Int J Comput Vis (2008) 78: 1–13
DOI 10.1007/s11263-007-0121-5

SHORT PAPER

Pairwise Matching of 3D Fragments Using Cluster Trees

Simon Winkelbach · Friedrich M. Wahl

Received: 22 January 2007 / Accepted: 26 December 2007 / Published online: 23 January 2008

Abstract We propose a novel and efficient surface matching approach for reassembling broken solids as well as for matching assembly components using cluster trees of oriented points. The method rapidly scans through the space of all possible contact poses of the fragments to be (re)assembled using a tree search strategy, which neither relies on any surface features nor requires an initial solution. The new method first decomposes each point set into a binary tree structure using a hierarchical clustering algorithm. Subsequently the fragments are matched pairwise by descending the cluster trees simultaneously in a depth-first fashion. In contrast to the reassemblage of pottery and thin walled artifacts, this paper addresses the problem of matching broken 3D solids on the basis of their 2.5D fracture surfaces, which are assumed to be reasonable large. Our proposed contact area maximization is a powerful common basis for most surface matching tasks, which can be adapted to numerous special applications. The suggested approach is very robust and offers an outstanding efficiency.

Keywords Fracture matching · Surface registration · 3D puzzle · Fragment alignment · Cluster tree · Broken objects

1 Introduction

How can we reassemble fragments (e.g. broken objects) automatically even without any knowledge of their original configuration and/or shape? An answer to this crucial question is very important in many fields, like archaeology (e.g. reconstruction of broken artifacts), surgery (e.g. reduction of bone fractures), bioinformatics (e.g. docking of proteins), and assistance in assembly (e.g. assembly planning in robotics). The problem of matching complementary fragments of broken objects is closely related to the problem of partially overlapping surface registration where the objects are e.g. represented as range images. But in case of broken objects additional challenges are faced, which are not met by most registration methods: (i) In general, a good initial guess, which can be used to iterate to the global optimum, is not available. (ii) Very large surface areas have no correspondence on the complementary part. (iii) Object intersections must be avoided. (iv) Missing fragments and additional material deterioration complicates the matching.

1.1 Related Work

An outline of the numerous publications dealing with registration techniques would go beyond the scope of this paper. Therefore, we only give an overview of the most related work. The literature in this area often distinguishes between 'fine registration' and 'coarse registration'.

Fine Registration Fine registration approaches can be used to optimize a given coarse solution. One of the most popular fine registration approaches for surfaces is the *iterative closest point* (ICP) algorithm by Besl and McKay (1992). The algorithm iteratively improves an initial solution according to some fitness criterion. Although many enhancements to the original method have been suggested (e.g. Dalley and Flynn 2002; Pottmann et al. 2006; Rusinkiewicz and Levoy 2001), it still requires a good initial guess to find the global optimum.

S. Winkelbach (✉) · F.M. Wahl
Institute for Robotics and Process Control,
Technical University of Braunschweig, Braunschweig, Germany
e-mail: s.winkelbach@tu-bs.de

Coarse Registration A coarse registration is necessary, if a coarse solution is not initially available. Most approaches in this area use local surface features to find corresponding point pairs. Features vary from simple properties, like *curvatures*, to complex vectors, like *point signatures* (Chua and Jarvis 1997), *surface curves* e.g. Papaioannou and Theoharis (1999), Sappa et al. (2001), Vanden Wyngaerd and Van Gool (2002), Krsek et al. (2002), *spin-images* (Johnson and Hebert 1997), or *salient points* (Schön and Häusler 2005). See e.g. Seeger and Labourex (2000) for an overview of further surface features. Using features can highly constrain the search space. However, their use cannot guarantee unique point correspondences if the features are not robust (e.g. due to noise or missing surface structures) or indiscriminate (correspondence problem). A well-known category dealing with object recognition and localization are the *pose clustering* approaches (also known as *hypothesis accumulation* or *generalized Hough transform*, see for example Stockman 1987; Linnainmaa et al. 1988, or Barequet and Sharir 1996). The basic idea is to accumulate low level pose hypotheses in a voting table, followed by a maxima/cluster search, which identifies the most frequent hypotheses. The drawback of voting tables is their high time and space complexity, particularly in case of large data sets and high-dimensional search spaces.

Reassembly of Broken Objects Many papers address matching of two-dimensional fragments like *jigsaw puzzles* or *thin-walled fragments* (e.g. sherds of pottery), see e.g. Cooper et al. (2002), Kampel and Sablatnig (2003a, b), da Gama Leitão and Stolfi (2002), Goldberg et al. (2004), Willis and Cooper (2004). However, these methods cannot be generalized to fragments of 3D solids. The problem of matching complementary three-dimensional fragments is rarely treated in the open literature up to now. One approach is proposed by Papaioannou et al. (2002), Papaioannou and Theoharis (1999); it is based on a pose error estimation using *z-buffers*, which are generated online for every hypothesized pose. The error minimization is performed by *simulated annealing* in a 7D search space of all possible poses of two fragments in relation to a separating projection plane. An efficient and robust approach for matching fragments is the *random sample matching* approach (Winkelbach et al. 2004, 2006), which is based on the *RANSAC* algorithm introduced by Fischler and Bolles (1981). The repetitive procedure is simple but powerful: First, a likely hypothesis is generated randomly from the input data set. Subsequently, the quality of the hypothesis (number of contact points) is evaluated. The main disadvantage of this randomized algorithm is its infinite search loop, which makes it impossible to decide, whether the best result was found or not. The most recent approach for reassembling broken 3D solids is from Huang et al. (2006). Their comprehensive reconstruction process consists of several steps: (i) a surface segmentation into faces that are bounded by sharp curves, (ii) a classification into original and fractured faces using a roughness criterion, (iii) a multi scale feature extraction based on volume distance descriptors, (iv) a pairwise matching of the faces based on feature correspondences, and (v) an iterative greedy multi-piece matching. In this paper we suggest an alternative technique for pairwise fragment matching, which neither relies on sharp curves nor requires any features.

Bounding Volume Hierarchies Another class of related methods is concerned with *collision detection* and *distance queries* in computer graphics applications and surface registration. The most popular and versatile approaches in this field of research are *bounding volume hierarchies* (BVH) and space-partitioning trees, like the *binary space-partitioning tree* (BSP tree) or kd-tree (see e.g. Ericson 2005). By arranging the objects into tree hierarchies, the collision and distance queries can be reduced to logarithmic time complexity. But up to now, no work has been published that simultaneously matches tree hierarchy representations of two objects. In the following, we present such a novel and highly efficient matching method of tree hierarchies, whose principle goes far beyond using hierarchies for simple distance queries.

1.2 Method Overview

Our suggested cluster tree matching strategy rapidly finds the global optimum in the space of all rigid transformations between two fragments without any knowledge of an initial guess. Here ‘global optimum’ means the relative pose with the largest surface contact. The tolerated distance radius between surface points that are considered to be in contact must be predefined. The aim of our approach is to find a good match without knowledge of an initial pose hypothesis. As soon as such an pose hypothesis was found, several known iterative fine registration approaches can be applied to optimize accuracy. The paper is organized as follows: Sect. 2 gives a theoretical concept for matching two point-sampled surfaces, but at this stage without using a hierarchical decomposition. Section 3 introduces the cluster tree matching strategy for 3D-puzzle-problems. In a preprocessing step, we store each fragment in a binary tree structure (cluster tree) by a hierarchical subdivision of the point sets into subsets using a divisive clustering method. Subsequently, we propose an efficient matching method, which descends the cluster trees simultaneously in a depth-first fashion by propagating each pose hypothesis from one tree level to the next. The descent along each branch is discontinued if the expected result is not better than the last best match. Thus, transformations with weak surface contact can

be discarded at a very early stage. The algorithm always terminates and returns the global optimum of the search space. In Sect. 4 we discuss strategies to accelerate execution and indicate further matching constraints. Finally, in Sect. 5 we evaluate the excellent performance of our new approach by matching different types of objects.

2 Preliminaries

Before discussing an efficient method for solving a 3D-puzzle-problem, we outline some mathematical basics. Let us assume that there is a set $\mathcal{P}_A$ of 3D point coordinates $\mathbf{p}_1, \ldots, \mathbf{p}_k$ of the surface of the first fragment A and a set $\mathcal{N}_A$ of corresponding 3D surface normals $\mathbf{n}_1, \ldots, \mathbf{n}_k$ at these points. Note that we consider outward-pointing *surface normals*, which are unit vectors perpendicular to the surface. Many surface reconstruction approaches can compute normal vectors as part of the reconstruction process. If surface normals are not given in advance, or if they are not robust due to heavy noise, an additional computation of robust surface normals is necessary. E.g. Mitra and Nguyen (2003) suggests a robust method for estimating surface normals of a point cloud in the presence of noise, which is based on local least square fitting.

