



Gegenstandserkennung und kategoriebasierter Transport anhand von kameraunterstützten NXT-Robotern

Studienarbeit

für die Prüfung zum
Bachelor of Engineering

von

Sebastian Hüther & Lorenzo Toso

23. April 2015

Bearbeitungszeitraum: 2 Semester

Matrikelnummer: 8853105 & 1906813

Kurs: TINF12B3

Studiengang: Informationstechnik

Ausbildungsfirma: Karlsruher Institut für Technologie

Betreuer: Gertrud Nieder

Inhaltsverzeichnis

Eidesstattliche Erklärung	IV
Abkürzungsverzeichnis	1
Abbildungsverzeichnis	2
1 Einleitung	3
2 Problemstellung	4
3 Materialien und Methoden	5
4 Hardwareumsetzung	6
4.1 Entwurf des NXT-Roboters	6
4.2 Steuerung des Roboters	6
4.3 Wahl des Kameramoduls	6
5 Softwareumsetzung	7
5.1 Wahl der Bildverarbeitungsbibliothek	7
5.2 Algorithmen zur Objekterkennung	8
5.3 Objektverfolgung	10
5.4 Entfernungsschätzung	13
5.5 Orientierung im Raum	15
5.6 Zielzonenbestimmung	15
5.7 Hauptschleife	16

Inhaltsverzeichnis	III
6 Tests des Robotersystems	17
6.1 Tests im gesicherten Rahmen	17
6.2 Realtests	17
7 Zusammenfassung und Ausblick	18
Literaturverzeichnis	V

Eidesstattliche Erklärung

Gemäß § 5 (3) der „Studien- und Prüfungsordnung DHBW Technik“ vom 22. September 2011.
Ich habe die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Ort, Datum

Unterschrift

Abkürzungsverzeichnis

Abbildungsverzeichnis

5.1	Beispiel einer Aufspaltung in RGB-Kanäle (links) und HSV-Kanäle (rechts)	9
5.2	Beispiel der Binarisierung des Sättigungskanals	9
5.3	Beispiel einer Segmentierung des binarisierten Sättigungskanals	10
5.4	Objektverfolgung durch Kriterium der lokalen Nähe	11
5.5	Objektverfolgung durch Größenkriterium	11
5.6	Objektverfolgung durch farbliche Ähnlichkeit	12
5.7	Stereokamerabasierte Entfernungsschätzung	14
5.8	Monokamerabasierte Entfernungsschätzung	14

1 Einleitung

Im Rahmen der Studienarbeit des fünften und sechsten Semesters der Prüfung zum Bachelor of Engineering, stellt diese Arbeit eine Dokumentation zur Entwicklung eines kameragestützten Roboters dar. Ziel der Arbeit ist es mit Hilfe eines Android-Smartphones und eines LEGO Mindstorm NXT-Kits einen Roboter zu entwerfen, der Gegenstände in einem Raum erkennt, anfährt und in eine vordefinierte Zielzone transportiert. Hierfür werden diverse Methoden der Bildverarbeitung eingesetzt, welche unter der Verwendung der OpenCV-Library [1] implementiert werden. Das Projekt wurde „CLEEN-R“ getauft. Der Name stellt hierbei ein Akronym für „Cleaning and Lifting Environment Evaluating NXT - Robot“ und stellt damit eine Anspielung an den Disney-Film „WALL-E“ dar

Das nachfolgende Kapitel beschreibt die Problemstellung und erklärt eine Grundproblematik der Zusammenarbeit der beiden Hardwaremodule. Kapitel drei beinhaltet genaue Daten zu den Hardwaremodulen und deren Zusammenspiel, sowie die notwendigen Grundlagen der Bildverarbeitung, die für spätere Methoden genutzt werden. Die beiden darauf folgenden Kapitel gehen gesondert auf den genauen Aufbau des Roboters samt Konstruktionsplänen, sowie Details zur Softwareimplementierung der Objekterkennung und der Roboteransteuerung ein. Kapitel 6 beschreibt durchgeführte Tests sowohl unter speziell präparierten Bedingungen, als auch Realbedingungen. Das abschließende Kapitel ist ein letztes Fazit, welches einen Überblick über die gesamte Arbeit bildet.

