



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Seminarausarbeitung WS06/07
Bildverarbeitung für das Projekt FAUST
Schwerpunkt Skelettierung
Ingmar Gründel

*Fakultät Technik und Informatik
Department Informatik
Masterkurs Informatik
Betreuer: Prof. Dr.-Ing. Bernd Schwarz
Datum: 15. Februar 2007*

*Faculty of Engineering and Computer Science
Department of Computer Science*

Inhaltsverzeichnis

Abbildungsverzeichnis	2
1 Einleitung	3
1.1 Motivation für die Bildskelettierung	3
1.2 Anwendungen der Skelettierung	3
2 Grundlagen	4
2.1 Charakterisierungen für Skelette	4
2.2 Skelette in elektronischen Bildern	5
2.3 Morphologische Operatoren und Strukturierende Elemente	5
2.4 Algorithmen zur Skelettierung	6
2.4.1 Skelettierung mit morphologischen Operatoren	6
2.4.2 Skelettierung mit der Euler'schen Charakteristik	7
2.4.3 Skelettierung nach Ji und Piper	8
3 Realisierung	9
3.1 Prototyping einer Algorithmenkette	9
3.2 Anwendung morphologischer Skelettierung und Ji und Piper Skelettierung	10
4 Zusammenfassung und Ausblick Masterarbeit	12
Literatur	13

Abbildungsverzeichnis

1 Systemarchitektur Videoüberwachung	4
2 Grassfeuer bei einem kontinuierlichen Quadrat	4
3 Skelett aus Kreisen mit maximalem Radius innerhalb des Segments	5
4 Strukturelemente	6
5 Beispiele Basisoperatoren	6
6 Strukturierende Elemente	7
7 Anwendung für das erste SE	7
8 Beispiele nach[15]: (a) $e=11, k=15, f=5 \Rightarrow E=1$. (b) $e=29, k=38, f=12 \Rightarrow E=3$. E ist die Anzahl der einfach zusammenhängenden Segmente	8
9 Beispiele für Linienenden in einer 3*3 Nachbarschaft	8
10 Punkte der Skelettlinie mit horizontalem, vertikalem und diagonalem Verlauf nach [19]	8
11 Impresario Software am Beispiel eines Skeletierungsgraphen.	10
12 Einfaches Beispiel zur Verdünnung und Form	10
13 Beispiel zur Verdünnung	11
14 Beispiel zur Verdünnung	11

1 Einleitung

Im Department Informatik an der HAW-Hamburg wird das Projekt „FAUST“ durchgeführt. Ziel ist es, eine autonome fahrbare Plattform aufzubauen, die in der Lage ist, sich ohne menschliches Eingreifen fortzubewegen. Dabei soll weder die Einrichtung, noch Menschen Schaden erleiden.

Die im Projekt genutzte Plattform wird in der Industrie als Trainingsgerät für größere Plattformen verwendet. Die Plattformen werden zum Beispiel in Fertigungsstraßen eingesetzt, um Material/das unfertige Produkt von einem Fertigungspunkt zum Nächsten zu bewegen.

Die FAUST Plattform wurde im Rahmen des Masterprojektes WS05/06 erfolgreich mit Hardware und Software ausgestattet, so dass sie durch einen Leitstand fernsteuerbar ist. Im Sommersemester 06 ist ein Laserscanner integriert worden, womit ein Ausweich- und Bremsassistent realisiert wurde. Das Masterprojekt WS06/07 integriert zur Zeit ein automatisches Einparksystem und setzt hierfür einen weiteren Laserscanner ein.

Bisher wurden im Rahmen des Projektes Laserscanner als Abstandssensoren verwendet. Das nächste Ziel soll eine Sensordatenfusion sein, bei der zusätzlich Videokameras mit hoher Bildrate (>50 fps) verwendet werden. Diese gewährleisten eine präzisere Objektverfolgung und Routenfindung.

1.1 Motivation für die Bildskelettierung

Die Bildverarbeitung ist ein breit gefächertes Gebiet und wird in vielen industriellen Anwendungen eingesetzt. Am Wirtschaftsstandort Hamburg gibt es eine Vielzahl an Firmen, wie zum Beispiel I-to-I[4], Olympus Industrial[5], Trioptics[7] oder Basler Vision Technologies[1], die Software- und Hardware-Systeme für die industrielle Bildverarbeitung entwickeln.

Die Skelettierung hat je nach Anwendungsfall unterschiedliche Aufgaben und soll für das Projekt FAUST als Werkzeug für die Objekterkennung hinterfragt werden. Die Übergänge zwischen Verdünnung von Linien und Objekten, hin zur Bestimmung von Mittellinien und Skeletten ist fließend. Dünne Linien können vektorisiert werden und sorgen für eine Bilddatenreduktion. Skelette bieten Parameter zum Vergleich von Formen um Objekte wiederzuerkennen.

Im Rahmen dieser Arbeit soll ein Einblick in die Verfahren der Skelettierung gegeben werden, des Weiteren werden gängige Algorithmen vorgestellt. Hinsichtlich einer zukünftigen Masterarbeit im Bereich Bildverarbeitung zeigt die Arbeit eine Softwareplattform zur Entwicklung von Algorithmenketten, in der als Beispiel einer der hier vorgestellten Algorithmen implementiert wurde. Die dort entwickelte Kette könnte dann in Spezialhardware, wie FPGAs oder System-On-Chip, realisiert werden.

1.2 Anwendungen der Skelettierung

Die Skelettierung eines Bildes hat mehrere Ziele, die sich grob in Verdünnung und Gewinnung von Formparametern unterscheiden lassen.