Let $\mathbf{u} := [\mathbf{p}_u, \mathbf{n}_u]$ be a 6D parameter vector, which combines the coordinates and the normal of a surface point with index u. Referring to Johnson and Hebert (1997) we call this vector an *oriented point*. This gives us the oriented point set $\mathcal{A}$ of fragment A and the oriented point set $\mathcal{B}$ of the counter fragment B

$$\begin{aligned} \mathcal{A} &:= \{\mathbf{u} = [\mathbf{p}_u, \mathbf{n}_u] \mid \mathbf{p}_u \in \mathcal{P}_A \text{ and } \mathbf{n}_u \in \mathcal{N}_A\}, \\ \mathcal{B} &:= \{\mathbf{v} = [\mathbf{p}_v, \mathbf{n}_v] \mid \mathbf{p}_v \in \mathcal{P}_B \text{ and } \mathbf{n}_v \in \mathcal{N}_B\}. \end{aligned} \tag{1}$$

The goal is to find the relative transformation that 'correctly' assembles the fragments. One of the most important criteria is the amount of contact between the fragments. An approach that considers the whole contact area promises more robust matching results than approaches that solely rely on local surface features. Two fragments are in contact if there exists at least one point, where both fragments touch each other tangentially.

Definition 1 (Tangential point-point contact) Given a relative transformation ${}^A\mathbf{T}_B$ (4×4 matrix in homogeneous coordinate notation) between fragment A and fragment B; two oriented points $\mathbf{a} \in \mathcal{A}$ and $\mathbf{b} \in \mathcal{B}$ are in tangential contact if the point coordinates coincide $\mathbf{p}_a = {}^A\mathbf{T}_B\, \mathbf{p}_b$ and the respective surface normals are directed against each other $\mathbf{n}_a = -{}^A\mathbf{T}_B\, \mathbf{n}_b$.

Definition 2 (Tangential point-fragment contact) Given a relative transformation, an oriented point $\mathbf{a}$ is in tangential contact with fragment B if there exists an oriented point $\mathbf{b} \in \mathcal{B}$ that is in tangential contact with $\mathbf{a}$.

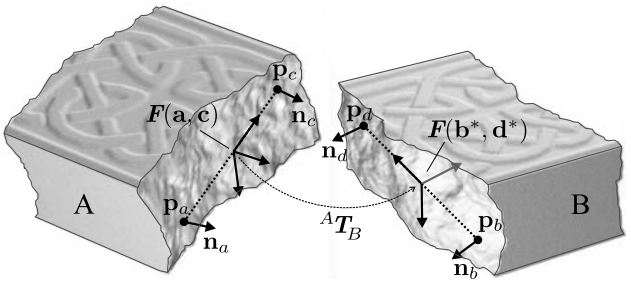


Fig. 1 The assumption of a tangential contact between two oriented point pairs can be used to define a relative transformation ${}^A\boldsymbol{T}_B$

Now we can specify the subset $\mathcal{C}$ of points of fragment A that are in tangential contact with fragment B given a pose ${}^A\boldsymbol{T}_B$

$$\begin{aligned} \mathcal{C} := \{\mathbf{a} \in \mathcal{A} \mid \exists\, \mathbf{b} \in \mathcal{B} : \\ \|\mathbf{p}_a - {}^A\boldsymbol{T}_B\, \mathbf{p}_b\| < \varepsilon_p \wedge (\mathbf{n}_a \cdot {}^A\boldsymbol{T}_B\, \mathbf{n}_b) + 1 < \varepsilon_n\}, \end{aligned} \tag{2}$$

where operator $\cdot$ indicates the dot product (i.e. scalar product) of two vectors. In contrast to Definition 1, this formula assumes a tangential contact if the point-point distance is smaller than ε_p and if the cosine of the angle between the normals is less than $\varepsilon_n - 1$. These tolerance values are necessary to handle numerical errors and noise, since point sets of real datasets never match perfectly. The optimal values for ε_p and ε_n depend on the application. In most cases ε_p and ε_n should be adapted to the surface accuracy (e.g. the standard deviation of the 3D scanner or the CT slice distance). The remaining problem is to find the relative pose ${}^A\boldsymbol{T}_B$ that maximizes the amount of contact.

Obviously, it is not efficient to exhaustively search through the 6D space of all relative poses. Therefore we only consider pose hypotheses with a contact between the fragments. We can construct such a pose hypothesis by assuming a contact between some surface points on each fragment. More precisely, four given oriented surface points $\mathbf{a}, \mathbf{c} \in \mathcal{A}$ and $\mathbf{b}, \mathbf{d} \in \mathcal{B}$ are sufficient if we assume a tangential contact between $\mathbf{a}$ and $\mathbf{b}$ as well as between $\mathbf{c}$ and $\mathbf{d}$. This assumption constrains all degrees of freedom of the relative transformation. As illustrated in Fig. 1, we can determine this homogeneous 4×4 transformation matrix ${}^A\boldsymbol{T}_B$ by means of two predefined frames (one coordinate system for each fragment):

$${}^A\boldsymbol{T}_B(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}) \;=\; \boldsymbol{F}(\mathbf{a}, \mathbf{c})^{-1}\, \boldsymbol{F}(\mathbf{b}^*, \mathbf{d}^*), \tag{3}$$

where the superscripted star indicates the surface normal inversion $\mathbf{b}^* := [\mathbf{p}_b, -\mathbf{n}_b]$, $\mathbf{d}^* := [\mathbf{p}_d, -\mathbf{n}_d]$, and the function

$\boldsymbol{F}(\mathbf{u}, \mathbf{v})$ is a homogeneous 4×4 transformation matrix, representing a coordinate system located between the oriented points $\mathbf{u}$ and $\mathbf{v}$

$$\boldsymbol{F}(\mathbf{u}, \mathbf{v}) := \begin{bmatrix} \frac{\mathbf{p}_{uv} \times \mathbf{n}_{uv}}{\|\mathbf{p}_{uv} \times \mathbf{n}_{uv}\|} & \mathbf{p}_{uv} & \frac{\mathbf{p}_{uv} \times \mathbf{n}_{uv} \times \mathbf{p}_{uv}}{\|\mathbf{p}_{uv} \times \mathbf{n}_{uv} \times \mathbf{p}_{uv}\|} & \frac{\mathbf{p}_u + \mathbf{p}_v}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

with the difference vector $\mathbf{p}_{uv}$ and the combined normal vector $\mathbf{n}_{uv}$

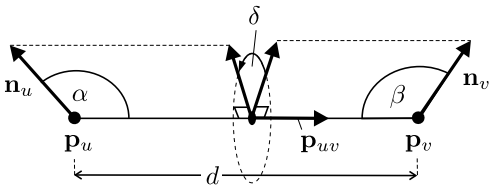
$$\mathbf{p}_{uv} := \frac{\mathbf{p}_v - \mathbf{p}_u}{\|\mathbf{p}_v - \mathbf{p}_u\|}; \qquad \mathbf{n}_{uv} := \mathbf{n}_u + \mathbf{n}_v. \tag{5}$$

To avoid singular frames, we must ensure that the length of $\mathbf{p}_{uv}$ and $\mathbf{n}_{uv}$ is not zero. The calculated transformation ${}^{A}\boldsymbol{T}_{B}$ aligns both point pairs. However, an exact tangential contact at two points is only possible if the relative distance of the points and the surface orientations at the contact points coincide. More precisely, we have to ensure that the oriented point pair $(\mathbf{a}, \mathbf{c})$ is geometrically congruent to the oriented point pair $(\mathbf{b}^*, \mathbf{d}^*)$. The relative transformation of one oriented point to another has four degrees of freedom (two translational and two rotational degrees). Therefore we have to compare at least four scalar quantities to check for congruency. To verify this constraint, we define a 4D relation vector of an oriented point pair consisting of four values (one distance and three angles) that define the relative pose between two oriented points without ambiguity

$$\begin{aligned} &\mathrm{rel}(\mathbf{u}, \mathbf{v}) \\ &:= \begin{bmatrix} d_{uv} \\ \cos\alpha_{uv} \\ \cos\beta_{uv} \\ \delta_{uv} \end{bmatrix} \\ &:= \begin{bmatrix} \|\mathbf{p}_v - \mathbf{p}_u\| \\ \mathbf{n}_u \cdot \mathbf{p}_{uv} \\ \mathbf{n}_v \cdot \mathbf{p}_{uv} \\ \mathrm{atan2}(\mathbf{n}_u \cdot (\mathbf{p}_{uv} \times \mathbf{n}_v), (\mathbf{n}_u \times \mathbf{p}_{uv}) \cdot (\mathbf{p}_{uv} \times \mathbf{n}_v)) \end{bmatrix}. \end{aligned} \tag{6}$$