2 Problemstellung

Ziel dieser Studienarbeit ist die praktische Anwendung gelernter Kenntnisse in Hard- und Software. Durch die Benutzung zweier verschiedener getrennter Module sind sowohl Kenntnisse in der Programmierung, als auch der Prozessautomatisierung und in der Entwicklung verteilter Systeme erforderlich.

Konkrete Aufgabenstellung ist es einen kameragestützten Roboter zu entwickeln, der autonom Gegenstände in einem Raum erkennt und in definierte Zielzonen transportiert. Der Prozess kann hierbei in drei Teilprozesse unterteilt werden.

Erstens muss der Roboter mit Hilfe von bekannten Verfahren der Bildverarbeitung Objekte auf Grund ihrer physikalischen Beschaffenheit, beispielsweise ihrer Größe, ihrer Form und ihrer Farbe, erkennen und zielgerichtet anfahren.

Zweitens soll sich der Roboter gegenüber des Gegenstands optimal positionieren und diesen mit Hilfe eines mechanischen Greifarms aufnehmen.

Zuletzt muss der Roboter den aufgenommenen Gegenstand kategorisieren, nach der entsprechenden Zielzone suchen diese Anfahren und den Gegenstand ablegen. Die Erkennung der Zielzone kann dabei durch Markierungen an Wänden und Böden des Raumes erfolgen.

Als Bewertungskriterien dienen hierbei beispielsweise ob der Roboter alle Gegenstände erfolgreich erkennt, diese korrekt kategorisiert und in korrekte Zielzonen bewegt, sowie die Zeit in der dies geschieht.

3 Materialien und Methoden

Dieses Kapitel vielleicht löschen?

NXT-Roboter

Android-Smartphone

Bluetooth-Verbindung

Some more Stuff

4 Hardwareumsetzung

4.1 Entwurf des NXT-Roboters

[INSERT PLANS HERE]

4.1.1 Sensoren

4.1.2 Aktorik

4.2 Steuerung des Roboters

Bluetooth Verbindung zu Smartphone

4.3 Wahl des Kameramoduls

5 Softwareumsetzung

Mit Hilfe des in Kapitel 4.3 beschriebenen Kameramoduls müssen verschiedene Aufgaben aus dem Bereich der Bildverarbeitung bewältigt werden.

5.1 Wahl der Bildverarbeitungsbibliothek

Die Umsetzung der zu bewältigenden Aufgaben kann durch die Wahl einer geeigneten Bildverarbeitungsbibliothek deutlich vereinfacht werden. Wichtige Kriterien für die Wahl der Bibliothek sind unter anderem Funktionsumfang, Dokumentation und Aktivität der Community.

5.1.1 LibCCV

LibCCV [2] ist eine open-source Bildverarbeitungsbibliothek, die viele bekannte Algorithmen implementiert. LibCCV steht unter einer BSD-Clause-3-Lizenz und kann somit für eine Studienarbeit problemlos unbegrenzt verwendet werden. Die Bibliothek ist größtenteils in C++ verfasst und somit potenziell auf einem Android-Smartphone verwendet werden. Die Verwendung auf dem Smartphone wird jedoch nicht offiziell unterstützt und kann potenziell weitere Schwierigkeiten mit sich bringen.

5.1.2 Imagemagick

Bei Imagemagick [3] handelt es sich um eine Bildverarbeitungsbibliothek, welche sehr viele Algorithmen bereits implementiert hat. Algorithmen zur Objekterkennung müssten jedoch

vollständig selbst implementiert werden, was zu einem großen zusätzlichen Aufwand führen kann. Imagemagick wird unter der Apache 2.0 Lizenz vertrieben.

5.1.3 OpenCV

OpenCV [1, 4] stellt eine der größten Open-Source-Bibliotheken für Bildverarbeitung da. Die Bibliothek hat einen starken Fokus auf Echtzeitverarbeitung und wird daher auch in vielen Projekten im Bereich der Robotik verwendet. OpenCV hat eine große aktive Community, wodurch eventuelle Fragen und Probleme schnell beantwortet werden können. Zusätzlich bietet OpenCV eine offizielle Version für Android und eignet sich somit ideal für diese Studienarbeit.

