

Robotik III

Merkmalsextraktion, Objekt- erkennung und Lageschätzung

Prof. Dr.-Ing. Tamim Asfour

Prof. Dr.-Ing. Rüdiger Dillmann

Dr.-rer.nat Eren Erdal Aksoy

Dipl.-Inform. Pascal Meißner



Forschungszentrum Karlsruhe
in der Helmholtz-Gemeinschaft



Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

- Korrelationsfunktionen
- Eckendetektoren
 - Moravec-Operator
 - Harris Corner Detector
 - Good Features To Track
- Einfache Deskriptoren
- Aktuelle Merkmale (Detektor + Deskriptor)
 - Scale Invariant Feature Transform (SIFT)
 - Speeded Up Robust Features (SURF)
 - Effiziente Punktmerkmale
- Maximally Stable Extremal Regions (MSER)

■ Objekterkennung auf der Basis von Punktmerkmalen

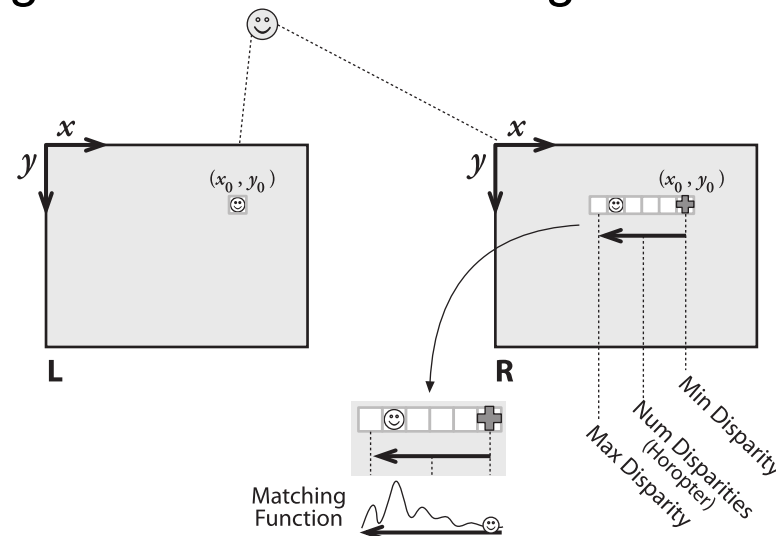
- Matching von Merkmalen
- „Bag of Features“
- RANSAC
- Methode der kleinsten Quadrate (Least Squares)

■ 6-DoF Lageschätzung

- Monokular
- Stereo

Korrelationsfunktionen I

- Ziel: Vergleich zweier Bildausschnitte
- Anwendungen
 - Lösung des Korrespondenzproblems bei Stereokamerasystemen
 - Objekterkennung
- Probleme
 - Invarianz bzgl. Helligkeitsveränderungen
 - Invarianz bzgl. Rotation / Skalierung / Affintransformation



[Bradski 08]

- Allgemeiner Ansatz:
Betrachtung der Grauwerte von zwei quadratischen Bildausschnitten
- Allgemeine Formel für nicht normalisierende Korrelationsverfahren in zwei Bildern mit Verschiebung (d_u , d_v):

$$c(I_1, I_2, u, v, d_u, d_v) = \sum_{i=-k}^k \sum_{j=-k}^k f(I_1(u+i, v+j), I_2(u+i+d_u, v+j+d_v))$$

- Suche auf dem Raum der Verschiebungen
- Die Funktion f ist durch Korrelationsverfahren bestimmt

■ Sum of Squared Differences (SSD)

- Quadrierte Euklidische Metrik
- Anfällig gegenüber Ausreißern
- Keine Invarianz bzgl. Helligkeitsunterschieden

$$SSD(I_1, I_2) = \sum_u \sum_v (I_1(u, v) - I_2(u, v))^2$$

■ Sum of Absolute Differences (SAD)

- Manhattan - Metrik
- Weniger anfällig gegenüber Ausreißern
- Keine Invarianz bzgl. Helligkeitsunterschieden

$$SAD(I_1, I_2) = \sum_u \sum_v |I_1(u, v) - I_2(u, v)|$$

- Erweiterungen um konstante Helligkeitsänderungen zu normalisieren.
 - Subtraktion des Mittelwertes
 - Bezeichnet als „Zero-Mean“-Normalisierung
 - Beispiel: Zero-Mean Sum of Absolute Differences (ZSAD)
 - erzielt Normalisierung bzgl. konstant additiver Helligkeitsänderungen
 - Division durch die Frobeniusnorm
 - Bezeichnet als „Normalized“
 - Beispiel: Normalized Sum of Squared Differences (NSSD)
 - erzielt Normalisierung bzgl. konstant multiplikativer Helligkeitsänderungen
 - Kombination beider Methoden
 - Beispiel: Zero-Mean Normalized Sum of Squared Diff. (ZNSSD)

Moravec-Operator I

- 1977 von Hans P. Moravec entwickelt
- Ziel:
Ähnliche Bereiche in aufeinanderfolgenden Kamerabildern wiederfinden.
- Hierzu wurde das Konzept der Punktmerkmale (engl. *Interest Points*) eingeführt
- Ein Punktmerkmal wird als ein Punkt definiert, von dem aus sich die Intensität in alle Richtungen stark verändert
- Nutzen von Autokorrelationsfunktion.

■ Vorgehensweise

- Betrachtung eines kleinen, quadratischen Fensters (z.B. 3×3 oder 5×5) mit dem Punkt (u, v) als Zentrum
- Verschieben vom Fenster mit anschließendem Vergleich in vier vordefinierte Richtungen (horizontal, vertikal, entlang Diagonalen).
- Als Maß für den Unterschied zwischen Ursprungsfenster und Verschobenem wird SSD (Sum of Squared Differences) verwendet:

$$D(u, v, s, t) = \sum_{(u_i, v_i) \in W(u, v)} (I(u_i + s, v_i + t) - I(u_i, v_i))^2$$

$W(u, v)$ bezeichnet das quadratische Fenster mit Zentrum (u, v)
(s, t) aus { (1, 0), (0, 1), (1, 1), (-1, 1) }

■ Folgende Fälle können auftreten:

1. Der Wert von D ist für alle untersuchten Verschiebungen gering
⇒ Das untersuchte Fenster liegt in einem (nahezu) homogenen Bereich
2. Der Wert von D ist für Verschiebungen längs einer bestimmten Richtung R aus S gering, für Verschiebungen senkrecht zu R jedoch hoch.
⇒ Das untersuchte Fenster enthält eine Kante, die entlang R verläuft.
3. Der Wert von D ist für alle untersuchten Verschiebungen hoch.
⇒ Das untersuchte Fenster enthält eine “Ecke”, also ein Punktmerkmal.

■ Algorithmus

- Das gesamte Bild wird abgesucht
- Maß für Eignung eines Punktes als Punktmerkmal soll bei Fall 1 und 2 kleine Werte und bei Fall 3 große Werte liefern:

$$M(u, v) = \min_{(s, t) \in S} D(u, v, s, t)$$

Eingabe: Graustufenbild $I(u, v)$, Schwellwert t

Ausgabe: Menge M der berechneten Punktmerkmale

$M := \emptyset$

for all pixels (u, v) in I **do**

$m := \text{unendlich}$

for all (s, t) in S **do**

$m := \min \{ m, D(x, y, s, t) \}$

if $m \geq t$ **then**

$M := M \cup \{ (u, v) \}$

return M

■ Nachteile

■ Anisotropische Antwort des Operators

Das Ergebnis des Moravec-Operators ist richtungsabhängig. Da nur vier vordefinierte Richtungen abgeprüft werden, kann das Ergebnis nicht invariant gegenüber beliebigen Rotationen sein.

