Technical Report

Dept. for Pattern Recognition and Image Processing
Institute for Automation
Technical University of Vienna
Treitlstr. 3/1832
A-1040 Vienna AUSTRIA
Phone: +43 (1) 58801-8161
Fax: +43 (1) 5054668
E-mail: roland@prip.tuwien.ac.at

PRIP-TR-44 23. September 1996

Robuste Eigenbildfunktionen
Aufgabenstellung der Diplomarbeit

Ebensperger Roland

Abstract

Die grundlegenden Einschränkungen der gegenwärtigen Eigenspace Methoden sind ihre Probleme mit gestörten Daten und überdeckten Objekten. Am Institut für Mustererkennung wurde eine neue Methode entwickelt, die diese Probleme löst. Im Rahmen der Diplomarbeit soll diese Methode in ein bestehendes Softwarepaket für appearance-based matching (SLAM) integriert werden. Weiters soll die Koeffizientenberechnung in einem diskreten Raum ausgeführt werden. Die Methode soll auf ihre Robustheit und Berechnungskomplexität hin untersucht und mit der Standardmethode verglichen werden.

1 Einleitung

Die Diplomarbeit beschäftigt sich mit Eigenspace Methoden, die ein wichtiger Vertreter der appearance-based matching Methoden sind. Der größte Vorteil dieser Methoden ist, daß sie zur Objekterkennung kein geometrisches Modell der Objekte benötigen. Mit Hilfe des Aussehens von drei-dimensionalen Objekten auf zwei-dimensionalen Bildern können sie die Repräsentation der Objekte automatisch lernen. Das Aussehen eines Objektes in einem zwei-dimensionalen Bild hängt von seiner Form, seinen Reflektionseigenschaften, seiner Lage und von den Beleuchtungsbedingungen ab [7]. Dabei sind Form und Reflektionseigenschaften eines Objektes konstant, während seine Lage und die Beleuchtungsbedingungen von Bild zu Bild variieren können. Appearance-based matching Methoden wurden bei Beleuchtungsplanung [8], Positionierung von Roboterarmen [9], Visueller Kontrolle [13], „Image spotting“ [6] und bei der Erkennung von Gesichtern [12, 2] erfolgreich eingesetzt.

Die wesentlichsten Einschränkungen der gegenwärtigen Eigenspace Methoden sind ihre Probleme mit überdeckten Objekten und gestörten Daten. Aus diesem Grund wurde am Institut für Mustererkennung eine Hypothese & Auswahl Methode entwickelt, die diese Probleme löst [4]. Der wesentlichste Unterschied zur Standardmethode ist die Art, wie die Koeffizienten der Eigenbilder bestimmt werden. Anstatt sie durch eine Projektion der Daten in den Eigenspace zu berechnen, werden sie durch Aufstellen von Hypothesen hergeleitet. Konkurrierende Hypothesen werden dann durch eine Auswahlprozedur eliminiert.

Dieser Technische Report ist folgendermaßen aufgebaut: Kapitel 2 behandelt die gegenwärtigen Eigenspace Methoden, Kapitel 3 beschreibt die Hypothese & Auswahl Methode, Kapitel 4 stellt die Ziele der Diplomarbeit vor und Kapitel 5 enthält einen Zeitplan für den Ablauf der Diplomarbeit.

2 Eigenspace Methoden

Eigenspace Methoden bestehen aus zwei Schritten. Zuerst wird das Aussehen eines oder mehrerer Objekte unter verschiedenen Orientierungen und Beleuchtungsrichtungen aufgenommen. Diese Menge von Bildern (Trainingsbeispiele) wird nun mit der Karhunen-Loève Transformation [1] komprimiert. Das Ergebnis der Transformation ist ein niederdimensionaler Eigenspace. Der zweite Schritt projiziert Teile eines Eingabebildes (diese Teile sind gleich groß wie die Trainingsbeispiele) in den Eigenspace. Die so berechneten Koeffizienten geben dann das erkannte Objekt, seine Orientierung und Beleuchtung an.

Für die mathematische Darstellung dieser Zusammenhänge wird folgende Notation verwendet: Ein Trainingsbild wird als Vektor $\mathbf{y} = [y_1, \ldots, y_m]^T \in \mathbb{R}^m$ dargestellt. Die Elemente y_i von $\mathbf{y}$ entsprechen den unverarbeiteten Helligkeitswerten des Bildes. Es können jedoch auch vorverarbeitete Bilder (geglättete Bilder, erste Ableitungen, Power spectrum des Helligkeitsbildes) verwendet werden, ohne daß sich etwas an der Methode ändert [7].

$\mathcal{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ bezeichnet eine Menge von Trainingsbeispielen (Abbildung 1). Um die Notation zu vereinfachen, wird angenommen, daß $\mathcal{Y}$ normiert ist und den Mittelwert Null besitzt.

