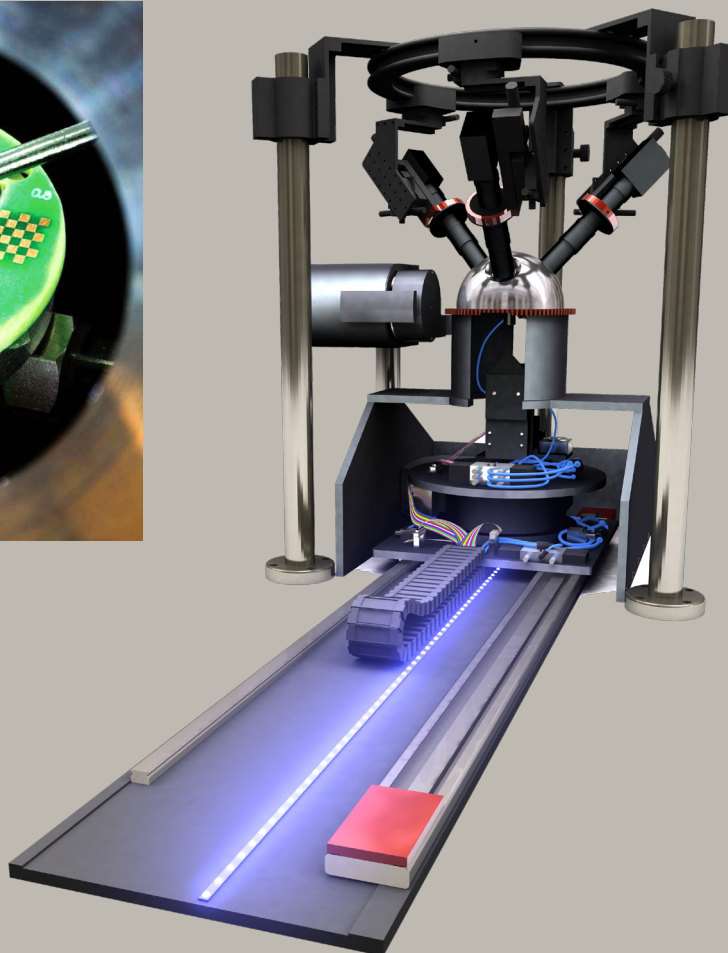
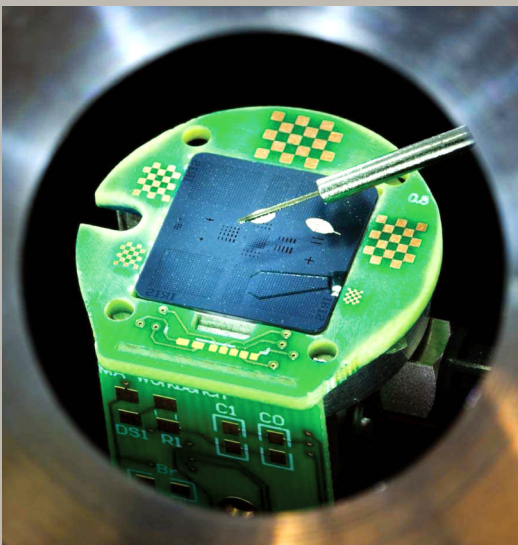


Martin Probst

Design of an Advanced Micro-Assembly System for the Assembly of Bio-Micro-Robots



DISS. ETH NO. 18137

Design of an Advanced Micro-Assembly System for the Assembly of Bio-Micro-Robots

A dissertation submitted to the

SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH

for the degree of

DOCTOR OF SCIENCES

presented by

Martin Probst

Dipl. Masch.-Ing. ETH
Born on December 27, 1977
Citizen of Basel-Stadt

accepted on the recommendation of

Prof. Bradley J. Nelson, examiner
Prof. Chris B. Rogers, co-examiner

2008

To Nathaly, Selma, Heinz, and Barbara,
for their support and encouragement.

Preface

This PhD thesis describes the work performed during an intensive time at the Institute of Robotics and Intelligent Systems of ETH Zurich. The primary objective of this project is to solve a number of complex real-world engineering problems in order to provide a revolutionary new set of tools. Since research and development went beyond the scope of the capabilities of a single ETH 200 % assignment the generous and enthusiastic assistance of many people is greatly appreciated. The interesting and challenging period with the result of a solid piece of hard- and software would not have been possible without their strong commitment. I am very thankful to all of them.

First of all, special thanks go to my supervisor and head of the laboratory PROF. BRADLEY J. NELSON for his academic support and guidance throughout the different phases of the project.

Special thanks also go to RUEDI BORER for the development of most electronic components and the numerous helpful discussions about all aspects of the micro-assembly project, and to the spirit of our lab, BRIGITTE GEISSMANN, for her generous support and cheerful nature.

I would like to express my gratitude to my best student CHRISTOPH HÜRZELER for his strong and valuable input in the field of software engineering. His coding and software design skills substantially contributed to the success of this project. I further acknowledge MARC ANDRÉ for his efforts in electronic component design.

I am also grateful for the services of MARTIN VOGT and his team from the workshop of the department of electrical engineering, which translated my ideas to actual hardware and helped in many aspects during the phase of design and manufacturing.

Many thanks also go to my co-supervisor PROF. CHRIS B. ROGERS for his support and the valuable input for the final version of this thesis.

Last but not least, I thank my partner NATHALY KRATZER, my parents SELMA UND HEINZ PROBST-GYSIN, and my sister BARBARA PROBST for their continuous support and encouragement.

Institute of Robotics and Intelligent Systems
ETH Zurich

Martin Probst
November 2008

Abstract

The field of bio-micro-robotics is gaining more and more importance, both in research and industrial applications. The reason for this trend is not least due to the significant increase in complexity of those miniature devices, that are capable of reaching in so far unreachable locations and thus solving a large variety of problems. A very promising field of application is medicine, where minimal- to non-invasive high precision surgical interventions performed by robotic devices more and more replace the classical type of procedures, and enable new ones. The production of micron-sized objects is mostly based on clean-room MEMS micro-fabrication processes and is thus limited to 2.5 dimensional shapes as well as certain compatible materials. However, the high degree of integration in bio-micro-robotic devices demands the manufacturing of full 3D hybrid MEMS devices of arbitrary shape. Micro-assembly is a modern and powerful technique that meets those requirements and reduces limits in creativity and technical complexity, thus offering a wide range of applications.

The main focus of this thesis is the development of an advanced and versatile micro-assembly station for the assembly of bio-micro-robotic components. It consists of a six degrees of freedom manipulator with sub-micrometer precision and accuracy, as well as a large cylindrical workspace of diameter 35.4 mm and height 25 mm. A simple gripper exchange mechanism offers high flexibility for various manipulation tasks and so far up to three individual gripper types are available. The workbench features a 15x15 mm MEMS fabricated silicon suction platform with through-hole features of various sizes, and can be driven out of the assembly area to a loading position in order to simplify part feeding and reduce damage to the grippers. An advanced vision and illumination system, featuring up to three CCD cameras, and a dome equipped with various independent high-power LEDs, provides crystal clear images of the assembly scene. The dextrous manipulator is controlled by two haptic input devices together with an advanced virtual reality environment that is continuously synchronized with the real hardware, thus providing an intuitive human interface to the micro-world.

The system also features a model based pose estimation module that is able to locate a given part in an unknown environment and estimate position and orientation with respect to a common base frame. The algorithm works with a combination of random sampling and linear refinement directly yielding a full 3D Euclidean error measure.

Zusammenfassung

Das Gebiet der Bio-Mikrorobotik gewinnt sowohl in der Forschung wie auch in industriellen Anwendungen mehr und mehr an Bedeutung. Der Grund dieser Entwicklung liegt nicht zuletzt an der bedeutenden Zunahme der Komplexität dieser Mini-Roboter, welche in bis anhin unerreichbare Gebiete vordringen und dadurch eine ganze Serie von Problemen lösen können. Ein vielversprechendes Anwendungsgebiet ist die Medizin, wo minimal- bis nicht-invasive Präzisionseingriffe mit Hilfe von Kleinstrobotern ausgeführt werden können, um schrittweise die klassischen Operationsmethoden zu ersetzen. Die Herstellung solcher Instrumente basiert auf MEMS Fabrikationstechniken, wodurch die Geometrie auf 2.5 Dimensionen beschränkt ist und nur untereinander kompatible Materialien verwendet werden können. Auf der anderen Seite verlangt die hohe Systemintegration der Bio-Mikroroboter die Fabrikation von 3D MEMS Hybriden ohne Einschränkungen. Das Zusammenbauen von Mikrokomponenten ist eine moderne und mächtige Methode, welche diese Anforderungen erfüllt und der Kreativität und technischen Komplexität viel Spielraum offen lässt.

Hauptthema dieser Arbeit ist die Entwicklung eines vielseitigen Mikro-Montage-Systems für den Zusammenbau von Mikrorobotern. Es besteht aus einem Manipulator mit sechs Freiheitsgraden, sowie einer Präzision und Genauigkeit im Sub-Mikrometer Bereich. Der zylindrische Arbeitsraum weist einen Durchmesser von 35.4 mm und eine Höhe von 25 mm auf. Ein einfacher Mechanismus zum raschen Wechseln von Greifern bietet höchste Flexibilität für zahlreiche Aufgaben, wobei aktuell drei verschiedene Greifer zur Verfügung stehen. Die Arbeitsplattform besteht aus einem 15x15 mm grossen Silizium-Quadrat mit durchgehenden Strukturen unterschiedlicher Grösse, welche mit Hilfe von Vakuum Bauteile ansaugen. Im weiteren kann die Vorrichtung in eine einfach zugängliche Ladeposition gefahren werden. Eine moderne Bildverarbeitungs- und Beleuchtungseinheit, bestehend aus maximal drei CCD Kameras und einem Beleuchtungsdom mit unabhängig regelbaren Hochleistungsleuchtdioden, liefert Bilder höchster Qualität. Die Steuerung des Manipulators geschieht über zwei haptische Eingabegeräte und einer virtuellen Umgebung, welche mit der Hardware synchronisiert ist. Dadurch wird ein intuitiver Zugang zur komplexen kinematischen Konfiguration gewährleistet. Im weiteren verfügt das System über eine Routine, welche ein gegebenes 3D Objekt im Raum erfassen und dessen Position und Orientierung gegenüber einem Basissystem bestimmen kann.

Contents

1	Introduction	1
1.1	Foundations of Micro-Robotics	1
1.2	Bio-Micro-Robotics	2
1.3	Objectives	5
1.4	Organization	5
2	Literature Review	7
2.1	Micro-Assembly System Concepts	7
2.1.1	Introduction	7
2.1.2	Serial Assembly	8
2.1.3	Parallel Assembly	13
2.1.4	Conclusions	15
2.2	Gripping and Handling Strategies	15
2.2.1	Introduction	15
2.2.2	Mechanical Gripping	17
2.2.3	Passive Gripping	18
2.2.4	Electrostatic Gripping	18
2.2.5	Cryogenic Gripping	19
2.2.6	Vacuum Gripping	19
2.2.7	Conclusions	19
2.3	Virtual Reality for Micro-Assembly	19
2.3.1	First Efforts	19
2.3.2	Recent Work	20
2.3.3	VR Support — A Case Study	21

2.3.4	Conclusions	21
2.4	Initial Pose Estimation (Localization)	21
2.4.1	Introduction	21
2.4.2	Conclusions	24
2.5	Conclusions	24
3	Micro-Assembly Station V2	25
3.1	Introduction	25
3.2	Design and Implementation	27
3.2.1	Design Requirements	27
3.2.2	Manipulator Unit	28
3.2.3	Grippers	31
3.2.4	Workbench	34
3.2.5	Calibration Units	36
3.2.6	Vision Units and Illumination	37
3.2.7	Glue Dispensing Unit	41
3.2.8	Pneumatic System	42
3.2.9	Mechanical Performance	43
3.3	Electronics and Control	43
3.3.1	Marvin Hardware Controller	43
3.3.2	Camera and Light Controller	45
3.3.3	Force Feedback Gripper Control	45
3.4	Software Components	48
3.4.1	Overview	48
3.4.2	ERASMUS: Hardware Controller	49
3.4.3	ADAGIA: Vision Server	50
3.4.4	COLLOQUIA: Client Workstation	53
3.4.5	Haptic Interfaces	56
3.5	Calibration and Initialization	57
3.6	Summary and Conclusions	57
4	Virtual Reality For Micro-Assembly	61
4.1	Motivation	61
4.2	Architecture	63
4.3	Scene Graph and Visualization	65

4.4	VR Control Center	66
4.5	Collision Avoidance	68
4.6	Event Recording/Replaying	69
4.7	Task Planning	71
4.8	Calibration and Synchronization	71
4.9	Performance	71
4.10	Summary and Conclusions	73
5	Model Based Pose Estimation	75
5.1	Introduction	75
5.2	Pose Estimation Algorithm	76
5.2.1	Strategies	76
5.2.2	The Algorithm	76
5.2.3	Pose Error	79
5.2.4	Center of Mass Error	81
5.3	Image Feature Extraction	82
5.3.1	Initial Experiments with Line Extraction	82
5.3.2	Microrobot Component Extraction	84
5.4	3D Model Preprocessing	85
5.4.1	Model Conversion	85
5.4.2	Additional Complexity Reduction	88
5.5	Correspondence Matching	89
5.5.1	Problem	89
5.5.2	Search Space Dimensions	89
5.6	Combining Pose Results	92
5.7	Implementation and Improvements	94
5.7.1	Ground Plane Constraints	94
5.7.2	Random Sampling Full Search Versus Random Sampling	94
5.7.3	Consistency	95
5.7.4	Multi-Threading	95
5.7.5	Tuning Parameters	96
5.8	Summary and Conclusions	96
6	Experimental Results	97
6.1	Introduction	97

6.2	Camera Calibration	97
6.3	Pose Estimation	99
6.3.1	Off-Line Experiments	99
6.3.2	Performance	101
6.4	Accuracy and Precision	104
6.5	Illumination	106
6.6	Assembly Experiments	108
6.6.1	Assembly of Bio-Micro-Robots	108
6.6.2	Assembly of Ultrasound Transceivers	112
6.7	Intercommunication Performance	116
7	Conclusions and Outlook	119
7.1	Contributions	119
7.2	Extensions and Improvements	120
7.2.1	Hardware Improvements	120
7.2.2	Software Improvements	124
7.3	Concluding Remarks	127
	References	129
A	Robot Kinematics	143
A.1	4 DOF Base Unit	143
A.1.1	Forward Kinematics	143
A.1.2	Inverse Kinematics	145
A.1.3	Jacobian	145
A.1.4	Camera Calibration Pattern Origins	146
A.2	2 DOF Gripper Unit	146
A.2.1	Forward Kinematics	147
A.2.2	Inverse Kinematics	148
A.2.3	Jacobian	148
A.3	6 DOF Manipulator	149
A.3.1	Forward Kinematics	149
A.3.2	Inverse Kinematics	150
A.3.3	Jacobian	150
A.4	Ring Units	151

B	Camera Calibration	153
B.1	The General Transformation Equation	153
B.2	Camera Calibration From Known Point Correspondences	153
B.3	Sampson Estimate of True Image and 3D Points	155
B.4	Maximum Likelihood Minimization	156
B.5	Decomposition of the Projection Matrix	158
B.6	Calibration Procedure	158
C	Computing the Rigid Body Motion	161
D	Computer Vision	165
D.1	Contours and Convex Hulls	165
D.2	OpenGL Matrices	166
D.3	Fields of View of EDMUND VZM-300i Microscopes	166
E	Geometric Transformations	169
E.1	General Transformations	169
E.2	Homogenous Coordinates	170
E.3	Euclidean Transformations	170
E.4	Affine Transformations	171
E.5	Projective Transformations	171
F	Geometric Entities	173
F.1	Plücker Lines	173
F.2	Hessian Normal Form	173
G	Software Modules	175
G.1	Overview	175
G.2	Specially Developed Libraries	176
G.3	Third Party Libraries	177
G.4	Dimensions	178
	Curriculum Vitae	179
	Index	183

List of Figures

1.1	Visions of micro-robotics.	2
1.2	PillCam from GIVEN IMAGING.	2
1.3	IRIS bio-micro-robot design and control setup.	3
1.4	Process steps in micro-fabrication.	4
1.5	MM3A Micromanipulator with MEMS force-feedback gripper.	4
1.6	Penetration of a mouse egg cell membrane.	5
1.7	The micro-assembly station V2 project website.	6
2.1	6 DOF micromanipulator and micro object handling system II.	9
2.2	Parallel cartesian mechanism and MiniPaR.	10
2.3	Mobile micro-manipulation platform MINIMAN III.	11
2.4	The main components of the minifactory.	12
2.5	Minifactory configuration for the assembly of microphones.	13
2.6	Parallel assembly of micro gearboxes and GaAs LEDs.	14
2.7	Forces at the micro-scale.	16
2.8	Microgripping strategies.	16
2.9	Interaction of localization and tracking.	22
3.1	Previous micro-assembly system prototypes.	25
3.2	Early assembly experiment.	26
3.3	Overview of the micro-assembly station V2.	28
3.4	Manipulator unit.	29
3.5	Kinematic configuration of the manipulator unit.	29
3.6	Exploded view of the 4 DOF base unit.	30
3.7	2 DOF gripper unit.	30

3.8	Main components of the arm assembly.	31
3.9	Main components of the wrist assembly.	31
3.10	Mechanical micro-gripper.	32
3.11	MEMS micro-gripper with force sensor.	33
3.12	Vacuum gripper.	33
3.13	Forces at the microscale.	34
3.14	Workbench assembly.	35
3.15	Work platform of the workbench.	35
3.16	Calibration mechanisms.	36
3.17	Laser calibration.	38
3.18	Ring-mounted pan-tilt units.	38
3.19	Position and naming scheme of ring units.	40
3.20	Vision dome.	41
3.21	Glue dispenser.	41
3.22	Reducing the quantity of liquid with a toothpick.	42
3.23	Pneumatic system setup.	43
3.24	Working area of the 6 DOF manipulator.	44
3.25	Camera-Light-Air controller board.	45
3.26	Physical arrangement of the camera-light-air controller unit.	46
3.27	Physical arrangement of the MEMS and mechanical gripper.	46
3.28	Linearity and dynamic range of the high voltage amplifier board.	47
3.29	High voltage DC-DC conversion circuit.	48
3.30	Computer infrastructure.	48
3.31	Software architecture of ERASMUS.	50
3.32	Software architecture of ADAGIA.	51
3.33	Software architecture of COLLOQUIA.	54
3.34	Main control panel widget.	55
3.35	Haptic input devices.	57
3.36	View of the assembly area through a camera hole.	58
3.37	Micro-assembly station V2.	59
4.1	Micro-assembly system setup without virtual reality support.	61
4.2	First generation assembly system.	62
4.3	Perspective view problem.	63

4.4	Assembly tree and part memberships.	63
4.5	Hard- and software setup with focus on the main workstation.	64
4.6	Internal structure of the virtual reality environment.	64
4.7	Scene graph data elements and simple example.	66
4.8	Model conversion process.	66
4.9	Virtual reality visualization widget.	67
4.10	Mapping motion commands to real axes.	67
4.11	Generic configuration of the VR control center.	68
4.12	Anatomy of a collision callback.	68
4.13	Collision detection system architecture.	69
4.14	Event recorder.	70
4.15	Event replaying mechanism.	70
4.16	Velocity commands and position measurements.	72
4.17	Closeup view of the virtual assembly area.	73
4.18	Virtual reality interface showing two live camera streams.	74
5.1	Pose estimation and tracking sequence.	76
5.2	Pose estimation strategies.	76
5.3	Visualization of point-line and point-plane constraints.	77
5.4	Pose error for point-line and point-plane constraints.	79
5.5	Euclidean distance measures.	80
5.6	3D pose error for correct and degenerate pose.	81
5.7	Center of mass error.	82
5.8	Line extraction from L-shape.	83
5.9	Line extraction from house.	83
5.10	Outer contour, convex hull and fit ellipse.	86
5.11	Microrobot extraction for different illumination.	86
5.12	3D model edge classification.	88
5.13	Fit-matrix representation of a correspondence space.	90
5.14	Pose estimation routine.	91
5.15	Bit-string representation of a correspondence set.	92
5.16	Hamming-distance-2 neighborhood of a given bit-string.	92
5.17	Threshold overlapping problem.	94
5.18	Threading scheme for multiple cameras.	95

6.1	Calibration patterns of the workbench.	98
6.2	Sample set of three camera calibration patterns.	98
6.3	Reprojection of calibration grid and coordinate frame.	99
6.4	Projection of the center of origin on the real scene I.	100
6.5	Projection of the center of origin on the real scene II.	101
6.6	Image processing and pose estimation result for L-shaped object.	102
6.7	Raw image and image processing result for a model house.	102
6.8	Pose estimation result for a model house.	102
6.9	Square micro-robot test case for computation time experiments.	103
6.10	Computation time comparison for point-plane constraints.	103
6.11	Computation time comparison for point-line constraints.	104
6.12	Comparison of computation time for pose estimation.	104
6.13	Precision measurement setup for the SUTTER MP-285.	106
6.14	Comparison of different illumination types.	107
6.15	Camera and light triggering scheme.	108
6.16	Initial assembly position.	109
6.17	Picking up and inserting an electroplated part.	110
6.18	Two-component bio-micro-robot.	110
6.19	Bio-micro-robot assembly — Part I.	111
6.20	Bio-micro-robot assembly — Part II.	113
6.21	Ultrasound transceiver developed at IRIS.	114
6.22	Operating principle of an ultrasound transceiver.	114
6.23	Ultrasonic transceiver assembly.	115
6.24	Mechanical test of an ultrasonic transceiver.	116
6.25	Magnetic test of an ultrasonic transceiver.	117
7.1	Second hand concepts.	121
7.2	Acrylic glass dust cover.	122
7.3	MEMS micro-grippers on manipulator and workbench.	123
7.4	Moving objects by pulling their handles.	124
7.5	Three basic relations of perceptual grouping.	126
7.6	Pose estimation using perceptual grouping.	126
7.7	Visual servoing control loop.	127
7.8	RB1: The most important haptic input device.	128

A.1	Coordinate frames for the D-H convention.	144
A.2	Origins of the calibration patterns.	147
A.3	Hierarchy of coordinate frames.	152
C.1	Illustration of a screw motion.	163
D.1	Convex and concave polygons and corresponding convex hulls.	165
D.2	Convex hull and convexity defect.	166
D.3	OpenGL [®] view volume.	167
D.4	Influence of camera tilt β on the field of view.	167
G.1	Location and interaction of specially developed libraries.	175

List of Tables

2.1	Comparison of serial and parallel micro-assembly.	15
3.1	Gripper comparison.	34
3.2	Calibration characteristics.	37
3.3	Viewing volumes for an EDMUND VZM-300i microscope.	39
3.4	Micro-assembly drive system parameters.	44
3.5	I03 output characteristics.	46
3.6	Configuration files for the hardware controller ERASMUS.	50
3.7	Configuration files for the vision server ADAGIA.	51
3.8	Configuration files for the client workstation COLLOQUIA.	56
5.1	Parameters of individual microrobot parts.	84
5.2	Summary of the microrobot extraction parameters.	87
5.3	Microrobot component parameters.	87
5.4	Relation types for correspondence sets.	90
5.5	Summary of the pose estimation parameters.	96
6.1	Pose estimation result for L-shaped object.	99
6.2	Pose estimation result for model house.	100
6.3	Pose estimation result for rectangular micro-robot part.	105
6.4	SUTTER MP-285 accuracy and precision results.	105
6.5	Accuracy and precision of the DC motor drives.	106
6.6	Network interface traffic between the three computing units.	117
A.1	Denavit-Hartenberg parameters for the base unit.	144
A.2	Denavit-Hartenberg parameters for the gripper unit.	147

xxii List of Tables

D.1 Fields of view and resolutions of an EDMUND VZM-300i microscope. 168

G.1 Total physical source lines of code (SLOC) for selected libraries. 178

Listings

3.1	Working with image processes.	52
4.1	Python TM module sample of the C++ replayer.	70

Nomenclature

Abbreviations & Acronymes

A/D	Analog to Digital (Conversion)
AFM	Atomic Force Microscope
ANSI	American National Standards Institute
ASIC	Application-Specific Integrated Circuit
CAD	Computer Aided Design
CCA	Canonical Correlation Analysis
CMD	Common Model Database
D/A	Digital to Analog (Conversion)
DCC	Digital Content Creation
DLT	Direct Linear Transformation
DOF	Degrees Of Freedom
EDM	Electro Discharge Machining
FOV	Field Of View
FPGA	Field Programmable Gate Array
GaAs	Gallium Arsenide
GUI	Graphical User Interface
IC	Integrated Circuit
ICA	Independent Component Analysis
ICP	Iterative Closest Point
LAN	Local Area Network
LDA	Linear Discriminant Analysis

LED	Light-Emitting Diode
LIGA	Lithographie ((X-ray) lithography), Galvanoformung (Electroplating), Abformung (Molding)
LP	Line-Plane
MEMS	Micro-Electro-Mechanical Systems
MOEMS	Micro-Opto-Electromechanical Systems
MOSFET	Metal Oxide Semiconductor Field-Effect Transistor
NA	Numerical Aperture
NMF	Non-negative Matrix Factorization
PCA	Principal Component Analysis
PID	Proportional Integral Derivative
PL	Point-Line
PP	Point-Plane
PWM	Pulse Width Modulation
RCM	Remote Center of Motion
SCARA	Selective Compliant Assembly Robot Arm or Selective Compliant Articulated Robot Arm
SEM	Scanning Electron Microscope
SLOC	Source Lines Of Code
SMA	Shape Memory Alloy
TEC	Thermo Electrical Cooler
TUI	Tangible User Interface
USAF	United States Air Force
UV	Ultra Violet (range of wavelength: 1–380 nm)
VR	Virtual Reality
WCS	World Coordinate System
XML	Extensible Markup Language

Greek letters

α	Camera pan movement	[rad]
β	Camera tilt movement	[rad]
ϵ	General variable for errors	
η	Attitude angle around x (roll)	[rad]
ω	Camera rotation around z	[rad]

θ	Attitude angle around z (yaw)	[rad]
ζ	Attitude angle around y (pitch)	[rad]

Roman letters

\mathbf{t}	Translation vector	
\mathcal{C}	Camera frame of reference	
\mathcal{I}	Image frame of reference	
\mathcal{L}	Calibration pattern frame of reference	
\mathcal{P}	Platform frame of reference	
\mathcal{T}	Tool frame of reference (TCP)	
\mathcal{V}	Vision frame of reference	
\mathcal{W}	World frame of reference (WCS)	
f	Frustration factor	
f_z	Camera movement along the optical axis (focus)	[m]
H	Homogenous transformation matrix	
P	Camera projection matrix from calibration pattern frame of reference \mathcal{L} to image plane \mathcal{I}	
P_W	Camera projection matrix from world frame of reference \mathcal{W} to image plane \mathcal{I}	
R	Rotation matrix	
T	Transformation matrix	
t	Time	[s]
x	First manipulator coordinate	[m]
y	Second manipulator coordinate	[m]
z	Third manipulator coordinate	[m]

Symbols

(a, b)	for intervals: $\{x \mid a < x < b\}$
$[a, b]$	for intervals: $\{x \mid a \leq x \leq b\}$
\varnothing_i	Inner diameter
\varnothing_o	Outer diameter
$\{a, b\}$	A set containing elements a and b

An expert is a person who has made all the mistakes that can be made
in a very narrow field.

Niels Bohr (1885 – 1962)

Introduction

This chapter gives an introduction to the field of micro-robotics and its potential both for research and industrial applications. Special focus is then put on one of its subsets called bio-micro-robotics together with a brief overview of current activities. Finally, objectives and organization of this thesis are presented.

1.1 Foundations of Micro-Robotics

The development of the first optical microscopes not less than four centuries ago opened up a sight on a completely different world. Even though mainly used for observation in the field of biology, people also started building miniature precision devices such as timepieces. The development of integrated circuit fabrication techniques roughly sixty years ago together with the growing interest in micro-biology eventually started pushing developments in the field of micro-robotics. The robotics research community started gaining interest only some twenty years ago and a significant amount of progress has since been achieved.

The broader public first came in touch with micro-robotics through the movie *Fantastic Voyage* (see Figure 1.1a) from 1966, in which a team of scientists together with their submarine are shrunk to microscopic size and injected into the human blood stream of a dying man, in order to save his life. This early vision of non-invasive surgery is completely ineffective since it neglects the scaling of physical effects (see Wautelet [171]). Micro-robotics cannot be reduced to the simple shrinking of macroscopic devices, but it involves developing a different intuition and its interdisciplinary nature requires knowledge in physics, materials science, as well as biology.

Micro-robotics is nowadays used in many areas, some of which include mechanisms (motors, gears, bearings: see Ehrfeld et al. [48]), hybrid components (MEMS, MOEMS, lasers), electrical systems (switches, connectors, lenses: see Yeh and Smith [177]) and biology (cell manipulation and puncturing, in-vitro fertilization: see Sun and Nelson [159]). However, most of these examples are situated in research environments and only a few devices are commercially available.

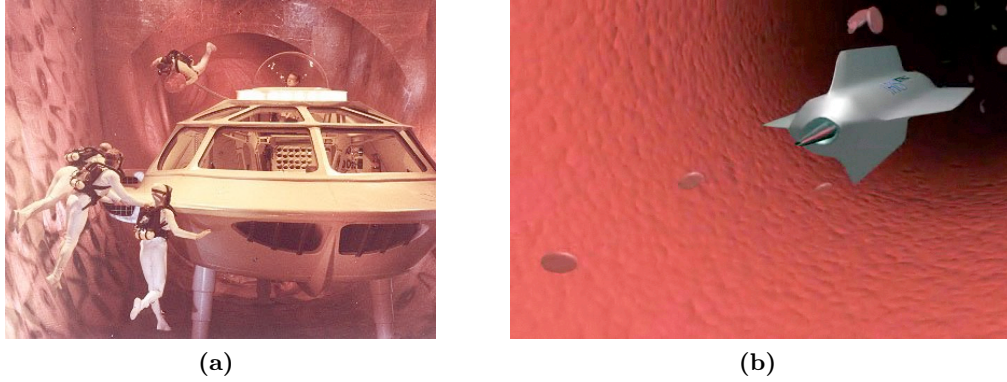


Fig. 1.1: Picture of the 1966 movie *Fantastic Voyage* (a) and a more recent and realistic concept developed at IRIS in 2004 showing a bio-micro-robot inside a simplified illustration of the human bloodstream (b).

1.2 Bio-Micro-Robotics

The application of microrobotics to biomedical systems similar to the above mentioned fictional idea is commonly referred to as bio-micro-robotics. Recent developments in sensor and actuator designs towards further miniaturization offer unprecedented possibilities, i.e., for non-invasive surgery and high-precision drug delivery. A prominent example of a commercial bio-micro-robot is the PillCam from GIVEN IMAGING (see Figure 1.2). This passive device is swallowed and travels through the body wirelessly transmitting images from the gastrointestinal tract at a frequency of around 2 Hz. The high integration of system components is remarkable but the overall size of 25x10 mm still quite large.

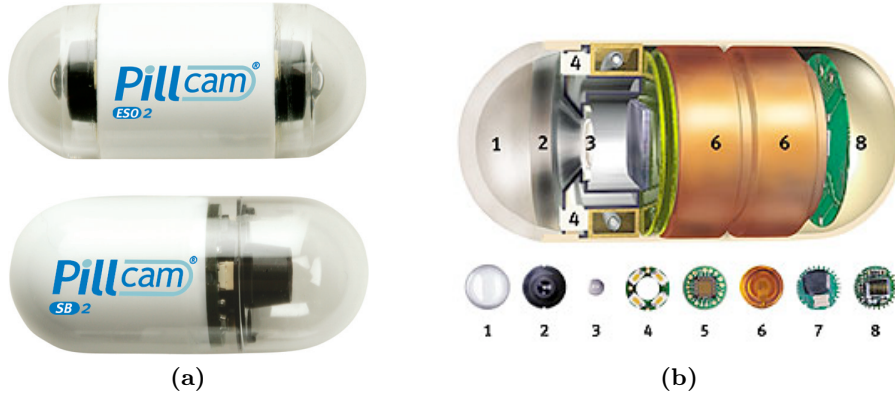


Fig. 1.2: PillCam from GIVEN IMAGING. ① optical dome, ② lens holder, ③ lens, ④ illuminating LEDs, ⑤ CMOS imager, ⑥ battery, ⑦ ASIC transmitter, ⑧ antenna.

The Institute of Robotics and Intelligent Systems introduced micro-robotics in the field of ophthalmologic surgery by developing a whole new concept of design and control strategies. The elliptic 3D bio-micro-robot shown in Figure 1.3a is injected into the vitreous humor of the eye and navigated by external magnetic fields with an example coil configuration shown in Figure 1.3b. The device can for example be equipped with a sensor for measuring oxygen concentration in order to locate inadequate oxygen supply (i.e., retinal hypoxia (Berkowitz and Wilson [16])) that is correlated with a number of major diseases such as glaucoma or retinal vein occlusions. Even though current research is focused on developments for eye surgery, the concept can also be used for medical analysis and treatment of other body parts, for example the human blood stream.

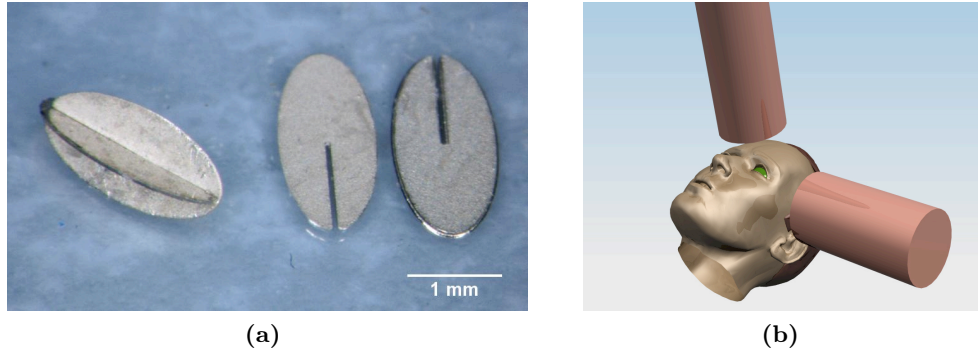


Fig. 1.3: *Microfabricated bio-micro-robot components (a) and a potential control environment based on a two-coil configuration (b).*

These examples of highly complex devices demand a high degree of integration of their building blocks. State of the art micro-fabrication processes are commonly used for manufacturing individual components, but they have their application limitations. The incompatibility of materials and/or manufacturing processes (e.g., CMOS, MOEMS, LIGA) confines design processes. Standard MEMS fabrication processes also limit designs to 2.5D which means that planar sketches on photomasks are linearly extruded along z without the possibility of creating undercuts (see Figure 1.4).

Micro-assembly can overcome these limitations and combine arbitrary components to complex hybrid 3D MEMS structures and thus enabling unprecedented capabilities. However, as already stated earlier, the vastly different physics at the microscale require special tools for part handling. Current micromanipulation devices, such as the one designed for an SEM chamber shown in Figure 1.5 or the cell handling station shown in Figure 1.6, require highly trained operators and a tremendous amount of experience, thus limiting the propagation of new ideas and technologies. In addition it also impairs industrial acceptance due to high costs and economic uncertainties.

The aggregation of complex hybrid 3D MEMS devices demands advanced micro-assembly systems which fulfil a set of requirements:

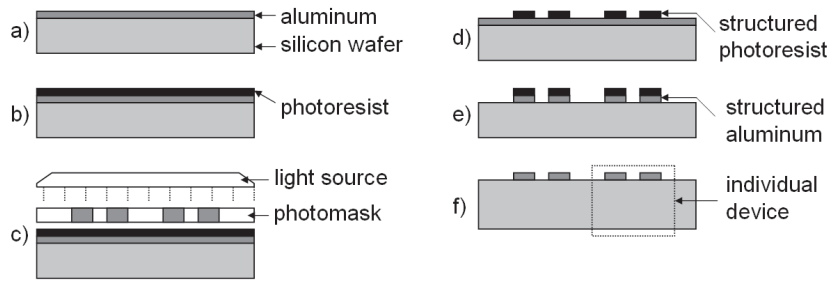


Fig. 1.4: Process steps in micro-fabrication showing the fabrication of aluminum electrodes. (a) Aluminum is deposited on a silicon wafer. (b) A thin layer of photoresist is deposited on the wafer and (c) exposed to UV light. (d) Immersion into developer causes the photoresist to selectively dissolve depending on its exposure to UV light. (e) The wafer is immersed into acid, etching the aluminum where it is not protected by photoresist. (f) The photoresist is stripped by a solvent. (Abbott et al. [1])

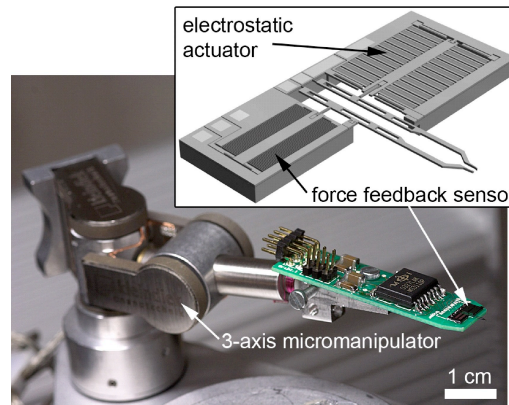


Fig. 1.5: MEMS force-feedback micro-gripper mounted on a KLEINDIEK MM3A three-axis piezo-driven micromanipulator.

- Tele-operated or automatic assembly of individual components to complex 3D devices with full 6 degrees of freedom.
- Access to the workbench for the easy population and removal of parts and assembled devices.
- Transparent kinematic configuration.
- Integration of advanced vision feedback.
- Flexible design for fast reconfiguration.
- Intuitive interface featuring visual feedback and haptic input devices.

A modern micro-assembly station should become standard laboratory equipment when working with microdevices and as easy to use as a regular microscope, so that only little training is required for efficient operation.

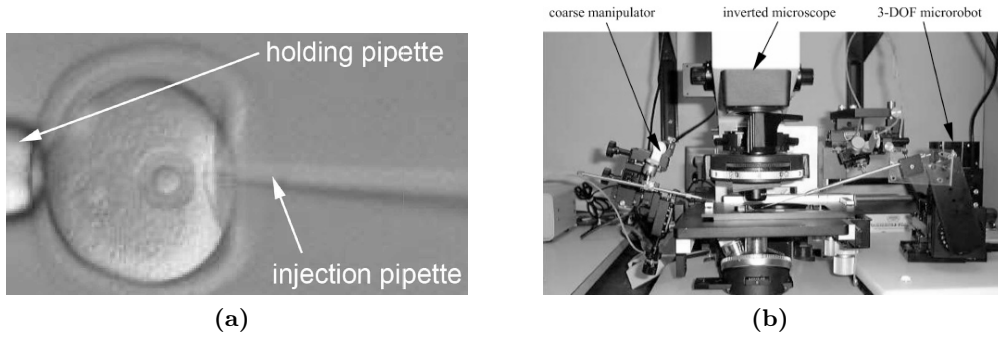


Fig. 1.6: Penetration of a mouse egg cell membrane (a) with an advanced micromanipulation station (b) (Sun and Nelson [159]).

1.3 Objectives

This thesis is about the design and implementation of a 6 DOF micro-assembly station for the assembly of bio-micro-robots. The primary goal is to develop a machine that can assemble devices unrivaled to the standards in micro-manipulation. A flexible modular design paired with an appealing user interface should provide an open gateway even for untrained operators. The focus of this development is in the field of human interaction and computer vision.

1.4 Organization

This report is a comprehensive summary of the research performed within the scope of this PhD thesis. Apart from pure scientific aspects it contains a large variety of applied engineering solutions that seem to be relevant for continuing development. Chapter 2 starts off with an extensive literature survey in the fields of micro-assembly related aspects as well as initial pose estimation techniques. The micro-assembly part contains an overview of existing micro-assembly systems and different gripping strategies as well as a brief excursion to developments of virtual reality support systems for micro-assembly. The description of the actual micro-assembly station can be found in Chapter 3 and is grouped into mechanical, electronics and essential software components. It follows a further elaboration of the virtual reality control software in Chapter 4, where many facets of the sophisticated control environment are explained. Chapter 5 deals with initial pose estimation, the problem of localizing and retrieving the pose of a known object in an unknown environment. Experimental results are presented in Chapter 6, and Chapter 7 concludes the work and dares a look into the future of this area of research.

The forward and inverse kinematic configurations of the micro-assembly system are derived in Appendix A. Appendix B deals with the complex mathematics of the camera calibration scheme and Appendix C with the screw notion for rigid body motion. Some aspects related to computer vision are summarized in Appendix D, and Appendices E

and F deal with geometric transformations and geometric entities, respectively. Appendix G finishes off with a brief overview of all custom and third-party software components involved in this project.

It is evident that space is not enough to include every single detail of all the work. However, more information is available for the interested reader on the micro-assembly project website <http://www.microassembly.ch> (see Figure 1.7). This page has a public section with lots of information about the system, research activities as well as a list of available student projects. The internal site is for registered users only and contains technical details in the domains of electronics, mechanics, and software, a logbook for developers as well as a link to a software bug tracking system.

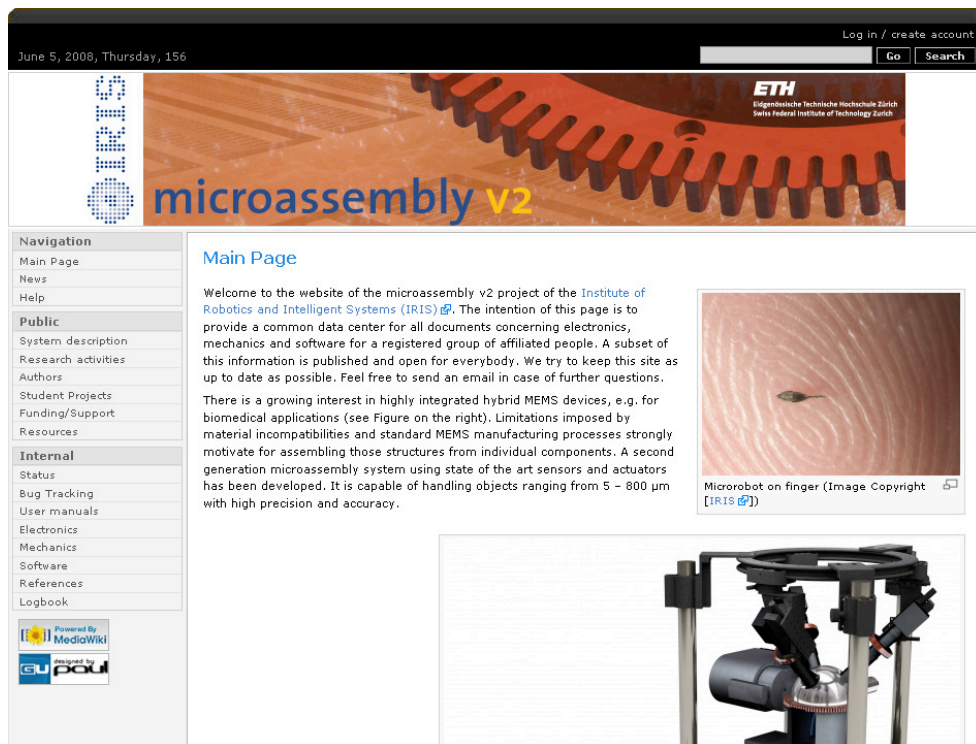


Fig. 1.7: The micro-assembly station V2 project website featuring public and private sections is located at <http://www.microassembly.ch>.

The source code of every single library developed within the scope of this project is thoroughly documented using Doxygen. The website directly links to this documentation.

Literature Review

The underlying chapter presents an extensive literature survey about micro-assembly systems, gripping and handling strategies, virtual reality for micro-assembly, and initial pose estimation.

2.1 Micro-Assembly System Concepts

2.1.1 Introduction

Micro-assembly systems can be divided into two sections, according to their principle of operation: serial assembly or parallel assembly (Böhringer et al. [21]). In serial assembly objects are assembled part by part following the traditional pick-and-place paradigm. Each element is picked from a part feeder by a robotic manipulator, translated, rotated, and then added to an intermediate position for re-grasping or to its final destination. It is a sequential process where one product after the other is assembled. Serial micro-assembly techniques require advanced sensory feedback of the assembly scene (vision feedback in most cases) as well as high precision positioning and part-handling tools. The automation of serial processes can vary between manual, tele-operated and fully automated assembly. For automated assembly systems, vision based control mechanisms are vital.

The throughput of serial micro-assembly is limited by the number of micro-manipulators in the array and their individual bandwidth. Efficient processing of parts from high-yield batch micro-fabrication processes for industrial applications thus relies on parallel assembly techniques. They can further be subdivided into deterministic approaches, based on a priori knowledge of the location of all parts, and stochastic approaches, where parts are distributed at unknown locations.

The following sections illustrate solutions and examples from both major assembly strategies. However, focus is set on serial assembly techniques since the system developed within the scope of this work belongs to this category, too. The structure is based on functional concepts and thus non-chronological.

2.1.2 Serial Assembly

The possibility of individually orienting micro-parts with N degrees of freedom generally allows serial assembly systems building more complex micro-structures than their parallel counterparts. Advanced sensory feedback and closed-loop control systems provide full control over the assembly sequence and unmatched flexibility. Serial assembly systems can be further subdivided into *master-slave systems*, *automatic assembly machines*, *assembly by micro-robots*, and *micro-factories*. They are illustrated in detail in the following.

Master-slave systems (Tele-Operation)

This is the basic type of a micro-assembly system and allows the manually controlled assembly of micro-structures using haptic input devices as well as visual and possibly force feedback. Commands from a human operator are scaled and directly translated to the hardware.

A tele-micro-surgical system which multi-modally transmits sensed information in the micro world to a surgeon, using visual, auditory, and tactile information, has been described by Mitsuishi et al. [107]. An experiment is presented in which a surgeon has successfully sutured an artificial blood vessel with a diameter of approximately 1 mm. In this particular case the operation room and surgery have been 700 km apart.

A prototype miniature robotic instrument consisting of a micro-gripper equipped with semiconductor strain-gauges as force sensors has been presented by Menciassi et al. [105]. Haptic interface force feedback that allows feeling pulsating fluids in micro-vessels has been demonstrated and finds applications in the area of biology, physiology and biomechanics.

Automatic assembly machines

Automating assembly processes is strongly demanded by industry and offers great potential in many fields. In order to reach a certain level of automation, continuous knowledge of the gripper position and orientation relative to the parts is required. Most systems rely on visual feedback since it has proven to be the most effective at those scales and can be used for visual servoing.

A 5 DOF robotic manipulator with $\pm 1 \mu\text{m}$ repeatability has been described by Dechev et al. [40, 41, 42]. The system is fitted with a compliant, passive micro-gripper that is able to grasp micro-parts, reorient them and assemble them to three-dimensional micro-structures using snap-lock joints. The same authors have recently presented a new version with 6 DOF (Dechev et al. [43]). It consists of an $xyz\alpha$ -stage as a base, and a robotic arm providing the two remaining degrees of freedom β and γ (see Figure 2.1a). The system is characterized by a single microscope providing an orthographic view of the working area, large ranges for all rotational axes, and an end effector repeatability of $\pm 2 \mu\text{m}$.

The development of a workcell for 3D wafer-level assembly of large numbers of micro-machined thin metal parts into holes of silicon wafers has been shown by Yang et al.

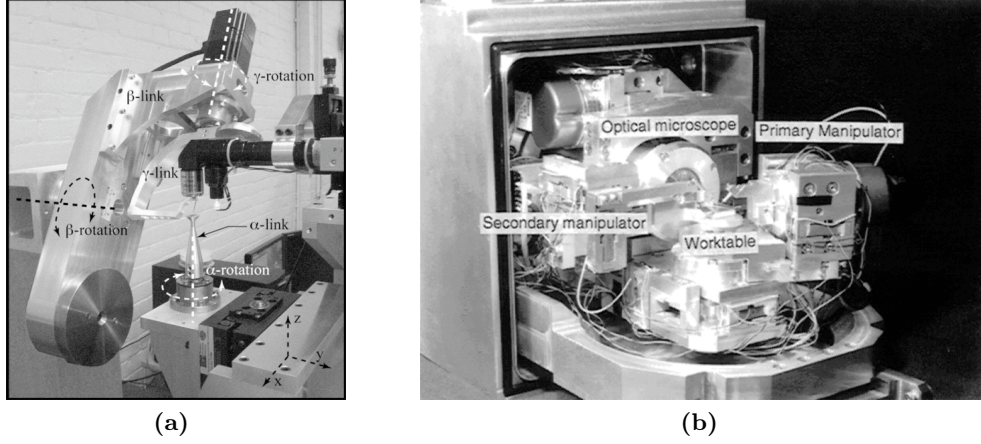


Fig. 2.1: (a) 6 DOF robotic micromanipulator (Dechev et al. [43]) and (b) micro object handling system II (Koyano and Sato [85]).

[175, 176]. It consists of a high precision 4 DOF micro-manipulator mounted on a 4 DOF course stage, as well as a multi-view imaging system. One of the primary goals of this project has been to demonstrate the feasibility of micro-assembly for combining incompatible materials and manufacturing complex 3D hybrid MEMS structures.

Another micro-object handling system designed for the use in the vacuum chamber of an SEM has been presented by Koyano and Sato [85]. It consists of two microscopes (one SEM and one regular optical microscope), two manipulators, and a worktable (see Figure 2.1b). The primary manipulator is equipped with two rotational and three highly precise translational degrees of freedom. The second manipulator is utilized to fix objects on the worktable that itself contains the last rotational DOF. All rotational axes are aligned around the center of the worktable, thus preventing the manipulation scene from leaving the fields of view of the microscopes during rotation.

The only commercially available micro-assembly system at present is the MEMblerTM from ZYVEX CORPORATION (Sarkar [135]). It is a closed-loop, automated 5 DOF robotic pick-and-place system with 25 nm positional accuracy used to assemble micro-components with minimum feature sizes better than 500 nm.

FRAUNHOFER IPT has developed a modular assembly system that offers 6 DOF with a total number of 7 axes (Weck and Peschke [172]). It combines individual, commercially available translational stages in a cartesian arrangement. The repeatability of each axis is in the order of $\pm 0.5 \mu\text{m}$ and $\pm 0.001^\circ$, respectively. Additionally, this system features a fully automatic end effector changing mechanism.

Another interesting research project runs under the name of MiniPaR (Bruzzone and Molino [24]), with the goal of developing a robot for micro-manipulation tasks and applying the peculiar properties of parallel robotics to it. In a parallel robot or parallel kinematics machine, the end-platform is carried in parallel by several elementary kinematic chains (legs), usually equipped with only one actuator. Parallel cartesian robots are

a special class of parallel kinematic machines. In these robots, the end-platform is carried in parallel by three legs. Each leg is a planar serial mechanism that leaves three DOF in its plane (see Figure 2.2a).

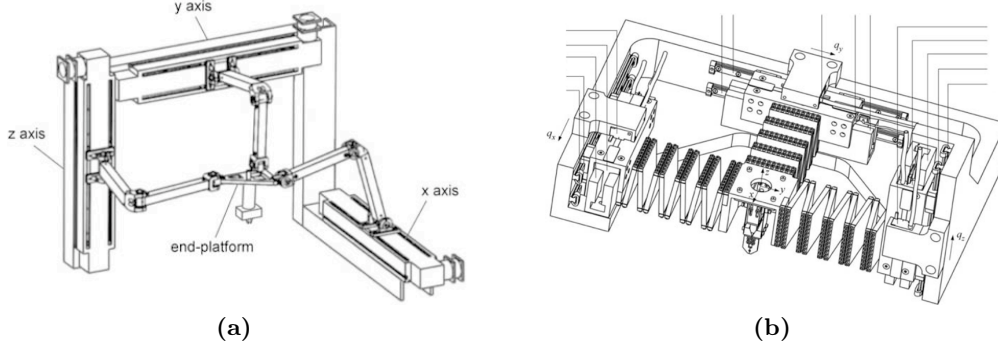


Fig. 2.2: Example of a parallel cartesian mechanism (a) and MiniPaR prototype (b).

The MiniPaR prototype configuration as seen in Figure 2.2b is a parallel cartesian robot with three translational degrees of freedom. The working platform is linked to three linear stages by flexure joints. Each of these flexure joints has a high stiffness in one cartesian direction, and is flexible in the other two directions. The robot design is very compact and features a relatively large workspace at good accuracy. However, the effects of unwanted rotations of flexure joints and hysteresis of the superelastic material are not negligible and must be properly compensated in order to fully exploit the potential performance of the robot.

A different focus on the environmental influences on micro-assembly processes as well as the construction of a controlled climate system can be found in Zhou et al. [186]. The assembly system has 3 DOF and is installed inside an environmental chamber that allows controlling temperature and humidity. Two gripper types for different objects sizes can be mounted in the system together with a micro droplet glue dispenser, and three microscopes are used to observe the scene from different perspectives. The system is operated in manual mode, tele-operated mode with a joystick or automatic mode based on off-line programming data.

Assembly by micro-robots

The manipulation accuracy of conventional macro-scale robots is limited by influences such as fabrication defects, friction, thermal expansion, or computational errors. Due to the high precision requirements these effects are not negligible in micro-assembly. In addition, macro-scale robots also suffer from mechanical wear and need periodic maintenance and calibration, which makes them expensive (Kratochvil et al. [87]). As an alternative micro-robot-based micro-assembly systems are proposed that consist of small robots with both transportation and manipulation skills. The devices can mostly be equipped with different types of tools for performing a large range of micro-manipulation tasks, such as

grasping, transportation, positioning and special processing steps like cutting, scratching and bonding (Koyano and Sato [85]). The large workspace, team work capabilities, as well as simple robot exchange offers great flexibility. Advanced control mechanisms, high-precision requirements, and inter-robot communication are key design issues for this type of assembly system. However, the main limitation of a micro-robot based assembly system is the insufficient energy autonomy of the micro-robots (Hollis et al. [72]).

The development of direct driven mobile robots only a few cm^3 in size has been described by Schmoeckel and Woern [138] for the use in a larger micro-manipulation station described in Quaid and Hollis [124]. These devices can perform high precision manipulations with a resolution of up to 20 nm and move at speeds of up to 3 cm/s. A version that is suited for use inside the vacuum chamber of an SEM is called MINIMAN III and shown in Figure 2.3. Piezoelectric actuators provide translation and rotation in the xy -plane using stick-slip motion (see Breguet et al. [22]).



Fig. 2.3: *Miniature mobile robotic platform for high-precision micro-manipulation MINIMAN III* (Schmoeckel and Woern [138]).

Higher positioning precision is achieved using a sneak-mode, in which the robot moves due to subsequent micro-steps, but shifts on the spot only by bending its legs within a $6 \times 6 \mu m$ radius. The achievable resolution of this motion is approximately 20 nm resulting from the resolution of the D/A converters.

The manipulation unit is mounted on the platform and consists of a steel ball carrying the end effector (i.e., a pair of micro tweezers). The steel ball is driven by three piezoelectric legs, too, and uses the same motion principle as the platform itself.

MINIMAN III is tele-operated with a spacemouse and closed loop control is proposed for further enhancements. The sensor system consists of a local and a global part. The global sensor system uses the image of a CCD camera supervising the workspace of the robot. It is used for navigating it into the field of view of the microscope, which is the local sensor and monitors the actual micro-manipulation task.

Micro-factories

Recently, more research has focused on the overall requirements for the automation of micro assembly processes. A product may have one or more interfaces to its environment, such as electrical, optical, fluidic, or mechanical contacts. It may need to be assembled in a process involving several different materials and several inconsistent manufacturing processes. The reasons for the assembly of the product to be automated can be various, such as component dimensions being too small to be effectively handled by people, the need for extremely high alignment accuracies between components, contamination by human operators, and the inability of human operators to focus on the micro-assembly task for long periods due to exhaustion. Finally, there are clear economic benefits in terms of rapid market response if the automation can be accomplished in a flexible way.

The concept of a highly reconfigurable, tabletop-sized micro-assembly system, capable of automatically assembling series of micro-products is referred to as a *micro-factory*. Ideally, the *micro-factory* consists of a clean box with a pallet entrance and an exit slot. Within the box are a wide-range manipulator and/or a courier robot, a high precision narrow-range robot and an image processing measurement and control system.

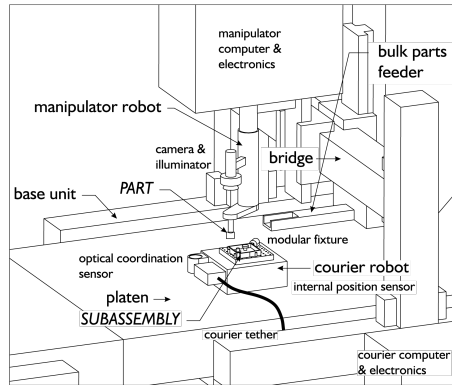


Fig. 2.4: The main components of the minifactory (Hollis and Gowdy [71], Hollis et al. [72]).

The Minifactory is a modular tabletop precision assembly system for the assembly of small mechatronic products and has been developed over the past years by Hollis and Gowdy [71], Hollis et al. [72], Quaid and Hollis [124]. It is essentially an implementation of the agile assembly architecture (AAA) proposed by Rizzi et al. [126].

The main components that form the Minifactory are shown in Figure 2.4. Multiple couriers equipped with planar linear motors glide on a factory floor which serves as the stator and a precision position reference. They carry sub-assemblies to multiple 4 DOF overhead manipulators which can access parts from nearby feeder units. The overhead manipulators have interchangeable modular end effectors and can be equipped with cameras and illumination devices. The strength of this concept has been shown by the automated assembly of small electret microphones and the corresponding machine configuration is shown in Figure 2.5.

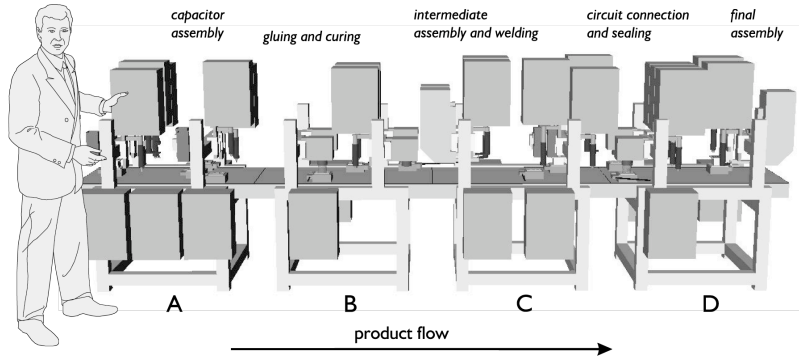


Fig. 2.5: *Minifactory configuration for the assembly of microphones* (Hollis and Gowdy [71], Hollis et al. [72]).

A similar idea called **MicroFab** for the assembly of optical lenses for endoscopes has been presented by Schuenemann et al. [140]. The MicroFab concept was developed to provide innovative medium sized companies with a customized concept for the commercially viable introduction of micro-fabrication technology. The micro-production process center consists of a number of tools, one or more handling units, and standardized input/output interfaces for wafers, substrates, and semi-finished components. The control is decentralized and the modular tools are capable of working separately as well as in linear or cluster-like arrangements.

