REGISTRATION OF 3D - PATTERNS AND SHAPES WITH CHARACTERISTIC POINTS

Darko Dimitrov, Christian Knauer and Klaus Kriegel
Freie Universität Berlin, Institute of Computer Science
Takustrasse 9, D-14195 Berlin, Germany

Keywords: Computational Geometry, Pattern Matching, 3D Image Registration.

Abstract: We study approximation algorithms for a matching problem that is motivated by medical applications. Given a small set of points $P \subset \mathbb{R}^3$ and a surface S, the optimal matching of P with S is represented by a rigid transformation which maps P as 'close as possible' to S. Previous solutions either require polynomial runtime of high degree or they make use of heuristic techniques which could be trapped in some local minimum. We propose a modification of the problem setting by introducing small subsets of so called characteristic points $P_c \subseteq P$ and $S_c \subseteq S$, and assuming that points from P_c must be matched with points from S_c. We focus our attention on the first nontrivial case that occurs if $|P_c| = 2$, and show that this restriction results in new fast and reliable algorithms for the matching problem. In contrast to heuristic approaches our algorithm provides guarantees on the approximation factor of the matching. Experimental results are provided for surfaces reconstructed from real and synthetic data.

1 MOTIVATION AND RELATED WORK

Today an increasing number of surgeries is supported by medical navigation systems. The basic task of such a system is to transform real world data (positions in the operating field) into a 3-dimensional model (CT or MR) and to display the transformed position in the model. Real world data are gaged by optical, electromagnetic or mechanical tracking systems. A common technique for computing the transformation is based on markers which are fixed on bones (landmark approach). The markers have to be fixed already during the model acquisition. Their positions in the model are computed using appropriate image processing methods. Later, at the beginning of the surgery, at least three markers must be gaged with the tracking system. Since the total number of markers is small, one could compute the correct matching transformation even by brute force techniques. A more advanced approach making use of geometric hashing techniques is presented in (Hoffmann et al., 1999).

There is a strong need to develop algorithmic methods for computing a transformation without using markers. The main reason for that is an anatomical one: in many cases (e.g. spinal surgery) it would be very hard or even impossible to fix markers before the surgery. One solution is to gage a few points on the surface of a bone and to compute the corresponding points in the model. This point registration is a hard algorithmic problem, which cannot be solved by the following standard approaches:
1) A combinatorial search does not work, because the gaged points could be anywhere on the model surface.
2) Surface matching algorithms do not help, because the number of gaged points is too small to apply a surface reconstruction.

There are heuristic methods that could be used, for example a combination of ICP (Iterative Closest Point) method with randomly generated starting configurations. The standard ICP approach, and several more efficient variants and generalizations of it can be found in (Besl and McKay, 1992), (Rusinkiewicz and Levoy, 2001), and (Mitra et al., 2004). However such heuristic methods have the fundamental drawback that they could be trapped in some local minimum, while failing to find the optimal solution. In contrast to this, we will present a registration algorithm that reports all possible registration transformations satisfying a given quality criterion. This is an essential property, especially in medical applications.

Matching problems have been studied intensively

Dimitrov D., Knauer C. and Kriegel K. (2006).
REGISTRATION OF 3D - PATTERNS AND SHAPES WITH CHARACTERISTIC POINTS.
In *Proceedings of the First International Conference on Computer Vision Theory and Applications*, pages 393-400
DOI: 10.5220/0001368303930400

in computational geometry, see (Alt and Guibas, 1999) for an overview. Nevertheless the best known approaches applicable to our problem require polynomial runtime of high degree (Hagedoorn, 2000). Since the registration is part of the surgery, real time algorithms are needed. In contrast to that, it is possible to spend more time for preprocessing the model.

Here, we try to adapt some ideas of the landmark approach to that new setting. The role of markers is played by so-called characteristic points. Such points can be determined automatically (see next two paragraphs) or manually, marking some anatomically significant points, e.g. the thorn of a vertebra or the root of the nose. If a set of characteristic points is fixed in the model and we can track at least three of them, the old landmark registration algorithms can be applied. Our main goal is to solve the registration problem if only two characteristic points can be tracked. To compute the transformation in that case, one must track some more (non-characteristic) points on the surface.

Different approaches can be applied to find characteristic points on the sampled surface. A lot of effort was devoted on determining the curvature estimation. A comprehensive survey of these methods is given in (Petitjean, 2002). Popular methods include quadric fitting, where the estimated curvature is the one of the quadric that best fits the sampled point locally. Other methods typically consider some definition of curvature that can be extended to the polyhedral setting. Estimating the curvature by a canonical mapping between the mesh describing the object and a standard spherical mesh was done in (Delingette et al., 1993). More examples of estimating the discrete curvature can be found in (Zhang, 1999), and (Kim et al., 1999). The definition of the curvature tensor in (Cohen-Steiner and Morvan, 2003) is based on the theory of normal cycles.

Sometimes technics to characterize points of the surface are strongly related with surface matching heuristics, e.g. the spin image approach in (Johnson and Hebert, 1997), surface point signatures in (Yamany and Farag, 2002), harmonic shape images in (Zhang and Hebert, 1999), and fingerprints (Sun and Abidi, 2001). A method described in (Wang et al., 2000) is based on geodesic distances in combination with additional local curvature parameters.

