

Konzepte zur Auslegung von Echtzeit-Bildverarbeitungssystemen für die Qualitätssicherung am Beispiel der Inspektion von Texturen

Herrn Prof. Dr. rer. nat. Dipl.-Ing. Henning Tolle zum 65. Geburtstag gewidmet

Georg Lambert, Jörg Amelung und Enis Ersü, Darmstadt



Dipl.-Ing. Georg Lambert ist wissenschaftlicher Mitarbeiter am Institut für Regelungstechnik der Technischen Hochschule Darmstadt, FG Regelsystemtheorie & Robotik (Leiter: Prof. Dr. rer. nat. Dipl.-Ing. Henning Tolle). Er beschäftigt sich mit der Untersuchung und Entwicklung von Algorithmen zur Oberflächeninspektion in Echtzeit, konturbasierter Bildverarbeitung, Verfahren zur Texturanalyse sowie Architekturen von Echtzeit-Softwaresystemen zur Bildverarbeitung.

Adresse: TH Darmstadt, Institut für Regelungstechnik, FG Regelsystemtheorie & Robotik, Landgraf-Georg-Straße 4, D-64283 Darmstadt, Tel.: (06151) 16-3542, Fax: (06151) 16-2507, E-Mail: lamb@rt.e-technik.th-darmstadt.de



Dr.-Ing. Jörg Amelung ist Projektleiter im Bereich Oberflächeninspektionssysteme bei der Firma ISRA Systemtechnik GmbH, Darmstadt. Hauptarbeitsfelder: industrielle Umsetzung von Texturanalyseverfahren, Klassifikatorauslegung und Design von Bildverarbeitungssystemen. Davor war er wissenschaftlicher Mitarbeiter am Institut für Regelungstechnik der Technischen Hochschule Darmstadt, FG Regelsystemtheorie & Robotik (Leiter: Prof. Dr. rer. nat. Dipl.-Ing. Henning Tolle). Aus dieser Zeit stammen Teile der in diesem Beitrag vorgestellten Untersuchungen und Ergebnisse.

Adresse: ISRA Systemtechnik GmbH, Industriestraße 14, D-64297 Darmstadt, Tel.: (06151) 948-0, Fax: (06151) 948-140



Dipl.-Ing. Enis Ersü ist Gründer und Geschäftsführer der ISRA Systemtechnik GmbH, Darmstadt, und Mitglied des Vorstandes der CRS Robotics Corporation, Kanada. In seiner Eigenschaft als Chief Operating Officer leitet er alle Unternehmenseinheiten in Burlington, Darmstadt, Schweden und Frankreich. Vor Gründung der ISRA Systemtechnik GmbH war er wissenschaftlicher Mitarbeiter am Institut für Regelungstechnik der Technischen Hochschule Darmstadt, FG Regelsystemtheorie & Robotik (Leiter: Prof. Dr. rer. nat. Dipl.-Ing. Henning Tolle).

Adresse: ISRA Systemtechnik GmbH, Industriestraße 14, D-64297 Darmstadt, Tel.: (06151) 948-0, Fax: (06151) 948-140

Bildverarbeitungssysteme werden in zunehmendem Maße für Aufgaben der Automatisierungstechnik und im besonderen in der Qualitätssicherung erfolgreich eingesetzt. Zur Lösung komplexer oder zeitkritischer Erkennungsaufgaben, wie z.B. in der Oberflächeninspektion und Texturanalyse, ist der Einsatz rechenintensiver Algorithmen erforderlich, so daß eine Softwarerealisierung nicht oder nur in speziellen Fällen möglich ist. Der vorliegende Beitrag zeigt Ansatzpunkte

und Methoden zur Auslegung von Softwaresystemen zur Qualitätssicherung in Echtzeit am Beispiel der Inspektion texturierter Oberflächen auf. Es werden merkmalsbasierte Verfahren diskutiert, wobei die ganzheitliche und zielorientierte Berücksichtigung aller Verfahrensschritte wesentlich ist. Zur Unterstützung des Designs von Bildverarbeitungslösungen werden automatische Verfahren der Systemkonfigurierung vorgestellt.

Concepts in designing real time machine vision systems for quality control

Machine vision systems are more and more successfully applied in the field of automation and specially for the purpose of quality inspection. To solve complex recognition tasks like surface inspection or texture defect analysis, the application of time consuming algorithms is required which makes software solutions work in only some specific cases. In this article ways and methods for implementing software systems for quality control in real time are pointed out and motivated by applications on texture inspection. Feature based systems are discussed. In this context it is essential to consider the overall system in a problem oriented way and not to focus on single aspects. To support the design of machine vision solutions algorithms for automatic system configuration are presented.

1 Einleitung

Die automatische Auswertung von Bilddaten besitzt ein hohes Potential für den Einsatz in der Automatisierungstechnik. Sowohl qualitative Aussagen, wie z.B. „Bauteil vorhanden“ (Vollständigkeitskontrolle) als auch quantitative Aussagen, wie Meßergebnisse, können von Bildverarbeitungssystemen in Sekundenbruchteilen getroffen werden. Das Spektrum des industriellen Einsatzes von Bildverarbeitungssystemen reicht von Kontrollaufgaben, 2D Meßaufgaben, 3D Meßaufgaben, Objekterkennung über Codeerkennung

(z.B. Barcodes) und Schrifterkennung bis zur Beurteilung und Inspektion von Oberflächen. Kennzeichnend für den Einsatz von Bildverarbeitungssystemen ist die Berührungsfreiheit des Verfahrens, wodurch der Einsatz am laufenden Prozeß möglich ist. Auch der nachträgliche Einsatz von Bildverarbeitungssystemen an Fertigungslinien ist in den meisten Fällen ohne wesentliche Umbaumaßnahmen leicht möglich.

Von einem echtzeitfähigen System wird immer dann gesprochen, wenn Zeitanforderungen des Prozesses, in den die Bildverarbeitung eingebettet ist, erfüllt werden können. Diese Zeitanforderungen können je nach Fertigungsprozeß und zu inspizierender Einheit zwischen $\frac{1}{\text{min}}$ z.B. bei der Vermessung eines komplexen Bauteils bis $500 \frac{\text{m}}{\text{min}}$ und mehr bei schnellaufendem

Bahnmaterial sein. Entsprechend ist die Komplexität der implementierten Algorithmik für die verschiedenen Aufgaben zur Qualitätssicherung stark unterschiedlich.

Traditionelle Einsatzgebiete der Bildverarbeitung sind Aufgaben quantitativer Natur, die als „maschinenfreundlich“ bezeichnet werden können. Hierzu zählen z.B. Vermessungsaufgaben, Vergleichsaufgaben oder auch das Lesen von Codes, die speziell für Maschinen entwickelt wurden. In den letzten Jahren wurden Bildverarbeitungssysteme vermehrt auch in Bereichen eingesetzt, wo sie die visuelle Inspektion bzw. Begutachtung komplexer Szenen durch den Menschen substituieren. Dies wurde durch große Fortschritte in der Forschung auf den Gebieten der Mustererkennung, der Signalanalyse und der künstlichen Intelligenz sowie der stetig steigenden Rechenleistung erzielt. Das Sehsystem des Menschen besitzt die bisweilen nicht nachzuahmende Fähigkeit, komplexe Szenen auf bestimmte Merkmale hin sehr schnell beurteilen zu können (z.B. [1; 2]). Diese Fähigkeit kann ausgebaut bzw. trainiert werden, um auf spezielle Merkmale zu reagieren. In diesem Kontext ist die visuelle Inspektion von komplexen Bauteilen oder texturierten Oberflächen durch den Menschen zu sehen. Beispiele sind:

- Warenschau (visuelle Inspektion von Bahnmaterial, insbesondere Stoffen),
- Parkettsortierung,
- Waverinspektion (Halbleiterstrukturen),
- Inspektion spanend bearbeiteter Metalloberflächen usw.