Verdünnte Linien lassen sich gut vektorisieren, was zu einer Datenreduktion führt. In einigen Fällen müssen vorher die störenden Äste entfernt werden, dieser Vorgang heißt Entbartung.

Außerdem kann ein Skelett benutzt werden, um Daten zu reduzieren, da der Vorgang mit bestimmten Skelettierungsalgorithmen reversibel ist[11].

Die Verdünnung wird im medizinischen und biologischen Bereich genutzt, um z.B. bei Wurzeln oder Blutbahnen die Gesamtlänge zu bestimmen[10]. Weitere Anwendungen finden sich im Bereich Schrift- und Fingerabdruckererkennung.

Die Gewinnung von Formparameter dient der Erkennung von Objektklassen, die sich durch ihre Form unterscheiden. Je nach Algorithmus erfahren die erkannten Objekte eine Unifikation, was im Bereich der KFZ-Erkennung bedeutet, dass die Form der Außenspiegel nicht so gravierend in ein Skelett eingeht, wie die Gesamtform des KFZ's. Die Skelettierung wird im Zusammenhang mit komplexen Algorithmen genutzt, die sich mit Form-matching beschäftigen. Allerdings weichen unter Umständen die dort benutzten Skelettierungsalgorithmen von den hier vorgestellten ab.

Im Bereich der Videoüberwachung kann die Skelettierung eingesetzt werden, um zum Beispiel KFZ-Typen, wie Pick-Up und Sportwagen zu unterscheiden, was in [16] gemacht wird. Des Weiteren ist mit einer Sternskelettierung die Bewegung von Fußgängern analysierbar. Siehe dazu [3].

Das in [2] vorgestellte Videoüberwachungssystem und Objekt-trackingssystem nutzt Skelettierung, um Objekte zu erkennen. Das dort verwendete Blockbild in Abbildung 1, kann als Vorlage für eine Anwendung im FAUST-Projekt dienen.

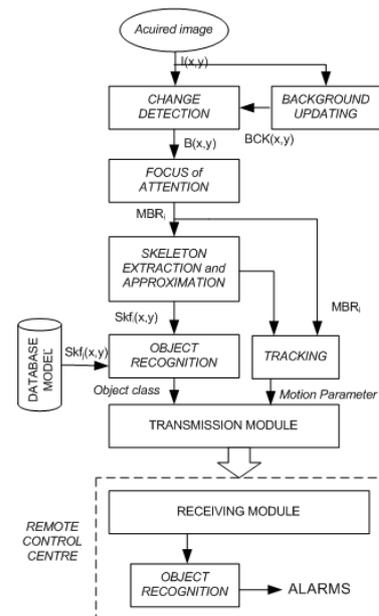


Abb. 1: Systemarchitektur Videoüberwachung

2 Grundlagen

2.1 Charakterisierungen für Skelette

In einer reellen zweidimensionalen Darstellung gibt es nach [18] mehrere mathematische Definitionen, die zu einem Skelett führen. Sie produzieren alle ein gleichwertiges Skelett.

- **Grasfeuerausbreitung.** Ein Feuer greift von allen Seiten mit der gleichen Geschwindigkeit an einer Menge X an. An den Stellen an denen sich die die Feuerfronten treffen, ist ein Skelettpunkt. Bei einem Quadrat wären die Diagonalen, die Mittelachse bzw. Skelettlinie, wie in 2 dargestellt. Im Falle eines Kreises wäre der Mittelpunkt die Skelettlinie.

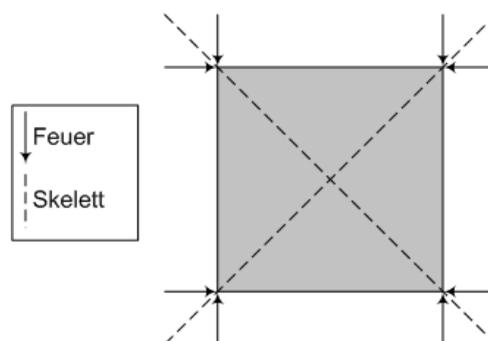


Abb. 2: Grassfeuer bei einem kontinuierlichen Quadrat

- **Maximale Scheiben.** Ein Skelett, definiert in Formel 1, ist die Linie der Mittelpunkte aus Kreisen mit maximalem Radius, die an mindestens zwei Punkten (y_1, y_2) den Rand des Segmentes X berühren. Dies ist in Abbildung 3 dargestellt.

$$x \in \text{Skelett}(X) \Leftrightarrow \exists y_1, y_2 \in \partial X \mid y_1 \neq y_2 \text{ und } d_\epsilon(x, \partial X) = d_\epsilon(x, y_1) = d_\epsilon(x, y_2) \quad (1)$$

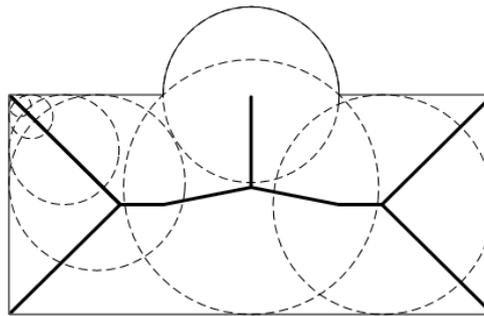


Abb. 3: Skelett aus Kreisen mit maximalem Radius innerhalb des Segments

- **Minimale Wege.** Der minimale Weg wird als kürzester Weg von einem Punkt in der Menge zum Rand der Menge interpretiert. Die Skelettlinie ist demnach die Menge aller Punkte, die nicht von einem minimalen Weg durchlaufen werden.