5.2 Algorithmen zur Objekterkennung

Aufgabe des Roboters, ist es Gegenstände in einem Raum mit Hilfe von Kamerabildern zu erkennen. Folglich spielt die Objekterkennung eine große Rolle.

5.2.1 Farbbasierte Objekterkennung

Einen einfachen Ansatz der Objekterkennung unter Verwendung von Methoden der in 5.1.3 beschriebenen Bibliothek OpenCV stellt eine farbbasierte Objekterkennung dar. Hierfür wird das Kamerabild zunächst vom RGB-Format [ZITAT EINFÜGEN] in das HSV-Format [ZITAT] konvertiert. Dies wird durchgeführt, da das HSV-Format unempfindlicher gegen Veränderungen in der Beleuchtung ist als das RGB-Format. Abbildung 5.1 zeigt die Aufteilung eines Bildes in die verschiedenen Kanäle.

Wie in Abbildung 5.1 zu sehen ist, eignet sich vor allem der Saturation-Kanal des Bildes um farbige Objekte zu erkennen, da dieser hohe Werte annimmt wenn die Farbintensität hoch ist.

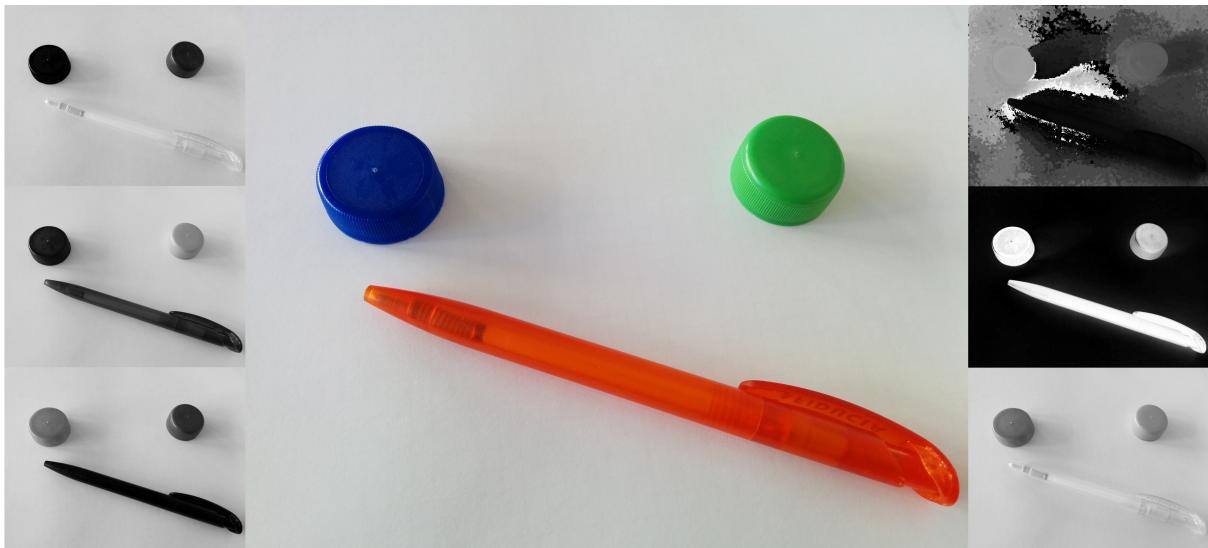


Abbildung 5.1: Beispiel einer Aufspaltung in RGB-Kanäle (links) und HSV-Kanäle (rechts)

Zuletzt erfolgt eine Binarisierung mit einem empirisch ermittelten Schwellwert von 50. Dies ist notwendig, da die, in der OpenCV-Bibliothek implementierte, Methode von Suzuki und Abe [5] zur Segmentierung von Objekten ein Binärbild erwartet. Aus dem in Abbildung 5.1 dargestellten Beispiel entsteht nach der Binarisierung schließlich Abbildung 5.2. Der Algorithmus zur Segmentierung erkennt daraus drei Objekte und zeichnet ihren Konturen, wie in Abbildung 5.3 ersichtlich, ein.

