Seminar für überfachliche Grundlagen

Medizinische Informatik

Thema:

Bildfusion

von

Stefan Scharinger

Inhaltsverzeichnis

1 Einleitung

Zu Beginn dieser Arbeit möchte ich zuerst einmal aufzeigen, was man sich eigentlich unter dem Begriff Bildfusion vorzustellen hat. Hierbei handelt es sich um das „Verschmelzen“ von mehreren Bildern zu einer einzigen Darstellung. Dies kann z.B. durch das transparente Überlagern eines Bildes mit einem anderen geschehen, auf die Möglichkeiten der Darstellung werde ich im Kapitel 2 Darstellung näher eingehen.

An dieser Stelle wird sich nun vielleicht die Frage stellen, wer überhaupt einen Grund haben soll, zwei verschiedene Bilder gleichzeitig darzustellen. Um dieser Frage gerecht zu werden möchte ich im Folgenden einige Beispiele anführen, bei denen die Bildfusion in der Medizinischen Informatik Anwendung findet und die Arbeit des medizinischen Personals deutlich erleichtert oder sogar die Qualität verbessern kann. Vorweg bleibt noch zu sagen, dass die Bildfusion nicht nur in der Medizin Anwendung findet, sondern auch z.B. zur Oberflächenanalyse in der Industrie verwendet wird, worauf ich aber an dieser Stelle nicht weiter eingehen möchte, da sich das Seminar in dessen Rahmen diese Ausarbeitung und der zugehörige Vortrag entstanden sind ausdrücklich auf Medizinische Informatik bezieht.

In der Medizin arbeitet man sehr oft mit Bilddaten über deren Erzeugung bereits in 2 Vorträgen in diesem Seminar berichtet wurde.

Für den Arzt von Interesse Zeitserien zu vergleichen; dabei werden von einem Patienten in bestimmten Abständen Aufnahmen angefertigt (meist mit demselben bildgebenden Verfahren), die anschließend auf eine Veränderung, z.B. das mögliche Wachstum eines Tumors hin verglichen werden sollen. In diesem Fall ist es deutlich einfacher eine Veränderung zu erkennen, wenn man die beiden Bilder genau überlagert. Veränderungen werden so auf einen Blick sichtbar. Man muss dadurch nicht in beiden Bildern die jeweiligen Größen der zu untersuchenden Strukturen bestimmen, die ohnehin nur schwer zu vergleichen wären, da nicht davon auszugehen ist, dass die Größenverhältnisse in beiden Aufnahmen exakt gleich sind. Ein weiterer Anwendungsfall ist die Kombination mehrerer Aufnahmetechniken. So kann man an einem CT die Anatomie eines Patienten sehr gut erkennen, mit einer PET hingegen kann man bei entsprechenden Einstellungen Stoffwechselfunktionen darstellen. Die Zuordnung eines solchen PET-Funktionsbildes zu den jeweiligen Teilen bestimmter Organe ist jedoch mit dem bloßen Auge nicht ohne weiteres zu bewerkstelligen. Auch hierbei bietet die Bildfusion Hilfe.

Wie oben bereits erwähnt werden bei der Bildfusion auch CT oder PET Datensätze verwendet, deshalb möchte ich darauf hinweisen, dass im Folgenden genauso wie oben mit Bildern, ober Bilddaten immer sowohl 2D als auch 3D Bilder (Volumen) gemeint sind. Diese Erweiterung ist von großer Bedeutung, da ein Großteil der für Bildfusion verwendeten Daten Volumen sind.

In Kapitel 3 werde ich mehrere Verfahren zur Registrierung, also der mathematischen Berechnung der Positionierung von Bildern näher vorstellen.

2. Darstellung

Zu den Möglichkeiten der Darstellung der registrierten Bilder muss im Grunde nur gesagt werden, dass prinzipiell alle möglichen Berechnungen für die Farb-/Grauwerte der Pixel/Voxel des neuen Bildes aus den entsprechenden beiden Pixeln der Ursprungsbilder vorstellbar sind. So kann man beispielsweise eines der Bilder als Hintergrund definieren und das andere transparent darüber legen, oder aber einen bestimmten Farbwert als Basis, entsprechend dem Mittelwert der Bilder benutzen und farbkodiert die Abweichung davon in den jeweiligen Bildern darstellen. Häufig gebraucht ist die Einbettung eines farbigen in ein schwarz-weißes Bild (im Beispiel PET und CT in Schichtaufnahmen und als Volumen).

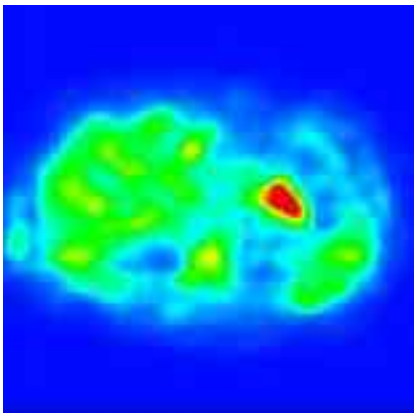

Abbildung 1PET des Abdomen - Schichtbild

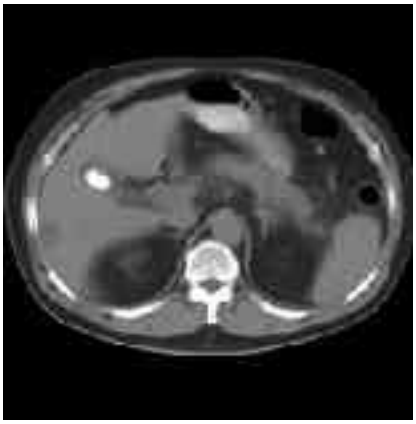

Abbildung 2CT des Abdomen - Schichtbild

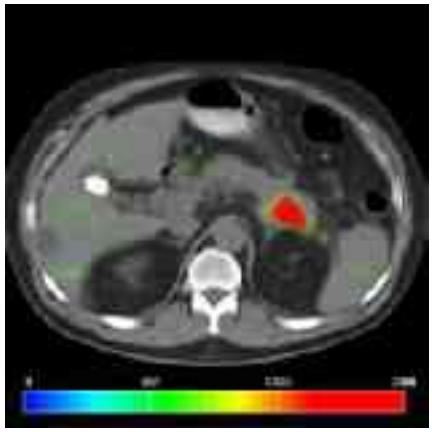

Abbildung 3CT und PET des Abdomen - Schichtbild

Entsprechend dieser 2D Bildfusion kann man auch ein PET Volumen in ein CT Volumen einfügen:

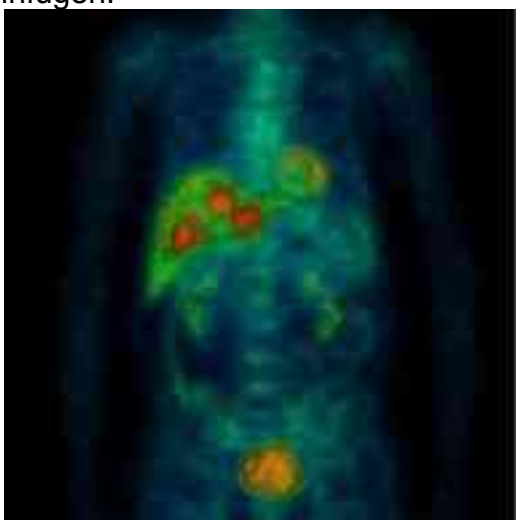

Abbildung 4PET Volumen

Abbildung 5PET Volumen in CT Volumen

Je nach den Anforderungen des Anwendungsfalles sind noch beliebige weitere Darstellungsformen denkbar, die jedoch im Vergleich zur Registrierung einfach zu implementieren sind, weshalb ich auf die Darstellung nicht weiter eingehen und im Folgenden den aufwändigeren Schritt der Registrierung behandeln möchte.

