

Seminararbeit zum Thema

Disparity and Occlusion Estimation in Multicocular Systems

Von Alexander Oertel

Seminar 3D-Rekonstruktion
LS Praktische Informatik IV
Universität Mannheim

Inhaltsverzeichnis

- I.) Einführung (Seite 3)
- II.) Beschreibung des Disparity Estimation / Occlusion Detection Algorithmus
 - A. Kamera Model (Seite 4)
 - B. Disparity Estimation und Occlusion Detection für 3D-Bildpaare (Seite 5)
 - C. Disparity Estimation und Occlusion Detection für Multiview Bildsequenzen (Seite 8)
 - D. Gleichzeitige Erkennung und Benennung von verdeckten Regionen (Seite 12)
 - E. R-L Disparity Feld und Verdeckungsschätzung (Seite 14)
 - F. Beispiel (Seite 15)

Abstrakt:

Es wird ein effizienter Algorithmus zur Disparity - Schätzung und zur Verdeckungserkennung in multiokularen Systemen präsentiert. Ein ‚dynamic programming‘ – Algorithmus, der sowohl ‚multiview matching‘ – Kosten, als auch einfache geometrische Beschränkungen verwendet, wird benutzt, um die Disparity zu schätzen und verdeckte Gebiete in der ganz linken und in der ganz rechten Sicht zu identifizieren. Ein wichtiger Vorteil des vorgeschlagenen Ansatzes ist, dass die genaue Anzahl der Sichten, in denen ein Punkt erscheint (nicht verdeckt ist) genau bestimmen werden kann. Die erhaltenen Disparity- und Verdeckungsinformationen können verwendet werden, um virtuelle Bilder von zwischenliegenden Standpunkten zu bilden.

I.) Einführung

Drei oder mehr Kameras werden verwendet, um ein multiokulares System für die Generierung von etlichen Bildsequenzen zu erstellen, die man von leicht verschiedenen Standpunkten erhält. Die Kosten der zugehörigen Aufnahme- und Übertragungssysteme steigen beträchtlich mit der Anzahl der Sichtpunkte. Um diese Kosten zu reduzieren müssen virtuelle Bilder von zwischenliegenden Standpunkten generiert werden. Hierbei werden die realen Standpunkte und zusätzliche Disparity - Informationen verwendet.

Effiziente Datenübertragung von Multiview - Bildsequenzen kann erreicht werden, indem man kodierte Bildsequenzen der realen Standpunkte zusammen mit Disparity- oder Tiefeninformationen überträgt. Zwischenliegende Bilder werden dann beim Empfänger erstellt. Dieser verwendet räumliche Interpolation auf der Basis von Lichtbildern und der übertragenen Disparity - oder Tiefeninformationen. Durch die Übertragung zusätzlicher Bitraten, die den Disparity- oder Tiefeninformationen entsprechen, wird die Generierung der zwischenliegenden Bilder erleichtert. Dadurch wird die Komplexität des Empfängers deutlich reduziert. Diesem Vorteil stehen die Kosten der Übertragung der zusätzlichen Bitraten gegenüber, die erheblich sein können. Um die Qualität der erzeugten zwischenliegenden Bilder zu verbessern, muss ein zuverlässiges Disparity - Feld geschätzt werden. Zuerst wird ein Algorithmus zur genauen Auswertung des Disparity - Feldes und zur Erkennung und Einordnung der Verdeckungen entwickelt, der dynamische Programmierung verwendet. Die übertragenen Informationen bestehen aus 1.) einem 3D-Bildpaar 2.) zwei Disparity - Feldern und 3.) einer oder zwei Verdeckungskarten.

Verfahren, die auf dynamischem Programmieren basieren, wurden für die Disparity - Schätzung und gleichzeitige Verdeckungserkennung aus 3D – Bildsequenzen (zwei Sichten) verwendet. Ein bedeutender Vorteil dieser Verfahren ist, dass sie eine globale Lösung der Disparity – Schätzung und der Verdeckungserkennung unter lokalen Beschränkungen bereitstellen.

Das erste Ziel dieses Papers ist die Ausgestaltung eines effizienten Algorithmus, der Disparity und Verdeckung so schätzt, dass die genaue Anzahl der Sichten, in denen jeder Punkt verdeckt (oder nicht verdeckt) ist, bestimmt werden kann. Aus diesem Grunde wird das Grundkonzept der Standard - Stereo - Matching - Algorithmen, das dynamisches Programmieren verwendet, für Multiview Bildsequenzen erweitert und angepasst. Der daraus resultierende Algorithmus vermeidet die Benutzung der Glättungsbeschränkungen und ist rechnerisch effizient, vor Allem wenn kleine Fenster für die Berechnung der Matching - Kosten verwendet werden. Weiterhin generiert der Algorithmus dichte Disparity – und Verdeckungskarten, wie es für die Erstellung von hochqualitativen zwischenliegenden Bildern vonnöten ist. Die Verbesserung von Disparity – und Verdeckungsschätzung in verdeckten Gebieten im ganz linken/ im ganz rechten Bild ist einer der wichtigsten Punkte.