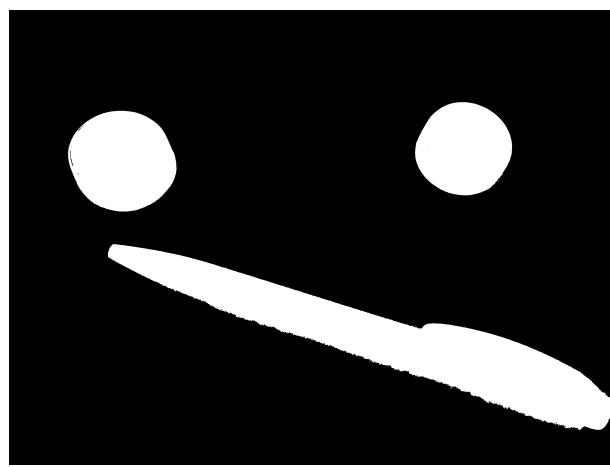


Abbildung 5.2: Beispiel der Binarisierung des Sättigungskanals

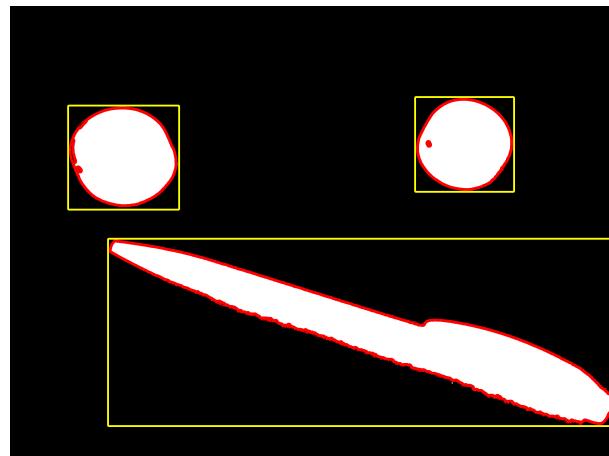


Abbildung 5.3: Beispiel einer Segmentierung des binarisierten Sättigungskanals

5.3 Objektverfolgung

Sobald ein Objekt durch die in Kapitel 5.2 dargestellten Algorithmen erkannt wurde, muss sichergestellt werden, dass die Fokussierung auf dieses Objekt beibehalten werden kann. Ein zufälliges Umspringen auf zufällige andere erkannte Objekte wäre für den Prozess sehr hinderlich. In den folgenden Abschnitten sind Kriterien und Probleme beschrieben, an Hand derer ein erkanntes Objekt in folgenden Bildern erkannt werden kann.

5.3.1 Ähnlichkeitskriterien

Um den Fokus auf ein Objekt zu behalten lassen sich verschiedene Ähnlichkeitskriterien formulieren durch die das erkannte Objekt in der Menge der im nachfolgenden Bild erkannten Objekte wieder gefunden werden kann.

Lokale Nähe

Das trivialste Kriterium stellt die lokale Nähe da. Geht man von einem Stillstand der Kamera und der aufgenommenen Szene aus, so befinden sich sämtliche Objekte in nachfolgenden Aufnahmen an der selben Stelle. Fokussierte Objekte können also rein aus ihrer Position wiedergefunden werden. Geschieht Bewegung, so kann nicht mehr von einer exakten Übereinstimmung der Koordinaten ausgegangen werden. Stattdessen muss eine gewisse Toleranz gegeben werden. Als Maß kann hierbei angenommen werden, dass sich die Position des Zentrums eines Objektes in fortlaufenden Aufnahmen um nicht mehr als beispielsweise 10% der Aufnahmegröße geändert hat. Abbildung 5.4 zeigt das beschriebene Verhalten anschaulich.

Abbildung 5.4: Objektverfolgung durch Kriterium der lokalen Nähe

Größenkriterium

Das Größenkriterium ähnelt zunächst dem Kriterium der lokalen Nähe, bezieht sich jedoch nicht auf den Mittelpunkt des Objekts, sondern auf die vom Objekt in der Aufnahme eingenommenen Fläche. Diese bleibt bei einer beliebigen Translation des Objektes in der Aufnahme konstant und eignet sich daher als Ähnlichkeitsmaß. Bewegt sich das Objekt jedoch auf die Kamera zu, oder von ihr weg, so verändert sich die eingenommene Fläche. Auch hier muss eine Toleranz gewählt werden. Diese kann entweder konstant gegeben sein (beispielsweise 10%) oder abhängig von der aktuellen Geschwindigkeit des Roboters gewählt werden. Abbildung 5.5 zeigt exemplarisch die Verfolgung eines Objektes dessen Größe sich im Verlauf der Aufnahmeserie verändert.