Für die Automatisierung der genannten Aufgaben durch Bildverarbeitungssysteme müssen die Beurteilungskriterien in Algorithmen gefaßt werden. Da in Bereichen wie der Inspektion von Teppichen oder Stoffen die Qualität auch durch die menschliche Empfindung bestimmt wird, müssen für das Design von automatischen Systemen entweder physiologische Untersuchungen herangezogen werden oder eine ausreichend große Stichprobe klassifizierter Muster vorliegen. Der Transfer zwischen Mensch und Maschine bzw. Algorithmik ist hier die wesentliche Ingenieuraufgabe.

Eine wichtige Vorarbeit beim Systemdesign einer Bildverarbeitungsarchitektur zur Qualitätssicherung ist

die Bereitstellung von Mustern. Nur dadurch kann vom Systemanbieter die Wirksamkeit der Trennlinie zwischen gut und fehlerhaft durch das Training vor Klassifikatoren bzw. das Erlernen von Modellen garantiert werden.

Der vorliegende Beitrag diskutiert das Design von Bildverarbeitungssystemen zur Qualitätssicherung in Echtzeit durch den Einsatz moderner Konzepte. Der Schwerpunkt liegt hierbei auf der integrativen Optimierung aller Bestandteile der Bildverarbeitungskette, von der Bildaufnahme bis zur Entscheidungsformulierung. Es soll dabei weniger auf starre Speziallösungen, die in vielen Fällen auf dedizierter Hardware aufbauen, als vielmehr auf flexible, echtzeitfähige Softwaresysteme eingegangen werden, die mit einem Minimum an spezieller Bildverarbeitungshardware auskommen. Die Aussagen werden an Beispielen aus der Inspektion von Texturen motiviert und konkretisiert (siehe z.B. [3; 4]). Es werden Mechanismen zur automatischen Konfiguration von Bildverarbeitungssystemen vorgestellt. Diese sollen den Ingenieur beim Design einer speziellen Problemlösung unterstützen, ihn jedoch nicht seiner intuitiven Entscheidungsautorität entheben. Ein neues Maß zur nichtparametrischen Beurteilung von Merkmalsüberlappungen wird vorgestellt und die Anwendung zur Merkmalsselektion aufgezeigt. Der Artikel geht nicht auf wissensbasierte Qualitätssicherungssysteme oder Expertensysteme ein (siehe hierzu z.B. [5; 6]). Vielmehr werden merkmalsbasierte statistisch ausgewertete Verfahren diskutiert.

2 Optimierung der Bildverarbeitungskette unter Echtzeitanforderungen

Ein Bildverarbeitungssystem besteht im allgemeinen aus einer komplexen Kombination von Verarbeitungsblöcken. In den meisten Fällen lassen sich jedoch die wesentlichen und vor allem zeitkritischen Verarbeitungsschritte durch die Bildverarbeitungskette in Form einer Pipeline darstellen (vergleiche Bild 1). Die Verarbeitungsblöcke aus Bild 1 können auf verschiedene Weise realisiert sein. Die Bilderfassung stellt im allgemeinen eine Hardwareeinheit dar. Alle anderen Verarbeitungsblöcke können sowohl Software- als auch Hardwaresysteme bzw. hybride Systeme sein. Bei der etablierten Bildverarbeitungssystemen auf VME-Bus-Basis ist der Aufbau der Pipeline aus Hardwarebausteinen durch spezielle, vornehmlich serielle Video-Bus-Systeme möglich. Bei modernen Framegrabbern für den PC, bei denen die schnelle Datenübertragung in der Hauptspeicher des Rechners über den PCI-Bus erfolgt sind die Verarbeitungsblöcke der Bildverarbeitungskette meist durch Software realisiert.

Unabhängig von der Realisierung der Struktur aus Bild 1, ist es beim Design von Echtzeitsystemen wesentlich, die einzelnen Komponenten aufeinander abzustimmen. Insbesondere bei der Inspektion von Bahnmaterial mit texturierter Oberfläche sind die

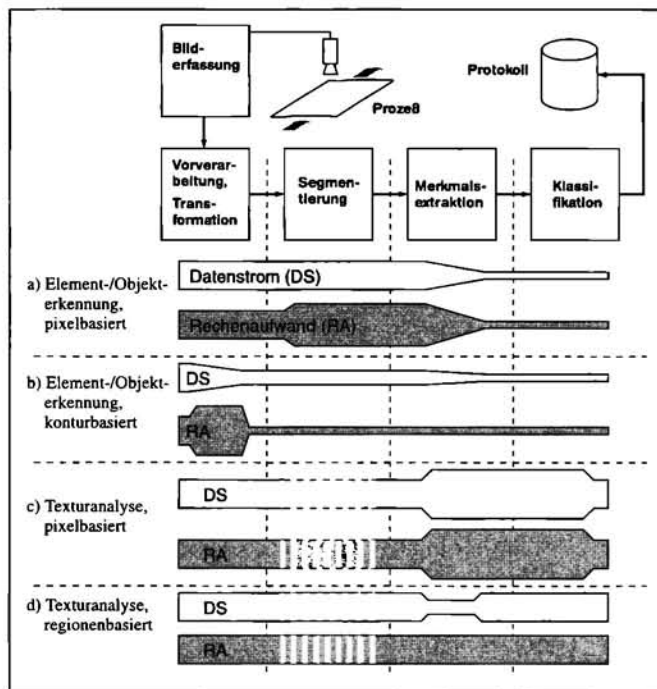


Bild 1: Die Bildverarbeitungspipeline mit Beispielen zur Verteilung von Datenstrom und Rechenaufwand.

Zeitanforderungen an moderne Rechensysteme enorm und erfordern daher die Optimierung der Bildverarbeitungskette.

In Bild 1 sind dazu einige Beispiele mit einer Darstellung des typischen Verlaufs von Datenstrom und Rechenaufwand entlang der Pipeline gegeben, um das Zusammenspiel der Verarbeitungseinheiten kritisch zu erläutern.

a) **Element-/Objekterkennung, pixelbasiert:** Von der Bilderfassung bis zur Merkmalsextraktion umfaßt der Datenstrom alle Bildpunkte und bleibt konstant. Die Verdichtung der Bildinformation findet bei der Segmentierung von Elementen bzw. Objekten statt, was sich im Rechenaufwand bemerkbar macht. Das Ergebnis der Segmentierung sind daher typischerweise Bildpunkte, die verschiedene Objektlabel tragen. Erst bei der Merkmalsextraktion nimmt der Datenstrom ab, da nur noch objektspezifische Merkmalsvektoren zur Klassifikation weitergegeben werden. Der Rechenaufwand ist hier dementsprechend gering.

