

# Das 3d-Puzzle-Problem

## Effiziente Methoden zum paarweisen Zusammensetzen von dreidimensionalen Fragmenten

Simon Winkelbach

Institut für Robotik und Prozessinformatik  
Technische Universität Braunschweig  
Mühlenpfordtstr. 23, 38106 Braunschweig  
s.winkelbach@tu-bs.de

**Abstract:** Das automatische Zusammenfügen von dreidimensionalen Objekten aus Einzelteilen (3d-Puzzle-Problem) ist in vielen wichtigen Forschungs- und Anwendungsbereichen von hoher Relevanz. In diesen Bereichen können insbesondere durch Methoden der Informatik und der 3d-Bildverarbeitung substantielle Fortschritte erzielt werden. Diese Arbeit gibt einen Überblick über die gesamte Prozesskette von der Datenakquisition, über die allgemeine Registrierung von Oberflächen, bis hin zu speziellen Anforderungen beim Zusammensetzen von Fragmenten in unterschiedlichen Anwendungsfällen. Insbesondere werden neuartige Ansätze vorgestellt, mit denen Fragmentoberflächen auf äußerst effiziente Weise und ohne Wissen über eine Initiallösung automatisch zusammengefügt werden können. Hierbei wird eine hohe Robustheit gegenüber Messungenauigkeiten, Fragmenteschädigungen und Materialverschleiß erreicht. Darüber hinaus gelingt es, durch Ausnutzung von Symmetrieebenen und Achsen gebrochene Oberschenkel- und Beckenknochen virtuell zusammenzufügen und somit einen wichtigen Baustein für die computerassistierte Frakturbehandlung in der Chirurgie zu schaffen.

## 1 Einleitung

Ein Mensch besitzt im Allgemeinen die kognitiven und motorischen Fähigkeiten, ein zerbrochenes Objekt wieder zu einem Ganzen zusammenzusetzen, in vielen Fällen sogar dann, wenn er die Form des ursprünglichen Objektes nicht kennt. Hierbei müssen die Bruchstücke (Fragmente) in die ursprüngliche (also vor dem Zerschneiden vorherrschende) relative räumliche Lage gebracht werden. Die Lösung dieses '3d-Puzzle-Problems' hat maßgebliche Relevanz in ganz unterschiedlichen Anwendungsgebieten (siehe Abbildung 1). Beispielsweise müssen in der Archäologie zerbrochene historische Artefakte rekonstruiert, in der Chirurgie höhere Genauigkeiten beim Richten von Knochenfrakturen erreicht, in der Biochemie Protein-Bindungen untersucht, in der Computergraphik Tiefendaten fusioniert und in der Robotik Bauteile mit unbekannter räumlicher Lage erkannt und gegriffen werden. In vielen Bereichen ist ein manuelles Zusammensetzen der Fragmente prinzipiell nicht möglich, zu ungenau oder mit zu hohen Kosten und Zeitaufwand verbunden. Im Fall von gebrochenen Knochen geht der behandelnde Chirurg ein nicht

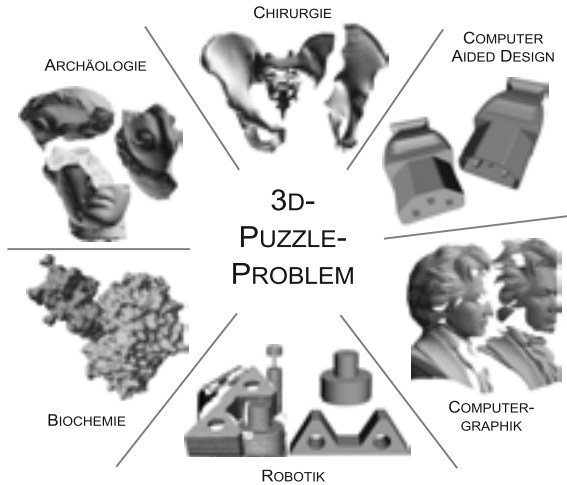


Abbildung 1: Die zentrale Bedeutung des 3d-Puzzle-Problems.

unerhebliches Gesundheitsrisiko ein, da der Fixierungsvorgang laufend durch eine Vielzahl von Röntgenbildaufnahmen (und damit verbundener Strahlenbelastung) überwacht werden muss. In der pharmazeutischen Forschung müssen beispielsweise große Strukturdatenbanken nach einem Protein durchsucht werden, welches eine stabile molekulare Verbindung mit einem anderen Protein bildet. Eine mögliche Anwendung ist die Suche nach einer Substanz, die ein Enzym blockieren (inhibieren) und damit eine Virusinfektion eindämmen kann. Aus diesen Gründen besteht großes Interesse an effizienten Methoden zur automatischen Lösung des 3d-Puzzle-Problems. Die Fragmentoberflächen müssen hierzu zuerst in eine geeignete digitale Form gebracht werden. Das Ziel ist dann, durch spezielle Methoden aus dem Bereich Bildverarbeitung, 3d-Computer-Sehen und Mustererkennung die einzelnen Fragmente virtuell zusammensetzen und somit 'Baupläne' oder Bindungsfähigkeitsinformationen zu generieren.