II.) Beschreibung des Algorithmus Disparity – Schätzung / Verdeckungserkennung

A. Kamera Model

Angenommen wird ein multiokulares System mit N Standpunkten O_k , $k = 1, \dots, N$. Wir nehmen an, dass Zentralprojektion verwendet wird, und dass alle optischen Achsen parallel sind. Ein trinokulares System ($N=3$) ist in Fig.1 dargestellt. Sei I_k das Bild, das dem Standpunkt O_k entspricht und seien (x, y, z) die Koordinaten eines Punktes W im 3D – Koordinatensystem, das sein Zentrum in O_1 hat. Wenn $X_k(W)$ die x-Koordinate der Abbildung von W auf I_k ist, dann gilt:

$$X_1(W) = f \frac{x}{z} \quad \text{und} \quad X_k(W) = f \frac{x - b_{1k}}{z} \quad k = 2, \dots, N \quad (1)$$

wobei $b_{1k} = \overline{O_1 O_k}$ die Grundlinie des entsprechenden Standpunktes ist. f ist die Brennweite.

Mit Bezug auf das erste (ganz linke) Bild I_1 wird jeder Punkt in den anderen Bildern mit entsprechender Disparity gesehen:

$$d_k(W) = \begin{cases} X_k(W) - X_1(W), & \text{wenn } W \text{ sichtbar in } I_k \\ \text{undefiniert}, & \text{wenn } W \text{ verdeckt in } I_k \end{cases} \quad (2)$$

mit $k = 2, \dots, N$. Wenn W in I_k sichtbar ist, ergeben (1) und (2):

$$d_k(W) = -f \frac{b_{1k}}{z}$$

(3)

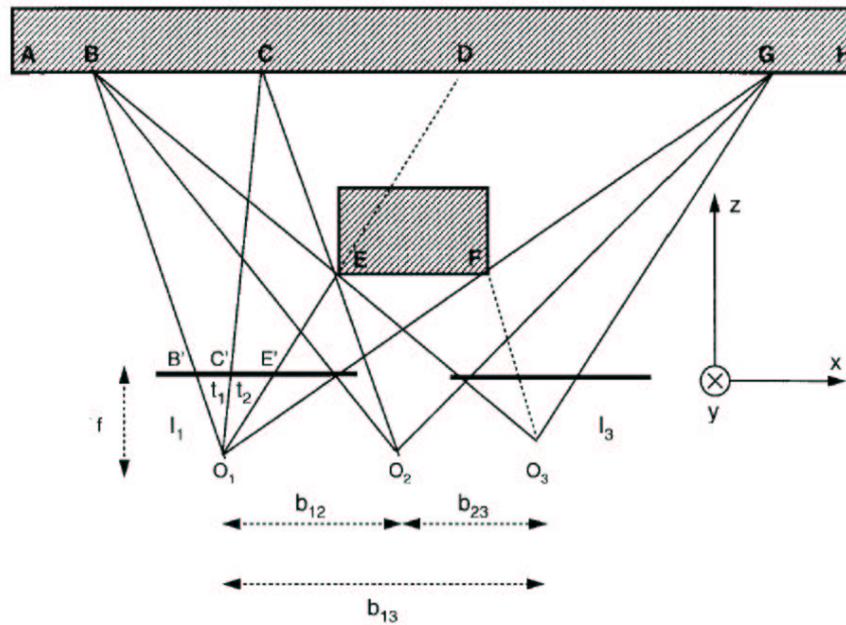


Fig. 1. System with three viewpoints.

B. Disparity Estimation und Occlusion Detection für 3D – Bildpaare

Wir werden jetzt den Spezialfall ausarbeiten, in dem nur zwei Standpunkte verfügbar sind (dargestellt in Fig.2).

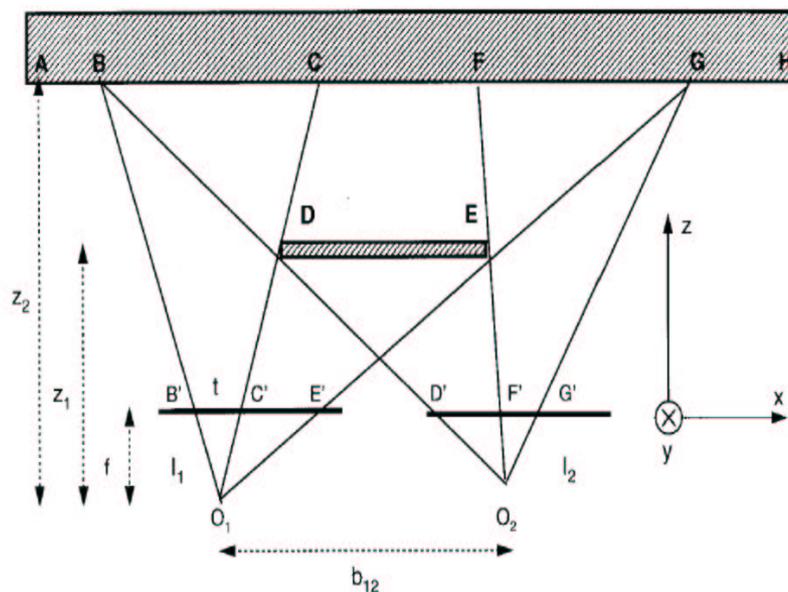


Fig. 2. Stereoscopic system (two viewpoints).

Aufgrund von perspektivischer Projektion gilt $X_1(D) = X_1(C) = X_1(C')$ und $X_2(D) = X_2(B) = X_2(D')$. Falls das verdeckende Objekt (DE) nicht zu schmal ist, liegt C links von F und das Segment $t = \overline{B'C'}$ ist im Bild I_2 verdeckt. Eigentlich ist leicht zu sehen, dass das Segment t im Standpunkt O_2 total verdeckt ist. Daraus folgt:

$$l = \overline{DE} > b_{12} \left(1 - \frac{z_1}{z_2}\right) \quad (4)$$

mit $z_1 = z(D) = Z(E)$ und $z_2 = Z(C) = Z(F)$.

