

Image Stitching

EDBV WS 2014/2015: AG_B4

J. Sebastian Kirchner (0926076)

Hanna Huber (0925230)

Patrick Wahrmann (1327120)

Ernad Sehic (1227865)

Nikolaus Leopold (1327344)

2. Januar 2015

1 Gewählte Problemstellung

(1-1,5 Seiten)

entspricht dem (aktualisierten) Konzept

1.1 Ziel

1.2 Eingabe

1.3 Ausgabe

1.4 Voraussetzungen und Bedingungen

1.5 Methodik

1.6 Evaluierungsfragen

2 Arbeitsteilung

(0,5 Seiten)

Wer hat welche Aufgaben übernommen (MATLAB-Funktionen, Abschnitt im Bericht, Evaluierung, Datenerfassung, etc.)?

Name	Tätigkeiten
Vorname1 Nachname1	Matlab-Funktion A, Bericht Abschnitt B...
Vorname2 Nachname2	Matlab-Funktion C, Bericht Abschnitt D...

3 Methodik

Ein wesentlicher Teil des Image Stitching ist die Korrespondenzanalyse. Dabei werden für den überlappenden Bereich der Bilder jene Bildpunkte in beiden Bildern gesucht, die dasselbe Objekt darstellen. Dieses Problem wird mithilfe von merkmalsbasiertem Matching gelöst, das gegenüber regionenbasiertem Matching den Vorteil hat, dass homogene und daher irrelevante Bildbereiche nicht in die Berechnung mit einfließen. Der Aufwand wird dadurch erheblich reduziert. [4]

Die Merkmale - Interest Points oder Keypoints - werden mittels Scale Invariant Feature Transform (kurz: SIFT) [3] gefunden. Dieser Algorithmus hat den Vorteil, dass das Ergebnis unabhängig von der Skalierung oder Orientierung der Interest Points im jeweiligen Bild ist [4].

3.1 SIFT

Der SIFT-Algorithmus ist in mehrere Schritte unterteilt, in denen Skalierung, Position und Orientierung der Interest Points ermittelt wird. Abschließend muss jeder Punkt eindeutig beschrieben werden, um die Korrespondenzanalyse zu ermöglichen.

3.1.1 Bildpyramiden

Um die Skalierungsinvarianz zu garantieren, ist eine Multiskalenanalyse der Bilder notwendig.

3.1.2 Position der Interest Points

Interest Points stellen besonders markante, in einer lokalen Umgebung möglichst einzigartige Bildpunkte mit einer klaren Position dar. Punkte, an denen sich im DoG-Bild ein lokales Minimum oder Maximum befindet, sind somit gute Kandidaten für Interest Points. Die Extrema werden in jeder Oktave der DoG-Bildpyramide gesucht. Für jedes Pixel werden dafür die umliegenden Nachbarn derselben, sowie der darüber und darunter liegenden Ebene betrachtet.

Um die Position des Extremums noch exakter - also im Subpixelbereich - zu bestimmen, wird danach noch eine Taylorapproximation durchgeführt. Schließlich werden Punkte mit zuringem Kontrast und jene, die an einer Kante liegen, verworfen.

3.1.3 Orientierung der Interest Points

Nun wird für jeden Keypoint die Orientierung im Bild bestimmt. Damit ist die Richtung des maximalen Gradienten gemeint. Die folgende Beschreibung des Punktes geschieht relativ zu dieser Orientierung. Dadurch ist der SIFT-Algorithmus rotationsinvariant.

3.1.4 Deskriptoren

Jeder Keypoint muss eindeutig beschrieben werden, damit Punkte verschiedener Bilder, die denselben Interest Point beschreiben einander zugeordnet werden können.

3.2 Matching

Nun müssen Keypoints, die denselben Interest Point beschreiben, einander zugewiesen werden. Wie bei Lowe[3] wird dazu der Nearest-Neighbor-Ansatz gewählt.

Allerdings wird statt der euklidischen Distanz der Winkel zwischen den Deskriptor-Vektoren der entsprechenden Keypoints zwischen den Vektoren minimiert. Dieser lässt sich leicht aus dem Skalarprodukt der beiden Vektoren ermitteln. Für jeden Deskriptor des ersten Bildes können die entsprechenden Skalarprodukte für alle Deskriptoren des zweiten Bildes mittels einer einzigen Vektor-Matrix-Multiplikation berechnet werden.