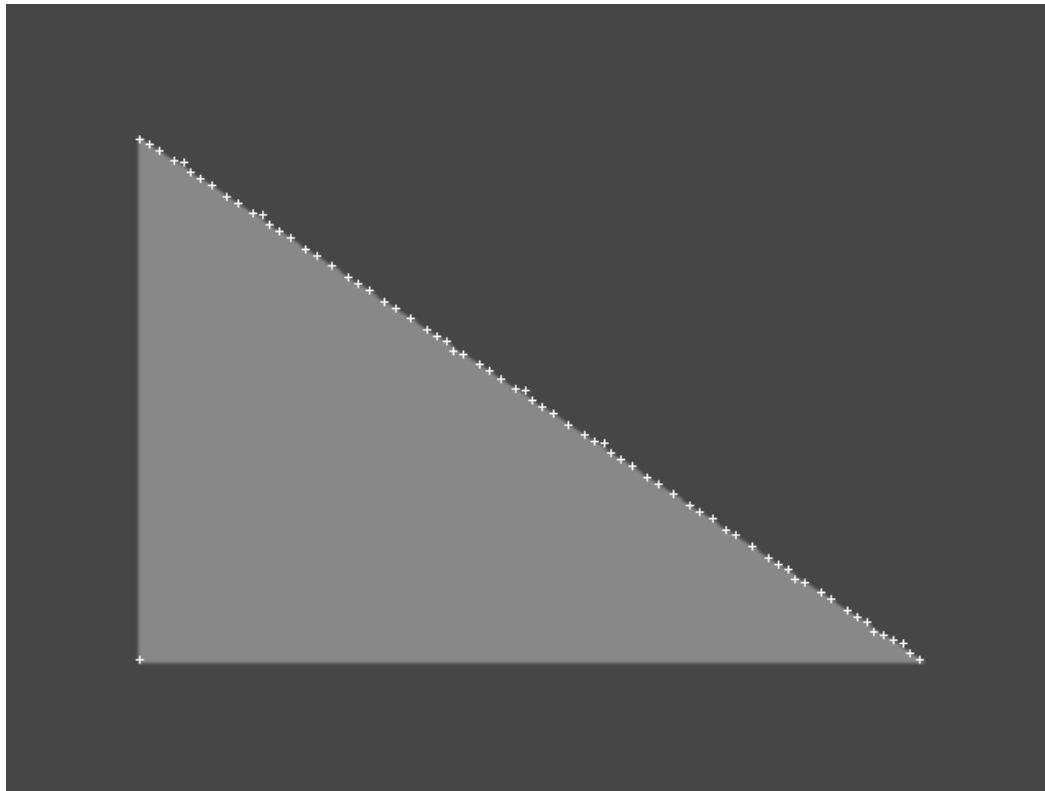
■ Verrauschte Antwort des Operators

Da das Fenster quadratisch und binär ist (und nicht kreisförmig), besitzen auch Pixel in den Ecken das gleiche Gewicht und können das Ergebnis verfälschen.

■ Starke Antwort bei Kanten

Der Operator findet Ecken bei Kanten, die nicht exakt entlang der vordefinierten Verschiebungsrichtungen verlaufen.

- Typisches Beispiel für die Nachteile des Moravec-Operators



Harris Corner Detector I

- 1988 von Chris Harris und Mike Stephens entwickelt
- Ziel:
Ersetzung der vier vordefinierten Richtungen durch ein Verfahren, welches eine feinere Schrittweite erlaubt
- Ansatz:
Ausgangsbasis ist die Taylorentwicklung erster Ordnung der Bildfunktion

- Taylorentwicklung erster Ordnung:

$$I(u + s, v + t) \approx I(u, v) + (I_x(u, v) \quad I_y(u, v)) \begin{pmatrix} s \\ t \end{pmatrix}$$

I_x und I_y bezeichnen die Richtungsableitungen, welche z.B. mit dem Prewitt- oder Sobel-Operator berechnet werden können. Stehen in gem. Skalarprodukt mit Verschiebungsparametern (s,t).

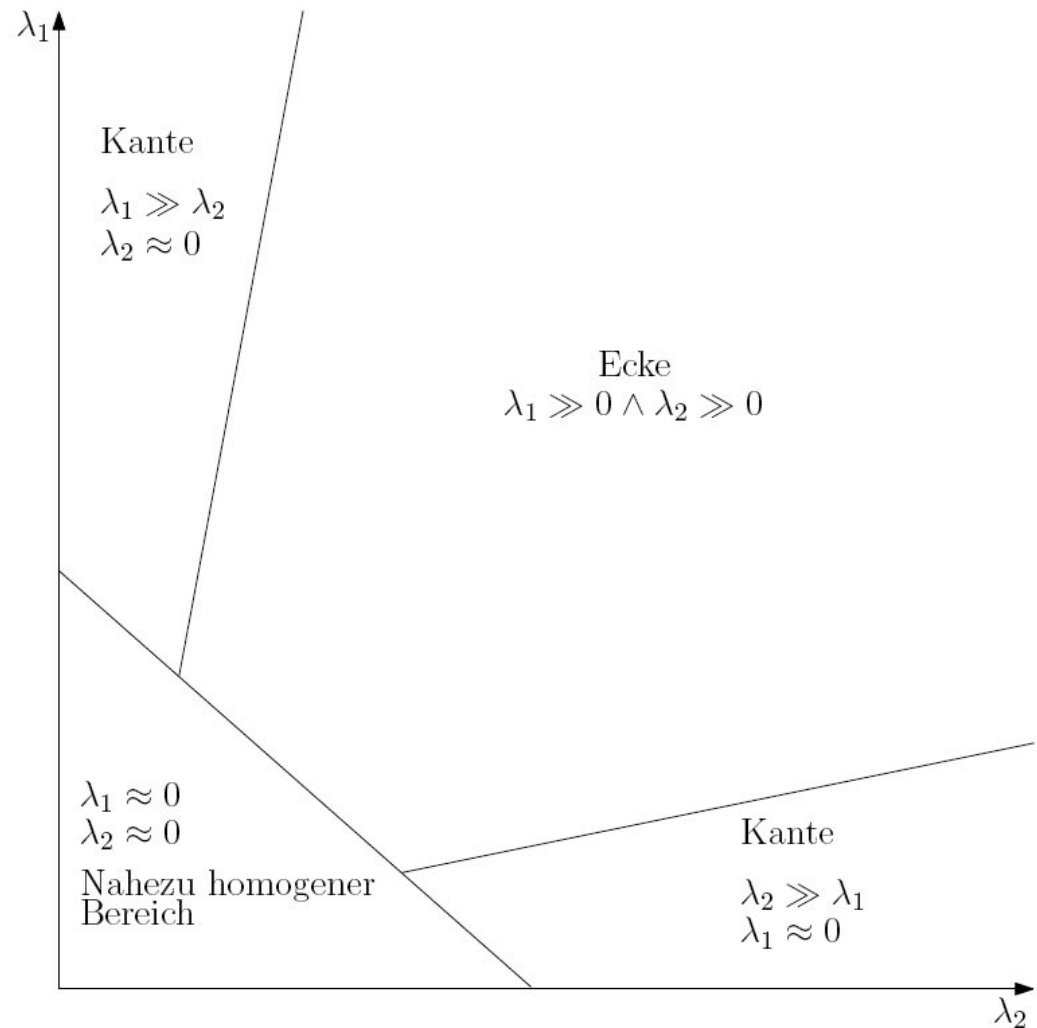
- Einsetzen in $D(u, v, s, t)$ (Moravec-Operator) ergibt:

$$\begin{aligned}
 D(u, v, s, t) &= \sum (I(u_i + s, v_i + t) - I(u_i, v_i))^2 \\
 &\approx \sum \left(I(u_i, v_i) + (I_x(u_i, v_i) I_y(u_i, v_i)) \begin{pmatrix} s \\ t \end{pmatrix} - I(u_i, v_i) \right)^2 \\
 &= \sum \left((I_x(u_i, v_i) I_y(u_i, v_i)) \begin{pmatrix} s \\ t \end{pmatrix} \right)^2 \\
 &= \sum \left(\begin{pmatrix} s & t \end{pmatrix} \begin{pmatrix} I_x^2(u_i, v_i) & I_x(u_i, v_i) I_y(u_i, v_i) \\ I_x(u_i, v_i) I_y(u_i, v_i) & I_y^2(u_i, v_i) \end{pmatrix} \begin{pmatrix} s \\ t \end{pmatrix} \right) \\
 &= \begin{pmatrix} s & t \end{pmatrix} M(u, v) \begin{pmatrix} s \\ t \end{pmatrix} \text{ mit } M(u, v) = \begin{pmatrix} \sum I_x^2(u_i, v_i) & \sum I_x(u_i, v_i) I_y(u_i, v_i) \\ \sum I_x(u_i, v_i) I_y(u_i, v_i) & \sum I_y^2(u_i, v_i) \end{pmatrix}
 \end{aligned}$$

- Approximation von Autokorrelationsfunktion durch quadratische Form über Matrix M (d.h. Gradienten).
- M enthält Information über die Struktur des Bildbereichs.
 - Ellipse konstanter Grauwertänderung.
- Aus Betrachtung der Eigenwerte von M (Kehrwerte der quadrierten Halbachsen) lässt sich diese Struktur ableiten:
 - Fall 1: Beide Eigenwerte sind klein; große Ellipse.
Das Fenster liegt in einem (nahezu) homogenen Bereich.
 - Fall 2: Ein Eigenwert ist groß und der andere klein; gestreckte Ellipse. Das Fenster enthält eine Kante.
 - Fall 3: Beide Eigenwerte sind groß; kleine Ellipse.
Das Fenster enthält eine "Ecke", also ein Punktmerkmal.

Harris Corner Detector V

- Analyse der Rolle der Eigenwerte λ_1, λ_2



- Anstatt einer expliziten Eigenwertzerlegung schlagen Harris und Stephens deren Zusammenfassen in Cornerness-Maß vor:

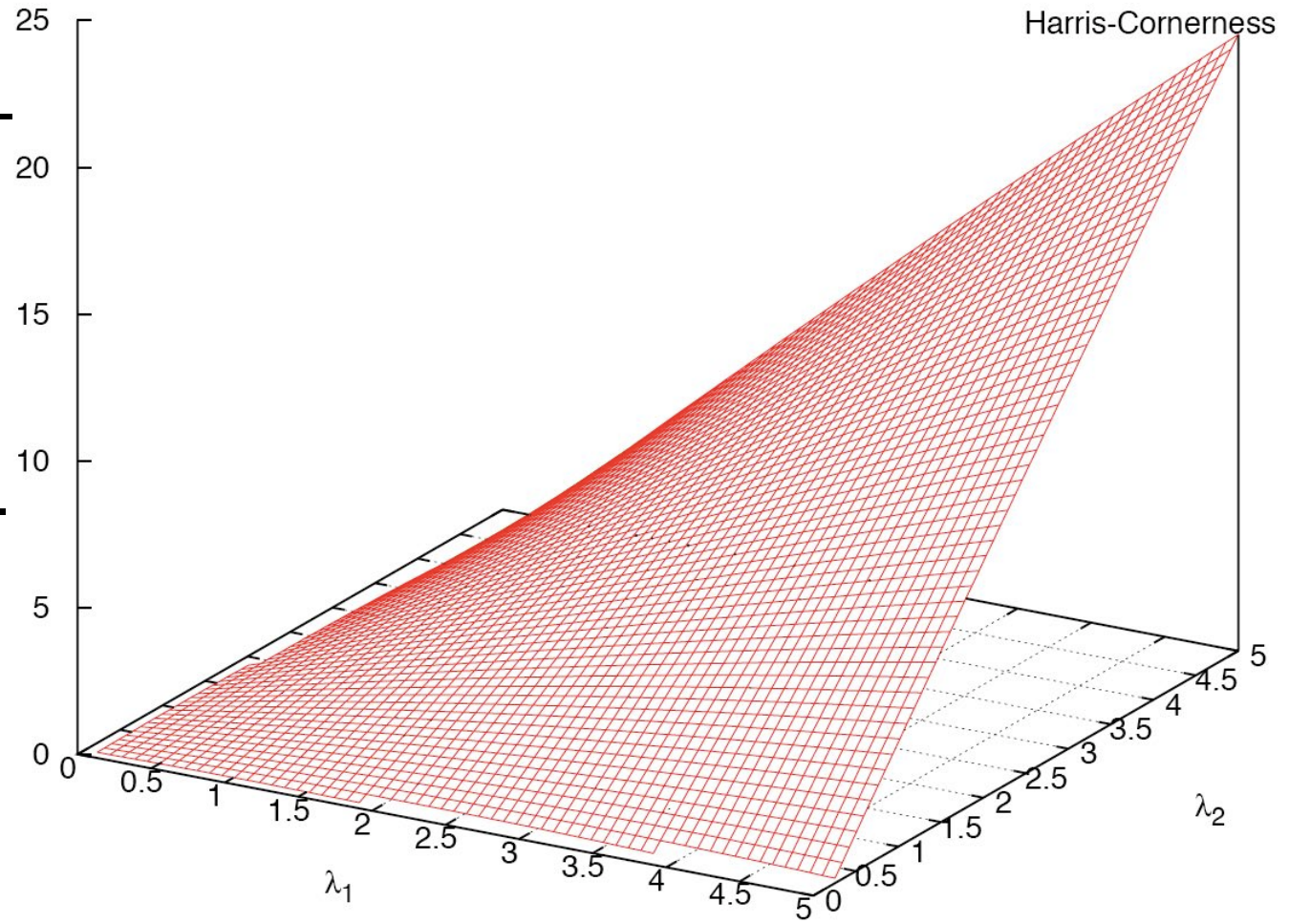
$$\begin{aligned}C(u, v) &= \lambda_1 \lambda_2 - \kappa (\lambda_1 + \lambda_2)^2 \\&= \det M(u, v) - \kappa (\text{spur } M(u, v))^2 \\&= m_{11}m_{22} - m_{12}m_{21} - \kappa (m_{11} + m_{22})^2\end{aligned}$$

- κ ist eine empirisch zu bestimmende Konstante. Üblicherweise werden Werte aus $[0.04, 0.06]$ verwendet.