Abbildung 5.5: Objektverfolgung durch Größenkriterium

Farbliche Ähnlichkeit

Ein drittes Ähnlichkeitsmaß stellt die farbliche Ähnlichkeit dar. Diese ist Abhängig vom Farbraum in dem die Aufnahme getätigt oder konvertiert wurde. Wie in Abschnitt 5.2 beschrieben, wird die Aufnahme bei der Objekterkennung in das HSV-Format konvertiert. Daher kann das Ähnlichkeitsmaß sich auf den Farbkanal, den Sättigungskanal oder den Intensitätskanal beziehen. Durch außschließen des Intensitätskanals kann eine weitgehend helligkeitsunabhängige Bestimmung erreicht werden. Daher ist das Filtern nach Objekten ähnlicher Farbe und Sättigung durchaus sinnvoll. Auch hier lässt sich eine Toleranz formulieren, wie etwa 10% Abweichung von der durchschnittlichen Farbe des zuvor ermittelten Objektes. Abbildung 5.6 zeigt die farbliche Ähnlichkeit zweier Objekte und die dadurch resultierende Wiederfindung des Objektes in der Folgeaufnahme.

Abbildung 5.6: Objektverfolgung durch farbliche Ähnlichkeit

5.3.2 Kurzzeitiger Verlust des Verfolgungsziels

Tendenziell muss bei der Objektverfolgung immer von einem Verlust des verfolgten Objektes ausgegangen werden. Dies kann permanent sein, wie beispielsweise bei apprupter starker Veränderung der Lichtbedingungen, oder lediglich temporär durch Fehler bei der Bildaufnahme geschehen. In diesem Abschnitt wird auf letzteres eingegangen, da das Verhalten bei längerfristigem Verlust wird in Abschnitt 5.7 beschrieben wird.

Es hat sich gezeigt, dass kleine Fehler oder Ungenauigkeiten bei der Bildaufnahme bereits dazu führen können, dass ein Objekt nicht in der darauf folgenden Aufnahme wiedergefunden werden kann. Ist dies nur von kurzer Dauer, muss dies gesondert behandelt werden um zu vermeiden, dass der Roboter einen plötzlichen Wechsel des Verfolgungsziels durchführt. Ähnlich der in Abschnitt 5.3.1 beschriebenen Ähnlichkeitskriterien muss auch hier eine Toleranz für

eine maximale Anzahl Aufnahmen gelegt werden, in denen das verfolgte Ziel nicht gefunden werden muss ohne, dass das Objekt als "verlorenängesehen wird. Die Wahl einer zu hohen Toleranz hat die Folge, dass der Roboter lange Zeit in einem undefinierten Zustand ein Objekt verfolgt, welches er nicht sieht. Ist die Schwelle zu niedrig gewählt, so kann dies ein häufiges Springen zwischen verschiedenen Objekten zur Folge haben, was sich sehr negativ auf die Laufzeit auswirken könnte. Empirisch hat sich gezeigt, dass eine solche Schwelle bereits bei etwa drei Aufnahmen ohne das verfolgte Objekt liegen kann. Wird ein Objekt verloren, so begibt sich der Roboter zurück in den Zustand der Objektsuche.

5.3.3 Verschwommene Bilder durch Bewegung des Roboters

5.4 Entfernungsschätzung

Die Entfernungsschätzung ist für mehrere Teile der Roboterroutine wichtig. Zunächst muss der Roboter, der ein Objekt per Kamera erkannt hat anfahren und aufheben. Hierfür muss er wissen ob der aufzuhebende Gegenstand in Reichweite des Greifarms ist. Weiterhin benötigt der Roboter die Entfernungsschätzung bei der Navigation. Er muss Hindernisse, wie beispielsweise Wände eines Raumes, erkennen bevor er kollidiert und unter Umständen seine Fracht verliert.

