Ein Freihand Modellierungstool für Surface™

Dennis Kilan

22. Juni 2011

Hochschule für Angewandte
Wissenschaften Hamburg

Inhaltsverzeichnis

Motivation

- Modellierung von Systemen wichtiger Schritt
- Variante 1: Zeichnen auf Papier
  - + Intuitiv, Einfach,Kollaborativ und Schnell
  - - keine Möglichkeit zu Speichern oder Ändern
- Variante 2: Zeichnen mit CASE-Tools
  - + Änderbar, Speicherbar,Verteilbar
  - - Oft Komplex, Kontraintuitiv, wenig Support für kollaboratives Arbeiten

Motivation

- Modellierung von Systemen wichtiger Schritt
- Variante 1: Zeichnen auf Papier
  - + Intuitiv, Einfach,Kollaborativ und Schnell
  - - keine Möglichkeit zu Speichern oder Ändern
- Variante 2: Zeichnen mit CASE-Tools
  - + Änderbar, Speicherbar,Verteilbar
  - - Oft Komplex, Kontraintuitiv, wenig Support für kollaboratives Arbeiten

Motivation

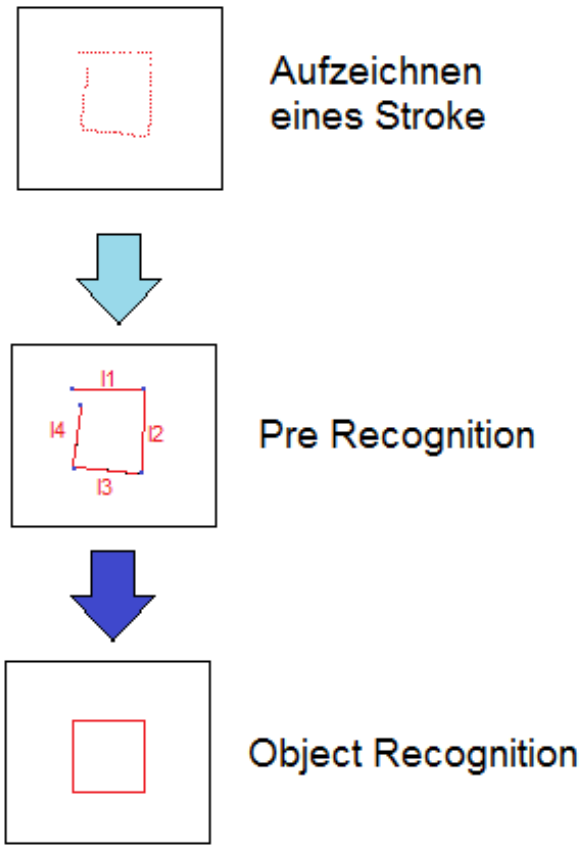
- Modellierung von Systemen wichtiger Schritt
- Variante 1: Zeichnen auf Papier
  - + Intuitiv, Einfach,Kollaborativ und Schnell
  - - keine Möglichkeit zu Speichern oder Ändern
- Variante 2: Zeichnen mit CASE-Tools
  - + Änderbar, Speicherbar,Verteilbar
  - - Oft Komplex, Kontraintuitiv, wenig Support für kollaboratives Arbeiten

Vision

- Zeichnen von Modellen über Touch Input auf dem Microsoft Surface™
- Erkennen der gezeichneten Elemente und Umwandlung in Modellelemente
- Export der gezeichneten Elemente in CASE Tools

Related Work

Sketch Recognition Architektur



1. Der Stroke wird über die Position des Eingabegerätegerätes in bestimmten Zeitintervallen aufgezeichnet
2. Der Prerecognizer erkennt aus diesen Punkten Ecken und primitive Formen wie z.B. Linien oder Kreise
3. Die Object Recognition setzt aus den primitiven Formen komplexe Formen zusammen, wie z.B. ein Rechteck

Sketch Based Interfaces[3]