3. Registrierung

a) Einleitung

Der wohl wichtigste Schritt bei der Bildfusion ist die Registrierung. Sie wird im Englischen oft mit matching bezeichnet, wofür es aber im Deutschen leider kein passendes Wort gibt. Deshalb werde ich in dieser Arbeit nur den Begriff der Registrierung verwenden. Unter Registrierung versteht man den Arbeitsschritt, in dem für eines der beiden zu verschmelzenden Bilder eine Transformation erzeugt wird, so dass sich die beiden Bilder genau überlagern.
Relativ trivial ist dies, wenn man die Möglichkeit hat die beiden Aufnahmen gleichzeitig und mit bekannten „Kamera“-positionen und -ausrichtungen anzufertigen. Dies ist z.B. bei kombinierten Geräten, die in einem Arbeitsschritt eine CT und eine PET anfertigen der Fall. Hierbei muss man nur bei der Konstruktion des Gerätes einmalig ein Transformation zwischen den jeweiligen Koordinatensystemen der Geräte berechnen und kann diese bei jeder angefertigten Aufnahme verwenden. Dieses Verfahren kann auch zur Anwendung kommen, wenn eine der beiden Aufnahmen mit einem kleinen transportablen Gerät angefertigt wird (z.B. Ultraschall, EKG,...). Dann muss allerdings die Position z.B. des Ultraschallgerätes exakt bestimmt und die Transformation angepasst werden.
Leider ist ein solches Verfahren eher die Ausnahme, teilweise auch nicht realisierbar, z.B. bei Zeitserien von Aufnahmen. Diese erstrecken sich oft über Wochen oder Monate und es ist unmöglich den Patienten bei jeder dieser Aufnahmen in exakt die selbe Position zu bringen. In anderen Fällen sind schlichtweg die Geräte zu groß und unflexibel um die Aufnahmen am selben Ort durchführen zu können. Aus diesem Grund verwendet man häufig andere Verfahren um die Bilder zu registrieren. Diese Verfahren verwenden nur die Daten, die die Bilder selbst liefern. Man unterscheidet hier aber zwischen den so genannten inneren (intrinsic) und den äußeren (extrinsic) Methoden. Der Unterschied besteht darin, dass bei den intrinsic Varianten nur anatomische Daten verwendet werden und in den extrinsic Varianten zusätliche Marker am Patienten angebracht werden. Diese Marker werden meist auf die Haut aufgeklebt. Die Verfahren sind jedoch nicht auf oberflächliche Marker festgelegt. Der große Vorteil dieser Marker besteht darin, dass sie speziell für diesen Zweck entwickelt wurden und deshalb auf den entsprechenden Bildern leicht zu erkennen und auch leicht durch automatisierte Prozesse zu identifizieren und zu segmentieren sind. Man kann also eine im Prinzip vollständig automatisierte Registrierung mit Hilfe der Marker durchführen. Die Platzierung der Marker, vor allem implantierter und nicht nur aufgeklebter Marker, ist allerdings für den Patienten eher unangenehm und kann zudem zu Nebenwirkungen führen, wenn diese Marker über einen längeren Zeitraum im Körper verbleiben. Als weiteres Problem eines derartigen Verfahrens vor allem bei Zeitserien ist, dass die aufgeklebten Marker oft schon nach einer relativ kurzen Zeit verrutschen und damit das Ergebnis der Registrierung verfälschen. Z.B. bei Krebs-Nachsorge, zu deren Zweck jedes halbe Jahr eine Aufnahme angefertigt wird ist das Anbringen und Verbleiben lassen von Markern nicht praktikabel. Außerdem besteht bei Marker basierten Verfahren nicht die Möglichkeit Bilder nachträglich zu registrieren, also z.B. aktuelle Aufnahmen mit schon früher gemachten zu vergleichen, da bei diesen die Marker nicht vorhanden, oder falsch positioniert sind. Anwendungen wie Augmented Reality, bei denen vorher gemachte Aufnahmen mit einer aktuellen Operation überlagert werden sollen sind jedoch nach heutigem Stand nicht ohne Marker zuverlässig realisierbar.

Neben der eben erläuterten Unterscheidung nach dem Ursprung der Informationen auf denen die Registrierungsalgorithmen beruhen kann man die Verfahren auch nach der Art der von ihnen verwendeten Information unterscheiden. Dabei sind Verfahren, die einzelne Punkte aus den Bildern verwenden genauso möglich wie die Verarbeitung von Geraden, komplett segmentierten Bildern, oder den gesamten Grauwerten eines Bildes.
Eine anderer wichtiger Unterschied zwischen den verschiedenen Registrierungsverfahren ist die berechnete Transformation. Dabei gibt es Verfahren die affine oder sogar sog. curved oder elastische (eine passendere Übersetzung konnte ich leider nicht finden) Transformationen erlauben. In medizinischen Anwendungen werden beinahe ausschließlich rigide Transformationen verwendet, die nur eine Translation, Rotation und Skalierung erlauben. Sie werden oft auch als Starrkörpertransformationen bezeichnet, da sich die Form eines transformierten Objekts nicht ändert. Bei affinen Transformationen sind zusätzlich perspektivische Abbildungen erlaubt, bei denen nur Geraden erhalten werden, nicht jedoch Winkel oder Parallelen. Diese Abbildungen finden in manchen Fällen auch in der Medizin Anwendung, wenn man z.B. ein 2D Röntgenbild und ein 3D Volumen registrieren will. Bei curved Transformationen werden nicht einmal Geraden erhalten, aus ihnen können beliebige Kurven werden. Die Übersetzung elastisch für curved ist nicht ganz adäquat, weil es genau genommen nur ein Teilgebiet der curved Transformationen beschreibt, da bei diesen normalerweise keine elastischen Gesetze berücksichtigt werden. Auf die in der Medizin verwendeten elastischen Transformationen trifft diese Einschränkung aber im Grunde genommen zu. In den wenigen Fällen in denen sie zum Einsatz kommen geht es darum Bilder zu registrieren, bei denen Organe elastisch verformt wurden.
Neben der Wahl einer der eben genannten Transformationsarten (Suchraum) und der zuvor erläuterten Festlegung der verarbeiteten Daten (Merkmalsraum, Punkte, Geraden, gesamtes Bild....) muss man sich bei allen Registrierungsverfahren im Voraus auf eine Suchstrategie festlegen. Ein letztes wichtiges Kriterium, das vor der Erstellung eines Algorithmus entschieden werden sollte, ist die Festlegung des gewünschten Ähnlichkeitsmaßes. Dieses ist vor allem bei iterativen Verfahren von sehr großer Bedeutung, da man anhand dieses Maßes feststellt, wann der Algorithmus terminieren soll.
Im Folgenden möchte ich zwei Gruppen dieser Registrierungsverfahren etwas genauer beleuchten. Zum einen Punkt-Korrespondenzen und zum anderen die sog. Mutual Information.

b) Punkt Korrespondenzen

bAllgemeines

Für die Berechnung einer Transformationsmatrix über Punkt Korrespondenzen benötigt man zu aller erst einen Satz Punkte im einen Bild und den jeweiligen entsprechenden im anderen. Bei der Beschaffung dieser Punkte liegt dabei die größte Fehlerquelle, falls sie nicht automatisch über Marker erkannt werden können. Meist kann man nur sehr schlecht Punktkorrespondenzen über vollautomatische Verfahren aus anatomischen Daten beziehen. Es ist allerdings auch denkbar die Festlegung der entsprechenden Punkte komplett dem Benutzer zu überlassen. Da die Weise, auf die die Punkte gewonnen werden keinen Einfluss auf das Verfahren der Registrierung haben möchte ich an dieser Stelle nicht weiter auf die Generierung der Punkte eingehen.
Im nächsten Abschnitt möchte ich eine geschlossene Form vorstellen, wie man eine solche Transformation mit Hilfe von Quaternionen und 3 oder mehr Punkten aufstellt.

Bestimmung der Transformation mit Quaternionen

Was sind Quaternionen

Quaternionen sind eine Art „erweiterte komplexe Zahlen“. Im Gegensatz zu gewöhnlichen komplexen Zahlen besitzen sie jedoch 3 verschieden Imaginärteile i , j und k. Ein Quaternion $\dot{q}$ hat die Form: $\dot{q}=q_0+iq_x+jq_y+kq_z$.
Für die Imaginärteile gelten folgende Regeln:

$$i^2=-1;\quad j^2=-1;\quad k^2=-1;$$

$$ij=k;\quad jk=i;\quad ki=j;$$

$$ji=-k;\quad kj=-i;\quad ik=-j;$$

Wie aus diesen Regeln bereits zu sehen ist sind die Imaginärteile und damit auch Quaternionen bezüglich der Multiplikation nicht kommutativ. Die Multiplikation von $\dot{q}$ mit dem Quaternion $\dot{r}=r_0+ir_x+jr_y+kr_z$ ergibt durch einfaches ausmultiplizieren:

$$\begin{aligned}\dot{r}\,\dot{q}=&(r_0q_0-r_xq_x-r_yq_y-r_zq_z)\\&+i(r_0q_x+r_xq_0+r_yq_z-r_zq_y)\\&+j(r_0q_y-r_xq_z+r_yq_0+r_zq_x)\\&+k(r_0q_z+r_xq_y-r_yq_x+r_zq_0)\end{aligned}$$

für diese Multiplikation gibt es zusätzlich 2 Möglichkeiten, sie als Produkt zwischen einer Matrix und einem Vektor darzustellen. Dabei kann entweder der erste, oder der zweite Quaternion zu einer orthogonalen 4x4 Matrix erweitert werden:

$$\dot{r}\,\dot{q}=\begin{bmatrix}r_0 & -r_x & -r_y & -r_z\\ r_x & r_0 & -r_z & r_y\\ r_y & r_z & r_0 & -r_x\\ r_z & -r_y & r_x & r_0\end{bmatrix}\dot{q}=\mathbb{R}\,\dot{q}$$

oder

$$\dot{q}\,\dot{r}=\begin{bmatrix}r_0 & -r_x & -r_y & -r_z\\ r_x & r_0 & r_z & -r_y\\ r_y & -r_z & r_0 & r_x\\ r_z & r_y & -r_x & r_0\end{bmatrix}\dot{q}=\bar{\mathbb{R}}\,\dot{q}$$

Achtung! : $\bar{\mathbb{R}}$ entsteht aus $\mathbb{R}$ durch transponieren der unteren rechten 3x3 Untermatrix.