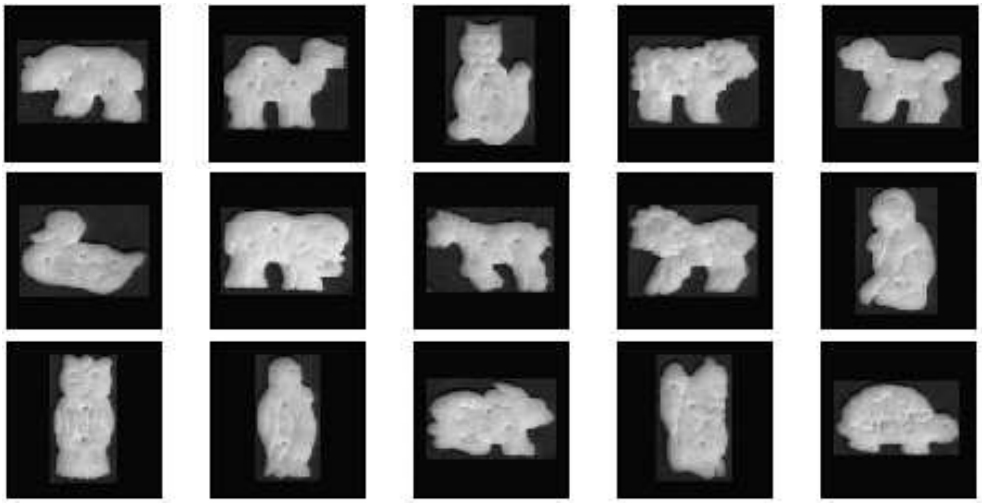


Abbildung 1: Menge von Trainingsbeispielen.

$\mathbf{Q}$ ist die Kovarianzmatrix der Vektoren in $\mathcal{Y}$. $\mathbf{e}_i$ sind die Eigenbilder (Abbildung 2) von $\mathbf{Q}$, λ_i die korrespondierenden Eigenwerte und $1 \leq i \leq n$. Die Eigenbilder sind entsprechend den korrespondierenden Eigenwerten ($\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$) geordnet.

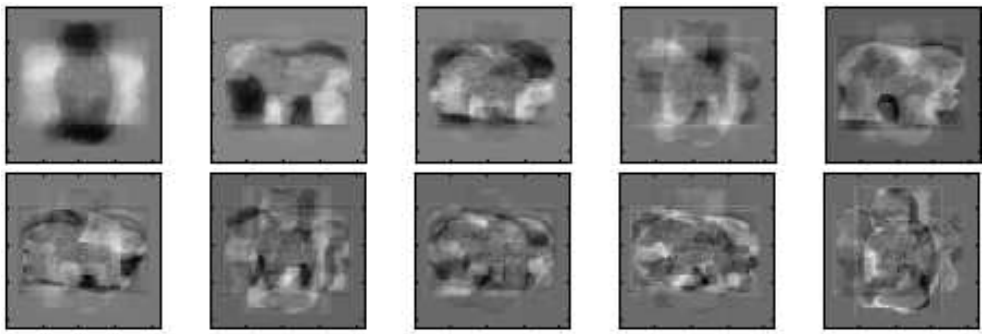


Abbildung 2: Die ersten zehn Eigenbilder der Trainingsbeispiele von Abbildung 1.

Abhängig von der Korrelation der Trainingsbeispiele braucht man nur p, $p < n$, Eigenbilder um die $\mathbf{y}_i$ mit hinreichender Genauigkeit als Linearkombination der Eigenbilder $\mathbf{e}_i$ zu repräsentieren (Abbildung 3):

$$\tilde{\mathbf{y}} = \sum_{i=1}^{p} a_i(\mathbf{y})\mathbf{e}_i \quad . \tag{1}$$

Der Raum, der von den ersten p Eigenbildern aufgespannt wird, wird *Eigenspace* genannt.

Die Parameter a_i werden durch Projektion des Datenvektors $\mathbf{x}$ in den Eigenspace bestimmt:

$$a_i(\mathbf{x}) = < \mathbf{x}, \mathbf{e}_i > = \sum_{j=1}^{m} x_j e_{ij} \qquad 1 \leq i \leq p \quad . \tag{2}$$

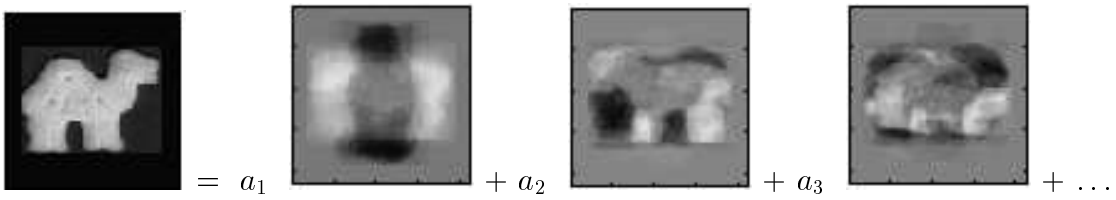


Abbildung 3: Rekonstruktion eines Kamels mit einer Linearkombination der Eigenbilder.

