



Hochschule **RheinMain**

FACHBEREICH DESIGN INFORMATIK MEDIEN

FACHSEMINAR

Zum Thema:

Skizzen-basierte Bildersuche

Name, Vorname	Vogt, Patrick
Studiengang	Informatik (M. Sc.)
Datum (erstellt)	30. Oktober 2011
Datum (fertiggestellt)	28. Dezember 2011

Abstract

Das Ergebnis einer text-basierten Bildersuche ist oft unbefriedigend, da es beispielsweise nicht möglich ist, eine Priorität zwischen den einzelnen Suchbegriffen festzulegen. Des Weiteren kann der Benutzer nicht bestimmen, wo sich die einzelnen Suchbegriffe auf dem Bild befinden sollen.

Alternative Ansätze für die Bildersuche befinden sich derzeit in der Entwicklung. In der vorliegenden Ausarbeitung wird auf die sogenannte *sketch-based image search*¹ eingegangen. Diese Variante der Bildersuche ermöglicht es dem Benutzer eine Skizze von dem zu suchenden Bild anzufertigen, die dann mit einer Reihe von Bildern verglichen wird. Hierzu wird für jedes Bild, nach dem gesucht werden kann, ein vereinfachtes Konturenbild² berechnet und abgespeichert. Bei der Suche wird dann die vom Benutzer angefertigte Skizze mit diesen Konturenbildern verglichen.

Diese Suchmethode könnte sowohl genauere Ergebnisse bei einer Bildersuche liefern, als auch die Bildersuche bei Geräten mit Toucheingabe vereinfachen bzw. beschleunigen.

¹Die *sketch-based image search* wird auch oft als *sketch-based image retrieval (SBIR)* bezeichnet.

²Ein Konturenbild wird auch oft *abstract image* genannt.

Inhaltsverzeichnis

1	Einleitung	4
1.1	Grenzen der text-basierten Bildersuche	4
1.2	Alternative Bildersuchen	5
1.2.1	Manuelle Bildersuche	5
1.2.2	Suche anhand von existierenden Bildern	5
1.2.3	Spezielle Bildersuchmaschinen am Beispiel <i>SkyFinder</i> . .	6
1.2.4	<i>Sketch-based image search</i>	6
2	Skizzen-basierte Bildersuche	7
2.1	Konzept der skizzen-basierten Bildersuche	8
2.2	Die Realisierung: Berechnung der Konturenbilder	9
2.3	Die Realisierung: Speicherung und Vergleich von Skizze und Konturenbildern	14
3	Fazit und Ausblick	17

1 Einleitung

In der vorliegenden Ausarbeitung werden der etablierte Standard der Bildersuche, als auch die neuen Entwicklungen in diesem Bereich diskutiert. Da dies eine Ausarbeitung für einen Informatik-Studiengang ist, liegt der Schwerpunkt dieser Ausarbeitung auf den mathematisch-technischen Grundlagen der Bildersuche, als auch auf den Herausforderungen bei der tatsächlichen Realisierung einer Bildersuchmaschine.

Die Bildersuche im Internet kann einen Benutzer schnell frustrieren. Die Ergebnisse einer Bildersuche sind häufig zu ungenau und die Bilder besitzen meist eine zu geringe Auflösung [TYS09]. Außerdem kann der Benutzer durch die textbasierte Bildersuche nur unzureichend festlegen, welche Objekte auf dem Bild zu sehen sein sollen. Es ist daher für den Benutzer schwierig, das Bild, welches er sich in Gedanken vorstellt, in eine passende textbasierte Suchanfrage zu transformieren [WLZ10].

Im Folgenden sollen kurz die einzelnen Bildersuchvarianten vorgestellt und mit einer neuartigen Methode der Bildersuche, die detailliert vorgestellt wird, verglichen werden.

1.1 Grenzen der text-basierten Bildersuche

Die heutigen Bildersuchmaschinen basieren auf einer textlichen Suchanfrage [TYS09], wie z. B. die Google Bildersuche. Mithilfe dieser Bildersuche kann der Benutzer anhand einer text-basierten Anfrage (eine Liste von Suchwörtern) spezifizieren, nach welchen Bildern bzw. Motiven er suchen will.

Neben der einfachen Suche ist es anhand einer erweiterten Suche möglich die Bildersuche einzuschränken. Mithilfe von speziellen Parametern kann Einfluss auf die Größe des gesuchten Bildes (Auflösung und Seitenverhältnis), den Bildtyp (Foto oder Clipart), die Bildquelle (bspw. *flickr.com*) oder eine dominante Farbe innerhalb des Bildes genommen werden. Alle Parameter, die bei der Google Bildersuche benutzt werden können, finden sich unter [Gooa] und [Goob].

Weitergehende Parameter, mit denen der Benutzer festlegen kann, was genau und an welcher Position innerhalb des Bildes zu sehen sein soll, sind bei der textbasierten Bildersuche nicht vorgesehen.

1.2 Alternative Bildersuchen

Neben der reinen text-basierten Bildersuche gibt es auch noch andere Möglichkeiten eine Bildersuchanfrage zu spezifizieren.

1.2.1 Manuelle Bildersuche

Im Internet existieren mittlerweile Webseiten, die sich auf die Bereitstellung von freien Bildern spezialisiert haben. Hier kann von Hand in einer (nach einer bestimmten Kategorie) sortierten Bildermenge nach einem passenden Bild gesucht werden. So kann der Benutzer bspw. Texturen von verschiedenen Oberflächen suchen und kommt so nach mehreren Schritten zu einer passenden Textur. Dieser Ansatz besitzt den Vorteil, dass nur Bilder bereitgestellt werden, die sich auch als Texturen eignen, da diese von den Betreibern einer entsprechenden Webseite vortriert werden. Diesen Vorteil gibt es bei einer automatisierten und generischen Bildersuchmaschine nicht. Ein solcher manueller Suchansatz funktioniert jedoch nur bei einer relativ kleinen Bildermenge und ist somit nicht für die Verwendung bei mehreren Millionen Bildern geeignet [TYS09].