Wie im oberen Fall werden Quaternionen oftmals auch in anderen Zusammenhängen als

4dim. Vektoren verwendet $\dot{q}=\begin{pmatrix} q_0 \\ q_x \\ q_y \\ q_z \end{pmatrix}$ .Dementsprechend ist für sie ein Skalarprodukt als Summe der Produkte definiert $\dot{p}\cdot\dot{q}=p_0q_0+p_xq_x+p_yq_y+p_zq_z$

Das Quadrat der Größe eines Quaternions ist also das Skalarprodukt des Quaternions mit sich selbst $\|\dot{q}\|^2=\dot{q}\cdot\dot{q}$

Ein Einheitsquaternion ist ein Quaternion mit der Größe 1. Die Konjugation kehrt das Vorzeichen aller Imaginärteile um: $\dot{q}=q_0+iq_x+jq_y+kq_z, \quad \tilde{\dot{q}}=q_0-iq_x-jq_y-kq_z$

Die zum Konjugierten von $\dot{q}$ gehörende 4x4 Matrizen sind genau die jeweils Transponierten der zu $\dot{q}$ gehörenden. Für diese Matrizen gilt weiterhin $QQ^T=q\cdot\tilde{q}I$ daraus folgt:

$\dot{q}\,\tilde{\dot{q}}=(q_0^2+q_x^2+q_y^2+q_z^2)=\dot{q}\cdot\dot{q}$; daraus wiederum folgt, dass es für jeden Quaternion $\dot{q}\neq 0$ ein Inverses gibt : $\dot{q}^{-1}=(\frac{1}{\dot{q}}\cdot\dot{q})\dot{q}$

Zuletzt noch einige Eigenschaften der Produkte:

Das Skalarprodukt bleibt erhalten:

$(\dot{q}\,\dot{p})\cdot(\dot{q}\,\dot{r})=(Q\,\dot{p})\cdot(Q\,\dot{r})=(Q\,\dot{p})^T(Q\,\dot{r})=\dot{p}^TQ^TQ\,\dot{r}=\dot{p}^T(\dot{q}\cdot\dot{q})I\,\dot{r}=(\dot{q}\cdot\dot{q})(\dot{p}\cdot\dot{r})$

$Daraus\ folgt: (\dot{q}\,\dot{p})(\dot{q}\,\dot{r})=(\dot{q}\cdot\dot{q})(\dot{p}\cdot\dot{r})$

Daraus ergeben sich 2 Spezialfälle:

$\dot{q}\ ist\ ein\ unit\ Quaternion: (\dot{q}\,\dot{p})\cdot(\dot{q}\,\dot{r})=\dot{p}\cdot\dot{r}$

$und\ (\dot{p}\,\dot{q})\cdot(\dot{p}\,\dot{q})=(\dot{p}\cdot\dot{p})(\dot{q}\cdot\dot{q})$

Was als, der Betrag des Produkts ist gleich dem Produkt der Beträge zu deuten ist.

Des weiteren folgt: $(\dot{p}\,\dot{q})\cdot\dot{r}=\dot{p}\cdot(\dot{r}\,\tilde{\dot{q}})$

Vektoren und Skalare können leicht als Quaternionen dargestellt werden:

Ein Vektor $r=\begin{pmatrix} x \\ y \\ z \end{pmatrix}$ entspricht dabei einem rein imaginären Quaternion

$\dot{r}=0+i\,x+j\,v+k\,z$

und ein Skalar q einfach dem rein realen Quaternion $\dot{q}=q$ Für die Matrizen der rein imaginären Quaternionen gilt außerdem:

$\mathbb{R}^T=-\mathbb{R}\ und\ \bar{\mathbb{R}}^T=-\bar{\mathbb{R}}$

Zusammenhang zwischen Rotation und Quaternionen

Soweit zur kurzen Einführung in das Rechnen mit Quaternionen und die dafür nötigen Grundregeln. Nun drängt sich vielleicht die Frage auf, welchen Sinn Quaternionen im Zusammenhang mit dem Finden einer Transformation haben. Deshalb möchte ich versuchen im folgenden Abschnitt aufzuzeigen, wie Rotationen und Quaternionen zusammenhängen und anschließend erläutern, warum es günstiger ist einen Quaternion anstatt einer Rotationsmatrix zu bestimmen.

Da die Rotation weder die Länge eines Vektors, noch die Winkel zwischen Vektoren verändert erhält sie also das Skalarprodukt. Eine Spiegelung würde das zwar auch, diese ändert aber die Richtung des Kreuzprodukts. Wenn es uns also gelingt eine Abbildung zu finden, die rein imaginäre Quaternionen wieder auf rein imaginäre Quaternionen abbildet und dabei das Skalarprodukt erhält und auch die Richtung des Kreuzprodukts nicht

verändert können wir Rotationen als Einheitsquaternionen darstellen.
Die Erhaltung des Skalarprodukts haben wir bereits bei der Multiplikation mit einem Einheitsquaternion gezeigt:
$(\dot{q}\,\dot{p})\cdot(\dot{q}\,\dot{r}) = \dot{p}\cdot\dot{r}$
Leider ist das Produkt eines rein imaginären Quaternionen mit einem Einheitsquaternion in den meisten Fällen nicht rein imaginär, weshalb wir die gewöhnliche Multiplikation nicht verwenden können. Was ich nun aber zeigen möchte ist, dass das gemischte Produkte $\dot{r}' = \dot{q}\,\dot{r}\,\tilde{\dot{q}}$ rein imaginär ist:

$$\dot{q}\,\dot{r}\,\tilde{q} = (Q\,\dot{r})\tilde{q} = \bar{Q}^T(Q\,\dot{r}) = (\bar{Q}^T Q)\,\dot{r}$$

mit Q und $\bar{Q}$ als zu $\dot{q}$ gehörende 4 x4 Matrizen:

$$\bar{Q}^T Q = \begin{bmatrix} \dot{q}\cdot\dot{q} & 0 & 0 & 0 \\ 0 & (q_0^2+q_x^2-q_y^2-q_z^2) & 2(q_x q_y - q_0 q_z) & 2(q_x q_z + q_0 q_y) \\ 0 & 2(q_y q_x + q_0 q_z) & (q_0^2-q_x^2+q_y^2-q_z^2) & 2(q_y q_z - q_0 q_x) \\ 0 & 2(q_z q_x - q_0 q_y) & 2(q_z q_y + q_0 q_x) & (q_0^2-q_x^2-q_y^2+q_z^2) \end{bmatrix}$$

Daraus folgt, dass r' rein imaginär ist, wenn es auch $\dot{r}$ war. Wenn $\dot{q}$ ein Einheitsquaternion ist sind Q und $\bar{Q}$ orthonormal, da in diesem Fall $\dot{q}\cdot\dot{q} = 1$ muss also die rechte untere 3x3 Teilmatrix von $\bar{Q}^T Q$ ebenfalls orthonormal sein. Diese Matrix ist genau die Rotationsmatrix R, die r in r' überführt:
r'=Rr
Für den Beweis, dass auch das Kreuzprodukt erhalten bleibt möchte ich noch zusätzlich eine weitere Betrachtungsweise für Quaternionen einführen. Neben den bereits bekannten zwei Sichtweisen als Summe eines Real- und 3 Imaginärteilen und der Betrachtung als 4 dim. Vektor kann man Quaternionen auch als Summe aus einem Skalar und einem 3 dim. Vektor betrachten, so dass
$\dot{q} = q_0 + q_x + q_y + q_z = q + \boldsymbol{q}$ *mit* $q = q_0$ *und* $\boldsymbol{q} = (q_x, q_y, q_z)^T$.
Die Multiplikation $\dot{p} = \dot{r}\,\dot{s}$ *mit* $\dot{r} = r + \boldsymbol{r}$, $\dot{s} = s + \boldsymbol{s}$ *und* $\dot{p} = p + \boldsymbol{p}$ kann damit in der komplexeren Form $p = rs - \boldsymbol{r}\cdot\boldsymbol{s}$ $\boldsymbol{p} = r\,\boldsymbol{s} + s\,\boldsymbol{r} + \boldsymbol{r}\times\boldsymbol{s}$ geschrieben werden. Eine weitere Vereinfachung ergibt sich, wenn die Vektoren $\dot{r}$ und $\dot{s}$ rein imaginär sind, was bedeutet, dass $r = s = 0$
In diesem Fall gilt: $p = -\boldsymbol{r}\cdot\boldsymbol{s}$, $\boldsymbol{p} = \boldsymbol{r}\times\boldsymbol{s}$
Angewendet auf die gemischte Multiplikation:
$\dot{r}' = \dot{q}\,\dot{r}\,\tilde{\dot{q}}$, $\dot{s}' = \dot{q}\,\dot{s}\,\tilde{\dot{q}}$, $\dot{p}' = \dot{q}\,\dot{p}\,\tilde{\dot{q}}$.
$\dot{r}'\dot{s}' = (\dot{q}\,\dot{r}\,\tilde{\dot{q}})(\dot{q}\,\dot{s}\,\tilde{\dot{q}}) = (\dot{q}\,\dot{r})(\tilde{\dot{q}}\,\dot{q})(\dot{s}\,\tilde{q}) = \dot{q}(\dot{r}\,\dot{s})\tilde{\dot{q}}$
Ebenso: $-\boldsymbol{r}'\cdot\boldsymbol{s}' = -\boldsymbol{r}\cdot\boldsymbol{s}$
Genauso ist $\boldsymbol{r}'\times\boldsymbol{s}'$ ein Ergebnis der Anwendung der gemischten Multiplikation auf $\boldsymbol{r}'\times\boldsymbol{s}'$.