Die Skelette reagieren sehr sensibel auf Störungen am Rand, was aus Abbildung 3 deutlich wird. Für jede Störung entsteht ein neuer Ast.

2.2 Skelette in elektronischen Bildern

Im diskreten 2D Bild gibt es keine mathematische und exakte Definition eines Skeletts. Die Schwierigkeiten für eine Definition entstehen durch die Diskretisierung. Bei einem Rechteck mit einer geraden Anzahl an Pixel für die Höhe, kämen zwei Punkte in der Mitte in Frage für die Mittellinie. Daher ist ein Skelett an eine Reihe von Forderungen geknüpft. Die aufgezählten Kriterien nach [15] und [10] sind gleichzeitig ein Beurteilungskriterium für skeletterzeugenden Algorithmen.

- Das Skelett einer Fläche sollte aus einer 1 Pixel dicken Linie bestehen.
- Das Skelett einer Fläche sollte zusammenhängend sein und die ursprüngliche Fläche widerspiegeln.
- Die Skelettlinie sollte ungefähr in der Mitte liegen.
- Der Skelettalgorithmus sollte nicht empfindlich gegenüber Störungen am Rand sein, da durch jede Störung eine Verästelung entsteht.
- Der Algorithmus sollte nach einer endlich Anzahl von Iterationen stabil werden.

2.3 Morphologische Operatoren und Strukturierende Elemente

Mit morphologischen Operatoren lassen sich verschiedene Aufgaben bewerkstelligen. Hierbei können, Formen verändert, Merkmale von Formen berechnet und nach bestimmten Formen gesucht werden. Sie lösen Aufgaben, wie Kantenextraktion, Segmentierung und natürlich Skelettierung.

Die morphologischen Operatoren basieren auf der mathematischen Morphologie, die sich mit der Verknüpfung von Mengen auseinandersetzt. Es wird ein Eingangsbild über einen Operator mit einem strukturierenden Element verknüpft. Zu sehen in Formel 2. X steht für die Menge der Eingangsbildpixel, SE für das strukturierende Element (SE) und \diamond für einen Operator wie Schnittmenge, Differenz, Vereinigung, usw.

$$Y = X \diamond SE \quad (2)$$

Strukturierende Elemente sind in Abbildung 4 dargestellt. Die weißen Felder entsprechen einem zu betrachtenden Bildpunkt, die schwarzen sind Ankerpunkte. Die Verwendung wird anhand der zwei Basisoperationen Dilatation in Abbildung 5(a) und Erosion in Abbildung 5(b) gezeigt. Alle Felder eines SE werden logisch mit dem Bild verknüpft. Im Falle der Dilatation wird mit \vee -verknüpft und das Ergebnis im Ankerpunkt gespeichert. Es entsteht eine Flächenvergrößerung der im Beispiel „schwarz“ dargestellten Vordergrundbildpunkte. Bei der Erosion wird mit \wedge -verknüpft, was zu einer Flächenverringering der Vordergrundbildpunkte führt.

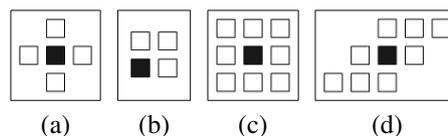


Abb. 4: Strukturelemente

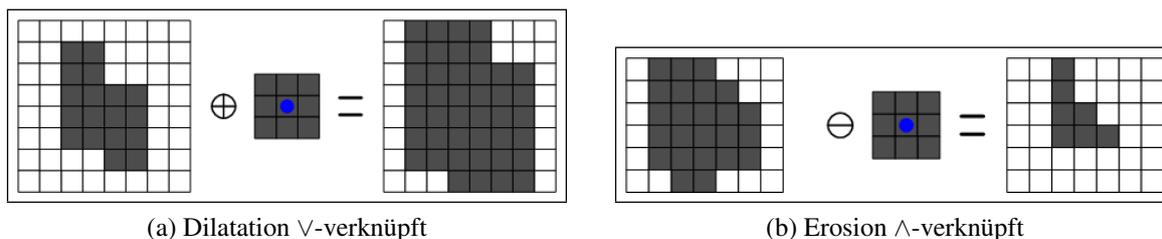


Abb. 5: Beispiele Basisoperatoren

Nach [15] gibt es eine schnelle Implementierungsmöglichkeit, in dem man das Eingangsbild in alle Felder des SE verschiebt. Anschließend wird das Eingangsbild mit der Verschiebung logisch verknüpft, wie bei den Basisoperationen erläutert. Das Ergebnis ist mit dem obigen Beispiel identisch.

2.4 Algorithmen zur Skelettierung

Es gibt eine Vielzahl unterschiedlicher Klassen von Skelettierungsalgorithmen, die auf Binärbildern arbeiten[15]. Es werden zwei Klassen vorgestellt, die Grundlagen vermitteln und in der Basisliteratur häufig genannt werden. Eine Klasse entscheidet für die Vordergrundpixel anhand vorher definierter SE, ob ein Pixel gelöscht werden darf oder nicht. Die gelöschten Pixel gehören dann zum Hintergrund. Eine weitere Klasse von Algorithmen arbeitet mit konturverfolgenden Verfahren und schaut sich die Randpunkte an, ob diese entfernt werden dürfen oder nicht. Die Algorithmen arbeiten iterativ und hören auf, wenn keine Vordergrundbildpunkte mehr zum Hintergrund gefügt werden können.