2.1.3 Parallel Assembly

As stated above, serial assembly enables the fabrication of highly complex micro-structures at the cost of small series production. However, some (industrial) applications demand high throughput rates at relatively low individual object complexity and this is where parallel assembly strategies are applied.

Deterministic Parallel Micro-assembly

Deterministic parallel micro-assembly refers to direct transfer of micro-structures from wafer to wafer. The placement of the structures is predetermined by their layout on the donor wafer.

A parallel deterministic approach using a modified flip-chip technique has been demonstrated by Singh et al. [148]. Flip-chip bonding has been used to connect IC chips to printed circuits since the sixties. Surface micro-machined resonators and rotary micro-actuators are transferred in a batch process to interface electronics with the MEMS devices. Experiments show the successful transferring of twelve parts in one step with sub-micron precision. The transfer method is relatively simple and has a yield of nearly 100%. However, part and final assembly complexity is strongly limited by this process.

A way of micro-fabrication which utilizes features of the LIGA fabrication process (Becker et al. [14]) together with a sequence of geometrically complementary micro-structures

from different materials is presented in Ehrfeld et al. [48]. This new process is based on a plate-shaped magazine with embedded and spatially separated micro-structures whose positions and orientations are precisely fixed in the magazine. To assemble two series of complementary parts, their magazines are pushed together making sure that the part features exactly fit. The structures can then be pushed out of the magazine into their mounting positions on the corresponding devices. The capacity of this process has been demonstrated by the manufacture of micro gear systems shown in Figure 2.6a and is believed to essentially contribute to a cost-effective mass fabrication of hybrid microsystems in the future.

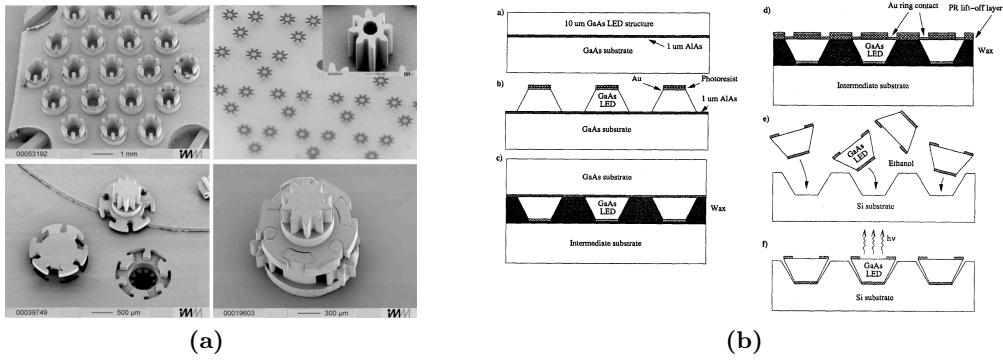


Fig. 2.6: *Parallel assembly of micro gearboxes (Ehrfeld et al. [48]) and fluidic self-assembly of GaAs LEDs (Yeh and Smith [177]).*

Stochastic Parallel Micro-assembly (Self-Assembly)

Self-assembly is a term used to describe processes in which a disordered system of pre-existing components forms an organized structure or pattern as a consequence of specific, local interactions among the components themselves without external direction. A self-assembly process can be described as an ensemble of parts or particles evolving towards a state of minimal potential energy.

The focus of these approaches lies on how to position and align an initially random collection of parts, so they can be further processed with deterministic parallel methods. Stochastic approaches use some type of force to agitate parts into known locations. The effects used as motive forces for stochastic self-assembly are fluidic or vibratory agitation and mating part shapes, manipulator surfaces (programmable force fields), vibratory agitation and electrostatic force fields, mating patterns of self-assembling monolayers, and nano-manipulation (inspired by chemical processes).

Motivated by the biological process in which antigens on viruses bind to antibodies through the matching of biochemicals, Yeh and Smith [177] have developed a process in which micro-structures are suspended in a carrier fluid prior to assembly. They are then dispensed over a host substrate with receptors. The carrier fluid drives the micro-

structures from receptor to receptor where they are captured similar to their biological analogon. The concept of fluidic self-assembly has been demonstrated by integrating GaAs optical and microwave electronic devices on Si wafers as illustrated in Figure 2.6b.

2.1.4 Conclusions

There is no unique direction in which current micro-assembly designs are heading. This is mainly due to the vastly different types of applications. However, for the case of serial assembly some sort of a trend towards multi-functional devices with 4 – 6 degrees of freedom and vision support can be observed. This development is in close agreement with advances in micro-technology with a frenetic demand for highly integrated devices. Some advances are also visible for the parallel assembly case with developments in MEMS fabrication processes. This approach is more visible in industry related projects where batch fabrication at low unit costs is a crucial factor. A short summary and comparison of the two major micro-assembly types can be found in Table 2.1.

	serial	parallel
object complexity limit	high	low
assembly tools	micromanipulator and -gripper	distributed manipulator, using e.g. electrostatic-, centrifugal-, capillary forces or vibration
sensory feedback requirement	vision, force-feedback	minimal
process control	direct, precise control feasible	indirect, precise control difficult
process flexibility	high	low
relative positioning accuracy	can be compensated by sensing	cannot be compensated
process efficiency	low – medium	high
error recovery	possible	difficult
system complexity	high	relatively low
system cost	high	relatively low

Table 2.1: *Comparison of serial and parallel micro-assembly* (Behera et al. [15]).

2.2 Gripping and Handling Strategies

2.2.1 Introduction

The key to successful and reliable manipulation of micro-parts lies in the selection and design of appropriate handling tools. As shown in Figure 2.7 the dominance of forces

changes dramatically the smaller the objects become. In order to account for these effects several gripping methods and types of grippers have been proposed and developed over the past years. This section compensates for the lack of a profound overview of those methods in literature.

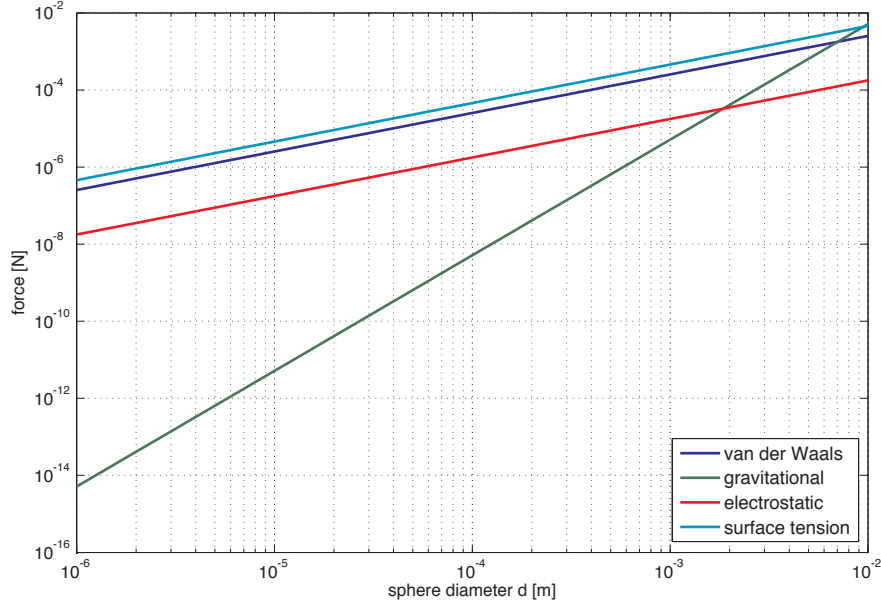


Fig. 2.7: Van der Waals, gravitational, electrostatic and surface tension forces for a plane-sphere model.

Gripping tools can be subdivided according to their principles of operation: mechanical, passive, adhesive, cryogenic, and vacuum (see Figure 2.8). The subsequent sections provide some examples for each category.

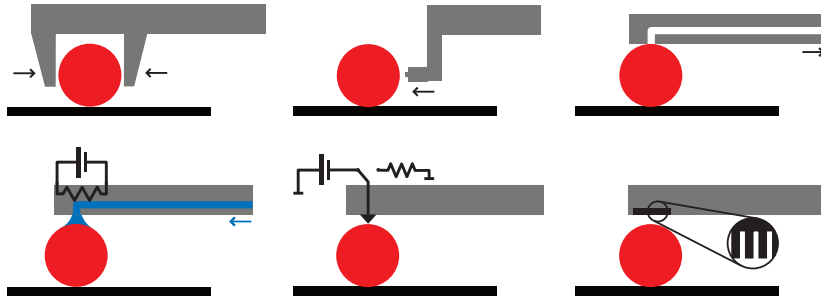


Fig. 2.8: Various types of strategies for gripping an object at the micro-scale (based on Arai et al. [9]).

2.2.2 Mechanical Gripping

A mechanical gripper consists of at least two main components: a gripping mechanism and a drive system. Some of the most popular drive systems are shape memory alloys SMA, piezoelectric, thermoelectric, and electrostatic actuators. In order to handle fragile objects like living cells, some devices are equipped with force sensors based on electrostatics, vision, and strain gauges (Kemper [79], Menciassi et al. [105], Park and Moon [119]). Releasing a grasped particle is one of the biggest challenges in micro-robotics and some mechanisms and strategies are described in Koyano and Sato [85].

The operating principle of a mechanical micro-gripper is the same as its counterpart from the macro-world: objects are grasped by closing the gap between two gripper arms. This simple basic concept has several advantages over other strategies like vacuum or adhesion force based tools. It can for example be used in virtually any surrounding condition, such as submerged in a liquid or in vacuum. In addition to that mechanical grippers can apply high forces.

A more exotic type of mechanical gripper is known as ortho-tweezers (Shimada et al. [144], Thompson and Fearing [161]) and consists of two probes mounted orthogonally to each other. Each finger has one degree of freedom perpendicular to its main axis and thus allows the rotation of parts around the z -axis.

Electrostatic Actuation

Electrostatic Actuation is realized by a series of comb drives whose total force is given by $F_e = n\epsilon \frac{tV^2}{d}$, where n is the number of capacitor elements, ϵ is the permittivity of the surrounding fluid (i.e. air), t is the thickness of the plates, d is the distance between the plates and V is the driving voltage. Last is in the order of 0–200 V yielding relatively high gripping forces compared to other methods. A nice example of electrostatic gripping has been presented by Beyeler et al. [18].

Thermal Actuation

Thermal Actuation is based on thermal expansion and contraction of materials and can be formulated as $\delta = \alpha \Delta T L$ with α being the thermal expansion coefficient, ΔT the temperature difference and L the initial length. The heat is usually induced by Ohmic resistance in a conducting actuator. A chevron-type actuator with mechanical amplification has been developed by Fraser et al. [59]. The advantage of this design is the large distance from the hot actuators (~ 500 K) to the gripper tips. In another design principle only one of the gripper arms is heated up whereas the other one stays “cold” (Chronis and Lee [34], Nguyen et al. [114]). Electrostatic gripping and electrothermal releasing of micro-parts has been demonstrated by Lee et al. [92], and Mayyas et al. [103] has analyzed static and dynamic performance of a thermal gripper.

Piezoelectric Actuation

Piezoelectric Actuation is a commonly used method since it allows ultra-precise movements at high resolution. The piezoelectric effect can be observed in a series of materials

(SiO₂, PZT, ZnO, AlN, PVDF, etc.) and is characterized by $\epsilon = dE$, where ϵ is the mechanical strain, d is the piezoelectric constant, and E the applied electrical field. Examples using this actuation type can be found in Agnus et al. [2], Park and Moon [119].

Shape Memory Alloy Actuation

Shape Memory Alloy Actuation is characterized by high force to volume ratios and thus very interesting for MEMS applications. Shape memory alloys regain their original geometry by themselves during heating. Micro-grippers based on SMA actuation are described in Houston et al. [75], Kohl et al. [83].

2.2.3 Passive Gripping

As opposed to mechanical grippers passive types are far less complex and thus a lot cheaper in manufacturing, and generally of higher robustness. In addition there is virtually no limit on the environment which they are used in. Since they usually do not need any support systems (i.e., power supplies, cooling, etc.) they can further be used for parallel assembly, which makes them attractive for industrial applications. The major drawback of passive grippers is the strict design requirement on the gripper-part-combination. An elegant solution is the use of snap-lock-joints and has been extensively described in Dechev et al. [41, 42], Tsui et al. [163]. Pushing parts to their final location can also be regarded as passive gripping and is shown in Zesch and Fearing [183].

2.2.4 Electrostatic Gripping

Electrostatic forces can be used to attract objects to specially designed gripper tips as shown in Lee et al. [92]. The small contact area between the object and the gripper simplifies the release process. Generally, this gripper type is of low complexity and does not require advanced fabrication techniques. The voltages needed for operation are in the range of 100–250 V which is why these models cannot be used in liquid environments.

Adhesive Gripping

Bark et al. [13] have developed a gripper with a simple design using low viscosity fluids. Picking up a part is done by dispensing a defined quantity of fluid to the gripper tip which forms a liquid bridge between the gripper and the micro-part. Slight misalignment of the two elements is automatically compensated by surface tension forces trying to reach an optimal energy level. The part can be released by evaporating, sucking back or blowing away the fluid by gas injection, or glue is used to simply hold the part at its final destination. The gripping force strongly depends on the surface condition, the gripping distance, as well as the type and volume of adhesive used.

2.2.5 Cryogenic Gripping

Cryogenic or freeze grippers operate by freezing a small amount of liquid (usually water) between gripper and part. The cooling energy is usually generated in a thermoelectrical cooler TEC, i.e., a Peltier element. Switching off the cooler or actively heating the tip releases the part as soon as the adhesion forces are overcome (CSEM [39], Lang et al. [90]). High gripping forces, no additional stress for the object, as well as the possibility to work in fluidic environments (Walle et al. [170]) are the big advantages of this method. It also allows cycle times of less than one second. However, freeze grippers are not suited for living cells and vacuum environments.

2.2.6 Vacuum Gripping

Vacuum grippers hold their objects by constant inward airflow in a pipette. Once the pipette has reached a certain vicinity of the object it sucks it to its tip. This in turn limits the density at which objects can be placed on a surface. Object surface roughness, pipette tip shape, inclination angle, vacuum pressure, etc. are only some of the parameters that influence the robustness of the gripping process. Generally, these systems yield low gripping forces and are of course not usable in a fluidic or vacuum environment. A detailed analysis as well as some experiments can be found in Zesch et al. [184].

2.2.7 Conclusions

There is no clear trend in the field of gripper developments due to the large variety of applications they are used for. In most cases releasing a part is the bigger challenge than actually picking it up. This notion becomes more and more relevant the smaller the parts are. For that reason, large efforts have been made in coating of materials in order to reduce attraction forces and simplify the release of parts from gripper tips. Additionally, a slight trend towards further miniaturization can also be observed by the increasing number of MEMS fabricated devices.

2.3 Virtual Reality for Micro-Assembly

2.3.1 First Efforts

The first reported use of virtual reality for micro-assembly can be found in Sulzmann and Jacot [154], Sulzmann et al. [155, 156, 157]. This work is based on a piezo-actuator driven mobile mini-robot equipped with a laser-machined micro-gripper. The unit is able to grip parts ranging from 20–200 μm . The workspace or the robot is observed by a camera and microscope, and vision algorithms extract position and orientation for updating the virtual reality environment. The approach also features virtual cameras to allow the user to set the viewpoint to arbitrary positions. However, real-time position update is not available.

Instead, the virtual world is used for designing assembly tasks which are then simulated. After a successful outcome, the trajectories are transferred to the robot and executed. Since the virtual model is not updated in real-time, safety measures are introduced to the robot controller that prevent the device from colliding.

Two 3 DOF micro-manipulators with vacuum grippers for objects larger than $10\mu\text{m}$ as well as a camera with a microscope observing the scene were introduced by Alex et al. [3]. The user moves parts by dragging a model in a virtual environment. The scene is then rendered to the same viewpoint as the one of the real camera by transforming 3D features of virtual objects to 2D screen coordinates and both virtual and real images are compared in image space. The resulting spatial differences are used for the visual servoing control loop.

2.3.2 Recent Work

The micro-assembly system of ENSIB-University of Orléans (Ammi et al. [6], Cassier et al. [29], Ferreira et al. [57], Hamdi and Ferreira [66]) consists of two high-precision micro-manipulators each of which is mounted on a coarse motion stage. A micro-conveyer system is used for part feeding and visual feedback provided by two microscopes. A sophisticated simulation package which takes into account predominant forces in the micro-scale is used together with a virtual reality environment for simulating assembly tasks. This powerful software package is shown to be very successful for millimeter-sized objects but reliability suffered due to the difficulty of accurately modeling physical effects at the micro-scale. Therefore, human operator supervision is still required, and its interaction with an automated control system is investigated in Ferreira [56]. In the same context Ammi and Ferreira [5] realized a manipulation system with an AFM-tip and a microscope. Image processing extracts all features in a scene including dust particles. This information is used as a base for rendering a virtual scene and helps the operator push parts along a collision-free trajectory. A head mounted display as well as a Phantom[®] input device are used for interaction with the operator. Again, physical models are used for the simulation of the interaction forces of all components involved in order to solve for an optimal trajectory.

A solution for a peg-in-hole assembly problem using virtual reality support is presented by Tan et al. [160]. The setup consists of a 3 DOF high precision manipulator that is mounted on a 6 DOF coarse motion stage. Apparently, vision and force feedback are only fed to a PID controller that gets its control input from the operator through a spacemouse, but not to the virtual environment. Instead, virtual reality is used for visualizing the deformation during the insertion process of the pegs.

A more recent publication by Cecil et al. [30] describes a simple setup with a 3 DOF base stage, a 1 DOF manipulator arm and a microscope. Computer vision algorithms implemented in Matlab are used for tracking gripper and micro-parts in the scene. Additional software components of this experimental setup include an assembly plan manager that reads user defined task descriptions, a collision detection engine, a virtual reality environment as well as a path planner. A motion tracked 3D mouse as well as stereoscopic goggles are used to fully integrate the operator into the environment.

2.3.3 VR Support — A Case Study

“Is virtual reality beneficial to micro-assembly systems? And if it is, what are the advantages and disadvantages. Which tasks can be simplified and which should not be touched? What are the boundary conditions that have to be taken into account?” Answers to those and other questions are presented in a case study by Zäh et al. [182]. Four test setups and around sixty participants which never had contact with a tele-operation system or virtual environment before were used for a series of experiments. The participants were divided in two groups, one half working with, the other half without, a specific feature to be investigated. The study shows that haptic feedback is a feature that drastically improves control over the assembly process and reduces the probability of damages to micro-parts. Acoustic feedback can help but not as much as haptic feedback. However, a gain in performance of about one third can be reached with the use of a virtual environment as opposed to a system with classical camera image feedback. Apparently, asynchronous feedback induced by system lag does not impair performance since the brain quickly learns to compensate for delays. This study is particularly interesting, because it is one of the few investigations of the benefits of visual, force and acoustic feedback systems in conjunction with micro-manipulation tasks. The results demonstrate that assembly performance is significantly higher, particularly if the information is consistently transmitted using various sensing modalities.

2.3.4 Conclusions

Even though some effort in using virtual reality for micro-assembly can be observed, it is all but widely used in reality. This is believed to be due to the strong focus on the technology of the assembly process itself and its juvenile phase that demands the knowledge of highly skilled researchers. No significant efforts in simplifying micro-manipulation tasks and bringing those systems closer to a larger community are visible in literature. The author believes that this will be a crucial development for the future in order to increase industry acceptance.

2.4 Initial Pose Estimation (Localization)

2.4.1 Introduction

The (automated) assembly of micro-components is strongly dependent on constant feedback. This is in strong contrast to the macro-world where pick-and-place operations are dominated by high relative gravitational forces and thus allow the “blind” execution of tasks in a repetitive manner. Different physics at the micro-scale with a changing balance of force create a challenging environment and demand constant observation of the parts to be manipulated. In practice, this means to extract position and orientation of the parts — also known as the pose (Grimson [64]) — at given time intervals t_i . Vision has proven to be the most effective type of sensor since it allows registering a relatively large working area.

The continuous observation of objects by means of visual feedback is generally referred to as *visual tracking*. It involves the detection of object features in the acquired images and the calculation of the relative or absolute pose given some information about the objects to be tracked. Several systems for visual guidance have been proposed (Allegro et al. [4], Bürkle et al. [26], Chen et al. [33], Fatikow et al. [50], Feddema et al. [53, 54], Koyano and Sato [85], Lee et al. [93], Sato et al. [136, 137], Shi and Tomasi [143], Sulzmann et al. [158], Vikramaditya and Nelson [166]), some of which offer higher degrees of automation than others. Most methods rely on template matching based on features selected by user input prior to tracking. More recent work uses 3D geometry from the CAD design process and is known as “model-based visual tracking” (Feddema et al. [53], Yeşin [178]).

Microscope optics closely approximate orthographic projection and therefore do not reflect changes in the image scale as objects move along the view axis (Nelson and Khosla [111]). Thus, a single microscope can only register three degrees of freedom (movement in xy in the image plane and rotation θ around z , the optical axis). Depth can be resolved by using depth from defocus methods (Allegro et al. [4], Sulzmann et al. [158], Vikramaditya and Nelson [166]) or structured illumination (Bürkle and Fatikow [25]).

The general “tracking assumption” states that position and orientation of an object are known at time t_{i-1} and only suffer small (definable) changes to the current timestep t_i . This neighborhood definition limits the search space and enables high refresh rates. Visual tracking systems therefore require human interaction in order to retrieve a rough initial pose as a starting condition so that the tracking assumption is fulfilled. On the other hand, initial pose estimation or localization methods are not limited in search space and thus include the whole sensory input (i.e., image) for pose retrieval. The combination of localization and tracking yields an efficient signal flow as shown in Figure 2.9.

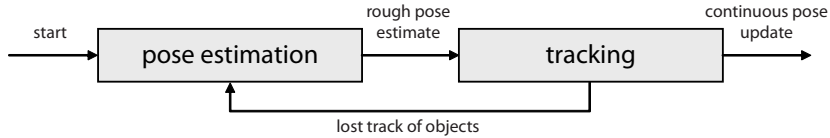


Fig. 2.9: Interaction of localization and tracking.

Initial pose estimation can also be looked at as an insurance for tracking. If the rigid body motions of an object become too large and violate the tracking assumption, localization is initiated again in order to regain control over objects in a scene.

Pose estimation has been studied since the beginning of computer vision and many solutions have been proposed. They can be roughly subdivided into appearance based and model based methods. Appearance based schemes encompass combined effects of shape, reflectance, pose and illumination that are acquired during a (automatic) learning phase. Instead of storing a large number of views representing most of these effects, subspace methods that compress and combine the amount of data are used (Bischof and Leonardis [19], Nayar et al. [110]). The basic idea is to represent an individual image as a linear combination of basis functions, such as

$$\mathbf{x}_i \approx \tilde{\mathbf{x}}_i = \sum_{j=0}^p q_j(\mathbf{x}_i) \mathbf{u}_j \quad (2.4.1)$$

where \mathbf{x}_i are the individual images, \mathbf{u}_j are orthonormal basis functions, and q_j are scalars used for scaling. One of the most prominent approach is the principal component analysis (PCA) where images are represented as points in an n -dimensional vector space. The PCA reduces the dimensionality of a given data set by transforming it to a new set of variables known as principal components. For visual recognition and pose estimation the coefficients of Equation 2.4.1 are calculated and the nearest point in the n -dimensional eigenspace is determined, which yields the pose of the object. Extensions to this basic concept as well as different schemes, such as linear discriminant analysis (LDA), canonical correlation analysis (CCA), independent component analysis (ICA), non-negative matrix factorization (NMF), and kernel methods have also been presented in literature (Chen et al. [32], Melzer et al. [104]).

Model based methods rely on specific geometric features of an object and can be partitioned into two separate problems (Rosenhahn et al. [132], Wunsch and Hirzinger [173]):

1. Correspondence problem: Determination of the mapping of 3D model elements and 2D image features.
2. Spatial fitting problem: Calculation of position \mathbf{t} and orientation R for every correspondence so that the spatial fit error is minimized.

Numerous solutions for the correspondence problem for different types of geometrical entities have been proposed. Point-based methods, that identify and locate feature points on an object from 2D image points, were introduced in the nineties and some pioneering work was done by Lowe [97, 98]. An extension to a fully projective formulation was later introduced by Araujo et al. [11]. Three non-collinear and non-coplanar or four non-collinear and coplanar points are the minimum required for retrieving a unique solution to the pose estimation problem (Goddard [63]). A new approach using compact constraint equations is presented in Rosenhahn et al. [131].

Methods using higher level primitives like lines and planes have also been of great interest over the past years since these entities are most often readily available from the CAD design process. Pose determination for line correspondences has been shown by Navab and Faugeras [109] and for line-plane correspondences by Chen [31]. The extension of the last method to point-line correspondences and the complete decoupling of the recovery of the rotational pose parameters from the translational ones has been presented by Phong et al. [122].

Free form features, such as NURBS surfaces, are the top level of geometrical entities. They are used for correspondence matching and their treatment can for example be found in Campbell and Flynn [27], Drummond and Cipolla [46].

Several algorithms for solving the second part of the pose problem, the spatial fitting, can be found in literature. A comparison of four approaches that deal with singular value decomposition, unit quaternion, dual quaternion and eigensystem computation can be

found in Lorusso et al. [95]. Rosenhahn [128] has introduced a gradient descent method and compares it with SVD and other approaches.

The combination of the iterative closest point (ICP) algorithm with a 2D correspondence operator in order to solve the registration problem has been presented by Wunsch and Hirzinger [173]. The proposed method relates image features to model data in 3D space and estimates the rigid body motion by minimizing an objective function.

An interesting approach was presented at an early stage by Lowe [96, 99, 100] based on the human vision system that is capable of filtering line groupings from a cluttered image. This process is known as perceptual grouping and dramatically reduces the search space. Three basic relations proximity, parallelism and collinearity are invariant to rotation, translation and scale over a wide range of viewpoints, and are used as key features for the subsequent matching process.

2.4.2 Conclusions

While the field of tracking is well covered in literature, less material is available for the initial pose estimation task. Many research and industrial applications deal well with the tracking assumption and do not mind to input additional knowledge from the user. In addition, the majority of projects focus on large scale objects that feature a different character. The use of model-based localization for the micro-world is yet to be fully established for a larger community.

2.5 Conclusions

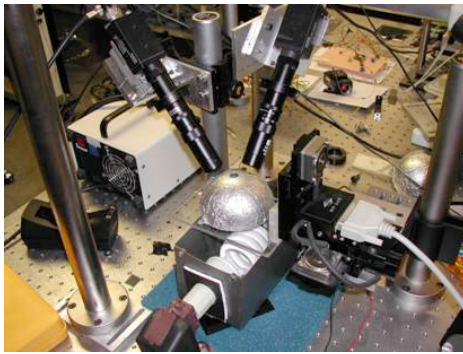
The primary goal of this chapter was to give an overview of existing micro-manipulation systems and their involved technologies gripping and handling, virtual reality, as well as localization. Demands on flexibility and robustness by this project have strongly voted for a serial system with classical mechanical grippers. However, none of the existing solutions meets the long list of criteria (as will be shown in Section 3.2.1) primarily required for the assisted assembly of sub-mm-sized bio-micro-robots as shown in Figure 1.3a. Thus, the development of a new micro-assembly station based on an existing configuration has been initiated.

Micro-Assembly Station V2

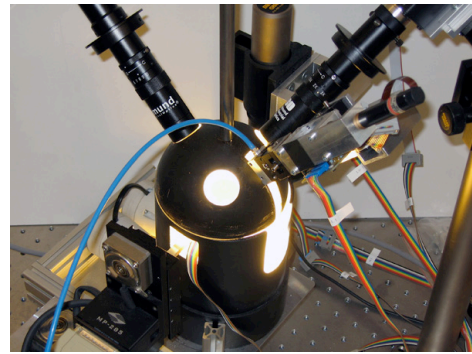
This chapter deals with the developments in the field of mechanics, electronics, and software within the scope of the micro-assembly system. The level of detail of explanations has been set to a moderate level in order to cope with both integrity and reader-friendliness. Methods for calibration and initialization of the system conclude this chapter.

3.1 Introduction

The Institute of Robotics and Intelligent Systems started working in the field of micro-assembly a few years ago (Yeşin [178]) and came up with an initial alpha version primarily for testing computer vision algorithms (see Figure 3.1a). This device featured a vision dome with multiple microscope views and a platform with three translational degrees of freedom. However, it lacked a manipulator with micro-gripping capabilities and any rotational degrees of freedom. A SUTTER stage allowed movements in xyz .



(a) *Alpha version*



(b) *First version*

Fig. 3.1: Previous micro-assembly system prototypes developed at the Institute of Robotics and Intelligent Systems.

These major drawbacks were compensated in the succeeding version with the development of an advanced manipulator configuration (see Figure 3.1b). The basic configuration with the xyz -stage was extended by a rotational degree of freedom θ around z as well as a separate kinematic chain providing $\eta\zeta$ -rotation and featuring a mechanically actuated micro-gripper. The machine was controlled using two standard PCs with external amplifiers and a rudimentary software solution. A number of successful assembly experiments were performed, one of which is shown in Figure 3.2.

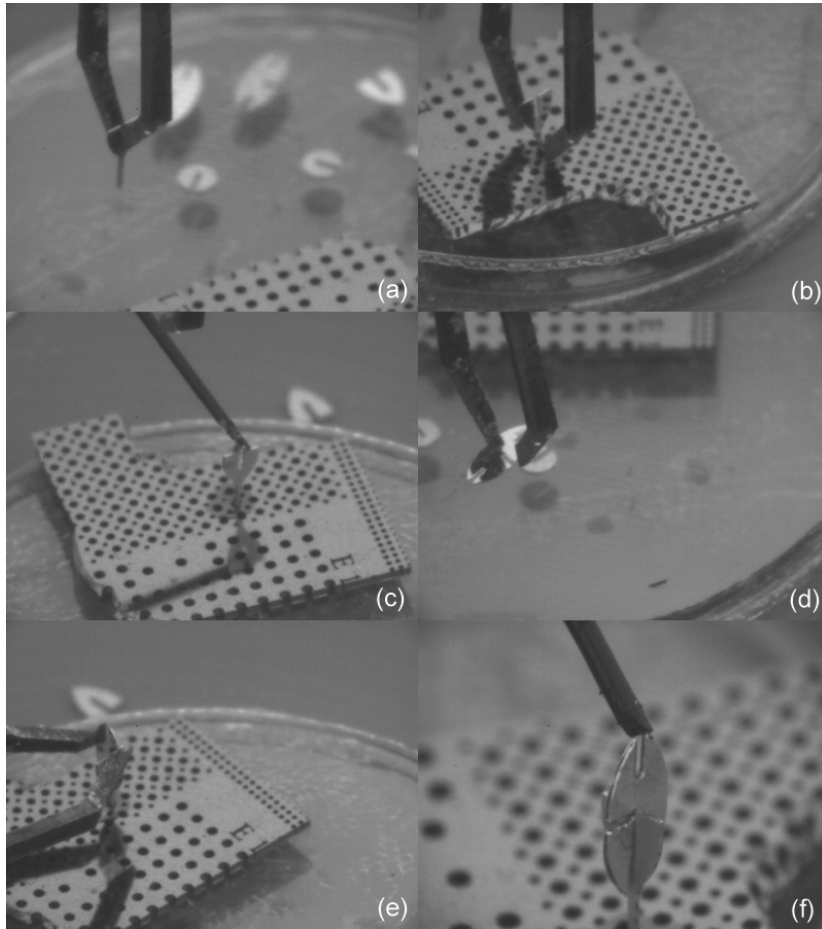


Fig. 3.2: *Early assembly experiment.*

The large number of experiments with this prototype revealed some issues and led to numerous ideas for further improvement.

Part feeding Loading micro-parts on the workbench required the manual lowering of the whole $xyz\theta$ -stage and most often the complete removal of the dome.

θ -stage Since the θ -axis was mounted at the end of the xyz -stage, it moved in xy and thus out of the intersection point of the $\eta\zeta$ -axes, which complicated the overall kinematic configuration.

Camera mounts Each camera was mounted on a single pole that had to be positioned around the dome on the vibration table's xy -grid. Precise circular alignment has proven to be very ineffective and painful.

Illumination The illumination of the workbench, which consisted of two regular energy saving lamps shining into the lower opening of the dome, has proven to be insufficient.

Robotic wrist The movement of the robotic wrist was abrupt and affected by large backlash.

Calibration The calibration of the individual axes has proven to be insufficient.

These and other issues finally led to the design described in the following sections. It starts off with a description of the mechanical hardware (Section 3.2), continues with the electronic components (Section 3.3) and finally explains the key software components, that have been specially designed within the scope of this project (Section 3.4).

3.2 Design and Implementation

The hardware section of the underlying micro-assembly system can be subdivided into two major components. One is the handling unit with all sensors and actuators for high precision movement. The other one is the vision unit containing adjustable cameras and illumination devices. In addition there are a number of auxiliary devices that are necessary for operation. This section describes the design considerations that led to the final design. Figure 3.3 shows a photo-realistic render of the micro-assembly station V2 as well as a zoom view of the workbench area with the gripper.

3.2.1 Design Requirements

The present system is based on a number of design requirements that have been specified at the beginning of the project. While some of them have been taken over from the previous system or intrinsically given by the available component (i.e., SUTTER stage), others have been defined in order to increase system performance. The key requirements are as follows:

- The system should be able to handle objects ranging from $50\text{ }\mu\text{m}$ – 5 mm .
- A large workspace in the order of $20\times 20\text{ mm}$ is required.
- The manipulator should allow movements in full 6 DOF.
- The kinematic chain(s) should follow the remote center of motion configuration.
- Full integration of computer vision, illumination and initial pose estimation capabilities is required.
- Easy access to the work platform has to be guaranteed.
- The system should feature an intuitive visual and haptic user interface.

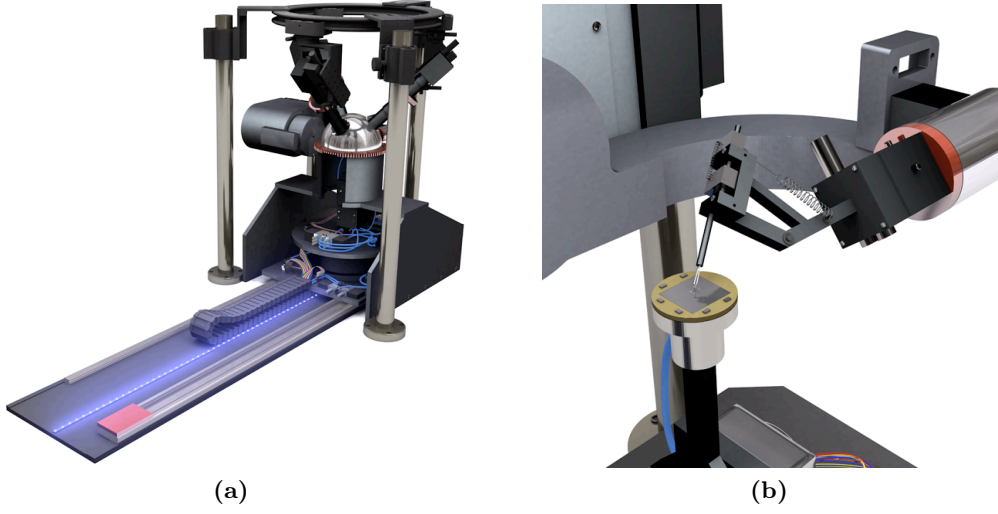


Fig. 3.3: Overview of the micro-assembly station V2 (a) and closeup of the gripper and workbench (b).

3.2.2 Manipulator Unit

The six degrees of freedom manipulator unit is the core component of the micro-assembly system. It consists of a four degrees of freedom base unit and a two degrees of freedom gripper unit as shown in Figure 3.4.

The strong dependency on visual feedback at the microscale requires that the view volume of the microscopes coincides with both the tip of the tool and the tool's center of motion TCM (see Sato et al. [136]). Since the TCM is defined by the configuration of the rotational axes and preferred to be stationary, all rotational axes have to intersect in one point, also referred to as the remote center of motion (RCM). The kinematic configuration in standard notation can be seen in Figure 3.5a.

The 4 DOF base unit consists of a large backlash-free high-precision rotation table providing rotation θ around the z -axis. Its fairly large diameter is reasoned in the fact that the weighty xyz -stage is eccentrically mounted on the output side on the top plate. In addition, the hollow construction with an inner diameter of 160 mm offers the integration of a rotary ring with a 2-channel pneumatic and a 36-channel electric feed-through, carrying pressurized air and vacuum as well as drive and other control signals, respectively (see Figure 3.6). This design allows infinite rotation around the z -axis and thus simpler kinematic calculations. Worries that the sliding contacts of the electrical feed-through would cause problems with digital signals (encoder feedback) could not be verified. The rotation table provides a theoretical mechanical resolution of 0.001° and a typical eccentricity of $1.4\text{ }\mu\text{m}$.

As already mentioned before the round plate on the output side carries the xyz -stage as well as pneumatic and vacuum valves. It also contains a laser diode whose beam is

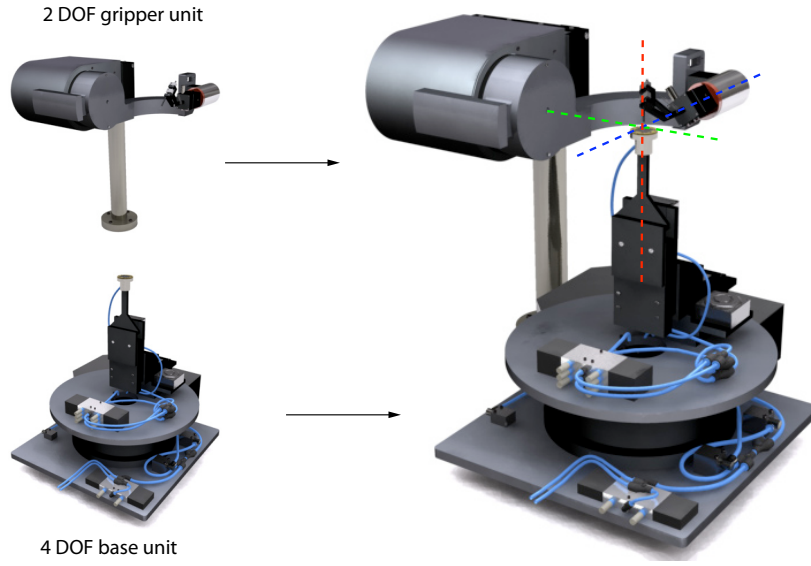


Fig. 3.4: The manipulator unit consists of a 4 DOF base and a 2 DOF gripper unit. The arrangement follows the remote center of motion (RCM) concept with all rotational axes meeting in one single point in space.

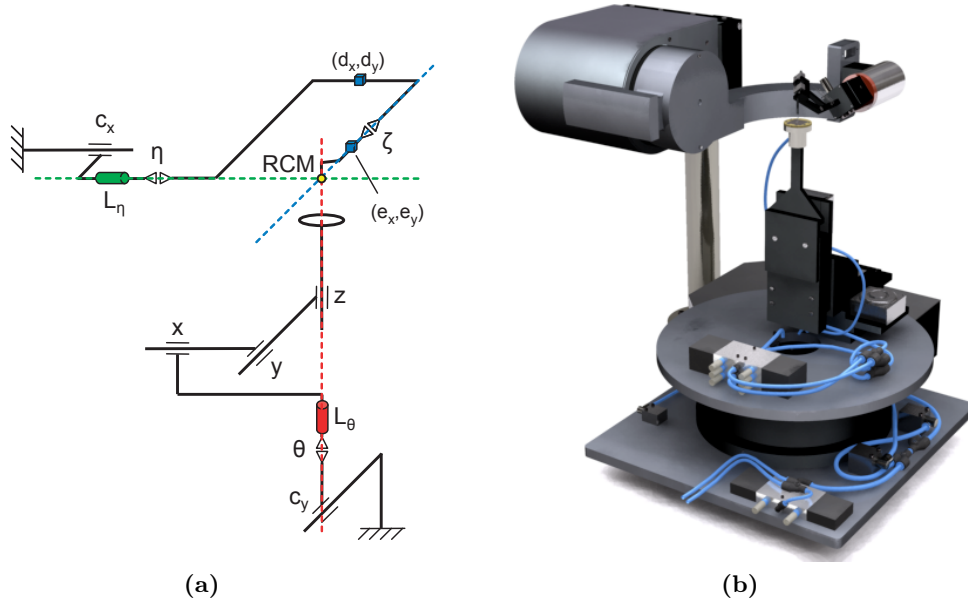


Fig. 3.5: Kinematic configuration of the manipulator unit.

coincident with the rotation axis used for calibration (see Section 3.2.5). Additional space allows mounting other components and thus offers flexibility for future developments.

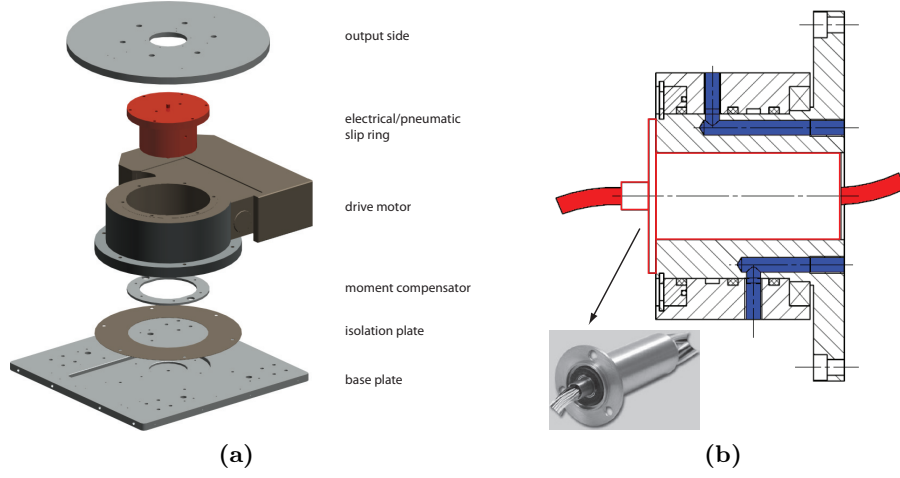


Fig. 3.6: Exploded view of the 4 DOF base unit (a) and the pneumatic and electrical multi-channel feedthrough (b).

The 2 DOF gripper unit provides the two remaining rotary degrees of movement η and ζ around axes x and y , respectively, and holds the end effector (micro-gripper). As shown in Figure 3.7 the unit consists of an arm that is arranged around the dome for design compactness. This arm is directly attached to a Harmonic Drive[®] driven by an electric motor and a drive belt. The backlash-free and quiet operation of the Harmonic Drive[®] as well as the high gear ratio of 560:1 allow movements at high resolution. Similar to the base unit (see Figure 3.6a) the gearbox also has a hollow shaft that contains two concentric tubes. The most inner one has a small diameter for a calibration laser beam, whereas the space between the outer and the inner one is used as a cable conduit. Electric wires as well as fibres for optical limit switches are all conducted inside special grooves of the curved arm. Figure 3.8 shows a simplified exploded view of the arm design.

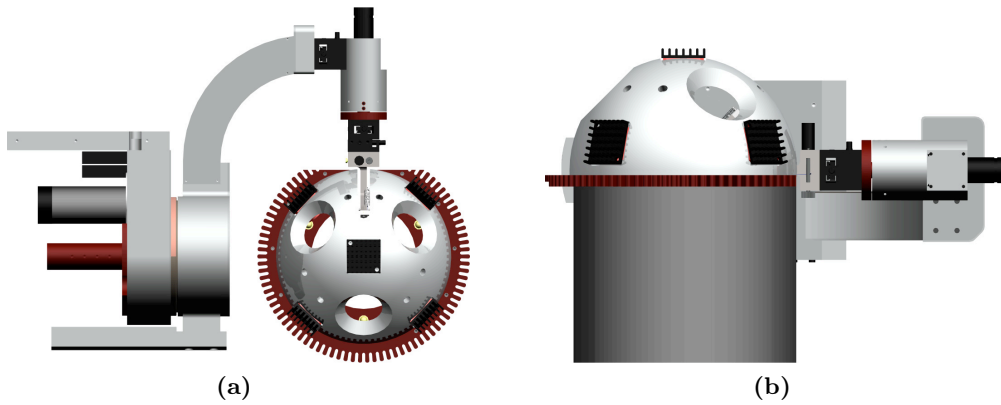


Fig. 3.7: 2 DOF gripper unit top (a) and side view (b).

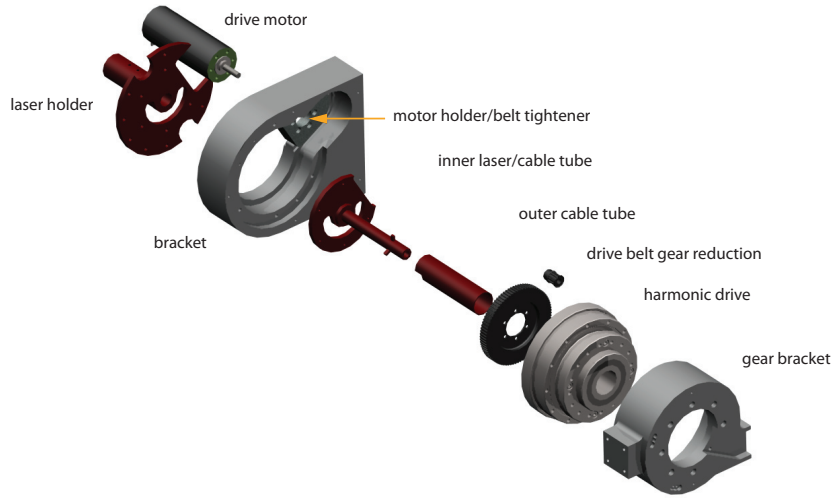


Fig. 3.8: *Main components of the arm assembly.*

The last rotational degree of freedom ζ is again realized with a miniature Harmonic Drive[®] combination that is attached to the very end of the arm unit. Two manual xy -stages mounted perpendicular to each other are used for alignment of the tool center point with the other rotation axes (see Section 3.2.5). See Figure 3.9 for an illustration of the situation.

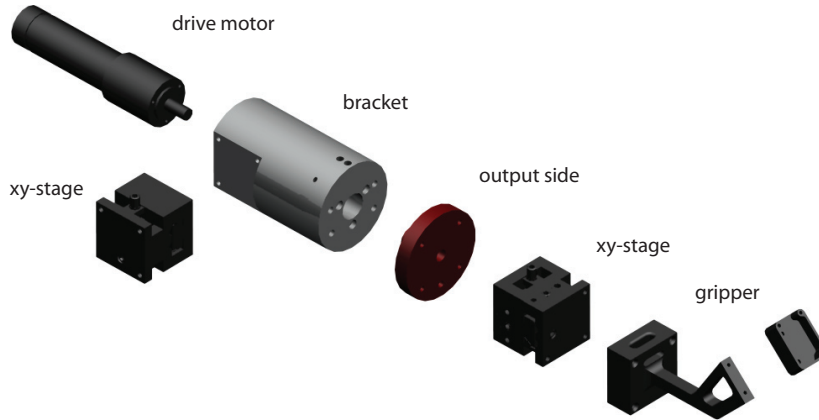


Fig. 3.9: *Main components of the wrist assembly.*

3.2.3 Grippers

The micro-assembly station V2 has a flexible tool exchanging mechanism that allows using different types of grippers depending on the demands of the application. Three grippers developed at this institute are briefly presented here.

Mechanical Micro-Gripper

The mechanical micro-gripper is basically a pair of tiny FEM optimized tweezers that are actuated by an enveloping sliding tube (see Figure 3.10). Due to limited space inside the dome the drive motor is externally mounted at the gripper flange. A tendon drive system transfers the linear motion to the sliding tube. The miniature tweezer is made out of spring steel, fabricated by micro-wire EDM, and can handle objects ranging up to $800\text{ }\mu\text{m}$ at a maximum gripping force of 72 mN . This gripper has proven to be very versatile for a large variety of applications and is extremely robust. Further details about this design can be found in Hess [68].

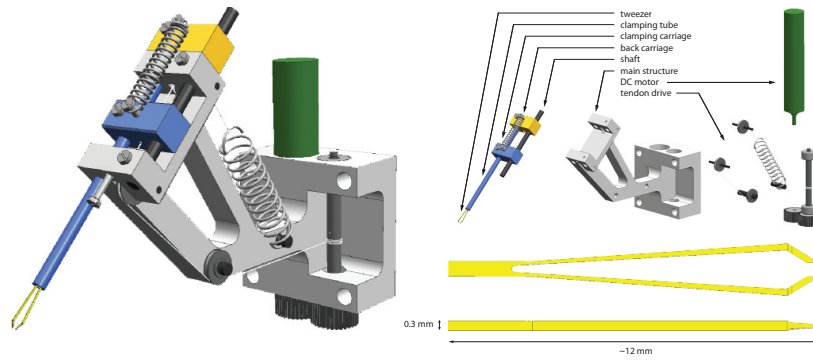


Fig. 3.10: Mechanical micro-gripper (left), its main components (top right) and the optimized tweezer (bottom right).

MEMS Force-Feedback Micro-Gripper

This micro-fabricated gripper has been developed at this institute and comes in different configurations. Their common actuation principle is based on electrostatic forces between numerous comb drives in an array (actuation voltage: $\sim 0 - 150\text{ V}$). The version used here has integrated force-feedback on the same device and can handle objects ranging from $5 - 200\text{ }\mu\text{m}$ with a maximum gripping force of 70 nN . Handling such small objects is certainly the big advantage of MEMS grippers. However, their silicon body is very fragile and makes the device less robust. Further details about the design of the gripper can be found in Beyeler et al. [18] and its use in the micro-assembly station V2 is shown in Section 3.3.3

Vacuum Micro-Gripper

As opposed to the tweezer-type mechanical grippers their vacuum counterparts hold parts by sucking them to a depression creating fixture. The big advantage is the low complexity of those devices since they do not have any moving parts. However, the main issue is

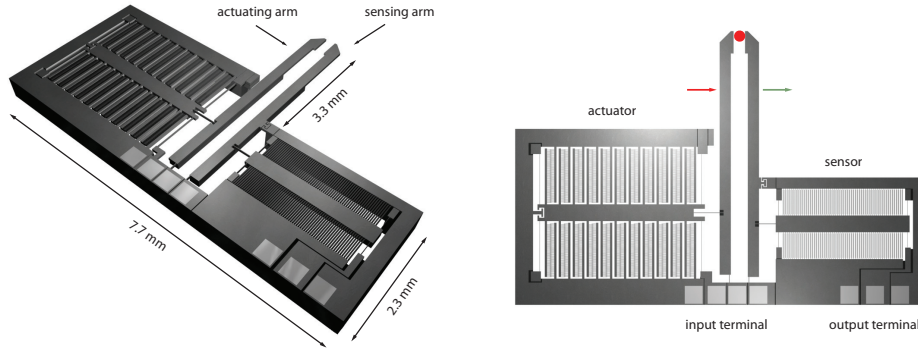


Fig. 3.11: MEMS micro-gripper with integrated force sensor (IRIS).

to control the orientation at which a part is aspirated and released. In addition, the contact surface on the part is not allowed to increase a certain roughness depending on the pressure of the vacuum.

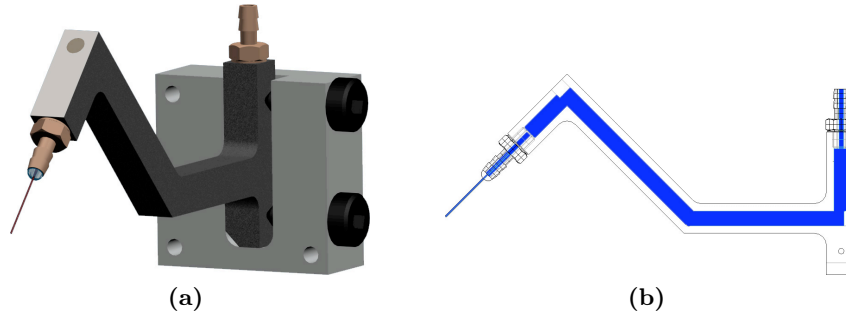


Fig. 3.12: Vacuum gripper assembly (a) and cross-sectional view of the flow path (b).

Table 3.1 shows a summary of the three gripper types previously discussed. The mechanical micro-gripper has proven to be the most effective for many assembly situations. For very small parts or biological cells the force-feedback MEMS gripper is the best solution. The vacuum gripper seems to be most unreliable and is therefore not considered in the subsequent explanations.

The mechanics of micromanipulation are strongly dependent on the ratio of forces acting on the parts. Figure 3.13 shows how the most important forces scale with object size. It is evident that gravity is negligible for objects with a diameter smaller than $100\text{ }\mu\text{m}$ where surface tension and van der Waals forces start dominating. This asks for strategies not only for gripping, but also for releasing parts from grippers. Even though special coatings allow reducing surface tension and van der Waals effects, other measures are required for robust operation under various operation conditions. The workbench plays an important role as is shown next.

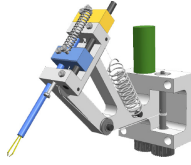

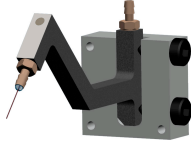
	mechanical	MEMS	vacuum
			
actuation principle	mechanical (DC motor)	capacitive	vacuum
min range [μm]	200	5	$f(p_{min})$
max range [μm]	800	200	$f(p_{max})$
force feedback	no	yes	no

Table 3.1: Comparison of three possible gripper types designed for the micro-assembly station V2. Parameters p_{min} and p_{max} are lower and upper air pressure levels for the vacuum, respectively.

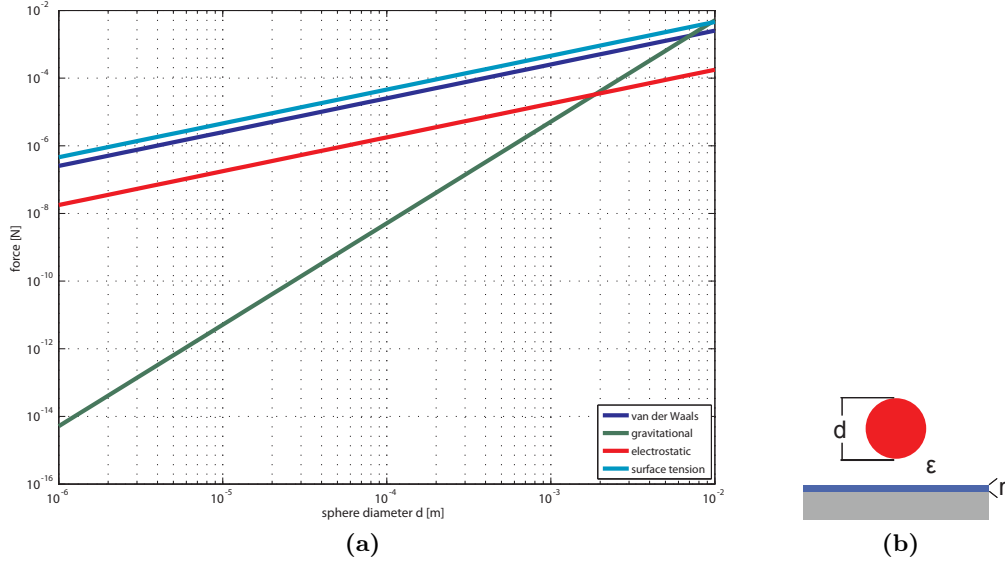


Fig. 3.13: Van der Waals, gravitational, electrostatic and surface tension forces (a) for a plane-sphere model (b), where d is the sphere diameter, ε is the dielectric coefficient, and r is the surface roughness of the plane (based on Arai et al. [9]).

3.2.4 Workbench

The workbench of the micro-assembly station is mounted on top of the xyz -stage of the base unit (see Figure 3.14). It consists of a microfabricated silicon platform surrounded by a printed circuit board which contains patterns for calibration (see Figure 6.1) and is designed for mounting a second force-feedback micro-gripper including readout electronics. Its support structure with an enclosed air-channel creates a low pressure environment just below the through-hole multi-purpose pattern of the platform (see Figures 3.14c and

3.15) and thus a constant air inflow. This design simplifies the release of micro-parts from the gripper, allows the dexterous handling of objects, and keeps them in place.

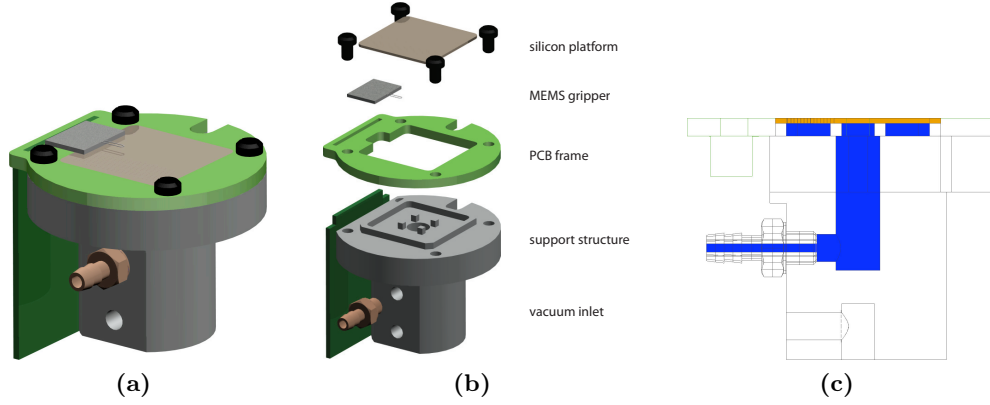


Fig. 3.14: Workbench assembly (a), individual components (b), and cross-sectional view of the vacuum flow path (c).

The upper and lower third of the pattern consist of hole arrays of the same size designed to keep flat parts in position. A large number of slots of different sizes for sticking in parts dominate the central area. For easier handling and flexibility, the vacuum can be turned on and off and its magnitude adjusted. The perforated areas feature a material thickness of $50\mu\text{m}$ whereas the adjacent areas are $500\mu\text{m}$ strong.

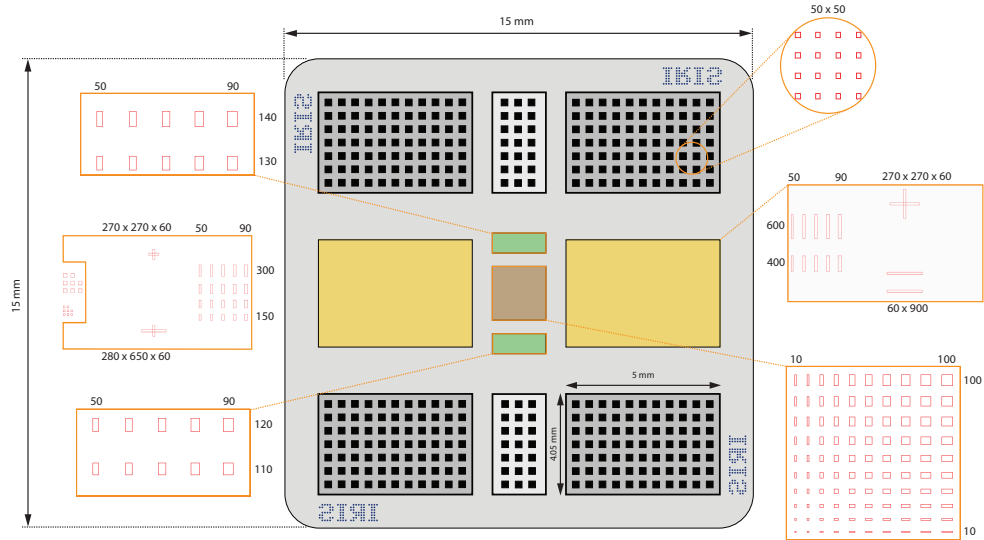


Fig. 3.15: Sections of the silicon workbench mask (all units in μm unless otherwise noted).

