

**UNIVERSITÄT
MANNHEIM**

**MULTIMEDIA-SEMINAR
BERECHNUNG DER PARAMETER DES KAMERAMODELLS
Seminar**

eingereicht im: April 2008

von: Sina Deibert
geboren am 25. Oktober 1978
in Mannheim

Matrikelnummer: 0828839

Betreuer: Dr. Stephan Kopf
Lehrstuhl für Praktische Informatik IV

Universität Mannheim
Lehrstuhl für ABWL und Wirtschaftsinformatik
D – 68131 Mannheim
Telefon: +49 621 1811691, Fax +49 621 1811692
Internet: <http://www.bwl.uni-mannheim.de/wifo1>

Inhaltsverzeichnis

Abbildungsverzeichnis	iii
1 Einleitung	1
2 Verfahren zur Berechnung von Parametern des Kameramodells	2
2.1 Überblick	3
2.2 Detaillierte Beschreibung von Verfahren	4
2.2.1 Iterative Bild-Registrierungs-Technik zur Ermittlung der Translationsparameter	4
2.2.2 Berechnung der Kamerabewegung für affine und projektive Kamerabewegung	8
2.2.3 Erzeugung von Panoramabildern	13
2.3 Fazit	14
Literaturverzeichnis	16

Abbildungsverzeichnis

2.1	Verschiebungsvektor zwischen $F(x)$ und $G(x)$ (Quelle: [4])	2
2.2	Verschiebungsvektor in x (Quelle: angelehnt an [4])	5
2.3	Fehler bei der Berechnung von h (Quelle: eigene Darstellung)	6
2.4	Affine Transformation eines Bildes (Quelle: eigene Darstellung)	8
2.5	Affine Transformation von $F(x)$ nach $G(x)$ (Quelle: eigene Darstellung) .	9
2.6	Projektion von $F(x)$ nach $G(x)$ (Quelle: eigene Darstellung)	12
2.7	Erstellung eines Panoramabildes (Quelle: eigene Darstellung)	14

1 Einleitung

Für das Ausrichten und Erzeugen von Panoramabildern ist es notwendig, die Bewegung der Kamera zwischen den einzelnen Bildern zu ermitteln ([3]). Diese kann mit Hilfe eines Kameramodells beschrieben werden. Die Ermittlung der Bewegung wird auch als Bildregistrierung bezeichnet. Sie kommt in unterschiedlichen Anwendungen zum Einsatz. Stereo vision, also die Aufnahme eines Bildes aus unterschiedlichen Kamerapositionen und die damit mögliche Ableitung der Objektgröße und Erzeugung eines 3D-Objekts sind ein mögliches Anwendungsgebiet. Weitere Anwendungsfelder sind die Bewegungsanalyse (sowohl der Kamera als auch der Objekte) sowie die Mustererkennung.

Die Bestimmung der Kameraparameter ist aufgrund von Veränderungen der Blickrichtung oder des Zooms zwischen den einzelnen Bildern notwendig. Bei der Analyse der Bewegung aller Pixel eines Bildes spricht man vom optischen Fluss. Dieser kann unterschiedliche Eigenschaften haben. Zum einen kann sich die Kamera selbst bewegen, zum anderen können sich Objekte innerhalb der Szenen bewegen. Auch eine Kombination von beiden Arten der Bildveränderung ist möglich. Dabei werden Objekte anhand der Bewegung von Bildpunkten segmentiert. Solche Bildpunkte, die sich gleichförmig bewegen, werden zu einem bestimmten Objekt gehörig interpretiert.

Die Berechnung der Kameramodell-Parameter kann als Festlegung der Kameraposition verstanden werden. Das Kameramodell kann beispielsweise über sechs Parameter definiert werden (drei für die Position, drei für die Ausrichtung). Zur Berechnung der Modellparameter werden korrespondierende Bildpunkte in beiden Bildern gesucht. Die Bestimmung der Parameter des Kameramodells kann nun mittels unterschiedlicher Verfahren durchgeführt werden. Dabei ist zu beachten, dass die verschiedenen Verfahren nicht nur hinsichtlich der Qualität der Ergebnisse sondern auch im Bezug auf die benötigte Berechnungszeit zu differenzieren sind. Im folgenden Kapitel werden unterschiedliche Verfahren der Parameterberechnung vorgestellt und kritisch auf deren Qualität und Beschränkungen eingegangen.

2 Verfahren zur Berechnung von Parametern des Kameramodells

Wie im vorherigen Kapitel bereits erwähnt wurde, unterscheiden sich Verfahren zur Berechnung des Kameramodells sowohl in der Qualität der Ergebnisse als auch im Berechnungsaufwand. Ein kurzer Überblick soll dies verdeutlichen bevor auf ausgewählte Verfahren im Detail eingegangen wird. Wie man mit Hilfe der Parameter der Kameramodells dann ein Panoramabild erstellt, wird in Abschnitt 2.2.3 erläutert. Die einfachste Bewegung zwischen zwei Bildern ist die Verschiebung (siehe Abbildung 2.1). Wie man hierfür die Parameter also den Verschiebungsvektor h berechnet, wird in Abschnitt 2.2.1 beschrieben. Zwischen zwei gegebenen Funktionen $F(x)$ und $G(x)$ wird für jedes Pixel x einer bestimmten Region R der Verschiebungsvektor h bestimmt.