Die Gültigkeit von (4) wird im Folgenden angenommen. Damit entspricht der Anteil t in I_1 , der im zweiten Standpunkt verdeckt ist

$$\begin{aligned} t = \overline{B'C'} &= X_1(C') - X_1(B') = X_1(D) - X_1(B) \\ &= X_1(D) - X_2(D) - X_1(B) + X_2(B) \\ &= -d_2(D) + d_2(B) = -\Delta d_2 \end{aligned} \quad (5)$$

weil $X_2(B) = X_2(D)$. Weiterhin gilt aus (5): $t = X_1(C') - X_1(B') > 0$ und damit $\Delta d_2 < 0$ und $d_2(D) < d_2(B)$. Daraus ergibt sich folgende Beobachtung, die im Folgenden sehr nützlich sein wird:

Beobachtung 1:

Die Anzahl von Pixeln in einem Liniensegment $\overline{A_1A_2}$, das in der ersten Kamera sichtbar ist, und in der zweiten verdeckt, gleicht der Disparity – Abnahme $d_2(A_1) - d_2(A_2)$.

Diese Beobachtung bleibt auch bei sich annähernden optischen Achsen gültig. Fig.3 zeigt diese Beobachtung mit einem Graphen der Disparity $d_2(x)$ der Pixel x in I_1 , wenn $|\Delta d_2| = t = 4$ ist. Pixel mit klar definierter Disparity sind schwarz, Pixel die in I_2 verdeckt sind, sind weiß.

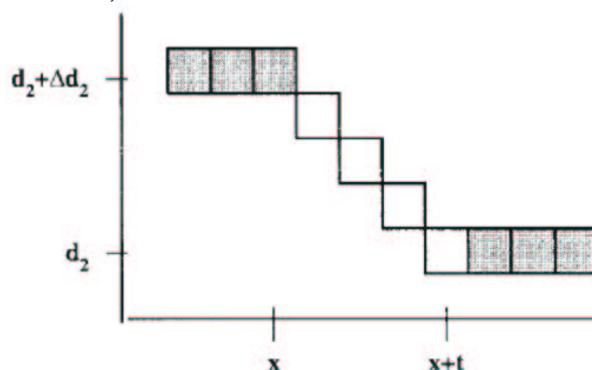


Fig. 3. Occlusions in the two-view case: the number of occluded pixels t is equal to the disparity change Δd_2 .

Für Pixel, die in I_1 verdeckt sind und in I_2 sichtbar sind, gilt $t' = \overline{F'G'} = d_2(G) - d_2(E) > 0$. Damit steigt die Disparity d_2 um die gleiche Anzahl Pixel, die in I_2 sichtbar und in I_1 verdeckt sind. Weiterhin ergibt sich aus (3), dass die Disparity d_2 für alle Punkte, die in I_1 und I_2 sichtbar sind, konstant bleibt. Dies führt zu folgender Beobachtung:

Beobachtung 2:

Sei t ein horizontales Liniensegment in I_1 , das aus Pixeln besteht, die auch in I_2 sichtbar sind. Dann ist die Disparity d_2 nicht absteigend, wenn man t von links nach rechts durchgeht. Diese Beobachtung drückt die wohl bekannte „monotonic ordering constraint“ aus.

Diese Beobachtungen liefern Beschränkungen für die Disparity – Veränderungen zwischen benachbarten Pixeln, die in Fig.4 schematisch zusammengefasst sind.

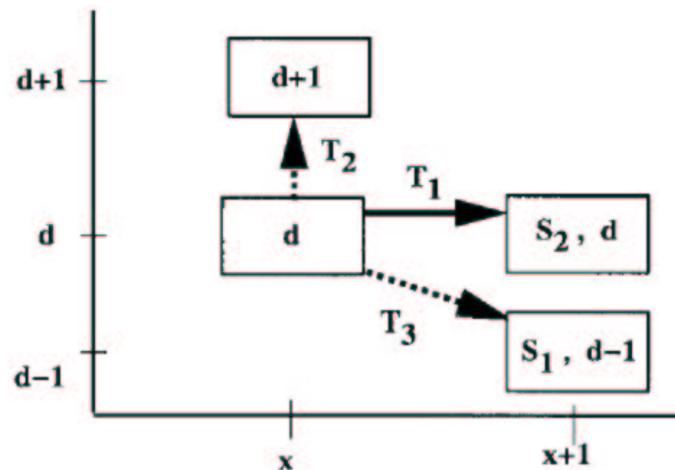
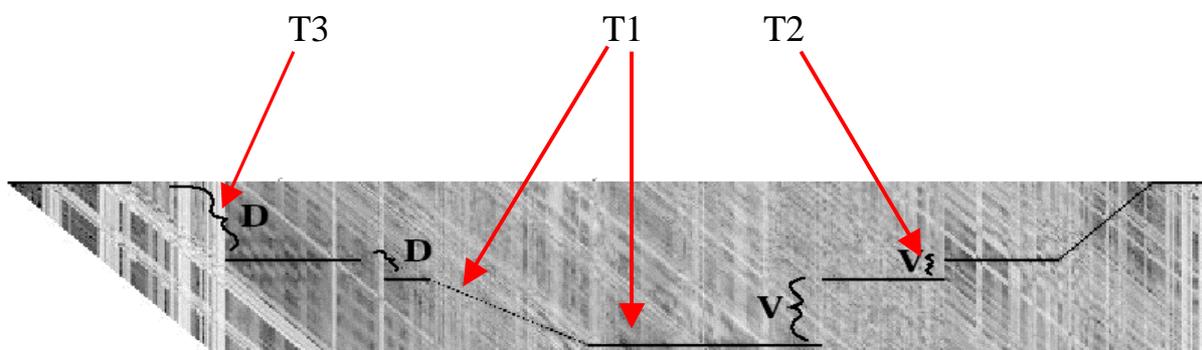