Daraus, dass Skalarprodukt und Kreuzprodukt, wie oben gezeigt erhalten bleiben kann man schließen, dass man das gemischte Produkt mit einem Einheitsquaternion zur Darstellung einer Rotation verwenden kann.

Ein weiterer mathematischer Einschub: Verkettung von Rotationen mit Quaternionen

$\dot{r}'' = \dot{p}\,\dot{r}'\,\tilde{\dot{p}} = \dot{p}(\dot{q}\,\dot{r}\,\tilde{\dot{q}})\tilde{\dot{p}}$, da $(\tilde{\dot{q}}\,\tilde{\dot{p}}) = \widetilde{(\dot{p}\,\dot{q})}$ gilt $\dot{r}'' = (\dot{p}\,\dot{q})\,\dot{r}\,\widetilde{(\dot{p}\,\dot{q})}$. Das zeigt, dass die Verkettung von Rotationen einer Multiplikation der entsprechenden Quaternionen entspricht ($\dot{p}\,\dot{q}$ entspricht der Gesamt-Rotation)

Warum benutzt man sie anstatt direkt Matrizen zu bestimmen?

Ein Grund liegt in der Berechnung der Verkettung von Rotationen, bei der man entweder das Produkt zwischen zwei Matrizen oder zwischen zwei Quaternionen berechnen muss. Hierbei ist wohl offensichtlich, dass das Produkt der Quaternionen deutlich weniger Rechenschritte benötigt und deshalb schneller auszuführen ist und auch geringere Rundungsfehler erzeugt. Durch die Rundungsfehler, die sich bei Berechnungen mit endlichen Maschinen nie vermeiden lassen tritt ein weiteres Problem auf; Produkte vieler orthonormaler Matrizen können durch Rundungsfehler unter Umständen nicht mehr orthonormal sein.
Ähnliches gilt für die Produkte vieler Einheitsquaternionen. Diese haben oft nicht mehr exakt die Länge 1. Es ist jedoch deutlich einfacher zu einem durch Rundungsfehler verfälschten Quaternion den nächsten Einheitsquaternion zu finden, als zu einer Matrix die näheste orthonormale Matrix.
Außerdem sind bei der Lösung über einen Einheitsquaternion nur 4 Unbekannte zu lösen, für die nur die Bedingung gelten muss, dass die Summe ihrer Quadrate 1 ist (Einheitsquaternion) im Gegensatz zur Findung einer Matrix, bei der 9 Variable zur Verfügung stehen, die das deutlich aufwändigere Kriterium zu erfüllen haben, dass sich aus ihnen eine orthonormale Matrix ergibt.

Wie bestimmt man mit ihnen Transformationen?

Vorweg ist zu sagen, dass man Einheitsquaternionen ausschließlich für die Findung einer Rotationsmatrix benutzt, da sie eben genau einer solchen entsprechen und nicht etwa einer beliebigen Transformation. Die Skalierung und Translation ist also separat zu berechnen, was relativ leicht fällt, wenn man die Rotation bestimmt hat. In dieser Arbeit soll eine Lösung in geschlossener Form vorgestellt werden, die darauf aufbaut die Rotation über 3 Vektoren je Bild zu berechnen.
Für die hier vorgestellte Lösung benutze ich ausschließlich 3 Punkte pro Bild, zwischen denen jeweils Korrespondenzen festgelegt sind. Diese Methode lässt sich auch auf einen allgemeineren Fall ausdehnen, vereinfacht sich jedoch enorm, wenn man nur 3 Punkte benutzt. Da man mindestens 3 Punkte mit Korrespondenzen benötigt, um eine derartige Registrierung durchzuführen ist es ein Leichtes, die gegebenen Punktkorrespondenzen durch einfaches Weglassen der „überflüssigen“ Korrespondenzpaare auf 3 zu beschränken. Für Punktmengen werde ich an anderer Stelle ein geeignetes Verfahren darstellen.
Als Basis der Registrierung hat man also in den beiden Bildern jeweils 3 Punkte $\boldsymbol{r}_{l,1}, \boldsymbol{r}_{l,2}, \boldsymbol{r}_{l,3} \text{ und } \boldsymbol{r}_{r,1}, \boldsymbol{r}_{r,2}, \boldsymbol{r}_{r,3}$. Für die Bestimmung der Rotation verwendet man ein Dreibein, das man in beiden Koordinatensystemen aufstellen kann. Der Ursprung liegt auf dem ersten Punkt, die x Achse ist in Richtung des zweiten. Die y-Achse ist dazu senkrecht in der Ebene, die durch die 3 Punkte aufgespannt wird. Die z-Achse ist zu den beiden anderen Achsen senkrecht, so dass sie die „reche Hand Regel“ erfüllt, was genau dem Kreuzprodukt der beiden Vektoren in x und y Richtung entspricht. Eine Rotation die diese Dreibeine ineinander überführt, überführt auch die zugrunde liegenden kartesischen Koordinatensysteme ineinander. In Zahlen sind diese Dreibeine:

$\boldsymbol{x}_l = \boldsymbol{r}_{l,2} - \boldsymbol{r}_{l,1}$, $\hat{x}_l = \boldsymbol{x}_l / \|\boldsymbol{x}_l\|$.
$\boldsymbol{y}_l = (\boldsymbol{r}_{l,3} - \boldsymbol{r}_{l,1}) - [(\boldsymbol{r}_{l,3} - \boldsymbol{r}_{l,1}) \cdot \hat{x}_l]\hat{x}_l$, $\hat{y}_l = \boldsymbol{y}_l / \|\boldsymbol{y}_l\|$,
$\hat{z}_l = \hat{x}_l \times \hat{y}_l$.

Entsprechend konstruiert man das Dreibein für das andere Bild. Aus diesen jeweils 3 Vektoren lassen sich zwei Matrizen aufstellen:

$M_l = [\hat{x}_l, \hat{y}_l, \hat{z}_l]$, $M_r = [\hat{x}_r, \hat{y}_r, \hat{z}_r]$

Mit Hilfe dieser Matrizen kann man bereits eine Rotation bestimmen: Der Vektor $\boldsymbol{r}_l$ lässt sich mit $M_l^T \boldsymbol{r}_l$ in das neue ($\hat{x}_l, \hat{y}_l, \hat{z}_l$) Koordinatensystem umwandeln. $\boldsymbol{r}_r$ lässt sich dann berechnen als: $\boldsymbol{r}_r = M_r M_l^T \boldsymbol{r}_l$ Die gesuchte Rotation ist also $R = M_r M_l^T$. Diese Matrix R ist zwar orthonormal, weil es auch $M_l und M_r$ sind, aber solange die verwendeten Punkte nicht absolut perfekt sind ist R von der (beliebig gewählten) Reihenfolge der Punkte abhängig. Außerdem besteht keine Möglichkeit bei diesem Verfahren mehr als 3 Punkte zu verwenden. Deshalb bietet es sich an eine sog. least squares Methode, also die Suche nach der Abbildung mit der kleinsten Summe der Quadrate der Fehler zu verwenden.
Bei dieser Methode kann man die Translation und Skalierung jedoch bereits bestimmen, bevor man die Rotation kennt, was ich im Folgenden zeigen werde.

Translation

Im allgemeinen Fall stehen n Punkte in beiden Bildern zu Verfügung $\boldsymbol{r}_{l,i} und \boldsymbol{r}_{r,i}$ jeweils i = 1 .. n. Gesucht ist dabei eine Transformation $\boldsymbol{r}_r = sR(\boldsymbol{r}_l) + r_0$. hierbei ist $R(\boldsymbol{r}_l)$ nicht zwangsweise in der Form Matrix mal Vektor. Wichtig ist nur, dass durch die Rotation die Längen unbeeinflusst bleiben: $\|R(\boldsymbol{r}_l)\|^2 = \|\boldsymbol{r}_l\|^2$
Da wir davon ausgehen, dass die Eingaben fehlerbehaftet sind, wird immer ein Fehler bei den Ausgaben verbleiben: $e_i = \boldsymbol{r}_{r,i} - sR(\boldsymbol{r}_{l,i}) - r_0$ Wie eingangs bereits erwähnt möchte ich versuchen die Summe der Quadrate der Fehler zu minimieren : $\sum_{i=1}^{n} \|e_i\|^2$.

Wie sich herausgestellt hat ist es hilfreich alle Messungen auf die Schwerpunkte zu beziehen:

$\bar{\boldsymbol{r}}_l = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{r}_{l,i}$, $\bar{\boldsymbol{r}}_r = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{r}_{r,i}$. mit den neuen Koordinaten:

$\boldsymbol{r}'_{l,i} = \boldsymbol{r}_{l,i} - \bar{\boldsymbol{r}}_l$, $\boldsymbol{r}'_{r,i} = \boldsymbol{r}_{r,i} - \bar{\boldsymbol{r}}_r$.

Dabei ist

$\sum_{i=1}^{n} \boldsymbol{r}'_{l,i} = 0$, $\sum_{i=1}^{n} \boldsymbol{r}'_{r,i} = 0$.

Der Fehler kann dann als

$e_i = \boldsymbol{r}'_{r,i} - sR(\boldsymbol{r}'_l) - \boldsymbol{r}'_0$, $mit\ \boldsymbol{r}'_0 = \boldsymbol{r}_0 - \bar{\boldsymbol{r}}_r + sR(\bar{\boldsymbol{r}}_l)$ geschrieben werden.