2.4.1 Skelettierung mit morphologischen Operatoren

Der morphologische Skelettierungsalgorithmus nach [15] wird mit der schnellen Methode(s. Kapitel 2.3) für die Verwendung morphologischer Operatoren implementiert. Es werden das Original-(Org) und 2 Hilfsbilder (H und T) benötigt. Die in Abbildung 6 dargestellten SE's dienen als Abarbeitungsvorschrift:

- 1 bedeutet SHIFT(Org,H,Richtung), NOT(H,H), AND(T,H,T)
⇒Extraktion von Punkten die für eine Löschung in Frage kommen
- 0 bedeutet SHIFT(Org,H,Richtung), AND(T,H,T)
⇒Löschung von Punkten die doch nicht gelöscht werden dürfen
- x ⇒keine Aktion

Die Vorschrift wird auf alle Felder des SE's um den Mittelpunkt angewendet. Zum Abschluss der Anwendung auf ein SE erfolgt eine XOR Verknüpfung des Hilfsbildes T mit dem Original. T ist eine Löschesmaske für nicht Skelettpunkte.



Abb. 6: Strukturierende Elemente

Alle SE werden der Reihe nach auf das Original angewendet, bis kein Vordergrundbildpunkte mehr entfernt wird. In Abbildung 7 sieht man einen Durchlauf für ein SE und am Ende das Ergebnis der Skelettierung eines Quadrats.

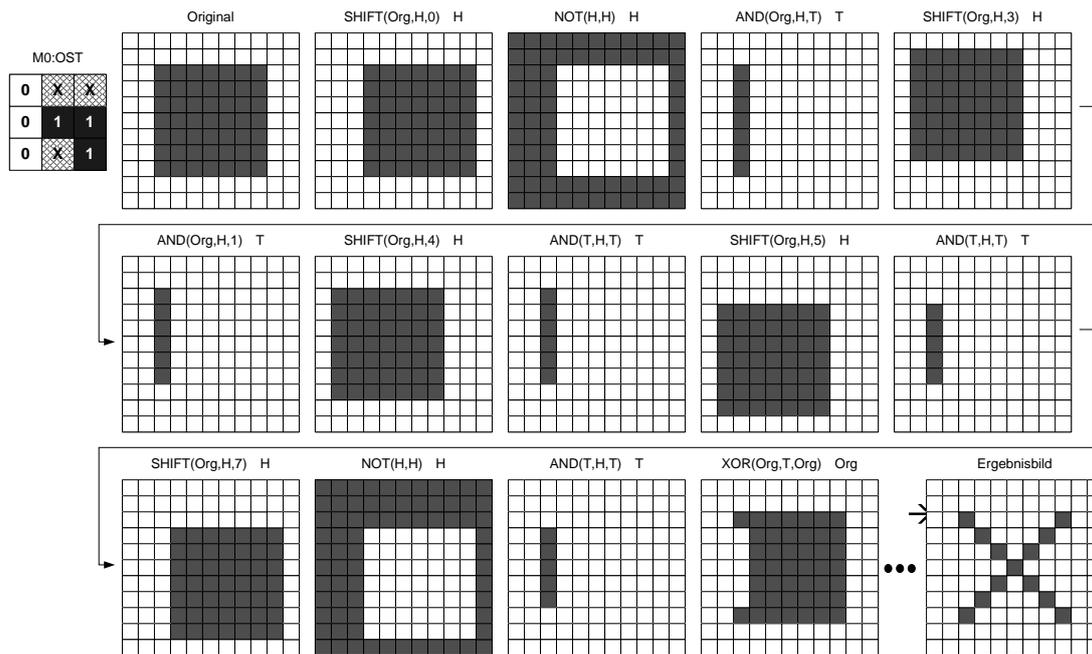


Abb. 7: Anwendung für das erste SE

2.4.2 Skelettierung mit der Euler'schen Charakteristik

Das Verfahren basiert auf der Euler'schen Charakteristik (vergl. [15]) für Polygonnetze und Polyeder. In der Formel 3 steht E für die Euler'sche Charakteristik, e für die Zahl der Ecken, k für die Anzahl der Kanten und f für die Anzahl der Flächen. Der Polygonsatz nach Euler besagt, dass ein Polygonnetz eine Charakteristik von eins und ein Polyeder von zwei hat. Berechnet man dies für einen einfachen Würfel, dann erhält man $E=2$, siehe Formel 4.

$$E = e - k + f \tag{3}$$

$$E = 8 - 12 + 6 = 2 \tag{4}$$

Diese Formel lässt sich auf ein binäres Bild übertragen, allerdings ändert sich leicht die Bedeutung der verwendeten Sumanden. e ist die Anzahl der Ecken der Vordergrundbildpunkte, k ist die Anzahl der Kanten und f ist die Anzahl der Hintergrundbildpunkte. E ist die Anzahl der einfach zusammenhängenden Segmente. Das Beispiel in Abbildung 8 zeigt die Berechnung der Charakteristik. Für (a) und (b) ergibt sich $E=1$ und für (c) $E=2$.