5.4.1 Stereokamerabasiert

Ein häufiger Ansatz für die Entfernungsschätzung stellt die Stereokamerabasierte Methode dar. Hierbei nimmt der Roboter seine Umgebung nicht mit nur einer Kamera, sondern mit mehreren, mindestens zwei, wahr. Durch einen bekannten Versatz zwischen den Aufnahmequellen der Bilder, lässt sich mittels Registrierung auch ein Versatz in den erzeugten Bildern ermitteln. Wie die beiden menschlichen Augen, erlaubt dies das Wahrnehmen der Umgebung in dritter Dimension. Abbildung 5.7 zeigt exemplarisch eine solche Entfernungsschätzung.

Abbildung 5.7: Stereokamerabasierte Entfernungsschätzung

Der Stereokamerabasierte Ansatz ist von der Ausführung sehr elegant, da er sich an der Natur orientiert. Problematisch ist jedoch, dass zusätzliche Hardware benötigt wird. Eine weitere Kamera als bildgebendes Mittel birgt einen neuen Grad der Komplexität, da diese mit dem vorhandenen Smartphone synchronisiert werden muss. Zusätzlich müssen Rechenintensive Registrierungen vorgenommen werden, wodurch die Echtzeitfähigkeit des Systems beeinträchtigt werden könnte.

[HIER FEHLT NOCH EINE QUELLE]

5.4.2 Monokamerabasiert

Der Monokamerabasierte Ansatz stützt sich auf allen vorhandenen Informationen. Hierbei wird mittels der bekannten Position und Inklination der Kamera, sowie ihres Streuwinkels ein virtuelles Blickfeld erstellt. Dieses Blickfeld breitet sich von der Kamera aus pyramidenförmig aus und trifft den als planar angenommenen Boden. Abbildung 5.8 zeigt den Aufbau eines solchen Systems.

Abbildung 5.8: Monokamerabasierte Entfernungsschätzung

Anhand der Position des detektierten Objektes im Kamerabild lässt sich daraufhin eine ungefähre Entfernungsschätzung liefern. Problematisch ist hierbei die Ungenauigkeit des Systems. Sowohl Fehler bei der Aufnahme, als auch Probleme durch eine maximale Größenbeschränkung des detektierten Objektes können in der Praxis zu Fehleinschätzungen und damit fehlerhafter Benutzung des Greifarms führen.

[HIER FEHLT NOCH EINE QUELLE]

5.4.3 Ultraschallbasiert

Die Ultraschallbasierte Entfernungsschätzung kann auch über Große Distanzen von bis zu zwei Metern zentimetergenaue Entfernungen messen. Die Messung wird durch einen Echo-Mechanismus mit, für den Menschen unhörbaren, Schallwellen im Ultraschallbereich durchgeführt. [HIER FEHLT NOCH EINE QUELLE] Zunächst wird jedoch, ähnlich dem in 5.4.1 beschriebenen Verfahren, zusätzliche Hardware in Form eines Ultraschallsensors benötigt. Ein solcher kann jedoch direkt von LEGOTM erworben und problemlos am Roboter angebracht werden.

Das Ultraschallbasierte Verfahren kann durch eine kleine Änderung des Roboters realisiert werden und liefert sehr genaue Ergebnisse. Die Koppelung mit dem Smartphone ist bereits gegeben und genügt daher Echtzeitanforderungen.

5.5 Orientierung im Raum

5.5.1 Kameragestützt

5.5.2 Streckenbasiert

Über gefahrene Strecke

5.5.3 Kombination unterschiedlicher Sensordaten

5.6 Zielzonenbestimmung

Zielzonen werden über App vor Start festgelegt.

5.7 Hauptschleife

Arbeitszustände

1. Objekt suchen
2. Objekt ansteuern
3. Objekt aufnehmen
4. Objekt kategorisieren
5. Zielbereich suchen
6. Zielbereich ansteuern
7. Objekt ablegen

[INSERT ZUSTANDSÜBERGANGSDIAGRAMM] Petrinetz?

6 Tests des Robotersystems

6.1 Tests im gesicherten Rahmen

6.2 Realtests

7 Zusammenfassung und Ausblick

Literaturverzeichnis

- [1] G. Bradski. Open cv. *Dr. Dobb's Journal of Software Tools*, 2000.
- [2] Libccv.
- [3] Imagemagick.
- [4] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. Ö'Reilly Media, Inc.", 2008.
- [5] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.