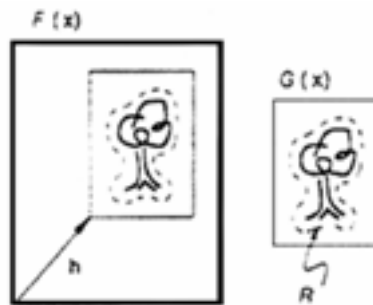


Abbildung 2.1: Verschiebungsvektor zwischen $F(x)$ und $G(x)$ (Quelle: [4])

In Abschnitt 2.2.2 wird von einer komplexeren Kamerabewegung (Zoom, Rotation, ..) ausgegangen und die Berechnung der entsprechenden Kameraparameter für die affine sowie die projektive Bewegung erläutert.

2.1 Überblick

Um die Bild-Registrierung also die Kameraparameter bzw. den Verschiebungsvektor zwischen zwei Bildern zu bestimmen, ist es nötig, Paare von korrespondierenden Bildpunkten zu kennen. Ausgehend von einer Menge dieser Paare können dann unterschiedliche Algorithmen angewendet werden.

Ein rechen- und somit zeitintensives Verfahren zur Bestimmung der Bild-Registrierung zwischen zwei Bildern, ist die erschöpfende Suche. Hierbei werden alle möglichen Translationen zwischen beiden Bildern getestet und der resultierende Fehler mittels einer Fehlerfunktion gemessen. Für kleine Bilder mit wenigen Translationen ist dieses Verfahren evtl. noch anwendbar, für größere Bilder muss ein besseres/schnelleres Verfahren herangezogen werden. Eine Möglichkeit ist zum Beispiel das Hill-Climbing-Verfahren. Hierbei startet man mit einem initialen Wert und sucht in einer begrenzten Nachbarschaft nach einem besseren Wert. Dieser dient dann als Ausgangswert für die nächste Iteration. Dieses Verfahren hat den Nachteil, dass zum einen Fehler auftreten können und zum anderen die Gefahr besteht, nur ein lokales Optimum zu finden. Ein anderes Verfahren, das nicht den Unterschied zwischen zwei Bildpunkten/Bildern sondern den Fehler bei der Berechnung des Unterschiedsvektors berücksichtigt, nutzt den Sequential Similarity Detection Algorithm (SSDA). Hierbei wird eine Grenze festgelegt, bei deren Unterschreitung zum nächsten Unterschiedsvektor übergegangen wird. Die Reihenfolge, in denen die Unterschiedsvektoren dabei gewählt werden, ist über das Verfahren allerdings nicht festgelegt. Eine Erweiterung von SSDA ist das veränderliche Festlegen der Grenze. Hierbei entspricht der Wert immer dem bis dahin kleinsten gefundenen Fehler.

Die oben erwähnten Methoden können zum Teil auch kombiniert werden, um zum Beispiel beim SSDA die Reihenfolge der Berechnung des Unterschiedsvektors zu bestimmen. Eine weitere Möglichkeit, die unterschiedlichen Verfahren anzuwenden und die Berechnung noch weiter zu beschleunigen ist die so genannte Grob-Fein-Suche. Dabei wird die Bild-Registrierung für eine niedrigere Auflösung der Bilder berechnet und anschließend die gewonnenen Ergebnisse auf eine höhere Bildauflösung angewendet.

2.2 Detaillierte Beschreibung von Verfahren

Im folgenden Abschnitt werden zwei ausgewählte Verfahren, die oben beschrieben Algorithmen zum Teil einsetzen, genauer beschrieben.

2.2.1 Iterative Bild-Registrierungs-Technik zur Ermittlung der Translationsparameter

Das Verfahren von Lucas und Kanade ([4]) wurde 1981 erstmals vorgestellt. Lucas et al. gehen dabei wie folgt vor. Zuerst werden Übereinstimmungen zwischen zwei Bildern gesucht, die dann in einem weiteren Schritt dazu genutzt werden, die Kameraparameter zu bestimmen. Dabei bestimmt das Verfahren die Reihenfolge in der die Unterschiedsvektoren ausgewählt werden. Es werden anschließend die Gradienten bestimmt, um die nächste Translation zu schätzen. Hierbei wird iterativ nach dem Newton-Raphson-Verfahren vorgegangen. Die Translation wird durch den Verschiebungsvektor h ausgedrückt (siehe Abbildung 2.1).

Die Erläuterung des Verfahrens nach Lucas et al. startet mit dem eindimensionalen Fall und wird dann auf den mehrdimensionalen Fall übertragen.