Mit dieser Formel können alle Segmente in einem Binärbild bestimmt werden. Bei der Anwendung werden 3×3 Umgebungen betrachtet, in denen dann entschieden wird, ob ein Vordergrundpixel entfernt werden darf. Die Löschung erfolgt, wenn sich die Euler'sche Charakteristik nicht

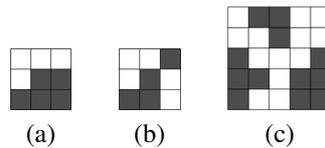


Abb. 8: Beispiele nach [15]: (a) $e=11, k=15, f=5 \Rightarrow E=1$. (b) $e=29, k=38, f=12 \Rightarrow E=3$. E ist die Anzahl der einfach zusammenhängenden Segmente

verändert.

Bei der Anwendung muß man die Linienendpunkte gesondert betrachten. Durch das Löschen des Endpunktes würde sich die Euler'sche Charakteristik nicht ändern. Daher dürfen Punkte nicht gelöscht werden, die nur einen Nachbarn haben, ansonsten würden keine Skelettlinien übrig bleiben. In Abbildung 9 sind Beispiele für Endpunkte gezeigt.

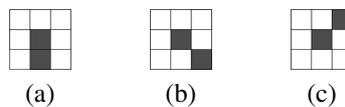


Abb. 9: Beispiele für Linienenden in einer 3*3 Nachbarschaft

Die übliche Bearbeitungsreihenfolge von oben links nach unten rechts ist in diesem Fall unangebracht, da sonst alle Skelettlinien am unteren Rand eines Segments liegen. Ein Ansatz ist, erst alle Bildpunkte (x,y) zu betrachten, wo x und y gerade und ungerade sind. Anschließend werden alle weiteren Punkte betrachtet. Man hat mindestens zwei Subiterationen. Abruchkriterium ist, dass kein weiteres Pixel entfernt wird.

Es gibt mehrere Arten um den Algorithmus zu beschleunigen. Eine ist, dass man das Bild in Subquadrate unterteilt und die bearbeitet, wo noch Pixel entfernt wurden. Eine Weitere ist das Ablegen aller drei mal drei Konstellationen in einer Liste, so dass die zweimalige Berechnung der Euler'schen Charakteristik gespart werden kann. Es gibt 512 Konstellation.

2.4.3 Skelettierung nach JI und Piper

Das Verfahren nach JI und Piper [8] ist ein Konturfolgealgorithmus, der im ersten Initialisierungsschritt alle Randpunkte in einen Puffer schreibt. Diese Punkte werden anschließend mit 3x3-Masken überprüft, ob es sich um Skelettpunkte handelt. Die gefundenen Skelettpunkte werden im Ausgangsbild abgelegt. Wird ein Randpunkt gelöscht, wird die Nachbarschaft darauf untersucht, ob ein neuer, in die Liste einzutragender Randpunkt, entstanden ist. In Abbildung 10 sind die Überprüfungsmasken dargestellt. Hier bedeutet „1“ Vordergrundbildpunkt, „0“ Hintergrundbildpunkt und „X“ ist nicht von Belang. Außer diesen Masken gibt es weitere, die an Endpunkten zum Einsatz kommen.

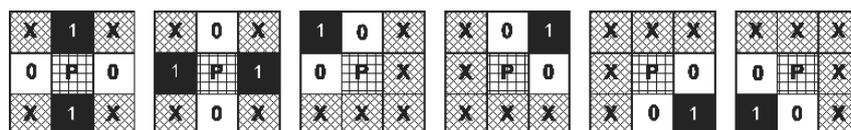


Abb. 10: Punkte der Skelettlinie mit horizontalem, vertikalem und diagonalem Verlauf nach [19]

3 Realisierung

3.1 Prototyping einer Algorithmuskette

Um ein Gefühl dafür zu erhalten, wie die gezeigten und weitere Algorithmen angewandt werden, ist es unerlässlich ein Software-Referenzmodell zu entwickeln, mit dem Prototypen von Algorithmusketten erprobt werden können. Erweist sich die getestete Kette als tauglich und muss Echtzeitfähigkeit hergestellt werden, ist eine Implementierung mit Spezialhardware zu prüfen. Als Spezialhardware kommen System-on-Chip Prozessoren in Frage, die aus einer PowerPC CPU und einem FPGA bestehen. Auf einem solchen System können Algorithmen durch Parallelisierung und Pipelining schneller ausgeführt werden.

Folgende Software wurde betrachtet:

- AdOculus [6] - Wird an der Hochschule für Angewandte Wissenschaften Hamburg im Wahlpflichtfach Digitale Bildverarbeitung[9] eingesetzt. Enthält gängige Algorithmen, die über graphische Elemente verkettet werden können. Skelettierung ist nicht enthalten. Die Software ist nicht erweiterbar.
- IGARIS Software aus Masterkurs Computergrafik und Bildverarbeitung[15] - Das Programm enthält eine Auswahl der im Buch vorgestellten Algorithmen. Die zum Buch erhältliche Version ist eine Light-Version, wo keine eigenen Bilder eingefügt werden können. Außerdem war die Bedienung umständlich, Algorithmusketten werden über Macros definiert. Es ist einer der zwei Skelettierungsalgorithmen aus dem Buch enthalten.
- Software aus Bildverarbeitung in der Praxis[19] - Hierbei handelt es sich um eine kostenlose DOS-Anwendung mit Algorithmen aus dem Buch. Es sind verschiedene Algorithmen realisiert unter anderem Skelettierung. Das Programm ist leider bedienerunfreundlich und Ketten können damit nicht realisiert werden. Der Sourcecode ist enthalten.
- LTI-Lib[17] - Diese Library wird an der HAW-Hamburg im Wahlpflichtfach Robot Vision [14] eingesetzt und enthält eine riesige Menge an Algorithmen. Eine Algorithmuskette muss dem Framework entsprechend von Hand codiert werden.
- Impresario[12] - Grafische Oberfläche für die Verknüpfung von Bildverarbeitungsalgorithmen aus der LTI-Lib. Die Funktionen die benötigt werden, können über eine DLL mit vorgegebenem Aufbau in die Software problemlos integriert werden.