Eine entsprechende Webseite zum Suchen von Texturen für 3D-Objekte findet sich bspw. unter [cgt].

1.2.2 Suche anhand von existierenden Bildern

Eine weitere Idee ist, anstatt einer text-basierten Anfrage, ein sogenanntes Referenzbild als Suchanfrage zu benutzen. Eine Suchmaschine, in der eine solche Suche anhand eines existierenden Bildes möglich ist, sucht dann nach Bildern, die diesem Referenzbild gleichen.

Das Problem bei diesem Ansatz ist, dass zuerst ein geeignetes Referenzbild gefunden werden muss, das dann als Eingabe für die Suchmaschine dienen kann. Somit wurde das Suchproblem nicht wesentlich vereinfacht. Der Benutzer kann eventuell auch nicht entscheiden, welches Referenzbild am besten geeignet ist, um sein gewünschtes Bild zu finden, da er die Erkennungsalgorithmen der entsprechenden Suchmaschine nicht kennt [TYS09]. Es existieren bspw. Ansätze, die den Inhalt eines Bildes erkennen und mit dem Referenzbild vergleichen [JBS⁺06] oder die zwei unterschiedliche Bilder anhand von sogenannten *color histograms* vergleichen [LHE⁺07], die die Verteilung der einzelnen Farbitensitäten in einem Bild wiedergeben.

1.2.3 Spezielle Bildersuchmaschinen am Beispiel *SkyFinder*

Neben den generischen Bildersuchmaschinen existieren auch solche, die sich auf spezielle Arten von Bildern spezialisiert haben. Bei dieser Variante der Bildersuchmaschine können Annahmen, über die zu suchenden Bilder gemacht werden oder die suchbare Bildermenge kann entsprechend vorsortiert bzw. eingeschränkt werden.

Als Beispiel für eine solche Bildersuchmaschine soll hier kurz das System *SkyFinder* [TYS09] erläutert werden. *SkyFinder* ist eine Attribut-basierte Bildersuchmaschine, die auf das Finden von Himmelsbildern ausgerichtet ist.

Damit die Himmelsbilder analysiert werden können, muss jedes Bild zuerst in einen „Himmel-Bereich“ und einen „Nicht-Himmel-Bereich“ eingeteilt werden. Erst danach können die Algorithmen zur Klassifizierung angewendet werden, die entscheiden, ob es sich um einen blauen Himmel / einen wolkgigen Himmel oder einen Sonnenuntergang handelt [TYS09].

Das *SkyFinder*-System nutzt im Wesentlichen neuronale Netze, um aus einer Bildermenge bestimmte Eigenschaften zu extrahieren bzw. diese Bilder in bestimmte Kategorien einzuordnen [TYS09]. Die Himmelsbilder, die von *SkyFinder* unterstützt werden, werden so bspw. in „wolkgig / blauer Himmel“ oder anhand der Position der Sonne in verschiedene Gruppen eingeordnet. Diese Extraktion ist zwar automatisiert, jedoch werden wie bei jedem neuronalen Netz zuerst Trainingsdaten benötigt, sodass für einen bestimmten Datensatz erst einmal von Hand die einzelnen Attribute extrahiert werden müssen. Nach dieser Extraktion der suchrelevanten Attribute können die Bilder, die nun in jeweils 3 Kategorien eingeteilt sind, als Punkt in einem drei-dimensionalen Raum angeordnet und sortiert werden [TYS09]. Somit können zu einem bestimmten Bild auch ähnliche bzw. nur leicht unterschiedliche Bilder in dessen Umgebung gefunden werden.

Nachdem für jedes Bild *offline* die jeweiligen Attribute extrahiert wurden, ist es möglich eine normale text-basierte Anfrage, wie z. B. „Landschaft bei Sonnenuntergang mit der Sonne am linken unteren Bildende“ zu spezifizieren und eine passende Bildermenge zu erhalten (siehe Abbildung 1) [TYS09].

1.2.4 *Sketch-based image search*

Die *Sketch-based image search* (im Folgenden mit dem deutschen Begriff „skizzenbasierte Bildersuche“ beschrieben) bietet dem Benutzer bei der Spezifikation seiner Suche viele Freiheiten. Der Benutzer spezifiziert hier das zu suchende Bild

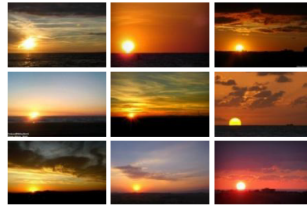


Abbildung 1: *SkyFinder*-Ergebnis für Anfrage „Landschaft bei Sonnenuntergang mit der Sonne am linken unteren Bildende“ [TYS09]

anhand einer von ihm gezeichneten Skizze [ERH⁺11]. Eine solche Skizze wird dann mit den in der Datenbank abgespeicherten Bildern verglichen. Dabei wird nicht das Bild direkt mit einer solchen Skizze verglichen, sondern aus den in der Datenbank gespeicherten Bildern wird ähnlich zu einer Miniaturansicht ein Konturenbild mit den groben Umrissen des jeweiligen Motivs berechnet und zusätzlich hinterlegt. Sollten sich das vorher berechnete Konturenbild und die vom Benutzer gezeichnete Skizze mehr als ein vorher definierter Grenzwert ähneln, wird das Bild dem Benutzer als Suchvorschlag angeboten. Andernfalls wird es ignoriert und dem Benutzer nicht als Suchergebnis vorgeschlagen.

Die skizzen-basierte Bildersuche wird schon seit Anfang der 1990er-Jahren intensiv erforscht [HK92]. Derzeit ist diese Forschung wiederbelebt worden, da Geräte mit Toucheingabe heute weit verbreitet sind. Es ist sicherlich einfacher eine Suchanfrage auf einem *Touchscreen* zu skizzieren, als diese auf einer virtuellen und kleinen Displaytastatur einzugeben.