Eindimensionaler Fall

Geht man im eindimensionalen Fall von zwei Kurven $F(x)$ und $G(x) = F(x + h)$ aus, so möchte man den Unterschiedsvektor h zwischen diesen beiden Kurven im Punkt x berechnen (Abbildung 2.2). h repräsentiert dabei die horizontale Verschiebung des Punktes. Hierbei untersucht man in der Nachbarschaft von x das Verhalten von $F(x)$ anhand der linearen Approximation (I). Dabei definiert man das Verhalten der Funktion über die Steigung der Funktion:

$$\text{I: } F'(x) \approx \frac{F(x+h)-F(x)}{h} = \frac{G(x)-F(x)}{h} \quad (h \neq 0).$$

Daraus folgt für h

$$\text{II: } h = \frac{G(x)-F(x)}{F'(x)} \quad (F'(x) \neq 0)$$

Voraussetzung für die erfolgreiche Schätzung ist dabei, dass h nicht zu groß ist. Die Ableitung der Funktion an der Stelle x ist dabei ein lokales Merkmal und die Schätzung

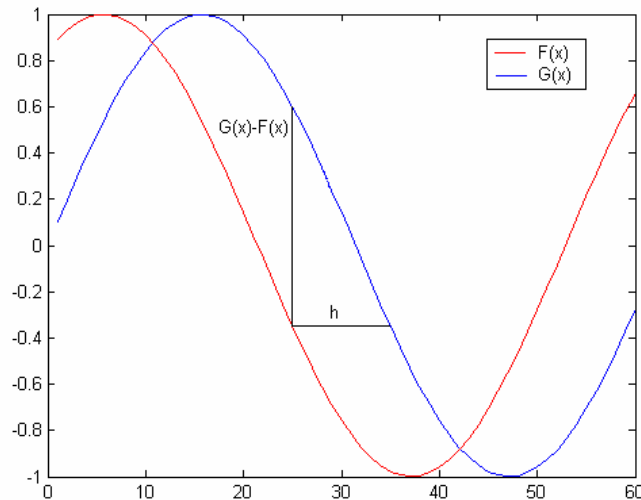


Abbildung 2.2: Verschiebungsvektor in x (Quelle: angelehnt an [4])

von h hängt davon ab. Zur Bestimmung von h kann eine Schätzung an mehreren Stellen vorgenommen werden und der daraus resultierende Mittelwert bestimmt werden:

$$\text{III: } h = \sum_x \frac{G(x) - F(x)}{F'(x)} / \sum_x (1).$$

Zu beachten ist dabei, dass die Näherung in (I) nur für diese Stellen der Kurve gute Werte liefert, an denen $F(x)$ nahezu linear ist. Ist $|F''(x)|$ allerdings groß, so führt dies zu einem schlechten Ergebnis. Diese Eigenschaft kann man zur Gewichtung der einzelnen Schätzergebnisse heranziehen, in dem man den Kehrwert der Schätzung von h ersetzt und somit diese Eigenschaft in die Gleichung integriert:

$$\begin{aligned} \text{IV: } F''(x) &= (G'(x) - F'(x))/h \\ \Rightarrow 1/h &= F''(x)/(G'(x) - F'(x)). \end{aligned}$$

Das Gewicht $w(x)$ wird dann ausgedrückt durch:

$$\text{V: } w(x) = 1/|G'(x) - F'(x)|.$$

Dies führt zum Ausdruck für h

$$\text{VI: } h = \sum((w(x)[G(x) - F(x)]/F'(x)) / \sum(w(x))).$$

Ein Beispiel zeigt, wie sinnvoll diese Veränderung der Schätzfunktion für *hist.* Betrachtet man beispielsweise die Kurven in 2.2 an deren Schnittpunkt, so ist dort zum

einen der Wert $h = 0$, da $F(x) = G(x)$. Diese Tatsache ist schlecht. Durch die Einbeziehung des Gewichts für h wird der Einfluss von diesem h am Schnittpunkt in den Durchschnittswert allerdings stark reduziert, da das Gewicht an dieser Stelle sehr klein ist, da sich $F'(x)$ und $G'(x)$ stark unterscheiden.

Nach der Ermittlung von (VI) kann man nun $F(x)$ mit Hilfe der Schätzung von h solange verschieben, bis h einen optimalen Wert hat. Diese Vorgehen entspricht dem Vorgehen beim Newton-Raphson-Algorithmus. Man berechnet aus h_k iterativ h_{k+1} :

$$h_0 = 0$$

$$\text{VII: } h_{k+1} = h_k + \frac{\sum (w(x)[G(x) - F(x + h_k)]/F'(x + h_k))}{\sum (w(x))}.$$

Mehrdimensionaler Fall

Der beschriebene Ansatz zur Berechnung von h mittels der linearen Schätzung eignet sich allerdings nur für den eindimensionalen Fall. Möchte man eine Vorgehensweise entwickeln, die auch für den zweidimensionalen Fall anwendbar ist, so kann dies über die Minimierung des quadratischen Fehlers der L2-Norm ($L2 - Norm = (\sum ([F(x + h) - G(x)]^2))^{0.5}$) geschehen. Mit dieser Vorgehensweise wird auch der Fall abgedeckt, dass $F'(x) = 0$, also die Kurve am Punkt x parallel zur x-Achse ist, abgefangen. Man kann nun $F(x+h) \approx F(x) + hF'(x)$ aus (I) nutzen, um ein h zu finden, das die L2-Norm für den Fehler zwischen den Kurven minimiert. Daraus ergibt sich die Summe der quadratischen Abweichungen über alle Pixel des Bildes (siehe Abbildung 2.3