Harris Corner Detector VII

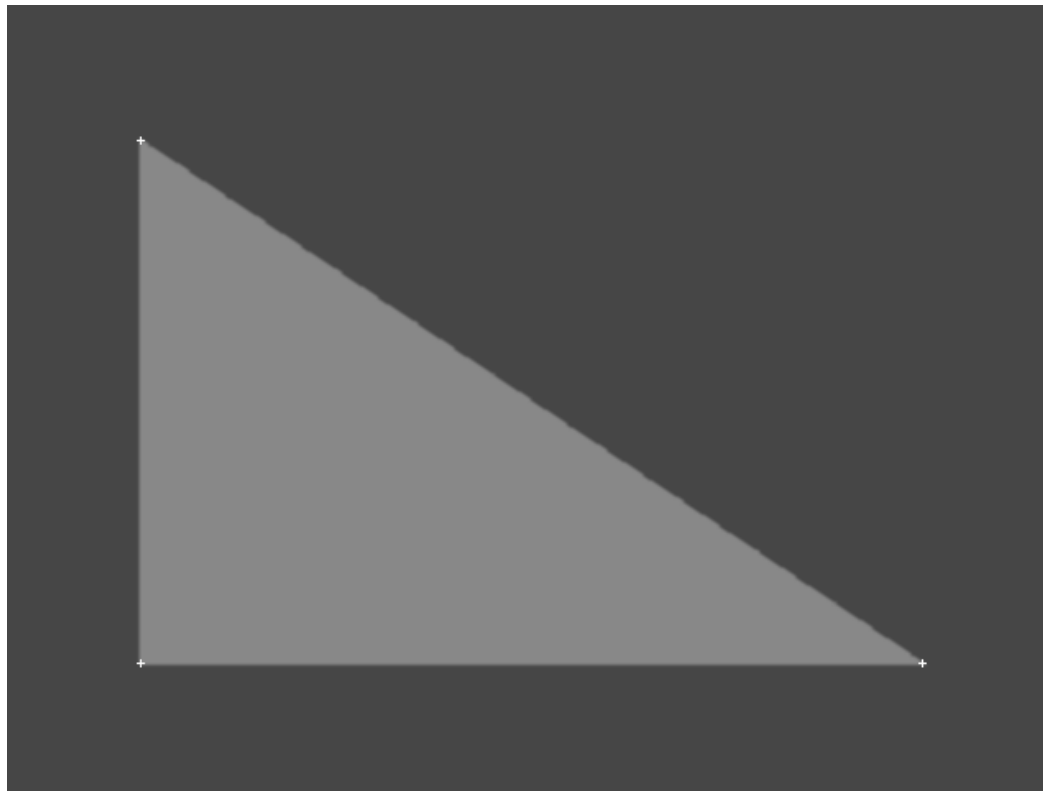
- Harris Cornerness-Maß für $\kappa = 0.04$

- Ist zu maximieren.



Harris Corner Detector VIII

- Beispiel: Harris Corner Detector behebt die Probleme des Moravec-Operators



■ Beispielanwendung für eine reale Szene



Good Features To Track

- 1994 von Jianbo Shi und Carlo Tomasi entwickelt
- Abstimmung der Merkmale auf deren eigenen Tracking-Algorithmus
- Verbesserte Fassung des Harris Corner Detektors
- Explizite Berechnung der Eigenwerte der Matrix $M(u,v)$
- Bedingung für Merkmale: $\min(\lambda_1, \lambda_2) > \lambda$ (Schwellwert), statt Schwellwert auf Cornernessfunktion. Analogie zu Moravec Operator.

■ Einfachster Ansatz:

- Beschreibung der lokalen Umgebung durch ein Quadrat um den zu betrachtenden Bildpunkt. Bildausschnitt als Merkmal.
- Vergleich zweier Ausschnitte durch Anwendung einer Korrelationsfunktion

■ Vorteile

- Einfach zu realisieren
- Geringer Rechenaufwand

■ Nachteil

- Keine Invarianz bzgl. Rotation und Skalierung

- Erweiterung nach [Lepetit et al., 2004]
 - Modellierung jedes Bildausschnitts durch ein sog. „view set“
 - Generierung von z.B. 100 Ansichten des lokalen Ausschnitts, welche den Raum der möglichen Erscheinungen abdecken
 - Skalierung
 - Rotation
 - Verzerrung
 - Abgleich (Matching) eines Bildausschnitts aus der aktuellen Szene mit der Datenbank durch Vergleich mit **allen** Ansichten aller eingelernten Bildausschnitte

- Ansichten können synthetisch generiert werden durch
 - Verzerrungsabbildung (Affine Transformation)
 - Texturiertes 3D-Modell und Rendering



100 synthetisch generierte Ansichten eines 32×32 Pixel großen Merkmals

■ Nachteil:

- Hoher Speicheraufwand: $100 \cdot 32 \cdot 32 = 102400$ Bytes / Merkmal
- Hoher Rechenaufwand: 102400 Vergleiche pro Merkmal

■ Lösungsansatz

- Komprimierung der Ansichten mithilfe der Hauptkomponentenanalyse (PCA)
- Reduktion von $32 \cdot 32 = 1024$ auf 20 Dimensionen
- Dadurch Beschleunigung und Reduktion des Speicheraufwands um in etwa Faktor 50

- Zu unterscheiden sind
 - Merkmalsdetektoren
 - Merkmalsdeskriptoren
- Nachfolgend werden Verfahren beschrieben, welche sowohl den Detektor als auch den Deskriptor spezifizieren
 - Scale Invariant Feature Transform (SIFT)
 - Speeded Up Robust Features (SURF)
 - Maximally Stable Extremal Regions (MSER)

Scale Invariant Feature Transform I

- 1999 von David G. Lowe publiziert, Verfeinerung des Verfahrens 2004
- Sehr populäre Punktmerkmale, vor allem in der Robotik
- Abkürzung: SIFT
- Ansatz
 - Generieren verschiedener Bildauflösung äquivalent zu sukzessivem Gaußglätten und Unterabtasten (sonst: Aliasing)
 - Detektor arbeitet auf der Basis einer Skalenraum-Analyse mit Difference of Gaussians (DoG)
 - Deskriptor besteht aus in Teil-Quadrate unterteiltem Gradientenhistogramm
- Das Verfahren erreicht Invarianz bzgl. Rotation, Skalierung und in einem kleinen Bereich bzgl. Verzerrung