Neben der reinen Anwendung für eine Bildersuche, könnte dieses Konzept jedoch auch zum Trainieren von Zeichnern benutzt werden [CNM05]. Beispielsweise könnte ein Programm die genaue Übereinstimmung (in Prozent) mit den berechneten Konturenbildern anzeigen, sodass der Zeichner versuchen kann, diesen Wert Schritt für Schritt zu erhöhen.

Die genaue Funktionsweise eines solchen Bildsuchsystems soll nun im nächsten Abschnitt genauer vorgestellt werden.

2 Skizzen-basierte Bildersuche

Die Motivation der skizzen-basierten Bildersuche ist, dass sie es dem Benutzer ermöglicht, genau das Bild zu finden, welches er sich in seinen Gedanken vorstellt [WLZ10]. Eine Transformation des vorgestellten Bildes in eine text-basierte

Anfrage oder in ein geeignetes Referenzbild entfällt bei dieser Variante der Bildersuchmaschine.

Damit eine skizzen-basierte Bildersuche funktionieren kann, müssen laut [ERH⁺11] folgende Punkte bei der Realisierung einer solchen speziellen Bildersuchmaschine beachtet werden:

- Die Suche muss trotz der geringen Eingabedaten (der eventuell sehr ungenauen Freihandskizze des Benutzers) genau sein und es dürfen dem Benutzer nur Bilder angezeigt werden, die (mit einer gewissen Toleranz) der Freihandskizze entsprechen [ERH⁺11].
- Die Suche ist unabhängig von der absoluten Position der Skizze, da die Skizze nur zum Festlegen des Motivs benutzt wird und nicht dessen absolute Position bestimmen soll [ERH⁺11].
- Die Suche soll tolerant gegenüber lokalen Deformationen sein, da eine Freihandskizze nicht den exakten Konturen eines fotografierten Motivs entsprechen wird [ERH⁺11].
- Die Suche soll interaktiv sein, sodass der Benutzer direkt, auch aus mehreren Millionen Bildern, eine entsprechende Ergebnismenge vorgeschlagen bekommt [ERH⁺11].

2.1 Konzept der skizzen-basierten Bildersuche

Der grundsätzliche Arbeitsablauf bei einer skizzen-basierten Bildersuche besteht aus den folgenden Prozessen [ERH⁺11]:

1. Der Benutzer malt eine Skizze des zu suchenden Bildes.
2. Der Benutzer bekommt interaktiv eine Bildermenge als Suchergebnis vorgeschlagen und kann anhand dieses Ergebnisses seine Skizze erweitern oder verwerfen.
3. Bei manchen Systemen (bspw. [ERH⁺11] und [TYS09]) ist es möglich ein Bild aus der Bildermenge zu selektieren und dieses weiter zu bearbeiten. Bei Photosketcher [ERH⁺11] ist es bspw. möglich aus mehreren gefundenen Bildern eine Bildkomposition zu erstellen.

Die hauptsächlichen Schwierigkeiten, die bei der Realisierung einer skizzen-basierten Bildersuche auftreten, ist die Extraktion eines Konturenbildes aus einer normalen Fotografie und der Vergleich dieser Konturenbilder mit der vom Benutzer spezifizierten Freihandskizze.

2.2 Die Realisierung: Berechnung der Konturenbilder

Eine der Vorbereitungen für ein System, das die skizzen-basierte Bildersuche unterstützen soll, ist die Berechnung der sogenannten Konturenbildern aus den zu suchenden Bildern. Wie kann nun aus einem normalen Bild bzw. Foto ein skizzenhaftes Konturenbild berechnet werden?

Die Kanten, die im Bild gefunden werden sollen, sollen einer Freihandskizze des jeweiligen Motivs ungefähr entsprechen, sodass dieses Konturenbild und eine Freihandskizze gut miteinander verglichen werden können. Solche Algorithmen bzw. Filter zum Finden von Kanten innerhalb eines Bildes werden bspw. in Bildbearbeitungsprogrammen, wie z. B. dem GNU Image Manipulation Program (GIMP) benutzt.

Diese Algorithmen sind jedoch (wie in Abbildung 2 zu sehen) zu ungenau, um direkt mit einer menschlichen Freihandskizze verglichen werden zu können. Das generierte Bild bzw. die gefundenen Kanten sind weiterhin zu detailliert.

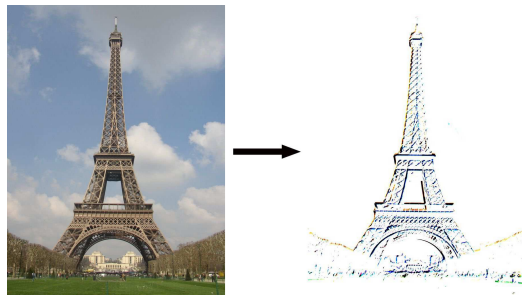


Abbildung 2: Berechnung eines Konturenbilds anhand der GIMP-Option „Kanten finden (Neon)“³

Ein Algorithmus, der auch aus beliebigen Fotografien skizzen-ähnliche Konturenbilder erzeugen kann, ist der sogenannte Canny-Filter⁴. Dieser Filter wurde von John F. Canny Mitte der 1980er Jahre entwickelt und in [Can86] veröffentlicht.

³Bildquelle: <http://www.eiffel-tower.us/Eiffel-Tower-Images/eiffel-tower-1.jpg>,
Letzter Abfruf: 23. November 2011

⁴Der Canny-Filter wird auch oft als *Canny edge detector* bezeichnet

Laut [Mar11, S. 347] ist der Canny-Filter ein effektiver Filter zum Erkennen von Kanten in Bildern und gleichzeitig der am häufigsten eingesetzte Filter für diesen Einsatzzweck. Das Ergebnisbild des Canny-Filters besitzt nur binäre Werte als einzelne Bildpunkte, dabei bedeutet „0“ = „keine Kante“ = „schwarzes Pixel“ und „1“ = „Kante“ = „weißes Pixel“.