Fig. 4. Allowed transitions between states when using two views. The working pixel is (x, y) with disparity $d = d_2$.



Jedem Pixel in I_1 wird ein Zustand S_k zugewiesen, je nach dem, ob es in $k=1$ oder $k=2$ Standpunkten sichtbar ist. Beginnend mit dem Pixel x mit der Disparity d , gibt es die erlaubten Übergänge T_1 , der zu gleicher Disparity des folgenden Pixels führt, und T_2, T_3 , die entsprechend jeweils zu einer Disparity Ab- oder Zunahme führen.

Jedem erlaubten Übergang T_1, T_2, T_3 werden Kosten für jedes Pixel im ersten Bild zugeordnet, und dynamisches Programmieren wird verwendet, um die globalen Kosten für jede Bild – Scanlinie y zu minimieren. Eine zusätzliche Beschränkung während dieser Prozedur ist, dass einem T_2 Übergang kein T_3 Übergang folgt und umgekehrt.

Die Matching-Kosten, die mit einem T_1 Übergang verbunden sind, können wie folgt definiert werden:

$$c(x, y; d) = \sum_{(p,q) \in W} [I_2(x + d + p, y + q) - I_1(x + p, y + q)] \quad (7)$$

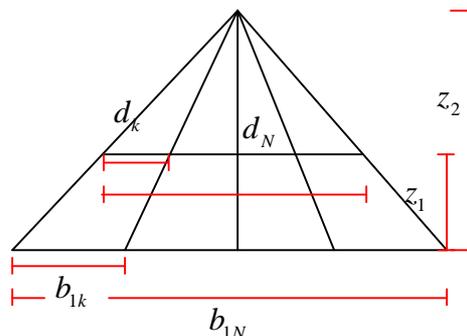
wobei W ein Fenster ist, das auf dem bearbeiteten Pixel (x,y) zentriert ist. Im Paper werden die zugehörigen Kosten für T_2, T_3 einander gleich gesetzt und bekommen den fixen Wert C_{change} .

C. Disparity Estimation und Occlusion Detection in Multiview Sequenzen

In multiokularen Systemen ist ein Pixel aus der ersten Sicht I_1 im Zustand k , wenn es in I_2, \dots, I_k Sichten sichtbar ist. Im Besonderen ist es im Zustand S_N , wenn es in allen Sichten sichtbar ist, im Zustand S_2 , wenn es nur in I_2 sichtbar ist, und im Zustand S_1 , wenn es, außer in I_1 , in allen Sichten verdeckt ist. Wenn d_k die Disparities der Sichten I_k mit $k = 2, \dots, N$ im Bezug auf I_1 sind, dann gilt für ein Pixel im Zustand S_N :

$$d_k = \frac{b_{1k}}{b_{1N}} d_N \quad (8)$$

Diese Formel ist leicht aus der folgenden Abbildung ableitbar. d_k sind die Disparities, b_{1k} sind die Grundlinien der Kamerapositionen.



Für Pixel im Zustand S_k werden modifizierte Matching - Kosten definiert, die aus dem gewichteten Durchschnitt der einzelnen Matching – Kosten bestehen.

$$C_k(x, y; d) = \frac{1}{k-1} \sum_{i=2}^k a_i c_i(x, y, d_i) \quad 2 \leq k \leq N \quad (9)$$

mit

$$c_i(x, y, d_i) = \sum_{(p,q) \in W} [I_i(x+p+d_i, y+q) - I_1(x+p, y+q)] \quad (10)$$

Die Disparity d_i , die durch (8) gegeben ist, wird auf den nächsten Integer gerundet. W ist ein Fenster, das auf dem bearbeiteten Pixel (x,y) zentriert ist. Die fixen Gewichte a_i können heuristisch gewählt werden. Die Fehler c_i , die den richtigen Übereinstimmungen zwischen den Sichten 1 und i entsprechen, nehmen mit größer werdendem i zu. Da die Übereinstimmungen zwischen Nachbarn zuverlässiger sind, sollten die Gewichte a_i mit zunehmendem i abnehmen. Die Wahl $a_i = 1-(i-2)/(N-1)$ lieferte befriedigende Ergebnisse.

Der Algorithmus aus dem vorherigen Abschnitt wird anschließend für die Schätzung der Disparity $d = d_N$ verwendet, wobei die modifizierten Kosten $C_N(x,y;d)$ eingesetzt werden. Ein erster „Dynamic Programming“ – Algorithmus für die Berechnung der Disparity d_N der Pixel in I_1 und für die Identifikation der Gebiete, die in mindestens einer Sicht verdeckt sind, basiert auf dem Algorithmus, der schematisch in Fig.5 dargestellt ist. Dieser Algorithmus ist eine direkte Verallgemeinerung des Zwei – Sichten – Algorithmus.