3.2.5 Calibration Units

Since the manipulator of the micro-assembly station V2 implements the remote center of motion concept (see Section 3.2.2) it is crucial to calibrate the system so that the three rotational axes all meet in one single point O_{RCM} together with the tool center point (TCP) of the current gripper. This can be reached by allowing movement of all rotational axes normal to the planes whose normal vectors are collinear with the rotational axes themselves, i.e., by introducing an xy -stage in every revolute joint. This is realized in the present case as shown in Figure 3.16. Axes ζ and η are both adjusted using individual miniature manual xy -stages with a sensitivity of $1\ \mu\text{m}$. Laser beam assistance is provided for axis η through a centered laser module L_η (see Figure 3.5a). Space limitations prevented the installation of a laser module for axis ζ so there is no visual feedback there. The adjustment of axis θ is separated into two components. Whereas y -movement is done using the large motorized driveout mechanism c_y (stage resolution: $1.6\ \mu\text{m}$), correction in x -direction is done by moving the whole 2 DOF gripper unit with a manually operated linear stage c_x (stage resolution $10\ \mu\text{m}$). Visual feedback is also provided for this axis by a centered laser module L_θ (see Figure 3.5a).

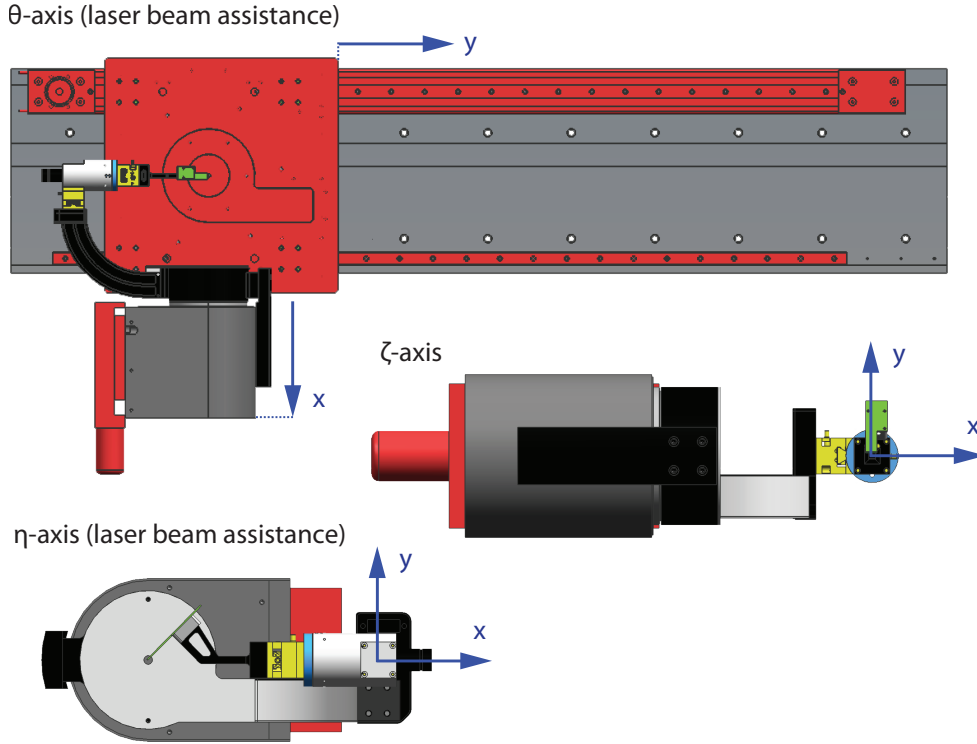


Fig. 3.16: Calibration mechanisms for the three rotational axes ζ , η , and θ of the manipulator. Axes η and θ are equipped with centered laser diodes L_η and L_θ (see Figure 3.5a) for visual feedback.

The built-in laser modules are economically priced types optimized for distance and not for beam collinearity. The minimum beam diameter in micrometers for the given lasers can be approximated by

$$d_{min} = \frac{l}{5} \quad (3.2.1)$$

where l is the distance in millimeters from the module to the point of interest. This yields values of around $80\text{ }\mu\text{m}$ for both axes. The laser diode manufacturer's specifications indicate a maximum beam diameter of $200\text{ }\mu\text{m}$. Figure 3.17 visualizes the situation of the laser beams hitting MEMS and mechanical gripper tips from the bottom (θ -axis) and the side (η -axis). The alignment strategy is to first locate the most outer contour of the gripper fingers by running a zigzag pattern towards this area (see Figure 3.17b). As soon as the beam hits a metallic surface light is heavily scattered in all directions and can be tracked by the mounted cameras. Due to the known gripper geometry the correction values for both x and y directions are determined and so is the final axis position.

The accuracy of this method firstly depends on the diameter of the laser beam which is a priori unknown. Additional to that, the final position accuracy also gets strongly influenced by the first contact position of the zigzag pattern search. Table 3.2 shows the geometrically approximated accuracy values.

axis	mechanical resolution	calibration accuracy	
		MEMS	mechanical
ζ	1	no laser	
η	1	$\leftrightarrow 150 - 160$	250
		$\updownarrow 100 - 180$	220 - 240
θ	10	$\leftrightarrow 60 - 100$	140 - 200
		$\updownarrow 80 - 100$	160 - 200

all values in μm unless otherwise noted

Table 3.2: Characteristics of the calibration devices (see Figure 3.17). The values seem rather large but do not heavily impair assembly operations. In order to reduce those values, an automated calibration sequence as well as high-accuracy laser diodes become necessary.

3.2.6 Vision Units and Illumination

As stated before vision sensory feedback is crucial for any micro-handling station. The micro-assembly station V2 features a maximum of three individually adjustable ring units observing the manipulation scene as shown in Figure 3.18a. The units are attached to a metal ring whose central axis is collinear with the rotation axis θ (see Section 3.2.5), and they can be configured with microscope-cameras, glue dispensers, etc., depending on the application. The kinematic configuration of each unit is depicted in Figure 3.18b.

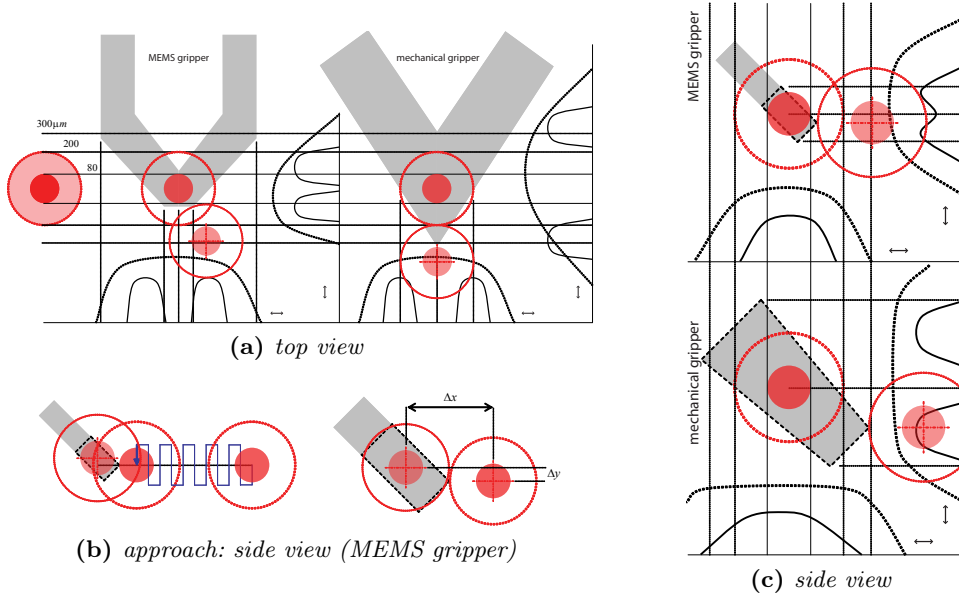


Fig. 3.17: Effect of calibration laser beam hitting the tips of the MEMS and mechanical gripper models (see Section 3.2.3). The top left image shows the situation for the θ -axis calibration whereas the right image stands for the η -axis (side view). The approach of the η -laser beam to the gripper tip is shown in (b).

The scene itself is surrounded by a metal dome in the form of a hemisphere providing customizable illumination conditions (see Figure 3.18c).

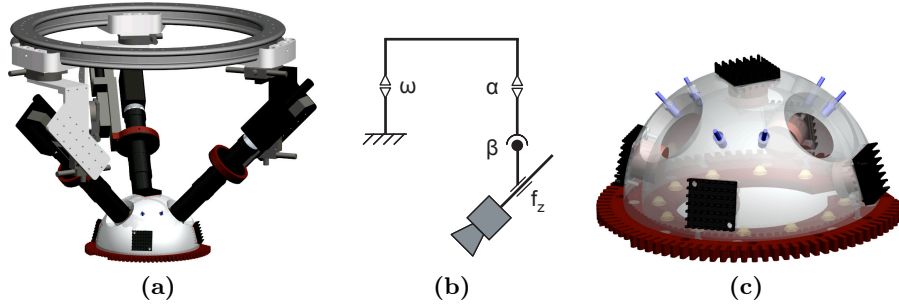
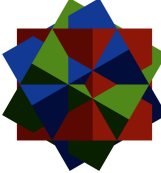
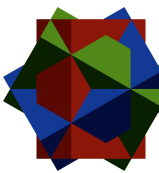
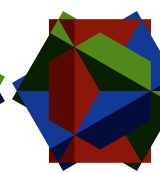
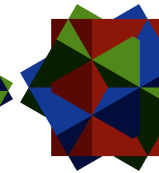




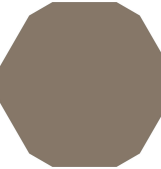
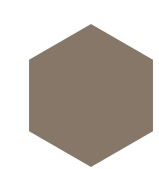
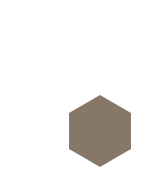



Fig. 3.18: Three pan-tilt units are connected to an upper ring which is centered around the vision dome (a) and each of them can be configured for specific needs. The kinematic configuration is shown in (b) and a closeup view of the vision dome in (c).

The main axes of the ring units are tilted down at a 45° angle from horizontal and equally spaced around the center. This configuration is beneficial for reducing the anisotropic sensitivity of microscope lenses that are unable to detect motion along the optical axis. It has been optimized by visualizing camera sensitivity through visual resolvability ellipsoids (refer to Nelson and Khosla [111, 112] for details). Table 3.3 shows the viewing volumes as well as some optical parameters for the current microscope arrangement. A viewing

volume represents the cuboid of each camera-microscope combination where a sharp image can be expected. The height of the cuboid is defined by the depth of field, and width and height by the field of view. Intersecting the bodies of all three cameras yields the volume where a sharp image for all camera entities can be expected. Again intersecting this with a plane at height z , which represents the workbench surface, results in the planar area of sharpness for all camera units.

PM	0.75x	1.0x	2.0x	3.0x
$\{R, G, B\}$				
$R \cap G \cap B$				
$R \cap G \cap B \cap H$				
SM	36x	48x	96x	144x
FOV	8.0	7.3	4.7	2.0
NA	0.016	0.030	0.040	0.045
DOF	3.82	1.28	0.59	0.42
d	1.3/3.1	2.1	0.9	0.6
h	8.0	4.18	1.926	1.372
V	209.0	18.3	1.7	0.6

all lengths in mm and volumes in mm³

Table 3.3: Viewing volumes for an EDMUND VZM-300i microscope for different levels of primary magnification (PM). The upper half shows the individual, unscaled viewing volumes (R stands for red, G for green, and B for blue color), their mathematical union as well as their union with plane H intersecting at $z = 0$. The lower half lists the system magnification (SM) using a $1/2$ " CCD chip, field of view (FOV), numerical aperture (NA), depth of focus (DOF), side length (d) and upright distance (h) of the section plane, and the viewing volume itself (V).

The present camera configuration scheme is shown in Figure 3.19. It consists of two BASLER A602fc-2 color cameras on positions RU_1 and RU_2 , since position RU_3 is occupied by a glue dispensing device. Depending on the type of application, this configuration can be changed within minutes.

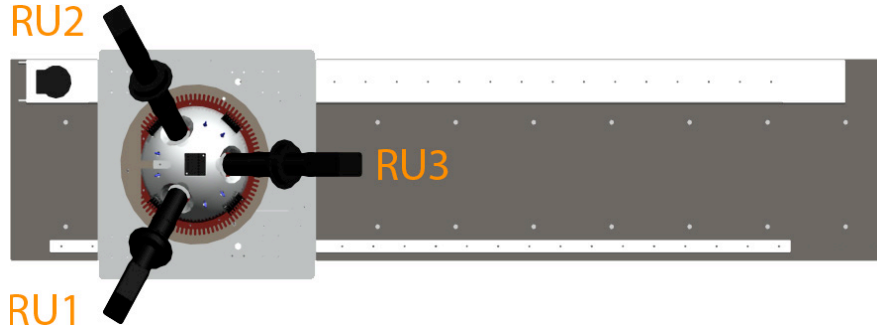


Fig. 3.19: Position and naming scheme of ring units RU_1 , RU_2 , and RU_3 , if looking down to the micro-assembly system.

A number of previous assembly experiments have shown that correct illumination is essential for micromanipulation tasks. Depending on the surface coating and the part shape one has to adjust light sources so that disturbing reflections and shadows are reduced to a minimum. The present illumination setup consists of three individual systems combined in a compact centered vision dome (Figure 3.20). The dome itself is not only a supporting structure and protection for the workspace, but also creates diffuse ambient (indirect) illumination by the light of twelve high power LEDs reflected at its inner matt-finished surface (Figure 3.20a). Each of these emitters provides around 80 lm @ 1 A, yielding a total of 960 lm. Indirect lighting or dome illumination provides a diffuse and homogeneous light without reflections, which becomes specifically useful when working with shiny spherical and convex objects.

The second illumination device is a single 1 W LED spotlight integrated in the summit of the dome (Figure 3.20a) and focused on the RCM (Figure 3.5a). It offers a bright and concentrated pool of light on the workbench and works best with non-reflective parts or in other situations where strong illumination is needed.

A set of four equally spaced 1 W LED spotlight units arranged around the dome at a low angle of 14° are designed to emphasize even the smallest deviations on the surface (Figure 3.20b). This results in high contrast images of the contours of any outstanding objects. As opposed to commercial low angle illumination systems, this implementation features two operation modes. In static mode, all four LEDs are triggered at the same frequency and zero phase shift. However, in rotation mode, they are triggered sequentially with the same frequency and a phaseshift of one cycle. This effect of rotating lightsources around the center creates shadows on alternating sides and thus allows the creation of depth images that give a more realistic view of the scene.

In addition, six high power UV LEDs are arranged at a 45° angle to the horizontal plane for curing UV glue during assembly sessions.

All light sources can be triggered and their power individually regulated in the range of 0 – 100 % using custom built electronics. As shown in Section 6.5 this allows creating illumination modes that are optimized for specific assembly tasks. Light control electronics are further explained in Section 3.3.2.

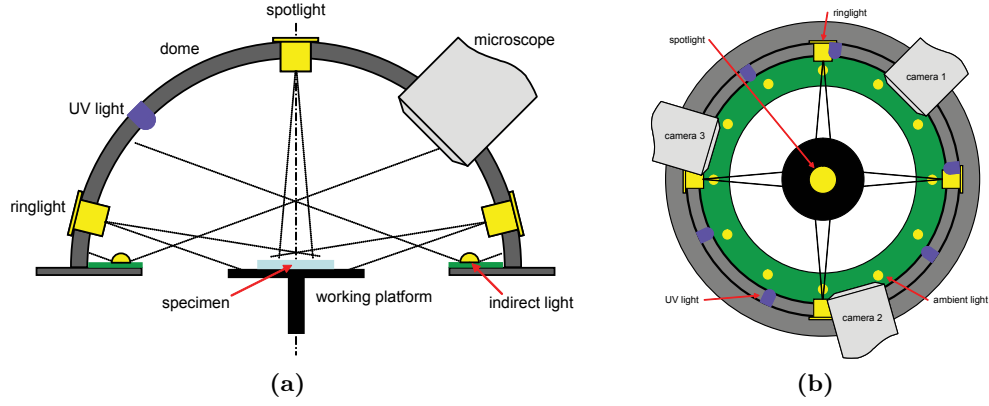


Fig. 3.20: Multi-purpose illumination dome made out of aluminium with spot-, ring-, ambient and UV-lighting. Cross-sectional view from the side (a) and the top (b).

3.2.7 Glue Dispensing Unit

One of the ring units can be equipped with a glue dispenser which basically consists of a fine needle attached to a glue reservoir that can be air pressurized in order to create fine glue droplets. Air pressure is prepared by a commercial microcontroller driven dispenser system with a minimum cycle time of 1 ms and adjustable input and output pressure levels.

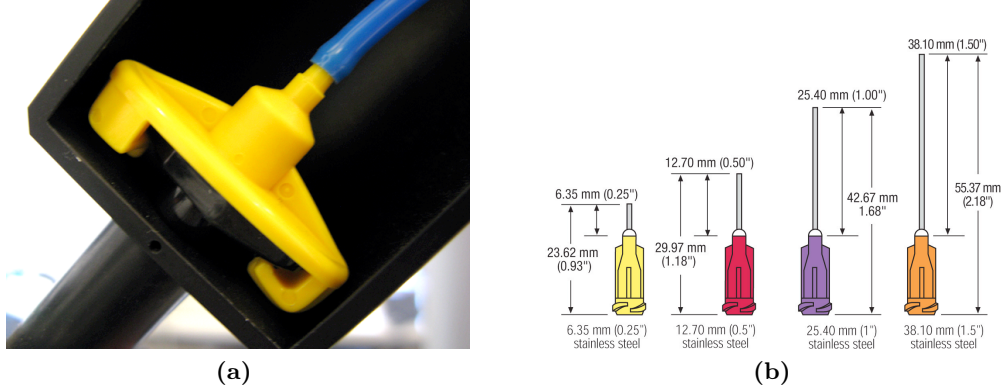


Fig. 3.21: Detail of the glue dispensing unit with the air pressure connection and barrel (a) as well as a series of stainless steel tips (b). All experiments have been performed with 1.5 " tips which are available in minimum tip dimensions of $\phi_i = 0.25$ mm and $\phi_o = 0.51$ mm.

Dosage of the correct amount of glue becomes a non-trivial task the smaller the parts are. Even the assembly of millimeter-sized objects yields surfaces to glue in the sub-millimeter region. Looking at the kinetic energy of a little droplet with radius r yields

$$E_{kin} = \frac{1}{2}mv^2 = \frac{1}{2}\rho Vv^2 = \frac{2}{3}\pi\rho v^2 r^3 . \quad (3.2.2)$$

The surface tension force for the same droplet is given by

$$E_{sur} = \sigma A = 4\pi\sigma r^2 \quad (3.2.3)$$

and thus

$$\frac{E_{kin}}{E_{sur}} \sim r \quad (3.2.4)$$

In other words the smaller the droplet is, the more dominant is the surface tension force and thus makes it difficult to release it from the needle. Industry states a minimum droplet size of around 1 μL ($r \approx 0.65 \text{ mm}$) for such tip contact systems. Experiments with the present system have shown larger droplet sizes in the order of around $r > 1 \text{ mm}$ which corresponds to the minimum cycle time of 1 ms of the air pressure controller.

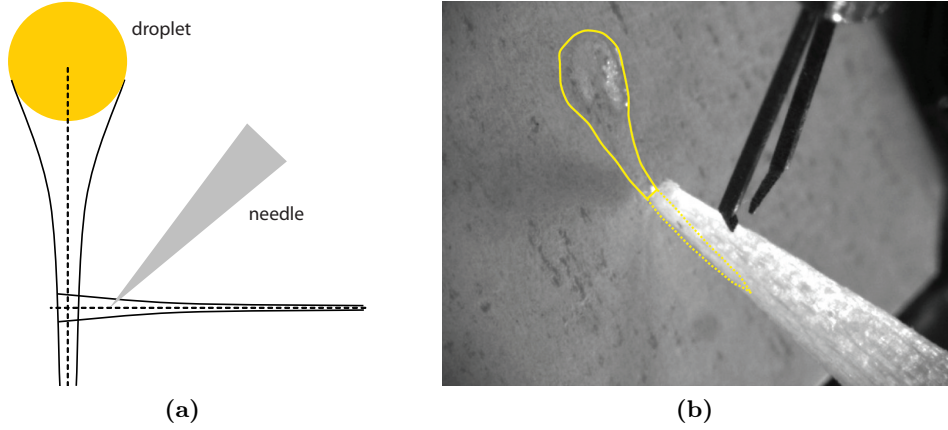


Fig. 3.22: A large droplet can be reduced by crossing it with a fine object, e.g., toothpick. The resulting lanes of various width can be used for dipping objects. However, this method makes precise glue deposition impossible and requires a designated and disposable area for glue deposition.

A possible method for creating smaller quantities consists of releasing a droplet on a flat surface and traversing it with a fine needle or toothpick (see Figure 3.22). The resulting fluid paths offer varying size but limit the application to convex surfaces that can be dipped. If local deposition is indispensable the use of piezoelectric driven valves, which shoot off droplets down to a diameter of 30 μm , offer a good solution.

3.2.8 Pneumatic System

The micro-assembly station V2 features a pneumatic system that delivers pressurized air and vacuum to virtually any location in the setup (see Figure 3.23 for an illustration). Compressed air is taken from the building and run through a conditioning unit for additional filtering. Several regulation stages drop the incoming high pressure to the required

levels. A venturi-type vacuum generator with an integrated vacuum switch and blow-off valve is mounted close to the conditioner. It provides low pressure air down to 350 mbar at a suction capacity of maximum 5.9 L/min. Both air and vacuum are also conducted through the θ rotation stage by a rotary ring. Devices fed by this system include the vibration table, brakes, a glue dispenser, etc.

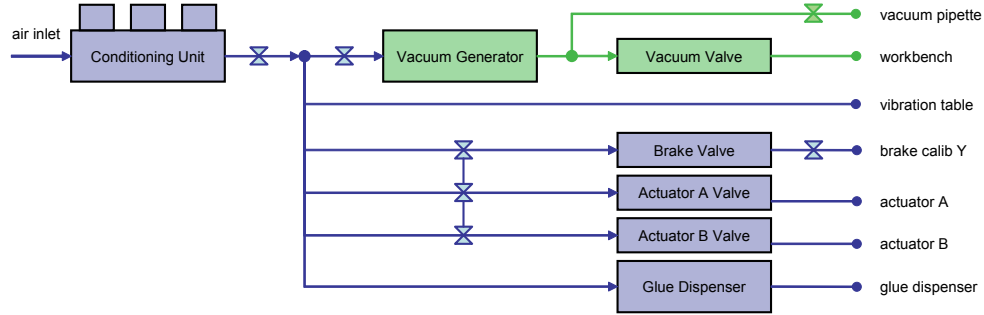


Fig. 3.23: *Pneumatic system setup. The hourglasses stand for manual pressure regulator valves.*

3.2.9 Mechanical Performance

Figure 3.24 shows the large workspace of the 6 DOF manipulator of the micro-assembly station. Due to the separation of the kinematic chains into 2 DOF and 4 DOF one can think of the workspace as follows: it is attached to the platform (4 DOF unit) for lateral movements in the xy -plane. For vertical movements along the z -axis the working area is fixed to the end effector (2 DOF unit).

Table 3.4 summarizes the performance of all motion devices in the system. Some values are not specified because they simply do not exist for the respective axes (i.e., SUTTER axes x, y, z and calibration axis c_x). The range values are for the design case and can be different on the real system due to their dependency on the adjustment of the limit switches.

3.3 Electronics and Control

3.3.1 Marvin Hardware Controller

The Marvin is a multipurpose hardware controller developed at Institute of Robotics and Intelligent Systems as a mobile robotics platform for education and research purposes. Its basic version consists of a rack with a seven slot CompactPCI (cPCI) bus and a Pentium-M processor board. Each slot can accommodate a carrier board that contains and interfaces to two ANSI mezzanine M-modules. The functionalities of these modules range from digital IO over DC motor to light controllers. The present system uses an extended and

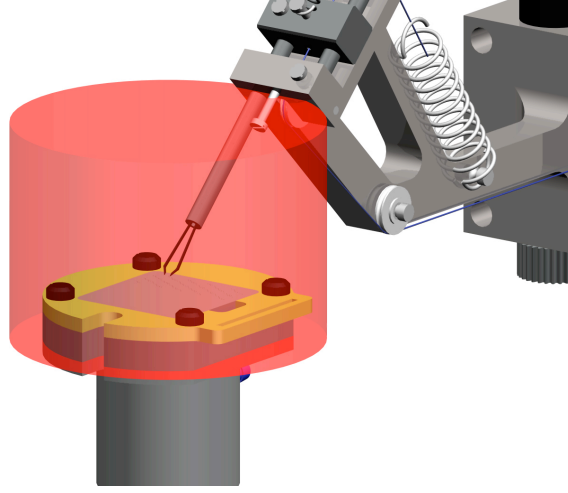


Fig. 3.24: Working area of the 6 DOF manipulator. The shaded cylinder has a diameter of 35.4 mm and a height of 25 mm.

	Motor		Encoder		Gear	Range		Performance		
	V_{cc}	P_{nom}	V_{cc}	ticks		x_{min}	x_{max}	resolution		v_{max}
x	24 V	15 W	—	—	1263:1	0 mm	25 mm	0.2	μm	0.2 mm/s
y	24 V	15 W	—	—	1263:1	0 mm	25 mm	0.2	μm	0.2 mm/s
z	24 V	15 W	—	—	1263:1	0 mm	25 mm	0.2	μm	0.2 mm/s
η	24 V	30 W	5 V	1024	560:1	-60°	$+45^\circ$	0.00016	$^\circ$	94.4 $^\circ/\text{s}$
ζ	18 V	13 W	5 V	512	100:1	-90°	$+90^\circ$	0.0018	$^\circ$	540 $^\circ/\text{s}$
θ	48 V	110 W	5 V	1000	90:1	$-\infty^\circ$	∞°	0.001	$^\circ$	80 $^\circ/\text{s}$
\mathbf{c}_x	—	—	—	—	—	0 mm	25 mm	1.0	μm	—
\mathbf{c}_y	24 V	20 W	5 V	1000	72:1	0 mm	591.5 mm	0.829	μm	231 mm/s
\mathbf{f}_z	3.6 V	1.2 W	5 V	16	4000:1	0 mm	30 mm	0.014	μm	0.375 mm/s

Table 3.4: Summary of the relevant parameters of the micro-assembly drive systems. All values are theoretical without taking into account system inaccuracies. Please refer to Section 6.4 for measurements and more details.

modified version of Marvin with a total of 14 slots fitted inside a standard 19-inch rack. The position of each M-module is fixed due to a backplane that provides additional features for safety and convenience. Motor drives and other units are attached to the backplane and not directly to the modules. Marvin has been extended by an additional rack which contains 5/12/24/48 V power supplies and power amplifier boards for stepper motors as well as high voltage generation.

The following two sections 3.3.2 and 3.3.3 describe two M-modules and their additional electronics that have been designed and built within the scope of this thesis.

3.3.2 Camera and Light Controller

All vision components explained in Section 3.2.6 are hooked up to a specially developed M-Module shown in Figure 3.25. It is built on the IRIS I00 base module that comes with an ALTERA FPGA with RAM, flash memory, and isolation circuits. The onboard power regulation provides the different voltage levels required by the FPGA. The remaining space is covered with switching devices and high-power (MOSFET) transistors that drive all vision components. The required power is provided externally and shielded from the bus electronics.

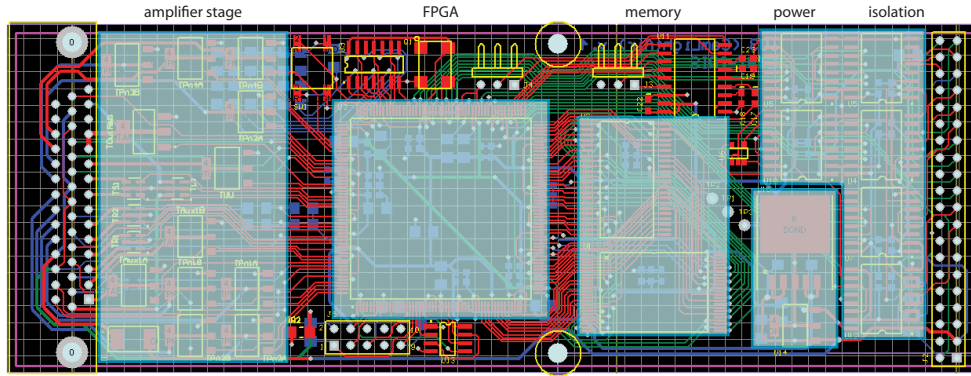


Fig. 3.25: IRIS *Camera-Light-Air* controller *M-module* board *I03*.

The module provides trigger outputs for cameras with variable frequency and duty cycle in 32-bit resolution. The lighting output channels are first combined in logical AND gates with the camera trigger as well as a PWM power regulation signal before they reach their individual MOSFET stages. Syncing illumination and image acquisition results in off-times and allows driving LED lightsources above their maximum power rating — a technique also known as LED turbocharging. A number of additional outputs, e.g. for pneumatic valve and laser control, can be PWM modulated but are not ANDed with the trigger signal. However, these combinations are not static since they are implemented within the FPGA and can be reprogrammed at any time. Table 3.5 shows some characteristics of the IRIS I03 module and Figure 3.26 the physical arrangement.

3.3.3 Force Feedback Gripper Control

In Section 3.2.3 a MEMS micro-gripper with force-feedback was introduced. In order to realize force-feedback control a number of electronic circuits are required. Those components as well as the signal path are shown in Figure 3.27. Note that the adaptor board mounted on the manipulator arm combines both signals from the high voltage DC amplifier and the DC motor driver thus offering high flexibility in the choice of the appropriate gripper.

	sync	power [0 – 100 %]	V_{\max} [V]	I_{\max} [A]
camera control				
trigger	✓	×	3.3	0.024
light control				
spot	✓	✓	20.0	2.7
rotation	✓	✓	20.0	2.7
ambient	✓	✓	55.0	3.8
gripper	✓	✓	20.0	2.7
UV	✓	×	20.0	2.7
laser	×	×	20.0	2.7
auxiliary devices				
valves	×	×	55.0	3.8
glue	×	×	5.5	0.128
aux1	×	✓	55.0	3.8
aux2	×	×	55.0	3.8

Table 3.5: *I03* output characteristics (*sync*: synchronized with the main trigger signal, *power*: output is ANDed with an adjustable PWM signal, V_{\max} : maximum output voltage, I_{\max} : maximum output current).

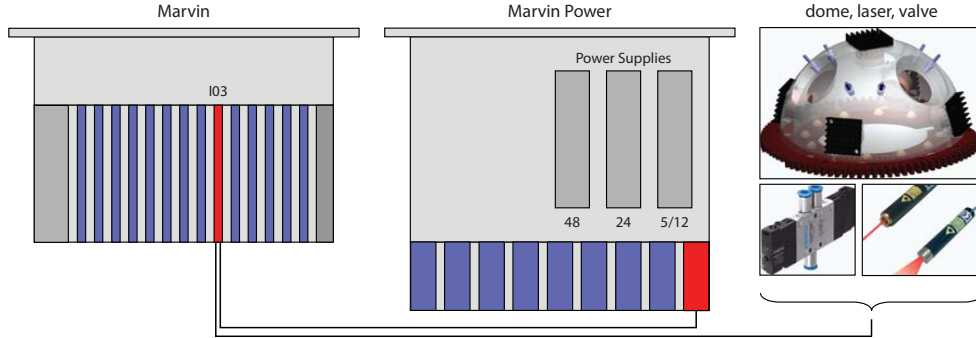


Fig. 3.26: Physical arrangement of the camera-light-air controller unit.

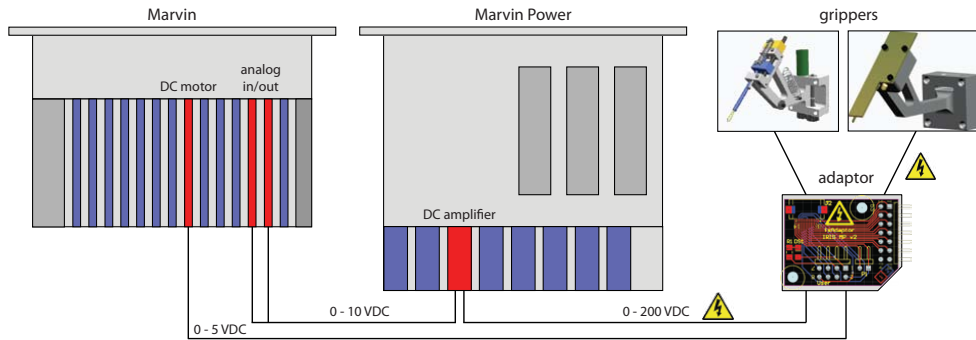


Fig. 3.27: Physical arrangement of the MEMS force-feedback and mechanical gripper.

The high voltage amplifier itself has two channels and works with APEX analog amplifiers. The circuit operates with an input voltage of $0 < V_{in} < 10$ V and provides a linear output range in the order of $0 < V_{out} < 200$ V at a maximum current of $1 - 2$ mA. Positive and negative supply voltages of $V_{s+} = 220$ V and $V_{s-} = -15$ V, respectively, are generated with standard DC-DC converter blocks. See Figure 3.29 for a visualization of the electronic circuitry. The excellent linearity of less than 0.1% and the large dynamic range of the whole circuit are shown in Figure 3.28.

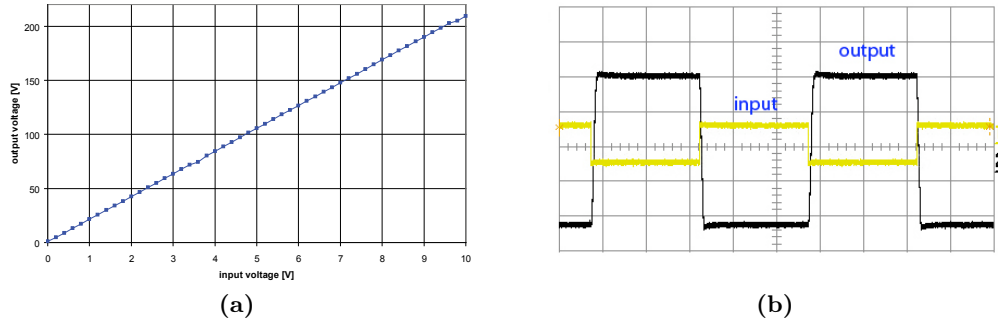


Fig. 3.28: Linearity (a) and dynamic range (b) of the high voltage amplifier board developed for the micro-assembly station V2. Note that the output signal of (b) is inverted with respect to the input signal.

The gripper control software component consists of an autonomous controller thread that is currently running a PD loop but can of course be extended to any other strategy. The control law is given by

$$V_c^{n+1} = V_c^n + K_p(V_d - V_r) + K_d \frac{V_d - V_r}{\Delta t} , \quad (3.3.1)$$

where V_c is the output control voltage, V_r is the measured input voltage and K_p and K_d are the control parameters. The desired voltage V_d is given by

$$V_d = \frac{F_d}{G} + V_o , \quad (3.3.2)$$

where F_d is the desired force, G is the gain of the characterized gripper ($45 \leq G \leq 60 \mu\text{N/V}$) and V_o is the offset voltage defined by the readout electronics.

Electrical connection from the adaptor to the different gripper types inside the dome is realized by HIROSE flexible flat cables.

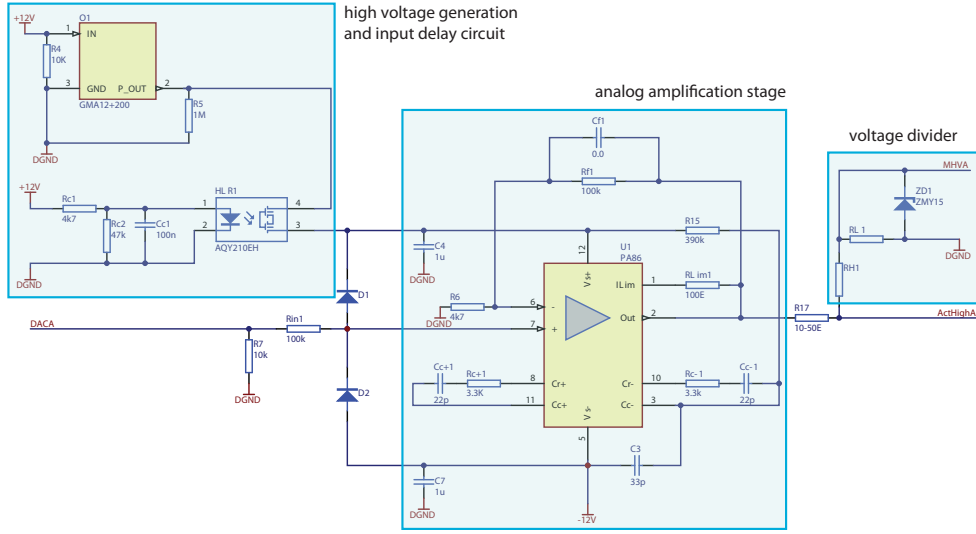


Fig. 3.29: One channel of the high voltage DC-DC conversion circuit.

3.4 Software Components

3.4.1 Overview

A lot of effort was put in the design of the computer system and the software architecture of the micro-assembly station V2, in order to simplify future developments and extension projects. This section briefly illustrates the involved hardware entities and the subsequent sections discuss the individual software components in more detail.

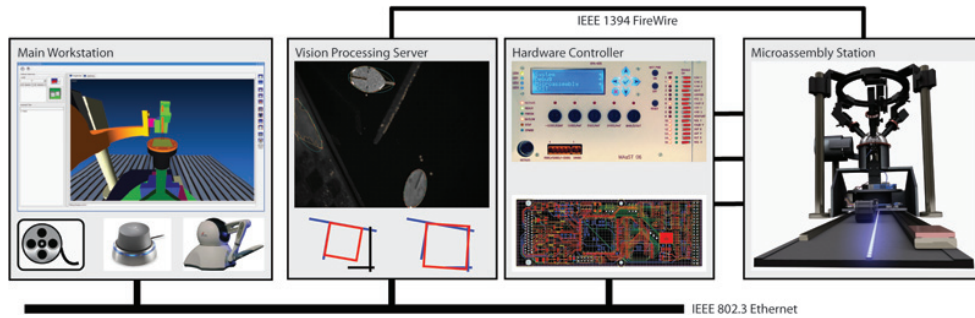


Fig. 3.30: Overview of the key computer and software elements designed and developed for the micro-assembly station V2.

Every single hardware component of the micro-assembly station V2 is directly attached to the Marvin robot controller. Marvin has been initially developed at IRIS for mobile robot competitions and is perfectly suited for this application due to its flexible design. A regular state of the art workstation hosts the main application and thus interfaces with

the user. It also processes events from haptic input devices. Due to the high processing power demanded by computer vision applications, a third computer was introduced at the very beginning. It is directly connected to the 1 – 3 FireWire cameras and executes vision processing tasks. See Figure 3.30 for a very simplified view of the situation.

Since Marvin is kind of the nerve center of the whole setup it is named ERASMUS after the great humanist and theologian DESIDERIUS ERASMUS ROTERODAMUS (*1465/69; †1536). The workstation with the user interface is called COLLOQUIA and the vision server ADAGIA — both extensive works written by ERASMUS.

All three machines are linked with each other over regular Gigabit Ethernet. Technically, this allows the client workstation COLLOQUIA and the vision server ADAGIA to be placed anywhere even though this was never a design objective. ERASMUS is of course bound to the hardware. Communication between all machines is done with Player¹ (Gerkey et al. [61]), an open source cross-platform robot device interface and server platform that has proven to be very reliable and flexible. ERASMUS and ADAGIA are both running Player servers whereas COLLOQUIA implements a client and connects to those servers and routes information. There is no direct connection between ERASMUS and ADAGIA.

The following sections give a brief overview of the software architecture and its major components. More in-depth information can be found on the project website <http://www.microassembly.ch> and in the documentation of the libraries themselves.

3.4.2 ERASMUS: Hardware Controller

The software for ERASMUS is built around an extensive Player driver (hardware control driver) that itself talks to the low-level drivers for the individual hardware components (see Figure 3.31). The frequency of the driver’s main routine defines how often commands and requests from connected clients are processed and since there are no control loops directly embedded, this value can be rather high (~300ms). As explained in Section 3.3.1 ERASMUS features a built-in watchdog that triggers an emergency power down in case of a software crash. Setting the watchdog bit is done in a second thread running independently from the main driver usually at higher speed. If the mounted gripper is a MEMS force-feedback type a third thread running a high-speed PID control loop is created. This configuration is disadvantageous since it consumes valuable processor time and future developments should consider rolling this out to electronics.

The main Player driver features a defined interface with a large set of commands and requests. In order to keep network traffic low, data from the manipulator axes and other components are regularly published without request whereas lower priority information is available upon demand only.

More detailed information about the hardware configuration of ERASMUS can be found in Section 3.3.1.

¹ <http://playerstage.sourceforge.net>

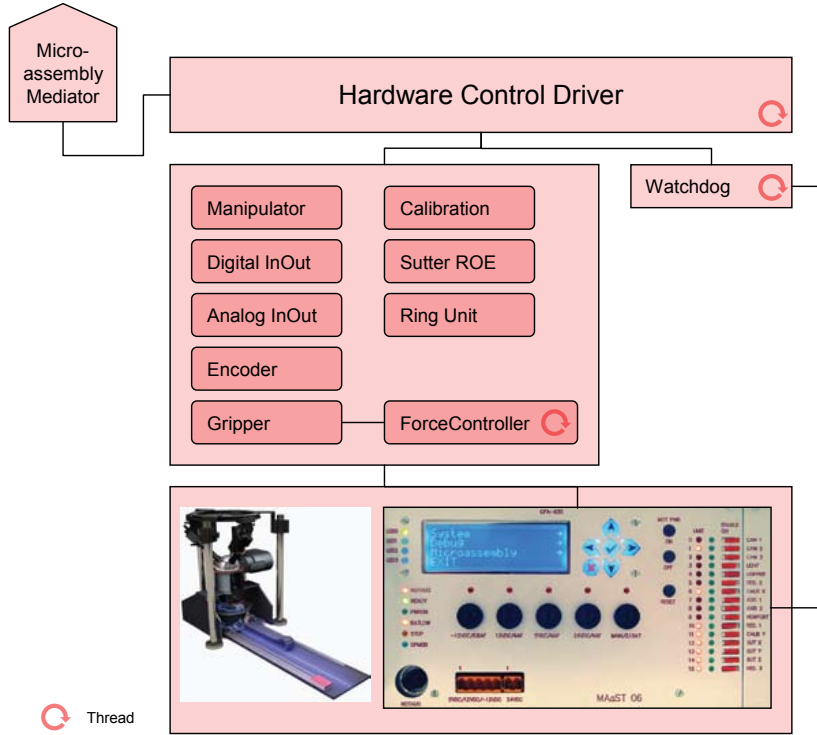


Fig. 3.31: *Software architecture of the Marvin hardware controller ERASMUS.*

description	filename
Player configuration file	/etc/microassembly/ma2_base.cfg
hardware definition file	/etc/microassembly/motor.xml

Table 3.6: *Configuration files for the hardware controller ERASMUS.*

3.4.3 ADAGIA: Vision Server

Similar to the hardware controller the vision server runs a main **Player** thread, which handles incoming requests and commands, as well as outgoing data packages.

The vision server is basically composed of three layers (see Figure 3.32 for clarification). Analogue to the hardware controller ERASMUS, the top layer is the main **Player** thread dealing with incoming and outgoing data packages from the network. This *vision control driver* owns the next layer called *object observer* which is the main interface class to the tracking threads and all other elements of the computer vision library. The last layer consists of the vision processing and tracking algorithms and is held by the *object observer*.

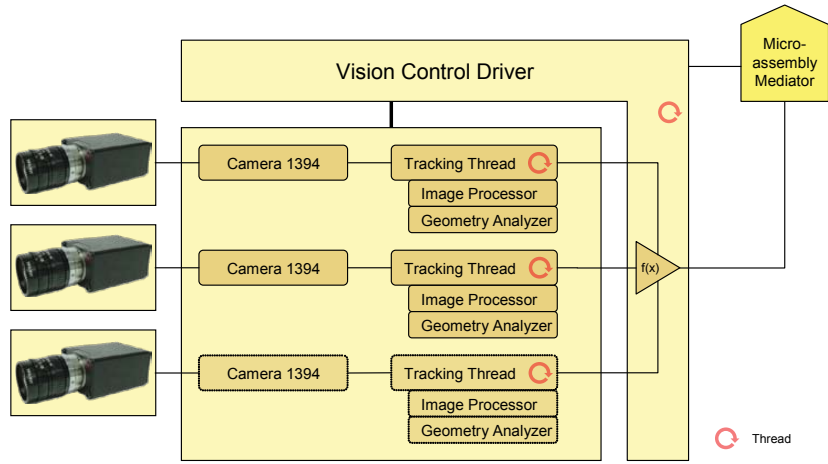


Fig. 3.32: Software architecture of the computer vision server ADAGIA.

Networking

Networking is solely based on the *Player* library by directly inheriting from its driver class. A clear interface provides a number of commands and data structures that can be used for setting parameters and gathering image information. The *vision control driver* is designed to directly talk to the *vision control center* which runs on the client workstation COLLOQUIA.

description	filename
Player driver config file	/etc/microassembly/visioncontrol.cfg
camera config file	/etc/microassembly/camera.xml
camera calibration data file	/etc/microassembly/calibrationData.xml
camera calibration config file	/etc/microassembly/camCalibrationConfig.xml

Table 3.7: Configuration files for the vision server ADAGIA.

Tracking

The *object observer* holds control interface classes to the IEEE 1394 (FireWire) cameras (*Camera 1394*) of the micro-assembly station V2, offering access to the most common options and features, such as grabbing images. In addition, it also contains individual *tracking threads* whose number is defined by the available FireWire cameras attached to the bus. Every single tracking thread and corresponding camera 1394 instance is assigned to one of the FireWire cameras plugged into the vision server. The tracking threads perform continuous image processing as well as pose estimation and tracking tasks based

Listing 3.1: Working with image processes.

```

1  ImageProcessor imgPrc;
   ImageProcess* proc = NULL;
3
   proc = new SmoothingProcess( CV_MEDIAN, 5, 0, 0.0, true );
5  imgPrc.addProcess( proc );
   proc = new EdgeDetectionProcess( 20.0, 100.0, 3, true );
7  imgPrc.addProcess( proc );
9
   imgPrc.process( srcImg, destImg );
   IplImage* p = imgPrc.getProcessedImage();
11
   imgPrc.destroyProcesses();

```

on incoming camera images, and run in separate independent threads. 3D geometry data for model-based predictors is directly sent from the client workstation’s common model database (CMD) to this driver and allocated for the tracking threads. Due to the constant need for “fresh” camera images for different parties they all share the same camera 1394 instances. If more than one camera is attached to the system, data from the individual trackers is combined with a function $f(x)$ that depends on the tracking scheme used. The client workstation can ask anytime for a set of results and is returned a data structure containing pose, error measures, and other relevant data.

Image Processing

A tracking thread consists of an *image processor* and a *geometry analyzer*. The image processor is a container with a list of image processes that it runs through sequentially. Each image process inherits from the same base class, offering a unified interface that basically consists of an input and output image as well as an execution command. Image processes can be filters, morphological operations, etc., but also feature extractors, such as edge detection algorithms. Each process has a variable argument list that can also be externally configured through the vision control driver. The underlying container architecture offers great flexibility and transparency to higher level classes. Adding or removing a process as well as changing the order of execution is very simple since each element is represented by two lines of code. Listing 3.1 shows a brief example of how this procedure looks like. For most of the basic image processing functions the open source computer vision library OpenCV is used.

The *geometry analyzer* is invoked after the image processing sequence has completed. Since this component is based on geometrical information extracted from camera images, it is logical that the last image process of the *image processor* has to be a feature extractor type. If this is not the case or the process does not find any features in the image, it cannot start. Depending on the configuration, the geometry analyzer initiates a localization or tracking routine.

Camera Calibration

The success of pose estimation and tracking algorithms strongly depends on the knowledge of the accurate position and orientation of the cameras with respect to a common base frame located on the workbench. The vision server features a semi-automatic camera calibration module which is based on direct linear transformation (DLT) and non-linear optimization algorithms (see Appendix B for a detailed mathematical explanation of those procedures). The whole calibration routine is implemented in **Matlab**, compiled to a dynamic library using the cross-compiler *mcc*, and called from C++ through a special interface. Calibration results and analysis can be found in Section 6.2.

3.4.4 COLLOQUIA: Client Workstation

The client workstation is the absolute nerve center of the whole system since it runs the main application **MAPILOT**, combines information from its satellite computers **ADAGIA** and **ERASMUS**, and processes input commands from the operator. The graphical user interface (GUI) serves as the visualization front end displaying the status of the whole micro-assembly system. It also allows sending commands to the application and the hardware via the control panel window and a number of plugin widgets that are accessible over a central toolbar. The control panel widget, which connects to the real hardware, as well as the VR view widget, which displays different artificial views synchronized with the real scene, are required for operation and thus loaded by default. The main application's tangible user interface (TUI) processes commands from the haptic input devices (3D spacemouse, Phantom[®], etc.) and prepares them for GUI interaction or micro-assembly commands. Communication between all these entities and the real hardware is loosely based on the mediator design pattern (refer to Gamma et al. [60]) and thus managed by a single controller instance. See Figure 3.33 for an overview of the software architecture.

Tangible User Interface (TUI)

The tangible user interface consists of a 3D spacemouse assigned to axes (x, y, z, θ) and a 6 DOF Phantom[®] Omni[™] with limited force-feedback for axes (η, ζ) . The specific separation of the 6 DOF into groups of 4 and 2 has shown to be favorable since it reflects the physical constellation of the 4 DOF base unit and 2 DOF gripper unit introduced in Section 3.2.2. Due to the different nature of the devices the 4 DOF are velocity controlled and the remaining 2 DOF position controlled. Velocity and position data are directly handled by the *axis control center* and the further actions depend on two control modes that can be toggled with one of the Phantom[®] Omni[™] buttons. The first one is the GUI mode, where the spacemouse allows changing the orientation of the selected virtual reality view window and the Phantom[®] Omni[™] emulates a regular mouse including the left mouse click by penetrating a virtual plane using force-feedback. This mode has been implemented mainly for the reason that the operator does not have to leave the haptic devices and switch to regular controls just for changing a setting or reorienting a view. The other mode is the robot control mode used for actual control of the machine. However,

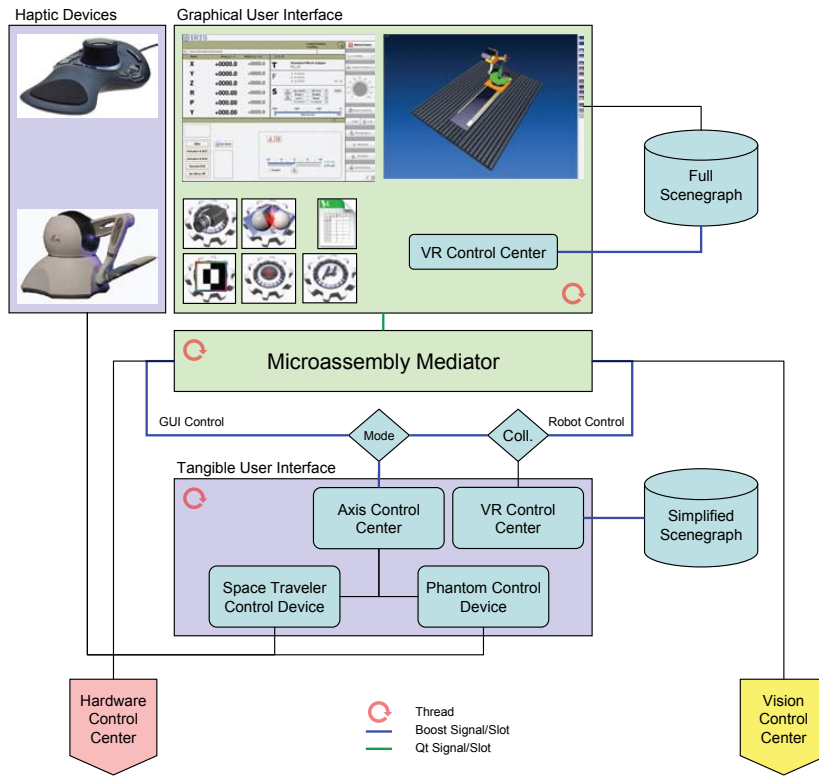


Fig. 3.33: *Software architecture of the client workstation COLLOQUIA.*

before any command is sent to the real hardware it is applied to a simplified virtual model in order to check for potential collisions (see Section 4.5). Safe commands are sent to the hardware whereas problematic ones result in the vibration of the Phantom® Omni™ and the absence of command transmission.

Graphical User Interface (GUI)

The graphical user interface abstracts the complexity of the underlying design and provides intuitive control over the whole process of micro-assembly.

The GUI is spread over two screens: by default the smaller one on the left contains the *main control panel* (see Figure 3.34) and the bigger one on the right the *VR view widget* (see Figure 4.9). A toolbar next to the control panel provides access to additional elements implemented as dynamically loaded plugins that are defined in a configuration file. Every plugin can be opened and closed as well as positioned anywhere on the large desktop, except the two above-mentioned default components *main control panel* and *VR view widget*.

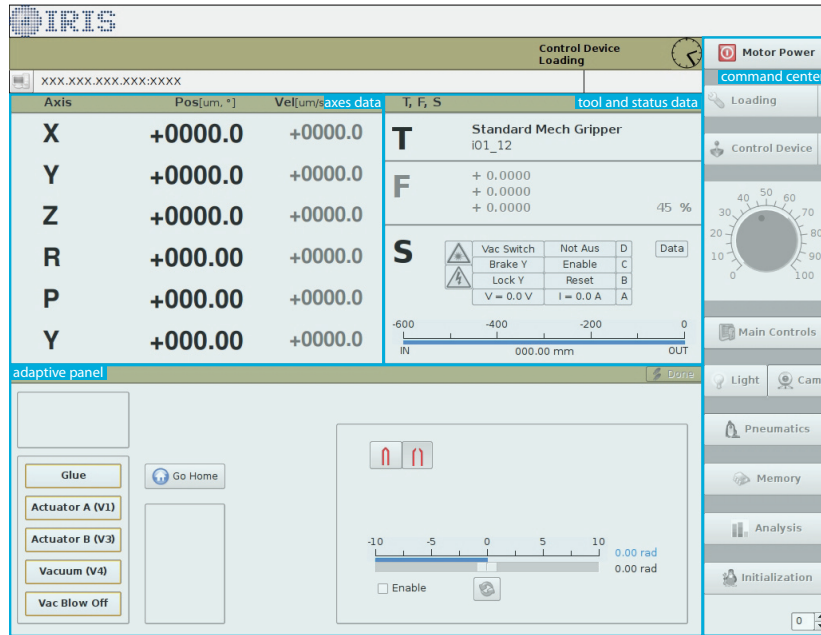


Fig. 3.34: The main control panel widget is the gate to the micro-assembly hardware.

The Plugin System

Dynamic loading and unloading of functional entities (plugins) adds to the overall flexibility of the application. The ability to include and exclude certain plugins from being loaded during the starting sequence of the main application is beneficial for debugging or other experimental purposes. So far, the following widgets have been implemented (in alphabetical order):

Control Panel The first widget loaded by default giving access to the whole micro-assembly station hardware.

Camera Control Widget Shows information about all cameras attached to the system and allows setting some parameters.

Collision Detection Widget Shows a list of collision pairs and the current status of the detector. It also provides controls for enabling/disabling collision detection and changing some parameters.

Data Viewer Data monitoring widget that shows a hierarchical representation of hard-coded data elements.

Event Recording Widget Recording and replaying of assembly sessions. This widget has not yet been fully implemented!

Image Processing Widget Controls image processing and tracking activities on the vision server ADAGIA.

Scene Tree Widget Shows the scenegraph of the loaded VR model of the micro-assembly station. It is linked to the VR view widget such that selected geometries in the VR view will be highlighted in the scene tree and vice versa.

VR Config Widget This widget allows loading a specific VR model as well as corresponding configuration XML files. Since the default startup sequence automatically loads those files this widget is primarily for debugging and testing purposes.

VR View Widget The second widget loaded by default providing different real and virtual views of the micro-assembly scene. The user-interactive viewports as well as a toolbar provide additional functionalities.

The concept of dynamic loading and dependency has been pursued throughout the main application. For example, if the image processing or camera control widgets are to be loaded they request the initialization of the low-level vision server so that their controls can be properly setup. If they are not loaded the vision server is not created either. Even though this results in a lack of functionality of the main application it does not impair the latter from running, at all. The concept simplifies debugging and the development of new features.

description	filename
Player configuration file	/etc/microassembly/maPilot.cfg
hardware controller client config	/etc/microassembly/maControl.xml
reduced scenegraph elements	/etc/microassembly/clone_nodes.xml
collision pair configuration	/etc/microassembly/collision_pairs.xml
joint control definitions	/etc/microassembly/joint_control.xml
joint limit definitions	/etc/microassembly/joint_limits.xml

Table 3.8: Configuration files for the client workstation COLLOQUIA.

3.4.5 Haptic Interfaces

The six manipulator axes of the micro-assembly system are mainly controlled over two haptic input devices: a Phantom[®] Omni[™] from SENSABLE as well as a SpaceTraveler3 from 3DCONNEXION (see Figure 3.35) that are directly connected to the main workstation. The six degrees of freedom are separated according to the real kinematic configuration 4 DOF + 2 DOF. The left hand controls $xyz\theta$ using the SpaceTraveler3 whereas the right hand moves the arm and wrist $\eta\zeta$ with the Phantom[®] Omni[™]. This configuration has proven to be intuitive and natural since it does not require the operator to perform any mental translation to the real situation.

A third haptic device can also be used for controlling any axis on the system. It is a modified three-axis input device ROE-285 from SUTTER INSTRUMENTS that is hooked up directly to the hardware controller and therefore provides lag-free control. It features

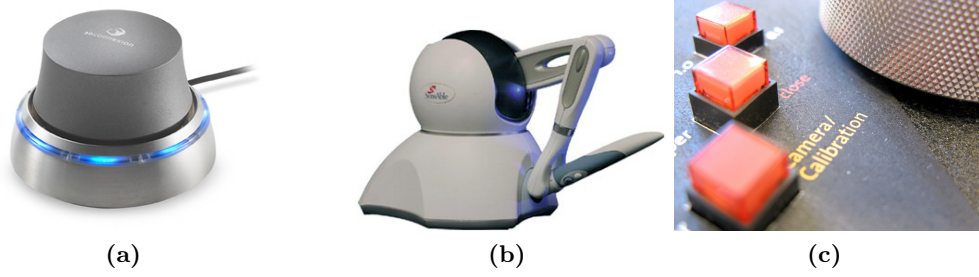


Fig. 3.35: Haptic input devices. 3DConnexion *SpaceTraveler3* (a), Phantom® *OmniTM* (b) and closeup of a modified SUTTER INSTRUMENTS ROE-285 (c).

three large knobs allowing precise movements and four buttons for selecting the desired axes.