Damit wird die Summe der Quadrate der Fehler:

$\sum_{i=1}^{n} \|\boldsymbol{r}'_{r,i} - sR(\boldsymbol{r}'_{l,i}) - \boldsymbol{r}'_0\|^2$ =

$\sum_{i=1}^{n} \|\boldsymbol{r}'_{r,i} - sR(\boldsymbol{r}'_{l,i})\|^2 - 2\boldsymbol{r}'_0 \cdot \sum_{i=1}^{n} [\boldsymbol{r}'_{r,i} - sR(\boldsymbol{r}'_{l,i})] + \|\boldsymbol{r}'_0\|^2$.

Die zweite Summe ergibt Null, da $\sum_{i=1}^{n} \boldsymbol{r}'_{r,i} = 0$ wie oben und

$\sum_{i=1}^{n} -sR(\boldsymbol{r}'_{l,i}) = -s\sum_{i=1}^{n} R(\boldsymbol{r}'_{i}) = -sR\sum_{i=1}^{n} \boldsymbol{r}'_{i} = 0$ ebenfalls wie oben und aufgrund der Linearität der Rotation.

Der gesamte Fehler ist also minimal, wenn $\boldsymbol{r}'_0 = 0, oder\ \boldsymbol{r}_0 = \bar{\boldsymbol{r}}_r - sR(\bar{\boldsymbol{r}}_l)$. Damit lässt sich die Translation bestimmen, wenn Rotation und Skalierung bekannt sind. Da nun $\boldsymbol{r}'_0 = 0$ kann der Fehler als $e_i = \boldsymbol{r}'_{r,i} - sR\boldsymbol{r}'_{l,i}$ geschrieben werden. Der zu minimierende Gesamtfehler ist also: $\sum_{i=1}^{n} \|\boldsymbol{r}'_{r,i} - sR(\boldsymbol{r}'_{l,i})\|^2$.

Die Skalierung

Da die Länge bei der Rotation erhalten bleibt gilt: $\|R(\boldsymbol{r}'_{l,i})\|^2 = \|\boldsymbol{r}'_{l,i}\|^2$, womit sich der Gesamtfehler schreiben lässt als $\sum_{i=1}^{n} \|\boldsymbol{r}'_{r,i}\|^2 - s\sum_{i=1}^{n} \boldsymbol{r}'_{r,i} \cdot R(\boldsymbol{r}'_{l,i}) + s^2 \sum_{i=1}^{n} \|\boldsymbol{r}'_{l,i}\|^2$, was man auch als $S_r - 2sD + s^2 S_l$ schreiben kann. Dabei sind $S_l\ und\ S_r$ jeweils die Summe der Quadrate der Eingabepunkte relativ zu ihrem Schwerpunkt und D ist die Summe der Skalarprodukte der Punkte im rechten System und den korrespondierenden, rotierten Punkten im Linken.

Anders geschrieben ergibt die obige Formel: $(s\sqrt{S_l} - D/\sqrt{S_l})^2 + (S_r S_l - D^2)/S_l$.

Dieser Term ist, wie nun leicht zu sehen, minimal, wenn

$s = D/S_l = \sum_{i=1}^{n} \boldsymbol{r}'_{r,i} \cdot R(\boldsymbol{r}'_{l,i}) / \sum_{i=1}^{n} \|\boldsymbol{r}'_{l,i}\|^2$ ist.

Ein Trick, um wie ich nun zeigen will die Skalierung ohne Kenntnis der Rotation zu finden ist die Ausnutzung der Symmetrie der Skalierung. Anstatt $\boldsymbol{r}_r = sR(\boldsymbol{r}_l) + \boldsymbol{r}_0$ suchen wir nun also $\boldsymbol{r}_l = \bar{s}\,\bar{R}(\boldsymbol{r}_r) + \bar{\boldsymbol{r}}_0$ und hoffen dabei genau die inverse Transformation zu finden

$\bar{s} = 1/s$, $\bar{\boldsymbol{r}}_0 = \frac{1}{s} R^{-1}(\boldsymbol{r}_0)$ und $\bar{R} = R^{-1}$.

Aufgrund der Fehler in den Eingabepunkten wird man dieses Ergebnis jedoch leider nicht erhalten, weshalb es von Vorteil ist, ein symmetrischeres Maß für den Fehler zu benutzen:

$e_i = \frac{1}{\sqrt{s}} \boldsymbol{r}'_{r,i} - \sqrt{s}\, R(\boldsymbol{r}'_{l,i})$ .Damit wird der Gesamtfehler zu

$\frac{1}{s}\sum_{i=1}^{n} \|\boldsymbol{r}'_{r,i}\|^2 - 2\sum_{i=1}^{n} \boldsymbol{r}'_{r,i} \cdot R(\boldsymbol{r}'_{l,i}) + s\sum_{i=1}^{n} \|\boldsymbol{r}'_{l,i}\|^2 = \frac{1}{s} S_r - 2D + sS_l = (\sqrt{s}\, S_l - \frac{1}{s} S_r)^2 + 2(S_l S_r - D)$

was minimal wird, wenn $s = S_r/S_l = (\sum_{i=1}^{n} \|\boldsymbol{r}'_{r,i}\|^2 / \sum_{i=1}^{n} \|\boldsymbol{r}'_{l,i}\|^2)^{1/2}$.

Nach diesem Ansatz muss man die Rotation so wählen, dass $\sum_{i=1}^{n} \boldsymbol{r}'_{r,i} \cdot R(\boldsymbol{r}'_{l,i})$ maximal wird.

Rotation

Für die Funktion der Rotation suchen wir nun einen entsprechenden Einheitsquaternion, so dass $\sum_{i=1}^{n}(\dot{q}\dot{r}'_{l,i}\tilde{\dot{q}})\cdot\dot{r}'_{r,i}$ maximal wird. Mit einem der obigen Ergebnisse kann man diese Summe auch als $\sum_{i=1}^{n}(\dot{q}\dot{r}'_{l,i})\cdot(\dot{r}'_{r,i}\dot{q})$ schreiben.

Angenommen $r'_{l,i}=\begin{pmatrix}x'_{l,i}\\y'_{l,i}\\z'_{l,i}\end{pmatrix}$ und $r'_{r,i}=\begin{pmatrix}x'_{r,i}\\y'_{r,i}\\z'_{r,i}\end{pmatrix}$, dann

$$\dot{q}\dot{r}'_{l,i}=\begin{bmatrix}0 & -x'_{l,i} & -y'_{l,i} & -z'_{l,i}\\ x'_{l,i} & 0 & z'_{l,i} & -y'_{l,i}\\ y'_{l,i} & -z'_{l,i} & 0 & x'_{l,i}\\ z'_{l,i} & y'_{l,i} & -x'_{l,i} & 0\end{bmatrix}\dot{q}=\mathbb{\bar{R}}_{l,i}\dot{q} \quad \text{und}$$

$$\dot{r}'_{r,i}\dot{q}=\begin{bmatrix}0 & -x'_{r,i} & -y'_{r,i} & -z'_{r,i}\\ x'_{r,i} & 0 & -z'_{r,i} & y'_{r,i}\\ y'_{r,i} & z'_{r,i} & 0 & -x'_{r,i}\\ z'_{r,i} & -y'_{r,i} & x'_{r,i} & 0\end{bmatrix}\dot{q}=\mathbb{R}_{r,i}\dot{q}$$

Die zu maximierende Summe kann damit geschrieben werden als

$$\sum_{i=1}^{n}(\mathbb{\bar{R}}_{l,i}\dot{q})\cdot(\mathbb{R}_{r,i}\dot{q})=\sum_{i=1}^{n}\dot{q}^T\mathbb{\bar{R}}_{l,i}^T\mathbb{R}_{r,i}\dot{q}=\dot{q}^T(\sum_{i=1}^{n}\mathbb{\bar{R}}_{l,i}^T\mathbb{R}_{r,i})\dot{q}=\dot{q}^T(\sum_{i=1}^{n}N_i)\dot{q}=\dot{q}^T N\dot{q}$$

$$mit \quad N_i=\mathbb{\bar{R}}_{l,i}^T\mathbb{R}_{r,i} \quad und \quad N=\sum_{i=1}^{n}N_i$$

Wie leicht nachzuweisen ist, ist jede der N_i Matrizen symmetrisch, also muss auch N symmetrisch sein.

Zur Berechnung von N führe ich nun eine Hilfsmatrix $M=\sum_{i=1}^{n}\boldsymbol{r}'_{l,i}\boldsymbol{r}'^{T}_{r,i}=\begin{bmatrix}S_{xx} & S_{xy} & S_{xz}\\ S_{yx} & S_{yy} & S_{yz}\\ S_{zx} & S_{zy} & S_{zz}\end{bmatrix}$

ein, wobei $S_{xx}=\sum_{i=1}^{n}x'_{l,i}x'_{r,i}$, $S_{xy}=\sum_{i=1}^{n}x'_{l,i}y'_{r,i}$ usw.