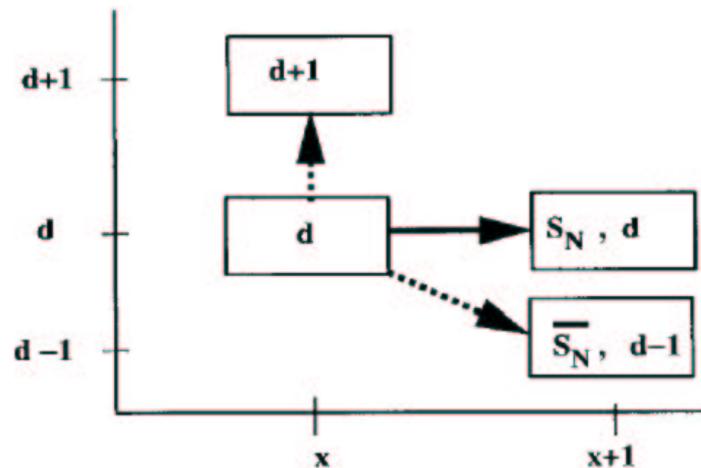


Fig. 5. Allowed transitions between states in the general (N -view) case. The disparity $d = d_N$ is estimated, and the occluded pixels in $\overline{S_N}$ (not in state S_N) are detected.

In diesem Fall sind nur zwei Zustände identifiziert. S_N , für Pixel die in allen Sichten sichtbar sind und $\overline{S_N}$ für Pixel, die in mindestens einer Sicht verdeckt sind. Auf diese Weise werden die verdeckten Gebiete entdeckt, aber nicht weiter analysiert. Eine genauere Analyse der verdeckten Gebiete liefert brauchbare Informationen, speziell für Anwendungen wie die Erzeugung von virtuellen

Bildern von zwischenliegenden Standpunkten. Es werden drei Algorithmen vorgestellt, die die verdeckten Regionen genauer untersuchen.

Algorithmus 1:

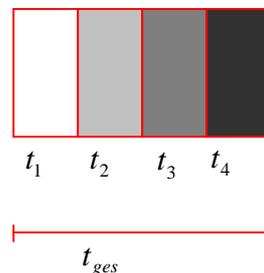
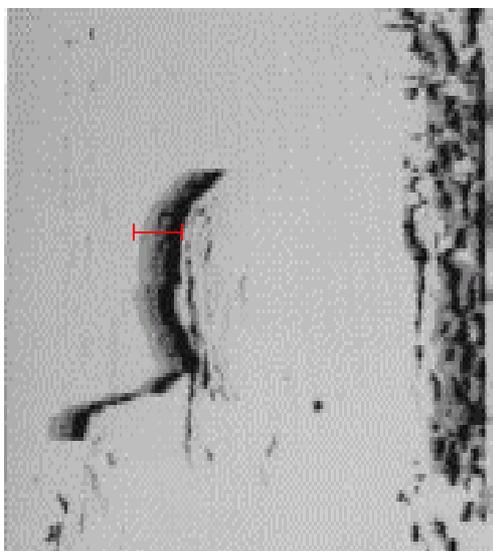
Geometrische Beobachtungen führen zu einem einfachen Benennungsvorgang für die erkannte verdeckte Region. Mit Bezug auf die Einfachheit des drei Sichten – Falls in Fig.1 gilt:

$$\frac{B'C'}{BC} = \frac{C'E'}{CD} \quad \text{und} \quad \frac{BC}{b_{23}} = \frac{CD}{b_{12}} \quad (11)$$

Falls $t_1 = B'C'$ und $t_2 = C'E'$, folgt:

$$\frac{t_1}{b_{23}} = \frac{t_2}{b_{12}} \quad (12)$$

Dies definiert einen einfachen ‚verdeckungsbenennenden‘ Vorgang, der das verdeckte Segment in N-1 Untersegmente teilt, deren Längen proportional zu den entsprechenden Grundlinien sind. Somit sind Pixel aus dem ersten Untersegment im Zustand S_{N-1} , Pixel aus dem zweiten Untersegment im Zustand S_{N-2} , und so weiter. Die entsprechenden Disparities sind durch (8) gegeben, wobei k der Pixelzustand und d_N die Disparity ist, die am Anfang des verdeckten Segments gemessen wurde. Im allgemeinen Fall, in dem die Grundlinien gleich sind, sind die Untersegmente offensichtlich auch gleich.



Algorithmus 2:

Algorithmus 1 basiert auf der impliziten Annahme, dass die Tiefe im gesamten verdeckten Segment konstant bleibt. Wenn das verdeckte Gebiet groß genug ist, werden innerhalb der verdeckten Regionen Tiefenunterschiede auftreten. In diesem Fall wird eine aufwendigere Disparity – Schätzung und Verdeckungserkennung für jedes verdeckte Segment benötigt. Um dies zu erreichen, wird der Algorithmus aus Sektion II-C rekursiv in $N-2$ Phasen in jeder verdeckten Region angewendet. In jeder Phase werden Disparity- und Zustandsinformationen für die Pixel dieser verdeckten Segmente bereitgestellt. Die genauen Phasen des Algorithmus sind wie folgt:

Phase 1: Identifikation von S_{N-1} und $\overline{S_{N-1}}$ aus den $\overline{S_N}$ - Verdeckungen unter Verwendung der Sichten 1 bis $N-1$