Für das weitere Vorgehen wurde sich für die Verbindung aus der LTI-Lib und der Impresario-Software entschieden, da diese erweiterbar und mit der Oberfläche komfortabel zu bedienen sind. Die Software ist an der Rheinische Westfälische Technische Hochschule entstanden und ist frei verfügbar. Das Konzept wird in [13] näher erläutert. Eingebundene Funktionen werden aus dem Menü auf die Oberfläche gezogen und mit einem Verbinder über Eingang und Ausgang verbunden. In der Abbildung 11 ist ein Beispiel zu sehen.

Im Rahmen der Seminararbeit wurde die Impresario Software um eine DLL für die Skelettierung erweitert. Diese DLL enthält Verweise auf den, in der LTI-Lib realisierten, Skelettierungsalgorithmus nach Ji und Piper[8]. Außerdem wurde der morphologische Skelettierungsalgorithmus aus [15] implementiert und ebenfalls in die DLL aufgenommen.

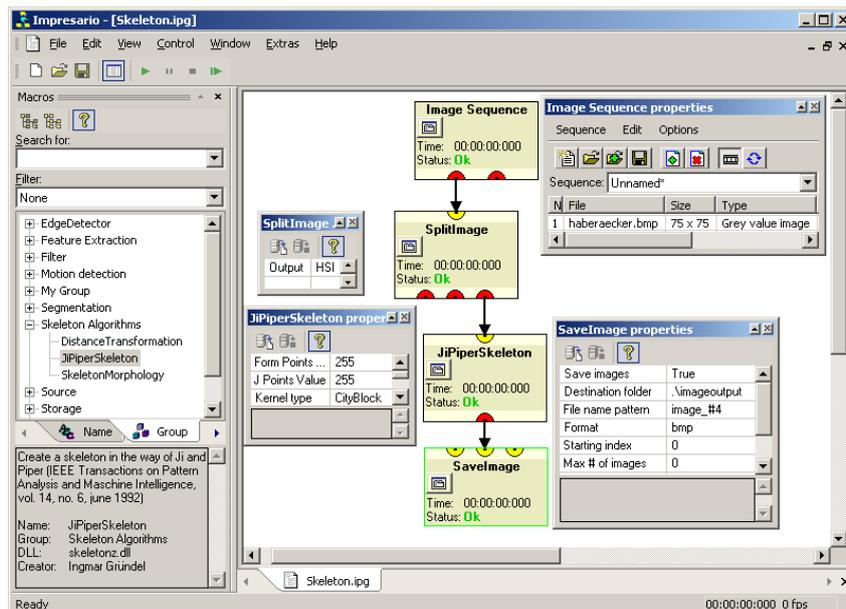


Abb. 11: Impresario Software am Beispiel eines Skeletierungsgraphen.

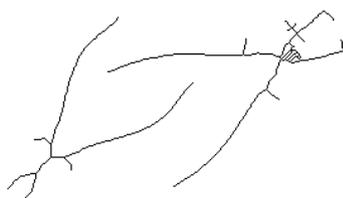
3.2 Anwendung morphologischer Skelettierung und Ji und Piper Skelettierung

In diesem Kapitel werden mehrere Beispiele gezeigt, die mit der, in Kapitel 3.1, erläuterten Software erstellt wurden. Zur Erzeugung der Ergebnisbilder wurden der morphologische Skelettierungsalgorithmus und der Ji und Piper Algorithmus verwendet.

Die in Abbildung 12(a) gezeigten Zangen, dienen als einfaches Beispiel. Hier kann man die Verdünnung und die Gewinnung von Formparametern erkennen. In den Ergebnisabbildungen (b) und (c) kann man den Unterschied zwischen den verschiedenen Zangenarten und zwischen den zwei Algorithmen sehen.



(a) Ausgangsbild



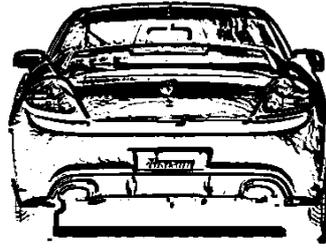
(b) Morphologische Skelettierung



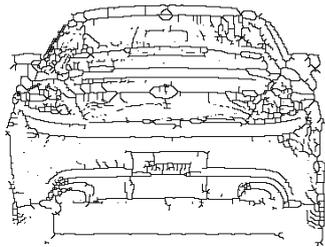
(c) Skelettierung nach Ji and Piper

Abb. 12: Einfaches Beispiel zur Verdünnung und Form

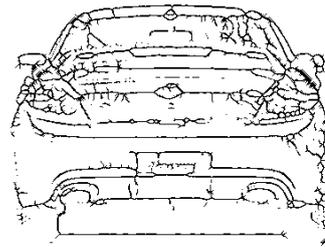
Abbildung 13 basiert auf einem Bild, welches durch einen Sobeloperator erzeugt wurde. Auf das Ausgangsbild 13a sind ebenfalls die beiden Algorithmen angewendet worden. Ziel war es die schwarzen Linien zu verdünnen, was in den beiden Abbildungen 13b und 13c zu sehen ist.