b) **Element-/Objekterkennung, konturbasiert:** Bei geeigneten Inspektionsproblemen führt die Extraktion von Konturen in einem frühen Verarbeitungsschritt zu einer erheblichen Reduktion des Datenstromes im gesamten Rest der Pipeline. Der größte Teil des Rechenaufwands fällt folglich bei der Vorverarbeitung an. Bei geeigneter Algorithmik, die auf den Konturdaten weiterarbeiten kann, ist eine schnelle Segmentierung, Merkmalsextraktion und Klassifikation möglich, ohne dabei nachträglich den Datenstrom zu erhöhen. Der entsprechend geringe Rechenaufwand darf jedoch nicht über die Kompliziertheit der erforderlichen, speziellen Algorithmik hinwegtäuschen.

c) **Texturanalyse, pixelbasiert:** Unter pixelbasierter Texturanalyse soll verstanden werden, daß jedem Bildpunkt Textureigenschaften in Form eines Merkmalsvektors zugeordnet werden. Dies hat zur Folge, daß bei der Merkmalsextraktion sowohl Datenstrom wie Rechenaufwand erheblich zunehmen. Erst bei der Klassifikation nimmt der Datenstrom wieder ab, da hier jedem Pixel eine skalare Texturklasse zugeordnet wird. Im Verlauf des meist zweistufigen Klassifikationsvorgangs werden in einem zweiten Schritt aus benachbarten Pixeln gleicher Klassen fehlerhafte oder fehlerfreie Regionen gebildet. Die Segmentierung fällt bei der Texturanalyse im allgemeinen weg.

d) **Texturanalyse, regionenbasiert:** Unter regionenbasierter Texturanalyse soll verstanden werden, daß aus mehreren Bildpunkten bestehenden Regionen bzw. Bildfenstern Textureigenschaften in Form von Merkmalsvektoren zugeordnet werden. Die Rasterung der ortsbezogenen Texturinformation wird daher gröber (Reduktion des Datenstromes), die Informationsmenge je Rasterpunkt nimmt jedoch durch die vektorwertigen Merkmale zu. Je nach Wahl von Regionengröße, Schrittweite der Regionenverschiebung und Dimensionalität der Merkmale, wird der Datenstrom vergrößert oder reduziert. Entsprechendes gilt für den Rechenaufwand. Eine Segmentierung wird im allgemeinen nicht durchgeführt.

Die Darstellung aus Bild 1 soll prinzipielle Ansatzpunkte zur Optimierung von Echtzeit-Bildverarbeitungssystemen geben. Bei sequentiellen Softwaresystemen, die mehrere Verarbeitungsschritte realisieren müssen, ist die Fläche des Rechenaufwandbalkens (RA) maßgebend. Bei Hardwarerealisierungen ist die Breite des Datenstrombalkens (DS) sowie die des Rechenaufwands ein Maß für die erforderliche Komplexität und Leistung der Baugruppe, die in der Pipeline arbeiten soll. Der Takt bzw. die Geschwindigkeit der Pipeline wird durch zwei Faktoren bestimmt:

1. Vorgabe des Eingangspixeltaktes, im allgemeinen durch bestehende Videonormen, oder
2. Vorgabe der Anzahl zu klassifizierender Einheiten pro Zeit z.B. durch den Kunden. Einheiten können hierbei sowohl einzelne Bildpunkte, wie Objekte sein.

Wie aus Beispiel b) in Bild 1 ersichtlich, ist die frühzeitige Extraktion von Konturen eine effektive Maßnahme, um Datenstrom und Rechenaufwand erheblich zu reduzieren. In [7] und [8] werden grundlegende Algorithmen zur Konturextraktion und konturbasierten Bildverarbeitung vorgestellt. Zur wissensbasierten Objekterkennung werden in [8] auf Konturinformationen basierende Ansätze aufgezeigt. Merkmalsbasierte Algorithmen werden in [9] und [10] vorgestellt, die auf den durch Polygone approximierten Bildkonturen arbeiten und eine schnelle Fehlererkennung in Texturen aus Texeln (Texturelementen) realisieren. Die Integration von Systemflexibilität und

Echtzeitfähigkeit wird durch die Hardwarerealisierung der Konturextraktion und dem sich daran anschließenden Softwaresystem erzielt.

Beispiel c) aus Bild 1 zeigt die ungünstige Auslegung eines Texturanalysesystems. Der Grund für die hohe erforderliche Systemleistung liegt an der Zuordnung vektorwertiger Texturmerkmale zu jedem Bildpunkt. Ein Beispiel hierfür ist die Anwendung richtungs- und frequenzselektiver Gaborfilter (siehe z.B. [11; 12]) zur Berechnung von Texturmerkmalen. Zur Erörterung der Frage nach der Notwendigkeit eines solchen Vorgehens soll der Texturbegriff genauer betrachtet werden:

- Textur ist eine Eigenschaft, die durch die Nachbarschaft mehrerer Bildpunkte erst entsteht.
- Das Zuordnen von Textureigenschaften zu einem einzelnen Bildpunkt ist möglich, die Eigenschaften sind jedoch nur durch dessen Nachbarschaft definierbar.
- Das Vorliegen eines isolierten Bildpunktes der Texturklasse \mathcal{A} mit einer Nachbarschaft aus Bildpunkten der Texturklasse \mathcal{B} ist daher im allgemeinen nicht detektierbar.

Die Konsequenz obiger Aussagen ist, daß die Detektion von Fehlern in einer texturierten Oberfläche bzw. die Segmentierung unterschiedlich texturierter Regionen, eine Mindestgröße der Regionen erfordert. Im Rahmen der Optimierung von Oberflächeninspektionssystemen ist daher eine regionenorientierte Texturanalyse (vergleiche Beispiel d) aus Bild 1) sinnvoll. Eine genaue Auslegung der Fenstergrößen und Schrittweiten der Fensterverschiebung sowie der Dimensionalität der Texturmerkmale muß problemspezifisch beim Design des Texturanalysesystems berücksichtigt werden. In Abschnitt 3 werden Verfahren der automatischen Konfigurierung von Texturanalysesystemen vorgestellt, die den Systemingenieur bei der Auslegung unterstützen.

2.1 Abstimmung von Merkmalen und Klassifikatoren

Es soll im folgenden auf die Optimierung des Zusammenspiels der Blöcke Merkmalsextraktion und Klassifikator aus der Bildverarbeitungs pipeline eingegangen werden. Die Leistung eines Mustererkennungssystems ist wesentlich von der Abstimmung dieser beiden Komponenten abhängig. Im Rahmen der Echtzeitanwendung ist die Komplexitätsminimierung bei Wahrung der erforderlichen Erkennungsleistung anzustreben.

Eine altbekannte Tatsache ist, daß auch der beste Klassifikator bei nicht signifikanten Merkmalen schlechte Ergebnisse liefern muß. Die Funktionsweise eines Erkennungssystems beruht auf der Tatsache, daß zu unterscheidende Klassen zu prinzipiell separierbaren Merkmalsverteilungen führen. Bild 2 zeigt unterschiedliche Situationen der Merkmalsverteilung im Merkmalsraum.