Phase 2: Identifikation von S_{N-2} und $\overline{S_{N-2}}$ aus den $\overline{S_{N-1}}$ - Verdeckungen unter Verwendung der Sichten 1 bis $N-2$

⋮

Phase $N-2$: Identifikation von S_2 und S_1 aus den $\overline{S_3}$ - Verdeckungen unter Verwendung der Sichten 1 und 2

In der $(N-k)$ ten Phase wird die Disparity d_k mit den erlaubten Übergängen, die in Fig. 6 dargestellt sind, geschätzt. Die Disparity d_k ist an den zwei extremen Punkten jedes verdeckten Segmentes, das aus der vorherigen Phase entsteht, wohl definiert. Somit wird in jeder Phase der optimale Weg zwischen diesen extremen Punkten, deren Disparities in der vorherigen Phase bestimmt wurden, gefunden.

Wie man sieht, verwendet der Algorithmus viele Disparities d_k , $k = 2, \dots, N$. Somit ergibt seine Implementierung keine direkten Informationen, die die Tiefenveränderungen in den verdeckten Regionen betreffen, und er ist nicht förderlich für sparsame Kodierungsverfahren.

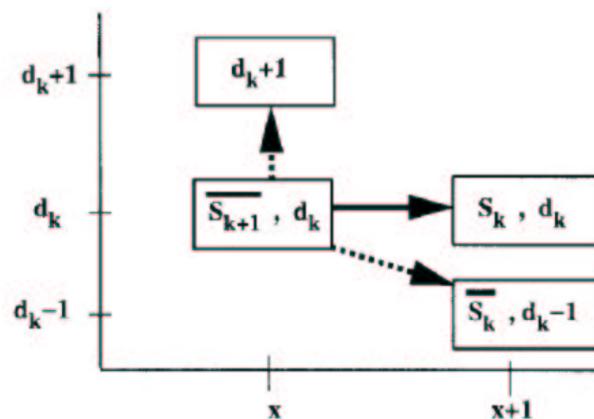


Fig. 6. Allowed transitions between states in the general (N -view) case. The working pixel is occluded in the $k+1$ view. The disparity d_k is estimated, and the occluded pixels in $\overline{S_k}$ (not in state S_k) in the k th view are detected.

Algorithmus 3:

Für eine alternative Implementierung, die in Fig.7 dargestellt ist, wird die vertikale Achse unter Verwendung von (8) in $d = d_N = (b_{1N}/b_{1k})d_k$ umbenannt. Nach jeder der $N-2$ Schätzungsphasen wird die Disparity d und der entsprechende Pixelzustand S_k für jedes Pixel geschätzt. Mit dieser Formulierung wird nur ein Disparity – Feld $d = d_N$ für alle Phasen gebraucht.

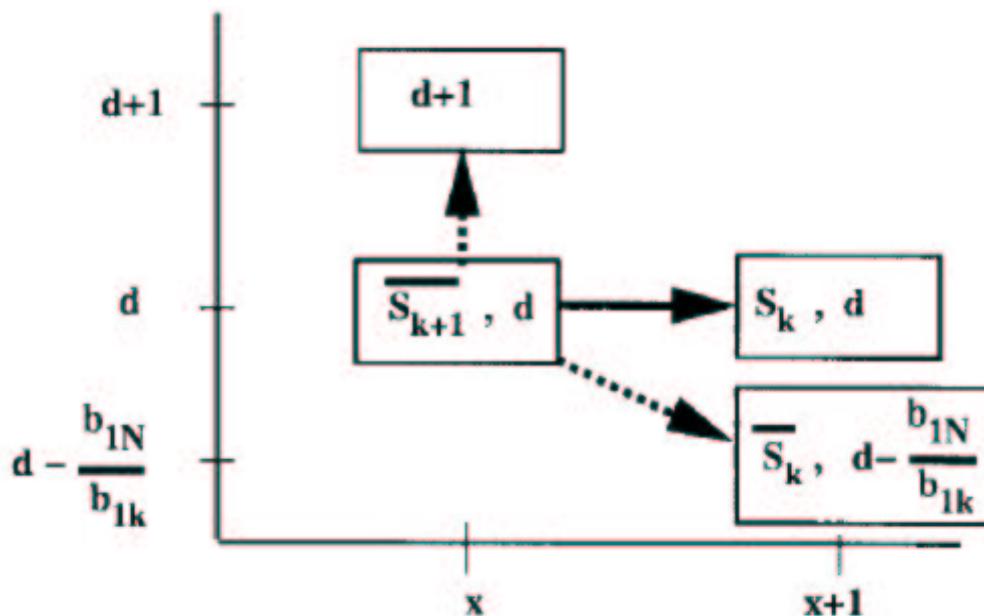


Fig. 7. Allowed transitions between states in the general (N -view) case. The working pixel is occluded in the $k+1$ view. The disparity $d = d_N$ is estimated, and the occluded pixels \overline{S}_k in the k th view are detected.

D. Simultane Erkennung und Benennung von verdeckten Regionen

In den bisherigen Methoden wurden die Erkennung und Benennung der verdeckten Regionen sequentiell implementiert. Ein anderer Ansatz ist es, dieselben Matching – Kosten, die durch (9) gegeben sind, in einem Algorithmus für die gleichzeitige Verdeckungserkennung und – benennung zu verwenden.