(a) Ausgangsbild Sobel



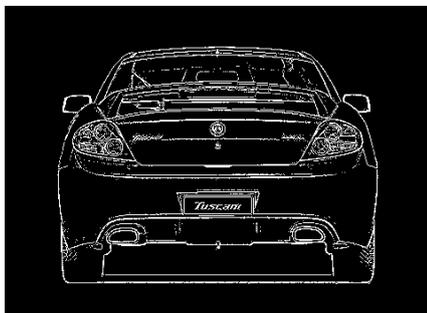
(b) Morphologische Skelettierung



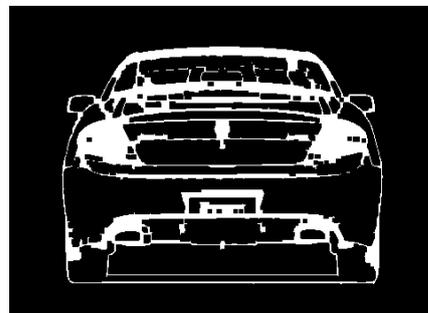
(c) Skelettierung nach Ji and Piper

Abb. 13: Beispiel zur Verdünnung

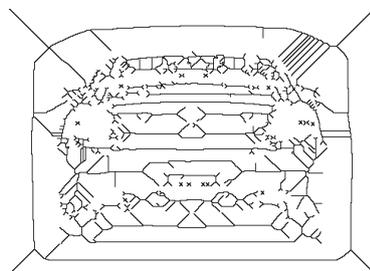
Im letzten Beispiel Abbildung 14 diene dasselbe Ausgangsbild 13a, allerdings nach einer Laplace-Filterung. Zusätzlich wurde eine Closing Operation ausgeführt, diese besteht aus den morphologischen Operationen Dilatation und Erosion. Sie sorgt dafür, dass Lücken und Öffnungen des Laplacelinienbildes geschlossen werden, so dass mehrere Flächen entstehen. Ziel der Anwendung der Algorithmen war es, die Skelette der Flächen zu finden.



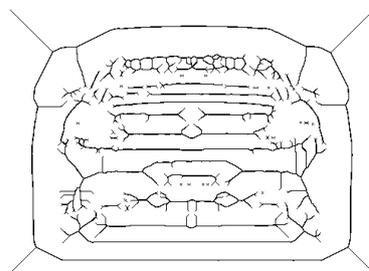
(a) Ausgangsbild Laplace



(b) Laplace nach Closing



(c) Morphologische Skelettierung



(d) Skelettierung nach Ji and Piper

Abb. 14: Beispiel zur Verdünnung

Würde man einen Vergleich zwischen diesem Fahrzeug und einem andersartigen Fahrzeug nach der Skelettierung durchführen, müsste man gravierende Abweichungen in der Form der Flächenskelette feststellen, ähnlich der Abweichungen beim Zangenbeispiel.

4 Zusammenfassung und Ausblick Masterarbeit

Das an der HAW Hamburg durchgeführte Projekt FAUST benötigt weitere Sensoren für die Umgebungserfassung. Zur Zeit werden Laserscanner zur Distanzemessung eingesetzt. Moderne Videosensoren bieten eine große Menge an erfassbaren Daten, diese müssen weiterverarbeitet werden, um relevante Informationen zu extrahieren. Objekte müssen erkannt und klassifiziert werden. Die Skelettierung, die in dieser Arbeit vorgestellt wurde, ist ein Werkzeug dafür. Es gibt die zwei Anwendungsbereiche Verdünnung und Extraktion von Formparametern. Die Verdünnung von Linien dient der Datenreduktion. Formparameter dienen der Widererkennung von Objekten oder der Erkennung von Objektklassen.

Es wurde definiert, was unter einem Skelett zu verstehen ist und wo der Unterschied zwischen der mathematischen Definition und Skeletten in elektronischen Bildern liegt. Der Hauptgrund für den Unterschied liegt in der Diskretisierung eines Bildes in Bildpunkte. Im Anschluss wurden morphologische Operatoren und strukturierende Elemente erläutert, da diese die Grundlage bei den Algorithmen bilden. Morphologische Operatoren speichern, die durch eine Rechenvorschrift verknüpften Werte eines SE, im Ankerpunkt.

Morphologische-, Euler'sche- und Ji und Piper Skelettierung sind Algorithmen zur Erzeugung von Skeletten. Die ersten beiden arbeiten auf den gesamten Punkten und der Letzte arbeitet auf den Randpunkten eines Binärbildes. Sie können alle für Verdünnung und Extraktion von Formparameter genutzt werden. Morphologische Skelettierung basiert auf den morphologischen Operatoren Dilatation und Erosion, die durch Verschiebungen und Verknüpfung der verschobenen Bilder erzeugt werden. Die Euler'sche Skelettierung benutzt die Euler'sche Charakteristik zur Berechnung von Polygonnetzen und Polyedern, welche außerdem für Binärbilder gilt. Der Ji und Piper Algorithmus benutzt SE ähnliche Masken zur Überprüfung der einzelnen Randpunkte und speichert gefundene Skelettpunkte im Ausgabebild.

Um die Algorithmen auf Beispielbilder anzuwenden, wurde nach einer Software gesucht. Bei den benutzten Programmen stellte sich heraus, dass sie nicht oder nur zum Teil die geforderten Funktionen enthielten und nicht erweiterbar waren. Allein die Kombination aus LTI-Lib und Impresario ist erweiterbar und es können Algorithmenketten aufgebaut werden. Der einzige Nachteil ist, dass nur die Ji und Piper Skelettierung enthalten war. In dieser Arbeit wurde die morphologische Skelettierung implementiert. Mit dieser Software wurden Beispielbilder erzeugt, die die Verdünnung und die Extraktion von Formparametern zeigen.