Die Eigenschaften des Canny-Filters sind⁵ [Can86]:

- Gute Erkennung der vorhandenen Kanten: Die Wahrscheinlichkeit, dass „wahre“ Kanten erkannt werden soll maximal sein. Entsprechend soll die Wahrscheinlichkeit, dass eine „falsche“ Kante erkannt wird gering sein [Can86].
- Gute Lokalisierung der Kanten: Die Punkte, die als Punkte einer Kante markiert werden, sollen so nah wie möglich an der „wahren“ Kante liegen [Can86].
- Eindeutigkeit des Ergebnisses: Eine Kante soll eindeutig bestimmt werden. Es kann also nicht sein, dass eine Kante in mehrere (Teil-)Kanten aufgeteilt wird [Can86].

Hier, in dieser Ausarbeitung, wird nur die grundsätzliche Idee bzw. die einzelnen Schritte des Canny-Filters erläutert. Die mathematischen Details finden sich in [Can86].

Der Canny-Filter besteht aus 7 Einzelschritten, die zur Kantenextraktion eines Bildes angewendet werden müssen [Mar11, S. 347f], [SB11, S. 272ff]:

1. Reduzierung des Bildrauschens (engl. *Noise Reduction*)
2. Bestimmung der Kantenstärke
3. Berechnung der Kantenrichtung (diskrete Werte)
4. Ausdünnung der temporär berechneten Kanten
5. Einteilung der Kanten in zu dünne, schmale und starke Kanten
6. Löschung von zu dünnen Kanten
7. Invertierung des Bildes (optional)

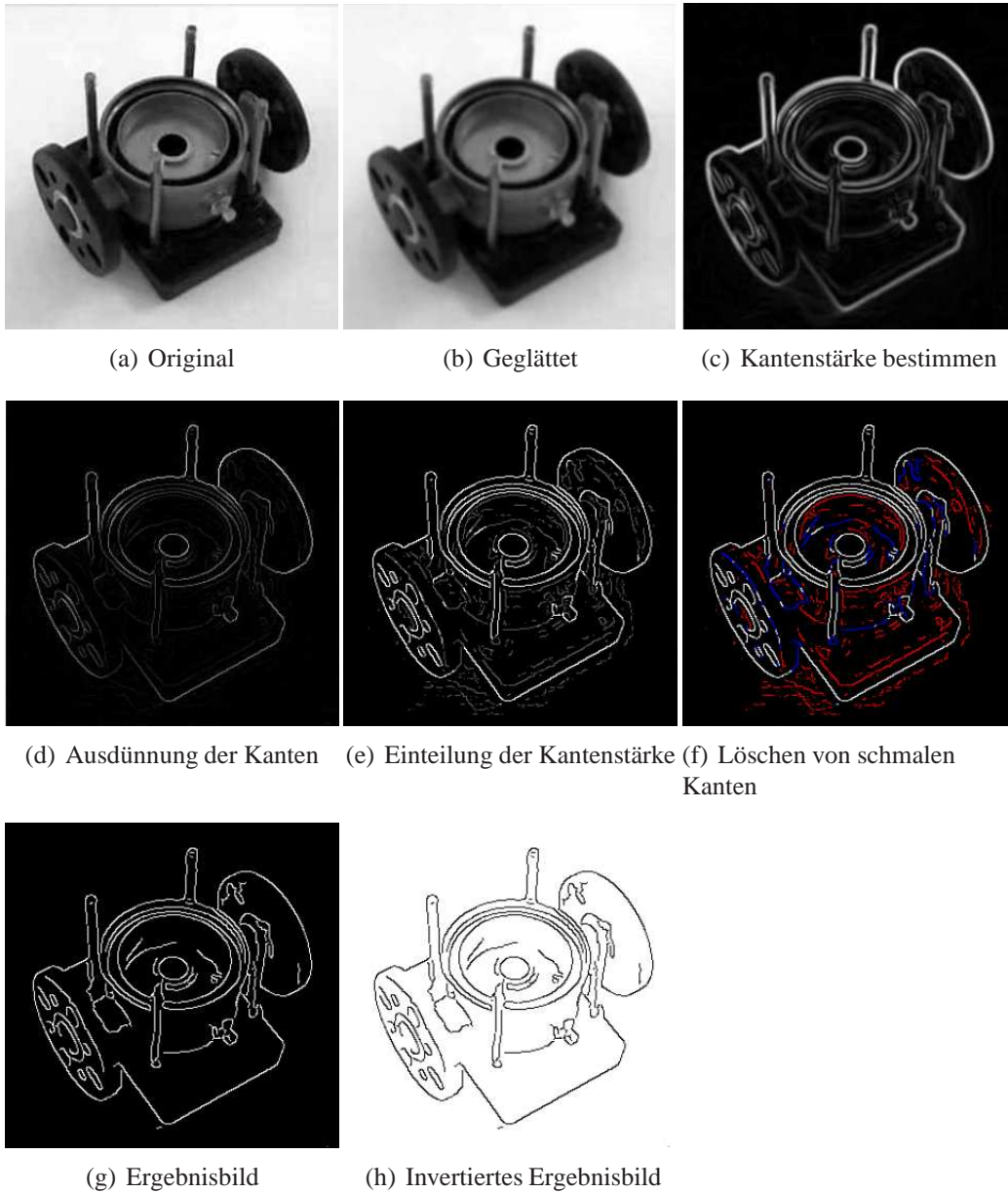


Abbildung 3: Die einzelnen Schritte des Canny-Filters angewendet auf ein Beispielbild (aus [cana])

Ein Beispiel für die einzelnen Schritte des Canny-Filters und dessen Teilergebnisse soll Abbildung 3 geben.

Im Folgenden sollen die einzelnen Schritte und deren Funktion kurz erläutert werden.

