

Two-dimensional graphics on personal workstations

Doctoral Thesis

Author(s):

Kohen, Eliyezer

Publication date:

1988

Permanent link:

<https://doi.org/10.3929/ethz-a-000510430>

Rights / license:

[In Copyright - Non-Commercial Use Permitted](#)

Two-Dimensional Graphics on Personal Workstations

**A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH
(ETH Zürich)**

**for the degree of
Doctor of Technical Sciences**

**presented by
Eliyezer Kohen, Diplom-Ingenieur
Rhein.-Westf. Technische Hochschule Aachen, W. Germany
born July 18, 1955
citizen of Turkey**

**accepted on the recommendation of
Prof. N. Wirth, examiner
Prof. J. Gutknecht, co-examiner**

Abstract

The project reported here comprises three distinct parts:

- A system for middle resolution font design
- Algorithms for some problems in two-dimensional computer graphics
- Module and data structures for a new kind of graphics editors.

The ETH Font Design System has been developed with the purpose of offering professional type designers tools for the production of middle and low resolution fonts of high aesthetic quality. The system starts with a curve-fitting program that extracts the outline information in the form of curves and control-points from scanned artwork. An outline editor provides a type designer with the facilities to design such curves from scratch, to correct existing outlines, and to specify characteristic horizontal and vertical lines which are later used for formatting text pieces. This meta description of the typeface serves as the basis for generating raster images in various sizes. A raster editor offers the tools to correct the errors introduced to the bitmap fonts by the inadequacies of the raster generator. The ETH Font Design System follows a graphic method. It emphasizes interactivity and easy usage by a typographer. As a consequence the raster generator has to fulfil high requirements.

Several algorithms have been developed in the course of this project. We analyze the problems of curve-tracing, region-filling, continuous dragging and raster generation. In each of these problems, we first classify the existing algorithms and then we present our own contribution.

The Rubi project aims at employing some of these algorithms in a graphics editor for the design of technical illustrations. Rubi has been structured so that a user can add new object classes, new attribute values, and new commands without having access to the source code. We realize the extensibility in object classes by installing a class handler and using it through messages sent to the objects. Similar upcall techniques can be used by the individual class handlers to implement the visual output for the various attributes. Two measures provide the extensibility in commands: The ability to add a new method to an existing class handler and the concept of interpreters. The latter is a new way of user interaction: Every object can be employed by a user to invoke a command. The implementation of the kernel of Rubi aims at dealing with overlapping 2-dimensional objects like textured areas, images, and text strings in a most efficient way.

Kurzfassung

Das hier beschriebene Projekt besteht aus drei getrennten Teilen:

- Einem System für die Entwicklung von Schriften im mittleren Auflösungsbereich
- Algorithmen für einige Aufgaben in 2-dimensionaler Computergraphik
- Modul- und Datenstrukturen für eine neue Art von Graphikeditoren.

Das ETH Font Design System soll Kalligraphen Werkzeuge bieten, um Schriften hoher typographischer Qualität für die mittlere und niedere Auflösung zu entwerfen. Ein Curvefitting-Programm kann bei gegebenen Bildern von Buchstaben deren Umrisslinie erkennen und diese in Form von Kurven und Stützpunkten ausdrücken. Mit Hilfe eines Kontour-Editors kann der Künstler solche Kurven von Grund auf neu entwerfen, bestehende Kurven korrigieren und spezifische horizontale und vertikale Begrenzungslinien angeben. Diese Meta-Beschreibung einer Schrift dient als Grundlage für die Generierung von Rasterbildern in verschiedenen Grössen. Mit Hilfe eines Raster-Editors kann der Künstler die Fehler korrigieren, die durch die Unzulänglichkeit des Rastergenerators entstanden sind. Unser System verfolgt eine graphische Methodik. Sein Schwerpunkt liegt auf der Interaktivität und der einfachen Benutzung durch einen Typographen. Als Folge dessen muss das Rasterprogramm besonders hohe Anforderungen erfüllen.

Das Projekt umfasst mehrere Algorithmen. Wir untersuchen die Schwierigkeiten des Kurven-Nachziehens, des Füllens einer gegebenen Fläche, des kontinuierlichen Verschiebens von Figuren und der Rastergenerierung. In jedem Thema stellen wir die bestehenden Algorithmen und unseren Beitrag vor.

Das Rubi-Projekt verwendet einige dieser Algorithmen im Rahmen eines Graphik-Editors für technische Zeichnungen. Rubi ist erweiterbar: Ohne Zugang zum Quellcode kann der Benutzer dem Editor neue Objektklassen, Attributwerte und Befehle hinzufügen. Die Erweiterbarkeit für Objektklassen erfolgt mit Hilfe von "Handlern", die zuerst installiert und dann durch Meldungen an die einzelnen Objekte verwendet werden. Ähnlich wird auch die Erweiterbarkeit für Attributwerte verwirklicht. Zwei Massnahmen gewährleisten die Erweiterbarkeit für Befehle: Die Möglichkeit, einem bestehenden Handler eine neue Methode hinzuzufügen und das Konzept der Interpreter. Das letztere ist eine neue Art der Benutzerinteraktion: Jedes graphische Objekt kann vom Benutzer zur Auslösung eines Befehls verwendet werden. Der Kern von Rubi zielt darauf hin, 2-dimensionale überlappende Objekte wie schraffierte Flächen, Bilder und Textstücke möglichst optimal zu behandeln.