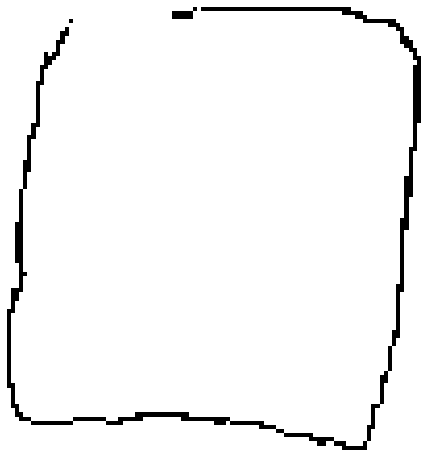
Vorstellung

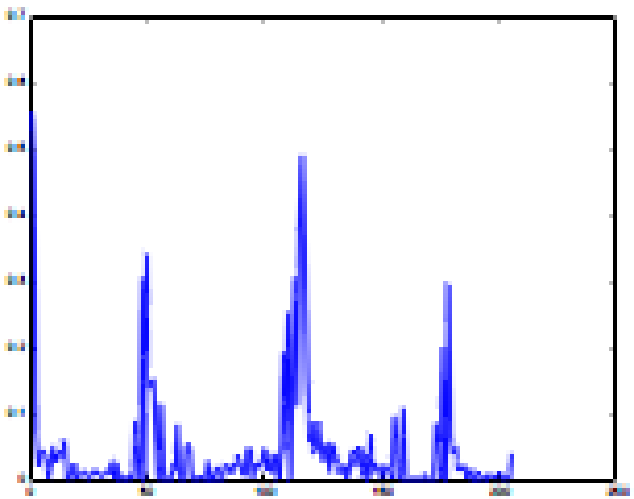
- 2001 von Sezgin, Stahovich (MIT) und Davis(CMU) entwickelt
- Ziel: Formerkennung unabhängig davon, wie sie gezeichnet wurde
- **Approximation** Versucht Linien und Kurven zu erkennen
- **Beautification** Glättet die Formen
- **Basic Recognition** Interpretiert die Zeichnung

Sketch Based Interfaces

Approximation(1)

- Im ersten Schritt: Erkennen von Ecken
- Eine Ecke weist eine hohe Krümmung auf
- Problem: Noise verursacht viele False Positives

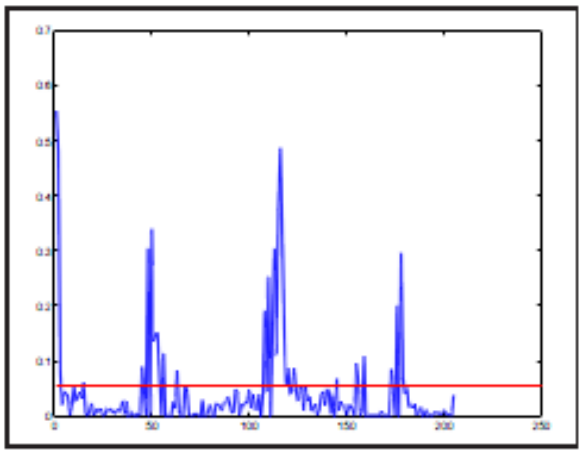


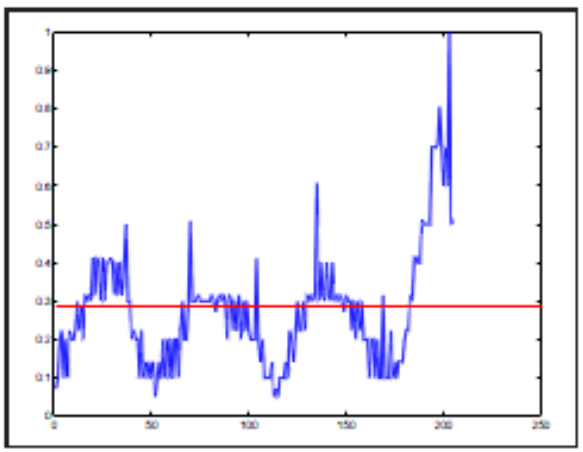


Sketch Based Interfaces

Approximation(2)

- Lösung: Average Based Filtering mit variablem Treshold(Mean)
- Absicherung mit Geschwindigkeitsdaten (lokale Minima)
- Das Finden von Ecken erfolgt über Hybrid Fit beider Daten
- Bester Fit: Wenigste Ecken