$\mathbf{a}(\mathbf{x}) = [a_1(\mathbf{x}), \ldots, a_p(\mathbf{x})]^T$ ist der Punkt im Eigenspace, den man bei Projektion von $\mathbf{x}$ in den Eigenspace erhält. Die $a_i(\mathbf{x})$ werden Koeffizienten von $\mathbf{x}$ genannt. Der rekonstruierte Datenvektor $\tilde{\mathbf{x}}$ wird folgendermaßen berechnet:

$$\tilde{\mathbf{x}} = \sum_{i=1}^{p} a_i(\mathbf{x})\mathbf{e}_i \ . \tag{3}$$

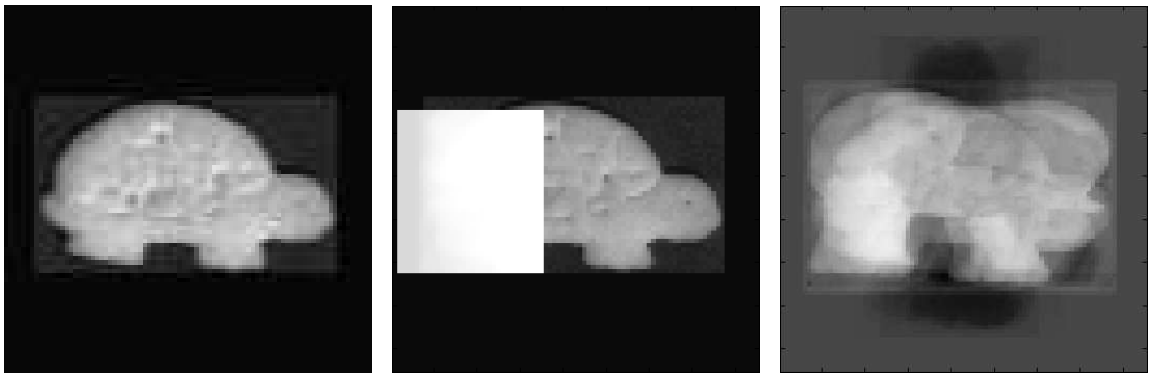
2.1 Vorteile und Nachteile der gegenwärtigen Eigenspace Methoden

Der größte Vorteil ist die Fähigkeit der Methoden, die Repräsentation eines Objektes automatisch zu lernen. Es ist nicht notwendig, ein geometrisches Modell des Objektes zu erstellen.

Die wesentlichsten Einschränkungen sind ihre Probleme mit überdeckten Objekten (Abbildung 4) und gestörten Daten. Diese Probleme werden dadurch verursacht, daß immer die ganze Datenmenge eines Bildes benötigt wird, um den Koeffizientenvektor $\mathbf{a}$ der Eigenbilder zu berechnen. Im folgenden werden die Schwächen der Methoden bei Überdeckungen und Outliern genauer untersucht.

Überdeckung Von $\mathbf{x}$ werden $m - r$ Komponenten auf 0 gesetzt. Dann ist $\hat{\mathbf{x}} = [x_1, \ldots, x_r, 0, \ldots 0]^T$ und $\hat{a}_i = \hat{\mathbf{x}}^T\mathbf{e}_i = \sum_{j=1}^{r} x_j e_{ij}$. Der Fehler bei der Berechnung von a_i ist $(a_i(\mathbf{x}) - \hat{a}_i(\hat{\mathbf{x}})) = \sum_{j=r+1}^{m} x_j e_{ij}$. Der Fehler bei der Rekonstruktion ist $(\tilde{\mathbf{x}} - \sum_{i=1}^{p}(\sum_{j=r+1}^{m} x_j e_{ij})\mathbf{e}_i)$. Das zeigt, daß sich der Fehler über den ganzen Vektor $\mathbf{x}$ verteilt. Abbildung 4 zeigt dieses Problem anhand eines konkreten Beispiels.

Outlier Zu x_j wird in Gleichung (2) ein Fehler δ addiert. Dann ist $\hat{a_i} = a_i + \delta e_{ij}$. Das zeigt, daß ein einziger falscher Punkt den Koeffizientenvektor $\mathbf{a}$ komplett verfälschen kann.



(a) Originales Bild (b) Überdecktes Bild (c) Rekonstruiertes Bild

Abbildung 4: Schwächen der gegenwärtigen Eigenspace Methoden.