These four values are illustrated in Fig. 2. Angle δ_{uv} denotes the dihedral angle between the plane with normal $\mathbf{n}_u \times \mathbf{p}_{uv}$ and the plane with normal $\mathbf{n}_v \times \mathbf{p}_{uv}$. Function $\mathrm{atan2}(x, y)$ is similar to calculating the arctangent of y/x except that the signs of both arguments are used to determine the quadrant of the return value

$$\mathrm{atan2}(y, x) := \begin{cases} \arctan\frac{y}{x} & \text{if } x > 0, \\ \arctan\frac{y}{x} + \pi & \text{if } x < 0,\ y \geq 0, \\ \arctan\frac{y}{x} - \pi & \text{if } x < 0,\ y < 0, \\ +\pi/2 & \text{if } x = 0,\ y > 0, \\ -\pi/2 & \text{if } x = 0,\ y < 0, \\ 0 & \text{if } x = 0,\ y = 0. \end{cases} \tag{7}$$



d distance between point $\mathbf{p}_u$ and point $\mathbf{p}_v$
α angle between the normal $\mathbf{n}_u$ and vector $\mathbf{p}_{uv}$ ($\mathbf{p}_{uv}$ is the unit vector pointing from $\mathbf{p}_u$ to $\mathbf{p}_v$)
β angle between the normal $\mathbf{n}_v$ and vector $\mathbf{p}_{uv}$
δ signed angle between $\mathbf{n}_u$ and $\mathbf{n}_v$ around $\mathbf{p}_{uv}$ (i.e. dihedral angle between the plane with normal $\mathbf{n}_u \times \mathbf{p}_{uv}$ and the plane with normal $\mathbf{n}_v \times \mathbf{p}_{uv}$)

Fig. 2 Geometric relationship between two oriented points $\mathbf{u}$ and $\mathbf{v}$

Note that the relation vector of (6) is invariant w.r.t. rotation and translation. In principle, every other relation vector that defines a unique relative pose between two oriented points works just as well. Incidentally, Wahl et al. (2003) showed that similar relation vectors can be accumulated in feature histograms for rapid 3D-shape classification.

Now consider the congruence relation ($\cong$) between oriented point pairs

$$(\mathbf{u}, \mathbf{v}) \cong (\mathbf{q}, \mathbf{r}) \quad :\Longleftrightarrow \quad \mathrm{rel}(\mathbf{u}, \mathbf{v}) \approx \mathrm{rel}(\mathbf{q}, \mathbf{r}), \tag{8}$$

which verifies if the relation vectors are equal and which is true if the given point pairs are geometrically congruent. Similar to the definition of contact point set $\mathcal{C}$ (2) we have to handle numerical errors and noise, since real datasets never match perfectly. Therefore we only check for approximate equality ($\approx$), which can be done using the tolerance values of (2)

$$\begin{aligned} \mathrm{rel}(\mathbf{u}, \mathbf{v}) \approx \mathrm{rel}(\mathbf{q}, \mathbf{r}) \quad \Longleftrightarrow \quad & |d_{uv} - d_{qr}| < \varepsilon_p \\ & \wedge\ |\cos\alpha_{uv} - \cos\alpha_{qr}| < \varepsilon_n \\ & \wedge\ |\cos\beta_{uv} - \cos\beta_{qr}| < \varepsilon_n \\ & \wedge\ |\cos\delta_{uv} - \cos\delta_{qr}| < \varepsilon_n. \end{aligned} \tag{9}$$

Using this relation, the set of valid pose hypotheses $\mathcal{H}$ can be specified by

$$\mathcal{H} := \{(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}) \mid (\mathbf{a}, \mathbf{c}) \cong (\mathbf{b}^*, \mathbf{d}^*);\ \mathbf{a}, \mathbf{c} \in \mathcal{A};\ \mathbf{b}, \mathbf{d} \in \mathcal{B}\}. \tag{10}$$

After we have defined how to construct appropriate pose hypotheses and that we want to find the pose with maximal surface contact, the question is how to find the best hypothesis in an efficient way. If we have n surface points per fragment, we already get n^2 different point pairs per fragment and this

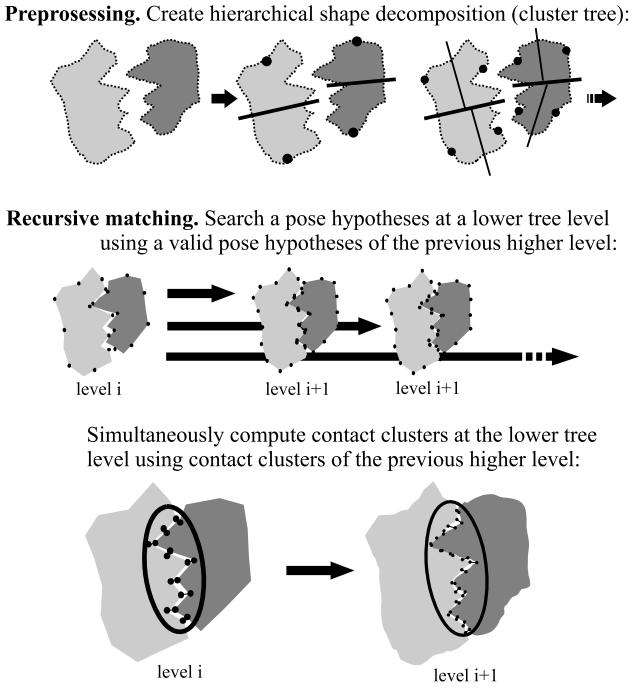


Fig. 3 Idea of the coarse-to-fine approach

leads to a reasonable amount of n^4 point pair combinations. Obviously, a naive algorithm that checks all point-pair combinations for geometrical congruency is not practical. Furthermore, we have to compare the number of contact points of every valid pose hypothesis. The classical method for finding the number of contact points is to calculate the relative transformation ${}^{A}\boldsymbol{T}_{B}$ of (3), then transform all points into a common coordinate system, and subsequently find contact point pairs. A straightforward implementation that checks every point of fragment A whether it is in tangential contact with one point of fragment B, will lead to an additional factor of n^2. However, in the following sections we will suggest an alternative approach that solves these problems in a very efficient way.

3 Top-Down Cluster Tree Matching

'Coarse-to-fine' or 'multi-resolution' strategies are well-known and common instruments in computer vision tasks. The aim is to solve the problem at a low resolution with a small amount of data and then to increase the resolution step-by-step and simultaneously refine the solution. Our matching method for 3D-puzzle-problems is also a form of coarse-to-fine approach. Figure 3 roughly illustrates the procedure. In a preprocessing step, we build a hierarchical shape decomposition called 'cluster tree'. The decomposition is based on a top-down clustering algorithm. After that, the recursive matching can be performed. The fragments at a higher tree level (represented by only a few clusters) are used to efficiently find contact hypotheses. At a high tree level it is sufficient to consider a small number of pose hypotheses and the contact computation is much faster. Now all valid high-level hypotheses can be recursively propagated down by descending the cluster trees. To compare the quality of the pose hypotheses the contact areas between both fragments are estimated. These contact areas can also be propagated from a higher to lower tree level. The hierarchical dependencies between different hierarchy levels can be used to accelerate the computation of contact areas. Furthermore, the pose hypotheses can be recursively refined. This implies that a high-level hypothesis may initiate multiple finer low-level hypotheses that are located close to the high-level one.

For efficiency reasons, not all hypotheses can be propagated to the lowest tree level (i.e. finest resolution). So we have to reject as many hypotheses as possible in order to prune the tree search early. *But the question is: how can we prevent the algorithm from rejecting coarse high-level hypotheses that might potentially turn out to be good at a lower tree level?* The answer is obvious: the computed measure of quality (e.g. the amount of contact) of a high-level hypothesis must always be a conservative upper bound of all underlying low-level hypotheses. Thus, we can prune the depth traversal whenever the upper bound of a coarse high-level hypothesis is worse than the last best match so far. Here 'last best match' denotes the hypothesis at the lowest tree level (leaf nodes of the cluster tree, where a confident contact estimation is possible). Consequently, we need a depth-first search. Furthermore, we have to consider that a high-level pose hypothesis is a representative of multiple underlying low-level hypotheses. Thus, we cannot assume that a high-level hypothesis is a single fixed pose, but rather have to allow pose tolerances. This is the reason why we cannot use simple relative transformations (i.e. homogeneous transformation matrices) for high-level pose hypotheses. This leads to an alternative 'transformation-free' contact computation.