Sketch Based Interfaces

Erkennen von Linien und Kurven

- **Linienerkennung** Abweichung der Datenpunkte von einer idealisierten Linie zwischen 2 Ecken
- Falls quadratische Abweichung unter Treshold: Linie
- **Kurven** Verhältnis von Bogenlänge zu euklidischer Distanz nahe 1 bei Linien, bei Kurven wesentlich höher
- Approximation der Kurve über Bézier Kurve mit 2 Kontrollpunkten
- Bewertung der Kurve über Diskretisierung und stückweiser Zerlegung in lineare Stücke

A Domain-Independent System for Sketch Recognition (2003)[5]

Vorstellung

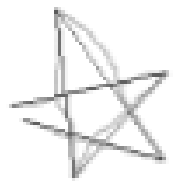
- 2003 von Bu Yi und Shije Cai an der Nanjing University in China entwickelt
- Nutzt ebenfalls Krümmungsdaten zum Finden von Ecken
- Es ist hier auch egal wie die Form gezeichnet wird
- Erkennen von Ecken und das Umwandeln in primitive Objekte in einem Schritt
- 2 Phasen: Imprecise Stroke Recognition(Erkennen von primitiven Shapes) und Post Process (Säubern der Zeichnung und Object Recognition)

A Domain-Independent System for Sketch Recognition

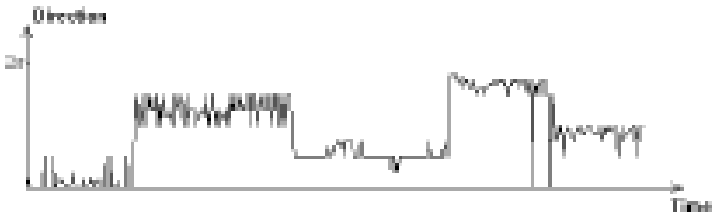
Imprecise Stroke Recognition

- Nutzt die Richtungs- und Krümmungsdaten eines Stroke, sowie die Feature Area zu einem Referenzobjekt
- Hohe Krümmung, Starker Richtungswechsel und die Feature Area bedeutet auch hier mögliche Ecke
- Verzichtet auf Geschwindigkeitsdaten, da diese zu noisy sind und nicht zu den visuellen Features gehören

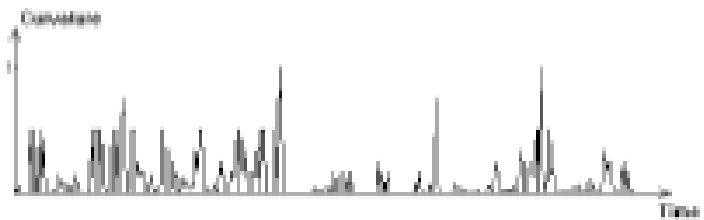
(a) Input stroke



(b) Approximation result



(c) Direction graph

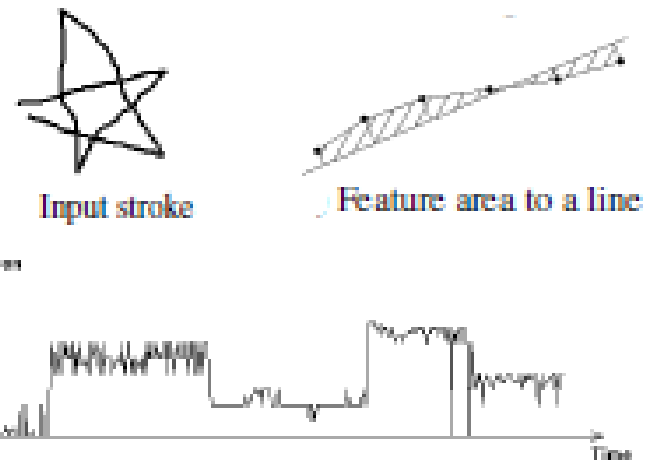


(d) Curvature graph

A Domain-Independent System for Sketch Recognition

Erkennen von Linien

- Direction Graph kann durch horizontale Linie genähert werden
- Echte Datenpunkte können durch Linie genähert werden
- Falls Feature Area und Fläche der echten Linie ca. gleich groß sind, ist eine Linie gefunden