■ Merkmalsdetektor

- Skalenraum wird aufgebaut durch Faltung mit der Gauß-Funktion (Tiefpassfilter mit Parameter σ wie aus letzter Vorlesung)

$$L(u, v, \sigma) = G(u, v, \sigma) * I(u, v)$$

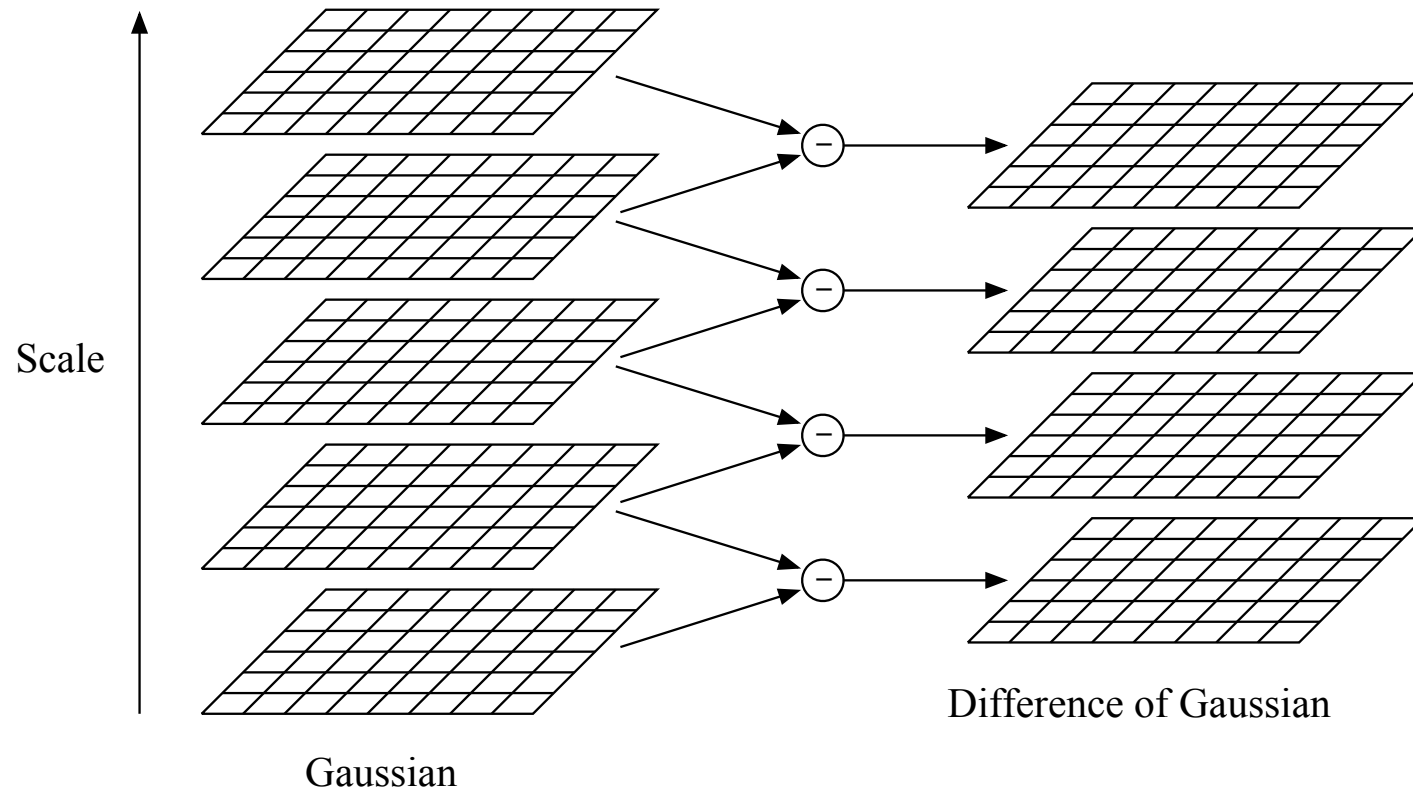
- Stabile Merkmalspunkte werden durch Bestimmung von Extrema der DoG-Funktion berechnet (Difference of Gaussians):

$$D(u, v, \sigma) = L(u, v, k\sigma) - L(u, v, \sigma)$$

- DoG-Funktion ist eine effiziente Approximation des skalen-normalisierten LoG-Operators (Laplacian of Gaussian), der ein Blob Detektor ist. LoG erkennt Blob nur bei passender Skalierung.

Scale Invariant Feature Transform III

- Raum der Skalen (Bilder) wird in sog. *octaves* eingeteilt
- Jede *octave* enthält s Intervalle mit $k = 2^{1/s}$

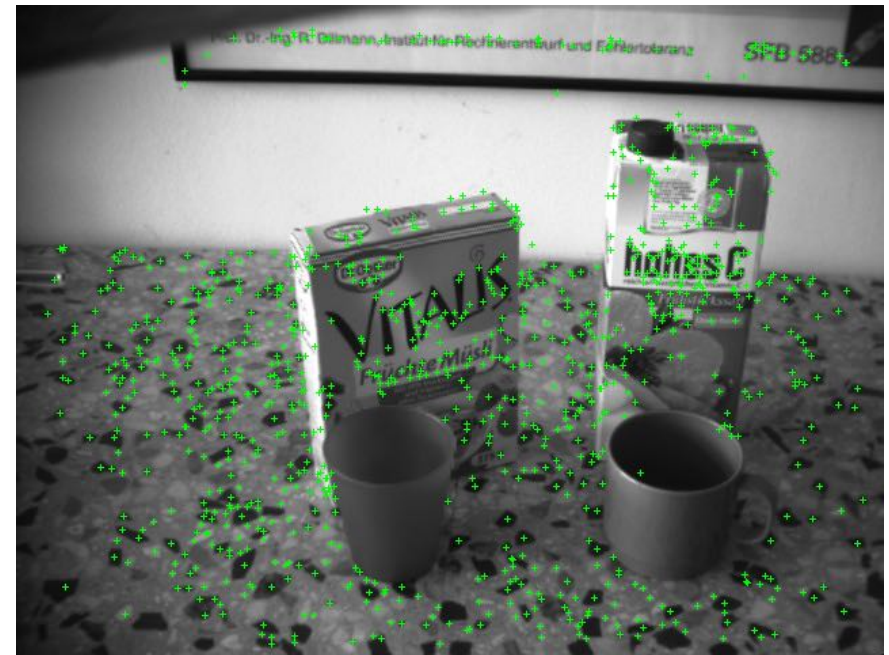


- Innerhalb einer *octave* wird geglättet statt skaliert und in DoG Bildern nach lokalen Extrema gesucht
- Zwischen aufeinanderfolgenden octaves werden die Bilder auf halbe Breite und halbe Höhe verkleinert
- Ein lokales Extremum ist ein lokales Minimum oder lokales Maximum unter Einbezug der
 - 8 direkten Nachbarn
 - jeweils 9 Nachbarn aus den benachbarten DoG Bildern (nächste größere und kleinere Skala)
- Insgesamt werden folglich $2 \cdot 9 + 8 = 26$ Nachbarn überprüft
- Damit s Ebenen bearbeitet werden können, müssen $s + 3$ Tiefpass-gefilterte Bilder innerhalb einer *octave* vorliegen

■ Harris Corner Detector vs. SIFT Keypoint Detector



Harris



SIFT

■ Deskriptor

- Das Tiefpass-gefilterte Bild entsprechend der Skala, auf welcher der Keypoint detektiert wurde, wird für die Berechnung des Deskriptors verwendet
- Zunächst: Berechnung der dominanten Richtung in Fenster

■ Schritt 1:

Berechnung der Gradienten in einem quadratischen Fenster (gewichtet mit Gauß-Funktion)

■ Schritt 2:

Berechnung des Betrags und der Richtung

$$m = \sqrt{g_x^2 + g_y^2} \quad \theta = \arctan \frac{g_y}{g_x}$$

- Schritt 3:
Berechnung eines Gradientenhistogramms mit 10°
Diskretisierung (36 Bins) über θ und m als Gewichtung
- Schritt 4:
Berechnung des globalen Maximums. Ausgabe dessen
und aller Einträge mit Werten $\geq 85\%$ des Maximums sind
Richtungen
- Schritt 5:
Für jede der berechneten Richtungen wird das Bild rotativ
entsprechend der Richtung ausgerichtet
(Rotationsinvarianz) und ein Deskriptor berechnet

■ Nun: Berechnung des Deskriptors

- Gradientenhistogramm in Sub-Fenstern mit jeweils 8 Richtungen
- Beispiel: 8x8 Fenster mit 4 Sub-Fenstern

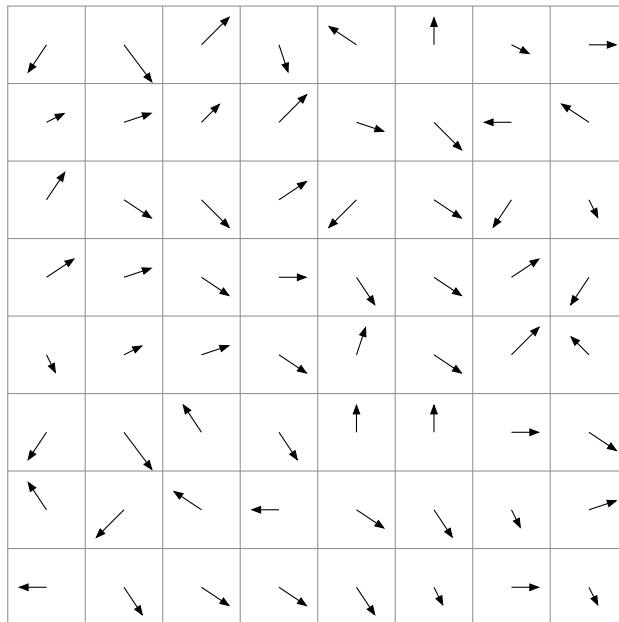
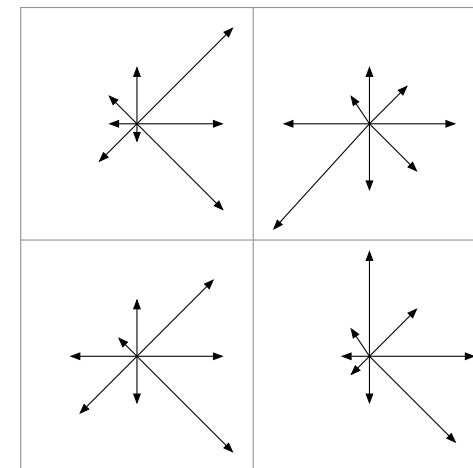


Image gradients



Keypoint descriptor

■ Üblicherweise verwendeter SIFT-Deskriptor

- 16×16 großes Fenster
- Sub-Fenster mit Größe $4 \times 4 \Rightarrow 16$ Sub-Fenster
- $16 \cdot 8$ (Bins) = 128 dimensionaler Merkmalsvektor
- Repräsentiert als float-Vektor $\Rightarrow$ 512 Bytes

■ Normalisierung

- Merkmalsvektor wird auf Länge 1 normalisiert
- Merkmalsvektor ist automatisch invariant bzgl. konstant additiven Helligkeitsunterschieden, da Gradienten-basiert
- Werte > 0.2 werden abgeschnitten (dann re-normalisiert)

■ Beispiel: Eingezeichnete SIFT-Merkmale inkl. Richtungen

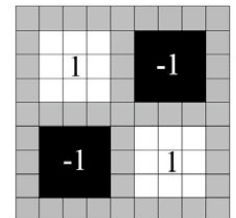
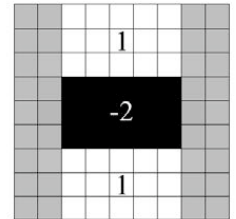


Speeded Up Robust Features

- 2006 von Herbert Bay, Tinne Tuytelaars und Luc Van Gool entwickelt

- Abkürzung: SURF

- Motivation:
Geschwindigkeitssteigerung gegenüber SIFT bei vergleichbaren oder verbesserten Eigenschaften



- Ansatz:
 - Verwendung von Rechtecksfilttern (beschleunigte Berechnung durch Integral Images) anstatt DoG
 - Deskriptor um Faktor 2 verkleinert: 64 Dimensionen

- Kombination von Eckendetektor und Deskriptor
 - Rechenintensive Skalenraumanalyse entfällt
 - Skalierungsinvarianz muss durch explizites Sampling auf vordefinierten Skalen erreicht werden
 - Ermöglicht Echtzeitverarbeitung (30 fps und schneller)
 - Beispiele
 - FAST-Detektor + SIFT/Ferns-Deskriptor [Wagner et al., 2008]
 - Harris-Eckendetektor + SIFT-Deskriptor [Azad, 2008], [Azad et al. 2009]

Maximally Stable Extremal Regions I

- 2002 von Jiri Matas et al. entwickelt
- Abkürzung: MSER
- Idee:
Stabile, homogene Regionen (mit kontrastierendem Rand) detektieren, ohne Vorgabe eines Schwellwerts
- Ansatz:
Analyse der Folge aller durch Schwellwertfilterung berechneter Binärbilder (Schwellwerte 0, ..., 255). Finden von Bildregionen, die über hohe Anzahl von Schwellwerten ihre Form & Ausdehnung nicht ändern.

■ Veranschaulichung

- Aufsteigende Folge der Schwellwerte: 0, ..., 255
- Bild zunächst weiß
- Schwarze Regionen wachsen

■ Maximal Regions (MSER+):

- Berechnung auf Original-Graustufenbild
- Verwendeter Schwellwertoperator: $\geq$

■ Minimal Regions (MSER-):

- Berechnung auf invertiertem Graustufenbild
- Verwendeter Schwellwertoperator: $\geq$ (wie bei Maximal Regions)

Maximally Stable Extremal Regions III

■ Beispiel: Berechnete MSER+



- Engl.: *feature matching*
- Problemstellung:
Gegeben eine Menge (Datenbank) von Merkmalen und einem aktuell zu betrachtenden Merkmal, Bestimmung des zu diesem Merkmal korrespondierenden Merkmals aus der Menge.
- Ansatz
Vergleich der Merkmalsvektoren mit entsprechendem Fehler- bzw. Korrelationsmaß.