3.1 Transformation-Free Contact Computation

In this section we will show how we can estimate the amount of contact of a pose hypothesis (at finest resolution) without using an explicit relative transformation. First, we extend the congruence relation of oriented point pairs (8) to oriented point triples

$$\begin{aligned}(\mathbf{u},\mathbf{v},\mathbf{w}) \cong (\mathbf{q},\mathbf{r},\mathbf{s}) :\Longleftrightarrow\\ |\mathbf{n}_{uv}\ \mathbf{p}_{vw}\ \mathbf{p}_{wu}| = |\mathbf{n}_{qr}\ \mathbf{p}_{rs}\ \mathbf{p}_{sq}| \ \wedge\ (\mathbf{u},\mathbf{v}) \cong (\mathbf{q},\mathbf{r})\\ \wedge(\mathbf{v},\mathbf{w}) \cong (\mathbf{r},\mathbf{s}) \ \wedge\ (\mathbf{w},\mathbf{u}) \cong (\mathbf{s},\mathbf{q}),\end{aligned} \tag{11}$$

where the first subcondition compares the relative sense of orientation to avoid mirror symmetrical solutions, and all

further subconditions verify whether the two oriented point triangles are geometrically congruent. Given the pose hypothesis $(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}) \in \mathcal{H}$, an oriented point $\mathbf{e} \in \mathcal{A}$ is in tangential contact with fragment B if there exists an oriented point $\mathbf{f} \in \mathcal{B}$ with $(\mathbf{a}, \mathbf{c}, \mathbf{e}) \cong (\mathbf{b}^*, \mathbf{d}^*, \mathbf{f}^*)$. (In this case, we can omit the second subcondition of (11), since $(\mathbf{a}, \mathbf{c}) \cong (\mathbf{b}^*, \mathbf{d}^*)$ is exactly the precondition of the pose hypothesis.) This gives us an alternative way to compute the desired amount of contact points $|\mathcal{C}|$ of (2).

Up to now, we did not care about fragment penetrations. A classical method to ensure that the fragments do not penetrate each other is to search for surface points of fragment B that are inside fragment A or vice versa. Unfortunately, this approach requires the calculation of ${}^{A}\boldsymbol{T}_{B}$. Alternatively, we suggest to search for points where the surfaces of A and B are intersecting each other. These points are very similar to contact points, except that the surfaces do not touch tangentially i.e. the surface normals are not opposed. Consequently, we can find surface intersection candidates while searching for contact points. A surface intersection occurs whenever the contact point condition $(\mathbf{a}, \mathbf{c}, \mathbf{e}) \cong (\mathbf{b}^*, \mathbf{d}^*, \mathbf{f}^*)$ fails solely as a result of misaligned surface normals $\mathbf{n}_e$ and $\mathbf{n}_f$. Since this intersection detection might be unstable in the presence of noise, we do not reject the hypothesis but subtract some quality penalty for each intersecting point.

3.2 Cluster Tree Construction

Due to the combinatorial explosion of all possible contact pairs and pose hypotheses, the theoretical concept of the last section is far from being suitable for an efficient implementation. To overcome this problem, we decompose each point set into a binary tree structure, which allows to apply an efficient tree matching strategy. The subdivision of the point sets is based on a hierarchical divisive (top-down) clustering algorithm. A common practice is to cluster the set of 3D point coordinates $\mathcal{P}_A$ and $\mathcal{P}_B$. Instead of that, we suggest the clustering of coordinates and surface normals simultaneously in the combined 6D-coordinate-normal space defined in (1). The clustering algorithm works as follows:

1. Scaling of all surface normals, in such a way that their variance roughly conforms to the coordinate variance.
2. Recursive splitting of each point set into two subsets.

The key point of our proposed matching algorithm is, that it does not depend on a stable/unique clustering. The approach does not rely on a similar decomposition of both fragments. Thus every fragment can be decomposed differently and independently from the others. Therefore, we apply a simple splitting rule that is based on the well-known *k-means clustering* approach (Hartigan and Wong 1978) in 6D-space with $k = 2$. Thus, the points are divided into clusters of similar location and surface orientation and rapidly

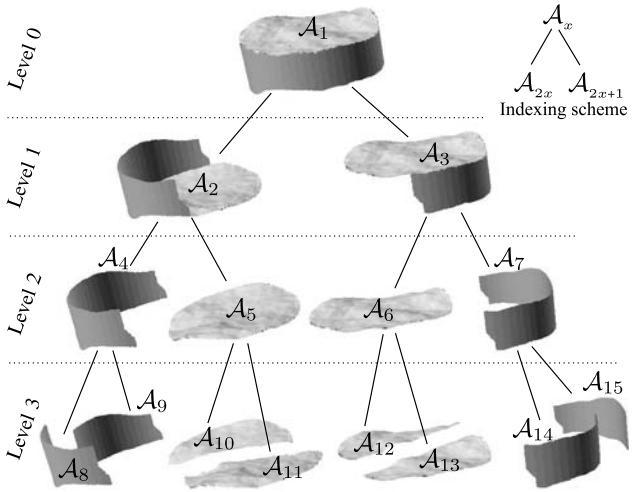


Fig. 4 Cluster tree example: first levels of the tree decomposition applied to a simple test fragment

decrease in size (both in coordinate and in orientation space) when descending the tree. The 6D splitting scheme is particularly favorable because the coordinate variance, as well as the normal orientation variance of each cluster are a crucial factor for the pose tolerance of a *high-level hypothesis* (will be discussed in Sect. 3) and consequently affect the efficiency of the whole matching approach. An example is given in Fig. 4, which shows the hierarchical clustering applied to a simple test fragment. As can be seen, our approach divides the set of oriented points into subsets of similar orientations and preferably splits along sharp edges. In the following the cluster tree nodes of fragment A are denoted as $\mathcal{A}_1, \ldots, \mathcal{A}_k$ with root $\mathcal{A}_1 := \mathcal{A}$ and conforming to the rule that each cluster node $\mathcal{A}_x$ is split into two children $\mathcal{A}_{2x}$ and $\mathcal{A}_{2x+1}$. The same applies to the cluster tree nodes $\mathcal{B}_1, \ldots, \mathcal{B}_l$ of fragment B.

3.3 Matching of Cluster Trees

After generating the cluster trees up to a certain level, the hierarchical matching can be performed. Our suggested matching approach is similar to the theoretical concept discussed so far, but instead of using individual oriented points, we now operate on oriented point clusters. Let us start with some definitions on cluster pairs: a tangential contact between two clusters $\mathcal{A}_a$ and $\mathcal{B}_b$ implies that there exists at least one tangential contact between two oriented points $\mathbf{a} \in \mathcal{A}_a$ and $\mathbf{b} \in \mathcal{B}_b$. We can construct a high-level pose hypothesis by assuming a tangential contact between some clusters of each fragment. More precisely, a high-level pose hypothesis $(\mathcal{A}_a, \mathcal{B}_b, \mathcal{A}_c, \mathcal{B}_d)$ is the assumption of a tangential contact between the clusters $\mathcal{A}_a$ and $\mathcal{B}_b$ as well as between the clusters $\mathcal{A}_c$ and $\mathcal{B}_d$. A tangential contact between two cluster pairs is only possible if their relative distances

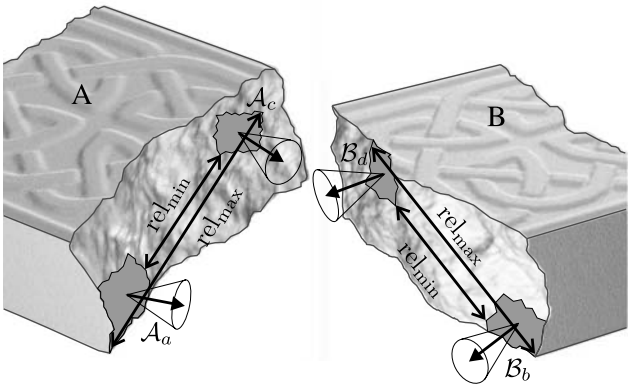


Fig. 5 A simplified illustration of two cluster pairs with overlapping relation intervals. A high-level pose hypothesis implies the assumption of a tangential contact between both pairs. Note that $\mathrm{rel}_{\min}$ and $\mathrm{rel}_{\max}$ define 4D relation intervals, but for purposes of clarity only the minimal and maximal cluster distance is visualized

and angles have overlapping intervals. To verify this constraint, we follow the idea of (8), where we compared 4D relation vectors of oriented point pairs. But instead of oriented point pairs, we now compare 4D relation intervals of cluster pairs. A simplified illustration of two cluster pairs with overlapping relation intervals if given in Fig. 5. Consider min/max operators that are defined element-wise on vector sets

$$\min\left\{\begin{bmatrix}x_1\\y_1\\\vdots\end{bmatrix},\ldots,\begin{bmatrix}x_n\\y_n\\\vdots\end{bmatrix}\right\}:=\begin{bmatrix}\min\{x_1,\ldots,x_n\}\\\min\{y_1,\ldots,y_n\}\\\vdots\end{bmatrix} \tag{12}$$

and $\max\{\ldots\}$ analogous. Then, we can define the relation interval $[\mathrm{rel}_{\min}(\mathcal{A}_a,\mathcal{A}_c),\mathrm{rel}_{\max}(\mathcal{A}_a,\mathcal{A}_c)]$ of a cluster pair $(\mathcal{A}_a,\mathcal{A}_c)$ by

$$\begin{aligned}\mathrm{rel}_{\min}(\mathcal{A}_a,\mathcal{A}_c)&:=\min\{\mathrm{rel}(\mathbf{u},\mathbf{v})\mid\mathbf{u}\in\mathcal{A}_a,\mathbf{v}\in\mathcal{A}_c\},\\ \mathrm{rel}_{\max}(\mathcal{A}_a,\mathcal{A}_c)&:=\max\{\mathrm{rel}(\mathbf{u},\mathbf{v})\mid\mathbf{u}\in\mathcal{A}_a,\mathbf{v}\in\mathcal{A}_c\}.\end{aligned} \tag{13}$$