A Domain-Independent System for Sketch Recognition

Erkennen von Kurven

- **Kreiserkennung**: Krümmungsgraph zu noisy
- Aber Richtungsgraph zeigt eine Kurve mit einer Steigung von $(-)2\pi/n$
- Angepasster Kreis: Mittelpunkt des Kreises ist Mittelpunkt der Bounding Box und der Radius ist mittlerer Radius
- **Bogenerkennung**: Ähnlich wie Kreis nur mit weniger Steigung
- Ambiguität zwischen Bogen mit wenig Krümmung und Linien

ShortStraw: A Simple and Effective Corner Finder for Polylines [4]

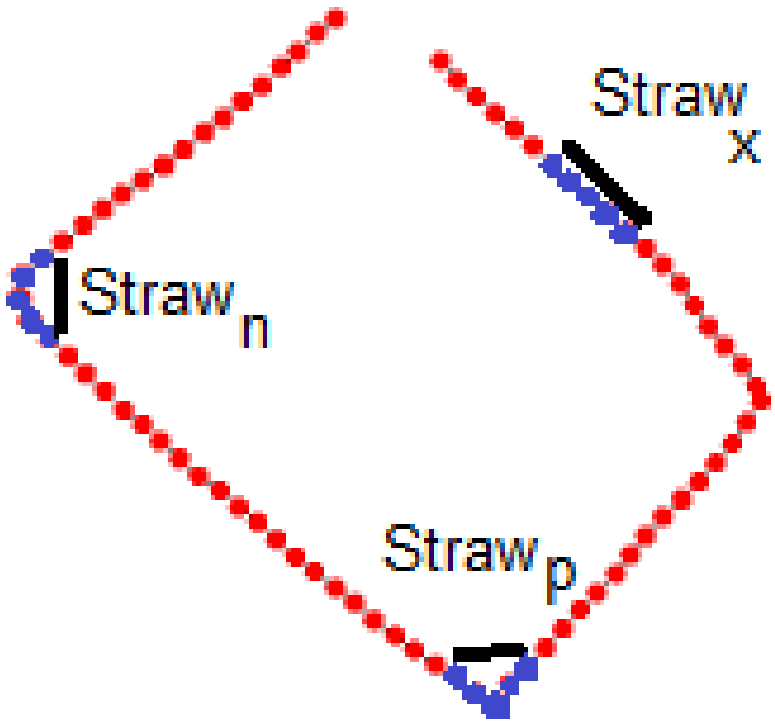
Einführung

- 2008 von Wolin an der Texas A&M University entwickelt
- Einfaches Finden von Ecken
- Ermittlung von Krümmungsdaten eines Stroke ungenau und kompliziert
- Idee: Aufteilen des Strokes in gleichmäßig verteilte Punkte

ShortStraw: A Simple and Effective Corner Finder for Polylines [4]

Einführung

- Ecken erkennt man durch die Länge von Straws
- Kürzerer Straw bedeutet Ecke
- High Level False Positive- und Negative Filterungen
- Jedoch keine Erkennung von Kurven =>viele False Positives



Object Recognition

LADDER [2]

Einführung

- 2007 von Tracy Hammond und Randall Davis am MIT entwickelt
- Domänenunabhängige Sprache zur Beschreibung von Shapes
- Beschreibt nicht nur das Aussehen eines Shapes sondern auch das Verhalten
- Aus der Beschreibung werden Shape Recognizer, Editing Recognizer und Shape Exhibitors erzeugt

LADDER

Beschreibung

```
(define shape Arrow
  (comment "An arrow with an open head.")
  (components
    (Line shaft)
    (Line head1)
    (Line head2))
  (constraints
    (coincident shaft.p1 head1.p1)
    (coincident shaft.p1 head2.p1)
    (equalLength head1 head2)
    (acuteMeet head1 shaft)
    (acuteMeet  shaft  head2))
  (aliases
    (Point head shaft.p2)
    (Point tail shaft.p1) )
  (editing
    ((trigger (holdDrag head))
      (action (rubberBand this tail head))
    ((trigger (holdDrag tail))
      (action (rubberBand this head tail))
    ((trigger (holdDrag this))
      (action (move this)))
  (display
    (originalStrokes shaft)
    (cleanedStrokes head1 head2)
    (color red))
)
```

LADDER

- Shapes können auch hierarchisch definiert werden
- Shapes können gruppiert werden
- Domänenspezifische Shapes werden über JESS Regelsystem erkannt
- Problem: Schlecht für kurvenreiche und irreguläre Shapes
- Problem2: Viele kleine Shapes und unerkannte Primitive bremsen LADDER aus