Wenn das aktuelle Pixel x in I_1 die Disparity d und den Zustand S_k hat, dann gibt es für den direkten Nachbarn $x+1$ folgende Optionen für Disparity und Zustand:

- Wenn $k \geq 2$ ist, steigt die Disparity entweder an, oder bleibt gleich. Der Zustand von $x+1$ im ersten Fall ist S_k und entweder S_k oder S_{k-1} in letzteren.

- Wenn $k=1$ ist, sinkt die Disparity entweder, oder bleibt gleich. Im ersten Fall hat $x+1$ den Zustand S_1 (ist in allen Sichten verdeckt) und (5) und (8) ergeben $t_2 = -\Delta d_2 = -(b_{12}/b_{1N})\Delta d$. Somit ergibt sich $\Delta d = -(b_{1N}/b_{12})$, da $t_2 = 1$.

Wenn die Disparity konstant bleibt, signalisiert ein Übergang in den Zustand S_N das Ende des verdeckten Segments. Wir nehmen oben implizit an, dass es links von jedem, in allen Sichten verdeckten Segment (Zustand S_1) ein Segment gibt, das in allen Sichten sichtbar ist (Zustand S_N).

Die obigen Beschränkungen, die aus einfachen geometrischen Überlegungen abgeleitet wurden, sind schematisch in Fig.8 (a) und (b) dargestellt. Durchgezogene Linien deuten eine Übereinstimmung an. In diesem Fall sind die zugewiesenen Kosten durch (9) gegeben. Gepunktete Linien deuten Disparity – Veränderungen an. In diesem Fall werden die Kosten gleich den fixen Kosten C_{change} gesetzt.

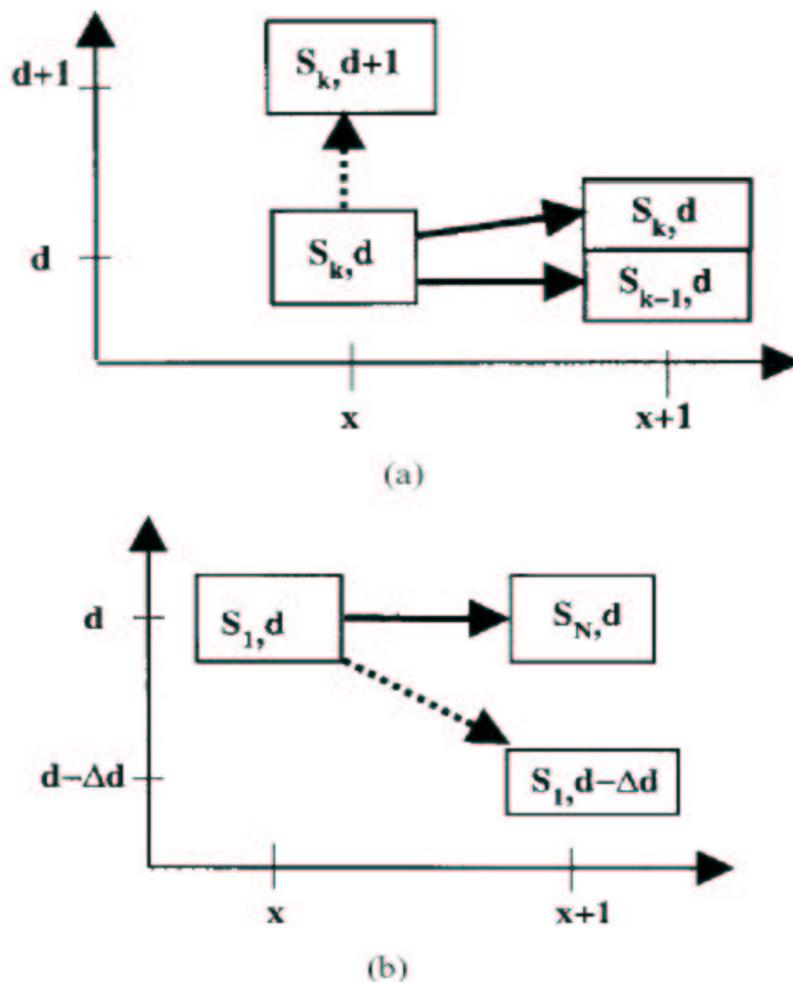


Fig. 8. Allowed transitions between states. The working pixel is (x, y) with disparity d and state S_k . (a) $k = 2, \dots, N$ (b) $k = 1, \Delta d = (b_{1N}/b_{12})$.

E. R-L Disparity – Feld und Verdeckungsschätzung

Die erhaltenen Disparity/Zustandsinformationen gehören zu jedem Pixel in der ersten Sicht (ganz linkes Bild) und können somit „L-R disparity“ Feld genannt werden, um es vom entgegengesetzten „R-L disparity“ Feld zu unterscheiden, das jedem Pixel in der N-ten Sicht (ganz rechtes Bild) entspricht. Wenn der Zustand eines Pixels p in der ersten Sicht (links) N ist, dann ist ein entsprechendes Pixel q in der N-ten Sicht definiert. Somit ergibt sich sofort ein entsprechender Vektor im R-L Disparity Feld für das Pixel q .