3.5 Calibration and Initialization

Calibration and initialization of the micro-assembly station V2 is an important step as emphasized again later in Chapter 4. It basically consists of the following steps:

1. Calibration axes c_x and c_y are initialized.
2. Rotation axis ζ is initialized by driving to the positive and negative limit switch and then to 50 % of the maximum travel.
3. Rotation axis η is initialized by driving to the lower limit switch and then to a pre-defined position, which has been mathematically determined and experimentally verified.
4. The tool center point TCP of the gripper is placed in the remote center of motion RCM as explained in Section 3.2.5.
5. The xyz -stage is initialized by driving each axis to both limit switches and then to 50 % of the maximum travel. The planar correction factors for x and y in order to align the center of the workbench with the laser axis L_θ are optically determined by rotating around θ .
6. The ring units are initialized by simply driving them to the most rear position and resetting their encoders.

3.6 Summary and Conclusions

This chapter describes in detail the design and implementation of an advanced micro-manipulation system for the primary application of bio-microrobot assembly. The system features six degrees of freedom with high precision stages, easy access to the workbench

for part feeding, advanced illumination and vision, as well as an extensive and flexible software package distributed on three computing entities.

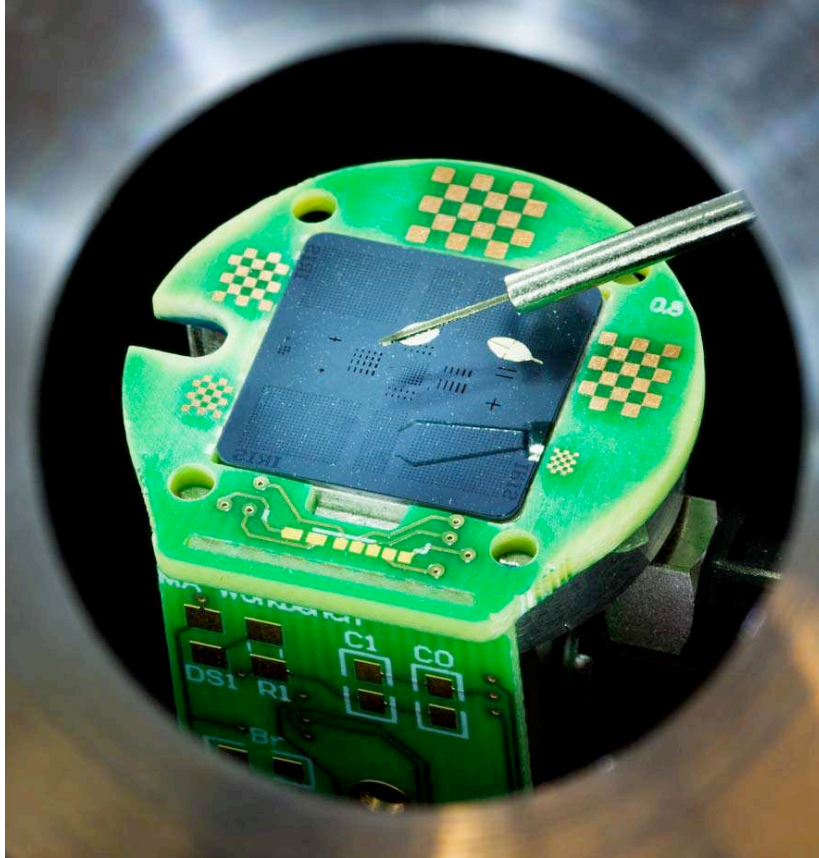


Fig. 3.36: View of the assembly area through a camera hole. Image courtesy of José Barea, for *GEO* magazine, 2008.

A particularity of the system is its flexibility and expandability in order to combine the most fancy structures and materials to complex three-dimensional hybrid MEMS structures. Figures 3.36 and 3.37 conclude this chapter with some artistic impressions of the micro-assembly station V2.

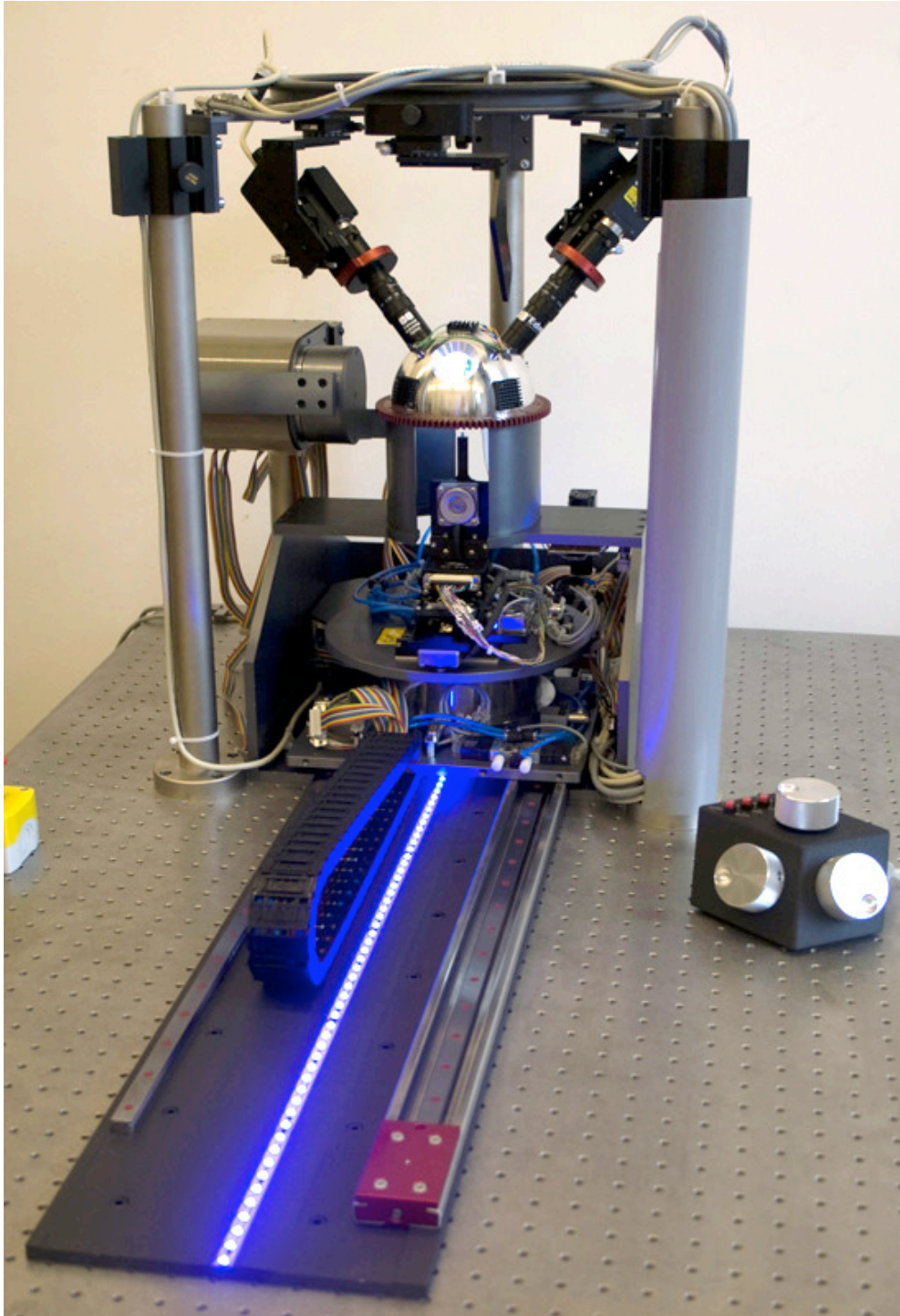


Fig. 3.37: *IRIS micro-assembly station V2. The blue warning light strip indicates potential movement along the calibration axis c_y .*

Virtual Reality For Micro-Assembly

The design and development of a versatile virtual reality environment capable of controlling any multi-axis robotic system is presented in this chapter. A detailed description of the system architecture together with every individual module is given in order to understand signal and data flow mechanisms.

4.1 Motivation

As mentioned in the previous chapter the concept and final design of the micro-assembly station V2 is based on a parent system. In contrast to the present configuration, that system was far simpler in terms of control electronics and software, as shown in Figure 4.1. Input commands were taken from a single 3D spacemouse which controlled all six degrees of freedom of the station. Vision feedback from the fixed camera views was given by third-party software which required opening a new instance for every single camera on the IEEE 1394 (FireWire) bus.

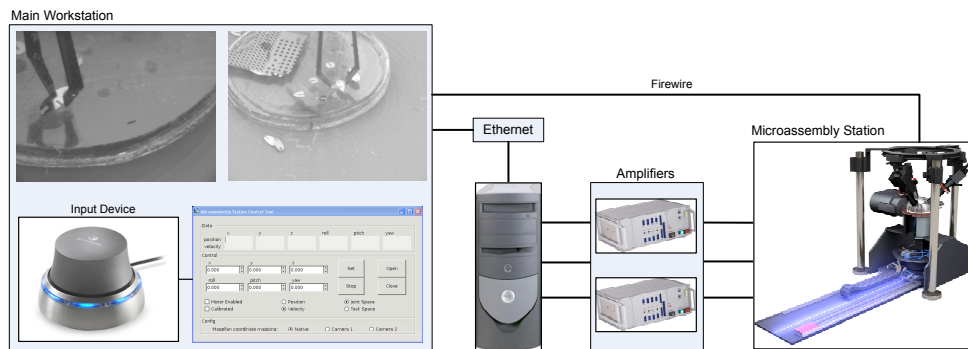


Fig. 4.1: Schematic of a version one micro-assembly system setup without virtual reality support. Images from two or three IEEE 1394 (FireWire) cameras surrounding the scene were directly displayed in the control center (top left) and provided the only information about the manipulation process.

Figure 4.2 shows some examples of fixed camera views from a regular micro-assembly session. It is evident that the poor illumination conditions and reflections are not beneficial for high-precision assembly. In addition, the fixed viewports make it virtually impossible to precisely align components for insertion unless the alignment axis coincides with the optical axis of one of the cameras. This becomes even more problematic due to the orthographic projection model of the microscopes as clearly illustrated in Figure 4.3.

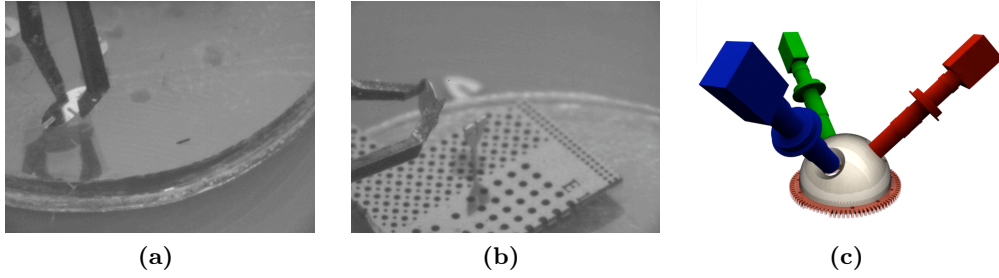


Fig. 4.2: Assembly of micro-robot components given fixed microscopic views. Figures (a) and (b) represent the views from cameras arranged as depicted in Figure (c). The given illumination conditions, shadows, reflections, and the cluttered scene create a difficult working environment for the operator.

Thus, the ability to change the viewport(s) to any desired location or to have multiple viewports of the same scene for precise alignment tasks is a big advantage over a fixed-viewport system. Since the physical movement of microscope-camera units is virtually impracticable and lacks full flexibility of motion, the solution is to recreate the working scene as a virtual environment. This requires three-dimensional models of all objects involved which are readily available from the CAD driven design process. A 3D representation also allows the selective hiding of parts, functional coloring, collision detection, etc. — features that simplify the assembly process and make it more safe and intuitive for the operator.

Since the model-based pose retrieval computer vision algorithms also depend on full 3D representations of components the setup of a common repository for 3D geometry becomes apparent. This concept of a *common model database* and its additional advantages are discussed in the subsequent sections.

The virtualization process can also be driven further towards a virtual assembly process as visualized in Figure 4.4. The goal is to keep a digital version of the current state and membership of all objects in the scene. For example, the gripping of a part A with gripper G releases A as a member of the platform P and assigns it to the gripper G. Of course, this technique relies on the full control of the real scene by computer vision.

The primary goal of this work is neither *teleoperation* (controlling a device over a distance) nor *augmented reality* (combination of real-world data with virtual elements in order to highlight certain elements). These terms are different from *virtual reality* (interactive computer simulated environment). However, parts of these other concepts might be implicitly used.

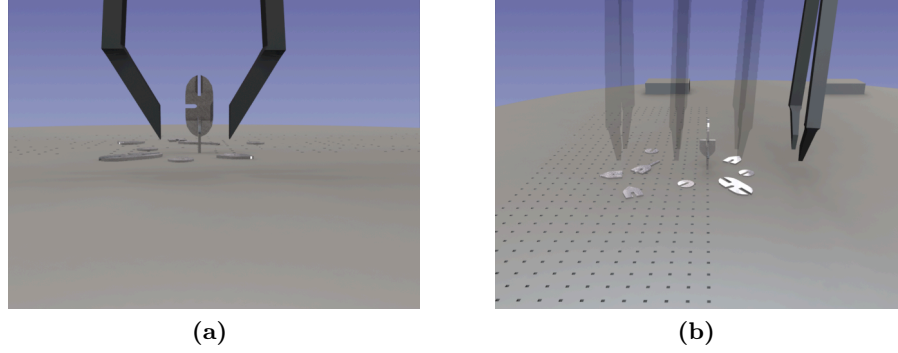


Fig. 4.3: Lack of perspective problem due to the orthographic projection of microscope lenses. The image on the left does not provide any depth information and, thus, hides the effective position of the gripper. Inclining the optical axis with respect to the workbench (45° in this system) reduces this effect. However, switching the viewport to an oblique position, as shown in the image on the right, provides full depth information.

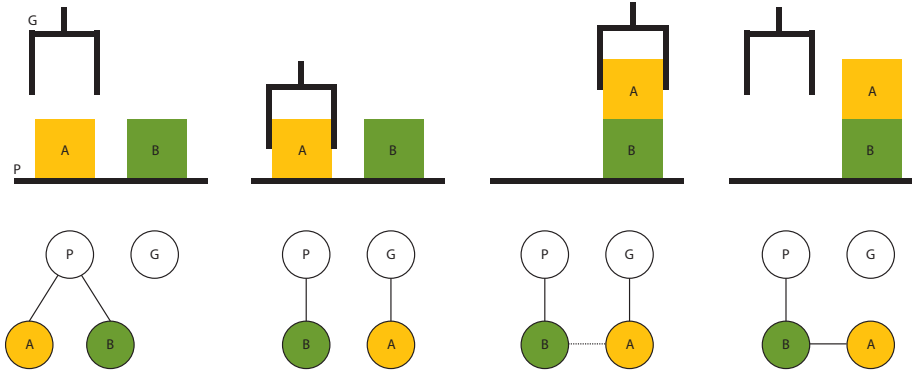


Fig. 4.4: Assembly tree and part memberships. From left to right: gripping of a part A with gripper G changes the membership of A from platform P to G. Consequently, stacking A on top of part B makes A a member of B and releases the membership from G.

4.2 Architecture

The basic software architecture and its separation on three different computing entities has already been described in Section 3.4. The focus of this section is on the software package creating the virtual environment which is running on the main workstation as shown in Figure 4.5.

Zooming in on the main workstation yields the more detailed structure as well as the internal signal flow as depicted in Figure 4.6.

1. Motion commands from a 3D input device or a task planner (also referred to as motion events) are sent to the axis control center and an event recorder.

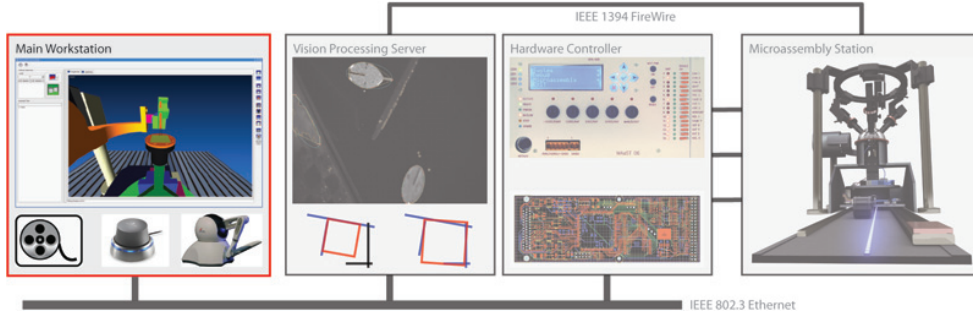


Fig. 4.5: Hard- and software setup with focus on the main workstation.

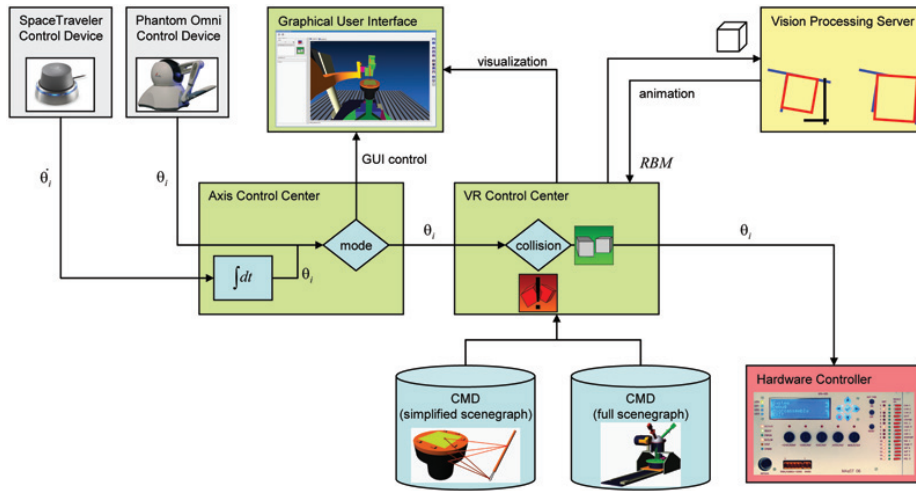


Fig. 4.6: Internal structure of the virtual reality environment running on the main workstation.

2. The motion events can be stored by an event recorder so that they can be replayed at a later stage.
3. Depending on the mode setting selected by a button on the Phantom[®] Omni[™] the axis control center handles the commands as follows:

GUI control This mode allows controlling the regular mouse with the input device Phantom[®] Omni[™] so that hands stay on the haptic devices all the time. The left mouse click is emulated by a virtual plane that has to be penetrated. The plane is defined by a force pointing along positive z direction. The 3D spacemouse is used for controlling the virtual model in this mode.

Robot control This mode is for directly controlling the micro-assembly station with both haptic input devices.

Motion commands are only processed from one of these two sources at the same time, in order to avoid conflicts and priority issues.

4. In robot control mode the events are handed over to the virtual reality control center which checks the incoming motion events for consistency and potential collisions. This is done by executing those commands in a reduced virtual environment which only contains the relevant geometry with a certain risk potential.
5. Harmless moves are directly passed to the robot controller for immediate execution whereas all others result in a visual and haptic notification of the user.
6. Position encoder feedback from the hardware is fed back to the virtual reality control center and results in an update of the reduced and the full virtual environment which is then rendered in the GUI.

This procedure ensures that only valid motion commands are sent to the real hardware while all others are blocked on the client side. However, the downside of this dry run is the introduction of a time lag between command and execution as discussed later in Section 4.9.

4.3 Scene Graph and Visualization

N -dimensional geometry is commonly stored in hierarchical tree structures known as scene graphs. Each hierarchical entity of a scene graph is called a node, which can hold multiple children, and itself be a child of several parents. Each node can hold multiple data objects called leaves which contain attributes and properties of the corresponding node. In order to create a virtual environment, that approximates a natural appearance for a human observer, some basic data elements as shown in Figure 4.7a are used. Scene graphs usually implement propagation methods so that properties affecting parents also affect their children. Figure 4.7b shows an example of a scene graph for a simple scene.

An extensive evaluation based on several weighted criteria finally led to the open source scene graph library **OpenSG**. This package provides excellent scene graph functionalities, integrates nicely with other libraries, and is widely used in a large community for a variety of applications. The existence of a number of collision detection libraries based on, or compatible with **OpenSG** was the decisive factor for the selection.

Setting up the scene graphs for both the micro-assembly station V2 and the micro-parts is done by importing 3D geometry that is readily available from the CAD driven design process. Format incompatibilities and some mandatory model modifications, which cannot be done in CAD software, require an additional conversion step that is performed in the DCC application **Maya**[®] (see Figure 4.8).

Now that the scene graph is converted to the proper format it can be manipulated within the host application by a set of specifically designed modules. Those include finding nodes, labeling nodes, dumping a scene graph or parts of it to the screen or a file, changing materials, hiding nodes, etc. In addition, navigation and picking functionalities have been implemented so that the geometry is updated according to user input.

The graphical user interface consists of an advanced scene graph visualization widget that dominates one of the two screens (see Figure 4.9). The center area can be configured to

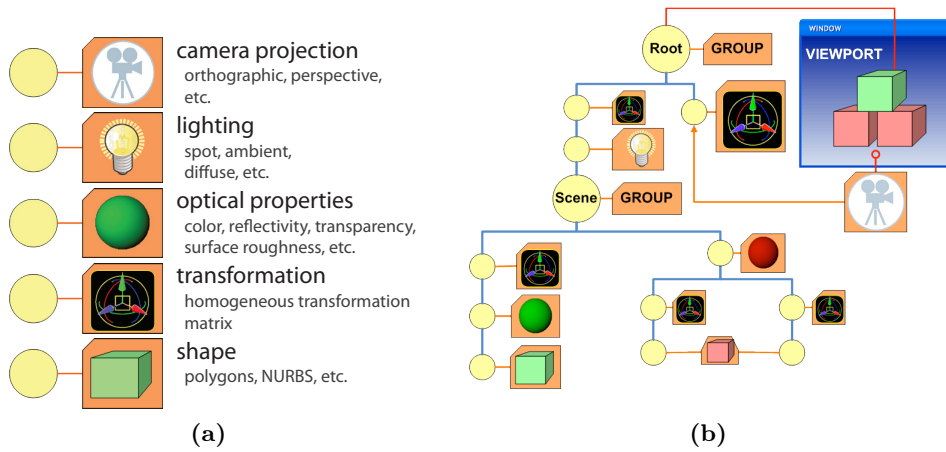


Fig. 4.7: Basic scene graph data elements (a) and an example representing a simple virtual scene (b).

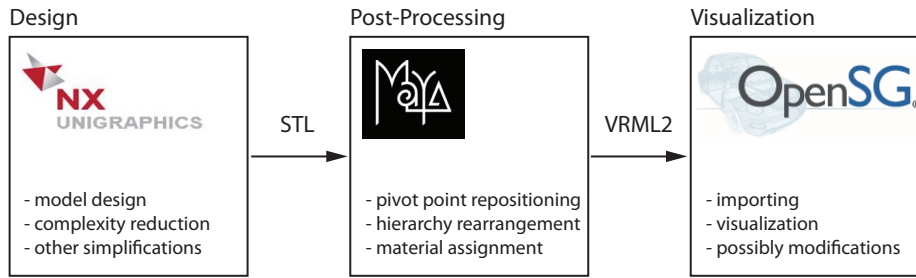


Fig. 4.8: Model conversion process from the CAD design program to the final scene graph.

show a single or four viewports, last with fixed camera orientations top, side, and front. Quick access buttons for zooming, hiding components, changing display modes, as well as screenshot and movie recording are accessible on the right. The scene graph can be displayed in a separate window on the left where there is also a control panel linked to the collision detection module is located (refer to Section 4.5 for more details).

4.4 VR Control Center

The core functionality and flexibility of the virtual reality environment is provided by the VR control center module. It basically handles and dispatches incoming motion commands for the visualization module (see Figure 4.10). The design of the VR control center enables free configuration of the mapping of incoming motion commands to outgoing actions in the transform control center. In order to define which element of a received command will control which degree of freedom of the 3D model, a simple XML configuration file (*joint control definition file*) is fed to the VR control center. In addition, it is possible to

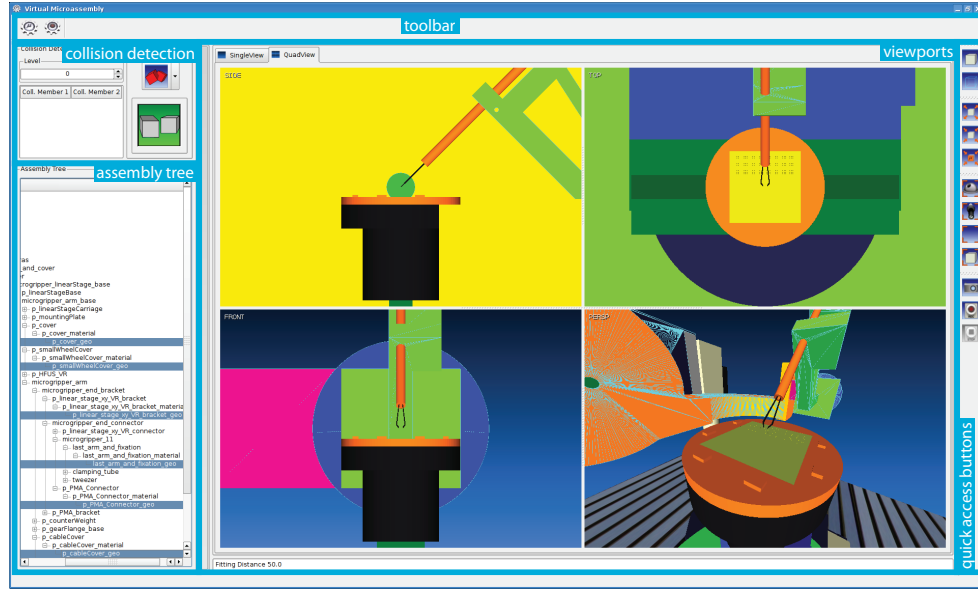


Fig. 4.9: Virtual reality visualization widget.

define joint limits (the range is limited to $[\theta_{min}, \theta_{max}]$) and joint constraints (a motion command can only be connected to a limited set of nodes in the transform control center $\{\theta_4\} \mapsto \{\eta, \zeta\}$) for every individual axis through XML files, too.

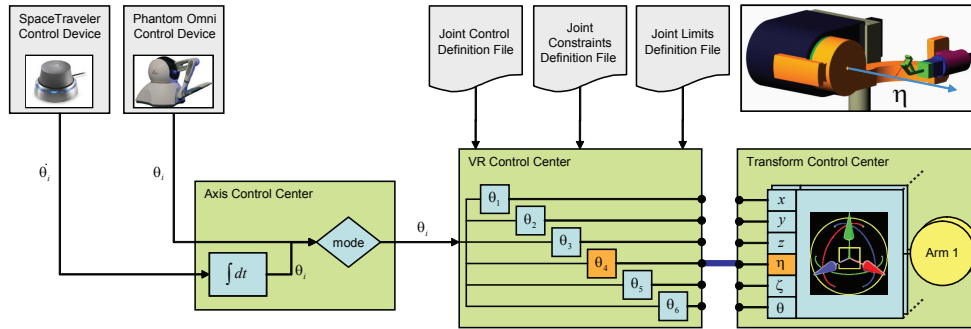


Fig. 4.10: Mapping motion commands to real axes in the VR control center.

The flexibility of this generic concept allows controlling not only the micro-assembly station V2, but also other robotic setups (see Figure 4.11), whereas exchanging the 3D model as well as the joint configuration, joint constraints, and joint limits files is sufficient. The present system implements the animation of the six degrees of freedom of the manipulator. All other axes, such as the ones for calibration (Section 3.2.5) and ring units (Section 3.2.6), are static within the virtual environment since they are not directly relevant to the manipulation procedure.

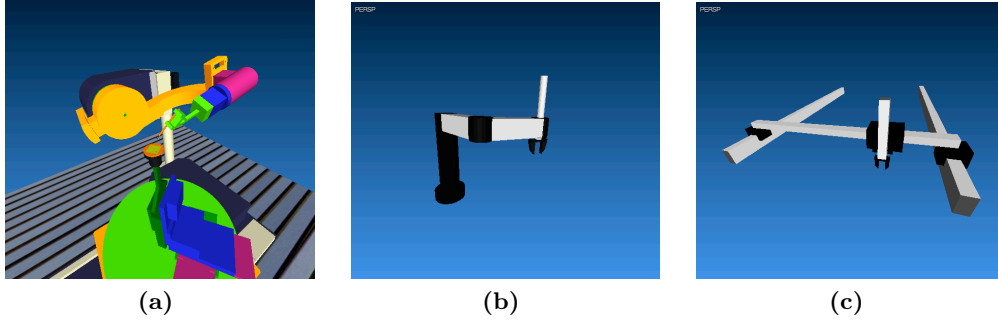


Fig. 4.11: The VR control center is built in a generic way so that it can handle various robot types from the micro-assembly station V2 (a) over a SCARA robot (b) to a gantry station (c).

4.5 Collision Avoidance

As already pointed out in Section 4.2, collision avoidance is a crucial component in this environment. Position, velocity or trajectory commands are first executed in the virtual reality environment and checked for collisions. Valid commands are then passed over to the hardware controller whereas invalid ones result in a visual and haptic notification of the operator. The collision detection library used here is named CollDet (Zachmann [179, 180], Zachmann and Weller [181]) and has the big advantage that it directly operates on an OpenSG scene graph.

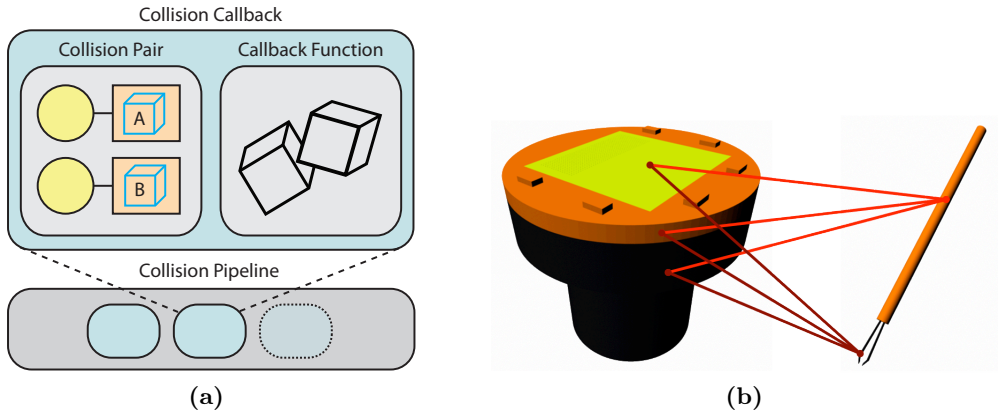


Fig. 4.12: Anatomy of a collision callback (a) and a set of collision pairs used in the present system (b).

Two individual model entities (*collision pair*) and a corresponding *callback function*, which defines the action to take upon a collision, define a *collision callback* (see Figure 4.12a for clarification). A set of collision callbacks is then inserted into a *collision pipeline* that sequentially executes collision detection on each member either on demand or continuously in a separate thread. The corresponding callback function is invoked if and only if a

collision has been detected. Apart from an appropriate colorizing of the objects involved, a notification is sent to the VR control center where further action is taken (i.e., the initial motion command is blocked) as shown in Figure 4.13.

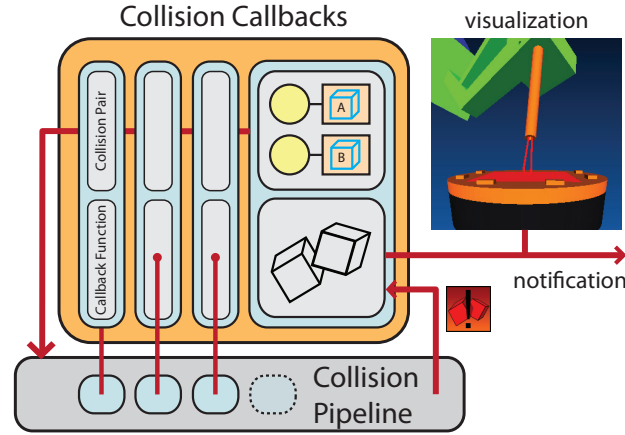


Fig. 4.13: Collision detection principle and integration into the virtual reality control center.

Collision detection is known to be a computationally intensive task where processing time mainly depends on the number of geometrical entities (collision pairs) as well as their complexity. The present configuration only includes objects that have a realistic potential for collision instead of the whole scene tree. Figure 4.12b shows a realistic set of six collision pairs which are dynamically loaded into the VR control center from an XML configuration file. The inclusion of micro-parts adds further entities to the queue. The model complexity, on the other hand, is dependent on the pre-processing step of the model conversion procedure (see Figure 4.8). This emphasizes again the importance of complexity reduction at this early stage.

4.6 Event Recording/Replaying

The virtual reality software environment contains a basic event recording and replaying mechanism. The recorder captures events from the haptic input devices and stores them in a file together with the corresponding timestamps (see Figure 4.14). The latter preserves the correct time-frame and sequence of commands for replaying. The format of the command file follows the PythonTM syntax where each line corresponds to a single function call in PythonTM.

The advantage of using PythonTM becomes apparent when looking at the replayer module (see Figure 4.15a). PythonTM's interpreter reads every single line from the data file and calls a type specific function of the replayer written in C++. Those functions create corresponding events and add them to an event pool for later execution. In order to call a C++ structure from PythonTM it has to be declared first in an event commands file (see

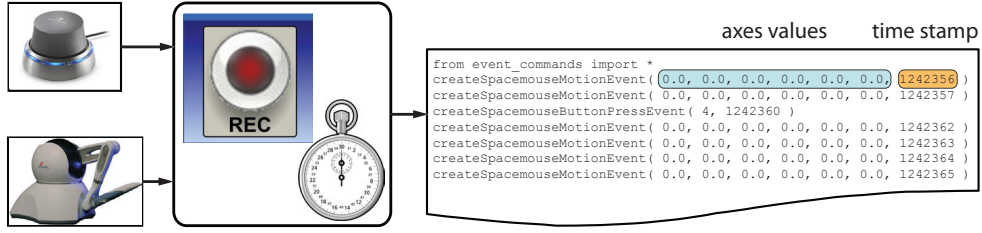


Fig. 4.14: Dataflow of the event recorder.

Listing 4.1: PythonTM module sample of the C++ replayer.

```
import Replayer python_replayer_ref = None
2
def import_cpp_instance( cpp_replayer_ref ):
4     print "importing cpp reference"
    global python_replayer_ref
6     python_replayer_ref = cpp_replayer_ref

8     def createSpacemouseMotionEvent( x, y, z, a, b, c, time_stamp ):
        if python_replayer_ref != None:
10             python_replayer_ref.addSpacemouseMotionEvent( x, y, z,
                                                            a, b, c, time_stamp, )
12     else:
        print "Error: Received null pointer!"
```

Listing 4.1 for an example). A third-party library then allows compiling C++ code into a PythonTM module for later importing. This makes almost all functionalities of a C++ class available to the PythonTM interpreter. See Figure 4.15b for details.

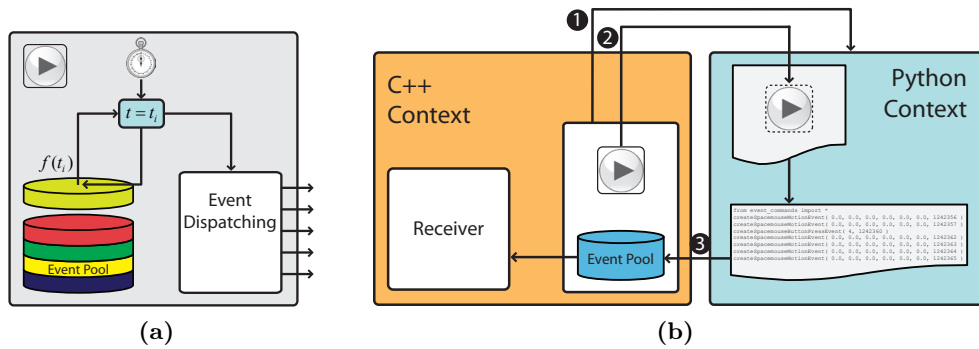


Fig. 4.15: Event replaying mechanism (a) and the interaction of the C++ and PythonTM contexts (b). Procedure: ① Start PythonTM interpreter and create PythonTM object, ② Declare replayer interface and pass C++ reference to PythonTM, ③ import session file as PythonTM module.

So far, commands for handling haptic input events have been defined within the PythonTM module. However, the extension to additional functionalities only requires the adaption of the PythonTM module, and, depending on the situation, the corresponding C++ code.

4.7 Task Planning

The same procedure used for the recorder-replayer mechanism can also be employed for use in a task planner. Instead of using the defined haptic input events in the PythonTM context, one can define additional functions, such as `moveTo()` or `pickPart()`, so that complex assembly operations can be coded directly in PythonTM. However, the proper integration of task planning also requires advanced path planning capabilities and thus requires additional modifications and extensions to the current system.

4.8 Calibration and Synchronization

The alignment of the virtual world with the real one is absolutely necessary for this setup. It can be separated between the initial alignment, known as calibration or initialization, and the continuous adjustment, referred to as synchronization. Any difference between the two worlds is called synchronization error ϵ_{sync} .

Assuming that the six manipulator axes are not affected by drifting effects (i.e., DC motor axes) and the stepper drives execute the commanded steps, ongoing synchronization can be neglected. The resulting synchronization error is then in the order of the accuracy and precision of the individual stages (see Section 6.4).

On the other hand, the initial alignment is less than trivial and the root for any future inaccuracies. The actual initialization routine simply centers all axes in their defined zero position and resets the position counters (encoders). Since the virtual world is continuously updated at defined time intervals according to the real manipulator values, it automatically reaches correct position and orientation. The initialization of the manipulator as well as the whole system is described in Section 3.5.

4.9 Performance

Figure 4.16 shows the results of a manipulation experiment using the micro-assembly station V2 hardware. Velocity commands c_v , generated by two haptic input devices (see Section 3.4.4), are processed by the virtual reality environment as previously explained and sent to the hardware controller over Ethernet. The resulting position measurement m is sent back to the workstation for updating the virtual model. For comparison, the integral of the velocity command s is also shown. In order to compensate for the low pass filter characteristics of the whole information flow, the input signal c_v is low pass filtered with a cutoff frequency of $f_c = 50$ Hz. Even though there is a lag of around 0.5 s between

command and result, the handling feels natural and the results are within expectance. As already mentioned, the system lag is due to collision detection as well as the Ethernet-based client-server data transfer scheme, which also involves traffic from and to the vision server. Additionally, Figure 4.16 reveals the fact that certain commands are omitted. This is due to the current configuration of the Player driver on the hardware controller which overwrites “old” commands in the queue with new ones.

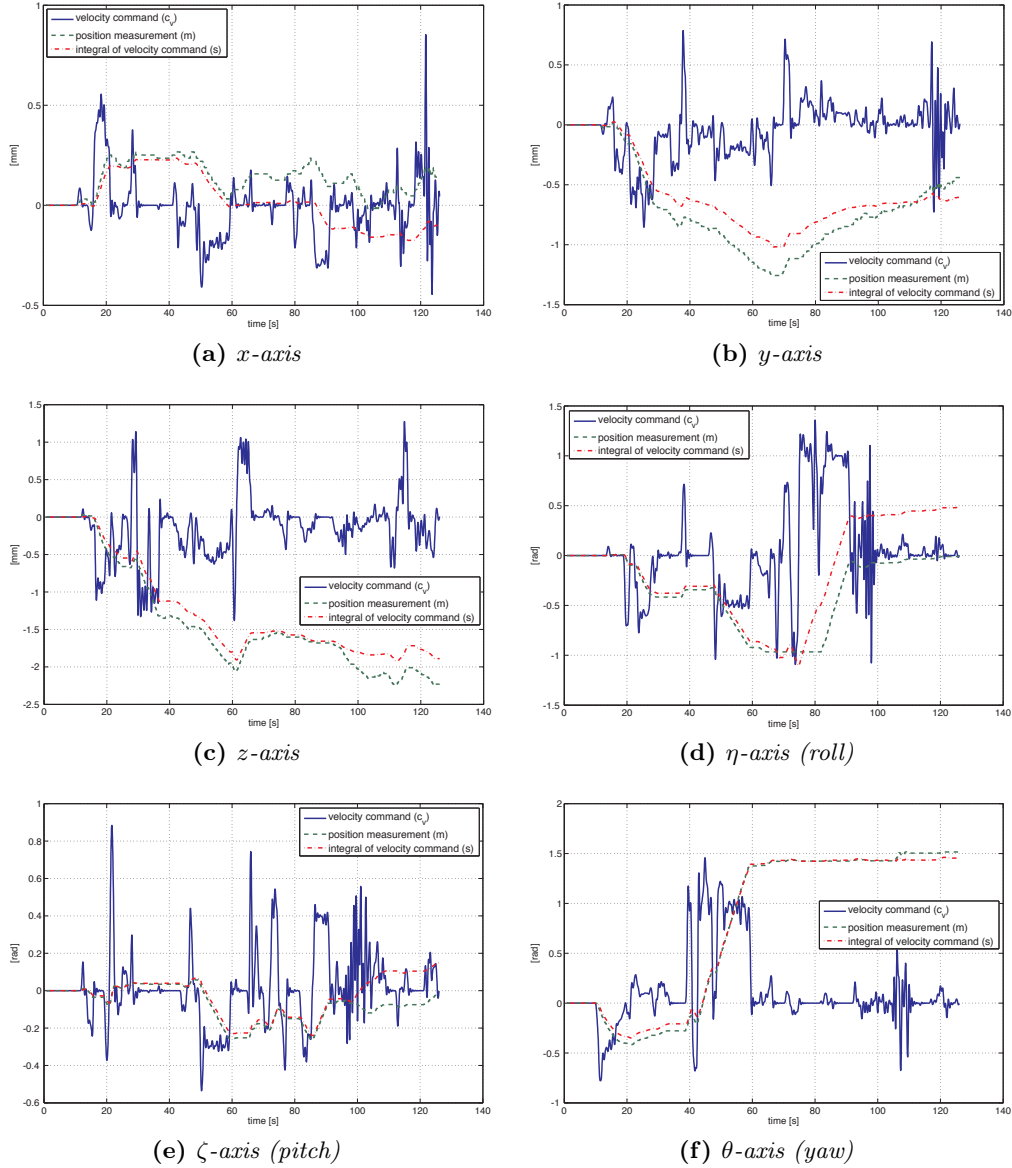


Fig. 4.16: Velocity commands c_v from an input device sent to the motor controller, its integral $s(t) = \int_{t=0}^x c_v dt$ and the measured position feedback of the main axes ($x, y, z, \eta, \zeta, \theta$). The input velocity commands have been filtered using a Butterworth lowpass filter ($n = 9$, $f_c = 50$ Hz).

4.10 Summary and Conclusions

The development and integration of a virtual reality environment is mandatory for providing a simple gateway to a rather complex machine such as the micro-assembly station V2. The present VR setup features an open source scenegraph package and a series of specially written tools allowing to visualize any translated 3D object in virtual space. The freely configurable motion mapping functionality opens this library for other projects and kinematic configurations. It also features collision avoidance and an event recording/replaying mechanism, which can be extended to a task planner in the future. Even though it is not (yet) possible to fully renounce live camera images, the power of virtual reality is obvious and essential for more and more complex assembly tasks.

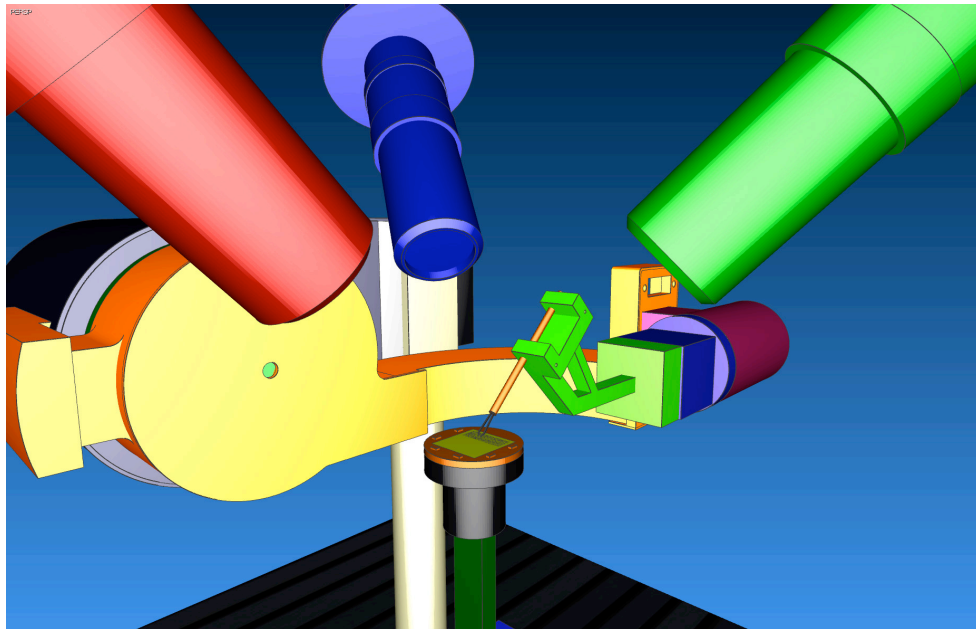


Fig. 4.17: *Closeup view of the virtual assembly area.*

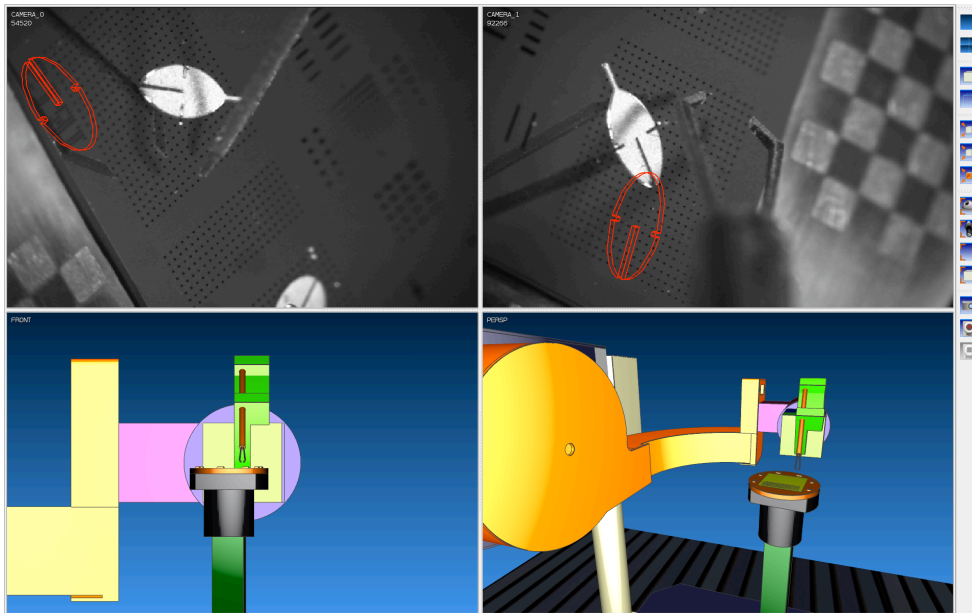


Fig. 4.18: *Virtual reality interface showing two live camera streams.*

Model Based Pose Estimation

This chapter elaborates the development and integration of a pose estimation method. After going into the details of the algorithm, in-depth information about image feature extraction, model preparation, as well as correspondence matching is given. The chapter concludes with details about implementation and potential improvements.

5.1 Introduction

The terms pose estimation, localization, and tracking have slightly inconsistent meanings in literature. In the following context they are used as follows:

(Initial) Pose estimation Retrieval of position and orientation with respect to an observing camera that results in a best fit of a 3D model representation to observed 2D image features. The search space is defined by the 2D image.

Localization More or less identical to pose estimation. The term tends to emphasize the process of actually locating an object in an unknown environment.

Tracking Identical to pose estimation but the search space is limited by the tracking assumption which states that the new pose T_t is in a definable vicinity of the previous pose T_{t-1} .

(Initial) pose estimation or localization routines are usually executed as a first step in order to limit the search space and provide a starting point for the tracking scheme (see Figure 5.1). While tracking is a continuous process running in real-time, the other task is usually executed once in a while.

Pose estimation itself deals with two major problems. The first one is the correspondence problem and often neglected when dealing with the geometric aspect only. It deals with matching extracted 2D image features to 3D model data and becomes the key issue when dealing with a large search space. The second problem is about solving for the pose $T = (R \mid \mathbf{t})$, that leads to a best fit of the model dataset with the actual extracted image entities.

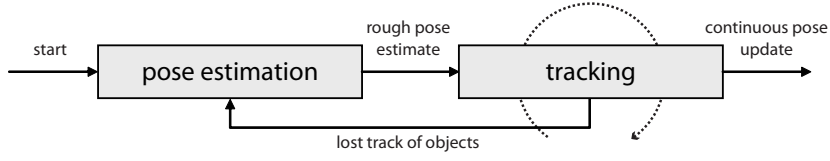


Fig. 5.1: Pose estimation routines are generally run in a single sequence at the beginning and followed by a continuous tracking algorithms. However, they can be initiated again as soon as tracking is lost.

5.2 Pose Estimation Algorithm

5.2.1 Strategies

There are two strategies for bringing 2D images features to congruence with 3D model data. The first method projects 3D model data to the image plane where it is compared with the extracted 2D image features. Depending on how this comparison in the projective or Euclidean plane is mathematically formulated, it results in non-linear or badly-conditioned equations or a loss in a distance measure. The second method projectively constructs 2D image features to 3D space where they are compared with the 3D model. This strategy is more efficient since it does not require the estimation of a projection for comparison in the image plane in each iteration step (see Figure 5.2). In addition, it has been shown that the formulation of constraints is simpler in 3D space and additionally provides an intuitive error measure that is directly linked to a spatial distance.

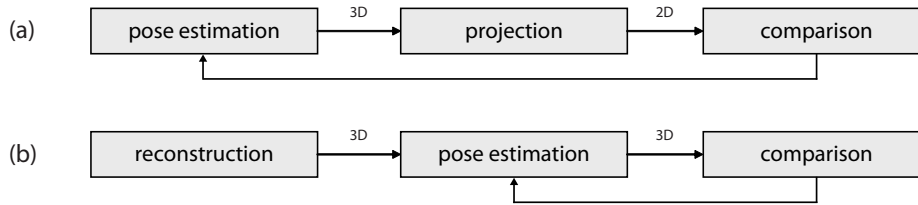


Fig. 5.2: Pose estimation strategies: comparison in the image plane (a) and in 3D space (b).

The following sections describe a scheme for solving the pose estimation problem, which has been developed by Rosenhahn [128]. Since this research is differently oriented, only the bare minimum of equations required for understanding the concept is presented. Readers interested in reproducing the mathematical foundations, which are related to the fields of geometric algebra, Plücker geometry, Lie groups, Lie algebras, etc., are recommended to consult Rosenhahn [128].

5.2.2 The Algorithm

The solution to the 2D-3D pose estimation problem is based on an algorithm developed by Rosenhahn [128]. The method overcomes the unsuited description of projective geometry

and kinematics in Euclidean space by transforming geometric entities to and solving constraint equations in the projective and conformal spaces. The constraint equations mathematically connect 3D model and 2D image features, where the latter are projectively constructed to three-dimensional entities. A big advantage of this method is that it directly yields a 3D error measure that can be used for assessing the pose result.

Three correspondence types for simple geometric features are available. They relate points to lines (PL-constraints), points to planes (PP-constraints) and lines to planes (LP-constraints). The first two types have been implemented within the scope of this thesis and are visualized in Figure 5.3.

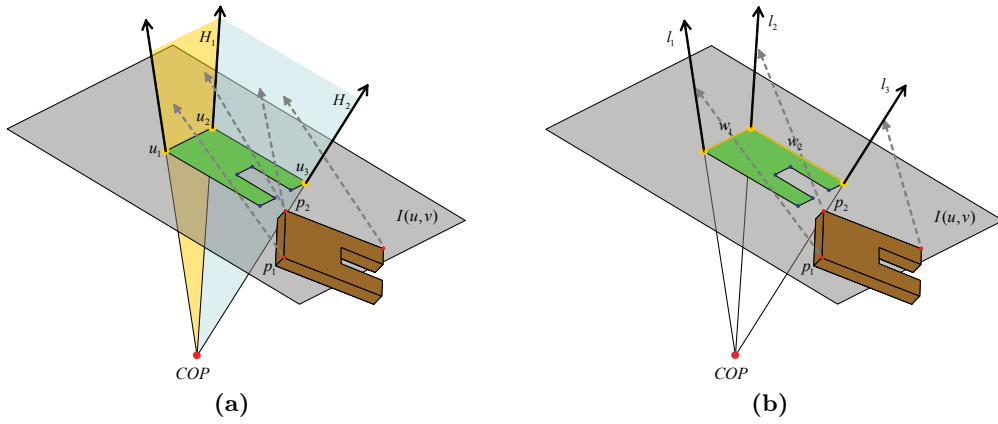


Fig. 5.3: Visualization of point-line (a) and point-plane (b) constraints.

Point-Line Constraints

Points (u_1, u_2, \dots, u_n) are extracted from the source image and used for constructing lines (l_1, l_2, \dots, l_n) with the common intersection point COP . The collinearity of a point $p_i = (p_{ix}, p_{iy}, p_{iz})$ with a line $l_i = \{(ld_{ix}, ld_{iy}, ld_{iz}), (lm_{ix}, lm_{iy}, lm_{iz})\}$ given in Plücker coordinates (see Appendix F.1) can then be written as a system of linear equations of the form $A_i x = b_i$ with

$$A_i = \begin{pmatrix} 0 & ld_{iz} & -ld_{iy} \\ -ld_{iz} & 0 & ld_{ix} \\ ld_{iy} & -ld_{ix} & 0 \\ -p_{iz}ld_{iz} - p_{iy}ld_{iy} & p_{ix}ld_{iy} & p_{ix}ld_{iz} \\ p_{iy}ld_{ix} & -p_{ix}ld_{iz} - u_{iz}ld_{iz} & p_{iy}ld_{iz} \\ p_{iz}ld_{ix} & p_{iz}ld_{iy} & -p_{iy}ld_{iy} - p_{ix}ld_{ix} \end{pmatrix} \quad (5.2.1)$$

and

$$b_i = \begin{pmatrix} -p_{iy}ld_{iz} + p_{iz}ld_{iy} + lm_{ix} \\ -p_{iz}ld_{ix} + p_{ix}ld_{iz} + lm_{iy} \\ -p_{ix}ld_{iy} + p_{iy}ld_{ix} + lm_{iz} \end{pmatrix} . \quad (5.2.2)$$

Note that every equation $A_i x = b_i$ stands for a single point-line constraint. Out of the three equations only two are linearly independent ($rank(A) = 2$) and thus at least three point-line constraints are required for a valid solution.

Point-Plane Constraints

Lines (w_1, w_2, \dots, w_n) are extracted from the source image and used for constructing planes (H_1, H_2, \dots, H_n) with the common intersection point *COP*. The coplanarity of a point $p_j = (p_{jx}, p_{jy}, p_{jz})$ with a plane $H_j = \{(Hn_{jx}, Hn_{jy}, Hn_{jz}), Hd_j\}$ given in Hessian normal form (see Appendix F.2) can then be written as a system of linear equations of the form $A_j x = b_j$ with

$$A_j = \begin{pmatrix} -Hn_{jx} & -Hn_{jy} & -Hn_{jz} \\ -Hn_{jz}p_{jy} + Hn_{jy}p_{jz} & Hn_{jz}p_{jx} - Hn_{jx}p_{jz} & -Hn_{jy}p_{jx} + Hn_{jx}p_{jy} \end{pmatrix} \quad (5.2.3)$$

and

$$b_j = -Hd_j + Hn_{jx}p_{jx} + Hn_{jy}p_{jy} + Hn_{jz}p_{jz} . \quad (5.2.4)$$

Note that every equation $A_j x = b_j$ stands for a single point-plane constraint. At least six point-plane constraints are required for a valid solution.

Both constraint types can either be individually used or combined yielding the final system of equations

$$\begin{pmatrix} A_1 \\ \vdots \\ A_n \end{pmatrix} x = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \quad (5.2.5)$$

which can for example be solved using least squares approximation. Vector x contains scaled twist parameters

$$x = \begin{pmatrix} m_1 \\ m_2 \\ m_3 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \quad (5.2.6)$$

from which the twist parameters θ , ω and m can be calculated as follows

$$\theta = \sqrt{\omega\omega^T} \quad (5.2.7)$$

$$\omega' = \frac{\omega}{\theta} \quad (5.2.8)$$

$$m' = \frac{m}{\theta} . \quad (5.2.9)$$

The rigid body motion parameters R and \mathbf{t} can then be evaluated using the formula of Rodrigues (see Appendix C).

5.2.3 Pose Error

The goal of the pose estimation algorithm is the minimization of the pose error which strongly depends on the correspondence set as well as on the quality of the approximation of the rigid body motion. As already mentioned this algorithm calculates the true Euclidean pose error in 3D space which is essentially the distance between the “updated” model points and the constructed lines and/or planes depending on the types of correspondences used. See Figure 5.4 for details.

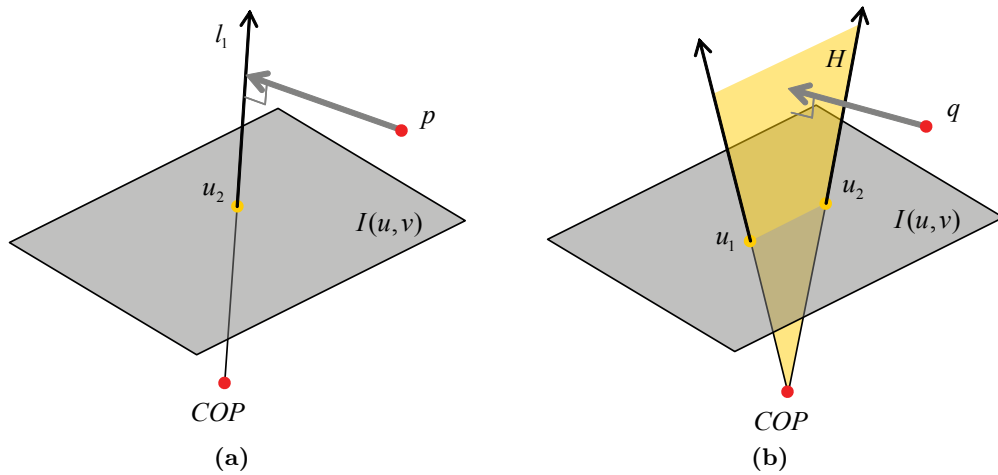


Fig. 5.4: Pose error for point-line (a) and point-plane (b) constraints.