Reduzierung des Bildrauschens: Um zu verhindern, dass das im Bild vorhandene Rauschen als mögliche Kanten erkannt wird, muss das Bild zuvor geglättet werden. Dies geschieht mithilfe eines Gauß'schen Tiefpassfilters. Dieser Tiefpassfilter kann über einen freien Parameter σ (für die Standardabweichung der mehrdimensionalen Gauß'schen Normalverteilung (siehe Abbildung 4 zur Visualisierung)) justiert werden. Je größer der Wert für σ gewählt wird, desto „glatter“ wird das Bild, dafür werden aber auch feine Kanten von den späteren Schritten nicht mehr erkannt und ignoriert [Mar11, S. 347]. Welchen Einfluss der freie Parameter σ auf das Ergebnis der Kantenextraktion haben kann, zeigt Abbildung 5.

Bestimmung der Kantenstärke und der Kantenrichtung: Nachdem das Ausgangsbild geglättet wurde, findet der Canny-Filter mögliche Kanten anhand der größten Änderung der Graustufenintensität im Bild [cana]. Hierzu werden für jeden Punkt im geglätteten Bild lokale Gradienten berechnet (dessen Richtung und Länge bzw. Intensität) [cana], [Mar11, S. 348]. Bild 3 (c) zeigt, dass die gefundenen Kanten noch recht grob sind. Es ist jedoch schon zu sehen, dass schmale Kanten von starken Kanten unterschieden werden können.

Ausdünnung der temporär berechneten Kanten: In diesem Schritt werden die relativ breiten Kanten, die durch die Glättung und die damit verbundene Unschärfe des Bildes entstanden sind, ausgedünnt, indem nur jeweils die Mitte der entsprechenden Kante übernommen wird [cana], [Mar11, S. 348].

Einteilung der Kanten in zu dünne, schmale und starke Kanten: Mithilfe zweier unterschiedlicher Schwellwerte T_{low} und T_{high} werden die Kanten in 3 verschiedene Arten von Kanten aufgeteilt. Die Stärke der Kante, die vorher bei der Berechnung der lokalen Gradienten (für jeden einzelnen Bildpunkt) errechnet wurde, wird nun zum Vergleich mit diesen Schwellwerten hinzugezogen. Alle Kantenpunkte, die eine kleinere Intensität bzw. Stärke als T_{low} besitzen, werden gelöscht, da diese Kanten zu schwach sind. Alle Kantenpunkte, die zwischen T_{low} und T_{high} liegen werden als Punkte einer schmalen Kante markiert. Kantenpunkte größer als T_{high} werden als zu einer starken Kante gehörig markiert und werden auf jeden Fall im Endergebnis als Kante übernommen [cana], [Mar11, S. 348].

⁵Diese Kriterien wurden von John F. Canny bei der Formulierung bzw. dem Design des Algorithmus festgelegt und in entsprechende mathematische Formeln transformiert [Can86].

Löschung von zu dünnen Kanten: Punkte, die zu einer Kante gehören, deren Intensität zwischen T_{low} und T_{high} liegt, werden nun gesondert behandelt. Solche Kantenpunkte werden nur dann in der endgültigen Kantenextraktion benutzt, falls eine solche schmale Kante mit einer starken Kante verbunden ist. Ansonsten werden diese Kanten ignoriert und finden sich nicht im endgültigen Konturenbild wieder [cana], [Mar11, S. 348].

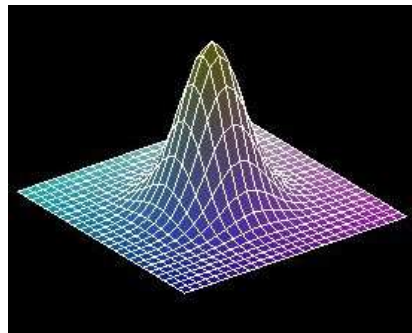
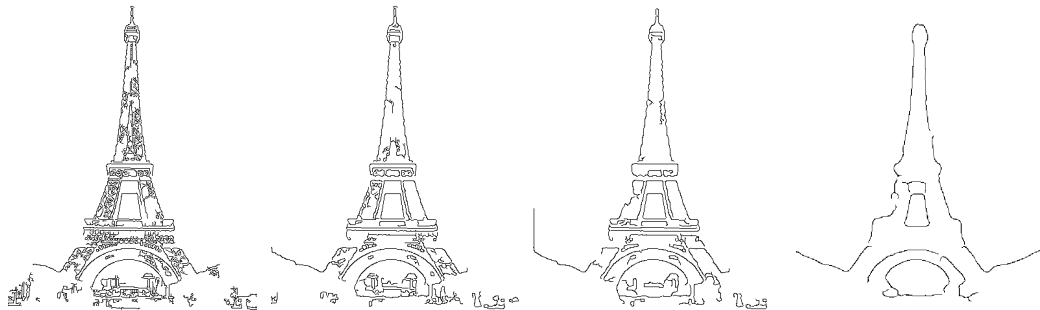


Abbildung 4: Visualisierung einer mehrdimensionalen Gauß'schen Normalverteilung⁶



(a) *Noise Reduction* mit $\sigma = 1.0$ (b) *Noise Reduction* mit $\sigma = 1.5$ (c) *Noise Reduction* mit $\sigma = 2.0$ (d) *Noise Reduction* mit $\sigma = 5.0$

Abbildung 5: Einfluss des freien Parameters σ auf das Ergebnis einer Kantenextraktion⁷

Invertierung des Bildes: Optional kann das so entstandene Bild (mit weißen Kanten auf schwarzem Grund) noch invertiert werden. Dies ist vor allem für die Verwendung von Konturenbildern bei der skizzen-basierten Bildersuche hilfreich, da

⁶Bildquelle: <http://upload.wikimedia.org/wikipedia/de/a/a8/MehrdimNormalvert.jpg.jpg>,
 Letzter Abfruf: 25. November 2011

⁷Dieses Bild wurde mithilfe von Abbildung 2 und der Webseite unter [canb] selbst erstellt.

die Freihandskizze ebenfalls mit schwarzen Konturen auf weißen Grund vorgehalten wird.