Matching von Merkmalen II

■ Einfachster Ansatz: NearestNeighbor

■ Algorithmus

- Eingabe: Anfrage-Merkmal $\mathbf{x}$, Menge von Merkmalen M ($|M| \geq 1$)
- Ausgabe: Bester “Match” aus M zu $\mathbf{x}$

- $d_{\min} := \infty$
- Für jedes Merkmal $m \in M$ berechne euklidische Distanz:
 $d := |\mathbf{x} - \mathbf{m}|$
 - Falls $d < d_{\min}$:
 - $d_{\min} := d$
 - $\mathbf{m}_{\min} := \mathbf{m}$
- Gebe $\mathbf{m}_{\min}$ zurück

■ Erweiterung: k-NearestNeighbor

■ Algorithmus

- Eingabe: Anfrage-Merkmal $\mathbf{x}$, Menge von Merkmalen M ($|M| \geq k$)
- Ausgabe: Liste der k besten “Matches” aus M zu $\mathbf{x}$

■ $d_{\text{entry}} := \infty$

■ Für jedes Merkmal $m \in M$ berechne euklidische Distanz:
 $d := |\mathbf{x} - \mathbf{m}|$

■ Falls $d < d_{\text{entry}}$:

- Entferne letzten Eintrag $\mathbf{l}_k$ aus L ($\mathbf{l}_k = \operatorname{argmax}_{\mathbf{l} \in L} |\mathbf{x} - \mathbf{l}|$)
- Füge $\mathbf{m}$ der sortierten Liste L hinzu
- Falls $|L| = k$ dann: $d_{\text{entry}} := |\mathbf{x} - \mathbf{l}_k'|$ ($\mathbf{l}_k' = \operatorname{argmax}_{\mathbf{l} \in L} |\mathbf{x} - \mathbf{l}|$)

■ Gebe Liste L zurück

- k-Nearest Neighbor: Auswertung der berechneten Liste
 - Übliche Anwendung: Klassifikation
- Variante von 2-Nearest Neighbor für Matching von Merkmalen
 - Wie bei herkömmlichem 2-Nearest-Neighbor: Berechnung der zwei nächsten Nachbarn $\mathbf{m}_1$ und $\mathbf{m}_2$ für ein Anfrage-Merkmal $\mathbf{x}$
 - Entscheidung über Akzeptanz:
 - Vergleich des Verhältnisses der Distanzen $d_1 := |\mathbf{x} - \mathbf{m}_1|$ und $d_2 := |\mathbf{x} - \mathbf{m}_2|$: $r := \min(d_1, d_2) / \max(d_1, d_2)$
 - r muss **unter** einem Schwellwert t mit $0 < t < 1$ liegen

■ Matching-Strategien

- Zu jedem Merkmal im Anfrage-Bild ordne ein entsprechendes Merkmal aus der Datenbank zu (Variante: Clustering der Trainings-Merkmale).
- Zu jedem Merkmal aus der Datenbank ordne ein entsprechendes Merkmal aus dem Anfrage-Bild zu.
- Kombination der beiden o.g. Strategien

- Ziel:
Erkennung und 2D-Lokalisierung von Objekten auf der Basis von Punktmerkmalskorrespondenzen



Zu analysierende Szene



Trainingsbild

- Einfachster Ansatz: “bag of features”
 - “Zählen” der Korrespondenzen für eine gegebene Objektrepräsentation bestehend aus einer Menge von Punktmerkmalen
 - Zuordnung erfolgt ausschließlich auf der Basis der Merkmalsvektoren
 - Objekt gilt als erkannt, wenn eine erforderliche Mindestanzahl an Korrespondenzen berechnet wurde
- Vorteil: Einfach zu realisieren
- Nachteil: Nicht robust, fehleranfällig

- Problem:
Auftreten falscher Korrespondenzen
- Lösung:
Ausnutzung der örtlichen Beziehungen der Merkmale
- Verfahren
 - RANSAC
 - Methode der kleinsten Quadrate (Least Squares)

- RANdom SAmple Consensus
- 1981 von Martin A. Fischler und Robert C. Bolles publiziert
- Iteratives Verfahren, um eine bezüglich eines gegebenen Modells konsistente Untermenge einer Menge robust zu bestimmen.
- Haupteinsatzzweck: Filtern von Ausreißern
- Diverse Einsatzgebiete

■ Gegeben:

- Ein Modell, welches mindestens n Datenpunkte benötigt, um eine Instanz eindeutig zu bestimmen
- $|P| \geq n$ Datenpunkte

■ Algorithmus

- Führe die Schritte 1-3 höchstens k Mal aus:
 1. Wähle zufällig eine Untermenge $S \subset P$ mit $|S| = n$ und berechne auf der Basis von S eine Instanz des Modells M
 2. Bestimme die Untermenge $S^* \subset P$ aller Punkte, welche der Instanz M des Modells innerhalb einer vorgegebenen Toleranz t genügen. S^* wird als „Consensus“ von S bezeichnet.
 3. Falls $|S^*|$ einen Schwellwert m überschreitet, gebe M, S^* zurück.

- Anwendung für die Filterung falscher Korrespondenzen
 - Als Modell wird eine Affintransformation oder Homographie verwendet
 - Mindestens 3 bzw. 4 Punkte sind notwendig, um diese aufzustellen: $n = 3$ (Affintransformation), $n = 4$ (Homographie)
 - Fehlertoleranz t : z.B. 5 Pixel Abstand
- 2D-"Homography"
(Formulierung mit homogenen Koordinaten):

$$\begin{pmatrix} u' \cdot s \\ v' \cdot s \\ s \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

- 2D-Affintransformation: $a_7 = a_8 = 0$

- Anwendung für die Berechnung einer Homographie bzw. Affintransformation
- Gegeben:
Menge von Punktkorrespondenzen
 $\{ \mathbf{x}_i = (u_i, v_i), \mathbf{x}_i' = (u_i', v_i') \}$ mit $i \in \{ 1, \dots, n \}$
- Ausmultiplizieren der Homography-Definition ergibt:

$$a_1 u + a_2 v + a_3 - a_7 u u' - a_8 v u' = u'$$

$$a_4 u + a_5 v + a_6 + a_7 u v' - a_8 v v' = v'$$

- LGS für Bestimmung einer Homographie ($n \geq 4$):

$$\begin{pmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1 u_1' & -v_1 u_1' \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -u_1 v_1' & -v_1 v_1' \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_n & v_n & 1 & 0 & 0 & 0 & -u_n u_n' & -v_n u_n' \\ 0 & 0 & 0 & u_n & v_n & 1 & -u_n v_n' & -v_n v_n' \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{pmatrix} = \begin{pmatrix} u_1' \\ v_1' \\ \vdots \\ u_n' \\ v_n' \end{pmatrix}$$

- LGS für Bestimmung einer Affintransformation ($n \geq 3$):

$$\begin{pmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_1 & v_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_n & v_n & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_n & v_n & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{pmatrix} = \begin{pmatrix} u_1' \\ v_1' \\ \vdots \\ u_n' \\ v_n' \end{pmatrix}$$

■ Bestimmung Affintransformation

- Robust
- Kann die realen Verhältnisse nicht vollständig beschreiben

■ Bestimmung Homographie

- Instabiler als Bestimmung der Affintransformation
- Anfällig gegenüber falschen und vertauschten Punktkorrespondenzen
- Kann bei planarer Annahme die realen Verhältnisse vollständig beschreiben