## 2 Gewinnung von Oberflächendaten

Je nach Anwendungsfall eignen sich unterschiedliche Aufnahmetechniken, um an die Oberflächendaten der Objekte zu gelangen. Aus diesem Grund habe ich mich früh mit der Gewinnung von Oberflächendaten beschäftigt, die die Basis aller weiteren Verfahren bilden. Hierbei entstand beispielsweise das mit „Shape from Stripe Pattern“ bezeichnete Verfahren [WW01], [WW02] (siehe Abbildung 2 (a)). Dabei wird per Lichtprojektor ein statisches Streifenmusters auf die Oberfläche des Objektes projiziert. Aus der Deformation des Streifensignals im Kamerabild kann dann die Form der Oberfläche rekonstruiert werden. Ein großer Vorteil dieser Technik besteht darin, dass zum einen nur ein kleiner kostengünstiger Festmusterprojektor benötigt wird und zum anderen die Akquisition von dynamischen Objekten möglich ist. Alternativ kann man auch einen Laserscanner oder

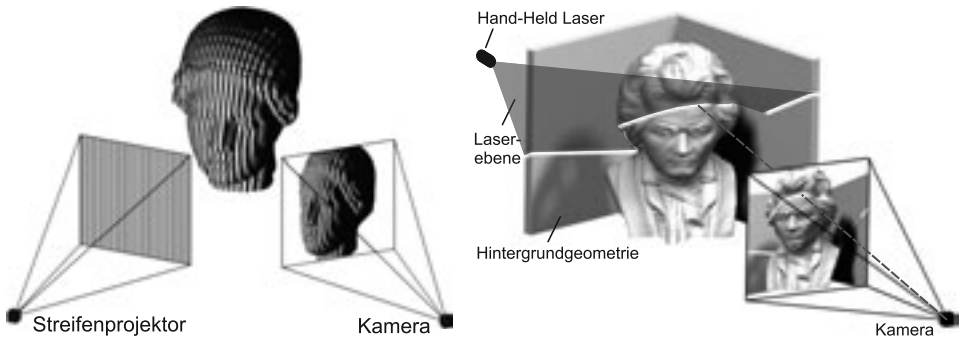


Abbildung 2: Alternative Methoden zur Gewinnung von Oberflächendaten: (links) Shape from Stripe Pattern; (rechts) Laserscanner mit manuell geführtem Laser.



Abbildung 3: (a) Kamerabild einer Beethovenbüste; (b) gescannte Oberfläche künstlich beleuchtet; (c) 3d Scan einer Tonfigur; (d) 3d Scan meines Kollegen Sven Molkenstruck.

Licht-Schnitt-Verfahren verwenden (siehe z.B. [Bla04] für einen umfassenden Überblick über bekannte Techniken). Auch hier haben wir einen neuen besonders kostengünstigen Ansatz entwickelt [WMW06], bei dem ein manuell geführter Laser eingesetzt werden kann (siehe Abbildung 2 (b)). Diese Methode hat unter anderem den Vorteil, dass die geringen Hardwarekosten auch von Studenten und privaten Entwicklern aufgebracht werden können. Die nötige Software ist inzwischen kostenlos im Internet erhältlich [WM07]. Die Knochenfrakturen sind natürlich nicht von außen sichtbar, sondern wurden aus Computertomogrammen rekonstruiert. Darüber hinaus wurden zu Testzwecken einige 3d-Objekte künstlich zerbrochen. Abbildung 3 zeigt exemplarisch zwei experimentelle Ergebnisse des Laserscanner-Ansatzes. Hier ist die erreichte Detailgenauigkeit gut zu erkennen. Untersuchungen der Messgenauigkeit mit einem bekannten Testobjekt, einem Triangulationswinkel von ca.  $30\text{-}35^\circ$  und einem Kameraabstand von 600 mm zum Messobjekt ergaben eine Standardabweichung von nur 0,37 mm.