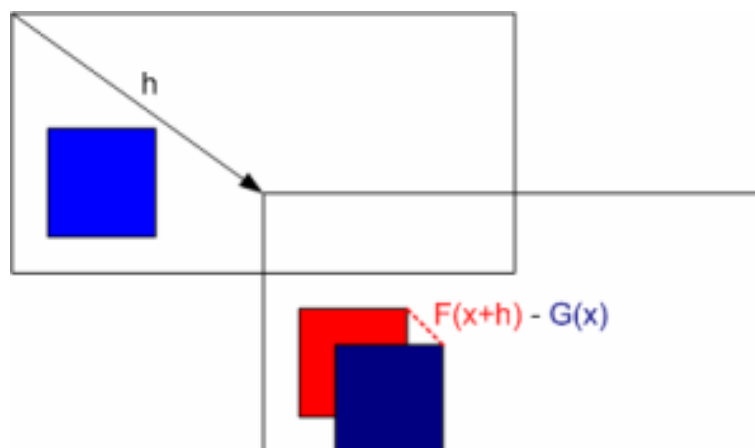


Abbildung 2.3: Fehler bei der Berechnung von h (Quelle: eigene Darstellung)

(bzw. endlich viele Werte im Definitionsbereich von F und G):

$$\text{VIII: } E = \sum [F(x+h) - G(x)]^2 \approx \sum [F(x) + hF'(x) - G(x)]^2$$

Zur Minimierung dieser Funktion (VIII) ist es nun notwendig, die Ableitung nach h gleich Null zu setzen. Daraus ergibt sich

$$\begin{aligned} 0 &= \frac{\partial E}{\partial h} \\ &\approx d/dh \sum ([F(x) + hF'(x) - G(x)]^2) \\ &= d/dh \sum (F(x)^2 - 2hF'(x)F(x) - 2F(x)G(x) - h^2F'(x)^2 - 2hF'(x)G(x) + G(x)^2) \\ &= \sum (2F'(x)[F(x) + hF'(x) - G(x)]) \\ &\Rightarrow IX : h \approx (\sum (F'(x)[G(x) - F(x)])) / \sum (F'(x)^2). \end{aligned}$$

Wie man sieht, entspricht dies der Funktion in (VI) mit $w(x) = F'(x)^2$. Hierdurch wird nun eine Division durch Null vermieden und das Verfahren ist auch auf den zweidimensionalen Raum anwendbar.

Kombiniert man (IX) nun mit (VII) so erhält man für die iterative Schätzung von h :

$$h_o = 0$$

$$\text{X: } h_{k+1} = h_k + \sum (w(x)F'(x+h_k)[G(x) - F(x+h_k)]) / \sum (w(x)F'(x+h_k)^2)$$

Der oben beschriebene Algorithmus kann so angepasst werden, dass er auch auf zwei- oder mehrdimensionale Fälle anwendbar ist. Hierzu werden x und h nicht mehr als Punkte, sondern als mehrdimensionale Vektoren betrachtet. Nun kann man die lineare Approximation auf $F(x+h) \approx F(x) + hF'(x)$ anwenden und erhält mit:

$$\frac{\partial}{\partial x} = [\frac{\partial}{\partial x_1} \frac{\partial}{\partial x_2} \dots \frac{\partial}{\partial x_n}]^T$$

$$\text{XI: } F(x+h) \approx F(x) + h \frac{\partial}{\partial x} F(x).$$

Mittels dieser Approximation wird wiederum E minimiert.

$$\begin{aligned} 0 &= \frac{\partial}{\partial h} E \\ &= \frac{\partial}{\partial h} \sum_x [F(x) + h \frac{\partial F}{\partial x} - G(x)]^2 \\ &= \sum_x 2 \frac{\partial F}{\partial x} [F(x) + h \frac{\partial F}{\partial x} - G(x)]. \end{aligned}$$

Dies führt wiederum zu

$$h = [\sum_x (\frac{\partial F}{\partial x})^T [G(x) - F(x)]] [\sum_x (\frac{\partial F}{\partial x})^T (\frac{\partial F}{\partial x})]^{-1}.$$

Diese Funktion ähnelt der Funktion für den Eindimensionalen Fall (IX) sehr. Das Verfahren kann noch weiter generalisiert werden, in dem man nicht nur eine einfache Bewegung

eines Objekts bzw. der Kamera, sondern auch die Rotation, Zoom berücksichtigt. Nicht alle affinen Bewegungen sind jedoch als Kamerabewegungen möglich wie z.B. Scherungen (siehe Abbildung 2.4).

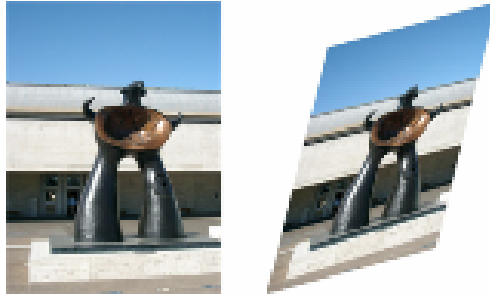


Abbildung 2.4: Affine Transformation eines Bildes (Quelle: eigene Darstellung)

Auch Veränderungen im Kontrast und Helligkeit müssen z.B. für die Berechnung bei Stereo Visions, in denen sich die beiden Bilder eben in diesen Punkten unterscheiden können, mit einbezogen werden. Anstatt den Verschiebungsvektor h zu berechnen, ermittelt man nun die Kameraparameter ([2]). Die affine Kamerabewegung wird durch die Matrix A ausgedrückt, die $F(x)$ in $G(x)$ transformiert. Somit gilt für $G(x)$:

$$G(x) = F(xA + h).$$

Die Veränderung des Kontrast bzw. der Helligkeit wird über die Parameter α und β ausgedrückt. Integriert man nun diese Parameter in die Gleichung für die einfache Verschiebung, so führt dies zu folgender Gleichung:

$$\text{XI: } E = \sum_x [F(xA + h) - (\alpha G(x) + \beta)]^2.$$

Im folgenden Abschnitt wird genauer auf die Bestimmung der Matrix A eingegangen.