For the point-line constraints, the Euclidean distance between a point p and a line l' is given by

$$\|p_2\| = \|\theta' - p \times \omega'\| \quad (5.2.10)$$

where $\theta' = \lambda\theta$ and $\omega' = \lambda\omega$ with the scaling factor $\lambda = \frac{1}{\|\omega\|} \in \mathbb{R}$ being the inverse norm of the direction vector (see Figure 5.5a). For the point-plane constraints, the Euclidean distance between a point q and a plane H is given by

$$h'_{d2} = h'_d - \mathbf{n}' \cdot q \quad (5.2.11)$$

where $h'_{d2} = \lambda h_{d2}$ and $\mathbf{n}' = \lambda \mathbf{n}$ with the scaling factor $\lambda = \frac{1}{\|\mathbf{n}\|} \in \mathbb{R}$ being the inverse norm of the normal (see Figure 5.5b).

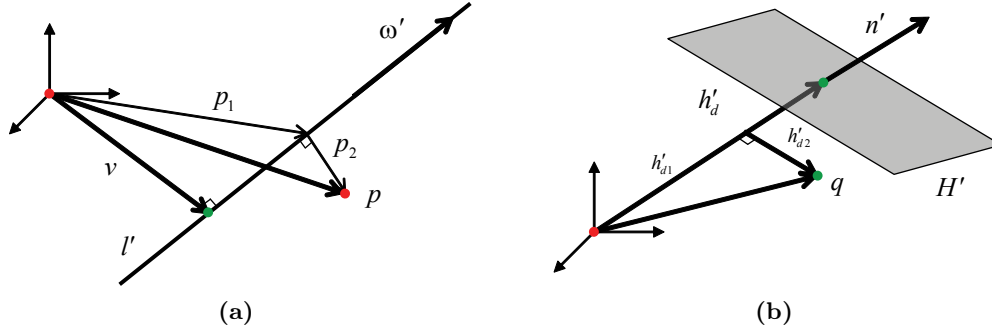


Fig. 5.5: Euclidean distances between a point p and a line l (a) as well as a point q and a plane H (b).

Summing up the individual distance values of each correspondence pair yields a first absolute error estimate

$$E_{abs} = \sum_{i=1}^m e_i \quad (5.2.12)$$

where e_i is the error value for a correspondence set i and m is the total number of model features. In order to compensate for the tendency that a small amount of correspondences results in a lower spatial error than a large amount, a relative pose error E_{rel} is defined as

$$E_{rel} = \frac{1}{n} E_{abs} = \frac{1}{n} \sum_{i=1}^m e_i \quad (5.2.13)$$

with n being the total number of correspondences. The problem is that a minimal amount of correspondences can lead to degenerate poses with very low relative errors. However, the

probability that the current estimated pose represents the true model location increases with increasing number of correspondences. Thus, the number of correspondences n is weighted with a constant σ that attenuates its effect for higher numbers. The final pose error E can then be written as

$$E = \frac{1}{n + \sigma} \sum_{i=1}^m e_i . \quad (5.2.14)$$

However, in most cases the pose error is not sufficient enough to filter bad matches. Degenerate configurations often lead to low error values as shown in the comparison in Figure 5.6. It is thus necessary to include additional measures that monitor the validity of a “good” pose. One possible solution related to the error measure is explained in the following section.

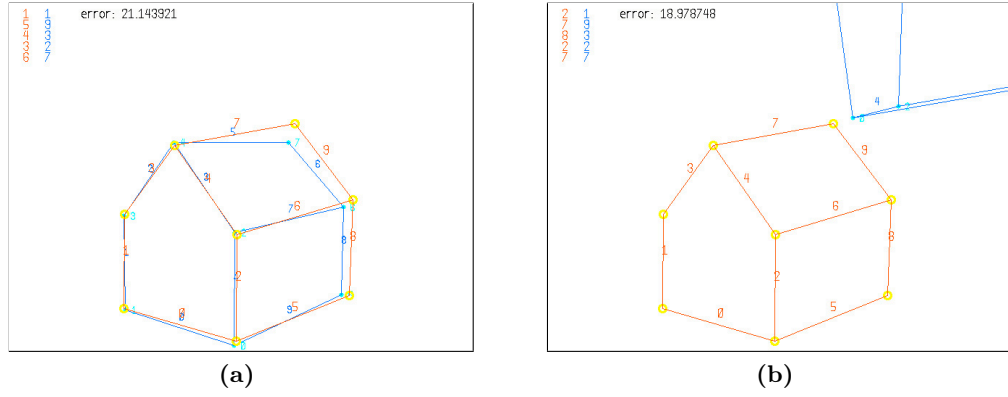


Fig. 5.6: 3D pose error for correct (a) and degenerate pose (b).

5.2.4 Center of Mass Error

One way to eliminate degenerate poses is the evaluation of the distance between the center of mass of the image and the center of mass of the model at the current position and orientation. This concept is visualized in Figure 5.7. The center of mass of a discrete system is defined as

$$\mathbf{r}_{cm} = \frac{\sum m_i \mathbf{r}_i}{\sum m_i} \quad (5.2.15)$$

where \mathbf{r}_i is the vector of point i and m_i its associated mass ($m_i = 1$ in this case). First, the 2D center of mass $\mathbf{r}_{cm,img}$ of the relevant extracted image features is calculated and its ray l_{COM} projectively constructed. Next, the 3D distance between the center of mass $\mathbf{r}_{cm,mod}$ of the model at the current location and l_{COM} is evaluated analogue to the calculation of the pose error shown in Section 5.2.3.

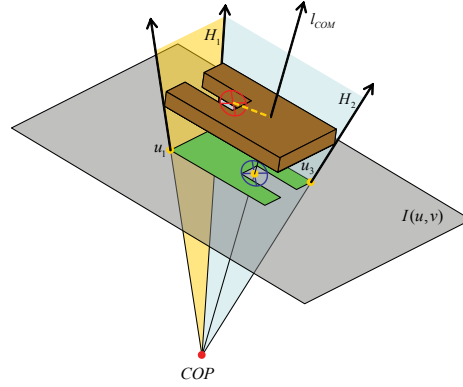


Fig. 5.7: Visualization of the center of mass error.

Due to noise in the source image and vision processing inaccuracies the center of masses will never exactly match. However, a carefully set threshold forces the algorithm to pose the model within the vicinity of the best fit.

This section explained the calculation and quantification of a pose under the assumption that a 3D model and image features are given as well as a set of correspondences are a priori known. Methods for preparing and calculating these input conditions for the pose processor are described in the following sections.

5.3 Image Feature Extraction

The two constraint types PL and PP define a limit on the types of objects that have to be extracted from the source images. Since PL constraints construct lines, points need to be extracted. On the other hand, PP constraints construct planes and thus require either lines or point pairs.

5.3.1 Initial Experiments with Line Extraction

Some initial experiments have been done with two objects that consist of a series of straight lines. The first one is a L-shaped metal piece (see Figure 5.8a) and the second a little model house with a single door (see Figure 5.9a).

For these experiments the Hough transform has been used. The Hough transform is a linear transformation for detecting straight lines. A straight line can be described as $y = mx + b$ in the image space. Due to the fact that parameters m and b go to infinity for vertical lines, lines are thus parameterized in their polar form as

$$y = \left(-\frac{\cos \theta}{\sin \theta} \right) x + \frac{r}{\sin \theta} \quad (5.3.1)$$

$$r = x \cos \theta + y \sin \theta \quad (5.3.2)$$

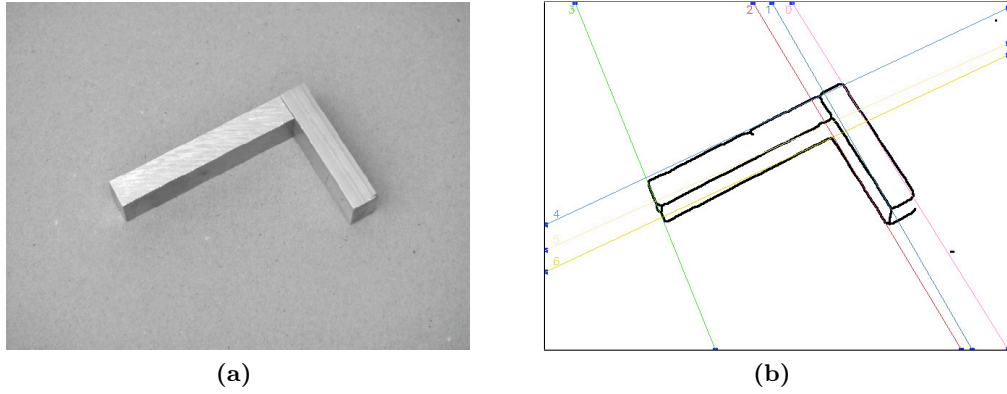


Fig. 5.8: *L-shape (a) and extracted Hough lines (b).*

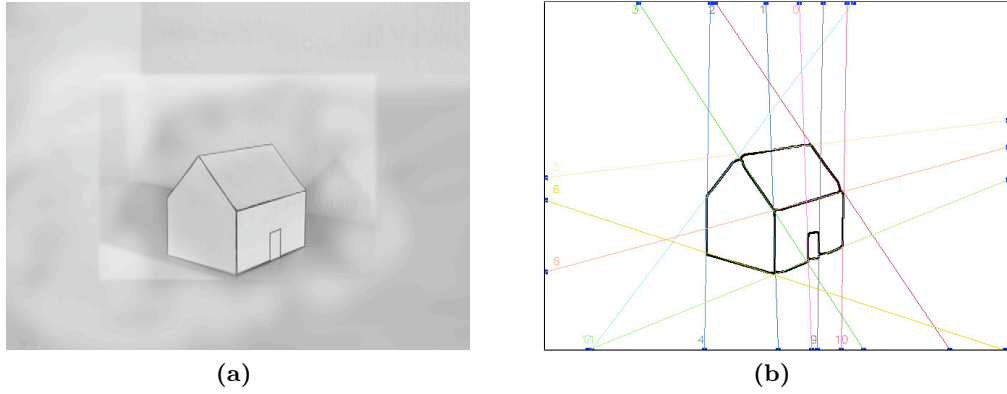


Fig. 5.9: *House (a) and extracted Hough lines (b).*

where r is the distance from the origin to the line and θ is the orientation of r with respect to the x -axis. Each line can now be represented as a pair of (r, θ) which is unique if $\theta \in [0, \pi]$ and $r \in \mathbb{R}$, or if $\theta \in [0, 2\pi]$ and $r \geq 0$. The (r, θ) -plane is referred to as Hough space for the set of straight lines in two dimensions.

Every point $p = (x, y)$ of a binary input image votes for all parameter pairs (r, θ) that could have possibly created it, i.e., for all lines passing through p . For every point of the binary image a curve in an accumulator array is drawn by increasing the corresponding values. The maxima in the accumulator array are candidates for lines in the image and can be found by thresholding.

Figures 5.8b and 5.9b show the result of the Hough transform for the sample images. Coordinates of these lines are then used for constructing the planes needed for the PP constraints. This preliminary version provides fast results within less than a second and is pretty robust even under difficult conditions.

5.3.2 Microrobot Component Extraction

Extracting microrobot components out of camera images is central to this project and thus explained in more detail. The components of the bio-microrobotic device of interest are shown in Figure 5.3. The first model consists of an elliptic body with (BP) and without (B), or two without (B) a pipette. Depending on the application two ribs (R) can be perpendicularly attached to each side. A second robot type consists of two rectangular elements (BR) and no ribs, at all. The efficient extraction of points and lines needed for the pose estimation step is described in the following list:

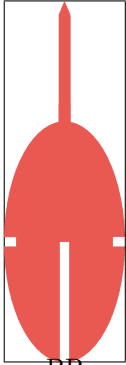
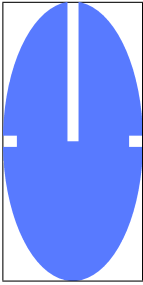

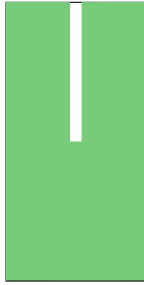
body pipette		body	rib	body rectangle
				
A_{rc}	$1 - A_{rc} \geq 0.1$	$1 - A_{rc} \geq 0.1$	$1 - A_{rc} \geq 0.1$	$1 - A_{rc} < 0.1$
S_{ab}	$S_{ab} < 0.6$	$S_{ab} - 0.5 < 0.05$	$S_{ab} > 0.75$	–
A_{ce}	$A_{ce} < 0.75$	$A_{ce} > 0.9$	$A_{ce} - 0.5 < 0.05$	–
A_{he}	$A_{he} > 0.8$	$A_{he} > 0.8$	–	–

Table 5.1: Relevant parameters of individual microrobot parts (ratio of the area of the minimal enclosing rectangle to the area of the contour A_{rc} , ratio of the semi-minor and semi-major axes of the ellipse S_{ab} , ratio of the contour area to the ellipse area A_{ce} , and ratio of the area of the convex hull to the ellipse area A_{he}).

1. The source image is smoothed by convolution with a 9×9 Gaussian kernel ($\sigma = 1.85$).
2. Hysteresis thresholding can be used for increasing the dynamic range of the input image and to automatically detect a reasonable threshold for the next step.
3. Next, a binary threshold filter is applied, where the relation of an output pixel u_d to an input pixel u_s is given by

$$u_d(u, v) = \begin{cases} I_{max} & \forall u_s(u, v) > I_{thresh} \\ 0 & \forall u_s(u, v) \leq I_{thresh} \end{cases}$$

4. Then, a contour finder extracts all external and closed contours from the binary image returning individual sequences of image points $C_i = \{u_{i1}, u_{i2}, \dots\}$. A mean threshold area \bar{A} is calculated by

$$\bar{A} = \frac{1}{n} \sum_{c=C_1}^{C_n} A_c$$

where A_c is the area of the current contour c . This is used for a preselection so that contours with an area lower than \bar{A} are neglected.

5. The convex hull as well as the relative convexity defect $\bar{d}_{c,i}$ are calculated for each contour C_i (see Appendix D.1). If $\bar{d}_{c,i}$ is below a given threshold $d_{c,rel}$ the contour is neglected. This ensures that low quality objects are rejected at an early stage.
6. Next, the minimal enclosing rectangle and its area for each contour are evaluated. The ratio of the area of the minimal enclosing rectangle to the corresponding contour area $A_{rc,i}$ distinguishes between rectangular and elliptic/circular components (see Table 5.1).
7. The features of the rectangular shape are then extracted by detecting the outer corners as well as the position of the slit. Sub-pixel corner refinement yields a set of maximal 16 vertices and 24 edges.
8. On the other hand, the elliptic/circular category is further analyzed by fitting an ellipse $E_{C_{h,i}}$ on the previously extracted convex hull $C_{h,i}$.
9. The following parameters based on $E_{C_{h,i}}$ are calculated and finally categorize the elliptic shapes as listed in Table 5.1.
 - A_{ce} : Ratio of the contour area to the ellipse area
 - A_{he} : Ratio of the area of the convex hull to the ellipse area
 - S_{ab} : Ratio of the semi-minor to the semi-major axis of the ellipse
10. Additional geometric evaluations finally yield the positions of the slits so that the image coordinates (u, v) of vertices and edges can be calculated.
11. Since vertex accumulations in small areas often cause degenerate poses, a simple proximity check is performed. It basically reduces vertex clouds with a given maximum radius d_{prox} to a single vertex.
12. The image processing ends with a list of the (u, v) -coordinates of unique image points and a list of tuples containing indices to the point list specifying edges.

Figure 5.10 shows the result of the microrobot extractor algorithm for an idealized dataset as well as a real camera image. Figure 5.11

5.4 3D Model Preprocessing

5.4.1 Model Conversion

The components of the bio-microrobotic device of interest are again shown in Table 5.3. The main device consists of an elliptic body with (BP) and without (B), or two without

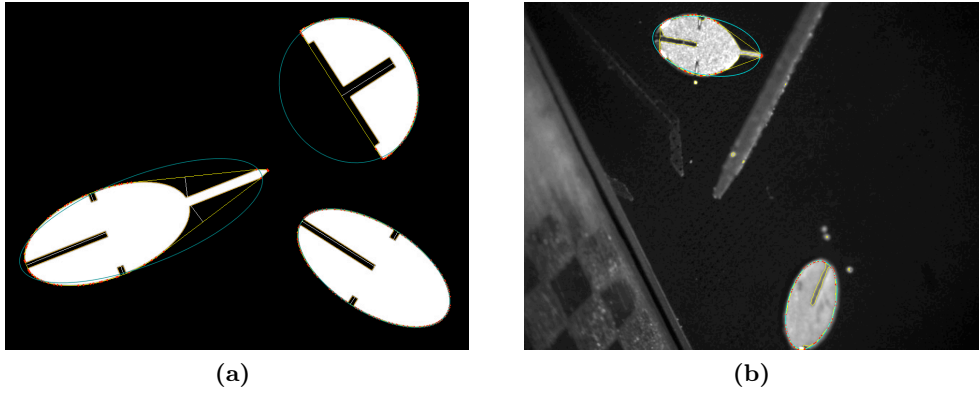


Fig. 5.10: Outer contour, convex hull and fit ellipse for an idealized case (a) and two real microrobot parts (b). The orange line represents the actual contour and the yellow line is the convex hull which the blue ellipse is fitted on.

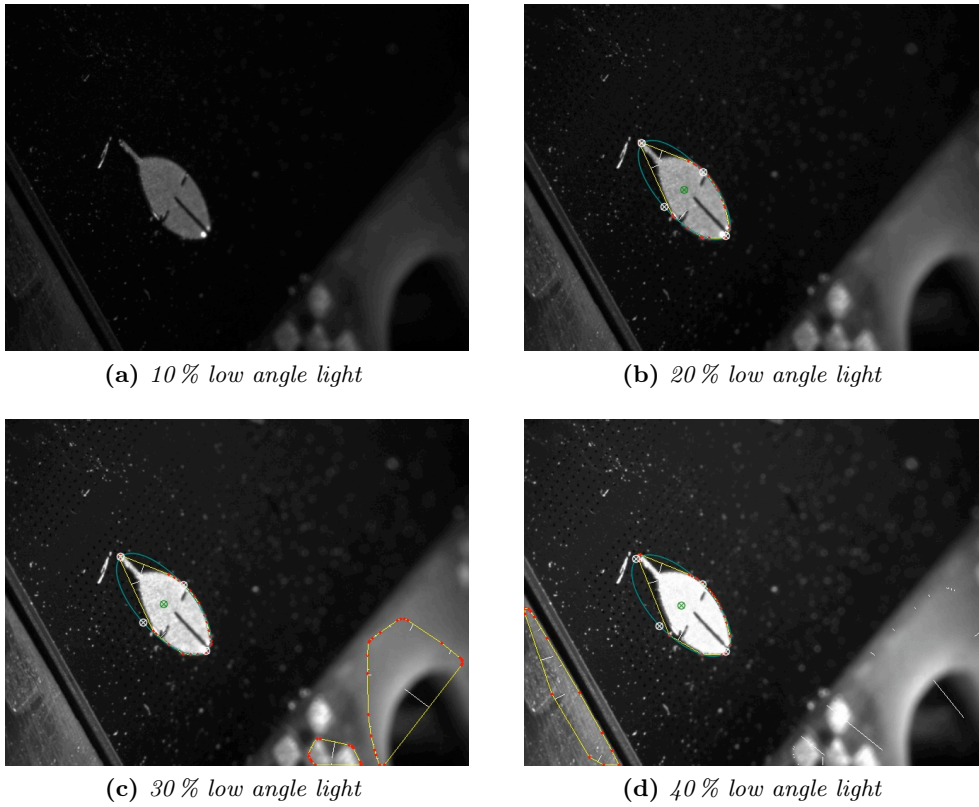


Fig. 5.11: Microrobot extraction for different power settings of the low angle illumination (see Section 3.2.6). A power setting within an optimal range yields one single match (b). This range depends on many factors, such as the object itself, the workbench surface, dust particles, etc. Lower or higher power output results in either no match (a) or multiple possibilities (c), respectively. However, the routine is able to discard false matches as shown in (c) and (d).

parameter	default	unit	description
robotType		–	type of robot to be searched for
$d_{c,rel}$	10.0	pixel	relative defect threshold
d_{mr}	0.1	mm	robot thickness
h_{mr}	1.5	mm	robot height
w_{mr}	3.0	mm	robot width
w_{pip}	0.1	mm	pipette width
d_{prox}	0.1	pixel	proximity threshold

Table 5.2: Summary of the microrobot extraction parameters.

(B) a pipette. Depending on the application two ribs (R) can be perpendicularly attached to each side. A second robot type consists of two rectangular elements (BR) and no ribs, at all.

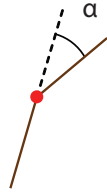
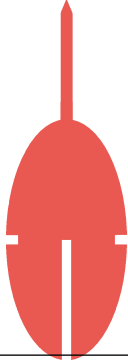
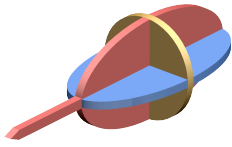
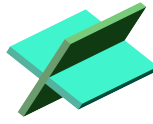
	body pipette	body	rib	body rectangle
				
designator	BP	B	R	BR
vertices	66	58	38	16
edges	80	70	48	24
normals	23	25	14	6

Table 5.3: Microrobot component parameters. Vertex, edge, and normal count are given for a simplified model where invisible and silhouette edges for $\alpha < 5^\circ$ have been removed.

Preprocessing of the model data is required for the pose estimation routine. The procedure can be subdivided into offline and online steps. The offline elements include exporting from the CAD program, adding of material, setting up a simple scene tree, etc., and finally converting the files to a binary object format OBJ that can be directly imported by the main application. The latter performs a series of online conversions, out of which

the complexity reduction is the most important step. Models are commonly stored as a series of triangular meshes (boundary representation) as shown in Figure 5.12a, and are primarily used for rendering purposes. The contours or sharp edges (see Figure 5.12b) are creating the actual shape of the model and thus most likely the ones that can be extracted from a camera image. The silhouette edges (see Figure 5.12c) are only visible if they belong to the object's boundary at the current position and orientation. Finally, the invisible edges (see Figure 5.12d) can be completely neglected since they do not contain any useful information for pose estimation.

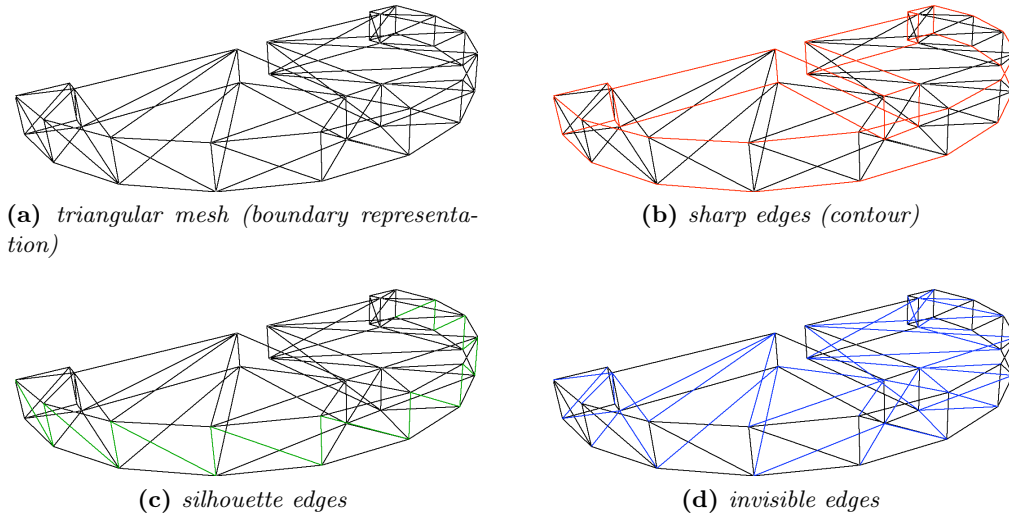


Fig. 5.12: 3D model edge classification based on Yeşin [178].

Edge classification is performed by analyzing the angle α of the normals of two adjacent triangles as follows:

$$\text{edge} = \begin{cases} \text{invisible} & \forall \alpha = 0 \\ \text{silhouette} & \forall 0 < \alpha \leq 70^\circ \\ \text{sharp} & \forall \alpha > 70^\circ \end{cases} \quad (5.4.1)$$

Table 5.3 summarizes the vertex, edge and normal count of the preprocessed 3D models.

5.4.2 Additional Complexity Reduction

MEMS fabricated microrobot components are 2.5D and thus all vertices can be categorized into two groups depending on their z -value. Due to the fact that the ratio of a planar dimension (width or height) to the thickness of MEMS fabricated microrobot components is in the order of 50, it is virtually impossible to extract features on both z levels, even if observing from a low angle. In other words, extracted image features are coplanar and

thus the model size can be reduced by only processing coplanar vertices. Splitting the number of model features in half results in a dramatic reduction of the search space and thus computation time, as shown in the next section.

5.5 Correspondence Matching

5.5.1 Problem

It has already been explained at the beginning of this chapter that localization consists of the two subproblems *correspondence matching* and *estimation of the pose parameters*. They both result in an optimization task with the final common goal of reducing a defined error measure, i.e., the pose error (see Section 5.2.3). Given a set of model M and a set of image features B , the space of all possible correspondences can be written as $S = M \times B$. A set of correspondences is a subset $s \subseteq S = M \times B$, that defines a set of constraint equations and thus a model pose. It can also be written as a map $f : M \mapsto B$ such that for every $m_i \in M$ there is a unique object $f(m_i) \in B$. The solution to the optimization problem finds the subset s that minimizes the pose error.

5.5.2 Search Space Dimensions

The set of subsets of a set S is called the power set of S , and the total number of distinct k -subsets on a set of n elements is given by the binomial sum

$$\mathcal{P}(S) = \sum_{k=0}^n \binom{n}{k} = 2^n . \quad (5.5.1)$$

For the current subset the number of elements can be written as $n = |M \times B|$ and thus the set of all subsets yields

$$\mathcal{P}(S) = 2^{|M \times B|} . \quad (5.5.2)$$

Given 12 model entities and 10 image features the search space has size $\mathcal{P}(S) = 2^{12 \times 10} = 1.33 \cdot 10^{36}$. Assuming an optimistic computation time of 1 μ s for one step results in a total of $4.2 \cdot 10^{22}$ years for the whole set. Reducing time consumption and thus overall performance can be done by (A) reducing the initial search space, or (B) by improving the search strategy, or both.

The search space can for example be reduced by limiting the number of model features than can correspond to an image feature and vice versa. Table 5.4 shows a list of possible limitations that can be applied. Limiting the search space of the example above to 1 : 1 relations, the size of the search space reduces to $\mathcal{P}(S) = \frac{12!}{(12-10)!} = 1.395 \cdot 10^8$.

The correspondence space spanned by m model entities and b image features can be represented by a $m \times b$ matrix (see Beveridge [17]). Such a correspondence table lists

relation	n	k	$\mathcal{P}(S)$
$m : b$	2	$ M \times B $	$n^k = 2^{ M \times B }$
$1 : 1$	$\begin{cases} m & \forall m \geq b \\ b & \forall m < b \end{cases}$	$\begin{cases} b & \forall m \geq b \\ m & \forall m < b \end{cases}$	$\frac{n!}{(n-k)!}$
$1 : b$	$\begin{cases} b & \forall m \geq b \\ m & \forall m < b \end{cases}$	$\begin{cases} b & \forall m \geq b \\ m & \forall m < b \end{cases}$	n^k
$m : 1$	$\begin{cases} m & \forall m \geq b \\ b & \forall m < b \end{cases}$	$\begin{cases} b & \forall m \geq b \\ m & \forall m < b \end{cases}$	n^k

Table 5.4: Different types of relations that can be applied on a correspondence space S . $m = |M|$ is the size (cardinality) of the set of model elements and $b = |B|$ is the size (cardinality) of the set of the image features.

potential, impossible and current correspondences in an intuitive form as shown in Figure 5.13.

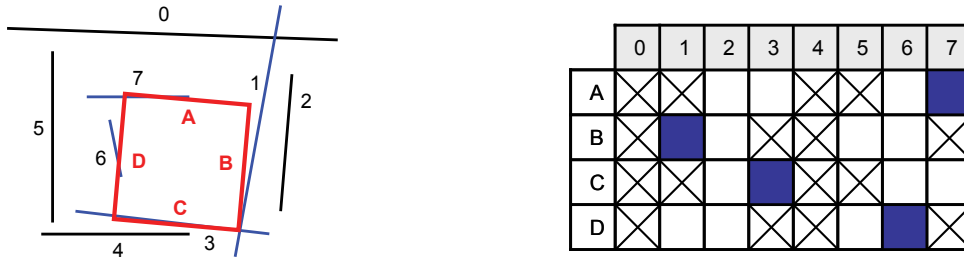


Fig. 5.13: Fit-matrix representation of a correspondence space consisting of four model lines (A–D) and eight image lines (0–7).

The search strategy used here is a modified random start local search algorithm as described in Beveridge [17]. It combines random sampling (see Fischler and Bolles [58]) with iterative (linear) improvement. The goal of this scheme is to avoid a time-intensive full search by randomly jumping to candidate areas in the search space and then to refine the search in order to find the global optimum. The algorithm works as follows (see Figure 5.14 for clarification):

Image Processing Computer vision algorithms extract features from camera images (see Section 5.3).

Model Processing The model of interest is processed and converted to the appropriate format (see Section 5.4).

Correspondence Table Setup A correspondence table containing the possible relations between model and image entities is created.

Random Sampling A subset of typically around five random model features is chosen and all possible correspondences to the image features are gone over. For every single combination the rigid body motion is calculated and the one with the lowest pose error (see Section 5.2.3) is taken as the best current fit.

Linear Refinement The initial pose is then refined by subsequently adding single model features and computing the pose error for all possible image entities. If a model feature worsens the previous pose, it is neglected and another feature is added. This procedure is repeated until the maximum correspondence count is reached. This number depends on the number of model and image features as well as on the relation type used (see Table 5.4). If linear refinement does not meet the error boundaries or consistency checks (see Section 5.7.3), a new random sampling step is initiated.

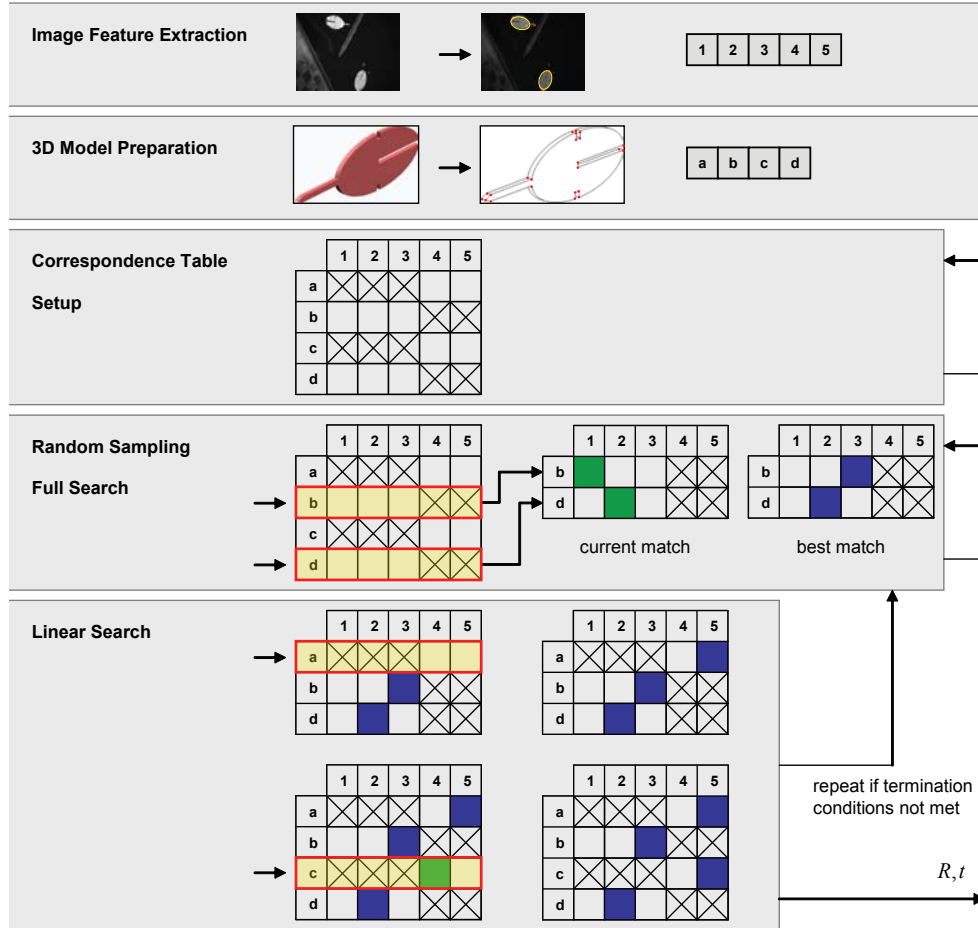


Fig. 5.14: Elements of the pose estimation routine (based on Rosenhahn [128]).

In order to efficiently define a neighborhood of a correspondence set, the Hamming distance n -neighborhood (refer to Beveridge [17] for details) is used, where n stands for the

number of different bits. Thus, the correspondence table is internally represented as a bit string where 0 and 1 stand for no match and a match, respectively. An example of this format is shown in Figure 5.15.

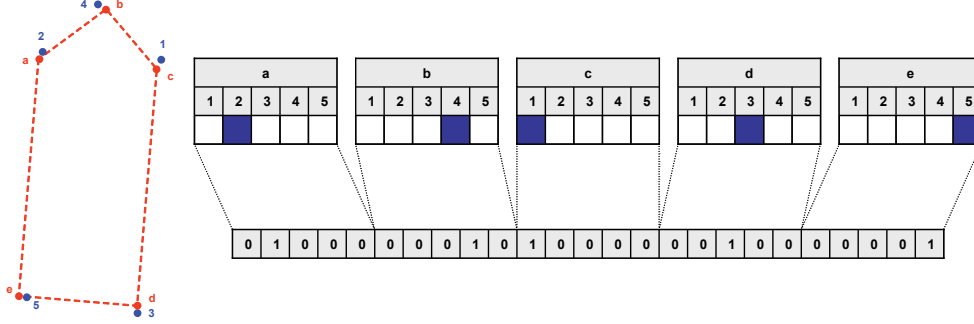


Fig. 5.15: Bit-string representation of a correspondence set consisting of five model points ($a-b$) and five image points ($1-5$).

An example of the Hamming distance 2-neighborhood used in this algorithm is shown in Figure 5.16.

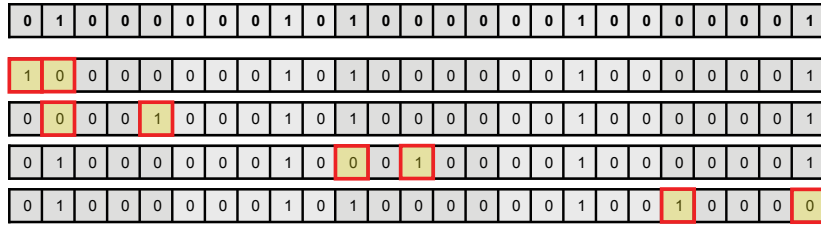


Fig. 5.16: Four examples of Hamming-distance-2 neighbors of a given bit-string.

5.6 Combining Pose Results

The preceding algorithm processes single camera images from a single view in order to find correspondences and extract the rigid body motion. Additional strategies are required when using multiple views, such as in the present configuration of the micro-assembly station V2. Since the correspondence matching is solely based on features from single images, a comparison of the rigid body motion is chosen. Given is an n -camera system with each camera projecting a different viewport of the same scene. For $n = 1$ the pose estimator yields a single homogeneous transformation matrix T_1 that approximates the true pose at the best. For $n \geq 2$ there are a number of solutions for the pose (T_1, \dots, T_n) that have to be combined. The goal is to make sure that all estimated poses approximate the same “true” pose.

Due to the fact that all cameras of this setup are calibrated with respect to the same coordinate frame, the pose estimator provides identical RBM matrices assuming an ideal case. However, wrong correspondence pairs bias the solution and thus (T_1, T_2) are not equal, but similar. The comparison of the rigid body motion can for example be done by comparing the twist parameters θ , ω and m , or, in case the homogeneous transformation matrices have already been calculated, by evaluating $(I - T_1 T_2^{-1})$ which should yield a matrix with values close to zero.

A more intuitive strategy is to extract the Z-Y-X Euler angles from each transformation matrix

$$T = \begin{pmatrix} R_{3 \times 3} & t_{1 \times 3} \\ 0_{3 \times 1} & 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

using

$$\begin{aligned} \beta &= \text{Atan2}(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}) \\ \alpha &= \text{Atan2}\left(\frac{r_{21}}{\cos \beta}, \frac{r_{11}}{\cos \beta}\right) \\ \gamma &= \text{Atan2}\left(\frac{r_{32}}{\cos \beta}, \frac{r_{33}}{\cos \beta}\right) \end{aligned}$$

where $\text{Atan2}(x, y)$ is the two-argument arc tangent function. Angles (α, β, γ) and translation values (t_x, t_y, t_z) can then be compared for each pose given individual threshold values $(\Delta_{rot}, \Delta_{trans})$ for rotation and translation.

Now that the equality of homogeneous transformation matrices can be quantified, a case differentiation is performed. For the case of a single camera ($n = 1$) no comparison is required and the single pose is taken. For $n = 2$ multiple cases exist. If the two poses are equal within the specified tolerance they are assumed to be correct. If they are not equal, additional information needs to be introduced in order to find out if one of the solutions is the true pose. Image quality, partial occlusions by the gripper or other parts, as well as degenerate configurations are some factors that indicate a high probability for false matches. Another option is to simply run an additional pose estimation step for all views until both poses match. For the case of $n \geq 3$ all poses are compared with each other and the outliers discarded. This method can run into a problem if the thresholds areas of different values overlap as visualized in Figure 5.17. The probability can be reduced by reducing the threshold values or by introducing additional information as explained for the case where $n = 2$.

The present mechanical configuration allows a maximum of three cameras (see Section 3.2.6) but only two are mainly used since the third ring unit holds a glue dispenser (see

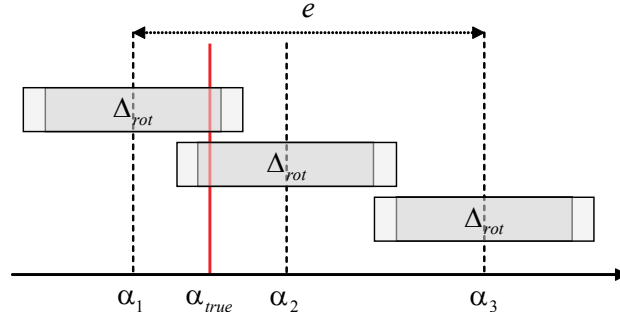


Fig. 5.17: *Threshold overlapping problem: Given are three different Euler angles α_1 , α_2 and α_3 that have been extracted from three transformation matrices. With the threshold value Δ_{rot} the pairs $\{\alpha_1, \alpha_2\}$ and $\{\alpha_2, \alpha_3\}$ are classified as similar, whereas $\{\alpha_1, \alpha_3\}$ is not. Since the position of the true pose α_{true} is unknown, additional information is required for selecting the correct pair. The illustration shows that a smaller Δ_{rot} can solve the problem in this case.*

Section 3.2.7). The present logic in the software compares two poses and rejects both of them if they do not match.

5.7 Implementation and Improvements

5.7.1 Ground Plane Constraints

If it is known that the objects of interest are lying flat on the workbench — an assumption that is usually valid when the system is initialized —, an additional set of constraints can be defined. The platform is then modeled as a horizontal plane and all coplanar points of the model are constrained to it using point-plane constraints (see Section 5.2.2). If this feature is turned on, the constraints are added regardless of the current state of the random sampling or linear refinement process.

Since ground plane constraints add more constraints to the system of equations, they induce a little increase in computation time. However, they dramatically constrain the problem and help reducing degenerate poses.

5.7.2 Random Sampling Full Search Versus Random Sampling

The random part of the correspondence matching process can be handled in two ways. The first method calculates the pose for all combinations of the randomly generated subset and takes only the best one for linear refinement. The second and preferred algorithm tries a linear refinement for every correspondence set that yields a pose error below a given threshold. The second version is preferred since it sends more, at first sight, “unsuited” correspondence sets to the linear refinement and thus generally reaches the optimum in a shorter amount of time. However, the average time consumption is not necessarily lower as shown in Section 6.3.

5.7.3 Consistency

Locating an object in unknown space is sometimes similar to finding the needle in the haystack and is therefore prone to degenerate or completely false results. Some of these can be filtered out by simple validity or consistency checks. The successful detection of outliers makes result combination more robust and reduces overall computation time. A simple method is the center of mass error already explained above in Section 5.2.4. It simply indicates if a reposed model is too far away from the one found in the image. Another simple method allows checking for results that transform a model “behind” the camera.

Time consumption is not relevant for this additional processing step since it cannot be compensated by simply running more iteration steps. The correct judgement of the quality of a pose is very important and more research will be required in this area in order to reach a robust system.

5.7.4 Multi-Threading

As already explained in Section 3.4.3 and visualized in Figure 3.32, each camera module is associated with a tracking thread each consisting of an image processor and geometry analyzer. Last is an abstraction for pose estimation (and/or tracking) routines. This also means that each estimator loop runs in its own thread. Even though all threads have the same level of priority, some of them are likely to finish their calculation loop earlier than others. This even more due to the random characteristic of the pose estimation algorithm. In order to have a common basis for pose comparison and combination (see Section 5.6), the individual entities have to be stopped upon the delivery of a new result. This behaviour is implemented using the barrier concept as shown in Figure 5.18.

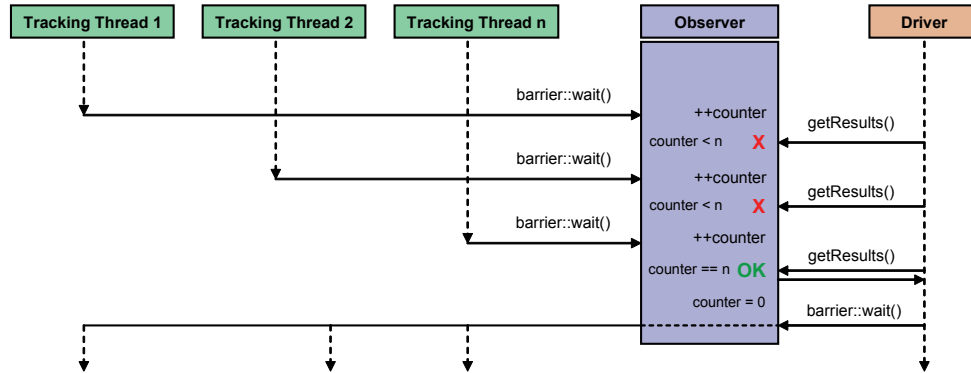


Fig. 5.18: Threading scheme for multiple cameras. Each thread i is stopped upon calling `barrier::wait()` and increases a thread safe counter. The polling main driver thread is only satisfied if the counter reaches the total number of running threads n , combines the results and releases the barrier with the same wait call. All threads are then simultaneously released and processing the next iteration step. The observer and driver shown above run in the same thread.

Each thread is stopped after calculation and only released after all threads are finished and the results combined by the main driver thread. This ensures each estimation loop works with images from the exact same scene and thus allows the comparison of the results.

5.7.5 Tuning Parameters

Table 5.5 shows a list of the relevant parameters that steer the behaviour of the pose estimation routine.

variable	description
initialCorr	initial number of correspondences
epsilon	pose calculation error termination criterion
minRSThreshold	random sampling lower threshold
maxRSThreshold	random sampling upper threshold
minLSThreshold	linear search lower threshold
maxLSThreshold	linear search upper threshold
maxCOMError	maximum center of mass error
maxModFeatPerImgFeat	maximum allowed number of model features per image feature
maxImgFeatPerModFeat	maximum allowed number of image features per model feature
GPC	include ground plane constraints
COMC	include center of mass error check

Table 5.5: Summary of the parameters controlling the pose estimation behaviour.

5.8 Summary and Conclusions

An existing pose estimation routine has been adapted and implemented for the specific needs of micro-assembly. The method consists of two major components, correspondence matching and pose estimation. The first step features a random component which selects a subsample of a large dataset and evaluates the best match. If and only if the sample satisfies a set of boundary conditions it is linearly refined in the second step until the maximum number of correspondences is reached. The calculation of the rigid body motion is based on constraint equations, which combine 2D points with 3D lines or 2D points with 3D planes. Implicit transformation to various mathematical spaces finally yields a true 3D error measure in Euclidean space.

Experimental Results

The focus of this chapter is on experimental verification of the various components of the micro-assembly system. It starts off with individual modules related to software and mechanics, and then shows the performance of the overall system by describing the assembly of two micro-robotic devices.

6.1 Introduction

The following sections show the results of some experiments performed with the micro-assembly station V2. It starts off with the performance of the camera calibration routine that is essential for correct pose determination as well as visualization purposes. Pose estimation results are shown next and are subdivided into off-line and on-line experiments, as well as a section about general performance. The subsequent section is about accuracy and precision of the whole system followed by a small discussion about illumination. The last paragraph shows some assembly results for a bio-micro-robot as well as an ultrasonic transceiver. The chapter finishes off with a closing note on general performance.

6.2 Camera Calibration

The camera calibration algorithm is extensively described in [Appendix B](#) and thus only the results are presented here. First, all mechanical components of the microassembly system are initialized and calibrated. This concretely means in subsequent order: calibration axes, manipulator (base and gripper unit), and ring units. For this case a calibration pattern size of 0.6 mm is chosen (see [Figure 6.1](#)) and a series of 5 images at levels of $d_z = 0.25$ mm apart are recorded (see [Figure 6.2](#)).

The patterns are registered by clicking the three outer points starting with the origin in clockwise direction and the software automatically interpolates the remaining grid points. The calibration process is started once all points of all images of one camera are determined. This process is repeated with all remaining cameras making sure that the first point clicked in every image is the same for all images (common origin).

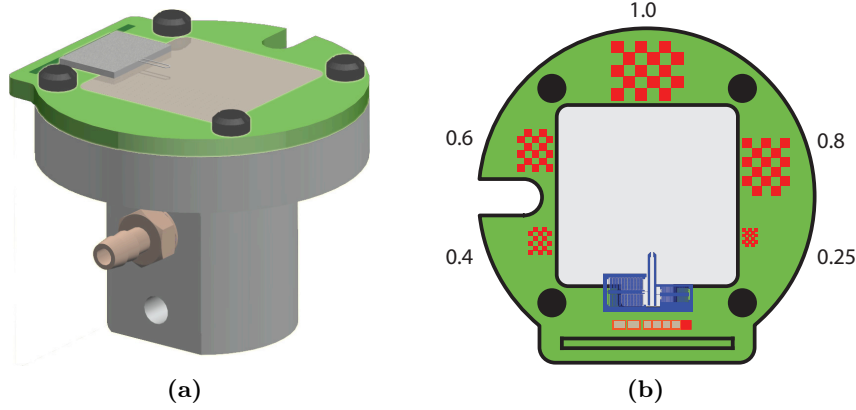


Fig. 6.1: Different calibration patterns (a) of the workbench (b). Numbers indicate the edge length of the individual squares in millimeters.

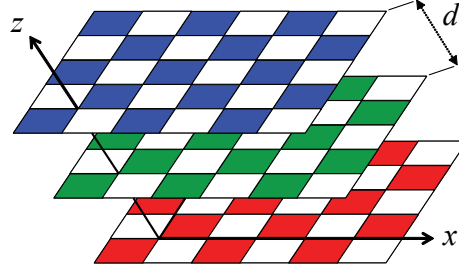


Fig. 6.2: Sample set of three camera calibration patterns at different z -levels. The main application *maPilot* automatically records the calibration images from each camera given a distance z and the total number of images.

Figure 6.3 shows the original refined corner locations together with the reprojected calibration pattern for two selected images from different views. A series of 10 calibration sequences yielded an RMS reprojection error of 2.13 pixels.

The projection of the center of origin of the platform frame of reference \mathcal{P} given real images is shown in Figure 6.4. It can be observed that the crosshair drifts with increasing or decreasing x and y value. The maximum deviation is in the order of $\pm 200 \mu\text{m}$ and is due to the imperfect alignment of the rotation axis θ (see Section 3.2.5) and the problematic decomposition of the projection matrix that requires an assumption for the focal length (see Appendix B.5).

A similar result is obtained by rotating around z by θ and the maximum deviation is also in the order of $\pm 200 \mu\text{m}$ for a full rotation of 360° . Figure 6.5 shows the results for three angular positions.

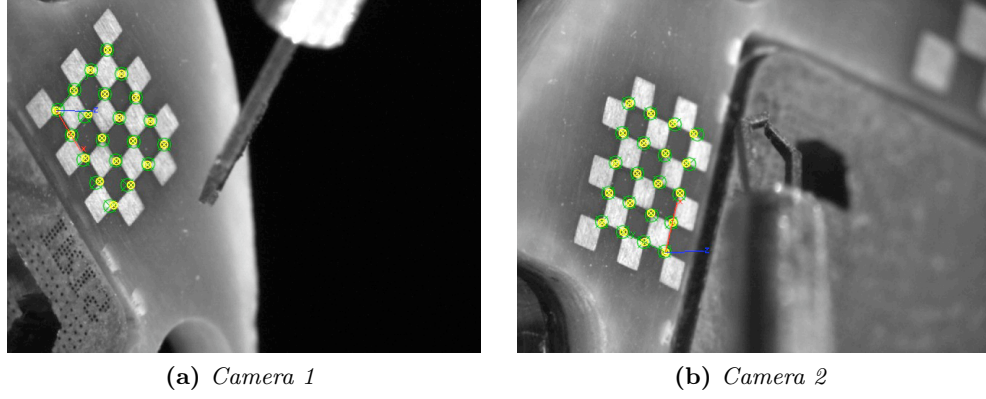


Fig. 6.3: Reprojection of calibration grid on the real image. The larger (green) markers indicate the location of the original pattern interpolated from three points given by user input. The smaller (yellow) markers are the reprojected corner positions using the calculated projection matrix P . Both images also show the projected common coordinate frame WCS (in RGB colors).

6.3 Pose Estimation

The results of the pose estimation routine are partitioned into off-line experiments, which work with single static images, on-line experiments, performed on the real micro-assembly setup, and general performance experiments.

6.3.1 Off-Line Experiments

A first series of experiments have been performed with an L-shaped object of dimension 115x74x21 mm that has been observed with a BASLER A602fc-2 color camera and a regular 12.5 mm lens. An array of 96 white high power LEDs, pulsed and synchronized with the camera trigger signal, provided balanced illumination. Table 6.1 shows the averaged results for a set of 100 experiments using point-plane constraints, as well as the best match from that set. The initial and final number of correspondences were 4 and 7, respectively.

value	best match	average for set
relative pose error	0.55 mm	1.2 mm
center of mass error	15.81 pixels	21.5 pixels
computation time	21.5 s	78.9 s

Table 6.1: Pose estimation result for L-shaped object.

Figure 6.6 shows two excerpts from image feature detection and pose determination of the L-shaped device.

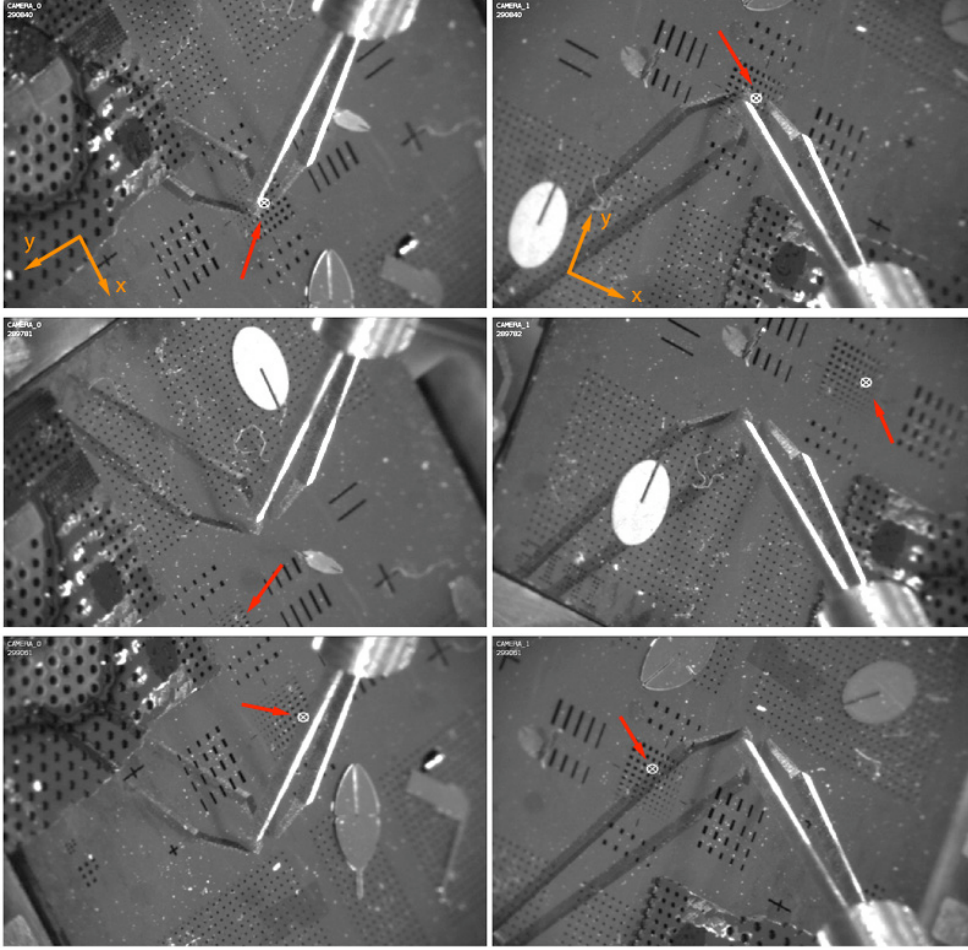


Fig. 6.4: Projection of the center of origin (red arrow) on the real scene. The two columns show live images from camera 1 and 2, respectively. The first row is for manipulator $(x, y, z, \theta) = (0, 0, 2, 0)$, the second for $(2, 2, 2, 0)$ and the third for $(-2, -2, 2, 0)$. All units are in mm or $^\circ$.

Similar experiments have been performed with a model house of dimensions 160x200x105mm and the same vision hardware. Again, results for a series of 100 experiments using point-plane constraints are shown in Table 6.2 and Figures 6.7 and 6.8 visualize a sample result set.

value	best match	average for set
relative pose error	0.62 mm	1.42 mm
center of mass error	21.7 pixels	24.1 pixels
computation time	5.7 s	55.2 s

Table 6.2: Pose estimation result for model house.

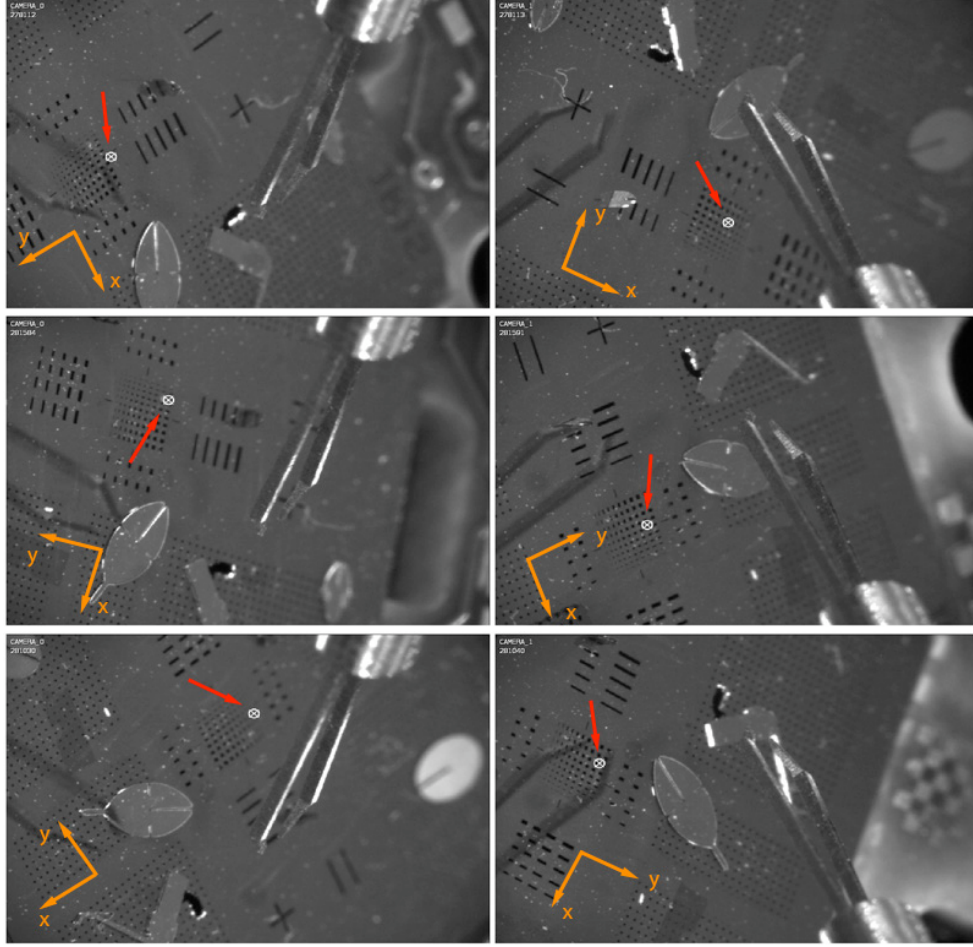


Fig. 6.5: Projection of the center of origin (red arrow) on the real scene. The two columns show live images from camera 1 and 2, respectively. The first row is for manipulator $(x, y, z, \theta) = (2, -2, 0, 0)$, the second for $(2, -2, 0, -45)$ and the third for $(2, -2, 0, -90)$. All units are in mm or $^\circ$.

6.3.2 Performance

As already mentioned in Section 5.5.2 the performance of the pose estimation routine is mainly influenced by the size of the search space and the strategy it is combed through with. Series of 1'000 trials haven been performed where each trial consists of a full localization step including model preprocessing, image processing, and pose determination. The number of initial correspondences have been varied from 2 – 4 for point-plane (PP) constraints and 3 – 4 for point-line (PL) constraints. Using only 2 initial correspondences for point-line constraints does not provide enough equations and thus leads to degenerate solutions. The model data used for the experiments is a square micro-robot that has already been introduced in Table 5.3. Only 8 features at level $z = 0$ are used in this case

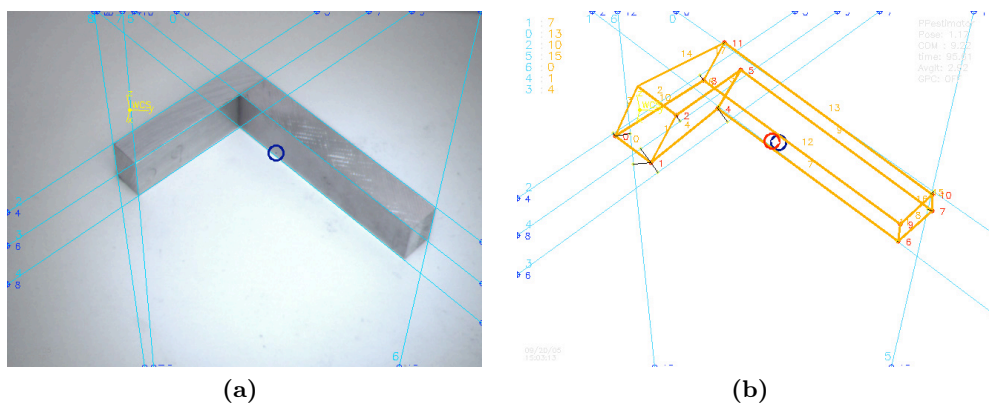


Fig. 6.6: Image processing (a) and pose estimation result (b) for L-shaped object.