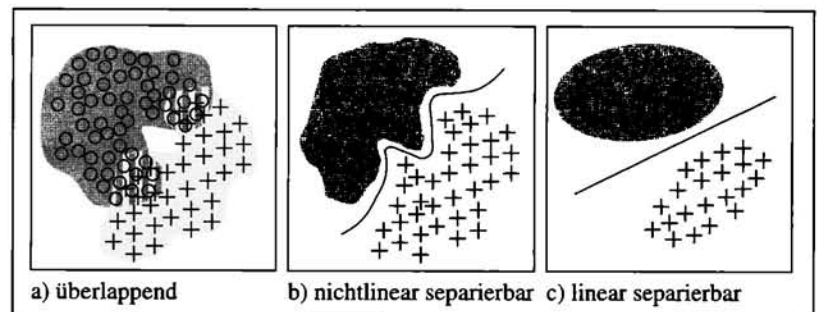
Im Fall überlappender Merkmalsverteilungen ist eine fehlerfreie Klassifikation nicht möglich. Wenn die geforderte Erkennungsleistung daher prinzipiell nicht erbracht werden kann, müssen die Merkmale verbessert werden. Im Fall separierbarer Merkmalsverteilungen entscheidet die Komplexität der Klassentrennlinie über die erforderlichen Fähigkeiten des anschließenden Klassifikators. Bei nichtlinearen Klassentrennlinien und klassenspezifischen Verteilungsdichtefunktionen $f_{\omega}(m)$ der Merkmale m , die keinen parametrischen Verteilungen zuzuordnen sind, ist der Einsatz nichtparametrischer Klassifikatoren bzw. universeller Funktionsapproximatoren zur Erzielung der maximalen Erkennungsleistung erforderlich. (Als parametrische Klassifikatoren werden solche bezeichnet, die von bekannten, mathematisch darstellbaren Verteilungsdichtefunktionen, z.B. normalverteilter Merkmale, ausgehen. Nichtparametrische Klassifikatoren können beliebige Merkmalsverteilungen modellieren. Siehe dazu z.B. [13; 14; 15].) Beispiele nichtparametrischer Klassifikatoren sind:

- KNN (K-nächste-Nachbarn) Klassifikatoren,
- BP (Back-propagation) Netzwerke,
- Kohonen-Netze mit Ausgangsschicht,
- RBF (Radial-Basis-Funktions-) Netzwerke,
- Fuzzy Klassifikatoren.

Im Fall einfach separierbarer Merkmalsverteilungen (Bild 2c) ist der Einsatz parametrischer Klassifikatoren mit niedriger Komplexität ausreichend und sinnvoll. Unter der Annahme normalverteilter Merkmale ist dies im allgemeinen der Bayes Normalverteilungsklassifikator oder andere aus diesem abgeleitete Klassifikatoren (Maximum-Likelihood Klassifikator, Mahalanobis-Distanz Klassifikator). Bei radialsymmetrisch um die Klassenzentren μ_k verteilten Merkmalen oder entsprechend großem Klassenabstand genügt der Einsatz eines Minimum-Distanz Klassifikators, der die Klassen lediglich durch deren Zentren μ_k modelliert.

Wesentlich bei der Auslegung eines merkmalsbasierten Erkennungssystems ist die ganzheitliche Berücksichtigung aller Verfahrensschritte. Die Fokussierung auf Einzelaspekte (z.B. ausschließliche Optimierung des Klassifikators) führt im allgemeinen nicht zum optimalen System. Dies soll durch folgenden Grundsatz hervorgehoben werden:

Bild 2: Unterschiedliche Verteilung von Merkmalen im Merkmalsraum als Ausgangspunkt zur Klassifikation.



Die Eignung eines Verfahrens bezüglich eines untersuchten Problemkreises spiegelt sich in der Güte der verfahrensspezifischen Merkmale in Verbindung mit der erforderlichen Komplexität des anschließenden Klassifikators wider.

Bild 3 zeigt mögliche Verteilungen des Rechenaufwands zwischen der Merkmalsextraktion und der Klassifikation. Wenn die Auswahl an prinzipiell geeigneten Verfahren für ein vorliegendes Erkennungsproblem es zuläßt, kann zwischen Verfahrensaufwand und Klassifikationsaufwand abgewogen werden. Bei einfachen, schnellen Analyseverfahren kann der Einsatz eines komplexen Klassifikators erforderlich sein (Bild 3b), wogegen komplexe Analysemethoden den Einsatz einfacher Klassifikatoren erlauben (Bild 3c).

Das ideale Merkmalsextraktionsverfahren leistet eine Abbildung aller Instanzen i_k (z.B. Objekte oder Regionen) der Klasse ω_k auf Punkte P_k im \mathbb{R}^n bei n -dimensionalen Merkmalen. Damit die Klassen ω_k und ω_l unterscheidbar sind, muß $P_k \neq P_l$ gelten. In diesem Fall muß der Klassifikator die Orte P_k lediglich mit der Bedeutung der Klassen belegen.

Sei also

$$\Omega = \{\omega_k\}, \quad k = 1 \dots K \tag{1}$$

die Menge aller zu unterscheidenden Klassen und

$$I_k = \{i_k | i_k \in \omega_k\} \tag{2}$$

die Menge aller Instanzen i_k oder Ausprägungen der Klasse ω_k , dann ist die ideale Merkmalsextraktion

$$IME: I_k \rightarrow \mathbb{R}^n \tag{3}$$

$$i_k \mapsto P_k, \quad P_k \neq P_l \text{ für } k \neq l.$$

Bild 4 zeigt ein Beispiel aus der Objekterkennung. Bei Merkmalsextraktionsverfahren, die analytisch so hergeleitet sind, daß Gl. (3) erfüllt ist, spricht man von invarianten Merkmalen. Durch stochastische Schwankungen bei den Instanzen i_k sind die Invarianten im allgemeinen als normalverteilt anzunehmen, was in Bild 4 durch die graue Hinterlegung der Punkte im Merkmalsraum angedeutet ist.

Bei Methoden der Merkmalsextraktion, die bezüglich des betrachteten Problemkreises Invarianten liefern, ist der Einsatz von parametrischen Klassifikatoren im allgemeinen ausreichend. Die Annahme von normalverteilten Merkmalen ist in den meisten Fällen gerechtfertigt.

Bei komplexen Erkennungsproblemen, die eine Herleitung von Invarianten nicht zulassen oder bei stark variierenden Instanzen innerhalb der verschiedenen Klassen, ist die Verteilung der Merkmale als nichtparametrisch anzunehmen. In diesem Fall ist der Einsatz komplexer Klassifikatoren bzw. universeller Funktionsapproximatoren zur Klassifikation angebracht.

2.2 Der Überlappungsgrad von Merkmalsverteilungen

Da die Überlappung der Merkmalsverteilungen zu unterscheidender Klassen, wie in Bild 2a dargestellt, zu unvermeidbaren Fehlklassifikationen führt, wird im

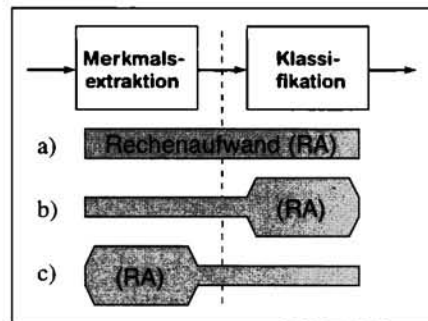


Bild 3: Möglichkeiten der Verteilung des Rechenaufwands zwischen Merkmalsextraktion und Klassifikation.