3 Hypothese & Auswahl Methode

Diese Methode [4] löst die Probleme mit gestörten Daten und überdeckten Objekten. Der wesentlichste Unterschied zur Standardmethode ist die Art, wie die Koeffizienten der Eigenbilder berechnet werden. Anstatt sie durch eine Projektion der Daten in den Eigenspace zu berechnen, werden sie durch Aufstellen von Hypothesen hergeleitet. Konkurrierende Hypothesen werden dann durch eine Auswahlprozedur eliminiert. Außerdem werden nicht mehr alle Bildpunkte zur Berechnung der Koeffizienten verwendet. Die Methode besteht aus vier Schritten: Generierung der Hypothesen, Auswahl, Fitting und Endauswahl.

3.1 Generierung der Hypothesen

Für die Berechnung der p Koeffizienten a_i (Gleichung 2) braucht es nur p Punkte $\mathbf{r} = (r_1, \ldots, r_p)$. Die Koeffizienten a_i können dann durch Lösen des folgenden Gleichungssystems bestimmt werden:

$$x_{r_i} = \sum_{j=1}^{p} a_j(\mathbf{x}) e_{jr_i} \quad 1 \leq i \leq p \ . \tag{4}$$

Mit den Koeffizienten a_i wird mit Gleichung (3) eine Hypothese generiert. Dann wird der Fehlervektor $\xi = (\mathbf{x} - \tilde{\mathbf{x}})^2$ berechnet. Die Punkte innerhalb einer Fehlergrenze Θ werden als *kompatible* Punkte bezeichnet. Eine Hypothese wird akzeptiert, wenn sie eine minimale Anzahl von kompatiblen Punkten hat. Die akzeptierte Hypothese ist durch den Koeffizientenvektor $\mathbf{a}$, den Fehlervektor ξ und der Menge der kompatiblen Punkte $D = \{j | \xi_j < \Theta\}$, $s = |D|$, charakterisiert.

Gleichung (4) kann aber nur dann angewendet werden, wenn alle Eigenbilder berücksichtigt werden, d.h. $p = n$, und wenn die Daten x_{r_i} nicht gestört sind. Werden nicht alle Eigenbilder benützt und sind die Daten gestört, muß ein überbestimmtes Gleichungssystem mit k Gleichungen ($p < k << m$) gelöst werden. Somit wird ein Vektor $\mathbf{a}$ gesucht, der folgende Funktion minimiert:

$$E(\mathbf{r}) = \sum_{i=1}^{k} (x_{r_i} - \sum_{j=1}^{p} a_j(\mathbf{x}) e_{jr_i})^2 \quad . \tag{5}$$

Die Minimierung von Gleichung (5) liefert nur korrekte Werte für den Koeffizientenvektor $\mathbf{a}$, wenn die Punktemenge r_i keine extrem verrauschten Punkte, Hintergrundpunkte oder Punkte aus einem überdeckten Bereich enthält. Dieses Problem wird folgendermaßen gelöst:

1. $k = 10p$ Punkte zufällig auswählen
2. Durch Minimieren von Gleichung (5) Koeffizienten berechnen
3. Mit Gleichung (3) Bild rekonstruieren
4. Fehlervektor berechnen
5. Menge der kompatiblen Punkte berechnen
6. k reduzieren
7. Wenn $k < 2p$, dann gehe zu Schritt 9
8. Aus den kompatiblen Punkten k Punkte zufällig auswählen und weiter bei Schritt 2
9. Generierung einer Hypothese

Nicht alle zufällig gewählten Punkte liefern auch gute Hypothesen. Für ein zu erkennendes Objekt werden deshalb im allgemeinen mehrere Hypothesen generiert. Eine Auswahlprozedur soll dann die besten Hypothesen finden.

3.2 Auswahl

Die Menge von generierten Hypothesen ist üblicherweise redundant. Die Aufgabe der Auswahlprozedur ist es nun, gute Hypothesen auszuwählen und überflüssige zu eliminieren. Dazu wird eine Optimierungsfunktion aufgestellt, die Informationen über die einzelnen Hypothesen enthält [5]. Diese Funktion hat die folgende Form:

$$F(\mathbf{h}) = \mathbf{h}^T \mathbf{C} \mathbf{h} = \mathbf{h}^T \begin{bmatrix} c_{11} & \dots & c_{1R} \\ \vdots & & \vdots \\ c_{R1} & \dots & c_{RR} \end{bmatrix} \mathbf{h} \quad . \tag{6}$$

Der Vektor $\mathbf{h}^T = [h_1, h_2, \ldots, h_R]$ bezeichnet eine Menge von Hypothesen, wobei h_i eine *Anwesenheitsvariable* ist, die den Wert 1 bei Anwesenheit und den Wert 0 bei Abwesenheit der Hypothese i in der resultierenden Beschreibung hat. Die Diagonalelemente geben die Kosten für eine spezielle Hypothese i an:

$$c_{ii} = \mathrm{K}_1 s_i - \mathrm{K}_2 ||\xi_i|| - \mathrm{K}_3 N_i \ . \tag{7}$$