### 3 Das 3d-Puzzle-Problem

Nachdem die Fragmente in eine geeignete digitale Form gebracht wurden, kann das Zusammensetzen der Oberflächen beginnen. Die meisten verwandten Arbeiten behandeln die Registrierung von partiell überlappenden Oberflächen (die z.B. per Tiefensensor aus unterschiedlichen Richtungen gemessen wurden) und verwenden lokale Oberflächenmerkmale für die Suche nach korrespondierenden Oberflächenpunkten. Ein Überblick über Methoden zur merkmalsbasierten Registrierung findet sich z.B. in [SL00]. Der Nachteil dieser Ansätze ist der verhältnismäßig hohe Zeitaufwand, der für die Merkmalsextraktion und Sortierung benötigt wird. Sind die Merkmale nicht robust messbar (z.B. aufgrund von Rauschen oder fehlenden Oberflächenstrukturen) oder wenig prägnant bzw. nicht eindeutig einander zuzuordnen (Korrespondenzproblem), können diese Verfahren für sich allein genommen das Problem nicht lösen, aber häufig den Suchraum erheblich einschränken. Das allgemeine 3d-Puzzle-Problem wurde nur sehr wenig untersucht. Einige interessante Arbeiten behandeln die Rekonstruktion von wertvollen zerbrochenen archäologischen Tongefäßen oder Keramiken anhand ihrer Bruchkanten, Texturen oder Rotationssymmetrieeigenschaften (siehe z.B. [KS03], [dS02], [GMB02], [WC04]). Diese Methoden lösen jedoch eher ein 2d-Puzzle-Problem und können im Allgemeinen nicht auf 3d Fragmente übertragen werden.

Bei näherer Betrachtung der Problemstellung stellt man fest, dass der sechsdimensionale Suchraum aller relativen Lagen zwischen zwei Fragmenten große Bereiche hat, in denen die Fragmente sich nicht berühren. Diese Bereiche sollten von vornherein nicht untersucht werden. Die meisten bekannten Optimierungsverfahren (wie Gradientenabstieg, 'hill climber', 'simulated annealing', etc.) entarten zu einer nahezu vollständigen Suche, da der Suchraum oft äußerst stark mit lokalen Optima frequentiert ist, wobei außerdem ein Großteil der Zeit für die unnötige Iteration in falsche lokale Optima verschwendet wird. Bei dem folgenden eigenen Algorithmus handelt es sich, ähnlich wie beim klassischen RANSAC-Algorithmus [FB81], um eine alternierende Hypothesengenerierung und Hypothesenverifizierung:

1. Generiere eine wahrscheinliche Lagehypothese durch Annahme eines tangentialen Kontaktes zwischen zwei zufällig ausgewählten Punktpaaren auf der Oberfläche.
2. Bewerte die Güte der Lagehypothese durch eine schnelle Hochrechnung der Kontaktfläche.
3. Wiederhole die Schritte 1 und 2 und speichere das jeweils beste Ergebnis, solange bis die Güte hinreichend hoch ist, alle Hypothesen getestet wurden oder die maximale Suchzeit abgelaufen ist.

Der Algorithmus ist aufgrund seiner Zufallskomponente nicht deterministisch, aber terminiert nach endlicher Zeit, falls in Schritt 1 dafür gesorgt wird, dass keine Punktpaarcombination doppelt ausgewählt wird.

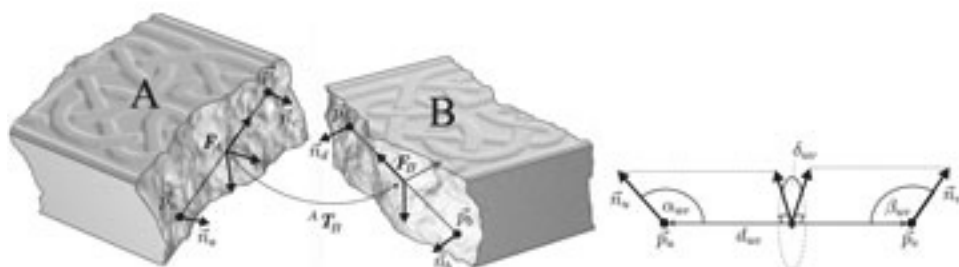


Abbildung 4: (Links) Berechnung der relativen Transformation  ${}^A T_B$ ; (Rechts) Relationen zwischen zwei orientierten Punkten  $\vec{u}$  und  $\vec{v}$