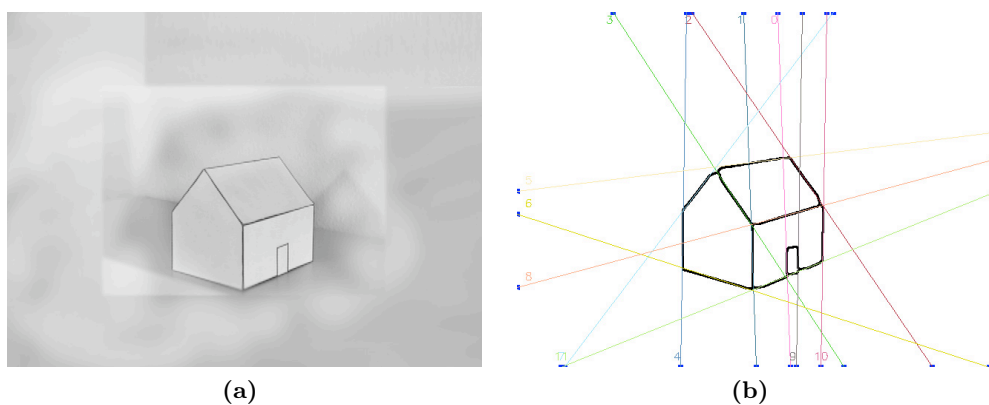


Fig. 6.7: Raw image (a) and image processing result (b) for a model house.

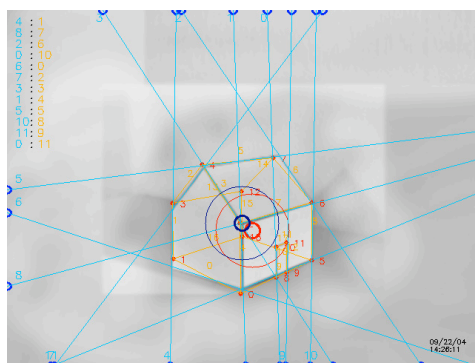


Fig. 6.8: Pose estimation result for a model house.

in order to reduce redundant information. In addition, the algorithm is configured to use ground plane constraints and allows only 1:1 mappings of constraints (see Table 5.4).

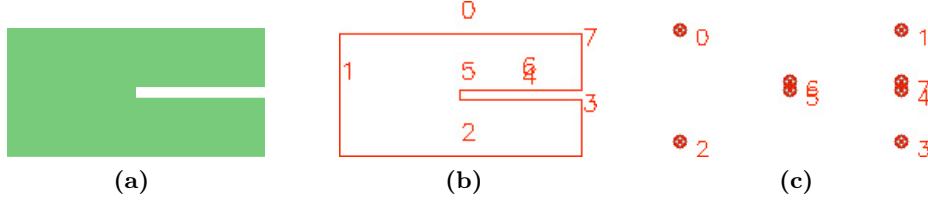


Fig. 6.9: Square micro-robot test case for computation time experiments. The images show the CAD model (a), the vertices (b) as well as the corner points (c), both for $z = 0$.

The individual incidents of the computation time experiments are shown in Figures 6.10 and 6.11. These plots show the computing time for successful trials, for different initial number of correspondences, for point-line and point-plane constraint types, and for the regular and full search random sampling scheme. It can be generally said that the cases using point-plane constraints are significantly faster than the ones using point-line constraints. This is mostly due to the fact that the system of equations for point-line constraints consists of three times as much equations as the one for point-plane constraints for the same number of correspondences (see Equations 5.2.1, 5.2.2, 5.2.3, and 5.2.4).

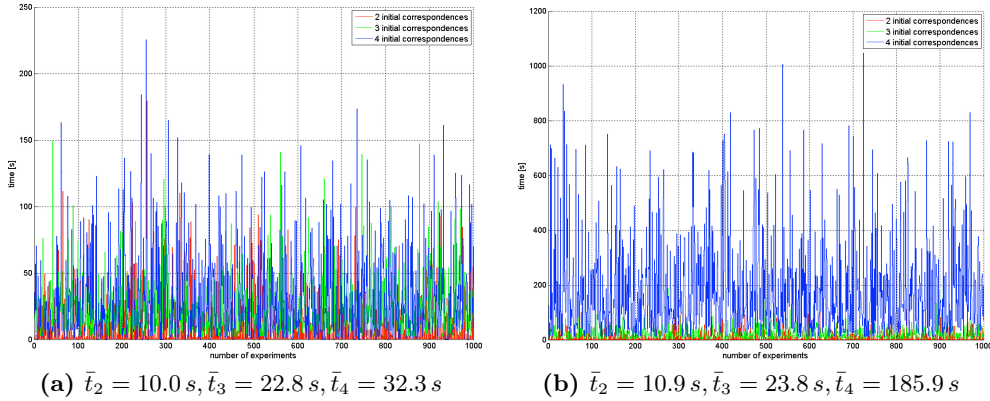


Fig. 6.10: Computation time for random sampling (a) and random sampling full search (b) for point-plane constraints and different numbers of initial correspondences.

Figure 6.12 combines data from Figures 6.10 and 6.11 into a single plot. Three of the graphs follow a trend that agrees well with the characteristics of the search space dimensions indicated by the violet line with circles. An interesting fact is that the point-plane type seems to show a linear behavior for the regular search with increasing number of correspondences. This is in agreement with the random behavior that does not exponentially change as opposed to the full search version.

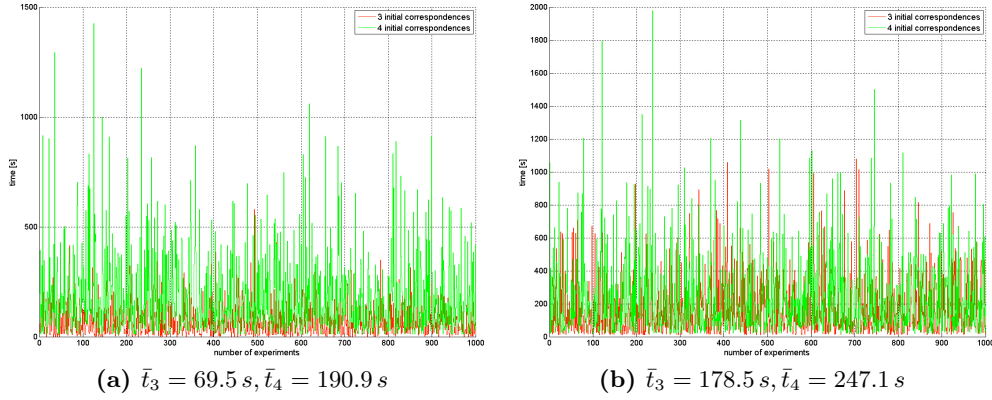


Fig. 6.11: Computation time for random sampling (a) and random sampling full search (b) for point-line constraints and different numbers of initial correspondences.

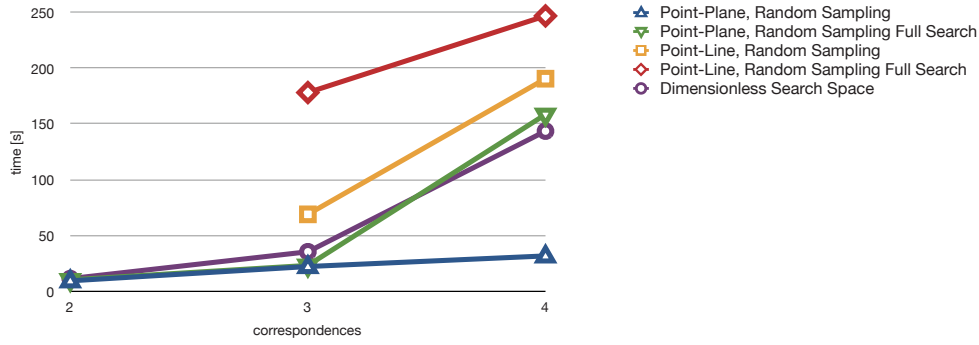


Fig. 6.12: Comparison of computation time for different initial correspondences. This plot basically combines data from Figures 6.10 and 6.11. In addition, the size of the search space has been added (violet line with circles). Note that there is no data available for two initial correspondences for both point-line experiments due to the fact that insufficient number of equations are generated leading to degenerate solutions.

6.4 Accuracy and Precision

Precision and accuracy of the SUTTER MP-285 *xyz*-stage is measured using an optical approach. It consists of a regular microscope that is mounted on a vibration table and faces down to a USAF resolution target with a minimum feature width of 228 cycles/mm mounted on the MP-285 (see Figure 6.13). Static calibration yields a one pixel resolution of $4.92 \cdot 10^{-5}$ mm at a precision of $\pm 0.75\text{ px} = \pm 0.037\text{ }\mu\text{m}$.

The measurement consists of driving to 16 individual locations and optically measuring the target position at each of them. This procedure is repeated three times in order to account for measurement errors. The optical accuracy is high enough to measure the length of a single regular step of the motor. However, since the accuracy is in the same order of magnitude as a single microstep, the resulting length from a microstep cannot

	value	best	average	unit
Point-Plane (RS)	relative pose error	0.071	0.196	mm
	center of mass error	0.013	0.016	pixel
	computation time	0.217	32.31	s
Point-Plane (RSf)	relative pose error	0.071	0.077	mm
	center of mass error	0.013	0.015	pixel
	computation time	11.86	185.94	s
Point-Line (RS)	relative pose error	0.071	0.077	mm
	center of mass error	0.013	0.015	pixel
	computation time	11.98	190.91	s
Point-Line (RSf)	relative pose error	0.058	0.058	mm
	center of mass error	0.0	0.0	pixel
	computation time	19.72	247.12	s

Table 6.3: Pose estimation result for rectangular micro-robot part. The number of initial correspondences is 4 for all cases.

be accurately determined. Hence they have been calculated from the measured length of a single regular step. A summary of the measurement results is shown in Table 6.4

	measurement		calculation
	\bar{x}	σ	
1 pulse	$0.195 \mu\text{m} \pm 0.025 \mu\text{m}$		$0.2 \mu\text{m}$
1 step	$1.950 \mu\text{m} \pm 0.250 \mu\text{m}$		$2.0 \mu\text{m}$
1 °	$2.150 \mu\text{m} \pm 0.280 \mu\text{m}$		$2.2 \mu\text{m}$

Table 6.4: SUTTER MP-285 accuracy and precision measurement and calculation results.

The play of the MP-285 is evaluated by approaching a position from the two opposing sides and yields $0.5 \mu\text{m} \pm 0.2 \mu\text{m}$ for the current configuration. Thus, the hardware play is two times bigger than a single microstep.

Precision and accuracy results for the DC-motor drives are listed in Table 6.5. The first column lists the maximum theoretical resolution of the whole drivetrain, including motor, encoder, and gear reductions. This value can only be reached with the complete absence of any mechanical backlash in combination with a perfect motor controller. Columns two and three show values closer to reality which are extracted from data sheets of the gearbox combinations. In the present case gearboxes generally reduce accuracy by a factor of 10 – 100 and thus become the limiting element in the system.

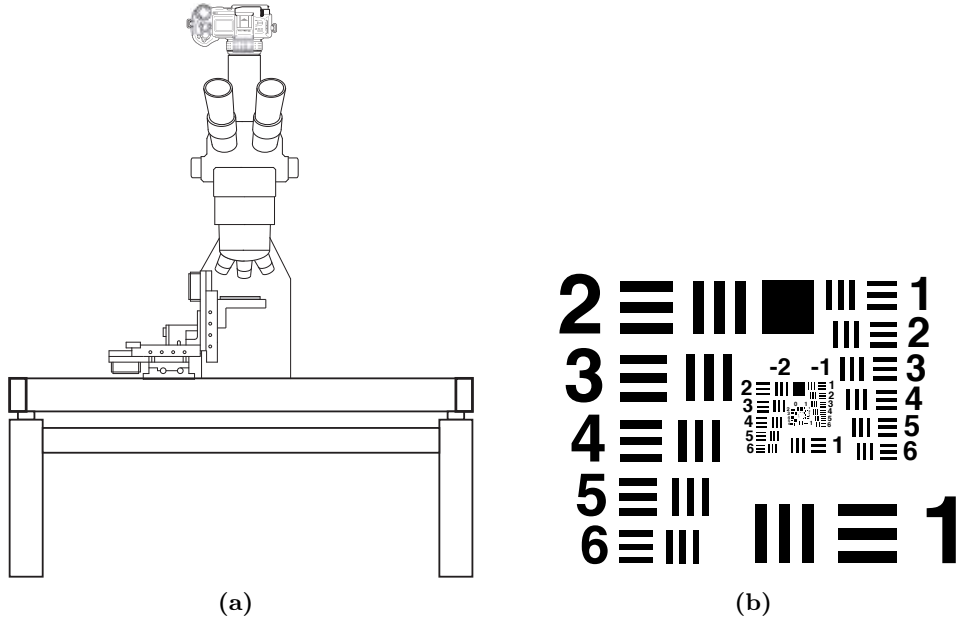


Fig. 6.13: Precision measurement setup for the SUTTER MP-285 (a) using the USAF resolution target (b).

		theoretical		practical	
				\bar{x}	σ
Axis 1	η	0.00016	°	0.016	± 0.0016 °
Axis 2	ζ	0.0018	°	0.075	± 0.025 °
NEWPORT	θ	0.001	°	0.01	± 0.001 °
Calib Y	c_y	0.829	µm	165.806	µm –
Camera 1	f_{z1}	0.014	µm	0.1	µm ± 0.01 µm
Camera 2	f_{z2}	0.014	µm	0.1	µm ± 0.01 µm
Camera 3	f_{z3}	0.014	µm	0.1	µm ± 0.01 µm

Table 6.5: Accuracy and precision of the DC motor drives. The table shows the theoretical values of the whole drivetrain (motor, encoder, gear reductions) as well as the practical numbers dominated by values from data sheets of the gearboxes.

6.5 Illumination

The novel advanced illumination system is presented in detail in Section 3.2.6. It yields crystal clear images of the workbench area and offers the possibility to highlight certain areas. Figure 6.14 shows four possible combinations of the individual light types together with the corresponding histogram. While the spotlight highlights the center area from the top, the low angle illumination visualizes edges and even surface roughness. A combination of those two (see Figure 6.14c) is a good choice for most applications. Diffuse or ambient illumination can be used for shadowless brightening of the scene.

Depending on the aperture setting of the microscope lenses the output power of the illumination is adjusted. With a 0.75x primary magnification and an aperture setting of roughly 20 %, a configuration used for all experiments shown here, an output power of less than 75 % was sufficient for all cases. The 25 % reserve will be required once the apertures are fully closed or if parts are coated with a dark material. Since the driver electronics are designed to drive even higher power outputs, the present LED set can be easily exchanged with newer and brighter models.

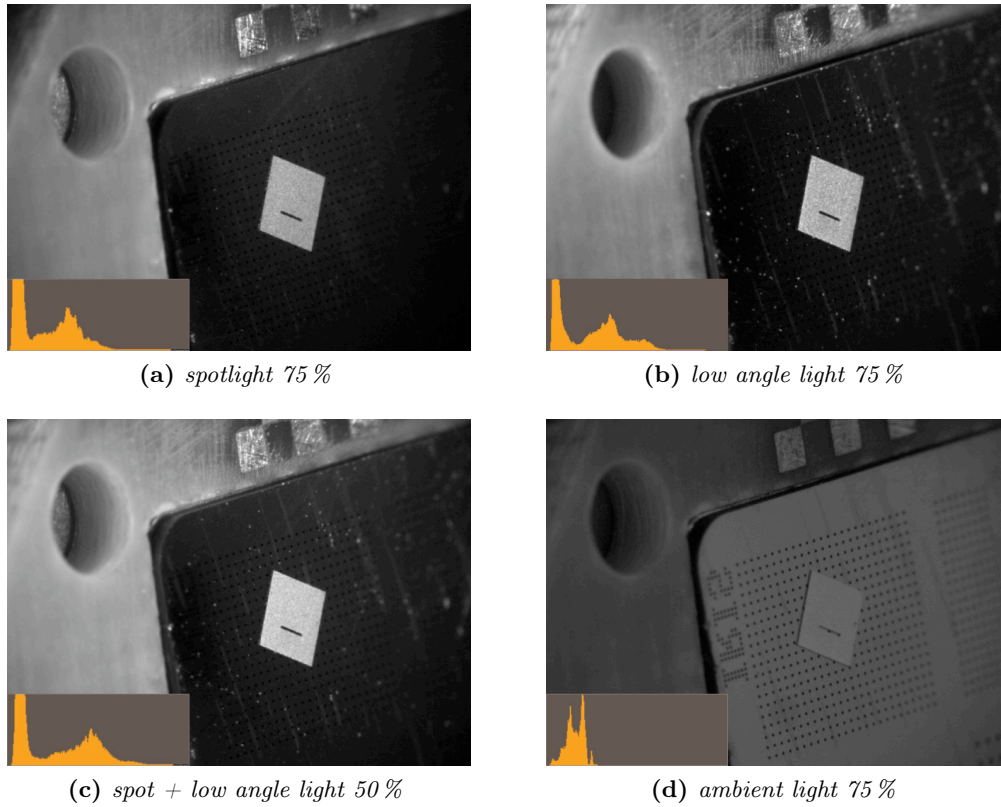


Fig. 6.14: Comparison of different illumination types. Ambient illumination (d) gives a narrow histogram representing the diffuse character of the light resulting in a shadowless scene. On the other hand, spot (a) and low angle illumination (b), as well as a combination of them (c), yield a histogram with two peaks that are beneficial for automatic image thresholding.

Figure 6.15 shows the timing diagram of the BASLER camera's internal logic. An external trigger signal $ExTrig$ induces a delayed exposure whereas the duty cycle is controlled over the time $ExTrig$ is logic high (Th_{ExTrig}). The camera signals an exposure cycle with an integrate enabled output signal ($IntEn$) that can be used to trigger flashes or other illumination devices. In the present configuration the lighting is in complete sync with the $ExTrig$ signal and this yields excellent results as already shown above. However, more sophisticated illumination patterns might be useful for future tasks. Since

the light controller electronics are based on a reprogrammable FPGA (see Section 3.3.2) optimizations can be quickly realized.

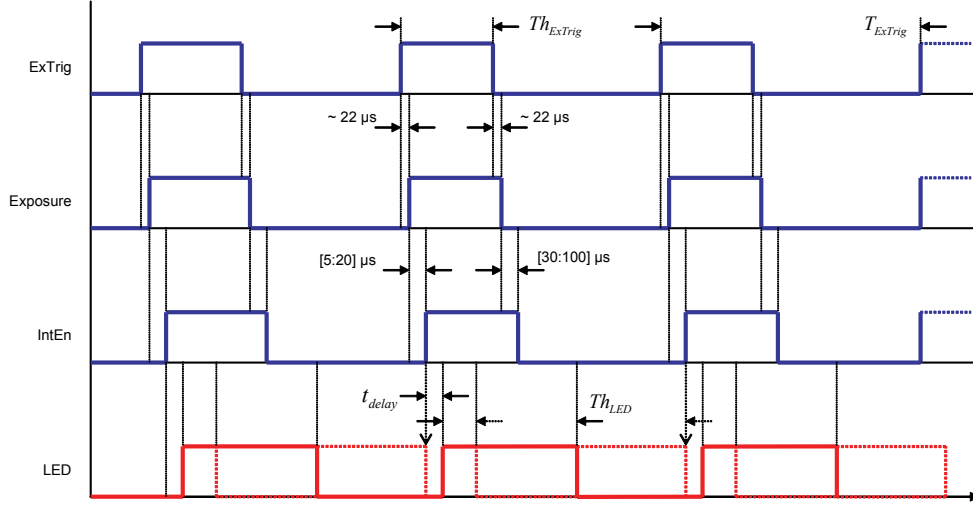


Fig. 6.15: Camera and light triggering scheme for BASLER A602fc-2 color cameras. An external trigger signal $ExTrig$ initiates frame exposure which is indicated by the camera through an integrate enabled signal $IntEn$. The illumination pattern LED is shown on the very bottom.

6.6 Assembly Experiments

This section shows some results of micro-part handling as well as assembly operations. All experiments are performed with the hardware that is extensively described in Chapter 3. The present configuration uses two microscopes on positions RU_1 and RU_2 as well as a glue dispenser on position RU_3 (see Figure 3.19). The microscopes are set to a primary magnification of 0.75x at an aperture level of $\sim 20\%$.

6.6.1 Assembly of Bio-Micro-Robots

The assembly of bio-micro-robotic devices starts with a preparation phase where the individual components are laid on the the workbench, which has been driven to the loading position beforehand. The location of placement is not critical since the manipulator is capable of reaching any location on the platform. However, putting the parts on the four large outer perforated areas is favorable in order to properly suck them down and not to obstruct any other patterns needed for assembly. The base unit is then driven to the assembly position which is determined by prior calibration. The camera view after the preparation phase is shown in Figure 6.16.

The microassembly station is now configured and ready for assembly. The next challenge is the grasping of parts. The following experiments have all been carried out under the

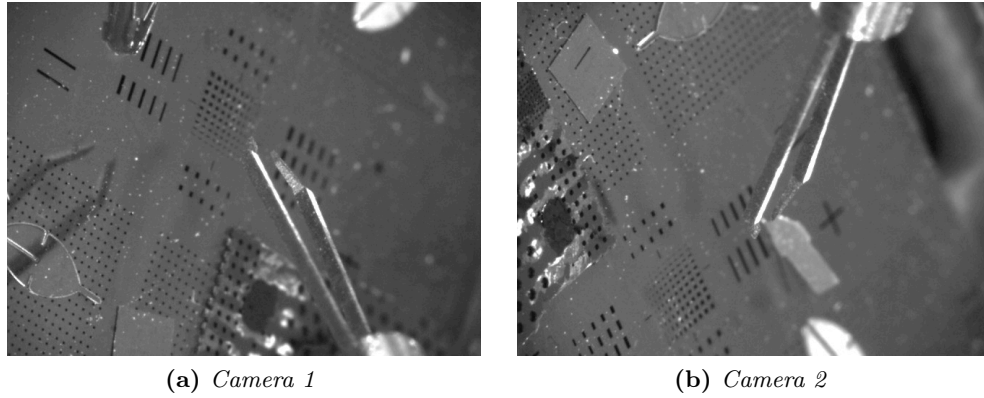


Fig. 6.16: *Camera views of the initial assembly position.*

assumption that all objects are lying on a flat surface with the shortest dimension pointing along positive z . This results in very limited space for gripping since the part thickness is in the order of $50\mu\text{m}$. Figures 6.17a and 6.17b shows the picking up process of an electroplated test head for harddisk inspection IC cards. The limited opening of the gripper requires grasping of the small end first and then a subsequent re-grasping by supporting the body on the platform. The gripper fingers are now in full contact with the object and can therefore exert higher forces and torques. This is particularly important if close tolerances are chosen.

Figures 6.17c and 6.17d show the insertion process of this single object into a slit featuring the same dimensions as its cross section. Even though a slight penetration can be observed, the tolerances are too tight for a complete insertion. The part is finally inserted with its smaller end. Pure assembly time for this simple manipulation is within reasonable 5 – 10 minutes. This is a significant improvement over the “old” system of at least 50 %.

The assembly of bio-micro-robots, such as the model shown in Figure 6.18 basically is a two step process. The first component is picked and placed in an upright position with its slit facing upwards along positive z . Then, the second component is grabbed and inserted as illustrated in Figure 6.18b. In reality, this process is more complex and subject to stochastic behavior. The first step is illustrated in Figures 6.19a – 6.19f where a $50\mu\text{m}$ thick part is inserted in a $60\mu\text{m}$ slit on the platform. Even though the gripper opening is not big enough to match the width of the part, it can be gripped by squeezing the sharp fingers into the $50\mu\text{m}$ slit. This procedure has proven to be very reliable even though the maximum allowed force and torque is limited. Despite the vacuum, adhesion forces tend to stick the object to the gripper with the result that it is extremely difficult to have it remain in a fully upright position (see Figure 6.19f). Even though the workbench features a limited pattern thickness of $50\mu\text{m}$ no relevant sideways tilting is observed over a series of experiments.

The second phase of the assembly process is visualized in Figures 6.20a – 6.20f. Again, the maximum opening of the gripper is not sufficient to get a hold of the small edge length, so the pipette is (mis)used. After a first rough placement there is a chance for

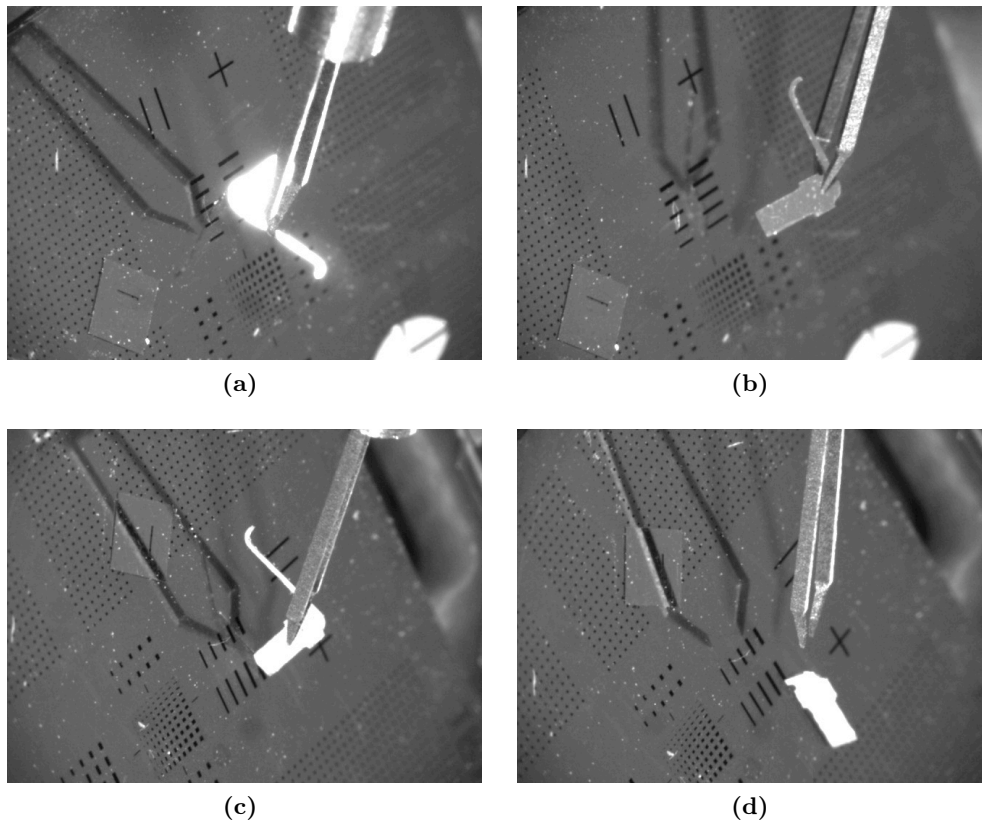


Fig. 6.17: Picking up and inserting an electroplated part in a slit of the workbench. The object is first gripped at its small end (a) and regripped (b) in order to be able to exert higher insertion forces. Trying to insert the larger end fails after multiple trials due to the tight tolerances (c), so the part is finally inserted with its smaller end (d).

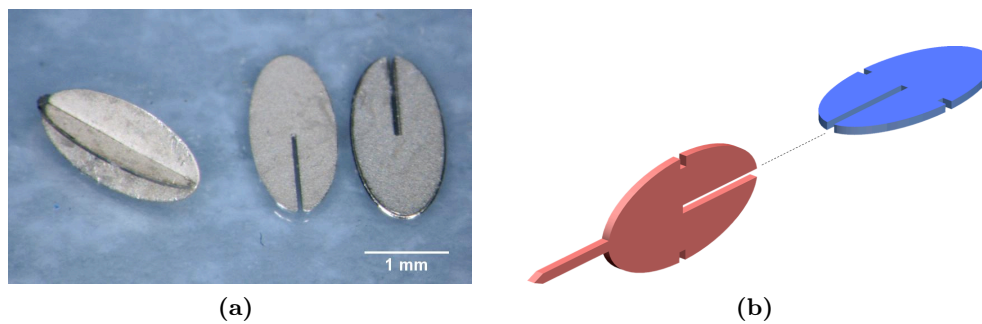
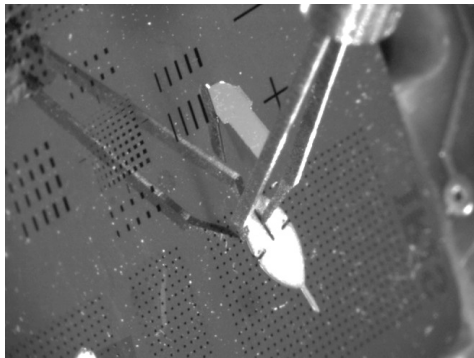
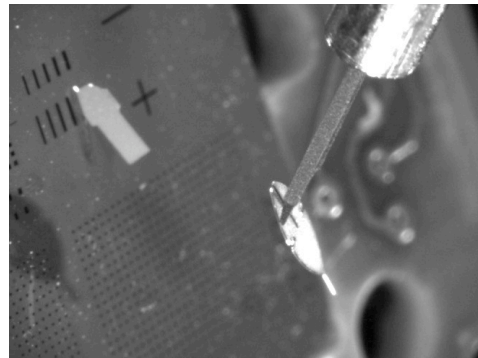


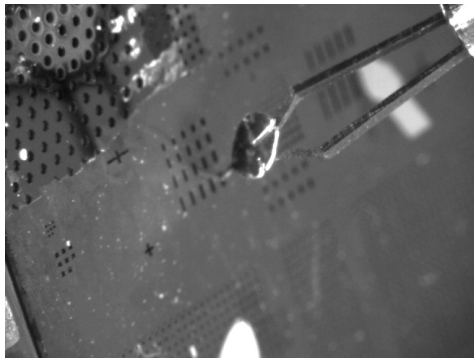
Fig. 6.18: Two-component bio-micro-robot. The simple version on the left (a) does not feature any pipettes as opposed to the model on the right (b).



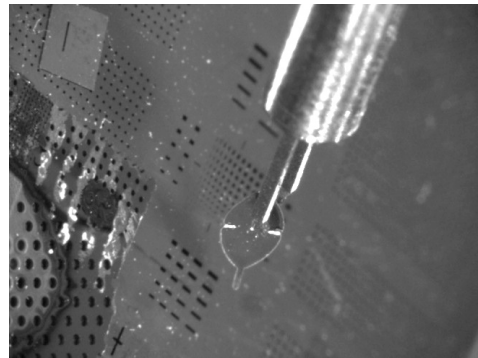
(a)



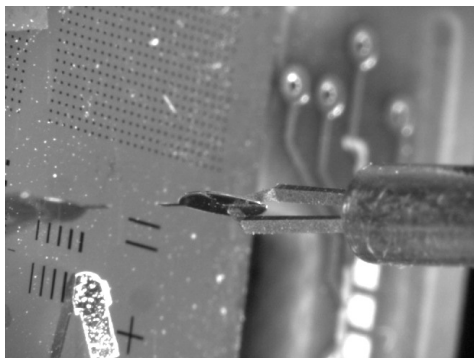
(b)



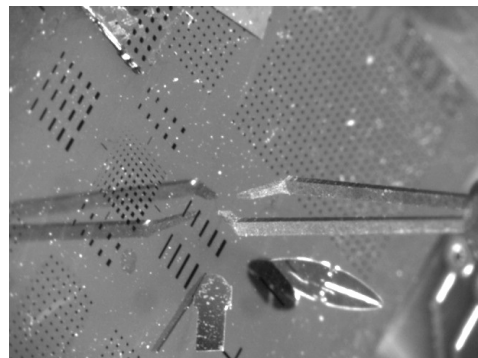
(c)



(d)



(e)



(f)

Fig. 6.19: *First part of the assembly of a bio-micro-robot.*

proper regripping and insertion. A slit width of $50\text{ }\mu\text{m}$ for both elements abruptly stop the sliding in process after half the body length (see Figure 6.20c). Since friction is not enough to create the necessary normal force, the gripper clasps the upper part and mechanically pushes it down until it reaches the platform. However, even the repositioning in a single hole (Figure 6.20e) does not bring the entities closer together. This is again due to the tight dimensions which also results in axial tilting as shown in Figure 6.20f (see white arrow indicators).

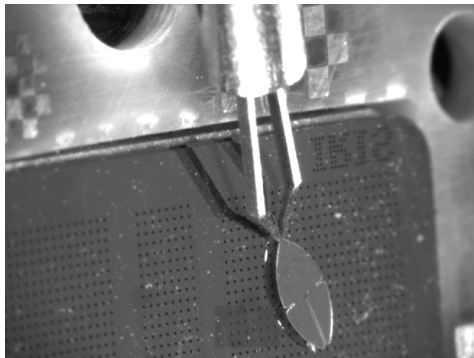
6.6.2 Assembly of Ultrasound Transceivers

An ultrasonic transceiver is a mechanical device that is externally excited by some type of force and reacts with the emission of sound waves in the ultrasonic band. The activation energy can for example consist of a magnetic field generated by a electric coil (see Figure 6.22). These devices gain more and more importance for bio-micro-robotic applications since they can be used for data transmission or localization purposes. Within the scope of a research project at Institute of Robotics and Intelligent Systems a custom ultrasound transceiver prototype module has been designed and manufactured. It basically consists of three parts as shown in Figure 6.21. A $20\text{ }\mu\text{m}$ thick bottom layer is made out of gold with a $1\times 1\text{ mm}$ nickel square in the center. On top there is a $110\text{ }\mu\text{m}$ thick rectangular plastic module interface. The last component is a $20\text{ }\mu\text{m}$ thick complex gold structure consisting of two beam springs holding a $1\times 1\text{ mm}$ nickel square in the center. When assembled, the two nickel bodies have a clearance of $100 - 200\text{ }\mu\text{m}$. Both gold and nickel structures are manufactured by electrodeposition.

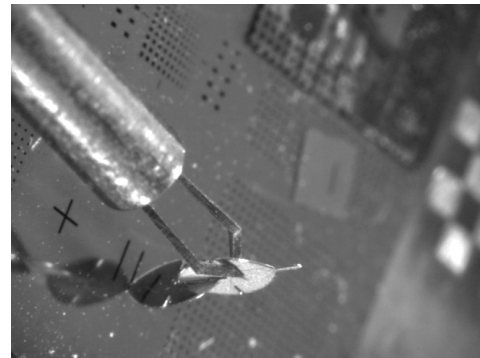
An external oscillating magnetic field ($f > 20\text{ kHz}$) excites the upper nickel body — an effect which is further amplified by the presence of the lower nickel body. This mechanical movement generates sound waves in the ultrasonic band that are then recorded by microphones for further analysis.

Precise placement of the three individual components is crucial for a successful operation of the device. Even slight misalignment of the nickel bodies immediately results in greater magnetic losses and thus impairs overall performance. Despite the relatively large size of the elements, manual assembly has proven to be inefficient, tiring for the operator and not precise enough. The micro-assembly station V2 is predestined for this type of operation as shown next.

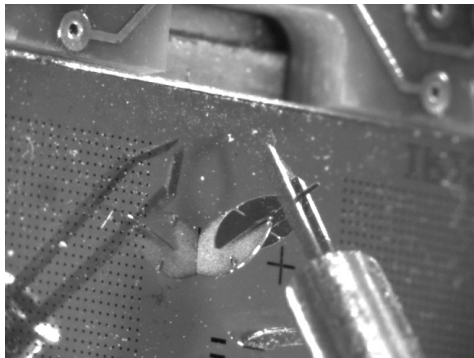
All three components shown in Figure 6.21 are placed on the workbench. Again, position and orientation is not relevant since the manipulator is able to reach any location. For reasons of simplicity, they are placed equally oriented next to each other in this experiment. First, some UV glue is applied to two opposite corners of the bottom layer as shown in Figure 6.23a. Next, the plastic spacer is gripped (Figure 6.23b) and aligned with the bottom layer (Figure 6.23c). Curing of the UV glue by targeted UV illumination fixates the first building blocks. Next, the top layer is gripped at one of its swing arms Figure 6.23d and the glue dispensing needle deployed. It is followed by the deposition of two small droplets of glue again on opposite corners Figure 6.23e and the exact alignment of the top layer (Figure 6.23f). Again, the glue is cured with intense UV light.



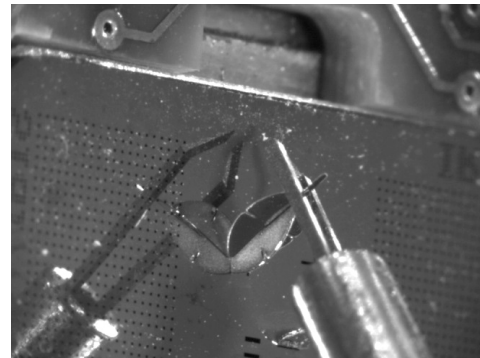
(a)



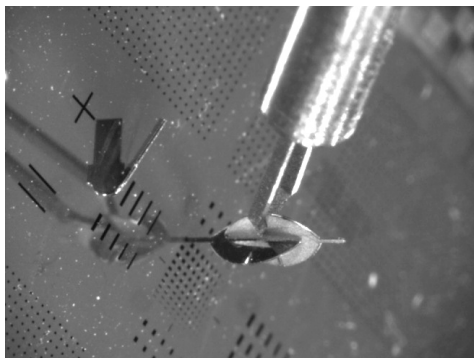
(b)



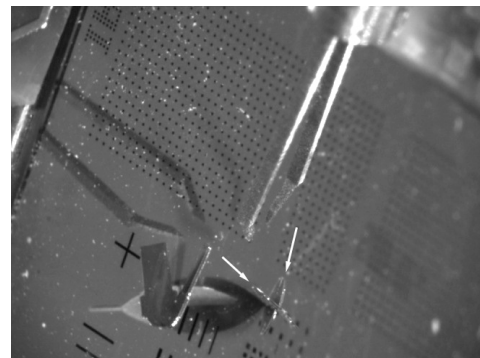
(c)



(d)



(e)



(f)

Fig. 6.20: *Second part of the assembly of a bio-micro-robot.*

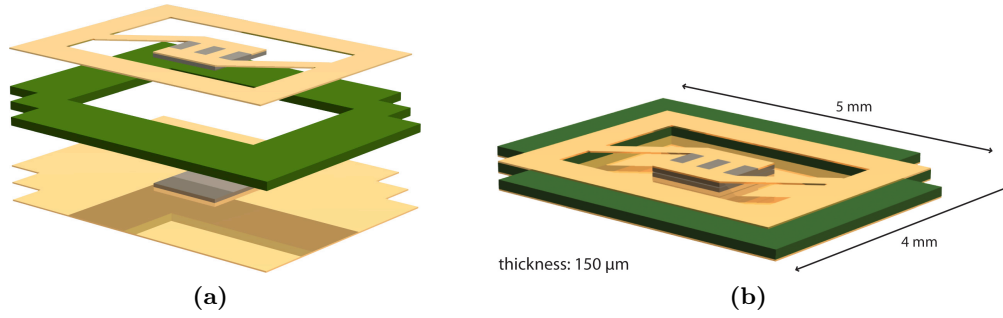


Fig. 6.21: *Ultrasound transceiver developed at IRIS. It consists of a bottom gold layer with a centered nickel insert, a plastic spacer (green), as well as a golden top layer with a suspended nickel body in the center.*

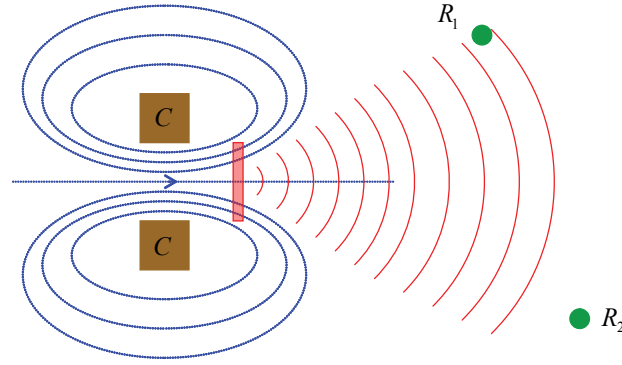


Fig. 6.22: *Operating principle of an ultrasound transceiver. An oscillating magnetic field generated by a coil C mechanically activates a swing arm on the transceiver, which itself emits ultrasound waves that are picked up by receivers R_1 and R_2 .*

A few dozen experiments have shown good performance and a high yield rate of more than 80 %. As opposed to the previous micro-robot case (see Section 6.6.1), picking up and inserting is not an issue, at all. This is mainly due to the fact that fabrication tolerances have little or no influence on the assembly process as a whole. The larger part dimensions are believed to have a little positive effect, too. Again the glue dispenser unit did not work as well as hoped. It often happens that the foremost glue at the end of the dispensing needle hardens and thus cuts off fluid flow. In most of these cases the replacement of the needle is indispensable.

A first quality check of the assembled ultrasound transducers can be readily performed on the micro-assembly station V2 once the glue has completely hardened out. For this purpose the gripper is used to push down the swinging mass as shown in Figure 6.24. The visual analysis of the upper and lower swing yields a maximum range of movement of $\sim 200 \mu\text{m}$.

A more realistic function check can be performed by using a magnetic field and observing the movement of the swinging mass. For that reason, the gripper of the manipulator is replaced by a custom aluminium holder which allows mounting the transducer assembly

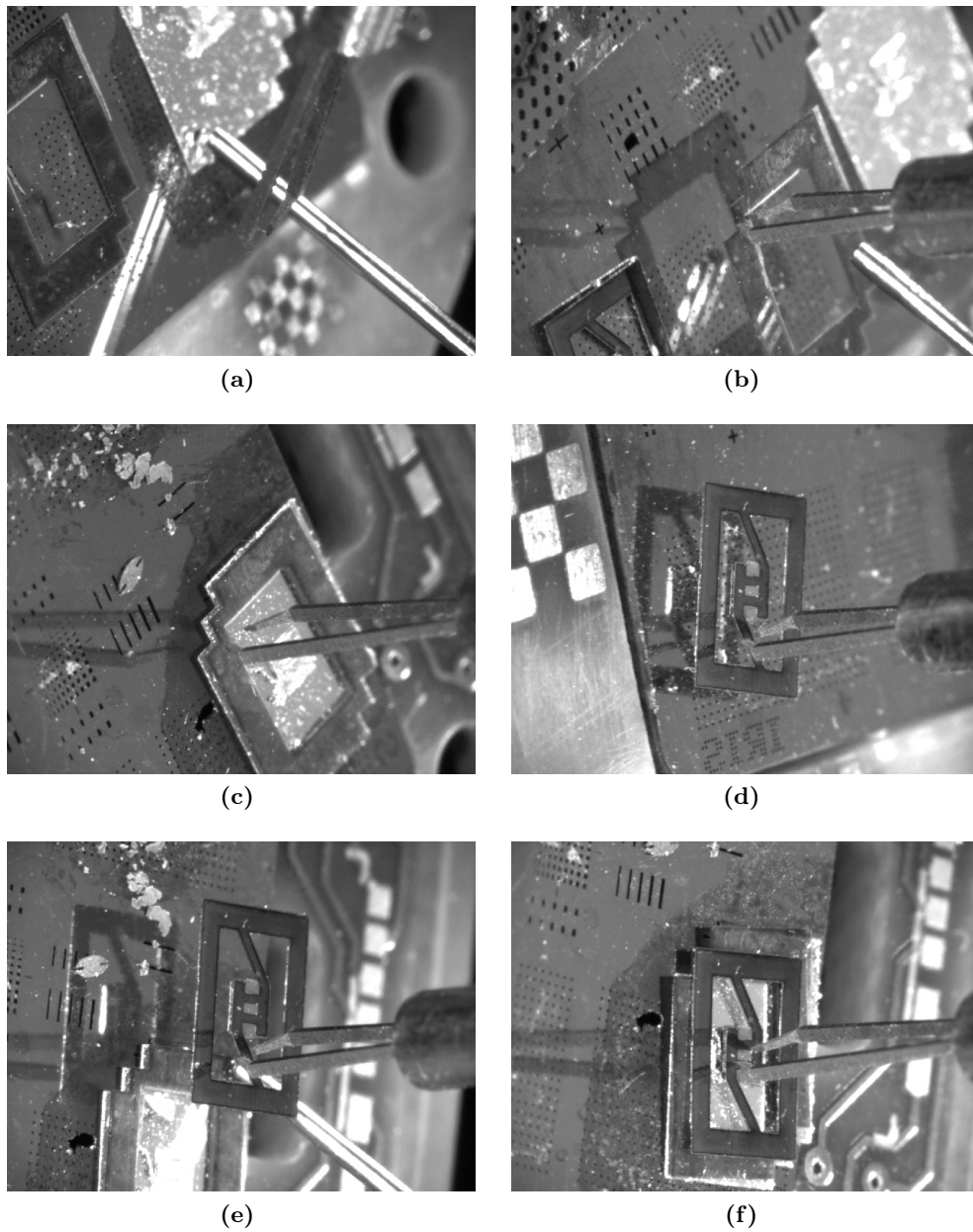


Fig. 6.23: *Assembly of an ultrasonic transceiver. Two droplets of UV activated glue are deposited on opposing sides of the bottom gold layer (a). The plastic spacer is gripped (b) and precisely aligned with the base (c). The same procedure is reproduced again for the top swing element (d,e). Finally the structure is compressed by pushing the gripper on the upper central nickel body (f) so that the glue can be hardened using UV illumination.*

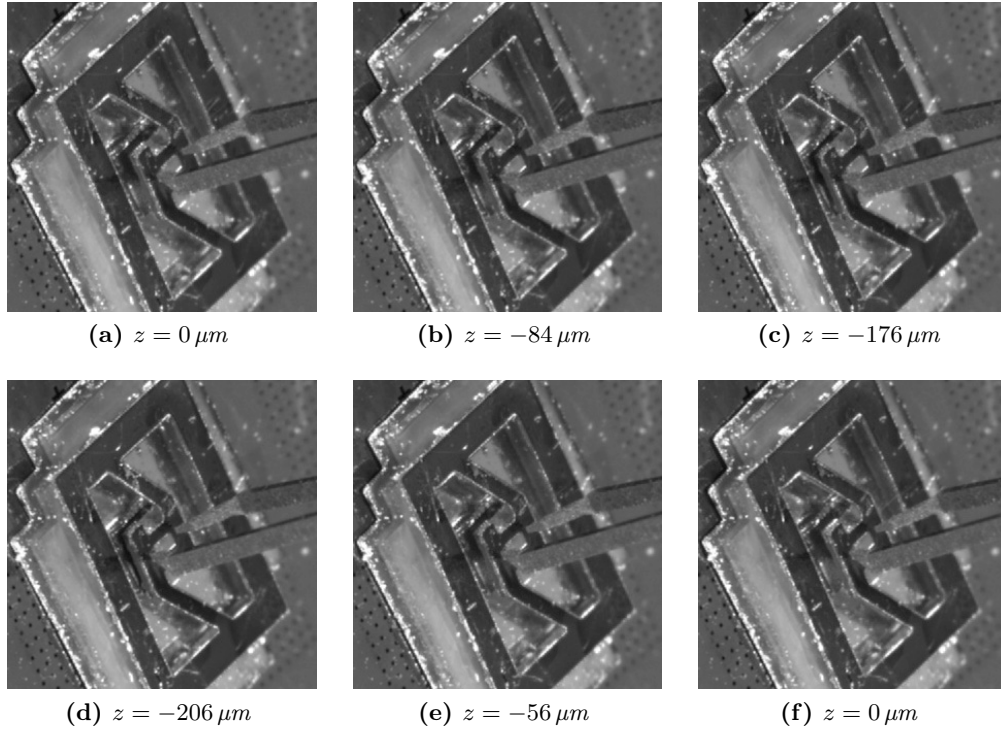


Fig. 6.24: Mechanical test of an ultrasonic transceiver by pushing the center region against the open gripper fingers. The values indicate the relative vertical displacements z_i with respect to $z = 0 \mu m$.

in the remote center of motion RCM. On the other hand, the workbench is replaced by a permanent magnet that can be driven to any location below the transducer. Due to the high stiffness of the swing arm and upper nickel body assembly, the magnetic field was insufficient to create any motion. Only the elimination of one of the arms improved the situation and the flapping movement is shown in Figure 6.25. Since the topic of this thesis is not related to ultrasound, at all, this result is not discussed any further. The only relevance to this project is to show that the micro-assembly station V2 can also be used for measurement and analysis, and to characterize devices.

6.7 Intercommunication Performance

As already described in Section 3.4 the three individual computing entities are connected over a regular Gigabit Ethernet switch that is also hooked up to the local area network (LAN). The average and peak traffic per second over the course of an assembly procedure (~ 30 min) is shown in Table 6.6. The hardware controller ERASMUS shows low rates which is in agreement with the small command and data structures sent back and forth. The incoming traffic to the vision server ADAGIA is equally low whereas the outgoing rates are about 25 times as high. This is expected since this computer transmits images

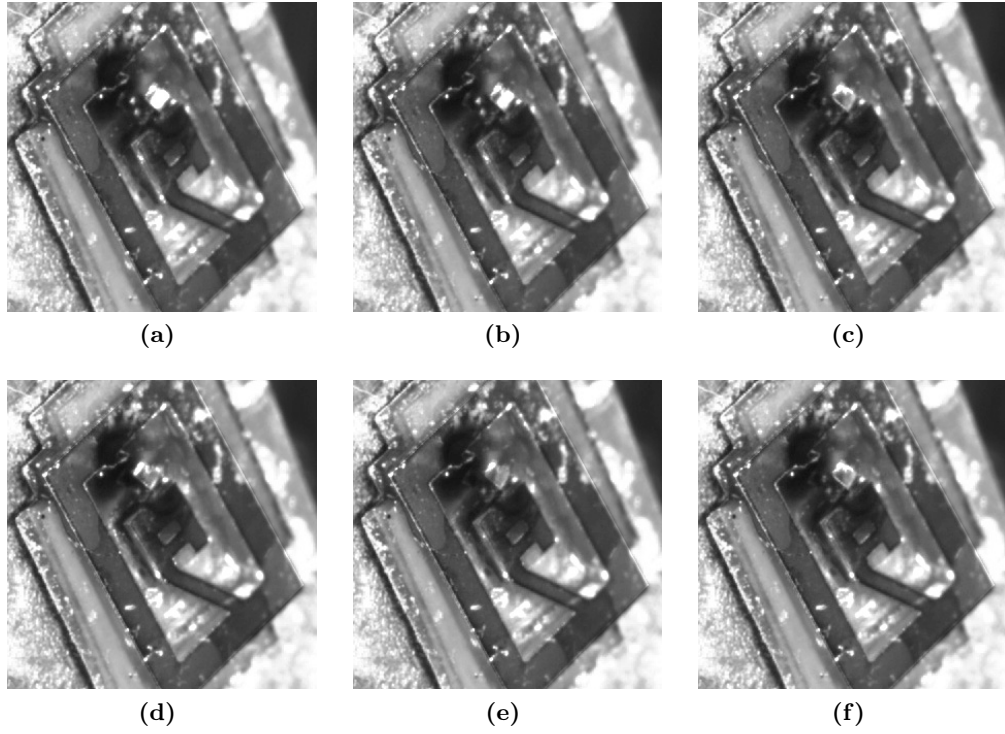


Fig. 6.25: Mechanical test of an ultrasonic transceiver by approaching the device with a permanent magnet from the bottom.

from all active cameras to the main workstation. Due to the fact that the images are compressed to JPEG, the outgoing rates have a high variance depending on the image composition (i.e. uniformly colored areas). The main workstation COLLOQUIA is the gate for all traffic and thus yields high incoming (image data) and low outgoing (command and data structures) data rates.

		COLLOQUIA	ADAGIA	ERASMUS
average rates	in	193.74	6.70	1.53
	out	9.06	166.64	6.11
	total	202.80	173.28	7.64
peak rates	in	286.74	9.83	1.94
	out	16.59	248.73	6.67
	total	303.33	258.56	8.23

all values in kB/s

Table 6.6: Network interface traffic between the three computing units.

Conclusions and Outlook

This chapter contains two main sections. The first one summarizes the contributions of this thesis to the related communities. The second section lists a selection of potential improvements in the fields of hard- and software. A few concluding remarks round off the main body of this work.

7.1 Contributions

The development of advanced 3D hybrid MEMS devices for bio-medical applications is impossible without feasible micro-assembly processes. However, complex three-dimensional designs, high integration, and different materials make it impossible to use standard micro-fabrication processes. Instead, micro-manipulation and -assembly concepts provide promising methods for a large variety of tasks. Unfortunately, the handling of tiny objects is as complicated as their design is, primarily due to the vastly different physics at the micro-scale. The understanding of those issues and the translation to real applications and hardware is therefore crucial for a wide acceptance of micro-assembly systems in research and industrial applications.

The main contribution to this research is the design and development of an advanced micro-assembly system capable of manipulating MEMS components of various sizes at high precision, using an intuitive haptic and visual interface. The machine features full six degrees of freedom in a remote center of motion configuration, and can be equipped with different types of grippers. The large linear axis provides easy access to the workbench for loading and unloading parts, and offers flexibility for the development of automatic part feeders in the future. A novel vision and illumination concept for micro-manipulation, consisting of multiple individually controllable high power LEDs as well as up to three CCD cameras with microscope lenses, has also been integrated. Custom electronics keep cameras and illumination in sync at any frequency and duty cycle, and the IEEE 1394 camera framework has been extended in order to make use of all available features. The device is designed to work under non-cleanroom conditions in regular atmosphere.

Based on this solid electro-mechanical framework, the second big achievement has been made in the field of user interaction. Intuitive interface concepts are one of the key ele-

ments for wide acceptance of these systems and thus it has been an early target from the very beginning on to improve the way operators deal with tiny movements and forces. The development and integration of an advanced virtual reality environment paired with two haptic input devices for both hands is a substantial contribution to the control of micro-manipulation systems. Clear and uncluttered views, movable viewports to any desired location, as well as full knowledge about object dependencies are only some of the key elements of the package. It has been proven that the design is robust and advantageous over previous techniques and generates a platform for a large number of new concepts.

The third component contributing to the field of modern micro-assembly is in the field of computer vision. The problem of localizing and retrieving the pose of known objects in a scene given their 3D models has been solved by introducing and adapting an existing algorithm. The core idea of the method is to randomly select an initial subset of correspondences, which map image to model features or vice versa, and refine them by linear search until a complete set of correspondences has been assembled. The actual extraction of position and orientation of the object involves the transformation of geometrical entities to as much as four mathematical spaces and yields a true 3D error measure in Euclidean space. While the model features are readily available from CAD data the image features are extracted by a special image processing unit. Last is capable of locating a series of single micro-robot parts and determine their features in 2D image space. Each camera has its own threaded image and pose processor which are automatically created when the system is started up.

A series of manipulation experiments focused on the assembly of bio-micro-robotic components have demonstrated the feasibility and ease of use of the system for a variety of applications. Even though the primary focus is on manipulation and assembly, the setup can be quickly configured for measurement purposes, where, for example, the mechanical or magnetic effects on miniature components have to be visually observed.

7.2 Extensions and Improvements

Even though the basic micro-assembly station V2 is an advanced system there are a number of improvements that can extend its capabilities. Some ideas are listed in the following two sections grouped into hardware and software related projects.

7.2.1 Hardware Improvements

Second Hand

The manual assembly of micro- or macroscopic parts is dominated by the enormous dexterity of human hands, which offer a large number of degrees of freedom. However, it has been shown that only the strong hand, holding a pair of tweezers or directly the component, makes use of the flexibility, whereas the other hand serves as a clamp primarily used for holding the target structure in place. This concept could also be adapted in the present system by introducing a second robust gripper on the platform. This device would

either be in a fixed horizontal or vertical position or able to rotate $\pm 45/90^\circ$. The current workbench is prepared for taking a second MEMS micro-gripper as shown in Figure 7.1a.

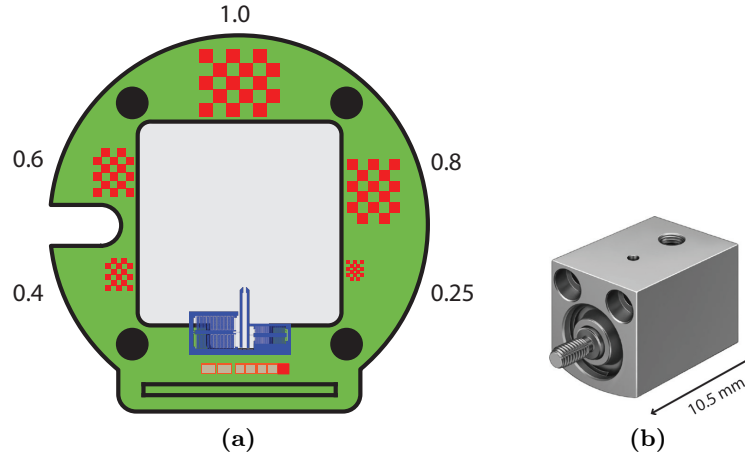


Fig. 7.1: *Second MEMS gripper mounted on the workbench (a) and pneumatic miniature actuator for the development of a larger scale micro-gripper (b).*

However, the integration of a larger scale mechanical micro-gripper, potentially with a rotational degree of freedom, requires additional developments. Tight space constraints most likely force the actuators to be positioned below the workbench and the use of tendon drives or rigid levers. The current setup already features two independently controllable miniature pneumatic actuators (see Figure 7.1b) as well as electrical contacts that could be used for driving a motor.

Joining Components

The present solution for making a permanent interconnection between two micro-components is based on a dispensing unit for UV curable glue (see Section 3.2.7). The individual air pressure and barrel vacuum control allows creating variable droplet sizes. However, close contact between the needle and the parts to be glued is required. Depending on the contact situation between the needle and the part, capillary and surface tension forces hold parts of the liquid on the needle at a position a little bit away from the tip, resulting in the accumulation of large quantities of glue after multiple cycles that cannot be sucked back by the vacuum. Dropping this excessive glue on a designated “deposit area” from time to time solves the problem of applying too much glue on a part and thus reduces hardening time. A more serious problem is the clogging of the needle due to UV illumination that can in most cases only be solved by replacing the needle.

One method to create smaller quantities consists of releasing a droplet on a flat surface and traversing it with a fine needle or toothpick (see Figure 3.22). The resulting fluid paths offer varying size but limit the application to convex surfaces that can be dipped.

If local deposition is indispensable the use of piezoelectric driven valves, that shoot off droplets down to a diameter of $30\text{ }\mu\text{m}$, offer a good solution.

A better but more expensive solution for applying liquid glue is the installation of a piezo driven droplet generator, which is able to precisely shoot tiny droplets on a target. This contactless approach allows creating extremely small droplets and is advantageous in every aspect.

Environmental Conditions

One advantage of micro-assembly is the fact that it does not require a clean room environment. In addition, the present system features a dome which provides constant illumination conditions as well as helps covering the platform and micro-parts. However, regular dust and other particles might impair the assembly process and thus a simple dust cover made out of acrylic glass can reduce this influence (see Figure 7.2). Apart from a door required for part feeding, the box is sealed and ventilated with pressurized air in order to create an outward air flow tendency.