Eine Möglichkeit, um den Canny-Filter online anzuwenden findet sich unter [canb].

2.3 Die Realisierung: Speicherung und Vergleich von Skizze und Konturenbildern

Die Speicherung eines Konturenbildes für jedes suchbare Bild erscheint zuerst als zusätzlicher *Overhead* zur Speicherung des eigentlichen Bildes. Jedoch werden bei der skizzen-basierten Bildersuche keine weiteren Schlüsselwörter oder weitere Metadaten benötigt, die zusätzlich zum Bild abgespeichert werden müssen [ERH⁺11]. Außerdem sind die Konturenbilder nur Schwarz-Weiß-Bilder und besitzen eine geringere Auflösung, als das eigentliche Bild [ERH⁺11].

In [ICWF01] wird ein einfacher Ansatz zum Vergleich zwischen gezeichneten und berechneten Skizzen bzw. Konturenbildern beschrieben. Hierfür wird für jede Skizze ein Histogramm berechnet, das die Skizze eindeutig wiedergeben bzw. charakterisieren soll.

Um eine Skizze in einem entsprechenden Histogramm zu charakterisieren wird zuerst die konvexe Hülle⁸ der jeweiligen Skizze bestimmt. Danach wird die Krümmung⁹ an den einzelnen Kanten der Skizze anhand einer Differentialgleichung (siehe Gleichung 1) für jedes Pixel entlang der Kante berechnet. Aus diesen beiden Informationen (der konvexen Hülle der Skizze und die Krümmung entlang der Skizzenkante) werden für die Skizze sogenannte dominante Punkte bestimmt, die die Skizze repräsentieren sollen.

$$k(t) = \frac{\dot{x}(t)\ddot{y}(t) - \ddot{x}(t)\dot{y}(t)}{\sqrt{(\dot{x}(t)^2 + \dot{y}(t)^2)^3}}$$

Gleichung 1: Differentialgleichung zur Berechnung der Krümmung $k(t)$ entlang der Skizzenkante, wobei $k(t) = [x(t), y(t)]$ und t ein freier Parameter ist (aus [ICWF01])

⁸In [PS85, S. 95] wird die konvexe Hülle anschaulich mit einem (losgelassenen) Gummiband verglichen, welches um eine Punktmenge herum angebracht wird.

⁹Die Krümmung in einem zwei-dimensionalen Pixel entlang der Skizzenkante kann vereinfacht mit der Änderung bzw. der Steigung der x - und y -Koordinate in dessen Umgebung verglichen werden.

Die dominanten Punkte werden nun entlang der konvexen Hülle über vordefinierte Kriterien bestimmt. Den größten Einfluss, ob ein Punkt ein dominanter Punkt wird, hat die vorab berechnete Krümmung an den jeweiligen Pixel der Kante. Eine hohe Krümmung in einem Pixel bedeutet, dass es an dieser Stelle in der Skizzenkante einen markanten Knick gibt. Diese Knicke bzw. der Punkt wo dieser Knick stattfindet wird als dominanter Punkt der konvexen Hülle übernommen (vgl. Abbildung 7). In Abbildung 7 kann jedoch erkannt werden, dass nicht alle Knicke der Skizzenkante (die auf der konvexen Hülle liegen) auch dominante Punkte repräsentieren. Das zweite Ohr des Hunds wird trotz der starken Krümmung nicht als dominanter Punkt aufgenommen. Das Verfahren, was hier verwendet wurde, garantiert u.a., dass dominante Punkte einen Mindestabstand besitzen müssen. Eine detaillierte Beschreibung des Algorithmus zur Bestimmung der dominanten Punkte einer Skizze findet sich in [ICWF01].

Nach der Bestimmung der dominanten Punkte wird der Mittelpunkt der konvexen Hülle berechnet und die dominanten Punkte werden durch einen sogenannten Spoke-Vektor¹⁰ ersetzt. Ein solcher Vektor wird zwischen jedem einzelnen dominanten Punkt und dem Mittelpunkt der konvexen Hülle gezogen [ICWF01]. Eine entsprechende Visualisierung dieses Konzepts für eine Beispielskizze befindet sich in Abbildung 7.

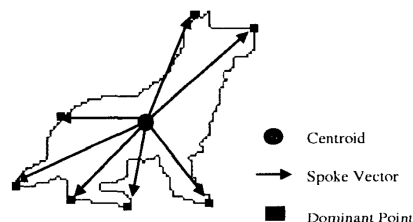


Abbildung 7: Die berechneten Werte einer Beispielskizze (aus [ICWF01])

Abschließend werden aus diesen Spoke-Vektoren zwei Eigenschaften extrahiert, der Winkel zwischen zwei benachbarten Spoke-Vektoren und die Fläche, die das entsprechende aufgespannte Dreieck zwischen den Vektoren enthält [ICWF01].

Diese beiden berechneten Eigenschaften können nun in einem Balkendiagramm bzw. in einem Histogramm dargestellt werden. In diesem Histogramm werden die kummulierte Summe der Dreiecksflächen in Abhängigkeit des Winkels zwischen den benachbarten Spoke-Vektoren aufgetragen. Damit dieses Histogramm nicht zu detailliert wird, werden die Winkel zuvor in 18 10°-breite Intervalle aufgeteilt

¹⁰Das englische Wort *Spoke* kann in diesem Fall am besten mit dem deutschen Wort „Speiche“ übersetzt werden.

[ICWF01]. Das Histogramm für die zuvor dargestellte Skizze befindet sich in Abbildung 8.