### 3.0.1 Generierung einer wahrscheinlichen Lagehypothese per Zufall

Angenommen es liegen die Oberflächen von zwei Fragmenten in Form von Punktwolken vor. Gegeben sei also die Menge  $\mathcal{P}_A$  der 3d Punktkoordinaten  $\vec{p}_1, \dots, \vec{p}_k$  auf der Oberfläche des Fragmentes A und die Menge  $\mathcal{N}_A$  von zugehörigen Oberflächennormalen  $\vec{n}_1, \dots, \vec{n}_k$ . Durch Kombination von Punktkoordinaten und Oberflächennormalen erhalten wir für jeden Oberflächenpunkt einen 6d Parametervektor, den wir im Folgenden *orientierten Punkt* nennen werden. Die Punktwolken können damit als eine Menge  $\mathcal{A}$  von orientierten Punkten auf Fragment A und die Menge  $\mathcal{B}$  von orientierten Punkten auf dem Gegenstück B aufgefasst werden

$$\mathcal{A} := \{ \vec{u} = [\vec{p}_u, \vec{n}_u] \mid \vec{p}_u \in \mathcal{P}_A \text{ und } \vec{n}_u \in \mathcal{N}_A \} \quad (1)$$

$$\mathcal{B} := \{ \vec{v} = [\vec{p}_v, \vec{n}_v] \mid \vec{p}_v \in \mathcal{P}_B \text{ und } \vec{n}_v \in \mathcal{N}_B \}. \quad (2)$$

Ein Oberflächenkontakt liegt vor, wenn sich die Oberflächen der Fragmente an mindestens einem Punkt tangential berühren. Wir können beispielsweise eine Lagehypothese zwischen zwei Fragmenten A und B konstruieren, indem wir einen tangentialen Kontakt zwischen den orientierten Punkten  $\vec{a} \in \mathcal{A}$  und  $\vec{b} \in \mathcal{B}$  sowie zwischen  $\vec{c} \in \mathcal{A}$  und  $\vec{d} \in \mathcal{B}$  annehmen. Diese Annahme beschränkt alle Freiheitsgrade der relativen Lage. Aus einer solchen Lagehypothese kann eine homogene  $4 \times 4$ -Transformationsmatrix berechnet werden. Abbildung 4 illustriert eine mögliche Vorgehensweise. Die relative Transformation  ${}^A T_B$  wird hier mittels zweier Koordinatensysteme  $F_A$  und  $F_B$  (jeweils ein Koordinatensystem pro Fragment) ermittelt. Allerdings ist ein exakter tangentialer Kontakt an zwei Punkten nur dann möglich, wenn die Abstände der Punktpaare und die Oberflächenorientierungen an den Kontaktpunkten übereinstimmen, bzw. wenn das orientierte Punktpaar  $(\vec{a}, \vec{c})$  geometrisch *komplementär* zu dem orientierten Punktpaar  $(\vec{b}, \vec{d})$  ist. Die räumliche Beziehung zwischen zwei orientierten Punkten auf einem Fragment kann eindeutig durch vier Relationen definiert werden. Diese sind im Folgenden zu einem 4d Relationsvektor zusammengefasst:

$$\text{rel}(\vec{u}, \vec{v}) := \begin{bmatrix} d_{uv} \\ \cos \alpha_{uv} \\ \cos \beta_{uv} \\ \delta_{uv} \end{bmatrix} := \begin{bmatrix} \|\vec{p}_v - \vec{p}_u\| \\ \vec{n}_u \cdot \vec{p}_{uv} \\ \vec{n}_v \cdot \vec{p}_{uv} \\ \text{atan2}(\vec{n}_u \cdot (\vec{p}_{uv} \times \vec{n}_v), (\vec{n}_u \times \vec{p}_{uv}) \cdot (\vec{p}_{uv} \times \vec{n}_v)) \end{bmatrix}. \quad (3)$$

Die Kernidee der Hypothesengenerierung besteht einerseits darin, nur relative Lagen zu prüfen, die sich an mindesten zwei Punkten tangential berühren und andererseits möglichst schnell viele dieser Lagehypothesen zu generieren. Hierzu wählen wir ein Punktpaar auf Fragment A aus und benötigen dann ein geometrisch komplementäres Punktpaar auf Fragment B. Die Lagehypothese ist dann die Annahme eines tangentialen Kontaktes zwischen beiden Punktpaaren. Angenommen wir haben zwei vollständig geometrisch komplementäre Oberflächen A und B die jeweils durch  $n$  orientierten Punkte repräsentiert werden. Die Frage ist: Wie lange dauert es, bis man zwei korrespondierende Punktpaare (eines auf A und eines auf B) findet? Hat man ein Punktpaar auf A gewählt, dann ist die Wahrscheinlichkeit, zufällig das korrespondierende Punktpaar auf B zu ziehen,  $1/n^2$ . Man muss also im Schnitt  $n^2+1$  Punktpaare betrachten, bis man zwei korrespondierende Paare erwischt (Ziehen mit Zurücklegen). Dies entspricht bereits einer nicht zu vernachlässigenden Komplexität von  $O(n^2)$ .