Diese Matrix M enthält all für die Berechnung nötigen Daten:

$$N=\begin{bmatrix}(S_{xx}+S_{yy}+S_{zz}) & S_{yz}-S_{zy} & S_{zx}-S_{xz} & S_{xy}-S_{yx}\\ S_{yz}-S_{zy} & (S_{xx}-S_{yy}-S_{zz}) & S_{xy}+S_{yx} & S_{zx}+S_{xz}\\ S_{zx}-S_{xz} & S_{xy}+S_{yx} & (-S_{xx}+S_{yy}-S_{zz}) & S_{yz}+S_{zy}\\ S_{xy}-S_{yx} & S_{zx}+S_{xz} & S_{yz}+S_{zy} & (-S_{xx}-S_{yy}+S_{zz})\end{bmatrix}$$

Wie sich zeigen lässt, ist der Einheitsquaternion, der $\dot{q}^T N\dot{q}$ maximiert der Eigenvektor zum größten (nicht betragsmäßig) Eigenwert von N. Angenommen λ_m sei der größte Eigenwert, dann kann man den zugehörigen Eigenvektor $\dot{e}_m$ durch lösen der homogenen Gleichung $[N-\lambda_m I]\dot{e}_m=0$ finden.

Zur Suche der Eigenwerte:

schreibt man die Matrix N als $N=\begin{bmatrix} a & e & h & j \\ e & b & f & i \\ h & f & c & g \\ j & i & g & d \end{bmatrix}$, dann kann man $det(N-\lambda I)=0$ schreiben als

$\lambda^4+c_3\lambda^3+c_2\lambda^2+c_1\lambda+c_0=0$

mit

$c_3=a+b+c+d=0,$

$c_2=(ac-h^2)+(bc-f^2)+(ad-j^2)+(bd-i^2)+(cd+g^2)+(ab-e^2),$

$c_1=-b(cd-g^2)+f(df-gi)-i(fg-ci)-a(cd-g^2)+h(dh-gj)-j(gh-cj)$
$-a(bd-i^2)+e(de-ij)-j(ei-bj)-a(bc-f^2)+e(ce-fh)-h(ef-bh),$

$c_0=(ab-e^2)(cd-g^2)+(eh-af)(fd-gi)+(ai-ej)(fg-ci)$
$+(ef-bh)(hd-gj)+(bj-ei)(hg-cj)+(hi-fi)^2$

Eingesetzt:

$c_2=-2(S_{xx}^2+S_{xv}^2+S_{xz}^2+S_{vx}^2+S_{vv}^2+S_{vz}^2+S_{zx}^2+S_{zv}^2+S_{zz}^2)$

c_2 ist also immer negativ.

$c_1=8(S_{xx}S_{yz}S_{zy}+S_{yy}S_{zx}S_{xz}+S_{zz}S_{xy}S_{yx})-8(S_{xx}S_{yy}S_{zz}+S_{yz}S_{zx}S_{xv}+S_{zv}S_{vx}S_{xz})=-8\,det(M)$

$c_0=det(N)$

c_1 kann sowohl positiv, als auch negativ sein, sind jedoch alle Punkte in einer Ebene, was bei drei Punkten immer der Fall ist, so ist $c_1=0$

Soweit zur allgemeinen Berechnung. Für den Fall, dass genau drei Punkte pro Bild gegeben sind, möchte ich hier eine weitere, speziell darauf abgestimmte Abwandlung dieser Methode vorstellen.
Die Vereinfachung ist dabei die Reduzierung des least-squares Problems (finden der kleinsten quadrierten Fehler) auf eine Ebene, anstelle des 3D Raums. Da die beiden Punktmengen jeweils koplanar sind (3 Punkte liegen immer in einer Ebene) kann man die gesuchte Rotation zerlegen in eine Rotation die die beiden Ebenen ineinander überführt und eine Rotation in der Ebene, die dann durch Fehlerminimierung gefunden werden muss. Die erste Rotation ist einfach eine Drehung um die Schnittgerade der beiden Ebenen, mit dem Winkel zwischen den Normalen der beiden Ebenen. Diese sind:

$\boldsymbol{n}_l=\boldsymbol{r}'_{l2}\times\boldsymbol{r}'_{l1}$, $\boldsymbol{n}_r=\boldsymbol{r}'_{r2}\times\boldsymbol{r}'_{r1}$ und $\bar{n}_l$, $\bar{n}_r$ als normalisierte Normalen (Länge 1)

$\boldsymbol{a}=\boldsymbol{n}_l\times\boldsymbol{n}_r$ ist dann die Richtung der Schnittgeraden, die in beiden Ebenen und damit senkrecht zu den beiden Normalen sein muss. Der gesuchte Rotationswinkel ϕ ist der Winkel zwischen den beiden Normalen. Für ihn gilt:

$\cos\phi=\bar{n}_l\cdot\bar{n}_r,\quad \sin\phi=\|\bar{n}_l\times\bar{n}_r\|,\quad wobei\ \ 0\le\phi\le\pi$

Der Einheitsquaternion für die Rotation ist dann

$\dot{q}_a=\cos\frac{\phi}{2}+\sin\frac{\phi}{2}(ia_x+ja_y+ka_z)$ die entsprechende Matrix ist R_a . Die rotierten Punkte von $\boldsymbol{r}'_{l,i}$ sind $\boldsymbol{r}''_{l,i}$.

Nun müssen wir noch die Rotation in der Ebene finden, die die Quadrate der Abstände der korrespondierenden Punkte minimiert:

$\sum_{i=1}^{n}\|\boldsymbol{r}'_{r,i}-\boldsymbol{r}''_{l,i}\|^2$

Es gilt:

$r'_{r,i}r''_{l,i}\cos\alpha_i=\boldsymbol{r}'_{r,i}\cdot\boldsymbol{r}''_{l,i},\quad unc$

$r'_{r,i}r''_{l,i}\sin\alpha_i=(\boldsymbol{r}'_{r,i}\times\boldsymbol{r}''_{l,i})\cdot\bar{n}_r\quad mit$

$r'_{r,i}=\|\boldsymbol{r}'_{r,i}\|,\quad r''_{l,i}=\|\boldsymbol{r}''_{l,i}\|=\|\boldsymbol{r}'_{l,i}\|$

und α_i als Winkel zwischen den Korrespondenzpunkten.
Da $\boldsymbol{r}'_{r,i} \times \boldsymbol{r}''_{l,i}$ parallel zu $\bar{n}_r$ ist das Skalarprodukt aus den beiden gleich $\|\boldsymbol{r}'_{r,i} \times \boldsymbol{r}''_{l,i}\|$. Wobei das Vorzeichen von der Ausrichtung von $\boldsymbol{r}'_{r,i}$ und $\boldsymbol{r}''_{l,i}$ in der Ebene abhängt.
Über die Kosinus-Regel für Dreiecke, erhalten wir $(r'_{r,i})^2 + (r''_{l,i})^2 - 2 r'_{r,i} r''_{l,i} \cos\alpha_i$ als quadratischen Abstand zwischen den korrespondierenden Punkten.
Wenn man eine Drehung um den Winkel θ durchführt reduzieren sich die Winkel α_i um θ . Man muss also $\sum_{i=1}^{n} f'_{r,i} r''_{l,i} \cos(\alpha_i - \theta) = C\cos\theta + S\sin\theta$ maximieren, wobei $C = \sum_{i=1}^{n} r'_{r,i} r''_{l,i} \cos\alpha_i = \sum_{i=1}^{n} (\boldsymbol{r}'_{r,i} \cdot \boldsymbol{r}''_{l,i})$ und $S = \sum_{i=1}^{n} r'_{r,i} r''_{l,i} \sin\alpha_i = \left(\sum_{i=1}^{n} \boldsymbol{r}'_{r,i} \times \boldsymbol{r}''_{l,i} \right) \cdot \bar{n}_r$.
Die Extrema dieses Ausdrucks sind bei $C\sin\theta = S\cos\theta$, oder $\sin\theta = \pm\frac{S}{\sqrt{S^2+C^2}}, \cos\theta = \pm\frac{C}{\sqrt{S^2+C^2}}$. Die zweite Rotation ist also um die Achse $\bar{n}_r$ mit dem Winkel θ und kann durch den Einheitsquaternion $\dot{q}_p = \cos\frac{\theta}{2} + \sin\frac{\theta}{2}(\dot{\imath}_x + jn_y + kn_z)$ dargestellt werden. Die gesamte Rotation ist also $\dot{q} = \dot{q}_p \dot{q}_a$
Mit diesem Verfahren ist es möglich die Transformation (Registrierung) zwischen zwei Bildern aufbauend auf die Bestimmung von 3 Korrespondenzpaaren in einer geschlossenen Form zu berechnen. Die einzig aufwändigeren Rechenschritte sind die Trigonometrischen Funktionen, die man allerdings nicht explizit berechnen muss, da man lediglich Kosinus und Sinus von halben Winkeln berechnen muss, wenn sie vom ganzen Winkel bereits bekannt sind. Diese Berechnung kann man im Bereich von $-\pi \le \theta \le \pi$ durch die Formeln $\cos\frac{\theta}{2} = [(1+\cos\theta)/2]^{1/2}, und \quad \sin\frac{\theta}{2} = sind\,\theta/[2(1+\cos\theta)]^{1/2}$ ohne Benutzung trigonometrischer Funktionen durchführen.