2.2.2 Berechnung der Kamerabewegung für affine und projektive Kamerabewegung

Die Betrachtung dieses Verfahren, das von Dirk Farin 2005 [1] entwickelt wurde, geschieht in mehreren Schritten. Zu Beginn wird eine einfache affine Bewegung zunächst für den ein- und dann den zweidimensionalen Fall betrachtet, bevor im Anschluss eine Erweiterung auf die projektive Bewegung stattfindet. Das Verfahren eignet sich dabei allerdings nur zur Berechnung der Bewegungen des Hintergrunds von Bildern. Bewegungen von Objekten im Bild können somit nicht berücksichtigt werden. Hierzu ist eine

Erweiterung des Modells notwendig, auf die im weiteren allerdings nicht eingegangen wird.

Eindimensionale affine Bewegung

Zwei typische Bewegungen, die durch eine Kamera ausgeführt werden können, sind zum einen das Zoomen sowie das Rotieren der Kamera. Beide lassen sich einfach durch eine lineare Gleichung integrieren. Sei ein Bildpunkt im eindimensionalen Raum durch x und der korrespondierende Bildpunkt nach der Bewegung durch x' ausgedrückt. Die Parameter zur Beschreibung der Bewegung sind a (Rotation) sowie b (Zoom). Dann lässt sich x' durch $x' = ax + b$ ausdrücken oder allgemein für alle Punkte der linearen Funktion: $x'_i = ax_i + b$. Dabei ist das Zentrum des Zooms der Schnittpunkt der beiden Geraden, an dem gilt $x_i = x'_i$. Abbildung 2.5 verdeutlicht die eindimensionale affine Transformation von $F(x)$ nach $G(x)$.

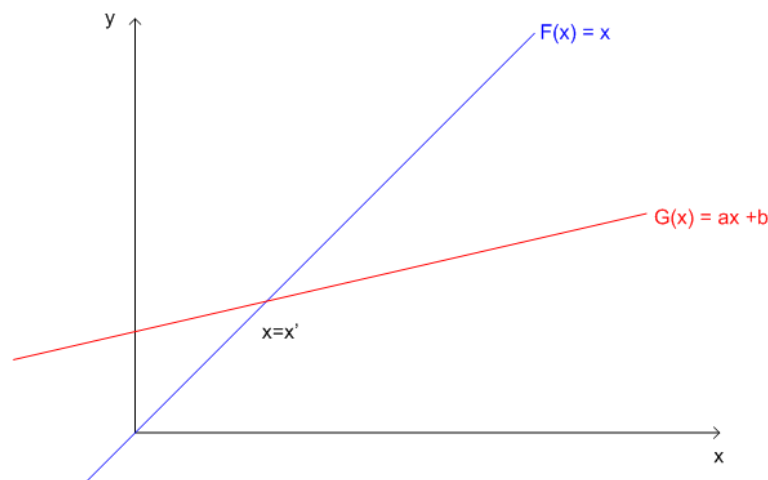


Abbildung 2.5: Affine Transformation von $F(x)$ nach $G(x)$ (Quelle: eigene Darstellung)

Ziel ist es nun, die Parameter a und b zu bestimmen. Dazu bedient man sich einer Menge von korrespondierenden Bildpunkten. Da man davon ausgehen muss, dass es sich nicht bei allen Bildpunkt-Paaren um tatsächlich Paare handelt, sondern auch einige Fehler bei der Berechnung dieser Korrespondenzen entstehen, muss man diese Fehler bei der Berechnung ebenfalls berücksichtigen. Dies geschieht durch die Berechnung des quadratischen Fehlers zwischen der gemessenen Position und der Position, die man aus dem

Modell erhält $\{x_i\}$ Menge der Bildpunkte im ersten Bild, $\{\hat{x}_i\}$ Menge der korrespondierenden Bildpunkte). Dies führt zu folgender Gleichung für den Fehler:

$$\text{I: } E = \sum_i ((ax_i + b) - \hat{x}_i)^2.$$

Um nun den Fehler wie im zuvor beschriebenen Verfahren zu minimieren, müssen die Ableitungen der Fehlerfunktion (I) gleich Null gesetzt werden. Daraus folgt:

$$\text{II: } \frac{\partial E}{\partial a} = \sum_i 2(ax_i + b - \hat{x}_i)x_i \quad ; \quad \frac{\partial E}{\partial b} = \sum_i 2(ax_i + b - \hat{x}_i),$$

$$\text{III: } \sum_i (ax_i^2 + bx_i - \hat{x}_i x_i) = 0 \quad ; \quad \sum_i (ax_i + b - \hat{x}_i) = 0.$$

Aus den beiden resultierenden Gleichungen kann ein lineares Gleichungssystem aufgestellt werden, um die Parameter a und b zu bestimmen.