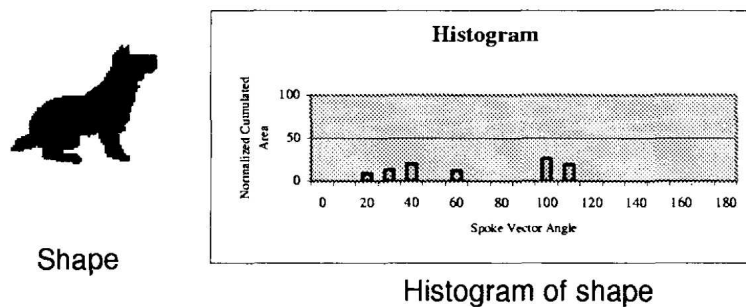


Abbildung 8: Der *Histogram Descriptor* für eine Beispielkontur (aus [ICWF01])

Der Vergleich zweier Skizzen kann nun auf die Differenz der entsprechenden Histogramme vereinfacht werden. Als Bilder werden nur solche Skizzen vorgeschlagen, bei denen die so berechnete Differenz minimal ist [ICWF01].

Dieser Ansatz besitzt jedoch den Nachteil, dass für jede einzelne Suchanfrage viel Daten berechnet werden müssen, nämlich jeweils die Differenz der beiden Histogramme. Der hier vorgestellte Algorithmus zum Vergleich von zwei schwarz-weiß Skizzen bzw. Konturenbildern wurde im Jahr 2001 lediglich mit einer Menge von 400 Bildern getestet [ICWF01]. Dabei lagen die Antwortzeiten des Systems bei weniger als einer Sekunde [ICWF01]. Dieser Ansatz ist also nicht auf eine Ansammlung von mehreren Millionen Bildern übertragbar [ICWF01].

Ein anderer Ansatz wird in [KKOH92] verfolgt. Dort werden das Konturenbild und die Freihandskizze zuerst in 8×8 -große Blöcke zerteilt. Danach wird die lokale Korrelation¹¹ zwischen den einzelnen Blöcken von Skizze und Konturenbild berechnet. Aus diesen für jeden Block berechnete lokale Korrelation wird dann eine globale Korrelation bestimmt, die letztendlich die Ähnlichkeit der Skizze mit dem in der Datenbank abgespeicherten Bild repräsentiert. Die Bilder mit den maximalen berechneten globalen Korrelationen sind der Freihandskizze also am ähnlichsten [KKOH92].

Bei der Speicherung der Bildersuche gibt es unterschiedliche Lösungen. Frühere Realisierungen von skizzen-basierten Bildersuchmaschinen benutzen normale Datenbanken zur Speicherung der suchbaren Bilder [KKOH92]. Neuere Realisierungen benutzen zum Vergleich zwischen den Skizzen bzw. Konturenbildern sogenannte *bag-of-features*-Ansätze. Nachdem aus der Skizze ein sogenanntes *Fea-*

¹¹Mithilfe der lokalen Korrelation kann zwischen zwei unterschiedlichen Signalen (hier: zweidimensionale Bilddaten) die Ähnlichkeit zwischen diesen Signalen bestimmt werden.

ture extrahiert wurde, wird dieses intern auf ein visuelles Vokabular abgebildet, welches den Index zu einer entsprechenden Ergebnisbildermenge darstellt. Für die genaue Erläuterung der Speicherung und die Bestimmung der Ergebnismenge für eine Skizze sei hier auf [CWZZ11] verwiesen.

3 Fazit und Ausblick

Es wurde kurz das Konzept der skizzen-basierten Bildersuche vorgestellt und die beiden hauptsächlichen Herausforderungen bei der Realisierung einer solchen Bildersuchmaschine vorgestellt: Die Kantenextraktion und der Vergleich der (vom Benutzer angefertigten) Freihandskizze und einem vorab berechneten Konturenbild.

Die Algorithmen, die hier vorgestellt wurden, wurden in der Zwischenzeit weiterentwickelt und verbessert. Eine verbesserte Variante des Canny-Filters findet sich bspw. in [BZW05]. In [CMN04] und [CNM05] wird die Beschreibung einer Skizze in einem Histogramm so erweitert, dass es sogar möglich ist, rotierte Bilder in einer entsprechenden Datenbank zu finden. Die Funktionsweise dieser verbesserten Methoden ist aber mit den hier vorgestellten Techniken vergleichbar [EHBA10]. Ein Survey-Artikel über die Entwicklungen der letzten Jahre findet sich in [EHBA10].

Die Menge der suchbaren Bilder stieg von ca. 400 Bildern [KKOH92] im Jahr 1992 bis hin zu 1.5 Millionen Bildern mit insgesamt 400GB Speicherplatz [ERH⁺11]. Eine produktive skizzen-basierte Bildersuche mit weit über 10 Millionen Bildern ist mit den vorgestellten Techniken nicht möglich, da diese nicht skalierbar sind [EHBA10].

Die Probleme liegen hier vor allem in der Erkennung und der gleichzeitigen Suche in der Datenbank. Diese beiden Themen sind eng miteinander verknüpft und werden in neueren Artikeln auch zusammen behandelt [CWZZ11].

Eine skizzen-basierte Bildersuche, die online getestet werden kann, ist *retrivr* [ske]. Des Weiteren existieren mehrere prototypische Entwicklungen von solchen Bildersuchmaschinen (bspw. [WLZ10] und [ERH⁺11]), diese können derzeit jedoch nicht online getestet werden.

Die skizzen-basierte Bildersuche befindet sich noch in der Entwicklung und wird (sofern sie erfolgreich realisiert werden kann) die text-basierte Suche ablösen bzw. verdrängen [EHBA10].

Ein Problem liegt in der geforderten Interaktivität eines solchen Systems. Laut [EHBA10] muss eine solche Bildersuchmaschine interaktiv sein und somit in Echtzeit dem Benutzer eine passende Ergebnisbildermenge vorschlagen. Nur dadurch kann der Benutzer nach mehreren Schritten, das Bild finden, welches er sich in Gedanken vorstellt [EHBA10]. Das Problem liegt weniger in der Speicherung der entsprechenden suchbaren Bilder, sondern vielmehr in der richtigen Extraktion von relevanten Informationen aus den suchbaren Bildern und dessen Verarbeitung bei einer Suchanfrage [EHBA10].