ICP Iterative Closest Point Algorithmus

Das Problem des soeben vorgestellten Algorithmus, auch wenn er für mehrere Punkte verwendbar ist, bleibt, dass man immer nur Korrespondenzpaare von Punkten verwenden kann. Gewinnt man die Punkte, die man verwenden möchte jedoch durch einen automatischen Vorgang, der z.B. Punkte auf einer detektierten Kante wählt, so erhält man in beiden Bildern eine Menge von Punkten. Man weiß dann zwar, dass sich die beiden Mengen in den jeweiligen Bildern entsprechen, weil sie dieselbe Kante repräsentieren, man kann aber keine Korrespondenzen zwischen den einzelnen Punkten herstellen. In derartigen Anwendungsfällen verwendet man den sog. ICP (iterative closest point) Algorithmus. Dieser bietet zwar, wie der Name bereits aussagt eine iterative und keine geschlossene Form der Lösung an, kann dafür auf großen Punktmengen mit unbekannten Korrespondenzen eingesetzt werden.
Anmerkung: Die Skalierung wird in diesem Algorithmus vernachlässigt, da er für eine Anwendung entstanden ist, in der es wichtig ist, die Größe von Objekten vergleichen zu können. Als Beispiel könnte man das Wachstum eines Tumors heranziehen.

Als Eingabe verwendet man zwei Punktmengen U und X
$U = \{u_i, i = 0,1,2,\ldots,n\}, \quad X = \{x_i, i = 0,1,2,\ldots,k$.
Dabei ist zu beachten, dass U und X nicht notwendigerweise die gleiche Anzahl an Elementen haben. O.B.d.A. werde ich annehmen, dass $k \ge n$. Es soll eine

Transformation berechnet werden, die U auf X abbildet. In jedem der iterativen Schritte soll nun U etwas näher in Richtung X gedreht und verschoben werden. q_R bezeichnet nun den Quaternion, der für die Rotation steht, q_T den 3D Vektor, der die Translation ersetzt und $q=(q_R, q_T)$ einen 7D Vektor, der einfach sämtliche Informationen, die für die Transformation nötig sind beinhaltet. Die Iteration wird nun mit den Werten $U_0=U$ und $q_0=(1,0,0,0,0,0,0)$ initialisiert. Die nachfolgenden Schritte werden so lange wiederholt, bis eine Konvergenz erreicht wurde:

1. Berechne die Menge $Y \subseteq X$ der U am nähesten Punkte. Zum Beispiel, für jeden Punkt $u \in U$ berechnet man den zu u nähesten Punkt in X. Y hat genauso viele Elemente, wie U.
2. Bestimme $q=(q_R, q_T)$ so, dass U_0 bestmöglich auf Y abgebildet wird. Da hier Korrespondenzpaare bekannt sind, kann man an dieser Stelle den oben vorgestellten Algorithmus für die Berechnung in geschlossener Form verwenden, wobei darauf zu achten ist, die Skalierung nicht zu berücksichtigen.
3. Wende die Translation und Rotation in q auf U_0 an.
4. Berechne den mittleren quadratischen Fehler
5. Ist der Fehler kleiner, als eine zuvor angegebene Schranke wird die Berechnung beendet, ansonsten wird sie ab Schritt 1 wiederholt.

Dieser Algorithmus konvergiert immer, leider nicht immer zum gewünschten Ergebnis. Ein möglicher Ausweg ist es verschiedene Start werte für den Algorithmus zu wählen und nach einigen Iterationen nur noch den bisher besten Auswertungspfad zu verfolgen. Es gibt noch weitere Optimierungstechniken, auf die ich aber an dieser Stelle nicht weiter eingehen möchte.

Andere Verfahren

Es gibt weiter Verfahren, wie Hauptachsen-Ausrichtung (principal axis alignment), oder Verfahren die auf einer Segmentierung, welche in einem eigenen Vortrag im Rahmen dieses Seminars erörtert wird, aufbauen. Auf diese Verfahren möchte ich an dieser Stelle nicht weiter eingehen, da sie extrem von den zu verarbeitenden Bilddaten und der Vorverarbeitung abhängen. Abgesehen davon soll diese Arbeit nur einen kleinen Überblick über die Möglichkeiten und Techniken der Bildfusion geben und nicht sämtliche Lösungsansätze im Detail erklären. Dies wäre ohnehin ein sinnloses Unterfangen, da allein eine meiner Quallen für diese Arbeit (eine Zusammenfassung der wichtigsten Arbeiten der letzten Jahre) mehr als 300 Quellen im Literaturverzeichnis führt, von denen viele eigene Ansätze für die Lösung spezieller Aufgaben beinhalten.

c) Mutual Information

Allgemeines über das Verfahren mit MI

Über die MI bestimmt man zumindest im hier vorgestellten Verfahren die statistische Abhängigkeit zwischen den Intensitäten der korrespondierenden Voxel in zwei Bildern. Es ist anzunehmen, dass diese Abhängigkeit maximal wird, wenn die beiden Bilder optimal registriert sind. Der große Vorteil dieser Methode ist, dass sie vollständig automatisch und ohne jegliche Vorverarbeitung der Bilddaten durchgeführt werden kann. Außerdem liefert sie in den allermeisten Fällen eine Genauigkeit der Registrierung, die auf Sub-Pixel Ebene liegt. Die Methode die ich hier vorstelle unterscheidet sich von den davor verwendeten Voxel-basierten Verfahren vor allem dadurch, dass nicht angenommen wird, der Zusammenhang zwischen den Intensitäten in den verschiedenen Bildern sei linear. Diese Erweiterung hebt die Einschränkung auf Bilder, die mit dem gleichen Verfahren erstellt wurden auf. Im konkreten Beispiel: Es besteht zwar ein Zusammenhang zwischen dargestellten Organen und ihrer Funktion, dieser ist im Normalfall aber nicht linear dargestellt. Im Extremfall werden Organe, die keine der untersuchten Funktionen aufweisen nur sehr schwach bis garnicht dargestellt, wohingegen die gesuchten Regionen mit sehr hohen Kontrastverhältnissen dargestellt werden. Für eine Registrierung derartiger Bilddaten ist also die Einbeziehung nichtlinearer Abhängigkeiten notwendig und sinnvoll. Im Grundprinzip gibt die MI an, wie viel Information über eine Variable (einen Voxelgrauwert) in einer korrespondierenden anderen enthalten ist.

Grundlagen

Eine kurze Erinnerung an einige statistische und stochastische Grundlagen und Einführung der MI:
Wahrscheinlichkeit für a als Wert der Variable A ist $p_A(a)$ ebenso für b und B $p_B(b)$
Die Variablen sind statistisch unabhängig, wenn $p_{AB}(a,b)=p_A(a)\cdot p_B(b)$ und maximal abhängig, wenn eine von ihnen durch eine 1:1 Abbildung aus der anderen gewonnen wurde: $T: p_A(a)=p_B(T(a))=p_{AB}(a,T(a))$. MI, $I(A,B)$ misst den Grad der Abhängigkeit zwischen A und B über den Unterschied zwischen der zusammengesetzten Wahrscheinlichkeit $p_{AB}(a,b)$ und der Wahrscheinlichkeit im Fall der totalen Unabhängigkeit $p_A(a)\cdot p_B(b)$:

(1) $$I(A,B)=\sum_{a,b} p_{AB}(a,b)\log\frac{p_{AB}(a,b)}{p_A(a)\cdot p_B(b)}$$

In Abhängigkeit von den Häufigkeiten $H(A)$ und $H(B)$ sowie den bedingten Häufigkeiten $H(A|B)$ und $H(B|A)$ und der gemeinsamen Häufigkeit $H(A,B)$:

(2) $I(A,B)=H(A)+H(B)-H(A,B)$
(3) $=H(A)-H(A|B)$
(4) $=H(B)-H(B|A)$
(5) $$H(A)=-\sum_a p_A(a)\log p_A(a)$$
(6) $$H(A,B)=-\sum_{a,b} p_{AB}(a,b)\log p_{AB}(a,b)$$

(7) $H(A|B) = -\sum_{a,b} p_{AB}(a,b) \log p_{(A|B)}(a|b)$

Einige wichtige Eigenschaften der MI:

$$
\begin{aligned}
&\textit{nicht}-\textit{negativ}: && I(A,B) \geq 0 \\
&\textit{Unabhängigkeit}: && I(A,B) = 0 \Leftrightarrow p_{AB}(a,b) = p_A(a) \cdot p_B(b) \\
&\textit{Symmetrie} && I(A,B) = I(B,A) \\
& && I(A,A) = H(A) \\
&\textit{Begrenztheit}: && I(A,B) \leq min(H(A),H(B)) \\
& && \leq (H(A)+H(B))/2 \\
& && \leq max(H(A),H(B)) \\
& && \leq H(A,B) \\
& && \leq H(A)+H(B) \\
&\textit{Datenverarbeitung}: && I(A,B) \geq I(A,T(B))
\end{aligned}
$$

Angenommen a und b sind die Grauwerte korrespondierender Pixel in den beiden zu registrierenden Bildern, dann stehen die Grauwerte über die geometrische Transformation T_α, definiert durch α in Relation. Das MI Kriterium legt fest, dass die beiden Bilder durch die Transformation $T_{\mathring{\alpha}}$, für die $I(A,B)$ maximal ist registriert sind. Am Beispiel eines CT und eines MR Bildes eines Schädels würde das bedeuten, dass im Registrierungsfall die im CT dunklen Regionen (innerhalb des Schädelknochens) auf helle im MR (Gehirnmasse) abgebildet werden. Dadurch erhält man zum Beispiel von den hellen Bereichen im CT relativ viel Information über die korrespondierenden Pixel im MR (diese sollten eine geringe Intensität aufweisen). Im nicht registrierten Fall geht diese Information völlig verloren.