N_i ist die Anzahl der Koeffizienten (Eigenbilder), s_i ist die Anzahl der kompatiblen Punkte und $||\xi_i||$ ist der Fehler. Die Koeffizienten K_1, K_2 und K_3 können automatisch bestimmt werden. K_1 bezeichnet die durchschnittlichen Kosten für das Beschreiben eines Datenpunktes, K_2 sind die durchschnittlichen Kosten für das Spezifizieren des Fehlers und K_3 sind die durchschnittlichen Kosten für das Spezifizieren eines Koeffizienten.
Die restlichen Elemente der Matrix $\mathbf{C}$ beschreiben die Interaktionen zwischen den überlappenden Hypothesen.

$$c_{ij} = \frac{-\mathrm{K}_1 |D_i \cap D_j| + \mathrm{K}_2 ||\xi_{i,j}||}{2} \quad , \quad ||\xi_{ij}|| = \max(\sum_{D_i \cap D_j} \xi_i, \sum_{D_i \cap D_j} \xi_j) \ . \tag{8}$$

D_i bezeichnet die Menge der kompatiblen Punkte der Hypothese i. Das Auswahlproblem ist so formuliert, daß seine Lösung dem Maximum der Optimierungsfunktion $F(\mathbf{h})$ entspricht. Da die Anzahl der Lösungen exponentiell mit der Größe des Problems zunimmt, ist es gewöhnlich sehr aufwendig, den gesamten Lösungsraum nach dem Maximum zu durchsuchen. Deshalb werden Lösungsverfahren verwendet, die zwar nicht unbedingt die beste, aber doch eine brauchbare Lösung finden. Bei diesen Verfahren handelt es sich um Tabu search [3, 11] und einen einfachen Greedy Algorithmus. Dabei ist Tabu search etwas aufwendiger zu berechnen, liefert jedoch im allgemeinen die besseren Resultate als der Greedy Algorithmus.

3.3 Fitting

Die Koeffizienten der generierten und ausgewählten Hypothesen wurden nur aus $2p$ Punkten berechnet. Um die Genauigkeit des Koeffizientenvektors $\mathbf{a}$ zu erhöhen, werden nun alle kompatiblen Punkte für die Berechnung von $\mathbf{a}$ mit Gleichung (5) verwendet. Das wird so oft wiederholt, bis Gleichung (5) konvergiert.

3.4 Endauswahl

Da das Fitting einige der Hypothesen verändert haben kann, wird noch eine Endauswahl durchgeführt. Dabei wird die gleiche Prozedur verwendet wie in Kapitel 3.2. Diese letzten beiden Schritte sind nicht sehr aufwendig, da sie nur auf einen kleinen Teil der Hypothesen angewendet werden.

4 Ziele der Diplomarbeit

4.1 SLAM

Die Hypothese & Auswahl Methode soll in SLAM integriert werden. SLAM (Software Library for Appearance Matching) [10] wurde an der Columbia University, New York, von S.A. Nene, S.K. Nayar und H. Murase entwickelt. SLAM ermöglicht dem Benutzer:

- Vorbereitung der Trainingsmenge (Segmentierung, Normierung von Größe und Helligkeit)
- Berechnung der Eigenbilder
- Projektion von Bildern in den Eigenspace
- Interpolation zwischen den projizierten Bildern (*parametric manifold*)
- Erkennung von Bildern

Diese Funktionalität wird dem Benutzer über eine X/Motif-Benutzeroberfläche (Abbildung 5), über command-line Programme und eine C++ Klassenbibliothek zur Verfügung gestellt.

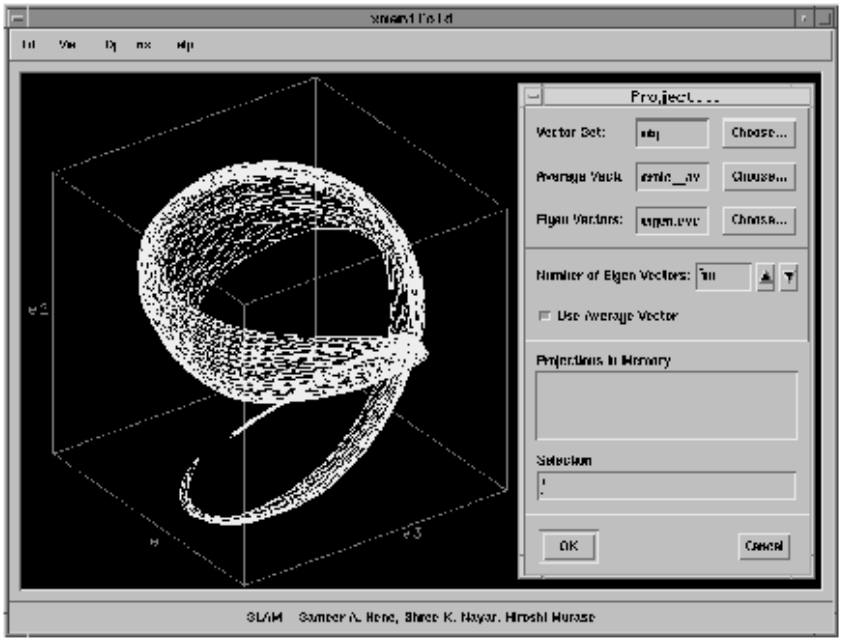


Abbildung 5: X/Motif-Oberfläche von SLAM.