Zweidimensionale affine Bewegung

Im zweidimensionalen Fall sind die korrespondierenden Bildpunkte nun durch zwei Koordinaten ausgedrückt. Die Menge der originalen sowie korrespondierenden Bildpunkte sind dann $\{(x_i, y_i)\}$ bzw. $\{(\hat{x}_i, \hat{y}_i)\}$. Die tatsächliche Position des Bildpunkts im zweiten Bild ist ausgedrückt durch (x'_i, y'_i) . Die Transformation zwischen jeweils zwei Bildpunkten wird ausgedrückt durch (x'_i, y'_i) . Dabei besteht die Transformation aus dem Verschiebungsvektor $h = \begin{pmatrix} t_x \\ t_y \end{pmatrix}$ sowie der affinen Bewegung, die repräsentiert wird durch

$$\text{die Matrix } A = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix}.$$

x' und y' lassen sich somit darstellen durch:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} \cdot x + \begin{pmatrix} t_x \\ t_y \end{pmatrix}.$$

Oder kurz:

$$\sum (A\vec{x} + \vec{b} - \vec{\hat{x}}).$$

Dies führt dann zu folgendem Gleichungssystem:

$$\text{IV: } (x') = [a_{00}a_{01}](x) + (t_x) \quad \text{und} \quad (y') = [a_{10}a_{11}](y) + (t_y)$$

Ziel ist es nun eine gute Schätzung für die Werte $\{a_{ik}\}$, t_x, t_y zu erhalten. Hierzu muss ähnlich wie im eindimensionalen Fall eine Gleichung für den Fehler aufgestellt und diese Ableitung gleich Null gesetzt werden:

$$V: E = \sum_i (x'_i - \hat{x}_i)^2 + (y'_i - \hat{y}_i)^2 = \sum_i (ax_i + b - \hat{x}_i)^2 + (ay_i + b - \hat{y}_i)^2.$$

Mit dieser Gleichung berechnet man den Euklidischen Fehler. Dabei wird davon ausgegangen, dass Fehler bei der Messung der Punkte nur im veränderten /zweiten Bild und nicht im originalen/ersten Bild auftreten. Um nun die Parameter zu bestimmen, werden wie im eindimensionalen Fall die Ableitungen bestimmt und gleich Null gesetzt. Hieraus entsteht da folgende Gleichungssystem:

$$a_{00} \sum_i x_i^2 + a_{01} \sum_i x_i y_i + t_x \sum_i x_i = \sum_i \hat{x}_i x_i,$$

$$a_{00} \sum_i x_i y_i + a_{01} \sum_i y_i^2 + t_x \sum_i y_i = \sum_i \hat{x}_i y_i,$$

$$a_{00} \sum_i x_i + a_{01} \sum_i y_i + t_x \sum_i 1 = \sum_i \hat{x}_i,$$

$$a_{10} \sum_i x_i^2 + a_{11} \sum_i x_i y_i + t_y \sum_i x_i = \sum_i \hat{y}_i x_i,$$

$$a_{10} \sum_i x_i y_i + a_{11} \sum_i y_i^2 + t_y \sum_i y_i = \sum_i \hat{y}_i y_i,$$

$$a_{10} \sum_i x_i + a_{11} \sum_i y_i + t_y \sum_i 1 = \sum_i \hat{y}_i.$$

Eindimensionale projektive Bewegung

Die Betrachtung im projektiven Modell unterscheidet sich hauptsächlich darin, dass die zu ermittelnden Bewegungen nicht mehr durch affine Transformation Beschrieben werden können. Der Bildpunkt im zweiten Bild wird nun ausgedrückt durch $x'_i = (ax + b)/(cx + 1)$ (siehe Abbildung 2.6).

Wendet man nun wieder das gleiche Verfahren wie im affinen Modell an, so erhält man für den Fehler ein nicht-lineares Gleichungssystem, das sehr schwer zu lösen ist:

$$VI: E(a, b, c) = \sum_i (x'_i - \hat{x}_i)^2 = \sum_i \left(\frac{ax_i + b}{cx_i + 1} - \hat{x}_i \right)^2.$$

Hier kann man sich aber eines einfachen Tricks bedienen, um das Gleichungssystem zu linearisieren. Man normiert einfach das Gleichungssystem mit $(cx_i + 1)$ und berechnet so den algebraischen Fehler.

$$VII: E_a(a, b, c) = \sum_i \left(\left(\frac{ax_i + b}{cx_i + 1} - \hat{x}_i \right) (cx_i + 1) \right)^2 = \sum_i (ax_i + b - \hat{x}_i (cx_i + 1))^2.$$

Nun kann man wieder die Ableitungen berechnen und diese gleich Null setzen, um die Parameter zu bestimmen.

$$a \sum_i x_i^2 + b \sum_i x_i + c \sum_i -x_i^2 \hat{x}_i = \sum_i x_i \hat{x}_i,$$