However, the recursive definition of all relation intervals (between clusters at the same tree level)

$$\mathrm{rel}_{\min}(\mathcal{A}_a,\mathcal{A}_c):=\begin{cases}\mathrm{rel}(\mathbf{u},\mathbf{v}) \text{ if } \mathcal{A}_a=\{\mathbf{u}\} \text{ and } \mathcal{A}_c=\{\mathbf{v}\} \text{ are singletons,}\\ \min\{\mathrm{rel}_{\min}(\mathcal{A}_x,\mathcal{A}_y)\mid\lfloor x/2\rfloor=a,\lfloor y/2\rfloor=c\} \text{ else,}\end{cases} \tag{14}$$

is computationally more efficient. The condition $\lfloor x/2\rfloor=a$ just follows the indexing scheme of Sect. 3.2 and stands for "cluster $\mathcal{A}_x$ is a child node of cluster $\mathcal{A}_a$". Now consider comparison operators on relation vectors that again are applied element-wise

$$\begin{bmatrix}x_1\\\vdots\\x_n\end{bmatrix}\lesseqgtr\begin{bmatrix}y_1\\\vdots\\y_n\end{bmatrix}:\Leftrightarrow x_i\lesseqgtr y_i\quad\text{for all } i=1,\ldots,n. \tag{15}$$

This enables us to compare the relation intervals and to set up a congruence relation between cluster pairs

$$\begin{aligned}(\mathcal{A}_u,\mathcal{A}_v)\cong(\mathcal{B}_q^*,\mathcal{B}_r^*):\Leftrightarrow\\ \mathrm{rel}_{\min}(\mathcal{A}_u,\mathcal{A}_v)\le\mathrm{rel}_{\max}(\mathcal{B}_q^*,\mathcal{B}_r^*)\wedge\mathrm{rel}_{\max}(\mathcal{A}_u,\mathcal{A}_v)\\ \ge\mathrm{rel}_{\min}(\mathcal{B}_q^*,\mathcal{B}_r^*),\end{aligned}$$

which holds if the intervals overlap and if a tangential contact is possible. An essential observation is, that this congruence relation on cluster pairs becomes equivalent to the congruence relation (8) on oriented point pairs when the cluster size converges to zero. It follows that the set of valid high-level pose hypotheses $\widehat{\mathcal{H}}$ can be specified similarly to the set of (low-level) pose hypotheses (10) by

$$\widehat{\mathcal{H}}:=\{(\mathcal{A}_a,\mathcal{B}_b,\mathcal{A}_c,\mathcal{B}_d)\mid(\mathcal{A}_a,\mathcal{A}_c)\cong(\mathcal{B}_b^*,\mathcal{B}_d^*)\}. \tag{16}$$

Now, all valid high-level hypotheses can be propagated down by descending the cluster trees. It is straightforward to show that an invalid hypothesis at level i implies that all hypotheses of their child clusters at level $i+1$ are invalid too (or in other words, if two cluster pairs can't be put into contact, then their children can't be put into contact either). This fact provides the opportunity to reject many hypotheses long before the depth-first search reaches a leaf.

To estimate the quality of a high-level pose hypothesis, we can use the approach suggested for low-level hypotheses (11) and extend the congruence relation of cluster pairs to cluster triples

$$\begin{aligned}(\mathcal{A}_u,\mathcal{A}_v,\mathcal{A}_w)\cong(\mathcal{B}_q^*,\mathcal{B}_r^*,\mathcal{B}_s^*):\Leftrightarrow\\ (\mathcal{A}_u,\mathcal{A}_v)\cong(\mathcal{B}_q^*,\mathcal{B}_r^*)\wedge(\mathcal{A}_v,\mathcal{A}_w)\\ \cong(\mathcal{B}_r^*,\mathcal{B}_s^*)\wedge(\mathcal{A}_w,\mathcal{A}_u)\cong(\mathcal{B}_s^*,\mathcal{B}_q^*).\end{aligned}$$

Note that this definition does not care about the relative sense of orientation and consequently accepts some mirror symmetrical solutions. But this property is not a drawback since mirror symmetrical solutions are sparse. In addition, they can be rejected by means of (11) as soon as the traversal reaches the leaf nodes. Given the high-level pose hypothesis $(\mathcal{A}_a,\mathcal{B}_b,\mathcal{A}_c,\mathcal{B}_d)$, a cluster $\mathcal{A}_e$ is potentially in contact with fragment B if there exists a cluster $\mathcal{B}_f$ at the same tree level with $(\mathcal{A}_a,\mathcal{A}_c,\mathcal{A}_e)\cong(\mathcal{B}_b^*,\mathcal{B}_d^*,\mathcal{B}_f^*)$. Once again, we can utilize the fact that a cluster pair can only be in contact if their parent clusters are in contact. The ratio of contact area to the total surface area serves as quality criterion. If the upper bound of the quality of a high-level hypothesis is worse than the last best match, we truncate the tree descent and continue to traverse the next higher branch.

Let us recapitulate and concretize the entire algorithm: In a preprocessing step (see Algorithm 1 lines 1–7) we build up

Algorithm 1 Preprocessing and Recursive Matching

1: Build a cluster tree for fragment A;
2: **for all** node combinations $\mathcal{A}_i$,$\mathcal{A}_j$ at same tree level **do**
3: Store $\mathrm{rel}_{\min}(\mathcal{A}_i, \mathcal{A}_j)$, $\mathrm{rel}_{\max}(\mathcal{A}_i, \mathcal{A}_j)$ in a matrix;
4: Repeat step 1–3 for fragment B;
5: $C^{(0)} \leftarrow \{(\mathcal{A}_1, \mathcal{B}_1)\}$; ▷ init contact list
6: $bestQuality \leftarrow 0$;
7: MATCH$(1, 1, 1, 1, 0)$; ▷ start recursion at root nodes
8: **procedure** MATCH(a, b, c, d, i)
9: **if** $a > b$ **then return;** ▷ avoids symmetrical hypotheses
10: **if** $(\mathcal{A}_a, \mathcal{B}_b, \mathcal{A}_c, \mathcal{B}_d) \notin \widehat{\mathcal{H}}$ **then return;** ▷ incongruent pair
11: $actQuality \leftarrow$ BUILDCONTACTLIST(a, b, c, d, i);
12: **if** $actQuality \leq bestQuality$ **then return;**
13: **if** i is maximal recursion depth **then**
14: $bestQuality \leftarrow actQuality$,
15: $bestHypothesis \leftarrow (\mathcal{A}_a, \mathcal{B}_b, \mathcal{A}_c, \mathcal{B}_d)$; ▷ memorize best
16: **else**
17: **for** $u \leftarrow 0 \ldots 1$ **for** $v \leftarrow 0 \ldots 1$ **do**
18: **if** i is odd **then**
19: MATCH$(a, 2b{+}u, c, 2d{+}v, i{+}1)$; ▷ decend tree of B
20: **else** MATCH$(2a{+}u, b, 2c{+}v, 2d, i{+}1)$; ▷ decend tree of A
21: **function** BUILDCONTACTLIST(a, b, c, d, i)
22: **if** $i = 0$ **then return** 100%;
23: $C^{(i)} \leftarrow \emptyset$; ▷ clear contact list of recursion depth i
24: **for all** $(\mathcal{A}_e, \mathcal{B}_f) \in C^{(i-1)}$ **do** ▷ for all pairs of list i-1
25: **for** $u \leftarrow 0$ to 1 **do**
26: **if** i is odd $\wedge$ $(\mathcal{A}_a, \mathcal{A}_c, \mathcal{A}_e) \cong (\mathcal{B}^*_b, \mathcal{B}^*_d, \mathcal{B}^*_{2f+u})$ **then**
27: $C^{(i)} \leftarrow C^{(i)} \cup \{(\mathcal{A}_e, \mathcal{B}_{2f+u})\}$; ▷ add contact pair
28: **if** i is even $\wedge$ $(\mathcal{A}_a, \mathcal{A}_c, \mathcal{A}_{2e+u}) \cong (\mathcal{B}^*_b, \mathcal{B}^*_d, \mathcal{B}^*_f)$ **then**
29: $C^{(i)} \leftarrow C^{(i)} \cup \{(\mathcal{A}_{2e+u}, \mathcal{B}_f)\}$; ▷ add contact pair
30: **return** percent of contact at level i; ▷ sum of surface areas in contact list

a cluster tree for each fragment and use the recursive interval definition (14) to store the relation intervals of each cluster pair in a matrix. We initialize a list of contact pairs $C^{(0)}$ with the root node pair and start the matching algorithm with $(\mathcal{A}_1, \mathcal{B}_1, \mathcal{A}_1, \mathcal{B}_1)$ as start hypothesis. The recursive matching procedure (lines 8–20) verifies whether the contact hypothesis is valid, estimates its contact quality, and descends the trees alternately if the actual quality is potentially better than the last best match. The contact quality (percentage of surface area in contact) is estimated by iterating the list of contact pairs C from one tree level to the next (lines 21–30). The total contact area correspond to the sum of surface area of clusters in the contact list $C^{(i)}$ at tree level i. The surface area of each cluster can be precalculated and stored simultaneous with the cluster tree construction.