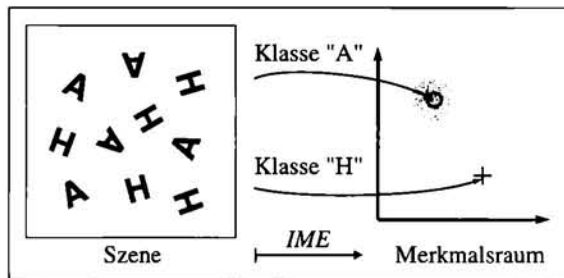


Bild 4: Ideale Merkmalsextraktion (IME) liefert klassenbezogene Invarianten.

folgenden ein neues Maß zur Bewertung von Merkmalsüberlappungen eingeführt. Der Überlappungsgrad q beurteilt folgende Problemstellungen bei beliebigen, nichtparametrischen Merkmalsverteilungen:

1. Auswahl zwischen merkmalsgebenden Verfahren (z.B. Texturanalyseverfahren), die auf Eignung zur Lösung eines Erkennungsproblems zu untersuchen sind.
2. Auswahl der optimalen Teilmenge aus der Gesamtmenge von Merkmalen eines merkmalsgebenden Verfahrens (Abschnitt 3.3, Merkmalsselektionsverfahren, siehe z.B. [14]).

Es gilt (vergleiche Bild 5):

$$q = \sqrt{\frac{N}{\sum_{j=1}^N \left(\frac{1}{q_j}\right)^\beta}}, \quad q_j = \frac{\sum_{i=1}^k q_{ji} + k}{\sum_{i=1}^k n_{ji} + k} \tag{4}$$

$$n_{ji} = 1 - \left(\frac{d_{NN_{ji}}}{d_{NN_{jk}}}\right)^\alpha, \quad q_{ji} = \begin{cases} n_{ji}, & \omega_j = \omega_i \\ -n_{ji}, & \omega_j \neq \omega_i \end{cases} \tag{5}$$

- N : Größe der Stichprobe,
- k : Anzahl der nächsten Nachbarn,
- $d_{NN_{ji}}$: euklidische Distanz zwischen Datenpunkt v_j und seinem i -ten nächsten Nachbarn v_{ji} ,
- $d_{NN_{jk}}$: euklidische Distanz zwischen Datenpunkt v_j und dem weitesten seiner k Nachbarn v_{jk} ,
- ω_j : Klasse des Datenpunktes v_j ,
- ω_i : Klasse des i -ten Nachbarn v_{ji} .

Für den Überlappungsgrad q gilt $0 \leq q \leq 1$, wobei $q = 1$ die höchste Güte darstellt. Das Maß q ergibt sich als das verallgemeinerte harmonische Mittel (Gl. (4))

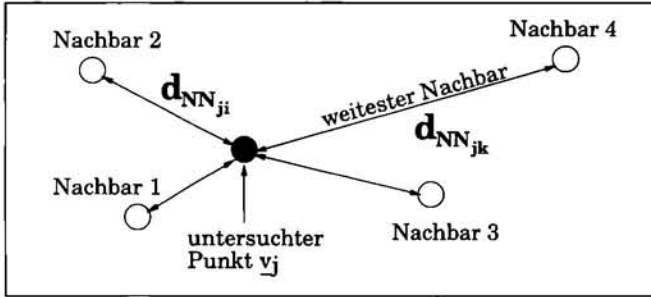


Bild 5: Erläuterung zum Überlappungsmaß q .

links) der lokalen Überlappungsgrade q_j . Diese werden für jeden Datenpunkt v_j bestimmt, wobei das Auftreten von Nachbarn anderer Klassen als ω_j mit einer Verringerung der Teilgüte q_j bestraft wird. Die Verwendung des harmonischen Mittels im Gegensatz zum arithmetischen Mittel führt dazu, daß schlechte lokale Überlappungsgrade q_j durch den Rest der Datenpunkte nicht weggemittelt werden. (Das verallgemeinerte harmonische Mittel hat für alle β die Eigenschaft, daß sich als Mittelwert von N Zahlen des gleichen Wertes x wiederum x ergibt.) Durch den Parameter β wird die Selektivität des Überlappungsgrades q bezüglich überlappender Regionen eingestellt. Bei großem β (z.B. $\beta = 4 \dots 8$) wird schon bei geringen Merkmalsüberlappungen der Wert von q merklich verkleinert. Durch den Parameter α kann die Entfernung von Datenpunkten gleicher bzw. unterschiedlicher Klassen innerhalb der k Nachbarn unterschiedlich bewertet werden. Der Überlappungsgrad q aus Gl. (4) stellt eine Erweiterung des in [16] hergeleiteten Maßes dar. Da sich q aus euklidischen Distanzen im Merkmalsraum berechnet, ist darauf zu achten, daß die Merkmale alle von gleicher Größenordnung sind. Gegebenenfalls müssen die Merkmale in einem vorverarbeitenden Schritt normiert werden.

3 Automatische Verfahrenskonfigurierung

Da der Bedarf an Rechenleistung bei der Auslegung von Echtzeit-Bildverarbeitungssystemen auch in heutiger Zeit extrem hoch ist und häufig an der Grenze des Realisierbaren liegt, ist die Optimierung der Verfahrensschritte und deren Zusammenspiel wesentlich. Vollautomatische Methoden zur Durchführung des Systemdesigns (Expertensysteme) sind im allgemeinen nicht einsetzbar, da die speziellen Echtzeit-Verfahren eine Optimierung auf unteren Ebenen verlangen und neue Algorithmen für neue Anwendungsprobleme entwickelt werden müssen. Zur Erfüllung der Echtzeitanforderungen können daher im allgemeinen keine Standardmodule mit Standardschnittstellen eingesetzt werden.

Trotzdem bleibt ein großes Potential für die Unterstützung des Systemdesigns durch den Rechner. Es handelt sich hierbei um Such- und Optimierungsver-

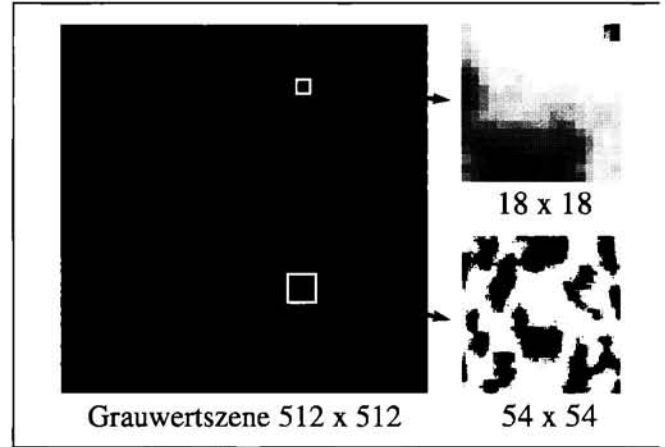


Bild 6: Inhalte von Fenstern der Größe 18×18 und 54×54 Bildpunkte einer Textur mit Fehlerstelle.

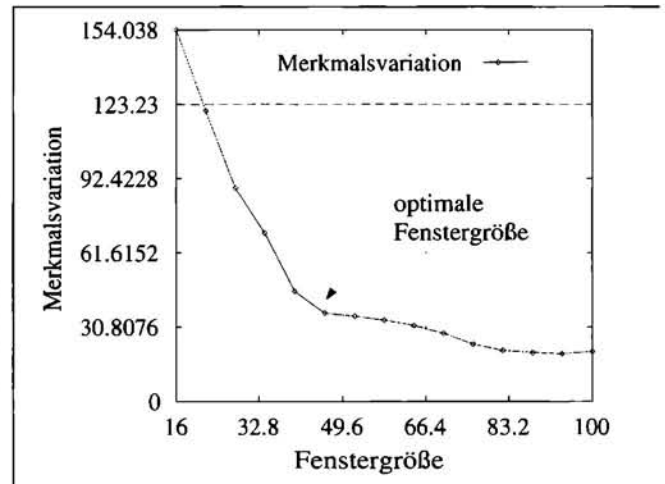
fahren auf mittleren und unteren Designebenen. Es wird auf die folgenden Methoden zur automatischen Verfahrenskonfigurierung eingegangen:

1. Fenstergrößeneinstellung bei Texturanalyseverfahren.
2. Bereichs- und Parametereinstellungen am Beispiel von Grauwert- und Lauflängenbereichen bei Texturanalyseverfahren.
3. Verfahrensspezifische Merkmalsselektion.
4. Automatische Klassifizierung von Stichproben.