Doch durch einen einfachen Trick geht es wesentlich schneller: Angenommen man trägt die gewählten Punktpaare in eine 'Hashtabelle' ein, wobei die Tabelle mit translations- und rotationsinvarianten Merkmalen (als Hashwert) indiziert wird. Nehmen wir weiterhin an, dass die Hashwerte eindeutig sind. Wenn man nun alternierend erst ein zufälliges Punktpaar von A ziehen und in die Hashtabelle einträgt und dann ein zufälliges Punktpaar von B ziehen und in die Hashtabelle einträgt, brauchen man im Schnitt nur noch ca.  $1,2 \cdot n$  Paare zu betrachten, bis es zu einer Kollision kommt. Diese Methode entspricht dem so genannten *Geburtstagsangriff* aus dem Bereich der Kryptologie, welcher dazu missbraucht werden kann, zwei verschiedene Dokumente mit der gleichen digitalen Signatur (Hashwerten) zu erzeugen.

Nach diesem Prinzip arbeitet auch der Algorithmus. Anstatt einer Hashtabelle verwenden wir jeweils eine 4d *Relationstabelle* pro Fragment. Diese Tabellen werden nun alternierend mit zufälligen Punktpaaren gefüllt. Als Tabellenindizes dienen uns die vier translations- und rotationsinvarianten Relationen eines orientierten Punktpaares  $(\vec{u}, \vec{v})$  aus Gleichung (3). Die Suchschleife des Algorithmus arbeitet nun folgendermaßen:

1. Wähle ein zufälliges orientiertes Punktpaar  $\vec{a}, \vec{c} \in \mathcal{A}$  und berechne den zugehörigen Relationsvektor  $\text{rel}(\vec{a}, \vec{c})$ .
2. Trage das Punktpaar in die Relationstabellen  $R_A$  ein:  $R_A[\text{rel}(\vec{a}, \vec{c})] \leftarrow (\vec{a}, \vec{c})$ .
3. Lies die andere Relationstabelle  $R_B$  an der gleichen Stelle aus:  $(\vec{b}, \vec{d}) \leftarrow R_B[\text{rel}(\vec{a}, \vec{c})]$ . Falls an dieser Stelle bereits ein Eintrag steht  $\Rightarrow$  neue Lagehypothese  $(\vec{a}, \vec{b}, \vec{c}, \vec{d})$ .
4. Wähle ein zufälliges orientiertes Punktpaar  $\vec{b}, \vec{d} \in \mathcal{B}$  und berechne den zugehörigen Relationsvektor  $\text{rel}(\vec{b}^*, \vec{d}^*)$ .
5. Trage das Punktpaar in die Relationstabellen  $R_B$  ein:  $R_B[\text{rel}(\vec{b}^*, \vec{d}^*)] \leftarrow (\vec{b}, \vec{d})$ .
6. Lies die Relationstabelle  $R_A$  an der gleichen Stelle aus:  $(\vec{a}, \vec{c}) \leftarrow R_A[\text{rel}(\vec{b}^*, \vec{d}^*)]$ . Falls an dieser Stelle bereits ein Eintrag steht  $\Rightarrow$  neue Lagehypothese  $(\vec{a}, \vec{b}, \vec{c}, \vec{d})$ .

Diese Schritte werden wiederholt, bis die Hypothese gut genug ist, alle Kombinationen getestet wurden oder das Zeitlimit abgelaufen ist. Natürlich müssen in Schritt 1 und 4

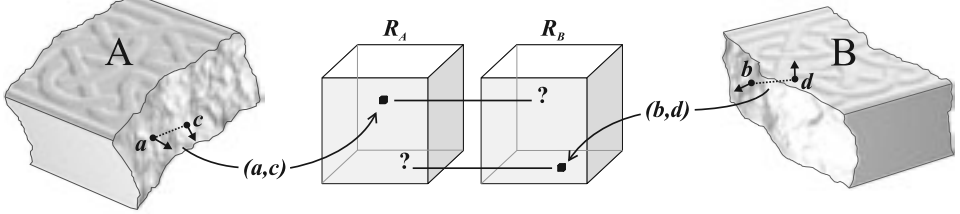


Abbildung 5: Generierung von Lagehypothesen: In einer Schleife werden alternierend Punktpaare in die Relationstabellen  $R_A$  und  $R_B$  eingetragen. Hierdurch füllen sich die Tabellen und es finden sich nach kurzer Zeit Punktpaare mit gleichen Relationen, die demnach per Rotation und Translation aufeinander abbildbar sind und als Lagehypothesen dienen.