- Beispiel:
Verbesserung der 2D-Lokalisierung durch Berechnung einer Homographie

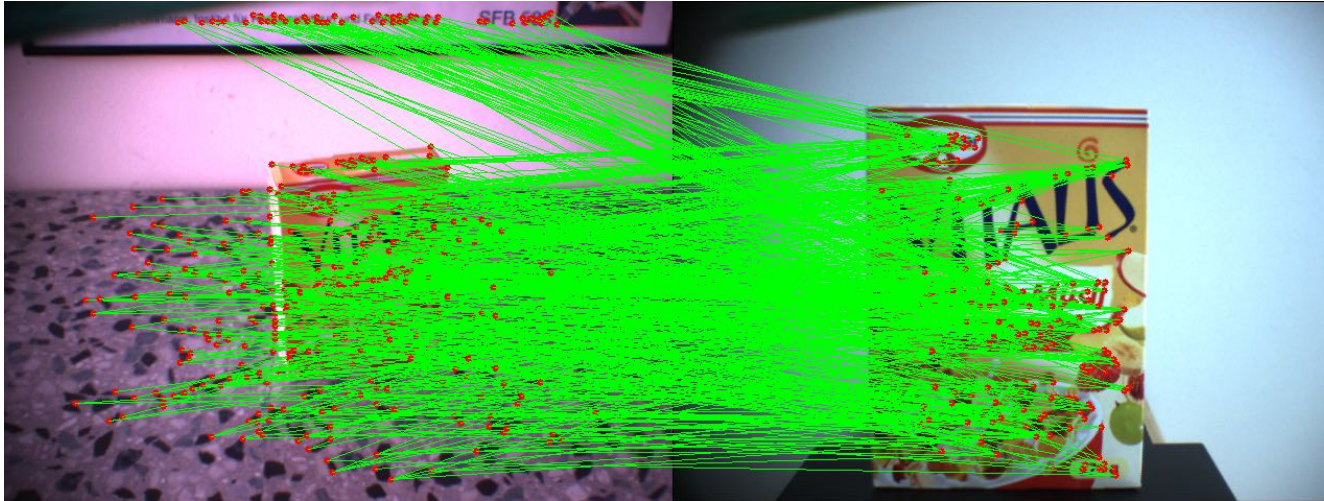


Ergebnis Affintransformation

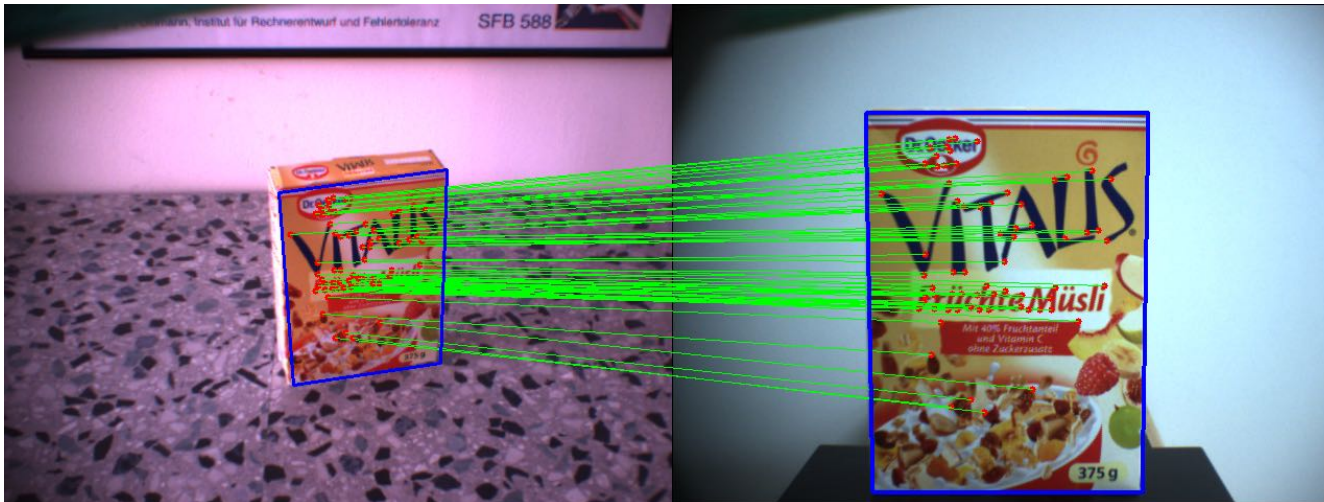


Ergebnis Homography

Linear Least Squares VI



Ungefilterte
Korrespondenzen
(toleranter
Schwellwert für
Matching)



Gefilterte
Korrespondenzen
nach RANSAC-
Verfahren und
Bestimmung der
Homographie
+ Ergebnis der 2D-
Lokalisierung (blauer
Rahmen, links)

6-DoF Lageschätzung

■ Problemstellung:

Gegeben ein 3D-Modell des Objektes, Bestimmung der 3D-Transformation (bestehend aus Rotation und Translation), welche die 6-DoF Lage (Degrees of Freedom) des Objektkoordinatensystems (Modell) im Weltkoordinatensystem beschreibt

■ Nachfolgend:

Weltkoordinatensystem = Kamerakoordinatensystem linke Kamera

■ Grundsätzlich zwei Alternativen

■ Monokular

- Operiert auf der Basis von 2D-3D Punktkorrespondenzen

■ Stereokamerasystem

- Operiert auf der Basis von 3D-Punkten aus Stereo-Triangulation

■ Grundlage

- 2D-3D Punktkorrespondenzen
 - 3D-Punkte des Modells (Weltkoordinaten)
 - 2D-Punkte aus der aktuellen Ansicht (Bildkoordinaten)
- Diese Korrespondenzen müssen beim Einlernen für das Trainingsbild bzw. die Trainingsbilder bestimmt werden
- Über die bei der Erkennung berechnete Homography können die 2D-3D Punktkorrespondenzen propagiert werden
oder
Propagation durch Tracking der 2D-Punkte, z.B. mit KLT-Tracker (Kanade-Lucas-Tomasi)

- Algorithmen zur Bestimmung der 6 DoF Lage aus 2D-3D Punktkorrespondenzen (sog. PnP-Problem, engl. *Perspective n-Point*)
 - POSIT (Pose from Orthography and Scaling with Iterations)
 - 1992 von Daniel F. DeMenthon und Larry S. Davis publiziert
 - In der Originalversion: 3D-Punkte dürfen nicht koplanar liegen
 - Erweiterung für koplanare 3D-Punkte: [Oberkamp et al., 1996]
 - Weitere Algorithmen hierzu
 - [Lu et al., 2000]
 - [Schweighofer and Pinz, 2006]
 - [Moreno-Noguer et al., 2007]
 - [Schweighofer and Pinz, 2008]

- Mehrere Ansätze, eine Möglichkeit:
 - Berechnung von 3D-Koordinaten für Merkmalspunkte über Korrelation und Stereo-Triangulation. Anschließend:
 - Fitting eines geometrischen 3D-Primitivs
 - Registrierung eines 3D-Objektmodells
- Vorteile
 - Robust aufgrund von Stereo-Triangulation
 - Je nach Aufbau höhere Genauigkeit (insbesondere Tiefe)
- Nachteile
 - Benötigt exakte Stereo-Kalibrierung
 - Ungenauigkeiten bei starken Linsenverzerrungen

Beispiel Lageschätzung

- 3D-Modell nach Anwendung der berechneten Lage als Overlay eingezeichnet



- Vertiefung – SIFT Merkmalsdetektion
 - Vorlesung „Blob Detection“ von S. Lazebnik.
- Dissertation „Visual Perception for Manipulation and Imitation in Humanoid Robots“ von Pedram Azad
 - Kapitel 2 (u.a. Lokale Methoden, POSIT)
 - Kapitel 4 (u.a. Korrelation, Homographie)