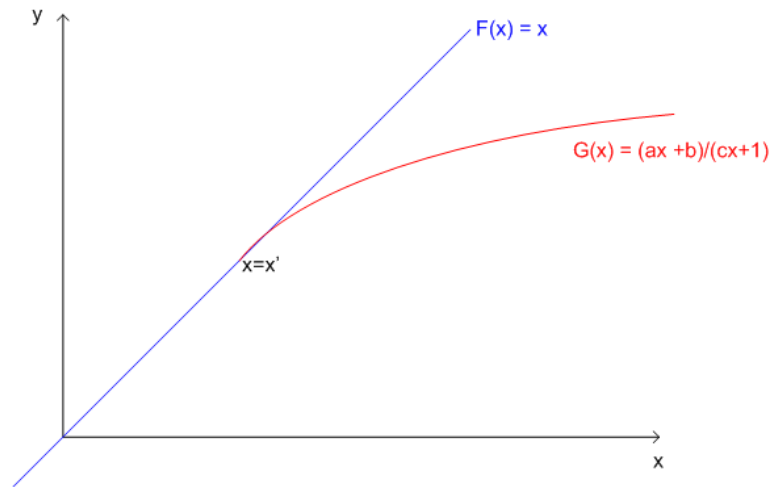


Abbildung 2.6: Projektion von $F(x)$ nach $G(x)$ (Quelle: eigene Darstellung)

$$a \sum_i x_i + b \sum_i 1 + c \sum_i -x_i \hat{x}_i = \sum_i \hat{x}_i,$$

$$a \sum_i x_i^2 \hat{x}_i + b \sum_i x_i \hat{x}_i + c \sum_i -x_i^2 \hat{x}_i^2 = \sum_i x_i \hat{x}_i^2.$$

Durch die Berechnung des algebraischen Fehlers ist es nun einfach, das Gleichungssystem zu lösen. Dies geht allerdings auf Kosten der Genauigkeit der Parameter. Anzumerken ist, dass die Ungenauigkeit für die Parameter relativ gering ist, sofern die Modelle wenig Rauschen aufweisen. Mit steigendem Rauschen der Modelle steigt allerdings auch der Fehler.

Zweidimensionale projektive Bewegung

Im zweidimensionalen Fall betrachtet man nun die homographie Matrix H . Eine homographie Matrix zeichnet sich dadurch aus, dass ein Punkt in einem Bild einem und genau einem Punkt im korrespondierenden Bild entspricht und umgekehrt. Mit Hilfe dieser Matrix wird nun die Bewegung eines Punktes p_i vom ersten Bild zum korrespondierenden Punkt $p'_i = Hp_i$ berechnet. Zu schätzen sind dann die Komponenten $\{ h_{ik} \}$ der Matrix H .

Ähnlich der Darstellung im eindimensionalen Fall geht man auch hier vor. Man geht davon aus, dass $h_{22} = 1$ gilt. Diese Annahme ist allerdings nur richtig für Bewegungen innerhalb eines Bildes. Für größere Rotationswinkel muss eine komplexere Berechnungsart gewählt werden, die in dieser Beschreibung allerdings nicht berücksichtigt wird. Somit

kann man also wieder die Koordinaten x' und y' mit $h_{22} = 1$ normieren und erhält:

$$\text{VIII: } x' = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + 1}, \quad y' = \frac{h_{10}x + h_{11}y + h_{12}}{h_{20}x + h_{21}y + 1}$$

Auch hier würde die Berechnung des Euklidischen Fehler (V) wieder zu einem sehr schwer zu lösenden Gleichungssystem führen. Darum wird wieder auf die Berechnung des Algebraischen Fehlers zurückgegriffen. Dies geschieht mittels der Normierung mit $(h_{20}x + h_{21}y + 1)^2$ und führt zu:

$$\begin{aligned} \text{IX: } E_a &= \sum_i ((x'_i - \hat{x}_i)^2 + (y'_i - \hat{y}_i)^2)(h_{20}x + h_{21}y + 1)^2 \\ &= (h_{00}x + h_{01}y + h_{02} - \hat{x}_i(h_{20}x + h_{21}y + 1))^2 + (h_{10}x + h_{11}y + h_{12} - \hat{y}_i(h_{20}x + h_{21}y + 1))^2 \end{aligned}$$

Auch hier wird wie in den anderen Fällen wieder die Ableitung aller Gleichungen gebildet und gleich Null gesetzt ($\partial E_a / \partial h_{ik} = 0$).

Sei nun

$$A_i = \begin{bmatrix} x_i^2 & x_i y_i & x_i & 0 & 0 & 0 & -x_i^2 \hat{x}_i & -x_i y_i \hat{x}_i \\ x_i y_i & y_i^2 & y_i & 0 & 0 & 0 & -x_i y_i \hat{x}_i & -y_i^2 \hat{x}_i \\ x_i & y_i & 1 & 0 & 0 & 0 & -x_i \hat{x}_i & -y_i \hat{x}_i \\ 0 & 0 & 0 & x_i^2 & x_i y_i & x_i & -x_i^2 \hat{y}_i & -x_i y_i \hat{y}_i \\ 0 & 0 & 0 & -x_i y_i & y_i^2 & y_i & x_i y_i \hat{y}_i & -y_i^2 \hat{y}_i \\ 0 & 0 & 0 & -x_i & y_i & 1 & x_i \hat{y}_i & -y_i \hat{y}_i \\ x_i^2 \hat{x}_i & x_i y_i \hat{x}_i & x_i \hat{x}_i & x_i^2 \hat{y}_i & x_i y_i \hat{y}_i & x_i \hat{y}_i & -x_i^2 \hat{s}_i & -x_i y_i \hat{s}_i \\ x_i y_i \hat{x}_i & y_i^2 \hat{x}_i & y_i \hat{x}_i & x_i y_i \hat{y}_i & y_i^2 \hat{y}_i & y_i \hat{y}_i & -x_i y_i \hat{s}_i & -y_i^2 \hat{s}_i \end{bmatrix}$$