die vier Relationen jeweils auf einen beschränkten ganzzahligen Wertebereich normiert werden, damit sie als Tabellenindizes dienen können. In den Experimenten haben sich 4d Relationstabellen mit jeweils  $32^4$  Zellen bewährt. Dies ergibt bei  $2 \times 2$  Bytes pro Zelle einen vertretbaren Speicherbedarf von vier Megabyte pro Tabelle. In Abbildung 5 wird die Vorgehensweise nochmals anschaulich dargestellt. Mit dieser Methode wird eine Laufzeitkomplexität von  $O(n)$  für die erste Hypothese erreicht; diese konvergiert jedoch für weitere Hypothesen gegen eine effiziente Laufzeit von  $O(1)$ , da die Relationstabellen immer weiter gefüllt werden.

### 3.1 Schnelle Bewertung von Lagehypothesen

Nachdem wir ein effizienten Algorithmus für die Hypothesengenerierung gefunden haben, brauchen wir noch eine schnelle Bewertung der Güte einer Lagehypothese. Als Gütemaß eignet sich die Größe der Kontaktfläche zwischen den beiden Fragmenten A und B, bzw. das Verhältnis  $\Omega$  zwischen Kontaktpunktzahl und Gesamtpunktzahl. Dieses Verhältnis kann auch als Wahrscheinlichkeit, dass ein zufälliger Punkt  $\vec{x} \in \mathcal{A}$  in der gegebenen Lage in Kontakt mit einem Punkt  $\vec{y} \in \mathcal{B}$  steht, interpretiert werden. Deshalb können wir  $\Omega$  durch eine effiziente *Monte-Carlo*-Hochrechnung abschätzen, indem wir eine Sequenz von Zufallspunkten auf Kontakt überprüfen. Der Vorteil hierbei ist, dass die Güteschätzung frühzeitig abgebrochen werden kann, falls der Erwartungswert weit unter dem bisher besten Matching-Ergebnis liegt. Angenommen wir haben eine Sequenz  $\vec{x}_1, \dots, \vec{x}_k$  von unabhängigen orientierten Zufallspunkten auf Fragment A, dann ist  $\Omega$  gegeben durch

$$\Omega := \lim_{k \rightarrow \infty} \frac{\sum_{i=1}^k \text{contact}_B(\vec{x}_i)}{k}, \quad (4)$$

wobei  $\text{contact}_B(\vec{x}_i)$  eine Funktion ist, die bestimmt, ob ein Punkt  $\vec{x}$  in Kontakt mit Fragment B steht

$$\text{contact}_B(\vec{x}) = \begin{cases} 1 & \text{falls } \|\vec{p}_x - \vec{p}_y\| < \varepsilon, & \text{(Kontakt)} \\ -4 & \text{falls } \|\vec{p}_x - \vec{p}_y\| \geq \varepsilon \wedge (\vec{p}_x - \vec{p}_y) \cdot \vec{n}_y < 0, & \text{(innen)} \\ 0 & \text{sonst.} & \text{(außen)} \end{cases} \quad (5)$$

Hierbei bezeichnet  $\vec{y} = [\vec{p}_y, \vec{n}_y]$  jeweils denjenigen orientierten Punkt auf Fragment B, der die kürzeste Distanz zu  $\vec{x}$  hat

$$\vec{y} := \arg \min_{\vec{b} \in \mathcal{B}} \|\vec{p}_x - \vec{p}_b\|. \quad (6)$$

Um Durchdringungen zu vermeiden, wurde außerdem in der Kontaktfunktion (5) eine Gütestrafe für Durchdringungspunkte (also Punkte des einen Fragments, die sich innerhalb des anderen Fragments befinden) subtrahiert. Für die Suche nach dem Punkt mit der kürzesten Distanz verwenden wir einen *kd-tree* [FBF77], wodurch eine logarithmische Zeitkomplexität erreicht wird. Im Gegensatz zu Gleichung (4) ist es jedoch nur möglich eine begrenzte Anzahl von Zufallspunkten auf Kontakt zu testen. Aus diesem Grund kann  $\Omega$  nur bis zu einem vorgegebenen Genauigkeitsgrad approximiert werden. Dies ist jedoch nur bei kontinuierlichen Oberflächen eine Einschränkung, da bei Punktwolken die Anzahl von vornherein begrenzt ist. Unter Berücksichtigung eines marginalen Fehlers kann  $\Omega$  für jeden zusätzlichen Zufallspunkt genauer approximiert werden:

$$\Omega \approx \frac{\sum_{i=1}^k \text{contact}_B(\vec{x}_i)}{k} \pm \frac{1,96}{2\sqrt{k}}. \quad (7)$$

Wobei der hintere Term ein Konfidenzintervall beschreibt, innerhalb dessen der tatsächliche Wert mit einer 95%-tigen Wahrscheinlichkeit liegt. Wenn die obere Grenze des Intervalls schlechter als das bisher beste Matching-Ergebnis ist, brechen wir die Berechnung ab und testen die nächste Lagehypothese. Auf diese Weise gewinnt die Güteberechnung während der Laufzeit mit jeder neuen besseren Hypothese an Geschwindigkeit.

