



Doctoral Thesis

## **Skizzieren in der Vorlesung Ein kombinatorischer Ansatz zur Erkennung handgezeichneter Skizzen**

**Author(s):**

Pomm, Konrad

**Publication Date:**

2006

**Permanent Link:**

<https://doi.org/10.3929/ethz-a-005270672> →

**Rights / License:**

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

This page was generated automatically upon download from the [ETH Zurich Research Collection](#). For more information please consult the [Terms of use](#).

Diss. ETH No. 16705, 2006

# **Skizzieren in der Vorlesung – Ein kombinatorischer Ansatz zur Erkennung handgezeichneter Skizzen**

Abhandlung zur Erlangung des Titels  
Doktor der Technischen Wissenschaften  
der Eidgenössischen Technischen Hochschule Zürich

vorgelegt von Konrad Pomm  
Assessor des Lehramts (Math., Kath. Theol., Inf.)  
Universität Tübingen  
geboren am 8.9.1969 in Stuttgart, Deutschland

Angenommen auf Antrag von  
Prof. Dr. Peter Widmayer, ETH Zürich, Referent  
Prof. Dr. Juraj Hromkovič, ETH Zürich, Korreferent  
Dr. Alex Hall, ETH Zürich, Korreferent

# Zusammenfassung

Mit der Verbreitung des Tablet PCs haben stiftbasierte Systeme an Bedeutung gewonnen. In der Tat lassen sich breit gefächerte Anwendungsmöglichkeiten finden, die den besonderen Nutzen von Stift-Interaktion eindrucksvoll belegen, beispielsweise in den Bereichen Design oder E-Teaching. Ein wichtiger Aspekt ist dabei die Erkennung der handschriftlich eingegebenen Informationen, um natürlich und fließend mit der verwendeten Applikation interagieren zu können. Sketch Recognition beschäftigt sich mit der Erkennung handschriftlicher Zeichnungen, sogenannten *Sketchszenen*. Ziel ist die Identifikation der vom Benutzer gezeichneten Symbole, welche durch *Strokes* in der Sketchszene repräsentiert sind.

In dieser Arbeit schlagen wir erstmals ein kombinatorisches Modell für Sketch Recognition vor. In unserem Modell fassen wir ein Symbol gleichsam als eine Schablone auf, die über eine Menge von Strokes gelegt werden kann, um festzustellen, wie gut die Schablone mit diesen Strokes zusammenpasst. Dazu benötigen die Strokes Platzhalter innerhalb der Schablone. Diese Platzhalter sind die elementaren Bausteine eines Symbols, repräsentieren einen bestimmten geometrischen Typ (Liniensegment, Kreis, Kreisbogen, etc.) und heißen *Komponenten*. Wie diese Komponenten miteinander in Beziehung stehen wird mit Hilfe von *Constraints* beschrieben. Constraints machen z.B. eine Aussage darüber, dass zwei Komponenten rechtwinklig zueinander stehen oder dass sie gleich lang sind. Der Grundgedanke des kombinatorischen Modells besteht darin, dass jeder Constraint eine Bewertungsfunktion besitzt. Diese Bewertung berechnet für Strokes, die den Komponenten zugewiesen werden, einen Konfidenzwert. Die Zusammenfassung aller Konfidenzen von Constraints eines Symbols ist die *Symbolkonfidenz*. Das zugehörige Optimierungsproblem, das unter allen möglichen Zuordnungen von Strokes zu Komponenten die Zuordnung mit der maximalen Symbolkonfidenz sucht, bezeichnen wir mit MAXSYMBCONF. Die Interpretation einer Sketchszene baut auf der Symbolkonfidenz auf. Wir betrachten die *Szenenkonfidenz* einer Sketchszene als die Zusammenfassung von maximalen Symbolkonfidenzen von möglichen Symbolen. Das Problem MAXSCENECONF sucht unter allen Interpretationen von Strokes eine solche, bei der die zugeordneten Symbole insgesamt zu einer maximalen Szenenkonfidenz beitragen.

Wir analysieren die Komplexität beider Optimierungsprobleme und stellen Algorithmen zu deren Lösung vor. Wir zeigen, dass MAXSYMBCONF ein NP-hartes und i.a. nicht approximierbares Problem ist. Für die praktische Bestimmung von MAXSYMBCONF haben wir einen Branch-and-Bound Algorithmus entwickelt, der mit einer Reihe von Heuristiken beschleunigt werden kann. Weiter zeigen wir, dass auch MAXSCENECONF NP-hart ist – unabhängig von MAXSYMBCONF. Für die Praxis schlagen wir daher eine leichte Einschränkung der Zeichenfreiheit vor: *Symbole dürfen nur aus aufeinanderfolgend gezeichneten Strokes bestehen*. Ein Symbol muss demnach erst komplett gezeichnet worden sein, bevor der Benutzer anfängt, ein weiteres Symbol zu zeichnen. Die zu einem bestimmten Symbol gehörenden Strokes sollen dagegen in beliebiger Reihenfolge gezeichnet werden können. Für das korrespondierende Problem MAXSCENECONF<sup>consec</sup> geben wir einen Algorithmus an, der polynomiell ist, falls die entsprechenden Symbolkonfidenzen gegeben oder in polynomieller Zeit berechenbar sind.