4.1.1 Parametric Eigenspace

In SLAM ist eine appearance-based matching Methode implementiert, bei der die Objekte durch einen *parametric eigenspace* [7] repräsentiert werden. Die Parameter sind dabei z.B. die Lage und die Beleuchtungsrichtung des Objektes.

Anhand von Abbildung 6(a) soll gezeigt werden, wie die Repräsentation eines Objektes im *parametric eigenspace* berechnet wird. Zuerst wurde das Aussehen des Objektes unter verschiedenen Orientierungen aufgenommen. Dazu befand sich das Objekt auf einem computergesteuerten Drehtisch, der zwischen zwei aufeinanderfolgenden Aufnahmen jeweils um 4 Grad weiterbewegt wurde. Da das Objekt auf diese Weise eine volle 360 Grad Drehung ausführte, ergab sich eine Trainingsmenge mit 90 Bildern. Nun wurden die Eigenbilder dieser Trainingsmenge berechnet. Anschließend wurden die einzelnen Trainingsbilder in den Eigenspace projiziert. Die Punkte in Abbildung 6(a) entsprechen diesen Projektionen.

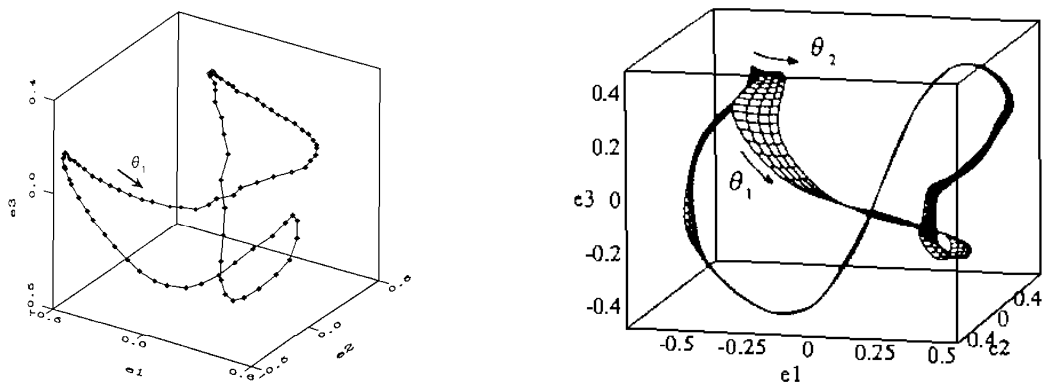
Der Eigenspace hat folgende wichtige Eigenschaft: der Abstand zweier Punkte im Eigenspace ist ein Maß für die Ähnlichkeit der korrespondierenden Bilder [7], d.h. je geringer der Abstand zweier Punkte im Eigenspace ist, desto ähnlicher sind sich die korrespondierenden Bilder. Da sich im konkreten Fall die Orientierung des Objektes zwischen zwei aufeinanderfolgenden Aufnahmen lediglich um 4 Grad verändert hat, sind sich zwei aufeinanderfolgende Bilder sehr ähnlich. Das bedeutet aber, daß der Abstand zwischen den korrespondierenden Punkten im Eigenspace sehr gering ist.

Angenommen, man hat nun ein neues Bild des Objektes, dessen Repräsentation in Abbildung 6(a) dargestellt ist und die Orientierung des Objektes in diesem Bild befindet sich zwischen zwei aufeinanderfolgenden Orientierungen, die für das Training verwendet wurden. Dann befindet sich auch die Projektion dieses Bildes im Eigenspace zwischen den Punkten der entsprechenden Trainingsbeispiele. Deshalb wird eine Interpolation der diskreten Punkte durchgeführt. Dadurch erhält man eine kontinuierliche Repräsentation des Objektes, die durch seine Orientierung parametrisiert ist. Mit Hilfe der Interpolation ist es möglich, das Objekt auch unter jenen Orientierungen zu erkennen, die nicht für das Training verwendet wurden.

Abbildung 6(b) zeigt die Repräsentation eines Objektes, das durch seine Orientierung (θ_1) und die Beleuchtungsrichtung (θ_2) parametrisiert ist. Durch die Interpolation ergibt sich deshalb eine Fläche.