## 4 Experimentelle Ergebnisse und Zusammenfassung

Das in dieser Arbeit vorgestellte Verfahren, sowie weitere Strategien zur Lösung des 3d-Puzzle-Problems wurden in [Win06] ausführlich untersucht. In der Vielzahl der Experimente haben sich die Matching-Ansätze als äußerst robust gegenüber Oberflächenrauschen und Ausreißern erwiesen. Die außerordentliche Effizienz der Methoden ist allen bis dato bekannten und vergleichbaren Verfahren überlegen. Selbst für große Datensätze wurden auf einem handelsüblichen PC Laufzeiten von unter einer Sekunde erreicht. Aufgrund des begrenzten Umfangs kann hier nur ein flüchtiger Eindruck der untersuchten Anwendungsmöglichkeiten vermittelt werden.

Abbildung 6 zeigt eine Auswahl von Testfragmenten aus unterschiedlichsten Anwendungsbereichen. Während für die Fragmente (a)–(e) der hier vorgestellte Algorithmus stets zum Erfolg führt, muss für die Fragmente (f)–(j) zusätzliches Vorwissen verwendet werden, da die relative Lage mit dem größten Oberflächenkontakt nicht zwangsläufig die gesuchte ist. Aus diesem Grund wird in [Win06] je nach Anwendung zusätzliches Wissen über Form und Symmetrien integriert. Insbesondere gelingt es, gebrochene Oberschenkel- und Beckenknochen durch eine automatische Schätzung von Achs- und Spiegelsymmetrien robust und präzise zusammensetzen. Hierdurch können in Zukunft Repositionierungsfehler weitestgehend vermieden und intraoperative Röntgenstrahlenbelastungen für Patient



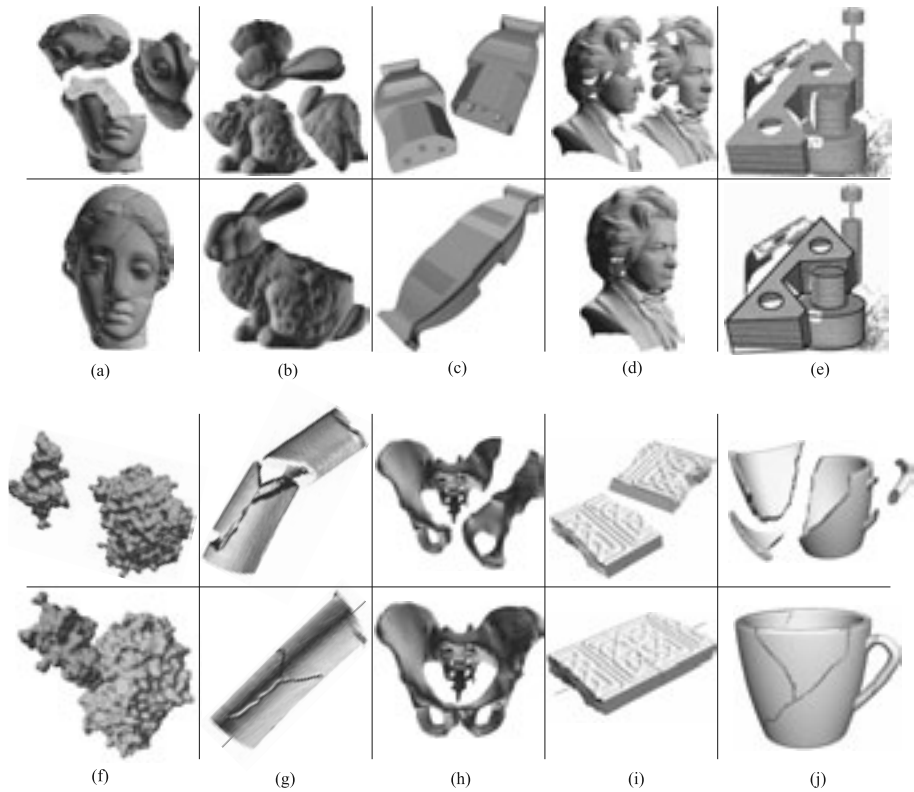


Abbildung 6: Verschiedene Testfragmente in ihrer initialen Lage (oben) und nach dem Matching (unten): (a) zerbrochener Venus-Kopf; (b) künstlich zerbrochener Stanford Bunny; (c) IEC-320 Stromverbinder; (d) 3d-Scans einer Beethoven-Büste; (e) 3d-Scan einer Testszene und hiermit geschätzte Lage der CAD-Modelle; (f) Docking zweier Proteine; (g) Schaftbereich einer Oberschenkelfraktur; (h) Beckenfraktur; (i) zerbrochener Quader; (j) zerbrochene Tasse.