F. Beispiel



Ausgangsbild

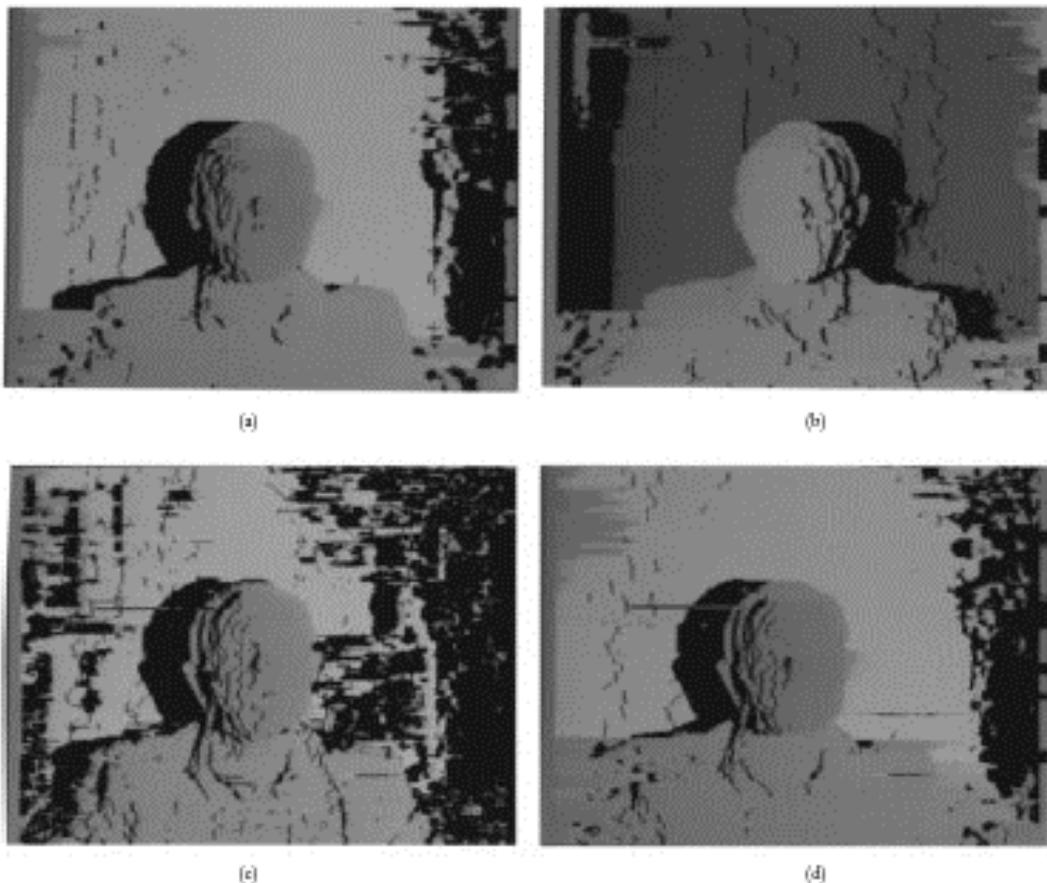
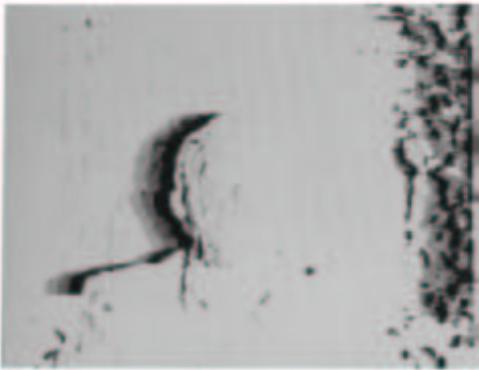


Fig. 10. Comparison of disparity estimation using four and two views. Occluded areas are shown in black. (a) $L \rightarrow R$ disparity field obtained after the initial disparity estimation procedure when using the proposed algorithm. (b) Corresponding $R \rightarrow L$ disparity field. (c) $L \rightarrow R$ disparity field when using only two views ($\|C_{\text{image}}\| = 2$). (d) $R \rightarrow L$ disparity field when using only two views ($\|C_{\text{image}}\| = 2$).

Bild oben links: L-R Disparity Field; 4 views
 Bild oben rechts: R-L Disparity Field; 4 views
 Bild unten links: L-R Disparity Field; 2 views
 Bild unten rechts: R-L Disparity Field; 2 views



(a)



(b)

Fig. 11. (a), (b) $L-R$ disparity field and corresponding state map using Algorithm 3 for the labeling of the occluded areas. Gray scale is used to indicate the states of the pixels. Thus, white corresponds to state S_0 (pixels visible all four views), light gray to state S_1 (pixels visible in three views), dark gray to state S_2 (pixels visible in two views), and black to state S_3 (occluded pixels).

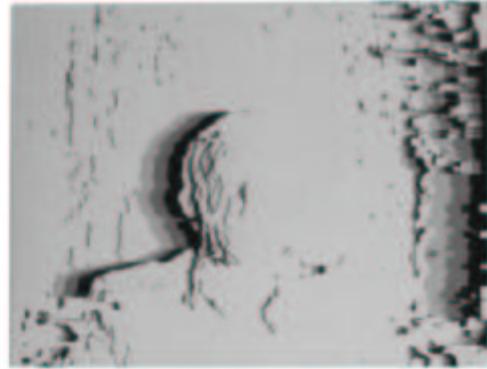


Fig. 12. State map for the $L-R$ disparity field obtained using Algorithm 1 for the labeling of the occluded areas.

Bild oben links: L-R Disparity Field
 Bild oben rechts: Zustandskarte nach Algorithmus 1
 Bild unten links: Zustandskarte nach Algorithmus 2