Wenn entweder $p_A(a)$ oder $p_B(b)$ von α unabhängig sind, was immer der Fall ist, wenn eines der beiden Bilder vollständig im anderen enthalten ist, dann reduziert sich das MI-Kriterium auf die Minimierung der bedingten Entropie $H(A|B)$ oder $H(B|A)$. Wenn sich die beiden Bilder jedoch nur teilweise überlagern, was zumindest während der Registrierung sehr oft passiert, dann hängen sowohl $p_A(a)$ und $p_B(b)$ als auch $H(A)$ und $H(B)$ im allgemeinen von α ab. Damit sich $I(A,B)$ für die Registrierung gutmütig verhält, sollte es sich als Funktion von $|\alpha - \mathring{\alpha}|$ nur langsam verändern.

Algorithmus zur Bestimmung einer Transformation mit MI

Jedes der Bilder erhält nun ein Koordinatensystem mit dem Ursprung in einer Ecke des Bildes, der x-Achse entlang der Reihen, der y-Achse entlang der Spalten und der z-Achse entlang der Ebenen. Eines der beiden Bilder wird das „Schwebende“ (floating image) **F**, aus dem einzelne Punkte $s \in S$ ausgewählt und in das andere, das Referenzbild **R** transformiert werden. S kann sowohl eine Ober- als auch eine Untermenge der Gitterpunkte in **F** sein, wobei Supersampling die Genauigkeit der Registrierung und Subsampling die Geschwindigkeit erhöht. Für den Parameter α werden jeweils nur die Punkte $s \in S_\alpha \subset S$ betrachtet, für die $T_\alpha(s)$ in **R** liegt. Hierbei wird für die Transformation, wie in medizinischen Anwendungen üblich angenommen, dass es sich um eine rigide Transformation handelt. Das Verfahren ist jedoch auch für beliebige andere Transformationen erweiterbar. In diesem Algorithmus wird des weiteren die Skalierung vernachlässigt, da die Daten nicht in Pixel sondern in Millimeter verarbeitet werden. Der Parameter α ist also ein 6-Komponenten-Vektor, der die Werte ϕ_x, ϕ_y, ϕ_z als Winkel für die Rotation in Grad und die Translationsparameter t_x, t_y, t_z gemessen in Millimeter beinhaltet. Die Transformation der Bildkoordinaten P_F nach P_R vom Bild **F** zum Bild

R wird festgelegt durch:

$$V_R \cdot (P_R - C_R) = R_x(\phi_x) \cdot R_y(\phi_y) \cdot R_z(\phi_z) \cdot V_F \cdot (P_F - C_F) + t(t_x, t_y, t_z)$$

Mit V_F und V_R als 3x3 Diagonalmatrizen mit den Voxel-Größen der Bilder **F** und **R**, C_F und C_R als Bildkoordinaten der Bildmitten, $R = R_x \cdot R_y \cdot R_z$ als 3x3 Rotationsmatrix mit den Matrizen R_x, R_y, R_z als Rotationen um die jeweilige Achse und t als Translationsvektor. f(s) sei die Intensität **F** an der Stelle s und $r(T_\alpha(s))$ die Intensität an der transformierten Stelle in **R.** Das Gesamthistogramm $h_\alpha(f, r)$ des überlagerten Volumens bei α wird durch eine Klasseneinteilung der Intensitätspaare $(f(s), r(T_\alpha(s)))$ für alle $s \in S_\alpha$ berechnet. Um dies effizient vornehmen zu können werden die Intensitäten der beiden Bilder zuerst linear in den Bereich $[0, n_F - 1]$ und $[0, n_R - 1]$ skaliert, $n_F \times n_R$ entspricht also allen Klassen im Gesamthistogramm. Meistens wird $n_F = n_R = 256$ verwendet.

Leider stimmt $T_\alpha(s)$ im Allgemeinen nicht mit einem Gitterpunkt von **R** überein und man muss den Intensitätswert $r(T_\alpha(s))$ an dieser Stelle interpolieren. Die nearest neighbour – Interpolation (NN) ist hierbei meist unzulänglich, da man mit ihr keine sub-Pixel Genauigkeit erreichen kann, weil sie auf Translationen von bis zu einem Pixel nicht reagiert. Andere Verfahren, wie trilineare Interpolation (TRI) erzeugen oft neue Intensitätswerte, die im Bild bisher nicht vorhanden waren und verändern damit das Histogramm und erzeugen unvorhersehbare Änderungen in der Verteilung $p_{R,\alpha}(r)$ von **R** bei kleinen Änderungen von α . Deshalb empfiehlt es sich die sog. trilineare partielle Volumenverteilungs-Interpolation zu benutzen, um das Histogramm für jedes Voxelpaar $(s, T_\alpha(s))$ zu aktualisieren. Anstatt in R neue Intensitäten zu erzeugen wird der Beitrag der Bildintensität $f(s)$ des Punktes s in **F** auf die Intensitätswerte aller 8 NN von $T_\alpha(s)$ auf dem Gitter von **R** verteilt. Dabei werden die selben Gewichtungen verwendet, wie bei einer TRI Interpolation. Dadurch verändert sich das Histogramm langsam und gleichmäßig, wenn α verändert wird. Näherungen für die einzelnen und die zusammengesetzten Verteilungen $p_{F,\alpha}(f), p_{R,\alpha}(r)$ und $p_{FR,\alpha}(f, r)$ werden durch Normalisieren von $h_\alpha(f, r)$ gewonnen:

(9) $$p_{FR,\alpha}(f, r) = \frac{h_\alpha(f, r)}{\sum_{f,r} h_\alpha(f, r)}$$

(10) $$p_{F,\alpha}(f) = \sum_r p_{FR,\alpha}(f, r)$$

(11) $$p_{R,\alpha}(r) = \sum_f p_{FR,\alpha}(f, r)$$

Das MI Kriterium $I(\alpha)$ wird dann ausgewertet durch:

(12) $$I(\alpha) = \sum_{f,r} p_{FR,\alpha}(f, r) \log_2 \frac{p_{FR,\alpha}(f, r)}{p_{F,\alpha}(f)\, p_{R,\alpha}(r)}$$

und der optimale Parameter $\mathring{\alpha}$ wird durch $\mathring{\alpha} = \arg\max_\alpha I(\alpha)$ gefunden.

Die Bilder werden zu Anfang der Suche so positioniert, dass ihre Bildmitten übereinstimmen und ihre Achsen gleich ausgerichtet sind. Für die Maximierung von $I(\alpha)$ wird „Powells Methode zur mehrdimensionalen Richtungsfestlegung“ (Powell's method for multidimensional direction set), aufbauend auf Brent's eindimensionaler Optimierung verwendet. Die Richtungsmatrix wird dabei initialisiert mit den Einheitsvektoren der jeweiligen Parameterrichtung. Die Festlegung der Reihenfolge, in der die Parameter optimiert werden sollen ist von großer Bedeutung, da sie die Stabilität der Optimierung stark beeinflussen kann. Bei der Registrierung des Bildes von einem Gehirn zum Beispiel werden die horizontale Translation und die Rotation um die vertikale Achse durch die Form des Gehirns deutlich besser eingegrenzt als die Rotation um die

horizontale Achse. Deshalb erleichtert es die Optimierung der Parameter außerhalb der Eben (ϕ_x, ϕ_y, t_z) , wenn man das Bild zuerst in der Ebene ausrichtet, indem man die Parameter $(t_x, t_y, \phi z)$ optimiert.
der Powell Algorithmus kann jedoch während seiner Ausführung die Reihenfolge der Optimierungen verändern.

Bewertung des MI-Algorithmus

In einer Vielzahl von Versuchen wurde nachgewiesen, dass der hier vorgestellte Algorithmus bei den allermeisten Anwendungen eine Sub-Pixel Genauigkeit bei der Registrierung liefert. Damit qualifiziert sich dieser Algorithmus für die Anwendung im praktischen Einsatz, da er zusätzlich ohne Vorverarbeitung und ohne Einwirkung des Benutzers abläuft. Die einzige Anwendungsspezifische Komponente hierbei ist die Reihenfolge der Optimierung der verschiedenen Parameter, die man für den jeweiligen Fall auswählen muss, wenn man die Stabilität entsprechend erhöhen möchte.

Literaturverzeichnis

Powell's Method for multidimensional direction set Kap10.5:
http://lib-www..lanl.glov/numerical/bookcpdf.html
Das entsprechende Kapitel befindet sich im Anhang

CT und PET, sowie die registrierten Bilder stammen von der Internetseite der Charite – Klinik für Strahlenheilkunde:
http://www.charite.de/rv/str/forschung/ippo/fusion/index.php

A survey of medical image registration
J.B. Antoine Maintz and Max A. Viergever
Medical Image Analysis (1998) volume 2, number 1, pp1-36
Oxford University Press

Chapter 11 aus
Volumetric Image Analysis
Lehmann

Closed-form solution of absolute orientation using unit quaternions
Berthold K. P. Horn
Department of Electrical Engineering, University of Hawaii at Manoa

Multimodality Image Registration by Maximization of Mutual Information
Frederik Maes, André Collignon, Dirk Vandermuelen, Guy Marchal und Paul Suetens, Member, IEEE
IEEE TRANSACTIONS ON MEDICAL IMAGING vol. 16 no. 2 april 1997