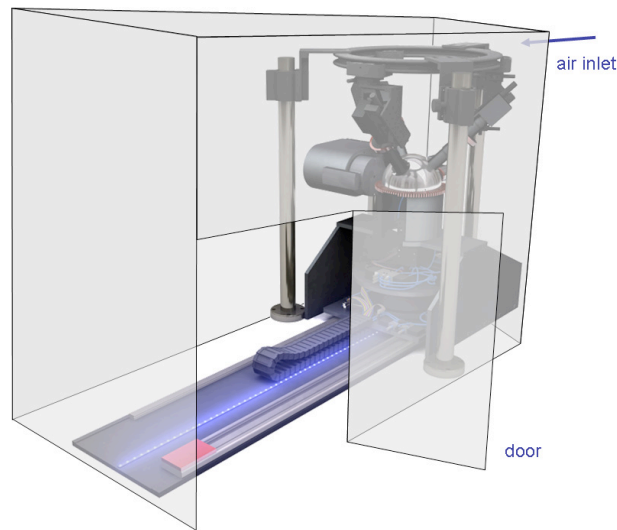


Fig. 7.2: Acrylic glass dust cover with door for accessing the workbench and inlet for pressurized air.

Computing Hardware Distribution

The balanced distribution of the three main tasks *hardware control*, *vision processing* and *haptic/visual interfaces* on three different hardware entities (Erasmus, Adagia, and Colloquia) has proven to be both necessary and effective. High demands on processing power for computer vision and visualization tasks require multiple processing units connected to

each other. However, the present interconnection over regular Gigabit Ethernet switches yields insufficient performance under certain situations (see Section 6.7), not least because of the additional external network traffic. In addition, the data flow is concentrated on the main application computer which makes it impossible to run close-to-realtime visual servoing routines.

For these reasons and additional future developments, an optimization of the communications architecture is required. Due to the growing availability of multi-core processor machines one could even think of using a single machine and thus gaining incredible performance in data transfer. The multi-threaded nature of all software components would simplify such a transition.

MEMS Force-Feedback Gripping

The possible use of a MEMS micro-gripper with integrated force-feedback is already mentioned in Section 3.2.3. The present system features all electronics to drive a gripper as an end effector as well as on the platform as a second hand (see Figure 7.3). As already mentioned, a software has been written up to stage that a desired voltage or force can be specified and is controlled using an implemented PID loop. The next step is to introduce this gripper type to the real system and experimentally verify its feasibility. It might also be beneficial to translate the PID controller to an FPGA in order to gain better performance and higher flexibility, as well as to reduce the computational load on the hardware controller.

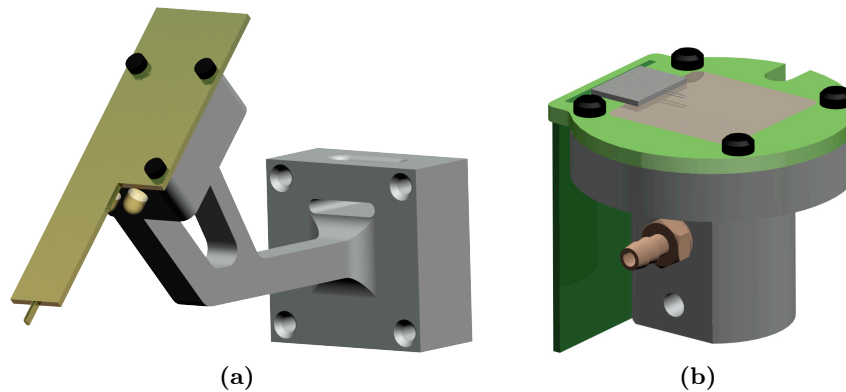


Fig. 7.3: MEMS micro-grippers mounted as an end effector on the manipulator (a) and on the workbench as a second hand (b).

Automatic Initialization of the Mechanical Gripper

Every axis of the micro-assembly station V2 can be automatically initialized to its zero position. This also applies to the calibration axes which store their zero positions after the calibration procedure. The only actuator unable to initialize itself at present is the

DC-motor driven mechanical micro-gripper. The simplest option is the use of a miniature switch that is triggered when the sliding tube reaches its rear position, i.e., an open gripper position. Applying a voltage to one of the gripper fingers or analyzing the gap using computer vision are other methods for first-time initialization.

7.2.2 Software Improvements

Drag and Move

The fundamental task of any micro-manipulation operation is to move an object from A to B. This task is usually executed by indirectly controlling a gripper using a haptic interface. A more intuitive solution would be to directly grab the object within the virtual environment and drag it to its new location B, with the consequence that the system automatically performs the required motions (see Figure 7.4). Such an environment demands accurate and robust visual servoing and path planning capabilities.

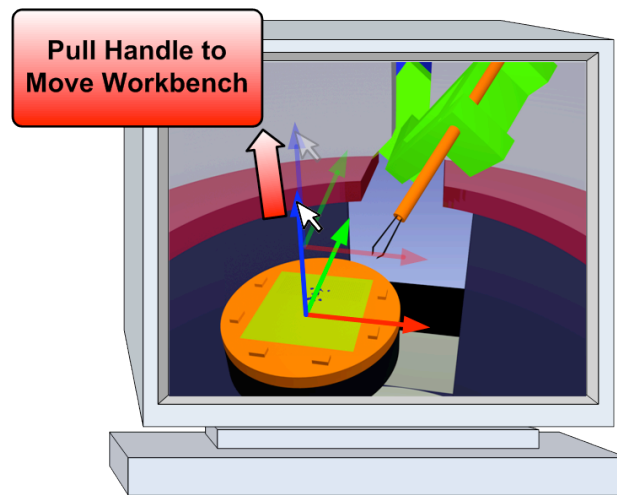


Fig. 7.4: *Moving objects by pulling their handles.*

Microphysics Simulation

The virtual environment can be used to perform dry runs in order to assess a programmed assembly sequence or, as already explained in Section 4.2, to avoid potential collisions. The integration of microphysics within the virtual environment would make the simulation more natural and be very beneficial for the analysis of complex assembly tasks. However, the big unknown is as to how well a physics engine is capable of capturing the numerous tiny influences on micro-parts. This problem set is certainly all but trivial and the final success unclear.

Visualization

Since the virtual reality interface is all but a static concept it provides room for a lot of improvements and additions. Some of them are briefly listed here:

- Integration of the glue dispenser needle into the scenegraph and thus visualizing it. A method for calibrating, i.e., aligning, the needle position will be required.
- Simultaneous visualization of multiple 3D objects. This also requires an interface class for accessing individual components and change their properties, i.e., position and orientation.
- Distance measurement tool in both the virtual reality as well as the camera views. Due to the known manipulator position and orientation as well as constant focal length, this extension should be simple but helpful.
- *Click and move* functionality for rapid manipulator movement to any desired location analog to *drag and move* explained above.
- Full implementation and visualization of the object dependency graph. This functionality is related to the development of a task planning system.

Pose Estimation

As explained in Chapter 5 the localization routine consists of three major problem sets, namely image feature extraction, correspondence matching and pose determination. The last one is fast and robust, and is therefore only a candidate for minor improvements. However, the other two areas are of higher complexity and have more potential for extensions.

The overall robustness and flexibility of the image feature extractor can be improved. Since the present version is limited to a small number of micro-robot parts and requires recompilation of the whole computer vision library upon any changes, the use of a plugin system would be beneficial. This way, individual algorithms could be quickly tested offline and then loaded by the system.

Improving the robustness of the feature detectors is going to be important when working with multiple objects and cluttered environments. Some experiments have already been made with a more sophisticated concept described in Lowe [96, 97, 98, 99, 100]. This method is based on the human vision system with its ability to group objects to higher order entities — a process known as perceptual grouping. Figure 7.5 shows three examples that satisfy the constraints of key features (invariance in rotation, translation and scale). The features are located using a probabilistic approach under the assumption that they are uniformly distributed.

Basic tests using images from the macro-world yielded impressive results (see Figure 7.6) and it is believed that the method might be well suited for the micro-world, too.

Additional work will also be necessary in the field of combining pose results from different cameras. Finding a strategy that checks for similar results but also makes the right decision upon non-similar matrices will be crucial. This could be integrated in some sort of health monitoring routine which continuously checks incoming results.

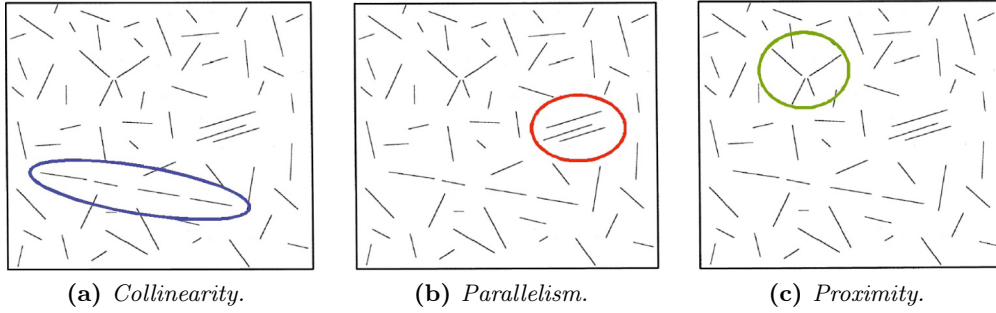


Fig. 7.5: *Three basic relations of perceptual grouping.*

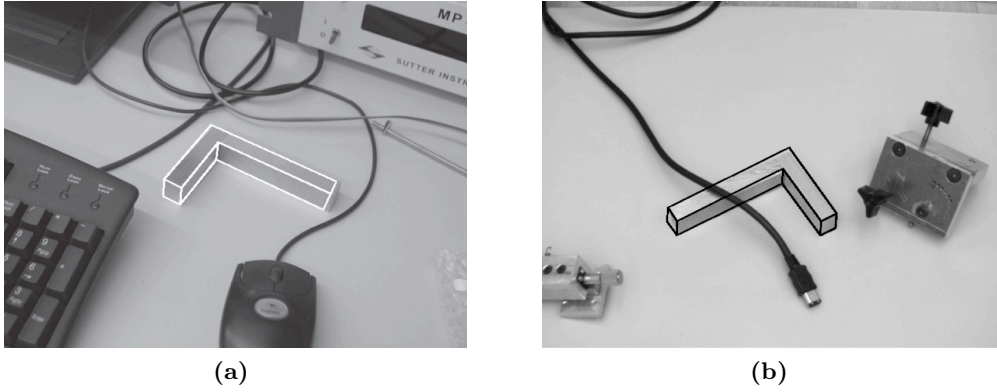


Fig. 7.6: *Pose estimation using perceptual grouping.*

Automatic Focus Adjustment

Every camera unit can move along its optical axis with high precision. The setup of a focus adjustment routine, based either on user or vision data, would be very useful. In the first case, the user could specify an object desired to be in focus and the stages would be adjusted according to the known kinematic configuration. In the latter case, a regular autofocus algorithm, similar to the ones in SLR cameras, would be used. In this case it might be necessary to define a limited active area in the image where the algorithm works on, in order to reduce problems with out of plane objects (e.g., gripper).

The implementation of an autofocus might also require updating the linear stages of the ring units. Even though they feature high precision ($0.014\mu\text{m}$) their maximum speed is rated at a low 0.375 mm/s . This might be too slow for given manipulation speeds.

Online Calibration

Camera calibration determines intrinsic and extrinsic parameters with respect to a common world coordinate frame \mathcal{W} , as explained in detail in Appendix B. This semi-automated procedure does not have to be repeated as long as the camera kinematics

(see Figure 3.18b) are kept constant. Once the cameras are re-initialized (after loss of power of the hardware controller), they can be driven back to their last known calibration position using the main control panel. In other words, movements along the motorized optical axis f_z can be compensated for. However, manual rotations around the center ω as well as pan α and tilt β motions cannot be detected and thus require re-calibration at least of the affected camera.

Continuous adjustment of the extrinsic camera calibration matrices upon camera movement is a useful and interesting feature to develop.

Tracking and Visual Servoing

The continuous observation of objects, known as visual tracking, and their guidance using visual feedback, known as visual servoing, are key elements for the automated assembly of micro-parts. A big leap forward is thus to setup a control loop as depicted in Figure 7.7.

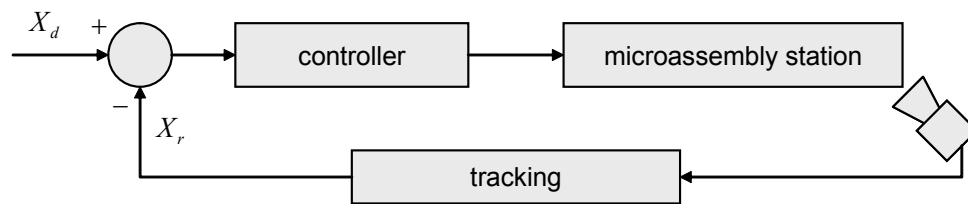


Fig. 7.7: Visual servoing control loop. The effective object pose X_r determined by visual tracking is subtracted from a desired pose X_d and the difference fed to a controller.

Since the hardware controller **Erasmus** is not directly linked with the vision server **Adagia** and all data is running over the workstation's mediator core (see Figure 3.33), there is an inevitable delay in the order

that prevents running an efficient control loop. Therefore, a future step is to directly link the vision server and the hardware controller.

Project Website

Improvements and extensions of any kind should be documented on the micro-assembly Wiki located at <http://www.microassembly.ch>. This page also features a link to a **Bugzilla** installation for tracking software related bugs.

7.3 Concluding Remarks

The field of micro-assembly is a fascinating world with its own laws. From an engineering standpoint of view it is particularly challenging due to the different involved competencies

in mechanical, electrical and software engineering, as well as micro-physics and micro-fabrication. In addition, the bright prospects of micro-assembly systems, both for research and industrial applications, are a central element for motivation and so is the readily visible success. However, the assembly of miniature devices is generally afflicted with a practical component, that can be best expressed as the frustration factor f and can be written in imperfect form as

$$f \approx \frac{1}{l^3} C \ .$$

Parameter l is the average characteristic length of the micro-parts involved in the experiment and C is a variable containing other influences that are neglected at this point. In other words, the frustration factor rapidly increases with reducing part dimensions. The author believes that the underlying system developed during the past years can in most cases scale the exponent of the previous equation down to 2, and future work is certain to carry on this trend.



Fig. 7.8: *RB1: The most important haptic input device...*

A few situations still have the potential to drive the frustration factor to infinity ($f \mapsto \infty$) and there is no apparent solution for this case. However, it has been observed that RB1 (see Figure 7.8) has a certain influence on long-tail claims. It is therefore advisable to keep this element in range of the operator at any time during assembly.

References

- [1] J. Abbott, Z. Nagy, F. Beyeler, and B. Nelson. Robotics in the small, part i: Microbotics. *IEEE Robotics & Automation Magazine*, 14(2):92–103, 2007.
- [2] J. Agnus, P. De Lit, C. Clevy, and N. Chaillet. Description and performances of a four-degrees-of-freedom piezoelectric gripper. *Assembly and Task Planning, 2003. Proceedings of the IEEE International Symposium on*, pages 66–71, 2003.
- [3] J. Alex, B. Vikramaditya, and B. Nelson. Teleoperated micromanipulation within a VRML environment using Java. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1747–1752, 1998.
- [4] S. Allegro, C. Chanel, and J. Jacot. Autofocus for automated microassembly under a microscope. In *Proc. International Conference on Image Processing*, volume 2, pages 677–680, 1996.
- [5] M. Ammi and A. Ferreira. Path planning of an AFM-based nanomanipulator using virtual force reflection. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, volume 1, pages 577–582, 2004.
- [6] M. Ammi, A. Ferreira, and J.-G. Fontaine. Virtualized reality interface for tele-micromanipulation. In *Proc. IEEE International Conference on Robotics and Automation ICRA '04*, volume 3, pages 2776–2781, 2004.
- [7] N. Ando, P. Korondi, and H. Hashimoto. Development of micromanipulator and haptic interface for networked micromanipulation. *IEEE/ASME Transactions on Mechatronics*, 6(4):417–427, 2001. ISSN 1083-4435.
- [8] S. Ando. Image field categorization and edge/corner detection from gradient covariance. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 179–190, 2000.
- [9] F. Arai, D. Ando, and T. Fukuda. Micro manipulation based on micro physics - strategy based on attractive force reduction and stress measurement. In *International Conference on Intelligent Robots and Systems*, volume 2, pages 236–241, 1995.

- [10] F. Arai, D. Andou, and T. Fukuda. Adhesion forces reduction for micro manipulation based on micro physics. *Micro Electro Mechanical Systems, 1996, MEMS '96, Proceedings. 'An Investigation of Micro Structures, Sensors, Actuators, Machines and Systems'. IEEE, The Ninth Annual International Workshop on*, pages 354–359, 1996.
- [11] H. Araujo, R. L. Carceroni, and C. M. Brown. A fully projective formulation for lowe's tracking algorithm. Technical report, University of Rochester, Nov. 1996 1996.
- [12] G. Bachler, M. Berger, R. Röhrer, S. Scherer, and A. Pinz. A vision driven automatic assembly unit. In *8th Conference on Computer Analysis of Images and Patterns CAIP99*, pages 375–382, Ljubljana, Slovenia, 1999.
- [13] C. Bark, T. Binnenboese, G. Voegelé, T. Weisener, and M. Widmann. Gripping with low viscosity fluids. *Proc. Eleventh Annual International Workshop on Micro Electro Mechanical Systems MEMS 98*, pages 301–305, 1998.
- [14] E. W. Becker, W. Ehrfeld, P. Hagmann, A. Maner, and D. Münchmeyer. Fabrication of microstructures with high aspect ratios and great structural heights by synchrotron radiation lithography, galvanofarming, and plastic moulding (LIGA process). *Microelectronic Engineering*, 4:35 – 56, 1986.
- [15] A. K. Behera, G. Kopec, S. G. Kapoor, and R. Devor. Lecture: Microassembly at the meso/micro scale.
- [16] B. A. Berkowitz and C. A. Wilson. Quantitative mapping of ocular oxygenation using magnetic resonance imaging. *Magnetic Resonance in Medicine*, 33(4):579–581, Nov 1995.
- [17] J. R. Beveridge. *Local Search Algorithms for Geometric Object Recognition: Finding the Optimal Correspondence and Pose*. Phd, University of Massachusetts, 1993.
- [18] F. Beyeler, D. J. Bell, and B. J. Nelson. Design of a micro-gripper and an ultrasonic manipulator for handling micron sized objects. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 772–777, 2006.
- [19] H. Bischof and A. Leonardis. Subspace methods for visual learning and recognition. Technical report, Institute for Computer Graphics and Vision, 2003.
- [20] K. F. Böhringer, K. Goldberg, M. Cohn, R. Howe, and A. Pisano. Parallel microassembly with electrostatic force fields. *Proc. IEEE International Conference on Robotics and Automation*, 2:1204–1211, 1998.
- [21] K. F. Böhringer, R. S. Fearing, and K. Y. Goldberg. Chapter microassembly. In S. Y. Nof, editor, *The Handbook of Industrial Robotics*. New York, NY: Wiley & Sons, 1999.
- [22] J. M. Breguet, E. Pernette, and R. Clavel. Stick and slip actuators and parallel architectures dedicated to microrobotics. *Proc. SPIE*, 2906:13–24, 1996.
- [23] H. V. Brussel, J. Peirs, D. Reynaerts, A. Delchambre, G. Reinhart, N. Roth, M. Weck, and E. Zussman. Assembly of microsystems. *Annals of the CIRP*, 49: 451–472, 2000.

- [24] L. Bruzzone and R. M. Molfino. A novel parallel robot for current microassembly applications. *Assembly Automation*, 26/4:299–306, 2006.
- [25] A. Bürkle and S. Fatikow. Laser measuring system for a flexible microrobot-based micromanipulation station. In S. Fatikow, editor, *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)*, volume 1, pages 799–804, 2000.
- [26] A. Bürkle, F. Schmoeckel, H. Wörn, B. P. Amavasai, F. Caparrelli, and J. R. Travis. A versatile vision system for micromanipulation tasks. In *International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2001)*, pages 271–276, Baden-Baden, 2001.
- [27] R. J. Campbell and P. J. Flynn. A survey of free-form object representation and recognition techniques. *Comput. Vis. Image Underst.*, 81(2):166–210, 2001. ISSN 1077-3142.
- [28] R. L. Carceroni and C. M. Brown. Numerical methods for model-based pose recovery. Technical Report 659, University of Rochester, 1997.
- [29] C. Cassier, A. Ferreira, and S. Hirai. Combination of vision servoing techniques and VR-based simulation for semi-autonomous microassembly workstation. In *Proc. IEEE International Conference on Robotics and Automation ICRA*, volume 2, pages 1501–1506, 2002.
- [30] J. Cecil, D. Powell, and D. Vasquez. Assembly and manipulation of micro devices—a state of the art survey. *Robotics and Computer-Integrated Manufacturing*, 23(5): 580–588, Oct. 2007.
- [31] H. Chen. Pose determination from line-to-plane correspondences: existence condition and closed-form solutions. In *Proc. Third International Conference on Computer Vision*, pages 374–378, 4–7 Dec. 1990.
- [32] L. Chen, L. Zhang, Y. Hu, M. Li, and H. Zhang. Head pose estimation using fisher manifold learning. In *Proc. IEEE International Workshop on Analysis and Modeling of Faces and Gestures AMFG*, pages 203–207, 2003.
- [33] M. Chen, M. Chen, S. Kume, A. Rizzi, and R. Hollis. Visually guided coordination for distributed precision assembly. In S. Kume, editor, *Proc. IEEE International Conference on Robotics and Automation ICRA*, volume 2, pages 1651–1656, 2000.
- [34] N. Chronis and L. P. Lee. Polymer MEMS-based microgripper for single cell manipulation. *Proc. 17th IEEE International Conference on Micro Electro Mechanical Systems (MEMS)*, pages 17–20, 2004.
- [35] C. Clévy, A. Hubert, and N. Chaillet. Micromanipulation and micro-assembly systems. *IARP’06 Workshop on Micro & nano robotics*, 2006.
- [36] A. Codourey, W. Zesch, R. Büchi, and R. Siegwart. A robot system for automated handling in micro-world. In *International Conference on Intelligent Robots and Systems*, volume 3, pages 185–190, 1995.

- [37] M. B. Cohn, K. F. Boehringer, J. M. Noworolski, A. Singh, C. G. Keller, K. A. Goldberg, and R. T. Howe. Microassembly technologies for MEMS. In A. B. Frazier and C. H. Ahn, editors, *Proceedings of the SPIE Conference on Micromachined Devices and Components IV*, volume 3515, pages 2–16. SPIE, 1998.
- [38] J. J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989. ISBN 0201095289.
- [39] CSEM. A gripper for miniaturized assembly. online, 1998.
- [40] N. Dechev, W. L. Cleghorn, and J. K. Mills. Construction of 3D MEMS microstructures using robotic microassembly. In *Sensing and Manipulation of Micro and Nano Entities: Science, Engineering, and Applications, Workshop, International Conference on Robots and Intelligent Systems (IEEE/RSJ IROS 2003)*, 2003.
- [41] N. Dechev, W. L. Cleghorn, and J. K. Mills. Microassembly of 3-D microstructures using a compliant, passive microgripper. *Journal of Microelectromechanical Systems*, 13(2):176–189, 2004.
- [42] N. Dechev, J. K. Mills, and W. L. Cleghorn. Mechanical fastener designs for use in the microassembly of 3d microstructures. In *Proceedings of the ASME International Mechanical Engineering Congress IMECE*, 2004.
- [43] N. Dechev, L. Ren, W. Liu, W. Cleghorn, and J. Mills. Development of a 6 degree of freedom robotic micromanipulator for use in 3D MEMS microassembly. In *Proceedings IEEE International Conference on Robotics and Automation ICRA*, pages 281–288, 2006.
- [44] L. Dorst. Geometric (clifford) algebra: A practical tool for efficient geometrical representation. Technical report, Intelligent Autonomous Systems, Department of Computer Science, May 1999.
- [45] E. R. Dowski and W. T. Cathey. Extended depth of field through wave-front coding. *Applied Optics*, 34(11):8, 1995.
- [46] T. Drummond and R. Cipolla. Real-time tracking of multiple articulated structures in multiple views. In *ECCV (2)*, pages 20–36, 2000.
- [47] D. W. Eggert, A. Lorusso, and R. B. Fisher. Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Mach. Vision Appl.*, 9(5-6):272–290, 1997. ISSN 0932-8092.
- [48] W. Ehrfeld, M. Begemann, U. Berg, A. Lohf, F. Michel, and M. Nienhaus. Highly parallel mass fabrication and assembly of microdevices. *Microsystem Technologies*, 7:145–150, 2001.
- [49] S. Ekvall, F. Hoffmann, and D. Kragic. Object recognition and pose estimation for robotic manipulation using color cooccurrence histograms. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, volume 2, pages 1284–1289, 27–31 Oct. 2003.
- [50] S. Fatikow, S. Fatikow, and U. Rembold. An automated microrobot-based desktop station for micro assembly and handling of micro-objects. In U. Rembold, editor, *Proc. IEEE Conference on Emerging Technologies and Factory Automation EFTA*, volume 2, pages 586–592, 1996.

- [51] S. Fatikow, J. Seyfried, S. Fahlbusch, A. Buerkle, and F. Schmoeckel. A flexible microrobot-based microassembly station. *Journal of Intelligent and Robotic Systems*, 27:135–169, 2000.
- [52] R. S. Fearing. Survey of sticking effects for micro parts handling. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 212–217, Pittsburgh, PA, 1995.
- [53] J. Feddema, J. Feddema, and R. Simon. Cad-driven microassembly and visual servoing. In R. Simon, editor, *Proc. IEEE International Conference on Robotics and Automation*, volume 2, pages 1212–1219, 1998.
- [54] J. T. Feddema, C. S. G. Lee, and O. R. Mitchell. Weighted selection of image features for resolved rate visual feedback control. *IEEE Transactions on Robotics and Automation*, 7(1):31–47, 1991.
- [55] J. T. Feddema, P. Xavier, and R. Brown. Micro-assembly planning with van der Waals force. *Proc. IEEE International Symposium on Assembly and Task Planning (ISATP)*, pages 32–38, 1999.
- [56] A. Ferreira. Strategies of human-robot interaction for automatic microassembly. In *Proc. IEEE International Conference on Robotics and Automation ICRA*, volume 3, pages 3076–3081, 2003.
- [57] A. Ferreira, C. Cassier, and S. Hirai. Automatic microassembly system assisted by vision servoing and virtual reality. *IEEE/ASME Transactions on Mechatronics*, 9(2):321–333, 2004. ISSN 1083-4435.
- [58] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, pages 726–740, 1987.
- [59] J. Fraser, T. Hubbard, and M. Kujath. Theoretical and experimental analysis of an off-chip microgripper. *Canadian Journal of Electrical and Computer Engineering*, 31:77–84, 2006.
- [60] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Professional, 1995.
- [61] B. Gerkey, R. Vaughan, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the International Conference on Advanced Robotics (ICRA 2003)*, pages 317–323, June 30 – July 3 2003.
- [62] O. Ghita and P. F. Whelan. A bin picking system based on depth from defocus. *Maschine Vision and Applications*, 13:234–244, 2003.
- [63] J. S. Goddard. *Pose And Motion Estimation From Vision Using Dual Quaternion-Based Extended Kalman Filtering*. PhD thesis, University of Tennessee, Knoxville, 1997.
- [64] W. E. L. Grimson. *Object Recognition by Computer*. The MIT Press, Cambridge, MA, 1990.
- [65] Y. Haddab, N. Chaillet, and A. Bourjault. A microgripper using smart piezoelectric actuators. *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)*, 1:659–664, 2000.

- [66] M. Hamdi and A. Ferreira. Microassembly planning using physical-based models in virtual environment. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, volume 4, pages 3369–3374, 2004.
- [67] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.
- [68] V. Hess. Processes and tools for robotic microassembly. Diplomarbeit, ETH Zurich, 2004.
- [69] T. Higuchi. Innovative actuators and tools for micro-nano mechatronics. *IEEE International Symposium on Micro-NanoMechatronics and Human Science*, pages 5–9, 2005.
- [70] B. Hohnhaeuser and G. Hommel. Object identification and pose estimation for automatic manipulation. In *International Workshop RobVis 2001*, Lecture Notes in Computer Science, pages 52–59, Auckland, New Zealand, 2001.
- [71] R. L. Hollis and J. Gowdy. Miniature factories for precision assembly. *Proc. International Workshop on Micro-Factories, Tsukuba, Japan*, pages 9–14, 1998.
- [72] R. L. Hollis, J. Gowdy, and A. A. Ritzi. Design and development of a tabletop precision assembly system. *Mechatronics and Robotics MechRob*, pages 1619 – 1623, 2004.
- [73] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4):629–642, 1986.
- [74] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America*, 5(7):1127–1135, 1988.
- [75] K. Houston, C. Eder, A. Sieber, A. Mencias, D. H. Kan, B. Kim, M. Carrozza, and P. Dario. Polymer sensorised microgrippers using SMA actuation. *IEEE International Conference on Robotics and Automation*, pages 820–825, 2007.
- [76] E. Huber and K. Baker. Using a hybrid of silhouette and range templates for real-time pose estimation. In *Proc. IEEE International Conference on Robotics and Automation ICRA*, volume 2, pages 1652–1657, Apr 26–May 1 2004.
- [77] R. J. K. Jacob, L. E. Sibert, D. C. McFarlane, and J. M. Preston Mullen. Integrality and separability of input devices. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 1(1):3–26, 1994. ISSN 1073-0516.
- [78] K. Kanatani. Analysis of 3-d rotation fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):543–549, 1994.
- [79] M. Kemper. Development of a tactile low-cost microgripper with integrated force sensor. *Proceedings of the 2004 IEEE International Conference on Control Application*, pages 1461–1466, 2004.
- [80] D. Kim, B. Kim, and D. H. Kan. Development of a piezoelectric polymer-based sensorized microgripper for microassembly and micromanipulation. *Microsystem Technologies*, 10:275–280, 2004.

- [81] V. C. Klema and A. J. Laub. The singular value decomposition: Its computation and some applications. *IEEE Transactions on Automatic Control*, AC-25(2):164–176, 1980.
- [82] A. Kochan. European project develops ice gripper for micro-sized components. *Assembly Automation*, 17:114–115, 1997.
- [83] M. Kohl, B. Krevet, and E. Just. SMA microgripper system. *Sensors and Actuators*, pages 646–652, 2001.
- [84] A. Kortschack and S. Fatikow. Development of a mobile nanohandling robot. *Journal of Micromechanics*, Vol. 2:249–269, 2004.
- [85] K. Koyano and T. Sato. Micro object handling system with concentrated visual fields and new handling skills. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2541–2548, Apr. 1996.
- [86] D. Kragic and H. Christensen. Model based techniques for robotic servoing and grasping. In H. Christensen, editor, *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 299–304, 2002.
- [87] B. E. Kratochvil, K. B. Yeşin, V. Hess, and B. J. Nelson. Design of a visually guided 6 DOF micromanipulator system for 3D assembly of hybrid MEMS. In *Proceedings of the 4th International Workshop on Microfactories*, Oct. 2004.
- [88] P. Krulevitch, A. P. Lee, P. B. Ramsey, J. C. Trevino, J. Hamilton, and M. A. Northrup. Thin film shape memory alloy microactuators. *Journal of Microelectromechanical Systems*, 5:270–282, 1996.
- [89] M. P. Kummer. Robotic wrist system for microassembly. Diplomarbeit, ETH Zurich, 2004.
- [90] D. Lang, M. Tichem, and F. Warner. An industrial prototype of a liquid solidification based micro-gripping system. *IEEE International Symposium on Assembly and Manufacturing ISAM*, pages 227–232, 2007.
- [91] M. Last, V. Subramaniam, and K. Pister. Out of plane motion of assembled microstructures using a single-mask soi process. In *The 13th International Conference on Solid-State Sensors, Actuators and Microsystems. Digest of Technical Papers. TRANSDUCERS*, volume 1, pages 684–687, 2005.
- [92] S. H. Lee, K.-C. Lee, S. S. Lee, and H.-S. Oh. Fabrication of an electrothermally actuated electrostatic microgripper. *12th International Conference on TRANSDUCERS, Solid-State Sensors, Actuators and Microsystems*, 1:552–555, 2003.
- [93] S. J. Lee, K. Kim, D.-H. Kim, J.-O. Park, and G. T. Park. Recognizing and tracking of 3d-shaped mirco parts using multiple visions for micromanipulation. In *International Symposium On Micromechatronics And Human Science*, pages 203–210, 2001.
- [94] S. Z. Li, Q. D. Fu, L. Gu, B. Schölkopf, Y. Cheng, and H. J. Zhang. Kernel machine based learning for multi-view face detection and pose estimation. In *International Conference on Computer Vision and Pattern Recognition*, pages 674–679, Vancouver, Canada, 2001.

- [95] A. Lorusso, D. W. Eggert, and R. B. Fisher. A comparison of four algorithms for estimating 3-d rigid transformations. In *British Machine Vision Conference*, pages 237–246, Birmingham, 1995.
- [96] D. Lowe. Object recognition from local scale-invariant features. In *Proc. IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157, 20–27 Sept. 1999.
- [97] D. G. Lowe. Solving for the parameters of object models from image descriptions. In *Proc. ARPA Image Understanding Workshop (College Park, MD)*, pages 121–127, 1980.
- [98] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.
- [99] D. G. Lowe. Fitting parameterized 3-d models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, 1991.
- [100] D. G. Lowe. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2):113–122, 1992. ISSN 0920-5691.
- [101] C. Lu, G. D. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622, 2000.
- [102] J. K. Luo, J. H. He, A. J. Flewitt, S. M. Spearing, N. A. Fleck, and W. I. Milne. Microgripper using a diamond like carbon/metal bilayer. *The Fifth International Conference on Advanced Semiconductor Devices and Microsystems ASDAM*, pages 283–286, 2004.
- [103] M. Mayyas, P. Shiakolas, W. H. Lee, D. Popa, and H. Stephanou. Static and dynamic modeling of thermal microgripper. *14th Mediterranean Conference on Control and Automation MED*, pages 1–6, 2006.
- [104] T. Melzer, M. Reiter, and H. Bischof. Appearance models based on kernel canonical correlation analysis. *Pattern Recognition*, 36(9):1961–1971, 2003. ISSN 0031-3203.
- [105] A. Menciassi, A. Eisinberg, G. Scaliari, C. Anticoli, M. Carrozza, and P. Dario. Force feedback-based microinstrument for measuring tissue properties and pulse in microsurgery. *Proceedings of the IEEE International Conference on Robotics and Automation*, 1:626–631, 2001. ISSN 1050-4729.
- [106] A. Menciassi, A. Eisinberg, M. Mazzoni, and D. P. Dario. A sensorized μ electro discharge machined superelastic alloy microgripper for micromanipulation: simulation and characterization. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1591–1595, 2002.
- [107] M. Mitsuishi, Y. Izuka, H. Watanabe, H. Hashizume, and K. Fujiwara. Remote operation of a micro-surgical system. *Proceedings of the IEEE International Conference on Robotics & Automation*, 2:1013–1019, 1998.
- [108] R. M. Murray, S. S. Sastry, and L. Zexiang. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., Boca Raton, FL, USA, 1994. ISBN 0849379814.

- [109] N. Navab and O. Faugeras. Monocular pose determination from lines: critical sets and maximum number of solutions. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR*, pages 254–260, 15–17 June 1993.
- [110] S. K. Nayar, S. A. Nene, and H. Murase. Subspace methods for robot vision. *IEEE Transactions on Robotics and Automation*, 12(5):750–758, Oct. 1996. ISSN 1042-296X.
- [111] B. J. Nelson and P. K. Khosla. The resolvability ellipsoid for sensor based manipulation. Technical report, The Robotics Institute, Carnegie Mellon University, Oct. 1993.
- [112] B. J. Nelson and P. K. Khosla. Vision resolvability for visually servoed manipulation. *Journal of Robotic Systems*, 13(2):75–93, 1996.
- [113] B. J. Nelson, Y. Zhou, and B. Vikramaditya. Sensor-based microassembly of hybrid mems devices. *IEEE Control Systems Magazine*, 18(6):35–45, 1998. ISSN 0272-1708.
- [114] N.-T. Nguyen, S.-S. Ho, and C. L.-N. Low. A polymeric microgripper with integrated thermal actuators. *Journal of Micromechanics and Microengineering*, 14:969–974, 2004.
- [115] K. Nickels and S. Hutchinson. Model-based tracking of complex articulated objects. *IEEE Transactions on Robotics and Automation*, 17(1):28–36, 2001.
- [116] P. L. Palmer, J. Kittler, and M. Petrou. An optimizing line finder using a hough transform algorithm. *Computer Vision And Image Understanding*, 67(1):1–23, 1995.
- [117] N. Papanikolopoulos, P. K. Khosla, and T. Kanade. Vision and control techniques for robotic visual tracking. In *International Conference on Robotics and Automation*, pages 875–864, Sacramento, California, 1991.
- [118] I. Pappas and A. Cordourey. Visual control of a microrobot operating under a microscope. In *International Conference on Intelligent Robots and Systems*, volume 2, pages 993–1000, 1996.
- [119] J. Park and W. Moon. A hybrid-type micro-gripper with an integrated force sensor. *Microsystem Technologies*, 9:511–519, 2003.
- [120] J. Park and W. Moon. The systematic design and fabrication of a three-chopstick microgripper. *The International Journal of Advanced Manufacturing Technology*, 26:251–261, 2004.
- [121] J. Peirs, D. Reynaerts, and H. V. Brussel. Scale effects and thermal considerations for micro-actuators. *IEEE International Conference on Robotics and Automation*, 2:1516–1521, 1998.
- [122] T. Phong, R. Horaud, A. Yassine, and P. Tao. Object pose from 2-d to 3-d point and line correspondences. *International Journal of Computer Vision*, 15(3):225–243, July 1995.
- [123] M. Probst, K. Vollmers, B. E. Kratochvil, and B. J. Nelson. Design of an advanced microassembly system for the automated assembly of bio-microrobots. In *Proc. 5th International Workshop on Microfactories*, Oct. 2006.

- [124] A. E. Quaid and R. L. Hollis. Cooperative 2-dof robots for precision assembly. *Proceedings of the IEEE International Conference on Robotics and Automation*, 3: 2188–2193, 1996.
- [125] R. Raskar, K.-H. Tan, R. Feris, and et al. Non-photorealistic camera: Depth edge detection and stylized rendering using multi-flash imaging. *ACM Transactions on Graphics*, 23(3):679–688, 2004.
- [126] A. A. Rizzi, J. Gowdy, and R. L. Hollis. Agile assembly architecture: An agent based approach to modular precision assembly systems. *Proc. IEEE International Conference on Robotics and Automation*, pages 1511–1516, 1997.
- [127] L. G. Roberts. Machine perception of three-dimensional solids. In J. Tippet, editor, *Optical and Electro-Optical Information Processing*, pages 159–197, 1965.
- [128] B. Rosenhahn. *Pose Estimation Revisited*. Ph.d. thesis, Christian-Albrechts-Universität, Kiel, Germany, Sept. 2003.
- [129] B. Rosenhahn and G. Sommer. Pose estimation in conformal geometric algebra. Technical report, Institut für Informatik und Praktische Mathematik, Nov. 2002.
- [130] B. Rosenhahn and G. Sommer. Adaptive pose estimation for different corresponding entities. *24th DAGM Symposium on Pattern Recognition*, 2449:265–273, 2002.
- [131] B. Rosenhahn, Y. Zhang, and G. Sommer. Pose estimation in the language of kinematics. In *AFPAC*, pages 284–293, 2000.
- [132] B. Rosenhahn, N. Krüger, T. Rabsch, and G. Sommer. Tracking with a novel pose estimation algorithm. In *International Workshop RobVis*, Lecture Notes in Computer Science, pages 9–18, Auckland, New Zealand, 2001.
- [133] B. Rosenhahn, C. Perwass, and G. Sommer. Pose estimation of free-form surface models. *DAGM Symposium on Pattern Recognition*, 2781:574–581, 2003.
- [134] J. M. B. Rubio, M. G. Hidalgo, and F. J. P. López. A new method for detection and initial pose estimation based on mumford-shah segmentation functional. Technical report, Departamento de Matemáticas e Informática Unidad de Gráficos y Visión Universitat de les Illes Balears, 2003.
- [135] N. Sarkar. Applying microassembly to real-world applications. *Zyver Application Note 9712*, 2005.
- [136] T. Sato, K. Koyano, M. Nakao, and Y. Hatamura. Novel manipulator for micro object handling as interface between micro and human worlds. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, 3: 1674–1681, 1993.
- [137] T. Sato, T. Kameya, H. Miyazaki, and Y. Hatamura. Hand-eye system in nano manipulation world. In *Proc. IEEE International Conference on Robotics and Automation*, volume 1, pages 59–66, 21–27 May 1995.
- [138] F. Schmoeckel and H. Woern. Remotely controllable mobile microrobots acting as nano positioners and intelligent tweezers in scanning electron microscopes (SEMs). *Proceedings of the IEEE International Conference on Robotics & Automation*, 4: 3909–3913, 2001.

- [139] D. Schreiner, M. Woo, J. Neider, and T. Davis. *OpenGL Programming Guide, The Official Guide to Learning OpenGL, Version 1.4*. Addison-Wesley, 2007.
- [140] M. Schuenemann, R. Grimme, T. Kaufmann, G. Schwaab, U. Bader, W. Schaefer, and J. Dorner. Manufacturing concepts and development trends in the industrial production of microelectromechanical systems. *SPIE Microrobots and Microsystem Fabrication*, 3202:1–12, 1997.
- [141] M. Sengel, M. Berger, V. Kravtchenko-Berejnoi, and H. Bischof. Fast object recognition and pose determination. In *International Conference on Image Processing*, volume 3, pages 349–352, 2002.
- [142] S. Shet, R. D. Revero, M. R. Booty, A. T. Fiory, M. P. Lepselter, and N. M. Ravindra. Microassembly techniques: A review. *Materials Science and Technology (MST)*, 1, 2006.
- [143] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, page 8, Seattle, 1994.
- [144] E. Shimada, J. A. Thompson, J. Yan, W. R., and R. Fearing. Prototyping millirobots using dextrous microassembly and folding. In *Proceedings of ASME IMECE / DSCD*, 2000.
- [145] E. Shimada, J. Thomson, J. Yan, R. Wood, and R. Fearing. Prototyping millirobots using dextrous microassembly and folding. *Symposium on Microrobotics ASME IMECE/DSCD*, pages 1–8, 2000.
- [146] R. Sim and G. Dudek. Learning and evaluating visual features for pose estimation. In *International Conference on Computer Vision*, volume 2, page 1217, Corfu, Greece, 1999.
- [147] R. Sim and G. Dudek. Comparing image-based localization methods. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.
- [148] A. Singh, D. A. Horsley, M. B. Cohn, A. P. Pisano, and R. T. Howe. Batch transfer of microstructures using flip-chip solder bump bonding. *Proc. International Conference on Solid-State Sensors and Actuators (Transducers '97)*, pages 265–268, 1997.
- [149] G. Sommer, B. Rosenhahn, and Y. Zhang. Pose estimation using geometric constraints. Technical report, Institut für Informatik und Praktische Mathematik, 2001.
- [150] M. W. Spong and M. Vidyasagar. *Robot Dynamics And Control*. Wiley & Sons, 1989.
- [151] J. Stavnitzky and D. Capson. Multiple camera model-based 3-D visual servo. *IEEE Transactions on Robotics and Automation*, 16(6):732–739, 2000.
- [152] S. A. Stoeter and N. Papanikolopoulos. Closed dynamic contour models that split and merge. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 3883–3888, New Orleans, Louisiana, USA, Apr./May 2004.

- [153] S. A. Stoeter, M. Probst, and M. A. Iranzo. Improving tracking precision for microassembly. In *IEEE International Conference on Robotics and Automation (ICRA)*, May 2006.
- [154] A. Sulzmann and J. Jacot. 3D computer graphics based interface to real microscopic worlds for & robot telemanipulation and position control. In *IEEE International Conference on Systems, Man and Cybernetics. 'Intelligent Systems for the 21st Century'*, volume 1, pages 286–291, Oct. 22–25 1995.
- [155] A. Sulzmann, J.-M. Breguet, and J. Jacot. Microvision system (MVS): a 3D computer graphic-based microrobot telemanipulation and position feedback by vision. *Proc. SPIE Photonics East*, 2593:38–49, 1995.
- [156] A. Sulzmann, J. Carlier, and J. Jacot. Distributed Microscopy: towards a 3D computer graphic-based multi user microscopic manipulation, imaging and measurement system. *Proc. SPIE Photonics East*, 2902, 1996.
- [157] A. Sulzmann, J. Carlier, and J. Jacot. Virtual Reality and high-accuracy vision feedback as key information for microrobot telemanipulation. *Proc. SPIE Photonics East Proc*, 2906, 1996.
- [158] A. Sulzmann, J.-M. Breguet, and J. Jacot. Micromotor assembly using high accurate optical vision feedback for microrobot relative 3D displacement in submicron range. In *IEEE International Conference on Solid-State Sensors and Actuators*, pages 279–282, Chicago, Illinois, USA, June 1997.
- [159] Y. Sun and B. J. Nelson. Biological cell injection using an autonomous microrobotic system. *International Journal of Robotics Research*, 21(10-11):861–868, 2003.
- [160] F. Tan, L. Sun, W. Rong, and J. Zhu. A collision response method in virtual environment of peg-in-hole microassembly. In *Fifth World Congress on Intelligent Control and Automation (WCICA 2004)*, volume 4, pages 3210–3213, 2004.
- [161] J. Thompson and R. Fearing. Automating microassembly with ortho-tweezers and force sensing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1327–1334, 2001.
- [162] M. Tichem, D. Lang, and B. Karpuschewski. A classification scheme for quantitative analysis of micro-grip principles. *Assembly Automation*, 24:88–93, 2004.
- [163] K. Tsui, A. A. Geisberger, M. Ellis, and G. D. Skidmore. Micromachined end-effector and techniques for directed MEMS assembly. *Journal of Micromechanics and Microengineering*, 14:542–549, Apr. 2004.
- [164] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.
- [165] R. T. Vaughan, R. Gerkey, and A. Howard. On device abstractions for portable, reusable robot code. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 2121–2427, Las Vegas, Oct. 2003.
- [166] B. Vikramaditya and B. Nelson. Visually guided microassembly using optical microscopes and active vision techniques. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 3172–3177, 1997.

- [167] B. Vikramaditya and B. J. Nelson. Modeling microassembly tasks with interactive forces. *Proceedings of the IEEE International Symposium on Assembly and Task Planning*, pages 482–487, 2001.
- [168] B. Vikramaditya, B. Nelson, G. Yang, and E. Enikov. Microassembly of hybrid magnetic MEMS. *Journal of Micromechatronics*, 1(2):99–116, 2001.
- [169] B. L. Walle, M. Gauthier, and N. Chaillet. A submerged freeze microgripper for micromanipulation. *IEEE International Conference on Robotics and Automation*, pages 826–831, 2007.
- [170] B. L. Walle, M. Gauthier, and N. Chaillet. Submerged freeze gripper to manipulate micro-objects. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 784–789, Oct. 2006.
- [171] M. Wautelet. Scaling laws in the macro-, micro-, and nanoworld. *European Journal of Physics*, 22(6):601–611, 2001.
- [172] M. Weck and C. Peschke. Equipment technology for flexible and automated micro-assembly. *Microsystem Technologies*, 10(3):241–246, 2004.
- [173] P. Wunsch and G. Hirzinger. Registration of CAD-models to images by iterative inverse perspective matching. In *13th International Conference on Pattern Recognition*, volume 1, pages 78–83, 1996.
- [174] Y. Xiong and S. Shafer. Depth from focusing and defocusing. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR*, pages 68–73, 1993.
- [175] G. Yang, J. A. Gaines, and B. J. Nelson. A flexible experimental workcell for efficient and reliable wafer-level 3d microassembly. *Proceedings of the 2001 IEEE International Conference on Robotics & Automation*, pages 133–138, 2001.
- [176] G. Yang, J. A. Gaines, and B. J. Nelson. A supervisory wafer-level 3D microassembly system for hybrid MEMS fabrication. *Journal of Intelligent and Robotic Systems*, 37(1):43–68, 2003.
- [177] H. J. Yeh and J. S. Smith. Fluidic self-assembly of microstructures and its application to the integration of GaAs on Si. *Proc. IEEE Workshop on Micro Electro Mechanical Systems (MEMS)*, pages 279–284, 1994.
- [178] K. B. Yeşin. *CAD Model Based Tracking for Visually Guided Microassembly*. Ph.d. thesis, University of Minnesota, Minneapolis, Minnesota, USA, June 2003.
- [179] G. Zachmann. Rapid collision detection by dynamically aligned DOP- trees. In *Proc. of IEEE Virtual Reality Annual International Symposium VRAIS*, pages 90–97, Atlanta, Georgia, Mar. 1998.
- [180] G. Zachmann. Minimal hierarchical collision detection. In *Proc. ACM Symposium on Virtual Reality Software and Technology (VRST)*, pages 121–128, Hong Kong, China, Nov. 11–13 2002.
- [181] G. Zachmann and R. Weller. Collision detection library using OpenSG. Source Code, 2007. <http://cg.in.tu-clausthal.de>.

- [182] M. Zäh, B. Petzold, O. Anton, M. Ehrenstrasser, and J. Schilp. Telepräsenz und Teleaktion in der Mikrosystemtechnik: Systeme zur Analyse und Bewertung zu unterstützender Fertigkeiten. In *Proc of HCRS*, pages 91–96, 2002.
- [183] W. Zesch and R. S. Fearing. Aligment of microparts using force controlled pushing. In *SPIE Conference on Microrobotics and Micromanipulation*, Boston, MA, USA, 1998.
- [184] W. Zesch, M. Brunner, and A. Weber. Vacuum tool for handling microobjects with a nanorobot. In *International Conference on Robotics and Automation*, pages 1761–1766, Albuquerque, New Mexico, 1997.
- [185] H. Zhang, Y. Bellouard, E. Burdet, R. Clavel, A.-N. Poo, and D. W. Hutmacher. Shape memory alloy microgripper for robotic microassembly of tissue engineering scaffolds. *IEEE International Conference on Robotics and Automation ICRA*, 5: 4918–4924, 2004.
- [186] Q. Zhou, A. Aurelian, B. Chang, C. del Corral, and H. N. Koivo. Microassembly system with controlled environment. *Journal of Micromechatronics*, 2:227–248, 2004.

A

Robot Kinematics

This section describes the forward and inverse kinematics of the micro-assembly station V2 in more detail. First, the coordinate frames are introduced and the homogeneous transformation matrices are derived. In addition, the Jacobian and inverse Jacobian matrices are derived from the homogenous transformation matrices. The pseudo inverse of the incomplete Jacobian matrix is neglected due to the supposition that the control will take place between a point in the gripper coordinate frame and a corresponding target point on the platform. In the present configuration the Jacobian is invertible for all angles the micro-assembly station V2 can reach.

The vision system generally references to a common base coordinate frame \mathcal{V} . For the case of pose estimation and tracking this means that the homogeneous transformation matrix H_E^V , that describes the position and orientation of the local coordinate frame \mathcal{E} attached to an object, is calculated with respect to the base frame \mathcal{V} . A 2D reference object (see Figure 6.1) is used as a reference for calibration of the n cameras.

Figure A.1 shows the coordinate frames introduced according to the Denavit-Hartenberg (D-H) scheme (see Craig [38] or Spong and Vidyasagar [150] for further details).

A.1 4 DOF Base Unit

The platform of the base unit has a coordinate frame \mathcal{P} with axes X_P , Y_P and Z_P . The origin is at the center of the platform with Z_P pointing up along θ_1 . In general, frame \mathcal{P} can have an arbitrary orientation with respect to the vision frame \mathcal{V} . The present configuration shown in Figure A.1b is for the case when the two coordinate frames are aligned.

A.1.1 Forward Kinematics

The translation and rotation angles, that define the different coordinate frames for the platform according to the D-H convention, are listed in Table A.1.

The corresponding transformation matrices are

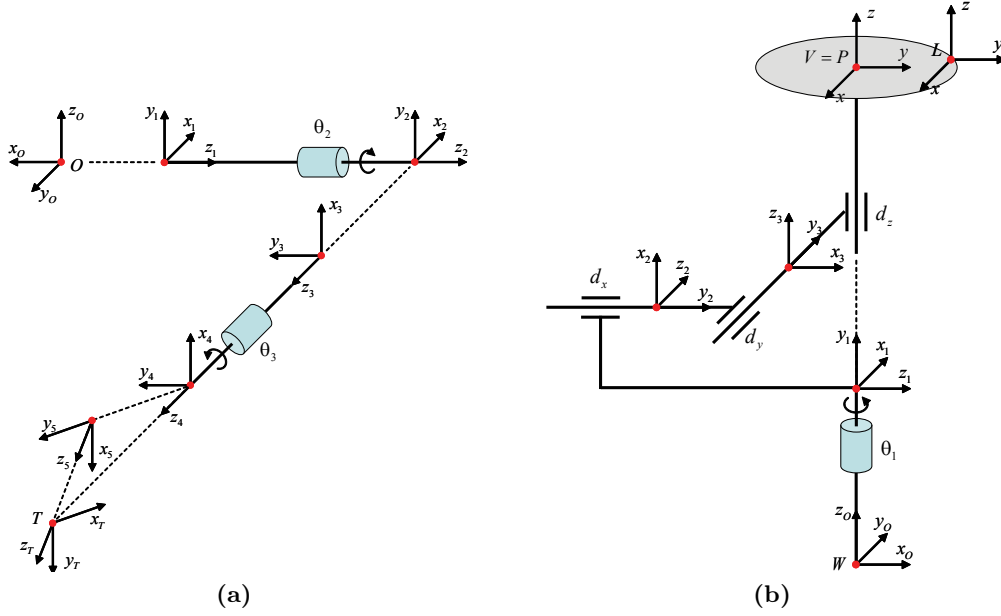


Fig. A.1: Coordinate frames for the D-H convention for the gripper unit (a) and the base unit (b).

	θ_i	d_i	a_i	α_i
1	$\theta_1 + 90^\circ$	0	0	90°
2	90°	d_x	0	90°
3	90°	d_y	0	90°
4	0	d_z	0	0

Table A.1: Denavit-Hartenberg parameters for the base unit.

$$A_1^W = \begin{pmatrix} -\sin \theta_1 & 0 & \cos \theta_1 & 0 \\ \cos \theta_1 & 0 & \sin \theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_2^1 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_x \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_3^2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_P^3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Multiplying these matrices in the appropriate order yields the forward kinematics of the system. The resulting matrix H_P^W describes the transformation from the platform coordinate frame \mathcal{P} to the frame \mathcal{W} as

$$P^W = H_P^W P^P \quad (\text{A.1.1})$$

where

$$H_P^W = A_1^W A_2^1 A_3^2 A_P^3 = \begin{pmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & d_x \cos \theta_1 - d_y \sin \theta_1 \\ \sin \theta_1 & \cos \theta_1 & 0 & d_y \cos \theta_1 + d_x \sin \theta_1 \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (\text{A.1.2})$$

A.1.2 Inverse Kinematics

In order to determine the coordinates of a point in the platform coordinate frame \mathcal{P} given its coordinates in the base coordinate frame \mathcal{W} , the inverse of the transformation matrix H_P^W is calculated. Any point in the platform coordinate frame \mathcal{P} is calculated using the transformation

$$P^P = H_W^P P^W \quad (\text{A.1.3})$$

where

$$H_W^P = (H_P^W)^{-1} = \begin{pmatrix} \cos \theta_1 & \sin \theta_1 & 0 & -d_x \\ -\sin \theta_1 & \cos \theta_1 & 0 & -d_y \\ 0 & 0 & 1 & -d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (\text{A.1.4})$$

A.1.3 Jacobian

The platform Jacobian matrix J_E^W , which describes the velocity induced on an object with coordinate frame \mathcal{E} with respect to the base frame \mathcal{W} can be formulated as

$$\begin{aligned} \begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}^W &= J_E^W \begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_z \end{pmatrix}^P \\ &= \begin{pmatrix} X_P^W & Y_P^W & Z_P^W & Z_P^W \times (T_E^W - T_P^W) \\ 0 & 0 & 0 & Z_P^W \end{pmatrix} \begin{pmatrix} \dot{x}_P \\ \dot{y}_P \\ \dot{z}_P \\ -\dot{\theta}_1 \end{pmatrix} \end{aligned} \quad (\text{A.1.5})$$

where $(T_E^W - T_P^W)$ is the position vector pointing from the origin of frame \mathcal{P} to frame \mathcal{E} described in the base frame \mathcal{W} . For an arbitrary point $p^P = (p_x, p_y, 0, 1)^T$ on the platform expressed in the base frame \mathcal{W} , the Jacobian yields

$$J_T^W = \begin{pmatrix} 1 & 0 & 0 & \cos \theta_1 (d_x + p_x) - \sin \theta_1 (d_y + p_y) \\ 0 & 1 & 0 & \sin \theta_1 (d_x + p_x) + \cos \theta_1 (d_y + p_y) \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (\text{A.1.6})$$

A.1.4 Camera Calibration Pattern Origins

The coordinate frame of the calibration pattern \mathcal{L} is not coincident with the one of the platform \mathcal{P} and the planar offset $\Delta x, \Delta y$ and orientation α depends on the size of the pattern used for camera calibration. The corresponding transformation matrix expressing a point in the calibration frame of reference \mathcal{L} in the platform frame of reference \mathcal{P} can be simply written as

$$H_L^P = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 & \Delta x \\ \sin \alpha & \cos \alpha & 0 & \Delta y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{A.1.7})$$

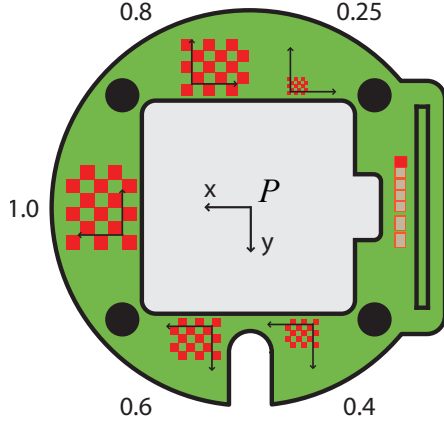
and its inverse as

$$H_P^L = \begin{pmatrix} \cos \alpha & \sin \alpha & 0 & -\Delta x \cos \alpha - \Delta y \sin \alpha \\ -\sin \alpha & \cos \alpha & 0 & -\Delta y \cos \alpha + \Delta x \sin \alpha \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (\text{A.1.8})$$

Table A.2 displays parameters $(\Delta x, \Delta y, \alpha)$ for the five calibration patterns. The pattern with $d = 0.6$ has proven to be most suitable for a primary microscope magnification of $SM = 0.75$.