3.1 Fenstergrößeneinstellung bei Texturanalyseverfahren

Wie in Abschnitt 2 erläutert, ist Textur durch die Ausprägung der Bildpunkte in Nachbarschaften charakterisiert. Texturmerkmale werden daher im allgemeinen in lokalen Fenstern berechnet. Etwaige Fehler auf einer texturierten Oberfläche ergeben sich als signifikante Abweichungen der Merkmale zum fehlerfreien Fall. Bild 6 zeigt die Inhalte von Fenstern der Größe 18×18 und 54×54 Bildpunkte einer Textur mit Fehlerstelle. Im Falle des kleinen Fensters können die Textureigenschaften nicht hinreichend erfaßt werden. Die Folge ist

Bild 7: Darstellung eines typischen Verlaufs von $\sum f d_i(w)$.



eine große Varianz der Texturmerkmale für verschiedene Fensterpositionen (vergleiche Bild 7). Andererseits können bei zu großen Fenstern kleine Fehler nicht mehr detektiert werden, da die Merkmalsabweichung zum Modell zu klein ist.

Zur Automatisierung der Fenstergrößeneinstellung bei Texturanalyseverfahren wurde in [17] ein Maß zur Erfassung der Merkmalsinhomogenität vorgeschlagen:

$$f d_i(w) = \frac{(\bar{m}_i(w) - \min\{m_i(w)\})^2 + (\bar{m}_i(w) - \max\{m_i(w)\})^2}{\bar{m}_i^2(w)}. \quad (6)$$

Die Merkmalsvariation $f d_i(w)$ erfaßt den gesamten Wertebereich eines Merkmals $m_i(w)$ bezogen auf das arithmetische Mittel $\bar{m}_i(w)$ für alle untersuchten Fenster der Fenstergröße w einer Szene. Der Mittelwert $\sum f d_i(w)$ der Merkmalsvariationen aller betrachteten Texturmerkmale wird zur Entscheidung über die geeignete Fenstergröße herangezogen. Ein typischer Verlauf von $\sum f d_i(w)$ ist in Bild 7 dargestellt. Das optimale Fenster ergibt sich aus der Überschreitung einer negativen Grenze der Kurvensteigung („Knick“ in der Kurve aus Bild 7).

3.2 Einstellung von Grauwert- und Lauflängenbereichen bei Texturanalyseverfahren

Zur Erfassung der Richtungsabhängigkeit bestimmter Grauwertkombinationen sowie anderer Texturmerkmale sind Verfahren populär geworden, die verschiedene Grauwertstatistiken in Form von Matrizen berechnen (Grauwert-Abhängigkeits-Matrix oder co-occurrence Matrix [18], Nachbarschafts-Grauwert-Abhängigkeits-Matrix [19], Grauwert-Differenzen-Abhängigkeits-Matrix [20], Nachbarschafts-Grauwert-Differenzen-Matrix [21], Grauwert-Lauflängen-Matrix [22]). Da diese Matrizen abhängig von der Anzahl unterschiedlicher Grauwerte und Grauwertlauflängen sehr groß werden können und deren Berechnung daher einen extrem hohen Rechenaufwand erfordert, ist die Beschränkung auf eine geringere Anzahl zu unterscheidender Grauwert- und Lauflängenbereiche wesentlich zur Beschleunigung der Berechnungen. In [23] werden Verfahren und deren Anwendung auf Beispiele aus der Texturanalyse vorgestellt, die eine Einteilung der Wertebereiche aller Grauwerte und Lauflängen auf optimale, dem Texturproblem angepaßte Intervalle realisieren. Bild 8 zeigt beispielhaft die mit den Merkmalen „gray level distribution“ GLD und „run percentage“ RP des Grauwert-Lauflängen Verfahrens erzielbaren Güten (zur Definition der Gütewerte siehe [23]) in Abhängigkeit der Anzahl unterschiedlicher Grauwertbereiche. Es ist festzustellen, daß nach Überschreiten einer Mindestanzahl unterschiedlicher Bereiche keine Verbesserung der Merkmalsgüten mehr auftritt.

Die Aufteilung einzelner Intervalle der auftretenden Grauwerte und Lauflängen auf die unterschiedlichen Bereiche kann äquidistant, gleichverteilt oder segmentierend erfolgen. Bei äquidistanter Einteilung wird der Eingangswertebereich äquidistant in Bereiche aufgeteilt, wogegen bei gleichverteilter Einteilung die

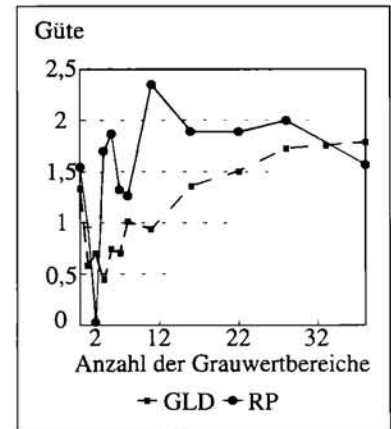


Bild 8: Darstellung der Güte von zwei Merkmalen in Abhängigkeit von der Anzahl unterschiedlicher Grauwertbereiche.

Bereichsgrenzen so gelegt werden, daß die Häufigkeit auftretender Grauwerte in allen Bereichen gleich ist. Die segmentierende Einteilung teilt die Bereiche nach Minima im Grauwert histogramm auf.

3.3 Verfahrensspezifische Merkmalsselektion

In Abschnitt 2.1 wurde die Wichtigkeit einer Abstimmung von Merkmalsextraktion und Klassifikation im Rahmen der Echtzeitproblematik erläutert. Ist beim Systemdesign die Entscheidung für den Einsatz einer speziellen Methode gefallen, so liefert diese Methode im allgemeinen eine maximale Anzahl von Merkmalen, die außerdem von der gewählten Ordnung bzw. Approximationsgenauigkeit des Verfahrens abhängen kann. Zur Einsparung von Rechenaufwand gilt es, die Menge der Merkmale und damit die Rechenkomplexität so gering wie möglich zu halten unter Wahrung der problemspezifischen Erkennungsleistung. Die Bestimmung der kleinsten Untermenge erforderlicher Merkmale kann durch Verfahren der Merkmalsselektion erfolgen.

Prinzipiell ist zwischen folgenden zwei Möglichkeiten der Vorgehensweise bei der Merkmalsselektion zu unterscheiden:

1. Bewertung des Klassifikationsergebnisses bezüglich Richtig- und Falschklassifikationen für eine bestimmte Merkmalskombination,
2. Bewertung der Verteilung der Merkmale im Merkmalsraum.