Im Rahmen einer Masterarbeit müssen weitere Skelettierungsalgorithmen implementiert und bewertet werden. Des Weiteren gibt es noch mehr Bildverarbeitungsalgorithmen, die neben der Skelettierung auf ihre Tauglichkeit zu untersuchen sind. Ziel der Arbeit ist es, ein Objekterkennungssystem in einem System-On-Chip(SoC) System zu realisieren, welches dann später in die Sensorfusion des FAUST-Systems integriert werden kann.

Die vorgestellte Applikation für das Prototyping wird von großem Nutzen sein, da erst eine Algorithmenkette entwickelt werden kann und dann das SoC programmiert wird.

Probleme könnte die Laufzeit der Algorithmen bereiten, da die Berechnung der Skelettbilder mit einem x86 PC mit 1,8 GHz zwischen drei und zehn Minuten dauert. Diese Rechenzeit könnte durch die Verwendung des FPGA-Teils eines SoC durch Pipelining und Parallelisierung verkürzt werden. Außerdem müssen noch Methoden entwickelt werden, durch welche die Skelette effizient verglichen werden können, da dieser Bereich bisher nicht behandelt wurde. Ein Ansatz wurde in [2] erläutert, in dem die Skelette approximiert und im Anschluss mit einer Datenbank verglichen wurden.

Literatur

- [1] AG, Basler: *Basler Vision Technologies Firmenhomepage*. Webseite, 2006. – <http://www.baslerweb.com>
- [2] FORESTI, Gian L.: Object Recognition and Tracking for Remote Video Surveillance. In: *IEEE Transactions on Circuits and Systems for Video Technology Vol. 9 No. 7* (1999), Oktober, S. 1045–1062
- [3] FUJIYOSHI, Hironobu ; LIPTON, Alan J.: Real-time Human Motion Analysis by Image Skeletonization. In: *IEEE Workshop on Applications of Computer Vision (WACV)* (1998), Oktober, S. 15–21
- [4] GMBH, I to-I Efkon m.: *I-to-I Firmenseite*. Webseite, 2006. – <http://www.i-to-i.de>
- [5] GMBH, OLYMPUS D.: *Olympus Industrial Firmenseite*. Webseite, 2006. – <http://www2.olympus.de/industrial>
- [6] GMBH, The Imaging Source E.: *Ad Oculos Software*. Webseite, 2006. – http://www.theimagingsource.com/de/products/software/windows_apps
- [7] GMBH, TRIOPTICS: *Trioptics Firmenseite*. Webseite, 2006. – <http://www.trioptics.com>
- [8] JI, Liang ; PIPER, Jim: Fast Homotopy-Preserving Skeletons Using Mathematical Morphology. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 14 No.6* (1992), Juni, S. 653–664
- [9] KÖLZER, Hans P.: *Wahlpflichtfach Digitale Bildverarbeitung*. Webseite, 2007. – <http://users.etch.haw-hamburg.de/users/koelzer/>
- [10] KOPP, Herbert: *Bildverarbeitung Interaktiv*. B. G. Teubner Stuttgart, 1997. – ISBN 3-5190-2995-2
- [11] KRESCH, Renato ; MALAH, David: Skeleton-Based Morphological Coding of Binary Images. In: *IEEE Transactions on Image Processing Vol. 7 No. 10* (1998), Oktober, S. 1387–1399
- [12] LIBUDA, Lars: *Impresario Software*. Webseite, 2005. – <http://www.techinfo.rwth-aachen.de/Software/Impresario/>
- [13] LIBUDA, Lars ; HAMACHER, Nico: Komfort als Entwicklungskriterium in der Systemgestaltung. In: *47 Fachausschusssitzung Anthropotechnik DGLR-Bericht 2005-05* (2005), Oktober, S. 19–39. ISBN 3-932182-44-8
- [14] MEISEL, Andreas: *Robot Vision*. Webseite, 2006. – <https://www.informatik.haw-hamburg.de/cms/?id=432>
- [15] NISCHWITZ, Alfred ; HABERÄCKER, Peter: *Masterkurs Computergrafik und Bildverarbeitung*. Friedr. Vieweg Sohn Verlag/GWV Fachverlage GmbH, Wiesbaden, 2004. – ISBN 3-528-05874-9
- [16] OZCANLI, Ozge C. ; TAMRAKAR, Amir ; KIMIA, Benjamin B. ; MUNDY, Joseph L.: Augmenting Shape with Appearance in Vehicle Category Recognition. In: *IEEE Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* (2006)
- [17] RHEINISCH WESTFÄLISCHE TECHNISCHE HOCHSCHULE(RWTH), Lehrstuhl für TI an d.: *LTI-Lib*. Webseite, 2005. – <http://ltilib.sourceforge.net/doc/homepage/index.shtml>
- [18] SOILLE, Pierre: *Morphologische Bildverarbeitung*. Springer Verlag, 1998. – ISBN 3-540-64323-0

-
- [19] STEINBRECHER, Rainer: *Bildverarbeitung in der Praxis*. München; Wien: Oldenbourg, 1993. – ISBN 3-486-22372-0. – <http://www.rst-software.de/dbv/download.html>
- [20] TÖNNIES, Klaus D.: *Grundlagen der Bildverarbeitung*. Pearson Studium, 2005. – ISBN 3-8273-7155-4