4.2 Diskrete Koeffizienten

In Kapitel 3.1 wird bei der Berechnung der Koeffizienten davon ausgegangen, daß sie jeden beliebigen Wert im Eigenspace annehmen können Deshalb werden sie als *kontinuierliche Koeffizienten* bezeichnet. Für die Koeffizienten sind aber nicht alle Werte zulässig. Wird zur Klassifikation verschiedener Objekte nur ein Trainingsbild für jedes Objekt verwendet, dann können die Koeffizienten auch nur die Werte dieser diskreten Punkte im Eigenspace annehmen. Wird ein Objekt unter verschiedenen Orientierungen und Beleuchtungsbedingungen aufgenommen, sodaß sich als Repräsentation dieses Objektes eine Interpolationskurve ergibt (Kapitel 4.1.1), dann können die Koeffizienten nur auf dieser Parameterkurve



(a) Interpolationskurve (b) Interpolationsfläche

Abbildung 6: Parametric Eigenspace.

liegen. Diese Einschränkungen sollen nun zur Verringerung der Berechnungskomplexität und zur Steigerung der Robustheit der Methode eingesetzt werden. Da die Koeffizienten bei dieser Art der Berechnung nicht mehr jeden beliebigen Wert im Eigenspace annehmen können, werden sie als *diskrete Koeffizienten* bezeichnet.

4.2.1 Kontinuierliche Koeffizienten

Die Koeffizienten (Punkt P in Abbildung 7) für das zu erkennende Objekt werden berechnet. Nun wird iteriert (Kapitel 3.1), damit der Punkt zu einem der Trainingsobjekte konvergiert. Mit den Koeffizienten des Punktes H wird dann eine Hypothese generiert.

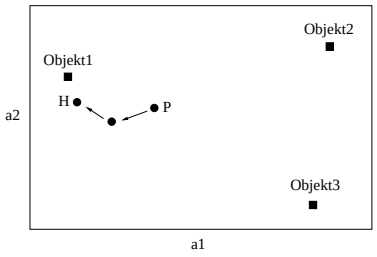


Abbildung 7: Beispiel für kontinuierliche Koeffizienten

4.2.2 Diskrete Koeffizienten

Die Koeffizienten (Punkt P in Abbildung 8) für das zu erkennende Objekt werden mit Gleichung (4) oder Gleichung (5) bestimmt. Es ist auch möglich, die Koeffizienten mit dem

kompletten Algorithmus aus Kapitel 3.1 mit wenigen Iterationen zu berechnen. Die genaue Vorgangsweise soll im Rahmen der Diplomarbeit untersucht werden. Nun wird das Trainingsobjekt mit geringstem Abstand zum Punkt P gesucht. Dann wird eine Hypothese mit den Koeffizienten dieses Trainingsobjektes generiert. Es soll außerdem untersucht werden, ob sich durch diese Berechnungsmethode die Schritte 3 und 4 (Fitting und Endauswahl) des Algorithmus einsparen lassen.

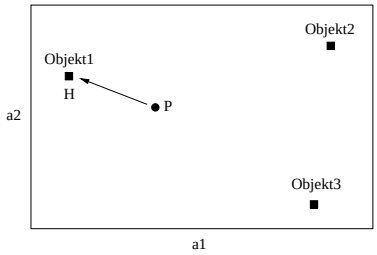


Abbildung 8: Beispiel für diskrete Koeffizienten

4.2.3 Diskrete Koeffizienten bei einer Interpolationskurve

Die Vorgangsweise bei einer Interpolationskurve ist ähnlich jener in Kapitel 4.2.2, nur daß hier der Punkt auf der Interpolationskurve mit geringstem Abstand zum Punkt P gesucht wird. Dann wird eine Hypothese mit den Koeffizienten dieses Punktes H generiert.

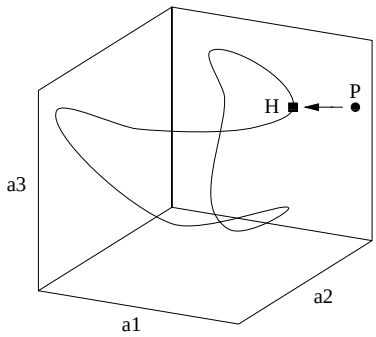


Abbildung 9: Beispiel für diskrete Koeffizienten

4.3 Robustheit

Die Robustheit der Hypothese & Auswahl Methode sowie der Methode mit diskreten Koeffizienten soll bei folgenden Störungen der Bilder untersucht werden:

- *Veränderter Hintergrund:* Objekte sollen auch auf einem Hintergrund erkannt werden, der nicht beim Training verwendet wurde.