4 Further Constraints and Improvements

Suppose we have n interface points between fragments A and B. Since almost every pair of these points of fragment A can be brought into contact with one corresponding pair at fragment B, we obtain up to n^2 valid hypotheses that all result in nearly the same pose. To reduce this huge amount of redundant hypotheses, we can constrain the contact pair search to a small subset of uniformly distributed surface points on fragment A. This reduction can be easily integrated by choosing a tree level at which the hypothesis iteration is constricted to the left tree branch of fragment A. Note that this hypothesis reduction does not influence the quality estimation, as we keep using all points for contact pair computation. Furthermore, we can shrink the cluster sizes slightly (particularly within the surface normal space) to eliminate outliers and to achieve an implicit smoothing, which will speedup the algorithm and additionally increases the accuracy in the presence of noise. On the one hand these extensions can significantly increase the performance, but on the other hand the global optimum is not guaranteed to be found any longer. Up to this point, we did not use any shape knowledge about the object. However, in many applications, additional information about the parts to be combined is available. This includes the knowledge

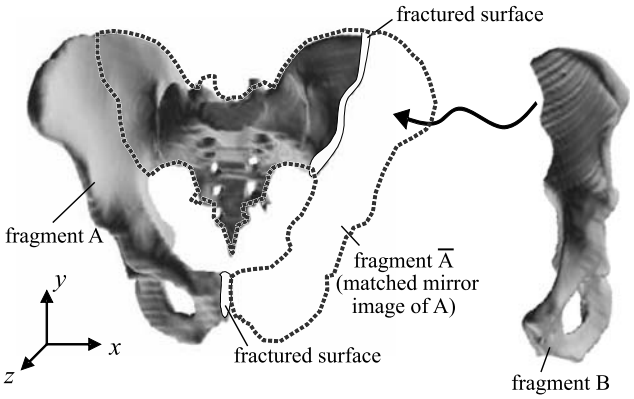


Fig. 6 The knowledge of symmetry as prior to match a broken pelvis

of axes (e.g. Winkelbach et al. 2003), symmetries, geometric features like sharp curvature transitions from intact surfaces to broken ones (e.g. Papaioannou and Theoharis 1999), or knowledge about the roughness of the fracture surface (e.g. Huang et al. 2006). An integration of these constraints into our approach is straightforward and can drastically limit the search space and thus reduce execution time. Figure 6 illustrates the use of knowledge about mirror symmetry to assemble mirror-symmetrical objects like a broken pelvis. First, we make a mirror symmetrical copy $\overline{A}$ of the larger fragment A. Then we invert the normals of fragment $\overline{A}$ and match it with fragment A using our matching approach. After that both fragments complement one another to a complete pelvis. Thus, we can compute the plane of symmetry and we get a 'virtual container', where the remaining fragment B must be fitted in (which can be done using the same matching approach). The result is a roughly assembled pelvis. Since human pelvises are not perfectly symmetrical, the symmetry-based rough alignment of the pelvic fracture is not accurate enough. This is particularly relevant if the fracture involve the sacral foramina (opening for sacral nerves and arteries). However, the problem can be tackled by a subsequent fine registration, where the rough alignment serves as basis for identifying the fracture surface. A detailed discussion about matching of pelvic and femoral fractures will be a subject of further publications.

5 Experimental Results and Conclusion

We have evaluated our new matching approach by means of point sampled surfaces of many objects with artificial and real fractures. Figure 7 shows ten representative test examples. The first two examples (a–b) are femoral (thigh bone) fractures, which have been extracted from computer tomographic scans. The matching of bone fragments is a very important field of application in computer aided fracture reduction in surgery. A robust and fast matching method promises better reposition results, a reduction of radiation exposure of surgeons, and a reduction of surgery time and cost (Winkelbach et al. 2003). The third example (c) are two fragments of a broken plaster cake, which are courtesy by Vienna University of Technology. Figure 8 shows a detail view of the matched plaster cake fragments compared to the rough alignment of the original data set. Two further data sets (d–e) are artificially broken 3D models. To provide realistic test conditions, we distorted the surface point coordinates of these fragments with additive white Gaussian noise (with an amplitude of 0.1% relative to the maximal object diameter). The examples (f–g) show combinable assembly components, which suggest the application of our approach in the field of assembly assistance and computer aided design. Example (h) is a pelvic fracture, which is also extracted from a computer tomographic scan. In the unconstraint case the algorithm finds the solution with the largest contact area, which is not the desired one. This occurs if the portion of the fracture interface is small in comparison to the total surface area, and if the intact object surface includes large smooth areas with complementary regions on the counterpart. Therefore, we apply the symmetry constraints of Sect. 4 to match the broken pelvis. Data set (i) illustrates the ability of the algorithm to match partially overlapping surfaces (in this case to surfaces of a Beethoven bust, which have been digitized from two different viewing directions using a simple low-cost laser scanner Winkelbach et al. 2006). And the last example (j) shows a 3D scan of overlapping assembly components. By using our approach to fit CAD-models into the scan, we can simply estimate the 3D pose of the objects. This example demonstrates the capability of our algorithm to solve the well-known bin-picking-problem (a robot that should grasp 3D objects in a bin).

Table 1 Performance evaluation: execution time (*time*) in seconds at tree level 12, angular inaccuracy (*rot.*), and translational inaccuracy (*trans.*) in percent w.r.t. the objects diameter

	time	*rot.*	*trans.*
Simple Fracture	3.26	2.48°	0.92%
Spiral Fracture	1.32	4.87°	2.35%
Plaster Cake	2.92	1.12°	2.07%
Venus A + B	2.62	1.55°	0.85%
Venus B + C	4.48	2.04°	0.74%
Bunny A + B	7.29	3.63°	0.71%
Bunny A + C	4.70	2.92°	0.95%
Connector Pair	5.45	1.87°	1.30%
Car Light	3.22	2.83°	1.52%
Pelvis	8.82	3.34°	3.61%
Beethoven	0.19	3.49°	1.90%

An evaluation of run-times and matching accuracies is given in Table 1. Table 2 shows the preprocessing time