A.2 2 DOF Gripper Unit

The gripper unit is separated from the platform and provides the two remaining degrees of freedom. As shown in Figure A.1a the θ_2 axis is along X^W and the θ_3 axis is along Y^W . At the end of the wrist a gripper with a TCP coordinate frame \mathcal{T} is mounted. Since the origin of \mathcal{T} is designed to be in the intersection point of all three rotational axes θ_1 , θ_2 , and θ_3 , there is no translational component involved. This alignment process is performed by calibration (see Section 3.2.5).



d	Δx	Δy	α
0.25	-3.0	-8.25	π
0.4	-4.6	8.4	0
0.6	2.6	8.5	0
0.8	4.0	-8.8	π
1.0	9.0	2.0	$-\frac{\pi}{2}$

values in mm or rad

Fig. A.2: Origins of the five calibration patterns \mathcal{L} with respect to the workbench coordinate frame \mathcal{P} . The table on the right shows the offset values $(\Delta x, \Delta y, \alpha)$ for each of the five pattern sizes d .

A.2.1 Forward Kinematics

The rotation angles that define the different coordinate frames for the gripper unit according to the D-H convention are listed in Table A.2.

	θ_i	d_i	a_i	α_i
1	0	0	0	-90°
2	θ_2	0	0	0
3	-90°	0	0	90°
4	$-\theta_3$	0	0	0
5	-90°	0	0	45°
6	-90°	0	0	0

Table A.2: Denavit-Hartenberg parameters for the gripper unit.

Note that no translations are performed to determine the coordinate frames of the gripper unit. The translations in Figure A.1a are just plotted for convenience of the reader. Translations are not necessary since the two axis of the swivel joints coincide at the point of origin. Therefore, all coordinate frames can be assumed to be fixed to this point. The corresponding transformation matrices are

$$A_1^W = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_2^1 = \begin{pmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 0 \\ \sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_3^2 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_4^3 = \begin{pmatrix} \cos \theta_3 & \sin \theta_3 & 0 & 0 \\ -\sin \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_5^4 = \begin{pmatrix} 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ -1 & 0 & 0 & 0 \\ 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_T^5 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Multiplying these matrices in the appropriate order yields the forward kinematics of the system. The resulting matrix H_T^W describes the transformation from the TCP coordinate frame \mathcal{T} to the frame \mathcal{W} as

$$P^W = H_T^W P^T \quad (\text{A.2.1})$$

where

$$H_T^W = A_1^W A_2^1 A_3^2 A_4^3 A_5^4 A_T^5 = \begin{pmatrix} -\frac{S_3}{\sqrt{2}} & -C_3 & -\frac{S_3}{\sqrt{2}} & 0 \\ \frac{C_2+C_3S_2}{\sqrt{2}} & -S_2S_3 & \frac{C_2+C_3S_2}{\sqrt{2}} & 0 \\ -\frac{C_2C_3+S_2}{\sqrt{2}} & C_2S_3 & -\frac{C_2C_3+S_2}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{A.2.2})$$

and $S_i = \sin \theta_i$ and $C_i = \cos \theta_i$.

A.2.2 Inverse Kinematics

In order to determine the coordinates of a point in the gripper coordinate frame \mathcal{T} given its coordinates in the zero coordinate frame \mathcal{W} , the inverse of the transformation matrix H_T^W is calculated. Any point in the gripper coordinate frame \mathcal{T} is calculated using the transformation

$$P^T = H_W^T P^W \quad (\text{A.2.3})$$

where

$$H_W^T = (H_T^W)^{-1} = \begin{pmatrix} -\frac{S_3}{\sqrt{2}} & \frac{C_2+C_3S_2}{\sqrt{2}} & -\frac{C_2C_3+S_2}{\sqrt{2}} & 0 \\ -C_3 & -S_2S_3 & C_2S_3 & 0 \\ -\frac{S_3}{\sqrt{2}} & \frac{C_2+C_3S_2}{\sqrt{2}} & -\frac{C_2C_3+S_2}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (\text{A.2.4})$$

A.2.3 Jacobian

The gripper Jacobian matrix J_T^W , which describes the velocity induced on an object in the coordinate frame \mathcal{T} with respect to the base frame \mathcal{W} can be formulated as

$$\begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}^W = J_T^W \begin{pmatrix} \omega_x \\ \omega_y \end{pmatrix}^T = \begin{pmatrix} X_T^W & Y_T^W & Z_T^W & Z_T^W \times T_T^W \\ 0 & 0 & 0 & Z_T^W \end{pmatrix} \begin{pmatrix} \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix}. \quad (\text{A.2.5})$$

The Jacobian matrix can be evaluated using the homogenous transformation matrix

$$H_T^W = \begin{pmatrix} -\frac{S_3}{\sqrt{2}} & -C_3 & -\frac{S_3}{\sqrt{2}} & A \\ \frac{C_2+C_3S_2}{\sqrt{2}} & -S_2S_3 & \frac{C_2+C_3S_2}{\sqrt{2}} & B \\ -\frac{C_2C_3+S_2}{\sqrt{2}} & C_2S_3 & \frac{-C_2C_3+S_2}{\sqrt{2}} & C \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{A.2.6})$$

with

$$A = \frac{1}{2} \left(\sqrt{2}(g_x + g_z)C_2 - S_2\sqrt{2}(g_x + g_z)C_3 + 2g_yS_3 \right) \quad (\text{A.2.7})$$

$$B = -g_yC_3 + \frac{(g_x + g_z)S_3}{\sqrt{2}} \quad (\text{A.2.8})$$

$$C = \frac{1}{2} \left(\sqrt{2}(-g_x + g_z)S_2 - C_2\sqrt{2}(g_x + g_z)C_3 + 2g_yS_3 \right) \quad (\text{A.2.9})$$

where g_i is the coordinate value in the gripper coordinate frame \mathcal{T} . For an arbitrary point $p^T = (p_x, p_y, 0, 1)^T$ on the end effector expressed in the base frame \mathcal{W} the Jacobian is given by

$$J_T^W = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -C_2 \\ 1 & 0 \\ 0 & S_2 \end{pmatrix}. \quad (\text{A.2.10})$$

A.3 6 DOF Manipulator

A.3.1 Forward Kinematics

From the forward and inverse kinematics of both the gripper and base unit the combined forward kinematics can be calculated. Any point p^T in the gripper coordinate frame \mathcal{T} can be expressed in the platform coordinate frame \mathcal{P} by

$$p^P = H_T^P p^T = H_W^P H_T^W p^T \quad (\text{A.3.1})$$

with

$$H_T^P = \begin{pmatrix} \frac{C_1(C_2-C_3S_2)-C_1S_3}{\sqrt{2}} & -C_1C_3 + S_1S_2S_3 \\ -\frac{C_1(C_2-C_3S_1)+S_1S_3}{\sqrt{2}} & -C_3S_1 - C_1S_2S_3 \\ -\frac{C_2C_3+S_2}{\sqrt{2}} & C_2S_3 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} -\frac{S_1(C_2+C_3S_2)+C_1S_3}{\sqrt{2}} & C_1d_x - S_1d_y \\ \frac{C_1(C_2+C_3S_2)-S_1S_3}{\sqrt{2}} & S_1d_x + C_1d_y \\ \frac{-C_2C_3+S_2}{\sqrt{2}} & d_z \\ 0 & 1 \end{pmatrix}. \quad (\text{A.3.2})$$

A.3.2 Inverse Kinematics

From the forward and inverse kinematics of both the gripper and base unit the combined forward kinematics can be calculated. Any point p^P in the gripper coordinate frame \mathcal{P} can be expressed in the platform coordinate frame \mathcal{T} by

$$p^T = H_P^T p^P = H_W^T H_P^W p^P \quad (\text{A.3.3})$$

with

$$H_P^T = \begin{pmatrix} \frac{S_1(C_2-C_3S_2)-C_1S_3}{\sqrt{2}} & -\frac{C_1(C_2-C_3S_2)+S_1S_3}{\sqrt{2}} \\ -C_1C_3 + S_1S_2S_3 & -C_3S_1 - C_1S_2S_3 \\ -\frac{S_1(C_2+C_3S_2)+C_1S_3}{\sqrt{2}} & \frac{C_1(C_2+C_3S_2)-S_1S_3}{\sqrt{2}} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} -\frac{C_2C_3+S_2}{\sqrt{2}} & \frac{S_3d_x+(C_2-C_3S_2)d_y+(C_2C_3+S_2)d_z}{\sqrt{2}} \\ C_2S_3 & C_3d_x + S_3(S_2d_y - C_2d_z) \\ \frac{-C_2C_3+S_2}{\sqrt{2}} & \frac{S_3d_x-S_2(C_3d_y+d_z)-C_2(d_y-C_3d_z)}{\sqrt{2}} \\ 0 & 1 \end{pmatrix}. \quad (\text{A.3.4})$$

A.3.3 Jacobian

The total Jacobian is a combination of both the Jacobian of the base unit and the Jacobian of the gripper unit

$$J' = (J_P^W \mid J_T^W) = \begin{pmatrix} 1 & 0 & 0 & C_1(d_x + p_x) - S_1(d_y + p_y) & 0 & 0 \\ 0 & 1 & 0 & S_1(d_x + p_x) + C_1(d_y + p_y) & 0 & 0 \\ 0 & 0 & 1 & d_z & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -C_2 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & S_2 \end{pmatrix}. \quad (\text{A.3.5})$$

Since the Jacobian matrix is regular within the joint limits the inverse can be determined by regular inversion. The inverse Jacobian can then be written as

$$J'^{-1} = \begin{pmatrix} 1 & 0 & 0 & T_2(-C_1(d_x + p_x) + S_1(d_y + p_y)) \\ 0 & 1 & 0 & -T_2(S_1(d_x + p_x) + C_1(d_y + p_y)) \\ 0 & 0 & 1 & -d_z T_2 \\ 0 & 0 & 0 & T_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{C_2} \end{pmatrix} \begin{pmatrix} 0 & -C_1(d_x + p_x) + S_1(d_y + p_y) \\ 0 & -S_1(d_x + p_x) - C_1(d_y + p_y) \\ 0 & -d_z \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{pmatrix} \quad (\text{A.3.6})$$

where $T_2 = \tan \theta_2$.

A.4 Ring Units

The rotation matrix R_L^C and translation vector ${}_{L}t_L^C$ describe position and orientation of the calibration pattern \mathcal{L} with respect to the camera \mathcal{C} . Additional transformations are required for correct referencing to the world coordinate frame \mathcal{W} as visualized in Figure A.3.

Thus, all camera transformation matrices are extended and expressed with respect to the fixed world coordinate frame \mathcal{W} using forward kinematics (see Section A.1) and the known manipulator values during calibration. The full camera transformation matrix can be expressed as

$$H_W^C = H_L^C H_P^L H_W^P \quad (\text{A.4.1})$$

where H_W^P is given by forward kinematics (Equation A.1.2), $H_P^L = (H_L^P)^{-1}$ is given by platform dimensions (Equation A.1.7), and H_L^C is the initial transformation matrix from camera calibration (see Section B.5).

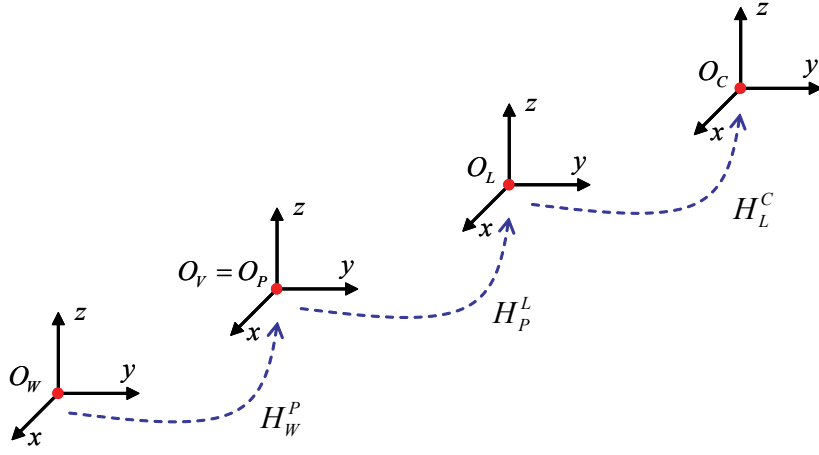


Fig. A.3: *Hierarchy of coordinate frames from camera \mathcal{C} , calibration pattern \mathcal{L} , platform \mathcal{P} , to world \mathcal{W} .*

B

Camera Calibration

B.1 The General Transformation Equation

Generally, mapping a point \mathbf{x} from an n -space to another point \mathbf{x}' can be described using a general transformation matrix H

$$\mathbf{x}' = H\mathbf{x} \quad . \quad (\text{B.1.1})$$

B.2 Camera Calibration From Known Point Correspondences

Camera calibration is usually referred to as the problem of finding a transformation equation between 3D points \mathbf{X} to 2D image points \mathbf{u} . For homogenous point coordinates this can be written as

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad \text{or} \quad \mathbf{u} = P\mathbf{X} \quad (\text{B.2.1})$$

where P is the general projection matrix defined as

$$P = K H_L^C = K R_L^C (I_3 |_L t_L^C) \quad . \quad (\text{B.2.2})$$

K is the camera intrinsic parameter matrix

$$K = \begin{pmatrix} \alpha_x & s & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{B.2.3})$$

where α_x and α_y stand for the focal length expressed in horizontal and vertical pixel dimensions, s is the skew parameter, and (u_0, v_0) are the pixel coordinates of the principal

point (intersection of the optical axis with the image plane). R_L^C is the rotation matrix from the calibration pattern coordinate frame \mathcal{L} to the camera coordinate frame \mathcal{C}

$$R_L^C = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \quad (\text{B.2.4})$$

and ${}_L t_L^C$ the vector from \mathcal{C} to \mathcal{L} expressed relative to \mathcal{L}

$${}_L t_L^C = \begin{pmatrix} {}_L t_x \\ {}_L t_y \\ {}_L t_z \end{pmatrix}. \quad (\text{B.2.5})$$

Vector \mathbf{u} is expressed in a metric unit relative to the camera coordinate frame \mathcal{C} and vector \mathbf{X} is expressed in a metric unit relative to the calibration pattern coordinate frame \mathcal{L} .

The goal of camera calibration is to find a matrix P for a given set of point correspondences $\mathbf{X}_i \rightarrow \mathbf{u}_i$. Equation B.2.1 can then be written as

$$\begin{aligned} \mathbf{u} \times P\mathbf{X} &= \mathbf{0} \\ &= \begin{pmatrix} u \\ v \\ w \end{pmatrix} \times \begin{pmatrix} \mathbf{p}_1^T \mathbf{X} \\ \mathbf{p}_2^T \mathbf{X} \\ \mathbf{p}_3^T \mathbf{X} \end{pmatrix} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} \times \begin{pmatrix} \mathbf{X}^T \mathbf{p}_1 \\ \mathbf{X}^T \mathbf{p}_2 \\ \mathbf{X}^T \mathbf{p}_3 \end{pmatrix} \\ &= \begin{pmatrix} u\mathbf{X}^T \mathbf{p}_3 - w\mathbf{X}^T \mathbf{p}_2 \\ w\mathbf{X}^T \mathbf{p}_1 - u\mathbf{X}^T \mathbf{p}_3 \\ u\mathbf{X}^T \mathbf{p}_2 - v\mathbf{X}^T \mathbf{p}_1 \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{0}_{4 \times 1}^T & -w\mathbf{X}^T & v\mathbf{X}^T \\ w\mathbf{X}^T & \mathbf{0}_{4 \times 1}^T & -u\mathbf{X}^T \\ -v\mathbf{X}^T & u\mathbf{X}^T & \mathbf{0}_{4 \times 1}^T \end{pmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix} = A\mathbf{p} \end{aligned} \quad (\text{B.2.6})$$

where $\mathbf{p}_i = (p_{i1} \ p_{i2} \ p_{i3} \ p_{i4})^T \ \forall i \in \{1, 2, 3\}$. Since the third row of matrix A in $A\mathbf{p} = \mathbf{0}$ is a linear combination of the first two rows it is usually omitted. For a minimal solution up to an arbitrary scale factor $n \geq 6$ equations are needed (every point correspondence delivers 2 equations). However, due to noise there is no exact solution \mathbf{p} for a given set of correspondences and Equation B.2.6 is extended by an error vector ε

$$\begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_j \\ \vdots \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{p}}_1 \\ \tilde{\mathbf{p}}_2 \\ \vdots \\ \tilde{\mathbf{p}}_3 \end{pmatrix} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_j \\ \vdots \end{pmatrix}. \quad (\text{B.2.7})$$

This system of equations cannot be solved directly anymore and iterative approximation methods that minimize the error vector ε have to be used. The more correspondences are fed to the system, the better the approximation of $\tilde{\mathbf{p}}$ is.

B.3 Sampson Estimate of True Image and 3D Points

According to Hartley and Zisserman [67] a first estimate for the true 3D points can be calculated after

$$\tilde{\mathbf{X}}_j = \mathbf{X}_j + \boldsymbol{\delta}_{X_j} \quad (\text{B.3.1})$$

where the error $\boldsymbol{\delta}_{X_j}$ is calculated from

$$\boldsymbol{\delta}_{X_j} = -J_j^T (J_j J_j^T)^{-1} \boldsymbol{\varepsilon}_j \quad (\text{B.3.2})$$

J_j is the Jacobian of the system $\boldsymbol{\varepsilon}_j = A_j \tilde{\mathbf{p}}_{DLT}$

$$\begin{aligned} \begin{pmatrix} \varepsilon_{jx} \\ \varepsilon_{jy} \\ \varepsilon_{jz} \end{pmatrix} &= A_j \mathbf{p} = \\ &= \begin{pmatrix} \mathbf{0}_{4 \times 1}^T & -w_j X_j & -w_j Y_j & -w_j Z_j & -w_j & v_j X_j & v_j Y_j & v_j Z_j & v_j \\ w_j X_j & w_j Y_j & w_j Z_j & w_j & \mathbf{0}_{4 \times 1}^T & -u_j X_j & -u_j Y_j & -u_j Z_j & -u_j \\ -v_j X_j & -v_j Y_j & -v_j Z_j & -v_j & u_j X_j & u_j Y_j & u_j Z_j & u_j & \mathbf{0}_{4 \times 1}^T \end{pmatrix} \\ &\quad (p_{11}, \dots, p_{14}, p_{21}, \dots, p_{24}, p_{31}, \dots, p_{34})^T \end{aligned} \quad (\text{B.3.3})$$

yielding

$$\varepsilon_{jx} = (-w_j p_{21} + v_j p_{31})X_j + (-w_j p_{22} + v_j p_{32})Y_j + (-w_j p_{23} + v_j p_{33})Z_j + (-w_j p_{24} + v_j p_{34}) \quad (\text{B.3.4})$$

$$\varepsilon_{jy} = (w_j p_{11} - u_j p_{31})X_j + (w_j p_{12} - u_j p_{32})Y_j + (w_j p_{13} - u_j p_{33})Z_j + (w_j p_{14} - u_j p_{34}) \quad (\text{B.3.5})$$

$$\varepsilon_{jz} = (-v_j p_{11} + u_j p_{21})X_j + (-v_j p_{12} + u_j p_{22})Y_j + (-v_j p_{13} + u_j p_{23})Z_j + (-v_j p_{14} + u_j p_{24}) \quad (\text{B.3.6})$$

and thus

$$J_j = \frac{\partial(\mathbf{y}_j = A_j \mathbf{p})}{\partial \mathbf{X}_j} = \begin{pmatrix} \frac{\partial y_{1j}}{\partial X_j} & \frac{\partial y_{1j}}{\partial Y_j} & \frac{\partial y_{1j}}{\partial Z_j} & \frac{\partial y_{1j}}{\partial W_j} \\ \frac{\partial y_{2j}}{\partial X_j} & \frac{\partial y_{2j}}{\partial Y_j} & \frac{\partial y_{2j}}{\partial Z_j} & \frac{\partial y_{2j}}{\partial W_j} \\ \frac{\partial y_{3j}}{\partial X_j} & \frac{\partial y_{3j}}{\partial Y_j} & \frac{\partial y_{3j}}{\partial Z_j} & \frac{\partial y_{3j}}{\partial W_j} \end{pmatrix} = \begin{pmatrix} -w_j p_{21} + v_j p_{31} & -w_j p_{22} + v_j p_{32} & -w_j p_{23} + v_j p_{33} & -w_j p_{24} + v_j p_{34} \\ w_j p_{11} - u_j p_{31} & w_j p_{12} - u_j p_{32} & w_j p_{13} - u_j p_{33} & w_j p_{14} - u_j p_{34} \\ -v_j p_{11} + u_j p_{21} & -v_j p_{12} + u_j p_{22} & -v_j p_{13} + u_j p_{23} & -v_j p_{14} + u_j p_{24} \end{pmatrix} \quad (\text{B.3.7})$$

The estimate for true image points can then be calculated using the initial projection matrix estimate calculated with the direct linear transformation (DLT) algorithm

$$\tilde{\mathbf{u}}_j = \tilde{P}_{DLT} \tilde{\mathbf{X}}_j . \quad (\text{B.3.8})$$

B.4 Maximum Likelihood Minimization

In order to find good estimates for $\tilde{\mathbf{X}}_j$, $\tilde{\mathbf{u}}_j$, and \tilde{P} they can be used as parameters of a probability density function. Assuming a probability model (i.e., the probability distribution of the errors of the image and 3D points) $Pr(\tilde{\mathbf{X}}_j, \tilde{\mathbf{u}}_j, \tilde{P})$ one can find $\tilde{\mathbf{X}}_j$, $\tilde{\mathbf{u}}_j$, and \tilde{P} such that the likelihood of the chosen model becomes maximal.

The errors for camera calibration are the distances $\{d_{X_j}, d_{u_j}\}$ between the measured points $\{\mathbf{X}_j, \mathbf{u}_j\}$ and the true points $\{\tilde{\mathbf{X}}_j, \tilde{\mathbf{u}}_j\}$. Assuming that they are Gaussian distributed the probability density function yields

$$Pr(d_{X_j}) = \left(\frac{1}{\sqrt{2\pi}\sigma_{X_j}} \right)^2 \exp \left(\frac{-d(\mathbf{X}_j, \tilde{\mathbf{X}}_j)^2}{2\sigma_{X_j}^2} \right) \quad (\text{B.4.1})$$

and

$$Pr(d_{u_j}) = \left(\frac{1}{\sqrt{2\pi}\sigma_{u_j}} \right)^2 \exp \left(\frac{-d(\mathbf{u}_j, \tilde{\mathbf{u}}_j)^2}{2\sigma_{u_j}^2} \right) \quad (\text{B.4.2})$$

$$= \left(\frac{1}{\sqrt{2\pi}\sigma_{u_j}} \right)^2 \exp \left(\frac{-d(\mathbf{u}_j, \tilde{P}\tilde{\mathbf{X}}_j)^2}{2\sigma_{u_j}^2} \right) . \quad (\text{B.4.3})$$

These two equations can be combined to

$$\begin{aligned} Pr(d_{X_j}, d_{u_j}) &= \\ &\left(\frac{1}{\sqrt{2\pi}\sigma} \right)^2 \exp \left(\frac{-(\lambda_u \sum_{j=1}^N d(\mathbf{u}_j, \tilde{P}\tilde{\mathbf{X}}_j)^2 + \lambda_X \sum_{j=1}^N d(\mathbf{X}_j, \tilde{\mathbf{X}}_j)^2)}{2\sigma^2} \right) \\ &= A \exp \left(\frac{-(\lambda_u \sum_{j=1}^N d(\mathbf{u}_j, \tilde{P}\tilde{\mathbf{X}}_j)^2 + \lambda_X \sum_{j=1}^N d(\mathbf{X}_j, \tilde{\mathbf{X}}_j)^2)}{B} \right) \end{aligned} \quad (\text{B.4.4})$$

with

$$A = \left(\frac{1}{\sqrt{2\pi}\sigma} \right)^2 \quad (\text{B.4.5})$$

$$B = 2\sigma^2 . \quad (\text{B.4.6})$$

This term is the probability that the erroneous measurements $\{\mathbf{X}_j, \mathbf{u}_j\}$ have been obtained given that the true measurements and the true projection matrix have been $\{\tilde{\mathbf{X}}_j, \tilde{\mathbf{u}}_j, \tilde{P}\}$. Estimating the most likely values for $\{\tilde{\mathbf{X}}_j, \tilde{\mathbf{u}}_j, \tilde{P}\}$ will result in a maximum likelihood of Pr minimizing the errors $\{d_{X_j}, d_{u_j}\}$ and thus delivering an optimal solution for \tilde{P} . Finding the maximum of the original likelihood function is equivalent to finding the maximum of the log-likelihood

$$\begin{aligned} \log(Pr(d_{X_j}, d_{u_j})) &= \\ \log(A) + \frac{-\left(\lambda_u \sum_{j=1}^N d(\mathbf{u}_j, \tilde{P}\tilde{\mathbf{X}}_j)^2 + \lambda_X \sum_{j=1}^N d(\mathbf{X}_j, \tilde{\mathbf{X}}_j)^2\right)}{B} \end{aligned} \quad (\text{B.4.7})$$

A and B are constants and can be neglected for the maximization problem, so $\log(A) = 0$, $B = 1$. Therefore the whole problem reduces to

$$\max \left\{ -(\lambda_u \sum_{j=1}^N d(\mathbf{u}_j, \tilde{P}\tilde{\mathbf{X}}_j)^2 + \lambda_X \sum_{j=1}^N d(\mathbf{X}_j, \tilde{\mathbf{X}}_j)^2) | \tilde{P}, \tilde{\mathbf{X}} \right\} . \quad (\text{B.4.8})$$

B.5 Decomposition of the Projection Matrix

At this point it is important to note that the projection matrix P is calculated with respect to the origin of the calibration pattern \mathcal{L} as clearly visible in Equation B.2.2. Since frame \mathcal{L} is situated on the workbench (see Figure A.2) and thus moves during assembly, a new projection matrix P_W with respect to the fixed base coordinate frame \mathcal{W} is formulated.

Once a good approximation of the true projection matrix P has been found it is usually required to decompose it to its components K , R_L^C and ${}_L t_L^C$ (see Equation B.2.2). Since K is an upper triangular matrix the QR decomposition algorithm can be used. However, this method has proven to yield degenerate values that do not match the real configuration.

A better approach is to calculate matrix K as shown in Equation B.2.3. While most parameters are a priori known from camera data sheets, the focal length f is technically an infinite number for an optical system approximating an orthographic projection model. The assumption of a large number has worked fine in this case. Given matrix K , the transformation from the pattern origin \mathcal{L} to the camera frame of reference \mathcal{C} can be evaluated by rewriting Equation B.2.2

$$H_L^C = K^{-1} P . \quad (\text{B.5.1})$$

Next, the total transformation matrix from world to camera frame of reference, given by equation A.4.1, can be computed. Finally, the total projection matrix mapping a point in the world coordinate frame \mathcal{W} to the image plane \mathcal{I} is given by

$$P_W = K H_W^C . \quad (\text{B.5.2})$$

B.6 Calibration Procedure

1. Find an initial estimate for \tilde{P} using the linear normalized direct linear transformation (DLT) algorithm.
 - a) Normalize all measured image and 3D points $\{\mathbf{X}_j, \mathbf{u}_j\}$ in the point clouds $\{\mathbf{X}, \mathbf{u}\}$. To do so, find transformations T_u and T_X such that the centroids of the clouds $\{\mathbf{X}', \mathbf{u}'\}$ holding the normalized points $\{\mathbf{X}'_j = T_X \mathbf{X}_j, \mathbf{u}'_j = T_u \mathbf{u}_j\}$ are $\{\mathbf{0}_{1 \times 3}, \mathbf{0}_{1 \times 2}\}$ and that the normalized average offset of the points from their new centroids is $\sqrt{2}$. This ensures that the DLT becomes invariant to the chosen coordinate frame for the points $\{\mathbf{X}_j, \mathbf{u}_j\}$.
 - b) Form the matrix A' (see Equation B.2.6) using the normalized point correspondences. The DLT solution $\tilde{\mathbf{p}}'$ for $A' \mathbf{p}' = \mathbf{0}$ with $\|\mathbf{p}'\| = 1$ is the column of the right-singular matrix V of the singular value decomposition $A'_{m \times n} = U_{m \times m} D V_{n \times n}$ corresponding to the smallest singular value of A' .
 - c) From $\tilde{\mathbf{p}}'$ assemble the initial estimate \tilde{P}' .
2. Find initial estimates $\{\tilde{\mathbf{X}}_j, \tilde{\mathbf{u}}_j\}$ for the true image and 3D points $\{\bar{\mathbf{X}}_j, \bar{\mathbf{u}}_j\}$ using the Sampson approximation method shown in B.3.

3. Minimize the geometric error using the maximum likelihood strategy explained in [B.4](#).
 - a) Choose a “reasonable” value for μ and σ .
 - b) Use maximum likelihood estimation to find better estimates for $\{\tilde{\mathbf{X}}_j, \tilde{\mathbf{u}}_j, \tilde{\mathbf{P}}\}$ minimizing the geometrical or the Mahalanobis distance between measured and “true” points.
4. Once a terminating condition is fulfilled denormalize the normalized projection matrix such that $\tilde{\mathbf{P}} = T_u^{-1} \tilde{\mathbf{P}}' T_X$.
5. Decompose the estimated projection matrix into its components K , R_L^C and ${}_L t_L^C$.

C

Computing the Rigid Body Motion

A rigid body is defined as an object whose distance between any two points remains constant at all times. Rigid body motion is the description of movement of such an object in space and can be written for homogenous coordinates as

$$P' = \begin{pmatrix} R_{3 \times 3} & \mathbf{t}_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{pmatrix} P = MP \quad (\text{C.0.1})$$

where R is a rotation matrix and \mathbf{t} a translation vector. A common representation of three-dimensional rotation is known as Z-Y-X Euler angles with subsequent rotation of α around Z , β around Y , and γ around X . The rotation matrix is

$$\begin{aligned} R_{zyx} &= R_z(\alpha) R_y(\beta) R_x(\gamma) \\ &= \begin{pmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{pmatrix} \end{aligned} \quad (\text{C.0.2})$$

$$= \begin{pmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{pmatrix} \quad (\text{C.0.3})$$

with $c\alpha = \cos(\alpha)$ and $s\alpha = \sin(\alpha)$, etc.

A group is a finite or infinite set of elements together with a binary operation (known as the group operation) that together satisfy the following four fundamental axioms of *closure*, *associativity*, *identity element* and *inverse element*. The following two groups are of interest here:

The special orthogonal group $SO(3)$ is a subgroup of the orthogonal group $O(n)$ and the rotation group on \mathbb{R} defined as

$$SO(3) := \{R \in \mathbb{R}^{3 \times 3} : RR^T = I, \det(R) = +1\} .$$

The special Euclidean group $SE(3)$ is the group of rigid transformations on \mathbb{R}^3 defined as the set of mappings $g : \mathbb{R}^3 \mapsto \mathbb{R}^3$ of the form $g(x) = Rx + p$, where $R \in SO(3)$ and $p \in \mathbb{R}^3$. An element of $SE(3)$ is written as $(p, R) \in SE(3)$. $SE(3)$ can be identified with the space of 4×4 matrices of the form

$$g = \begin{pmatrix} R & p \\ 0 & 1 \end{pmatrix} ,$$

where $R \in SO(3)$ and $p \in \mathbb{R}^3$.

$SO(3)$ and $SE(3)$ are *Lie groups* with a corresponding *Lie algebra*

$$so(3) = \{A \in \mathbb{R}^{3 \times 3} | A = -A^T\} \quad (\text{C.0.4})$$

$$se(3) = \{(v, w) | v \in \mathbb{R}^3, w \in so(3)\} . \quad (\text{C.0.5})$$

and are connected through an exponential function. Assuming the rotation of a point p around an axis ω its velocity can be written as

$$\dot{q}(t) = \omega \times q(t) = \hat{\omega}(t) \quad (\text{C.0.6})$$

where $\hat{\omega}$ is defined as

$$\hat{\omega} := \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} . \quad (\text{C.0.7})$$

This is essentially the tangents of the Euler matrices at identity

$$\hat{\omega} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_1 \\ 0 & \omega_1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & \omega_2 \\ 0 & 0 & 0 \\ -\omega_2 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & -\omega_3 & 0 \\ \omega_3 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (\text{C.0.8})$$

$$= \frac{\partial R_x}{\partial \gamma}(0) + \frac{\partial R_y}{\partial \beta}(0) + \frac{\partial R_z}{\partial \alpha}(0) . \quad (\text{C.0.9})$$

The time-invariant linear differential equation C.0.6 can be integrated to

$$q(t) = e^{\hat{\omega}t} q(0) \quad (\text{C.0.10})$$

with $q(0)$ being the initial condition at $t = 0$ and the matrix exponential

$$e^{\hat{\omega}t} = \sum_{k=0}^{\infty} \frac{(\hat{\omega}t)^k}{k!} . \quad (\text{C.0.11})$$

Again, for a point p rotated around axis ω at θ units of time, the (skew-symmetric) rotation matrix can be formulated as

$$R(\omega, \theta) = e^{\hat{\omega}\theta} . \quad (\text{C.0.12})$$

The equivalent of $R(\omega, \theta)$ for the group of 3D rigid motions $SE(3)$ is known as a twist $\xi = (v, \omega) \in se(3)$ and yields

$$\hat{\xi} = \begin{pmatrix} \hat{\omega} & v \\ 0_{1 \times 3} & 0 \end{pmatrix} . \quad (\text{C.0.13})$$

The rigid motion $e^{\theta\hat{\xi}}$ corresponds to a screw motion and the group action can be evaluated using the formula of Rodrigues

$$e^{\hat{\xi}\theta} = \begin{pmatrix} e^{\theta\hat{\omega}} & (I - e^{\theta\hat{\omega}})(\omega \times v) + \omega\omega^T v\theta \\ 0_{1 \times 3} & 1 \end{pmatrix} \quad (\text{C.0.14})$$

with

$$e^{\theta\hat{\omega}} = I + \hat{\omega} \sin(\theta) + \hat{\omega}^2 (1 - \cos(\theta)) . \quad (\text{C.0.15})$$

Variable θ is also referred to as the moment of a screw motion.

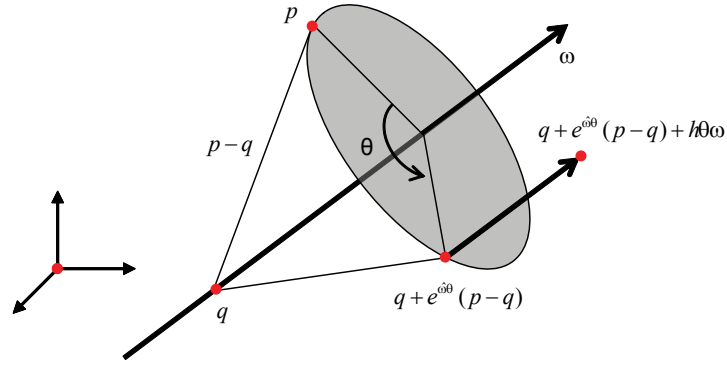


Fig. C.1: Illustration of a screw motion by rotation of moment θ around axis l and translation by $h\theta$ parallel to l .

D

Computer Vision

D.1 Contours and Convex Hulls

A planar polygon is convex if it contains all the line segments connecting any pair of its points. A regular pentagon is convex (Figure D.1a left) whereas an indented pentagon is not (Figure D.1a right).

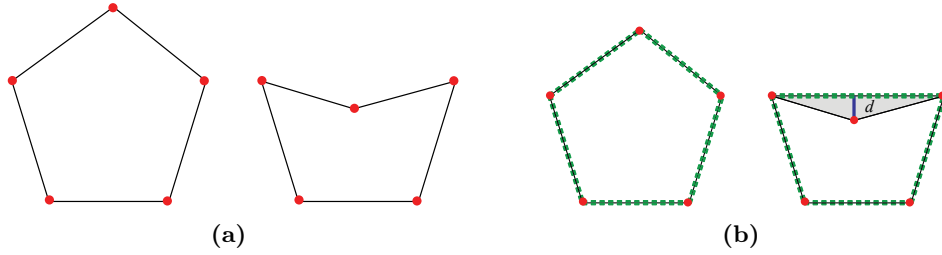


Fig. D.1: *Convex and concave polygons (a) and their corresponding convex hulls (b).*

The convex hull of a set of points S in n dimensions is the intersection of all convex sets containing S . For N points p_1, \dots, p_N , the convex hull C_h is then given by the expression

$$C_h \equiv \left\{ \sum_{j=1}^n \lambda_j p_j : \lambda_j \geq 0 \forall j \text{ and } \sum_{j=1}^n \lambda_j = 1 \right\} . \quad (\text{D.1.1})$$

The single convexity defect $d_{c,i}$ denotes the maximum distance from a segment of the convex hull to the contour within that segment as illustrated in Figure D.2. The total convexity defect d_c is the sum of all single defects $d_{c,i}$. The relative convexity defect \bar{d}_c is the total convexity defect d_c divided by the number of segments

$$\bar{d}_c = \frac{1}{m} \sum_{i=1}^m d_{c,i} \quad (\text{D.1.2})$$

where m is the total number of elements in the convex hull.

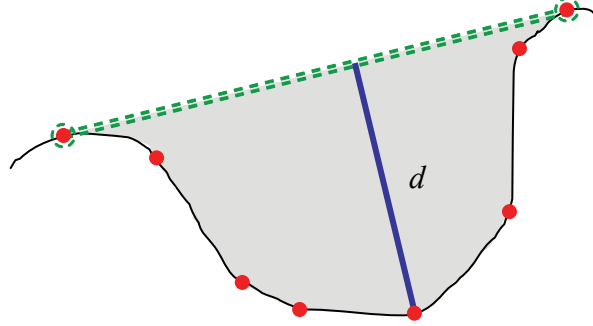


Fig. D.2: *Convex hull and convexity defect d .*

D.2 OpenGL® Matrices

In order to display geometry in OpenGL® the projection matrix has to be decomposed into two special forms, namely the modelview matrix M_{mv} , which is equivalent to the full camera transformation matrix H_W^C (see Equation A.4.1),

$$M_{mv} = \begin{pmatrix} R_W^C & ct_W^C \\ \mathbf{0} & 1 \end{pmatrix}$$

and the projection matrix M_{proj} , that is related to the camera intrinsics matrix K (see Equation B.2.3),

$$M_{proj} = \begin{pmatrix} \frac{k_x f}{u_0} & 0 & 0 & 0 \\ 0 & -\frac{k_y f}{v_0} & 0 & 0 \\ 0 & 0 & \frac{c_f + c_n}{c_f - c_n} & -\frac{2c_f c_n}{c_f - c_n} \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

where f is the focal length, k_x and k_y are the reciprocals of the pixel sizes in x and y direction, u_0 and v_0 are the coordinates of the principal point, and c_n and c_f are the near and far clipping planes, respectively (see Figure D.3).

D.3 Fields of View of EDMUND VZM-300i Microscopes

The micro-assembly station V2 uses up to three microscopes that are tilted at $\beta = 45^\circ$ with respect to the horizontal plane. This results in a larger field of view as visualized in Figure D.4. Since there is no angular movement perpendicular to the rotation axis of

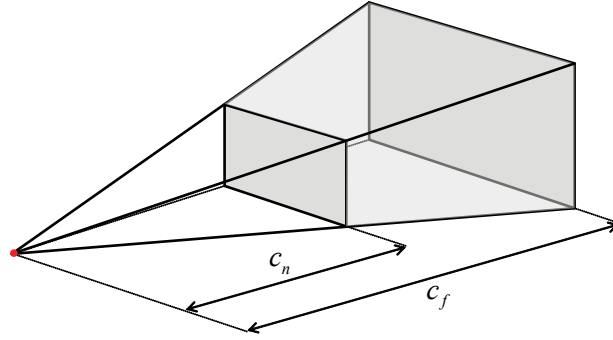


Fig. D.3: OpenGL[®] view volume with near and far clipping planes.

β , the field of view is only affected in one direction. Due to the fact that in the present configuration each camera is additionally rotated counterclockwise around its optical axis by 90° , a change in β affects the horizontal field of view.

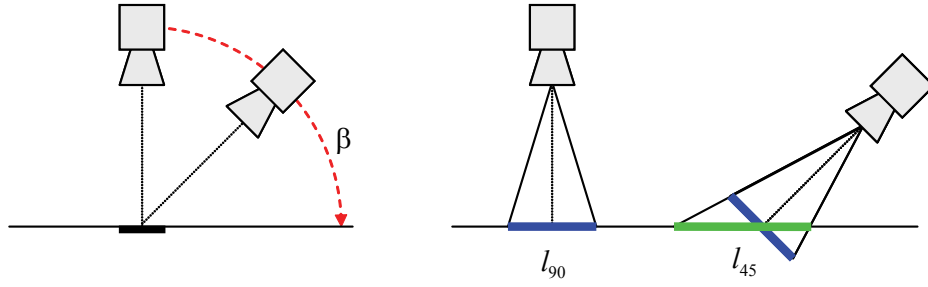


Fig. D.4: Influence of camera tilt β on the field of view. On the right the present case for an angle of $\beta = 45^\circ$ is shown.

Table D.1 shows the fields of view and image space resolutions for $\beta = 90^\circ$ and $\beta = 45^\circ$. Particularly the values of the resolutions are useful for determining lengths in camera images.

primary magnification	0.75x	1.0x	2.0x	3.0x
field of view $\beta = 90^\circ$	8.0	7.3	4.7	2.0
	6.0	5.475	3.525	1.5
resolution $\beta = 90^\circ$	82.0	89.86	139.57	328.0
	81.67	89.5	139.0	326.67
field of view $\beta = 45^\circ$	11.31	10.32	6.65	2.83
	6.0	5.475	3.525	1.5
resolution $\beta = 45^\circ$	58.0	63.57	98.65	231.80
	81.67	89.5	139.0	326.67

fields of view in mm and resolution in pixels/mm

Table D.1: *Fields of view and resolutions of an EDMUND VZM-300i microscope for different viewing angles β .*

E

Geometric Transformations

E.1 General Transformations

A linear transformation in Euclidean space is a mapping $L : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ such as for a given point $p = (x, y, z)$

$$L(x, y, z) = (ax + by + cz + d, ex + fy + gz + h, ix + jy + kz + l) \quad (\text{E.1.1})$$

with $a, b, c, d, e, f, g, h, i, j, k, l \in \mathbb{R}$. Equation [E.1.1](#) can also be written in matrix form

$$L(x, y, z) = (x, y, z) \begin{pmatrix} a & e & i \\ b & f & j \\ c & g & k \end{pmatrix} + \begin{pmatrix} d \\ h \\ l \end{pmatrix} . \quad (\text{E.1.2})$$

A translation $T(h, k, l)$ can be applied by simply adding the translation vector (h, k, l) to point p

$$T(h, k, l)(x, y, z) = (x + h, y + k, z + l) = (x, y, z) + (h, k, l) . \quad (\text{E.1.3})$$

A scaling with respect to the origin is a transformation of the form

$$S(s_x, s_y, s_z)(x, y, z) = (s_x x, s_y y, s_z z) = (x, y, z) \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{pmatrix} . \quad (\text{E.1.4})$$

A rotation is a transformation of the form

$$R(\alpha, \beta, \gamma)(x, y, z) = R_z(\gamma)R_y(\beta)R_x(\alpha)(x, y, z) . \quad (\text{E.1.5})$$

This notation is known as the Z-Y-X Euler angle representation and the individual rotation matrices are

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix} \quad (\text{E.1.6})$$

$$R_y = \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix} \quad (\text{E.1.7})$$

$$R_z = \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{E.1.8})$$

E.2 Homogenous Coordinates

Homogenous coordinates utilize a mathematical trick to embed three-dimensional coordinates and transformations into a four-dimensional matrix format. As a result, inversions or combinations of linear transformations are simplified to inversion or multiplication of the corresponding matrices. Homogenous coordinates also make it possible to define perspective transformations. A single three-dimensional vector

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (\text{E.2.1})$$

can be expressed in homogenous coordinates as

$$\begin{pmatrix} w \cdot x \\ w \cdot y \\ w \cdot z \\ w \end{pmatrix}. \quad (\text{E.2.2})$$

The relation between homogenous and Euclidean coordinates is given by

$$\begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} \Leftrightarrow \begin{pmatrix} x/w \\ y/w \\ z/w \end{pmatrix}. \quad (\text{E.2.3})$$

E.3 Euclidean Transformations

The most common transformation is the Euclidean transformation, which can be either a translation, a rotation, or a reflection. Generally, a point x is transformed to x' by

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (\text{E.3.1})$$

where r_{ij} describes a rotation matrix and t_k a translation vector.

Euclidean transformations preserve length and angle measure. Moreover, the shape of a geometric object will not change. That is, lines transform to lines, planes transform to planes, circles transform to circles, and ellipsoids transform to ellipsoid. Only the position and orientation of the object will change.

E.4 Affine Transformations

Affine transformations are generalizations of Euclidean transformations. Under affine transformations, lines transform to lines, but circles become ellipses. Additionally, length and angle are not preserved. The transform is given by

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \quad (\text{E.4.1})$$

E.5 Projective Transformations

Projective transformations are the most general “linear” transformations and need to use homogeneous coordinates. Given a point in space in homogeneous coordinates (x, y, z, w) and its image under a projective transform (x', y', z', w') , a projective transform has the following form

$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}. \quad (\text{E.5.1})$$

F

Geometric Entities

F.1 Plücker Lines

Plücker coordinates are a way to assign six homogenous coordinates to each line in projective 3-space. A line l in 3D Euclidean space is determined by two distinct points expressed as position vectors $\mathbf{x} = (x_1, x_2, x_3)$ and $\mathbf{y} = (y_1, y_2, y_3)$. A Plücker line is defined as

$$\mathbf{d} = \mathbf{y} - \mathbf{x} \tag{F.1.1}$$

$$\mathbf{m} = \mathbf{x} \times \mathbf{y} \tag{F.1.2}$$

where \mathbf{d} is the *direction* and \mathbf{m} the *moment* of the line. Up to a scale factor, \mathbf{d} and \mathbf{m} do not depend on the specific \mathbf{x} and \mathbf{y} .

F.2 Hessian Normal Form

The Hessian normal form is a specific way to describe planes and is obtained from the general equation of a plane

$$ax + by + cz + d = 0 \tag{F.2.1}$$

by defining the components of the unit normal vector $\mathbf{n} = (n_x, n_y, n_z)$

$$n_x = \frac{a}{\sqrt{a^2 + b^2 + c^2}} \tag{F.2.2}$$

$$n_y = \frac{b}{\sqrt{a^2 + b^2 + c^2}} \tag{F.2.3}$$

$$n_z = \frac{c}{\sqrt{a^2 + b^2 + c^2}} \tag{F.2.4}$$

and the constant

$$p = \frac{d}{\sqrt{a^2 + b^2 + c^2}} . \quad (\text{F.2.5})$$

Then, the Hessian normal form is defined as

$$\mathbf{n} \cdot \mathbf{x} = -p \quad (\text{F.2.6})$$

with p being the distance of the plane from the origin. Additional to that, the sign of p determines on which side of the plane the origin is located. For $p > 0$ the origin is in the half-space determined by the direction of \mathbf{n} and for $p < 0$ it is in the other half-space. The distance from an arbitrary point x_0 to a plane is given by

$$d = \mathbf{n} \cdot \mathbf{x}_0 + p \quad (\text{F.2.7})$$

where d is positive if it lies in the half-space determined by the direction of \mathbf{n} , and negative otherwise.

Software Modules

G.1 Overview

This chapter briefly shows the C++ libraries used within the scope of this project. Figure G.1 shows the presence of individual libraries on each computing center.

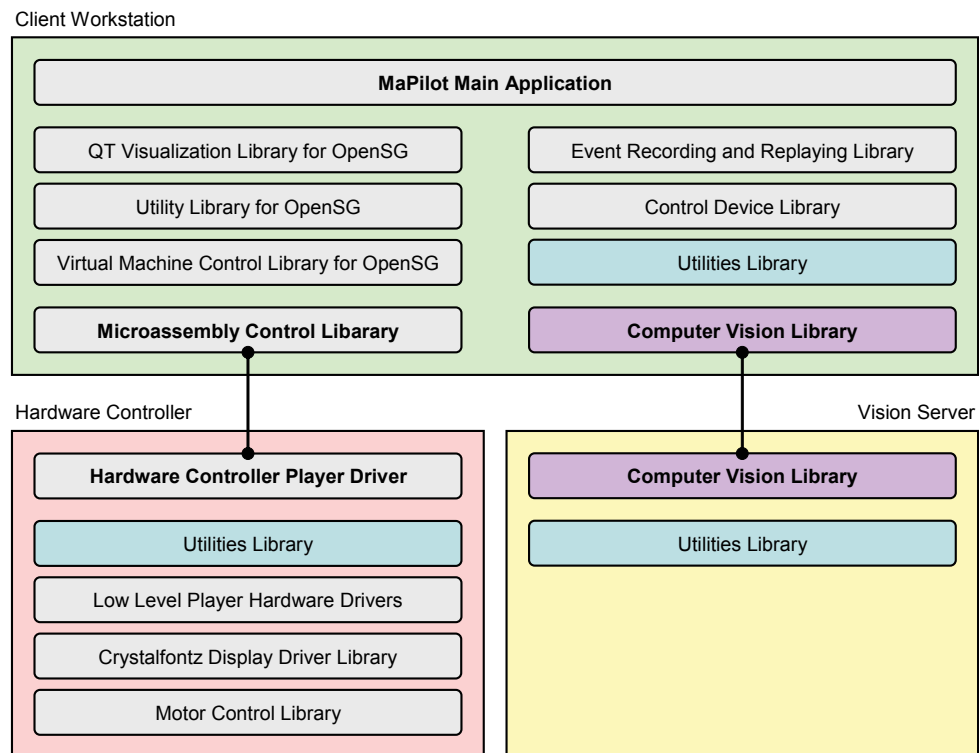


Fig. G.1: Location and interaction of specially developed libraries.

Since some libraries have been specially developed whereas others were taken off the shelf, the following list is split accordingly. Symbol ► designates binary files and ▷ shared library (*.so) files.

G.2 Specially Developed Libraries

Main Application This is the actual main application with the graphical (GUI) and tangible user interface (TUI). It communicates with both the hardware controller and the vision server and its architecture is based on a dynamic plugin loading mechanism for great flexibility.

► mapilot

▷ libCollDetWidget, libControlPanel, libDataViewer, libEventRecordingWidget, libSceneTree, libCameraControl, libImageProcessing, libVRConfigWidget, libVRView

Microassembly Control Library Provides a client interface to the hardware controller. Abstracts motion and other commands and handles communication over the network.

▷ libmacontrol

Qt Visualization Library for OpenSG Allows visualizing OpenSG scenegraphs within the Qt framework.

▷ libAV_SGWidgetOSG, libCameraViewWidgetOSG, libRenderWidgetOSG, libSGTreeOSG, libSGWidgetOSG, libVEWidgetOSG

Utility Library for OpenSG A set of utilities for managing and manipulating OpenSG scenegraphs.

▷ libosgutils

Virtual Machine Control Library for OpenSG Virtualization of a kinematic chain given a scenegraph and joint definition files.

▷ libvrcontrol

Event Recording and Replaying Library Recording and replaying of motion events.

▷ libeventrecording

Hardware Control Library High-level interface to a large set of hardware elements, such as motors.

▷ libAinM34, libAinM36, libAoutM33, libCamLightI03, libDioM66, libEncoderM72, libLog, libMotorI01, libMotorI04

Low-Level Player Hardware Drivers Accessing individual hardware elements through the Player interface.

▷ libi01_position1d, libi01_position3d, libi03_camLightAirOpaque, libi04_position1d, libi04_position3d, libm33_aout, libm34_ain, libm36_ain, libm66_dio, libm72_enc

Hardware Controller Player driver Main driver providing a Player interface for high level commands.

▷ libmadriver

LCD display driver Marvin LCD display driver module.

- ▶ crystalMenu
- ▷ libcrystalfontz, libcrystalfontz++, libnetstats, libvalve, libinitManipulator

Control Device Library Provides base classes for haptic input devices and implements some of them.

- ▷ libcontroldevice

Microassembly Utilities Library Large utility library providing and extending capabilities of the boost library, of math and computer vision algorithms, etc. ▷ libmautils

Computer Vision Library Image acquisition, image processing, camera calibration, etc.

- ▶ cameraCalibration
- ▷ libmacv

G.3 Third Party Libraries

The following libraries are third party products and are required for compiling the modules mentioned in the previous section. Some of those packages have been adapted, mostly so that they can be used with the standard automake process flow.

Collision Detection Library Provides collision detection on an OpenSG scenegraph. Copyright by Prof. G. Zachmann [179–181]. Adapted so that it can be used with automake.

- ▷ libcolldet

TinyXML Simple XML parsing library. Adapted so that it can be used with automake.

- ▷ libtinyxml

Matrix Template Library MTL Advanced vector and matrix manipulation templates for C++.

OpenCV Open source computer vision library.

Qt4 State of the art windowing toolkit used for any graphical user interface in the system.

Boost Libraries This project uses boost threads (synchronization primitives and threads), boost signals (very powerful signal/slots implementation), and boost python (communication between C++ and python programs)

OpenSG Portable scenegraph system to create realtime graphics programs.

libdc1394 IEEE 1394 (Firewire) Support.

libraw1394 IEEE 1394 (Firewire) Support.

libXtst X window system client interface.

OpenHaptics Support for haptic input devices.

Player Network client/server solution for robot control.

MDIS Low-level drivers for MEN bus systems.

G.4 Dimensions

In order to get a feeling about the dimensions of the code project, the total physical source lines of code (SLOC) for C and C++ has been evaluated and listed in Table G.1. Note that the physical count is higher than the logical one, since it also includes non-statement lines (but no comments).

library	line count
mapilot	18'875
libmacontrol	2'630
libqvrvis	4'131
libosgutils	6'487
libvrcontrol	1'457
libeventrecording	440
libio	15'347
libplayer	2'370
libmadriv	2'597
crystalMenu	3'154
libcontroldevice	1'462
libmautils	6'428
libmacv	15'760
Total	81'138
generated using D. A. Wheeler's SLOCCount	

Table G.1: Total physical source lines of code (SLOC) for selected libraries.

Curriculum Vitae

Personal Data

Name	Martin Probst
Date of Birth	December 27 th , 1977
Hometown	Basel-Stadt, Reigoldswil (BL)
Nationality	Swiss

Education

Swiss Federal Institute of Technology Zurich (ETH) Institute of Robotics and Intelligent Systems <i>Doctor of Science (Ph.D.)</i>	2003 – 2008
Whittle Laboratory Cambridge University, United Kingdom <i>Diploma Thesis</i>	2002
Swiss Federal Institute of Technology Zurich (ETH) <i>Diploma in Mechanical Engineering (Dipl. Masch.-Ing. ETH)</i>	1997 – 2002
Wimbledon School of English London, United Kingdom <i>Cambridge Advanced Certificate (CAE)</i>	1997
Gymnasium Kohlenberg Basel, Switzerland <i>A-Levels (Matura) type B (languages)</i>	1988 – 1996

Work Experience

SR Technics, Zurich Airport Aircraft Engine Service Center <i>Implementation of a concept study</i>	2002 – 2003
Turbomachinery Laboratory (ETH), TU Berlin and EC Lyon <i>Participation in an International Gas Turbine Project</i>	2001 – 2002
COR Infexpert AG Frenkendorf, Basel <i>Software engineering and application development</i>	1997 – 2001
SR Technics, Zurich Airport Aircraft Engine Service Center <i>Internship</i>	2000 – 2001
Institute for Mechanical Systems Swiss Federal Institute of Technology Zurich (ETH) <i>Software engineering and application development</i>	1997 – 2000
Sulzer Turbo Zurich <i>Internship in mechanical workshop</i>	1996

Publications

1. M. Flückiger, Z. Nagy, M. Probst, O. Ergeneman, S. Pané, B. J. Nelson. A Microfabricated and Microassembled Wireless Resonator. Submitted to *Sensors and Actuators A: Physical*, November 2008.
2. M. Probst, C. Hürzeler, B. J. Nelson. A Microassembly System for the Flexible Assembly of Hybrid Robotic MEMS Devices. *International Journal of Optomechatronics*, Accepted.
3. M. Probst, M. Flückiger, S. Pané, O. Ergeneman, Z. Nagy, B. J. Nelson. Manufacturing of a Hybrid Ultrasonic Transducer Using an Advanced Microassembly System. *IEEE Transactions on Industrial Electronics*, Accepted.
4. Z. Nagy, M. Flückiger, M. Probst, B. J. Nelson. A Wireless Acoustic Emitter for Passive Localization in Liquids. *International Conference on Robotics and Automation*, Kobe, Japan, May 2009.
5. F. Beyeler, M. Probst, B. J. Nelson. A Microassembly System with Microfabricated Endeffectors for Automated Assembly Tasks. Workshop on Robotic Microassembly of 3D Hybrid MEMS, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2007.
6. M. Probst, C. Hürzeler, R. Borer, B. J. Nelson. Virtual Reality for Microassembly. *International Symposium on Optomechatronic Technologies*, Lausanne, Switzerland, October 2007.

7. M. Probst, R. Borer, B. J. Nelson. Microassembly Processes and Microassembled Devices. Keynote at the *Conference on Multi-Material Micro Manufacture*, Borovets, Bulgaria, October 2007.
8. M. Probst, R. Borer, B. J. Nelson. A Microassembly System for Manufacturing Hybrid MEMS. *12th IFToMM World Congress*, Besançon, June 2007.
9. M. Probst, K. Vollmers, B. E. Kratochvil, B. J. Nelson. Design of an Advanced Microassembly System for the Automated Assembly of Bio-Microrobots. *Proc. 5th International Workshop on Microfactories*, Besançon, October 2006.
10. S. A. Stoeter, M. Probst, M. A. Iranzo. Improving Tracking Precision for Microassembly. *Proc. of the IEEE International Conference on Robotics and Automation*, Orlando, Florida, May 2006.

- image feature extraction 82
- image processing 52
- initial pose estimation 21, 75
- L**
- linear refinement 90, 91
- localization 21, 75
- M**
- manipulator 28, 149
 - forward kinematics 149
 - inverse kinematics 150
 - Jacobian 150
- maPilot 53
- maximum likelihood minimization 156
- micro-factory 12
- microassembly 7
- microassembly station V2 25
 - calibration 57
 - initialization 57
 - mechanical performance 43
- microphysics 124
- microrobot
 - extraction 84
- microrobotics
 - bio-microrobotics 2
 - foundations 1
- model based pose estimation 75
- N**
- networking 51
- O**
- OpenGL matrices 166
- P**
- parallel assembly 13
 - deterministic 13
 - self-assembly 14
 - stochastic 14
- perceptual grouping 125
- Plücker Lines 173
- pneumatics 42
- point-line constraint 77
- point-plane constraint 78
- pose error 79
- pose estimation 75, 125
 - algorithm 76
 - combining results 92
- projective transformation 171
- R**
- random sampling 90, 91
- rigid body motion 161
- ring unit
 - kinematics 151
- S**
- Sampson estimate 155
- scenegraph 65
- search space 89
- second hand 120
- serial assembly 8
- software 48
 - client workstation 53
 - graphical user interface 54
 - hardware controller 49
 - plugins 55
 - tangible user interface 53
 - vision server 50
- synchronization 71
- T**
- task planning 71
- tele-operation 8
- tracking 51, 75
- U**
- ultrasound transceiver assembly 112
- V**
- virtual reality 19, 61
- vision unit 37
- visualization 125
- VR control center 66
- W**
- workbench 34