Wir haben das vorgestellte Sketch Recognition Modell und die entsprechenden Algorithmen in *SketchWork* implementiert. SketchWork ist ein Framework, das die Umsetzung von Sketch Recognition in stiftbasierten Applikationen unterstützt.

Die wichtigsten Beiträge der vorliegenden Arbeit sind die folgenden. (1) Die Modellierung von Sketch Recognition als ein kombinatorisches Optimierungsproblem macht den Vergleich und die Diskussion über verschiedene Lösungsansätze erst möglich. Alle uns bekannten Ansätze sind heuristischer Natur, deren Qualität sehr schwer zu bewerten ist, da kein klares Ziel definiert wird. (2) Für die bei der Modellierung sich natürlicherweise ergebenden Teilprobleme erörtern wir genau, wo in Abhängigkeit der Stellenzahl von Constraints und der Anzahl von Komponenten eines Symbols die Grenzen zwischen polynomieller Berechenbarkeit und NP-Härte liegen. (3) Wir präsentieren Algorithmen und Heuristiken zu deren Beschleunigung. Wir schlagen vor, eine einfache Anforderung an den Zeichenstil zu stellen, wonach Sketchszenen effizient interpretiert werden können. (4) Unser Ansatz steht in einem einsatzfähigen System zur Verfügung. Die mächtige Beschreibungssprache erlaubt auf einfache Weise, Symbole für neue Anwendungsbereiche zu definieren oder bestehende Bereiche zu erweitern. Wir stellen Symbole für die Bereiche *Graphen*, *Geometrie* und *Buttons* zur Verfügung, die sich z.B. für den Einsatz in einer Vorlesung eignen. Die experimentelle Auswertung gibt Aufschluss über die Leistungsfähigkeit unseres Ansatzes.

---

## Abstract

Pen-based systems have become more and more important with the dissemination of the Tablet PC. Indeed, one can find a wide variety of tasks for demonstrating the particular benefits of pen-interaction, e.g., in the field of design or e-teaching. One important aspect is the recognition of the user's handwritten input to support natural interaction with the underlying software application. Sketch recognition deals with the interpretation of hand-drawn sketches, so-called *sketch scenes* which consist of *strokes* representing the pen input. The goal of sketch recognition is to identify user-drawn symbols.

In this thesis we propose the first combinatorial model for sketch recognition. In our model we see a symbol as a kind of template which can be laid over a set of strokes to verify how well the template and the strokes fit together. Therefore, the template needs to provide placeholders for the strokes. These placeholders are the elementary building blocks of a symbol. They represent a certain geometric type (line segment, circle, arc, etc.) and are called *components*. How these components are related to each other can be described via *constraints*. Constraints enforce that, e.g., two components are perpendicular to each other or that they have the same length. One of the basic ideas of the combinatorial model is that every constraint has an associated evaluation function. This function computes a confidence value for the strokes which are assigned to components of a symbol. The combination of all confidence values of the constraints in a symbol is called *symbol confidence*. We call the related optimization problem  $\text{MAXSYMBCONF}$ , where the aim is to find the maximum symbol confidence among all possible assignments of strokes to components. For the interpretation of an entire sketch scene  $\text{MAXSYMBCONF}$  appears as a subproblem. We consider the *scene confidence* of a sketch scene as the combination of maximum symbol confidences of possible symbols. Among all the interpretations of strokes the problem  $\text{MAXSCENECONF}$  searches for an interpretation such that the assigned symbols contribute to an overall maximum scene confidence.

We analyze the complexity of the two optimization problems and present algorithms for their solution. We show that  $\text{MAXSYMBCONF}$  is an NP-hard and in general not approximable problem, unless  $P = NP$ . To determine  $\text{MAXSYMBCONF}$  in practice we develop a branch-and-bound algorithm and several heuristics to increase its performance in practice.

Furthermore, we show that  $\text{MAXSCENECONF}$  is NP-hard – independent of  $\text{MAXSYMBCONF}$ . To deal with the interpretation of sketch scenes in practice we propose a modest restriction of the user’s drawing freedom: *Symbols may only consist of consecutively drawn strokes*. Hence, a symbol must have been drawn completely before the user starts to draw another symbol. In contrast, the strokes which belong to a certain symbol may be drawn in any order. For the corresponding problem  $\text{MAXSCENECONF}^{\text{consec}}$  we present an algorithm that is polynomial, if the required symbol confidences are given or can be computed in polynomial time.

We have implemented the proposed sketch recognition model and the related algorithms in *SketchWork*. SketchWork is a framework that supports the realization of sketch recognition in pen-based applications.

The main contributions of this thesis are the following. (1) The modeling of sketch recognition as a combinatorial optimization problem now allows to compare and discuss different approaches and also solutions to specific instances. All recent sketch recognition approaches that we are aware of are of heuristic nature. This makes it difficult to evaluate their quality as the goal is not clearly defined. (2) Two subproblems arise naturally in the proposed model. For these subproblems we investigate the borderline between polynomial computability and NP-hardness depending on the number of components in the symbols and the number of arguments of the constraints. (3) We present algorithms and heuristics. Furthermore, we propose a simple restriction of drawing freedom, in which case sketch scenes can be interpreted efficiently. (4) Our approach is implemented in a deployable system. The easy to use and powerful symbol description language allows the realization of sketch recognition applications in a variety of domains. We provide different examples of sketch domains which can be used directly, e.g., in a lecture to support teaching. Our experimental analysis demonstrates the performance of our approach.