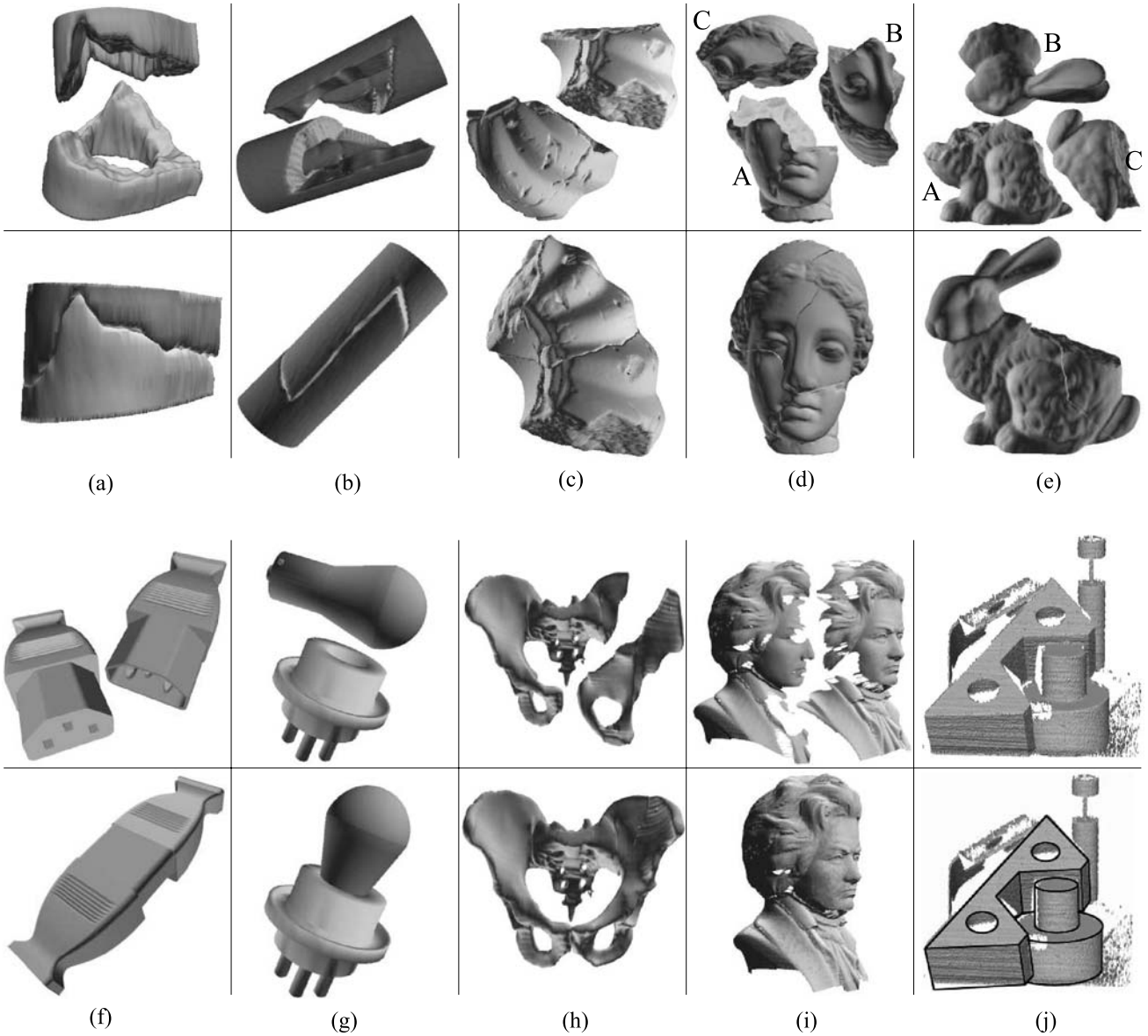


Fig. 7 Some test fragments in initial pose and their pose after matching: (**a**) simple thigh bone fracture; (**b**) spiral fracture; (**c**) fragments of a broken plaster cake; (**d**) broken Venus; (**e**) broken bunny; (**f**) IEC-320 power connector; (**g**) car light components; (**h**) pelvic fracture; (**i**) 3D scans of a Beethoven bust; (**j**) 3D scan of overlapping assembly components and matched CAD-polyhedrons

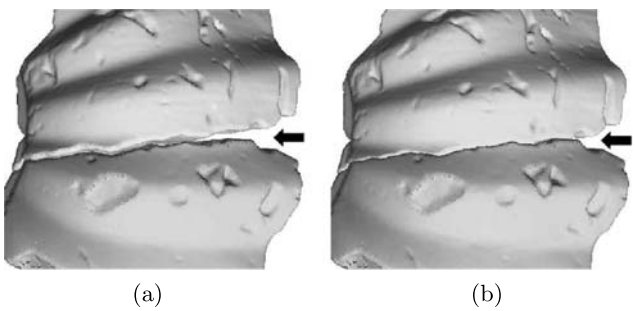


Fig. 8 Detail view of the matched plaster cake fragments: (**a**) rough alignment of the original data set provided by Vienna University of Technology; (**b**) matching result of our approach

for generating a cluster tree and its corresponding relation interval matrix. All tests were performed on an AMD Athlon 64/2.2 GHz based PC. For all test cases, adequate matching results can be achieved in less than nine seconds. The results of four case studies are plotted in Fig. 9. As can be seen, the execution time increases nearly linearly with the number of points. The accuracy is high, as long as the sampling rate lies above a certain threshold (below this threshold the matching fails due to an insufficient surface approximation). The approach performs very well with a variety of objects (3D fragments as well as industrial components).

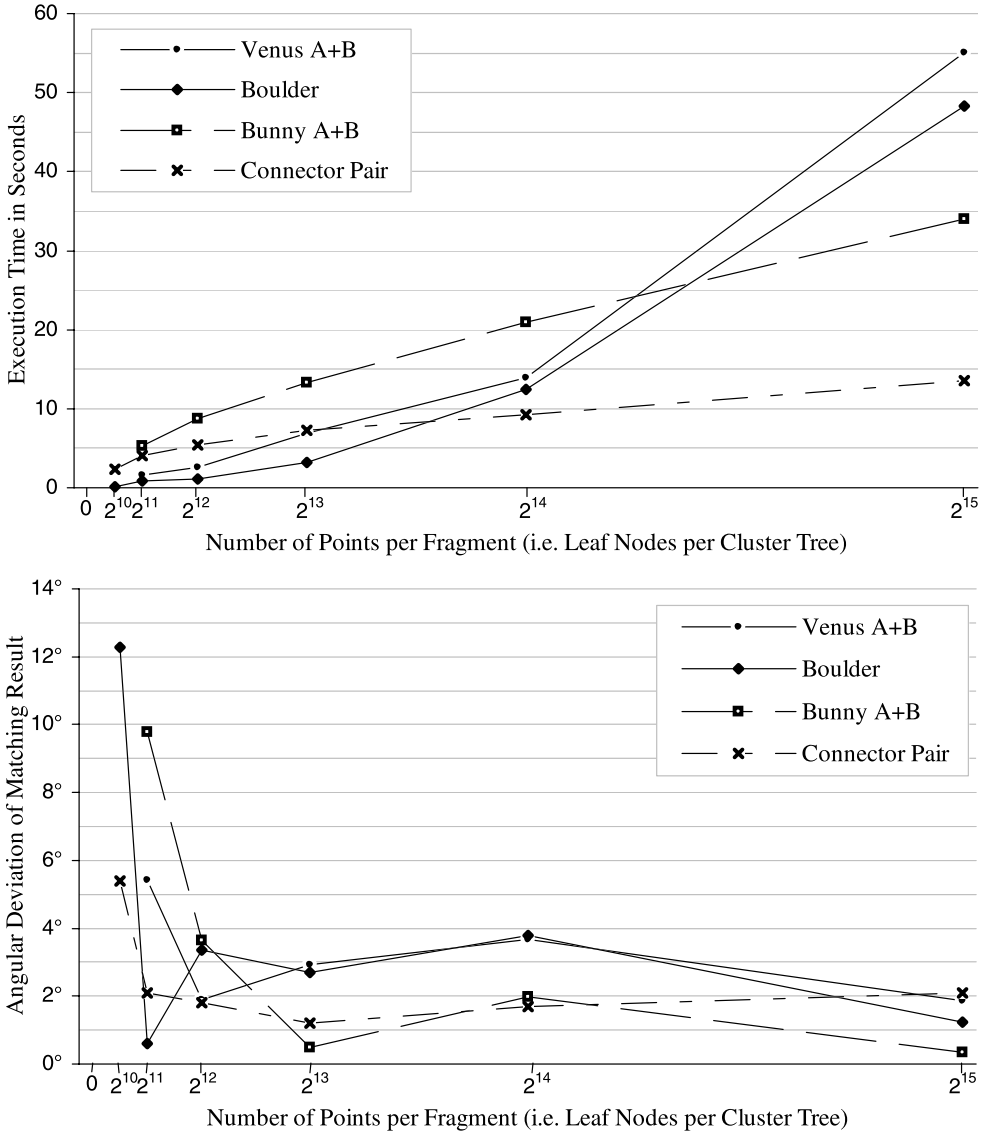


Fig. 9 Matching results of four examples in relation to the number of surface points (i.e. cluster tree depth)

Table 2 Preprocessing time (generation of one cluster tree and corresponding relation interval matrix)

Tree depth	10	11	12	13	14	15
Time in seconds	0.86	0.96	1.10	1.39	2.23	5.06

To analyze the effect of noise we applied the matching algorithm on an artificial broken bolder with different noise level. Figure 10 shows the broken boulder with no noise (a), and with 100% noise level (b). Here 100% noise level stands for an additive Gaussian noise with a standard deviation that equals the average distance of adjacent surface points. Table 3 shows an evaluation of run-times and matching accuracies. As expected, the matching accuracy decreases with the magnitude of noise. But the algorithm always finds a solution that is close to the desired one, which confirms the robustness against noise. Furthermore it can be seen that the execution time increases with the magnitude of noise. This can be explained by the fact that the coordinate and normal

Table 3 Effect of a varying noise level (a noise level of 100% corresponds to an additive Gaussian noise with a standard deviation that equals the average distance of adjacent surface points): execution time (*time*) in seconds at tree level 13, angular inaccuracy (*rot.*), and translational inaccuracy (*trans.*) in percent w.r.t. the objects diameter