und

$$b_i = (x_i \hat{x}_i \quad y_i \hat{x}_i \quad \hat{x}_i \quad x_i \hat{y}_i \quad \hat{y}_i \quad x_i \hat{s}_i \quad y_i \hat{s}_i)^T$$

mit der Abkürzung $\hat{s}_i = (\hat{x}_i^2 + \hat{y}_i^2)$. Dann lässt sich das Gleichungssystem ausdrücken mit

$$(\sum_i A_i)h = \sum_i b_i$$

und die Lösung durch

$$h = (h_{00} \quad h_{01} \quad h_{02} \quad h_{10} \quad h_{11} \quad h_{12} \quad h_{20} \quad h_{21})^T.$$

2.2.3 Erzeugung von Panoramabildern

Möchte man ein Panoramabild aus mehreren Einzelbildern zusammensetzen, so setzt dies voraus, dass die einzelnen Bilder aus der gleichen Perspektive aufgenommen worden sind.

Dies ist beispielsweise der Fall, falls die einzelnen Bilder mit einer festen Kameraposition photographiert wurden. Diese Art der Kameraführung ist aber nur in selten Fällen vorhanden. Zumeist werden Bilder von Hand aufgenommen und es verschiebt sich somit die Ausrichtung der Kamera. Oft kommt zusätzlich noch eine Zoombewegung hinzu. Möchte man nun die so entstandenen Bilder aber zu einem Panoramabild zusammen setzen, ist es notwendig, diese Kamerabewegung zu kennen und bei der Zusammensetzung die Parameter entsprechend anzupassen. Kennt man zum einen das Kameramodell und zum anderen (einige) korrespondierende Pixel in zwei Bildern, so kann man daraus die Parameter der Kamera berechnen. Kombiniert man dieses Wissen über die Ausrichtung der Kamera dann mit einer eventuell notwendigen Transformation der Helligkeit, so kann man die einzelnen Bilder zu einem Panoramabild zusammensetzen (siehe Abbildung 2.7).



Abbildung 2.7: Erstellung eines Panoramabildes (Quelle: eigene Darstellung)

2.3 Fazit

Beide vorgestellten Modelle betrachten zuerst den einfachen eindimensionalen Fall und wenden dort lineare Verfahren an, um den Fehler bzw. die Parameter zu berechnen. In einem zweiten Schritt werden die Verfahren dann auf mehr Dimensionen ausgedehnt und der quadratische Fehler in die Berechnung integriert.

Beide vorgestellten Verfahren zur Bestimmung der Parameter des Kameramodells eignen sich in dieser Form nur für einfache Bewegungen. Darunter sind entweder reine Bewegungen der Kamera oder reine Bewegungen der Objekte im Bild zu verstehen. Um sowohl die Bewegung der Kamera in Kombination mit Bewegungen von Objekten zu kombinieren, kann man beispielsweise das zweite Verfahren mit Hilfe des RANSAC Algorithmus erweitern. Hier haben Tests allerdings gezeigt, dass die tatsächliche Leistung des Algorithmus nicht an die theoretische Leistung des Algorithmus heranreicht ([1]). Ein weiterer Nachteil des RANSAC-Algorithmus ist, dass er nur mit einer gewissen

Wahrscheinlichkeit zu einem korrekten Ergebnis gelangt. Zur Bestimmung der Parameter in komplexen Bewegungen sind also weitere Algorithmen und Vorgehensweisen nötig.

Im zweiten vorgestellten Verfahren wird ausserdem davon ausgegangen, dass bei der Berechnung der korrespondierenden Bildpunkte nur Fehler bei der Messung im Abbild des Originals auftreten. Dies ist allerdings nicht realistisch, da auch bei der Berechnung der ursprünglichen Bildpunkte Fehler auftreten können. Um dies zu berücksichtigen, ist eine weitergehende Betrachtung des Fehler notwendig. Hier reicht es also nicht mehr, den Euklidischen Abstand des Fehlers im zweiten Bild in die Berechnung einzubeziehen, sondern es muss eine symmetrische Fehlerdefinition wie z.B. der Reprojektionsfehler genutzt werden. Dies führt dann zu einem weitaus komplexeren Verfahren, das allerdings nur wenig mehr Genauigkeit beim Ergebnis liefert.

Literaturverzeichnis

- [1] Dirk Farin. *Automatic Video Segmentation Employing Object/Camera Modeling Techniques*. PhD thesis, University of Eindhoven (Netherlands), 2005.
- [2] B. Lucas and T. Kanade. Optical navigation by the method of differences. *International Joint Conference on Artificial Intelligence*, pages 981–984, 1985.
- [3] B.D. Lucas. *Generalized Image Matching by the Method of Differences*. PhD thesis, Carnegie-Mellon University, Dept. of Computer Science, 1985.
- [4] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 674–679, April 1981.