- *Gaußsches Rauschen:* Alle Bildpunkte weichen gemäß einer Normalverteilung von ihrem korrekten Wert ab.
- *Impulsartige Störungen (Outlier):* Einzelne Bildpunkte sind falsch. Hier soll die Frage beantwortet werden, wieviel % der Bildpunkte gestört sein dürfen und wie groß die Störungen sein dürfen, damit das Objekt trotzdem noch erkannt wird.
- *Überdeckungen:* Ganze Teile eines Objektes sind verdeckt. Es soll untersucht werden, wieviel % des Objektes verdeckt sein können, damit es noch erkannt wird.

4.4 Berechnungskomplexität der Koeffizienten

Die Berechnungskomplexität der Koeffizienten der Hypothese & Auswahl Methode und der Methode mit diskreten Koeffizienten soll untersucht werden. Die Standardmethode hat konstante Berechnungskomplexität, da sie die Koeffizienten durch eine Projektion des Bildes in den Eigenspace berechnet.

Zur Berechnungskomplexität sollen folgende Fragen beantwortet werden:

- Wieviele Hypothesen werden für ein Bild benötigt? Es ist anzunehmen, daß umso mehr Hypothesen benötigt werden, je gestörter ein Bild ist.
- Wie aufwendig ist die Berechnung einer Hypothese?
- Wie aufwendig ist das Lösen des least-squares Problems?
- Wie ändert sich die Komplexität bei Störungen? Es wird erwartet, daß die Berechnungskomplexität zunimmt, da bei gestörten Bildern die Wahrscheinlichkeit, daß schlechte Hypothesen generiert werden, größer ist.
- Unterschiede zwischen der Hypothese & Auswahl Methode und der Methode mit diskreten Koeffizienten? Hier ist anzunehmen, daß die Methode mit diskreten Koeffizienten eine geringere Berechnungskomplexität hat.

4.5 Bildmaterial für die Tests

Es ist bereits entsprechendes Bildmaterial am Institut für Mustererkennung vorhanden (z.B. Face-database, Biscuit animals). Die Tests sollen auch mit dem Bildmaterial durchgeführt werden, das mit SLAM mitgeliefert wird. Dieses Material besteht aus vorverarbeiteten Bildern (segmentiert, Größe und Helligkeit normiert) von 20 Objekten und unverarbeiteten Bildern von 5 Objekten. Jedes Objekt wurde dabei unter 72 verschiedenen Orientierungen aufgenommen.

5 Zeitplan

Phase	*geplantes Ende*
Präsentation der Problemstellung	02.07.96
Implementierung	20.09.96
Testen und Auswerten	30.10.96
Diplomarbeit schreiben	10.12.96
Abschlußpräsentation	Ende Jänner 96
Diplomarbeit einreichen	Anfang Februar 96
2. Diplomprüfung	18.03.97

Literatur

[1] T. W. Anderson. *An Introduction to Multivariate Statistical Analysis.* New York: Wiley, 1958.

[2] D. Beymer and T. Poggio. Face recognition from one example view. In *Proceedings of 5th ICCV'95.* IEEE Computer Society Press, 1995.

[3] F. Glover and M. Laguna. Tabu search. In C. R. Reeves, editor, *Modern heuristic techniques for combinatorial problems*, pages 70–150. Blackwell Scientific Publications, 1993.

[4] A. Leonardis and H. Bischof. Dealing with occlusions in the eigenspace approach. In *Proceedings of the CVPR'96, San Francisco*, June, 1996. To appear.

[5] A. Leonardis, A. Gupta, and R. Bajcsy. Segmentation of range images as the search for geometric parametric models. *International Journal of Computer Vision*, 14(3):253–277, 1995.

[6] H. Murase and S. K. Nayar. Image spotting of 3D objects using parametric eigenspace representation. In G. Borgefors, editor, *The 9th Scandinavian Conference on Image Analysis*, volume 1, pages 323–332, Uppsala, Sweden, June 1995.

[7] H. Murase and S. K. Nayar. Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[8] H. Murase and S.K. Nayar. Illumination planning for object recognition using parametric eigenspaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(12):1219–1227, 1994.

[9] S. K. Nayar, H. Murase, and S. A. Nene. Learning, positioning, and tracking visual appearance. In *IEEE International Conference on Robotics and Automation*, San Diego, May 1994.

[10] S.A. Nene, S.K. Nayar, and H. Murase. SLAM: Sofware library for appearance matching. Technical Report CUCS-019-94, New York: Columbia University, Department of Computer Science, September 1994.

[11] M. Stricker and A. Leonardis. ExSel++: A general framework to extract parametric models. In V. Hlávač and R. Šára, editors, *6th CAIP'95*, number 970 in Lecture Notes in Computer Science, pages 90–97, Prague, Czech Republic, September 1995. Springer.

[12] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[13] S. Yoshimura and T. Kanade. Fast template matching based on the normalized correlation by using multiresolution eigenimages. pages 2086–2093.