SketchML [1]

Vorstellung

- 2009 von Avolo,Del Buono,Gianforme,Paolozzi (in Rom) und Wang (Saarland Universität) entwickelt
- Framework zum Erstellen und Erkennen von Shapes bestimmter Domänen
- Nutzt SVG/XML Dateien zum Beschreiben von Shapes
- 2 Teile: Library Editor Enviroment(LEE) und Sketch Recognition Enviroment(SRE)

```
<Individual>
  -----
  <Orientation item="Line-1" degree="horizontal"/>
  <Orientation item="Line-2" degree="vertical"/>
  <Orientation item="Line-3" degree="horizontal"/>
  <Orientation item="Line-4" degree="vertical"/>
  -----
</Individual>
  -----
<Internal>
  -----
  <Coincident first="Line-1" second="Line-2"/>
  <GreaterLength first="Line-1" percent="40" second="
  <RightAngle first="Line-1" second="Line-2"/>
  <Coincident first="Line-1" second="Line-4"/>
  <GreaterLength first="Line-1" percent="40" second="
  <RightAngle first="Line-1" second="Line-4"/>
  <Coincident first="Line-2" second="Line3"/>
  <GreaterLength first="Line-2" percent="40" second="
  <RightAngle first="Line-3" second="Line-2"/>
  <Coincident first="Line--3" second="Line-2"/>
  <GreaterLength first="Line--3" percent="40" second=
  <RightAngle first="Line--3" second="Line-2"/>
  -----
</Internal>
```

SketchML

Library Editor Enviroment

- 4 Phasen: **First Layer Segmentation:** Zerlegt Stroke in Closed Regions und Polylinien
- **Second Layer Segmentation:** Nutzt First Layer Segmentation um Stroke in Ellipsen und Linien einzuteilen
- **Constraints Determination:** Prüft Constraints der einzelnen Elemente
- **SketchML Descriptor:** Übersetzt die gefunden Elemente in das SVG Format

SketchML

Sketch Recognition Enviroment

- Matcht gezeichnete Elemente mit den Elementen der geladenen Library(Domäne)
- Idee: Sowohl das gerade gezeichnete Element und die Elemente liegen im SVG Format vor
- Jedes SVG Element ist eine Node die mittels Constraints in Beziehungen zu den anderen Nodes stehen
- Problem: Matching aller Elemente sehr aufwändig
- Nachteil: Das Matching erfolgt nicht dynamisch beim Zeichnen

Abgrenzung

Abgrenzung(1)

- Vertex Detection über hohe Krümmung lieferte keine guten Ergebnisse
- Eigene funktionierende Idee: Vertex Detection über Veränderung der Richtung
- Funktioniert nur über Resampling wie bei Wolin[4]
- False Positive Filterung über Steigung zwischen 2 Punkten

Abgrenzung(2)

- Erkennung von Primitives ähnlich wie bei Yu[5]
- Object Recognition ähnlich wie LADDER[2]
- Unterschied: Möglichst die vorhandenen Manipulationsgesten von Surface nutzen
- Objekterkennung ohne Semantik, Interpretation erfolgt in höherer Ebene

Ende

Vielen Dank fürs Zuhören und viel Erfolg bei den Klausuren

[1] Danilo Avola, Andrea Del Buono, Gianfor Giorgio, Stefano Paolozzi, and Rui Wang. Sketchml a representation language for novel sketch recognition approach. In *Proceedings of the 2nd International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '09, pages 31:1–31:8, New York, NY, USA, 2009. ACM.

[2] Tracy Hammond and Randall Davis. Ladder, a sketching language for user interface developers. In *ACM SIGGRAPH 2007 courses*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[3] Tevfik Metin Sezgin, Thomas Stahovich, and Randall Davis. Sketch based interfaces: early processing for sketch understanding. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, New York, NY, USA, 2006. ACM.

[4] Aaron Wolin, B.Eoff, and Tracy Hammon. Shortstraw: A simple and effective corner finder for polylines. In *EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (2008)*, 2008.

[5] Bo Yu and Shijie Cai. A domain-independent system for sketch recognition. In *Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, GRAPHITE '03, pages 141–146, New York, NY, USA, 2003. ACM.