Um die Eindeutigkeit der Zuweisung zu garantieren, wird der kleinste Winkel mit dem zweitkleinsten verglichen [3]. Nur wenn deren Verhältnis unter einem Schwellwert liegt, wird die Zuweisung akzeptiert.

Zusätzlich wird überprüft, ob der Deskriptor mit minimalem Winkelabstand bereits einem zuvor betrachteten Deskriptor zugewiesen wurde. In diesem Fall wird dieser durch den aktuellen Deskriptor ersetzt.

3.3 Transformation

Um die beiden Bilder zu einem Bild zusammenzuführen, muss eine Homographie-Matrix H ermittelt werden, die die Koordinaten der Bildpunkte eines Bildes in entsprechende Koordinaten im Koordinatensystem des anderen Bildes umwandelt. Dies gilt insbesondere für Keypoint-Paare, also $X_2 = H \cdot X_1$ für ein Keypoint-Paar (X_1, X_2) . H wird mithilfe des Random-Sample-Consensus-Algorithmus[1] berechnet.

Die Information über die Koordinaten der Keypoint-Paare wird verwendet, um ein lineares Gleichungssystem aufzustellen, dessen Lösung die

Koeffizienten der Matrix H liefert. Dafür werden vier Keypoint-Paare benötigt. [2]

Der RANSAC-Algorithmus wählt diese in jedem Iterationsschritt - deren Anzahl wird zuvor festgelegt – zufällig aus und berechnet die zugehörige Matrix H . Daraufhin wird die Homographie auf alle zu einem Keypoint-Paar (X_1, X_2) gehörigen Punkte X_1 angewendet. Liegt der transformierte Punkt innerhalb eines Toleranzbereichs um den Punkt X_2 , wird er als *Inlier* bezeichnet. In jedem Iterationsschritt, also für jedes H , wird die Anzahl der Inlier berechnet. Am Ende wird die Matrix mit den meisten Inliers als Homographie-Matrix gewählt.

Da eine korrekte Lösung des oben beschriebenen Gleichungssystems stark von Ursprung und Skalierung des Koordinatensystems der Bilder abhängt, werden die Koordinaten der Keypoint-Paare zuvor normalisiert. Um die endgültige Homographie-Matrix zu erhalten, wird die Matrix H am Ende noch mit den entsprechenden Transformationsmatrizen multipliziert.[1]

3.4 Stitching

Abschließend werden die beiden Bilder zu einem Bildmosaik zusammengefügt.

4 Implementierung

(1-X Seiten)

Hier gebt ihr einen Überblick über eure Implementierung:

Wie habt ihr die im vorhergehenden Abschnitt vorgestellte Methodik praktisch umgesetzt? Wie werden die einzelnen Methoden kombiniert (zB. Implementierungspipeline)?

Hier ist Platz für Implementierungsdetails wie zB. gewählte Parameter.

Wie startet der User das Programm? Welche Parameter hat der User zu setzen?

Auch in diesem Abschnitt können Referenzen und Zitate notwendig sein.

5 Evaluierung

(2-X Seiten)

Hier stellt ihr euren Datensatz vor und beantwortet Evaluierungsfragen: z.B. Fakten zum Datensatz: Anzahl der Bilder, Größe der Bilder, Quelle

des Datensatzes (falls selbst aufgenommen: Aufnahmegerät, Einstellungen,... / falls nicht selbst erstellt: Datenbank vorstellen...)

Diskussion der Evaluierungsfragen: Beantwortung der Fragen, Diskussion anhand von Beispielen, Diskussion von Grenzfällen: für welche Bilder funktioniert die Implementierung, für welche nicht? Worin unterscheiden sich diese Bilder? etc.

6 Schlusswort

(max. 1 Seite)

Hier fasst ihr Ergebnisse eures Projekt zusammen:

Welche Schlussfolgerung lässt sich ziehen? Gibt es offene Probleme? Wie lässt sich eure Lösung noch verbessern? etc.

Literatur

- [1] Elan Dubrovsky. Homography Estimation. Master's thesis, The University of British Columbia, 2009.
- [2] David Kriegman. Homography Estimation, 2007.
- [3] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [4] Robert Sablatnig and Werner Purgathofer. Einführung in Visual Computing: Skriptum zur VU, 2014.