Die erste Möglichkeit hat den Vorteil, daß die Kombination von Merkmalen bzw. Verfahren und Klassifikator in einem Schritt bewertet wird. Nachteile dieses Vorgehens liegen in der Tatsache, daß die erzielte Erkennungsleistung nicht in der Güte der Merkmale bzw. des Klassifikators lokalisierbar ist. Außerdem ist eine Initialisierung bzw. ein Training des Klassifikators für jede Merkmalskombination erforderlich, was einen erheblichen Rechenaufwand bedeutet. Wenn mehrere Klassifikatoren untersucht werden sollen, muß die Einheitlichkeit der Schnittstelle für alle Verfahren gewährleistet sein.

Für die zweite Vorgehensweise zur Merkmalsselektion, die auf der Bewertung der Merkmale im Merkmalsraum basiert, zeigt Bild 9 vereinfacht das

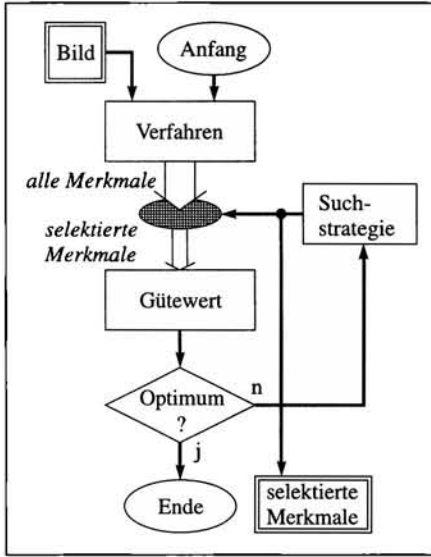


Bild 9: Merkmalsselektion mit direkter Bewertung der Merkmalsverteilung.

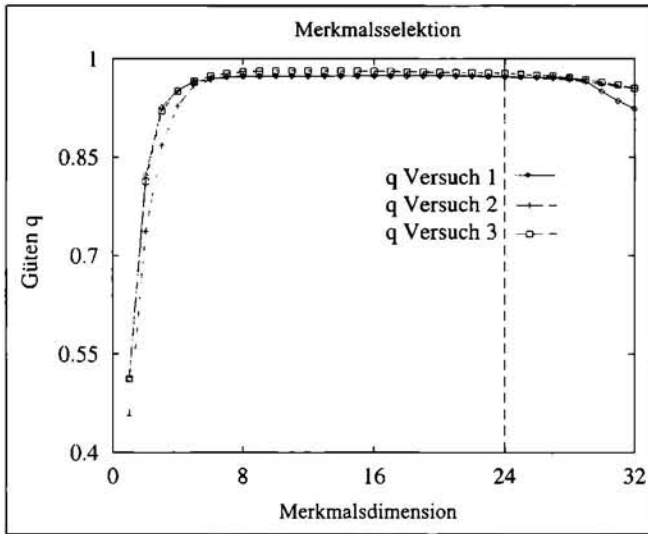


Bild 10: Überlappungsgrade q aus Gl. (4) als Gütwerte bei der Merkmalsselektion.

Vorgehen. Wie in Abschnitt 2.2 erläutert, folgt aus der Überlappung von Merkmalsverteilungen verschiedener Klassen eine obere Grenze der erzielbaren Erkennungsleistung. Der in Gl. (4) vorgestellte Überlappungsgrad ist ein geeignetes Gütekriterium für den Algorithmus nach Bild 9. Mittels der dargestellten Vorgehensweise kann bei vergleichsweise geringem Rechenaufwand die optimale (bezogen auf den Suchraum) Merkmalskombination ermittelt werden. Anschließend kann die erforderliche Komplexität des auf dem optimalen Merkmalsatz arbeitenden Klassifikators bestimmt werden.

Wie Bild 9 verdeutlicht, besteht die Merkmalsselektion aus zwei wesentlichen Schritten:

1. der Suchstrategie,
2. der Berechnung eines Gütwertes.

Da die vollständigen Merkmalsräume so groß sein können, daß eine Bewertung aller möglichen Merkmalskombinationen in einem sinnvollen Zeitrahmen nicht möglich ist, muß eine Suchstrategie über die Selektion der Merkmale entscheiden. Die Güte des

gesamten Merkmalsselektionsverfahrens ist daher durch die Qualität des Gütwertes und dem Erfolg der Suchstrategie gegeben. In [24] sind die Ergebnisse eines Vergleichs verschiedener Suchstrategien dargestellt. Populär sind heuristische Suchverfahren, wobei ausgehend von der leeren Merkmalsmenge sequentiell die jeweils besten Merkmale hinzugefügt werden (sequentielle Vorwärtsselektion) oder ausgehend von der Gesamtmenge aller Merkmale, die jeweils schlechtesten Merkmale gelöscht werden (sequentielle Rückwärtsselektion). Bild 10 zeigt das Verhalten der Überlappungsgrade q aus Gl. (4) als Gütwerte bei einer sequentiellen Rückwärtsselektion in Abhängigkeit der Merkmalsdimension. Es wurden 3 Verfahren der Texturanalyse untersucht (Versuch 1 bis 3 in Bild 10), die jeweils 32 Merkmale liefern. Man erkennt, daß in einem Bereich zwischen 7 und 24 Merkmalen die Güte auf hohem Niveau etwa gleich bleibt. Eine Reduktion des Merkmalsatzes auf 7 Merkmale ist hier sinnvoll zur Reduzierung des Rechenaufwands. Eine bessere Überdeckung des Suchraums läßt sich durch den Einsatz genetischer Algorithmen (GA) erreichen, die jedoch im allgemeinen mehr Rechenaufwand erfordern.

3.4 Automatische Klassifizierung von Stichproben

Zum Training von Klassifikatoren bzw. zur Modellierung der den logischen Entscheidungen zugrundeliegenden Signale sowie zur Anwendung von Merkmalsselektionsverfahren sind Stichproben mit bekannter Interpretation erforderlich. Zu Beginn eines Systemdesigns müssen daher alle Mustervorlagen gesichtet und interpretiert, d.h. mit Bedeutung versehen werden und in ein rechnerlesbares Datenformat gebracht werden. Im Fall der Oberflächeninspektion müssen die Muster digitalisiert werden, es müssen „gut“ und „schlecht“ Regionen sowie unterschiedliche Fehlertypen definiert und verschiedenen Klassen zugeordnet werden. Anschließend müssen die Merkmale in den festgelegten Regionen berechnet und mit den zugehörigen Klassenlabels als Trainings- und Testdaten abgelegt werden.

Dieser Prozeß ist im allgemeinen sehr zeitaufwendig, aber entscheidend für die Qualität des ausgelegten Systems. In vielen Anwendungsfällen kann die Festlegung zu unterscheidender Regionen oder Objekte sowie die Zuweisung unterschiedlicher Klassen durch Verfahren der Datenclustering (siehe z.B. [14]) automatisiert oder teilautomatisiert werden. Voraussetzung für die erfolgreiche Anwendung von Clusteringalgorithmen ist die Existenz verschiedener Ballungsräume (Cluster) der Merkmale für verschiedene Klassen. Da dies in der Praxis oft nur teilweise erfüllt ist, können die Ergebnisse von Clusteringalgorithmen als Vorschlag zur Einteilung in verschiedene Klassen angesehen werden und damit das Systemdesign interaktiv unterstützen.