und Chirurgen verringert werden. Schließlich wurde das große Potenzial der Verfahren in weiteren wichtigen Anwendungsgebieten (wie der Rekonstruktion von archäologischen Artefakten, der Registrierung von Oberflächendaten und der 3d Objekterkennung und Lageschätzung) demonstriert.

## Literatur

- [Bla04] F. Blais. Review of 20 Years Range Sensor Development. *Journal of Electronic Imaging*, 13(1), 2004.
- [dS02] H. C. da Gama Leitão und J. Stolfi. A Multiscale Method for the Reassembly of Two-Dimensional Fragmented Objects. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(9):1239–1251, September 2002.

- [FB81] M. A. Fischler und R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, Juni 1981.
- [FBF77] J. H. Friedman, J. L. Bentley und R. A. Finkel. An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Transactions on Mathematical Software*, 3(3):209–226, September 1977.
- [GMB02] D. Goldberg, C. Malon und M. Bern. A global approach to automatic solution of jigsaw puzzles. In *Annual Symposium on Computational Geometry*, Seiten 82–87. ACM Press, 2002.
- [KS03] M. Kampel und R. Sablatnig. Profile-based Pottery Reconstruction. *Conference on Computer Vision and Pattern Recognition Workshop*, 1:4, 2003.
- [SL00] S. Seeger und X. Labourex. Feature extraction and registration: An overview. *Principles of 3D Image Analysis and Synthesis*, Seiten 153–166, 2000.
- [WC04] A. Willis und D. Cooper. Bayesian Assembly of 3D Axially Symmetric Shapes from Fragments. *IEEE Conference on Computer Vision and Pattern Recognition*, 1:82–89, 2004.
- [Win06] S. Winkelbach. *Das 3d-Puzzle-Problem – Effiziente Methoden zum paarweisen Zusammensetzen von dreidimensionalen Fragmenten*. Fortschritte in der Robotik, Band 10. Shaker Verlag, 2006.
- [WM07] S. Winkelbach und S. Molkenstruck. DAVID-Laserscanner, Low-Cost 3D Laser Range Scanning for Everyone, [www.david-laserscanner.com](http://www.david-laserscanner.com), Stand:12. April 2007.
- [WMW06] S. Winkelbach, S. Molkenstruck und F. Wahl. Low-Cost Laser Range Scanner and Fast Surface Registration Approach. In *Pattern Recognition, 28th DAGM Symposium*, Lecture Notes in Computer Science 4174, Seiten 718–728. Springer, 2006.
- [WW01] S. Winkelbach und F. M. Wahl. Shape from 2D Edge Gradients. In *Pattern Recognition, 23th DAGM Symposium*, Lecture Notes in Computer Science 2191, Seiten 377–384. Springer, September 2001.
- [WW02] S. Winkelbach und F. M. Wahl. Shape from Single Stripe Pattern Illumination. In *Pattern Recognition, 24th DAGM Symposium*, Lecture Notes in Computer Science 2449, Seiten 240–247. Springer, September 2002.



**Dr.-Ing. Simon Winkelbach** wurde am 12. März 1976 in Göttingen geboren. Von 1996 bis 2001 studierte er Informatik an der TU-Braunschweig. Für seine Diplomarbeit zum Thema „Gradienten basierte Rekonstruktion von 3D Oberflächen“ erhielt Herr Winkelbach 2001 den Jubiläumspreis der Siegfried Werth Stiftung. Nach seinem Abschluss als Diplom-Informatiker begann er die Promotion am Institut für Robotik und Prozessinformatik. Für seine wissenschaftlichen Publikationen gewann er zweimal den Hauptpreis der Deutschen Arbeitsgemeinschaft für Mustererkennung. Außerdem betreute er die Übung zur Vorlesung „Computer-Sehen“ und hält inzwischen selbst die Vorlesung „Algorithmen und Programme“.

Die Arbeiten zum Thema „3d-Puzzle-Problem“ wurde im Rahmen des DFG-Projektes WA 848/14–1 finanziert. Am 18.09.2006 beendete Herr Winkelbach seine Promotion mit dem Gesamtprädikat „Mit Auszeichnung bestanden“.