Artikel

- [BZW05] BAO, Paul ; ZHANG, Lei ; WU, Xiaolin: Canny Edge Detection Enhancement by Scale Multiplication. In: *IEEE Trans. Pattern Anal. Mach. Intell.* (2005), S. 1485–1490
- [Can86] CANNY, John: A Computational Approach to Edge Detection. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* (1986), S. 679 –698
- [CMN04] CHALECHALE, A. ; MERTINS, A. ; NAGHDY, G.: Edge image description using angular radial partitioning. In: *Vision, Image and Signal Processing, IEE Proceedings* (2004), S. 93 – 101
- [CNM05] CHALECHALE, Abdolah ; NAGHDY, Golshah ; MERTINS, Alfred: Sketch-based Image Matching Using Angular Partitioning. In: *Systems, Man and Cybernetics, Part A: Systems and Humans* 35 (2005), S. 28–41
- [CWZZ11] CAO, Yang ; WANG, Changhu ; ZHANG, Liqing ; ZHANG, Lei: Edgel Inverted Index for Large-Scale Sketch-based Image Search. In: *The 24th IEEE Conference on Computer Vision and Pattern Recognition* 24 (2011), S. 761–768
- [EHBA09] EITZ, Mathias ; HILDEBRAND, Kristian ; BOUBEKEUR, Tamy ; ALEXA, Marc: PhotoSketch: a sketch based image query and compositing system. (2009), S. 60:1–60:1
- [EHBA10] EITZ, Mathias ; HILDEBRAND, Kristian ; BOUBEKEUR, Tamy ; ALEXA, Marc: An evaluation of descriptors for large-scale image retrieval from sketched feature lines. In: *Computers and Graphics* (2010), S. 482 – 498
- [ERH⁺11] EITZ, Mathias ; RICHTER, Ronald ; HILDEBRAND, Kristian ; BOUBEKEUR, Tamy ; ALEXA, Marc: Photosketcher: Interactive Sketch-Based Image Synthesis. In: *Computer Graphics and Applications* 31 (2011), S. 56 – 66
- [HK92] HIRATA, Kyoji ; KATO, Toshikazu: Query by visual example. In: *Lecture Notes in Computer Science - Advances in Database Technology - EDBT '92* (1992), S. 56–71

- [ICWF01] IP, H.H.S. ; CHENG, A.K.Y. ; WONG, W.Y.F. ; FENG, Jun: Affine-invariant sketch-based retrieval of images. In: *Computer Graphics International 2001. Proceedings* (2001), S. 55–61
- [JBS⁺06] JOHNSON, M. ; BROSTOW, G. J. ; SHOTTON, J. ; ARANDJELOVIC, O. ; KWATRA, V. ; CIPOLLA, R.: Semantic Photo Synthesis. In: *Computer Graphics Forum 25* (2006), S. 407–413
- [JFS95] JACOBS, Charles E. ; FINKELSTEIN, Adam ; SALESIN, David H.: Fast multiresolution image querying. In: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (1995), S. 277–286
- [KKOH92] KATO, T. ; KURITA, T. ; OTSU, N. ; HIRATA, K.: A sketch retrieval method for full color image database-query by visual example. In: *Pattern Recognition, 1992. Vol.1. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on* (1992), S. 530–533
- [LHE⁺07] LALONDE, Jean-François ; HOIEM, Derek ; EFROS, Alexei A. ; ROTHER, Carsten ; WINN, John ; CRIMINISI, Antonio: Photo clip art. In: *ACM Trans. Graph.* 26 (2007)
- [TYS09] TAO, Litian ; YUAN, Lu ; SUN, Jian: SkyFinder: attribute-based sky image search. In: *ACM Trans. Graph.* 28 (2009), S. 68:1–68:5
- [WLZ10] WANG, Changhu ; LI, Zhiwei ; ZHANG, Lei: MindFinder: image search by interactive sketching and tagging. In: *WWW '10 Proceedings of the 19th international conference on World wide web 19* (2010), S. 1309–1312

Bücher

- [Mar11] MARQUES, Oge: *Practical Image and Video Processing Using MATLAB*. Wiley-IEEE Press, 2011. – ISBN 978–0470048153
- [PS85] PREPARATA, Franco P. ; SHAMOS, Michael I.: *Computational Geometry: An Introduction*. 1st Edition. Springer, 1985. – ISBN 978–0–387–96131–6
- [Rus11] RUSS, John C.: *The Image Processing Handbook*. 6th Edition. CRC Press, 2011. – ISBN 978–1439840634

- [SB11] SOLOMON, Chris ; BRECKON, Toby: *Fundamentals of Digital Image Processing - A Practical Approach with Examples in MATLAB*. Wiley blackwell, 2011. – ISBN 978–0470844731

Internet

- [cana] *Advanced Image Processing - Canny Edge Detection*. http://www.cvmt.dk/education/teaching/f09/VGIS8/AIP/canny_09gr820.pdf, Abruf: 23. November. 2011
- [canb] *Web-enabled image processing operators - Canny edge detector*. <http://matlabserver.cs.rug.nl>, Abruf: 23. November. 2011
- [cgt] *CG Textures - Textures for 3D, graphic design and Photoshop!* <http://www.cgtextures.com/>, Abruf: 19. November. 2011
- [Gooa] GOOGLE INC. (Hrsg.): *Google - Erweiterte Bildersuche*. http://www.google.de/advanced_image_search?hl=de, Abruf: 19. November. 2011
- [Goob] GOOGLE INC. (Hrsg.): *Google - Suchhilfe: Verfeinerung Ihrer Suche*. <http://www.google.de/intl/de/help/refinesearch.html>, Abruf: 19. November. 2011
- [min] *MindFinder - TechDemo (Video)*. <http://research.microsoft.com/en-us/projects/mindfinder/mindfinder-techfest2011.wmv>, Abruf: 16. November. 2011
- [ske] *retrievr - Search by sketch*. <http://labs.systemone.at/retrievr/>, Abruf: 02. Dezember. 2011