Noise level	*time*	*rot.*	*trans.*
0%	2.83	2.70°	1.15%
10%	3.53	2.89°	0.68%
20%	4.11	3.74°	2.32%
30%	6.44	3.83°	2.08%
40%	7.85	4.83°	3.27%
50%	14.20	5.27°	0.61%
100%	95.78	8.20°	2.17%

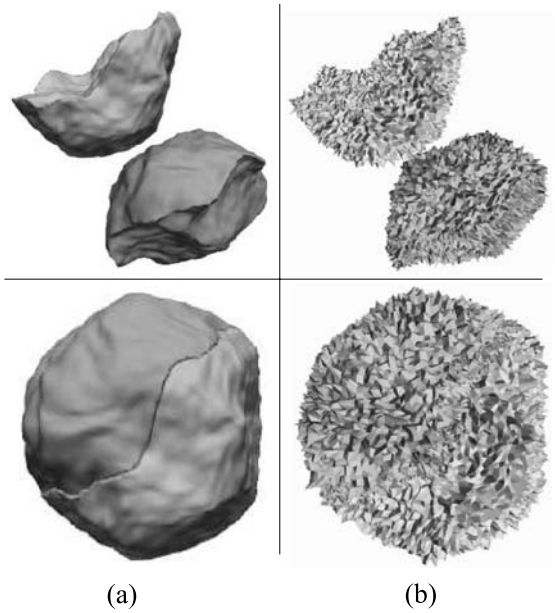


Fig. 10 Broken boulder in initial pose and pose after matching: (**a**) no noise; (**b**) 100% noise level (additive Gaussian noise with a standard deviation that equals the average distance of adjacent surface points)

variances of each 'noisy cluster' is higher, hence rejection of worse or invalid pose hypothesis must take place at a lower tree level.

Still open is the problem how to extend our approach with regard to matching of multiple fragments. It is obvious, that we can match multiple fragments by a pair-wise combination of two fragments (as shown), but the result may not be optimal in a global sense. One possible solution is a subsequent global pose optimization. Papaioannou et al. have presented a solution to this problem using global optimization of pairwise matching (Papaioannou et al. 2000). However, the outstanding efficiency of our matching approach can be regarded as an essential step towards an efficient multi-fragment matching.

References

Barequet, G., & Sharir, M. (1996). Partial surface matching by using directed footprints. In *Proc. of the 12th annual symposium on computational geometry* (pp. 409–410). New York: ACM Press.

Besl, P. J., & McKay, N. D. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*(2), 239–258.

Chua, C. S., & Jarvis, R. (1997). Point signatures: A new representation for 3D object recognition. *International Journal of Computer Vision*, *25*(1), 63–85.

Cooper, D., Willis, A., Andrews, S., Baker, J., Cao, Y., Han, D., Kang, K., Kong, W., Leymarie, F. F., Orriols, X., et al. (2002). Bayesian pot-assembly from fragments as problems in perceptual-grouping and geometric-learning. *International Conference on Pattern Recognition*, *16*(3), 297–302.

da Gama Leitão, H. C., & Stolfi, J. (2002). A multiscale method for the reassembly of two-dimensional fragmented objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(9), 1239–1251.

Dalley, G., & Flynn, P. (2002). Pair-wise range image registration: a study in outlier classification. *Computer Vision and Image Understanding*, *87*(1-3), 104–115.

Ericson, C. (2005). *Real-time collision detection*. *Kaufmann series in interactive 3D technology*. Amsterdam: Elsevier.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381–395.

Goldberg, D., Malon, C., & Bern, M. (2004). A global approach to automatic solution of jigsaw puzzles. *Computational Geometry*, *28*(2–3), 165–174.

Hartigan, J. A., & Wong, M. A. (1978). A k-means clustering algorithm. *Applied Statistics*, *28*, 100–108.

Huang, Q.-X., Flöry, S., Gelfand, N., Hofer, M., & Pottmann, H. (2006). Reassembling fractured objects by geometric matching. In *SIGGRAPH '06* (pp. 569–578). New York: ACM Press.

Johnson, A., & Hebert, M. (1997). Recognizing objects by matching oriented points. In *Proc. IEEE conf. computer vision and pattern recognition (CVPR'97)* (pp. 684–689).

Kampel, M., & Sablatnig, R. (2003a). Profile-based pottery reconstruction. *Conference on Computer Vision and Pattern Recognition Workshop*, *1*, 4.

Kampel, M., & Sablatnig, R. (2003b). Virtual reconstruction of broken and unbroken pottery. In *International conference on 3-D digital imaging and modeling (3DIM)* (pp. 318–325).

Krsek, P., Pajdla, T., & Hlaváč, V. (2002). Differential invariants as the base of triangulated surface registration. *Computer Vision and Image Understanding*, *87*, 27–38.

Linnainmaa, S., Harwood, D., & Davis, L. S. (1988). Pose determination of a three-dimensional object using triangle pairs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *10*(5), 634–647.

Mitra, N. J., & Nguyen, A. (2003). Estimating surface normals in noisy point cloud data. In *SCG '03: Proceedings of the nineteenth annual symposium on computational geometry* (pp. 322–328).

Papaioannou, G., & Theoharis, T. (1999). Fast fragment assemblage using boundary line and surface matching. In *IEEE/CVPR workshop on applicat. of computer vision in archaeology*.

Papaioannou, G., Karabassi, A., & Theoharis, T. (2000). Automatic reconstruction of archaeological finds—a graphics approach. In *Proc. 4th int. conf. computer graphics and artificial intell.* (pp. 117–125).

Papaioannou, G., Karabassi, E., & Theoharis, T. (2002). Reconstruction of three-dimensional objects through matching of their parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(1), 114–124.

Pottmann, H., Huang, Q.-X., Yang, Y.-L., & Hu, S.-M. (2006). Geometry and convergence analysis of algorithms for registration of 3D shapes. *International Journal of Computer Vision*, *67*(3), 277–296.

Rusinkiewicz, S., & Levoy, M. (2001). Efficient variants of the icp algorithm. In *International conference on 3D digital imaging and modeling (3DIM 2001)* (pp. 145–152).

Sappa, A., Restrepo-Specht, A., & Devy, M. (2001). Range image registration by using an edge-based representation. In *International symposium on intelligent robotic systems (SIRS'01)* (pp. 167–176).

Schön, N., & Häusler, G. (2005). Automatic coarse registration of 3D surfaces. In *Vision, modeling, and visualization 2005* (pp. 71–178).

Seeger, S., & Labourex, X. (2000). Feature extraction and registration: An overview. In *Principles of 3D image analysis and synthesis* (pp. 153–166).

Stockman, G. (1987). Object recognition and localization via pose clustering. *Computer Vision, Graphics, and Image Processing*, *40*(3), 361–387.

Vanden Wyngaerd, J., & Van Gool, L. (2002). Automatic crude patch registration: Toward automatic 3D model building. *Computer Vision and Image Understanding*, *87*, 8–26.

Vienna University of Technology. 3D Puzzles—reassembling fractured objects by geometric matching. www.geometrie.tuwien.ac.at/geom/ig/3dpuzzles.html, undated.

Wahl, E., Hillenbrand, U., & Hirzinger, G. (2003). Surflet-pair-relation histograms: A statistical 3D-shape representation for rapid classification. In *Proc. 4th international conf. on 3-D digital imaging and modeling (3DIM'03)* (pp. 474–481). IEEE Computer Society Press.

Willis, A., & Cooper, D. (2004). Bayesian assembly of 3D axially symmetric shapes from fragments. *IEEE Conference on Computer Vision and Pattern Recognition*, *1*, 82–89.

Winkelbach, S., Westphal, R., & Gösling, T. (2003). Pose estimation of cylindrical fragments for semi-automatic bone fracture reduction. In B. Michaelis & G. Krell (Eds.), *Lecture Notes in Computer Science: Vol. 2781. Pattern recognition, 25th DAGM symposium* (pp. 566–573). Berlin: Springer.

Winkelbach, S., Rilk, M., Schönfelder, C., & Wahl, F. M. (2004). Fast random sample matching of 3D fragments. In C. E. Rasmussen, H. H. Bülthoff, H. H. Giese, & B. Schölkopf (Eds.), *Lecture Notes in Computer Science: Vol. 3175. Pattern recognition, 26th DAGM symposium* (pp. 129–136). Berlin: Springer.

Winkelbach, S., Molkenstruck, S., & Wahl, F. (2006). Low-cost laser range scanner and fast surface registration approach. In K. Franke, K.-R. Müller, B. Nickolay, & R. Schäfer (Eds.), *Lecture Notes in Computer Science: Vol. 4174. Pattern recognition, 28th DAGM symposium* (pp. 718–728). Berlin: Springer.