Die Güte des ausgelegten Systems ist sehr eng mit der Signifikanz der Stichproben verbunden. Die Stichproben müssen den Problembereich der Erkennungsaufgabe umfassend dokumentieren. Gerade hier liegen häufig die Schwierigkeiten bei der Realisierung von

Bildverarbeitungssystemen für industrielle Kunden. Oft existieren nur wenige Muster fehlerhafter Produkte oder eine Zusammenstellung der Muster ist nur mit großem Aufwand möglich. Intensive Kommunikation zwischen Systemanbieter und Kunde ist unerlässlich zur exakten Definition und Lösung der Erkennungsaufgabe.

4 Zusammenfassung

Im vorliegenden Beitrag werden Aspekte der Echtzeit-Bildverarbeitung zur Anwendung in der Qualitätssicherung diskutiert. Der Schwerpunkt liegt dabei nicht auf der Anwendung von Spezialhardware zur Erfüllung der Echtzeitanforderungen, sondern vielmehr auf der Optimierung von Softwaresystemen. Mit der steigenden Leistung freiprogrammierbarer Prozessoren ist der Einsatz flexibler Software zur Lösung von Erkennungsproblemen aus dem Bereich der industriellen Qualitätssicherung möglich geworden.

Es werden Ansätze diskutiert, die sowohl Datenstrom als auch Rechenaufwand entlang der Bildverarbeitungs-pipeline reduzieren. Beispiele hierfür sind die konturbasierte Bildverarbeitung sowie Verfahren zur regionbasierten Texturanalyse. Die gemeinsame Optimierung der Verfahrensschritte Merkmalsextraktion und Klassifikation wird betrachtet unter dem Aspekt einer Reduktion der Gesamtkomplexität. In diesem Zusammenhang wird der Einsatz verschiedener Klassifikator-typen diskutiert.

Zur Erzielung eines optimierten Systemdesigns werden Methoden der automatischen Verfahrenskonfigu-rierung anhand mehrerer Beispiele vorgestellt. Es werden Ansätze zur automatischen Fenstergrößeneinstellung sowie zur Wahl von Grauwert- und Lauflängenbereichen bei der Texturanalyse vorgestellt, die zu einer erheblichen Einsparung an Rechenaufwand führen. Nach der Einführung eines neuen Maßes zur Beurteilung der Überlappung von Merkmalsverteilungen werden Methoden der Merkmalsselektion und automatischen Klassifizierung von Stichproben diskutiert. Die Berücksichtigung der dargestellten Prinzipien bei der Auslegung von Bildverarbeitungssystemen zur Qualitätssicherung unter Anwendung automatischer Konfigurierungsmethoden führt zur Realisierbarkeit von Echtzeit-Softwaresystemen für komplexe Erkennungsaufgaben.

Literatur

- [1] Marr, D.: Vision. W. H. Freeman, San Francisco, 1982.
- [2] Nothdurft, H. C.: Structural and functional organization of the neocortex, Kapitel Cortical properties of preattentive vision, Seiten 375–384. Springer Verlag Heidelberg, 1994.
- [3] Tatari, S. M.: Auswahl von Texturanalyseverfahren zur Automatisierung industrieller Sichtprüfungen. Nummer 67 in VDI Fortschrittberichte, Reihe 10. VDI Verlag Düsseldorf, 1987.
- [4] Nickolay, B.: Überwacht lernendes Bildauswertungssystem zur Erkennung von Oberflächenfehlern. Nummer 84 in Produktionstechnik-Berlin. Hanser-Verlag, 1990.
- [5] Niemann, H. und Bunke, H.: Künstliche Intelligenz in Bild- und Sprachanalyse. B. G. Teubner, Stuttgart, 1987.
- [6] Liedtke, C.-E. und Ender, M.: Wissensbasierte Bildverarbeitung. Springer-Verlag, Berlin u.a., 1989.
- [7] Bässmann, H. und Besslich, P. W.: Konturorientierte Verfahren in der digitalen Bildverarbeitung. Springer-Verlag, Berlin u.a., 1989.
- [8] Abou-Nabout, A.: Modulares Konzept und Methodik zur wissensbasierten Erkennung komplexer Objekte in CAQ-Anwendungen. Nummer 92 in VDI Fortschrittberichte, Reihe 20. VDI Verlag Düsseldorf, 1993.
- [9] Amelung, J., Lambert, G. und Pfister, J.: Ein vektorbasiertes Verfahren zur schnellen Fehlererkennung in strukturierten Texturen. In Mustererkennung 1994, 16. DAGM Symposium, Seiten 666–675. Springer, September 1994.
- [10] Lambert, G., Gao, H., Hohm, K. und Amelung, J.: Linienmomente und Invarianten zur Echtzeitverarbeitung vektorisierter Konturen. In Mustererkennung 1995, 17. DAGM Symposium, Seiten 570–577. Springer, September 1995.
- [11] Bovik, A. C. und Clark, M.: Multichannel Texture Analysis Using Localized Spatial Filters. IEEE Trans. on Pattern Analysis and Machine Intelligence, 12(1):55–73, 1990.
- [12] Dunn, D., Higgins, W. E. und Wakeley, J.: Texture Segmentation Using 2-D Gabor Elementary Functions. IEEE Trans. on Pattern Analysis and Machine Intelligence, 16(2):130–149, Februar 1994.
- [13] Niemann, H.: Klassifikation von Mustern. Springer-Verlag, Berlin, Heidelberg, 1983.
- [14] Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press, Boston, 1990.
- [15] Ripley, B. D.: Pattern Recognition and Neural Networks. Cambridge University Press, 1996.
- [16] König, A.: Neuronale Strukturen zur sichtgestuetzten Oberflächeninspektion von Objekten in industrieller Umgebung. Dissertation, Technische Hochschule Darmstadt, Institut für Mikroelektronik, 1995.
- [17] Amelung, J. und Vogel, K.: Automated window size determination for texture defect detection. In Hancock, E. (Hrsg.), 5th British Machine Vision Conference, Seiten 105–114, York, UK, Sept 1994. BMVA Press.
- [18] Haralick, R. M., Shanmugam, K. und Dinstein, I.: Textural Features for Image Classification. IEEE Transactions on Systems, Man and Cybernetics, 3(6):610–621, 1973.
- [19] Sun, C. und Wee, W. G.: Neighboring Gray Level Dependence Matrix for Texture Classification. Computer Vision, Graphics and Image Processing, 23:341–352, 1982.
- [20] Dapeng, Z. und Zhongrong, L.: Digital image texture analysis using gray level and energy cooccurrence. In Proc. Conf. Applications of Artificial Intelligence IV, SPIE, Band 657, Seiten 152–156, 1986.
- [21] Amadasun, M. und King, R.: Textural Features Corresponding to Textural Properties. IEEE Trans. on Systems, Man and Cybernetics, 19(5):1264–1274, 1989.
- [22] Galloway, M. M.: Texture Analysis Using Gray Level Run Lengths. Computer Graphics and Image Processing, 4:172–179, 1975.
- [23] Amelung, J.: Automatische Bildverarbeitung für die Qualitätssicherung. Dissertation, Technische Hochschule Darmstadt, Institut für Regelungstechnik, 1995.
- [24] Zongker, D. und Jain, A.: Algorithms for Feature Selection: An Evaluation. In 13th International Conference on Pattern Recognition, Band Volume II, Track B aus Pattern, Recognition and Signal Analysis, Seiten 18–22. IEEE Computer Society Press, August 1996.

Manuskripteingang: 19. November 1996.