In this paper we present a new geometric approach to the matching problem with characteristic points. Some preliminary ideas were already presented in the form of an extended abstract in (Dimitrov et al., 2005). In the next section we introduce necessary notations and give a formal definition of the problem. In section 3 we present the basic algorithm and show how to use this method for the approximation of the optimal matching. Experimental results on real and synthetic data are provided in section 4, and conclusions and future work are given in section 5.

2 FORMAL PROBLEM DESCRIPTION

We consider two point sets P and S in $\mathbb{R}^3$. Usually we assume that S is the (infinite) set of points on a triangulated surface. The corresponding triangulation will be denoted by $\mathcal{S}$. However, this assumption is not crucial. The algorithms presented in the next sections can be applied with small changes if S is a finite, dense sample of points on a surface.

Our main goal is to register P into a model S. The quality of the registration will be evaluated by the *directed Hausdorff distance*. The distance between a point a and a compact point set B in d-dimensional space $\mathbb{R}^d$ is defined as

$$\text{dist}(a, B) = \min_{b \in B} ||a - b||$$

where $||\cdot||$ is the Euclidean norm in $\mathbb{R}^d$. For two compact sets A, B in the one-sided Hausdorff distance from A to B is defined as

$$\overrightarrow{H}(A, B) = \max_{a \in A} \text{dist}(a, B) = \max_{a \in A} \min_{b \in B} ||a - b||.$$

The size of a problem instance $(P, \mathcal{S})$ depends on two parameters: k, the number of points in P, and n, the number of triangles in $\mathcal{S}$. We remark that in our applications $k << n$ can be assumed. Moreover, we assume that two subsets of characteristic points $S_c \subseteq S$ and $P_c \subseteq P$ are given. For a precise analysis of our algorithms we introduce the additional parameters $k_c = |P_c|$ and $n_c = |S_c|$. Both parameters should be seen as some reasonable constants. The special role of characteristic points is expressed by the additional requirement, that each $p \in P_c$ must be mapped onto (or close to) a characteristic point $q \in S_c$.

First we have to classify several types of matchings.

Definition 2.1 *Given two parameters $\mu, \eta \geq 0$ a rigid transformation $T : \mathbb{R}^3 \to \mathbb{R}^3$ is called (μ, η)-matching if the following two conditions hold:*

1. $\mu(T) := \overrightarrow{H}(T(P \setminus P_c), S) \leq \mu$, *and*
2. $\eta(T) := \overrightarrow{H}(T(P_c), S_c) \leq \eta$.

If ϵ is an upper bound for $\mu(T)$ and $\eta(T)$ we denote T as an *ϵ-matching*. In line with the notations above, we define $\epsilon(T) = \max(\mu(T), \eta(T))$. The minimal $\epsilon(T)$ over all rigid transformations T is denoted by ϵ_{opt}, and a corresponding matching is an *optimal matching*. For a given $\lambda > 1$, a matching T is a *λ-approximate matching*, if $\epsilon(T) \leq \lambda \epsilon_{opt}$.

Furthermore, we introduce the notion of *semi-optimal matchings*. To this end we fix a sequence $\overline{S} = (\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_{k_c})$ of predefined matching positions for the sequence of characteristic points $(p_1, p_2, \ldots, p_{k_c})$. We restrict our attention to matchings T with $T(p_i) = \bar{s}_i$ for $i = 1, \ldots, k_c$. Let us

denote this set of matchings by $\mathcal{M}_{\overline{S}}$. We assume that P_c and $\overline{S}$ are congruent, because otherwise $\mathcal{M}_{\overline{S}}$ is empty.

A matching $T \in \mathcal{M}_{\overline{S}}$ is a $(\mu(T), \eta_0)$-matching, where $\eta_0 = \overrightarrow{H}(\overline{S}, S_c)$ is a common value for all matchings in $\mathcal{M}_{\overline{S}}$. A matching $T \in \mathcal{M}_{\overline{S}}$ is called *semioptimal matching* (with respect to $\overline{S}$) if $\mu(T)$ is minimal.

A trivial case with $|\mathcal{M}_{\overline{S}}| = 1$ occurs, if P_c contains three or more non-collinear points. Thus, we will focus our attention to matchings with two characteristic points. In a first step we design an algorithm to compute semioptimal matchings for a given set $\overline{S}$. Then, based on the semioptimal solution, we show how to compute a λ-approximate matching for any $\lambda > 1$.

3 THE 2 POINT CASE

3.1 Semioptimal Matchings

In general a rigid transformation in 3-dimensional space has six degrees of freedom: three for fixing a translation and three more describing a rotation. The first step of our approach is devoted to a very special case with one degree of freedom only. Assuming a pair $\overline{S} = (\overline{s_1}, \overline{s_2})$ of predefined matching positions for the two characteristic points p_1, p_2, the matchings in $T \in \mathcal{M}_{\overline{S}}$ have a fixed translation $(T(p_1) = \overline{s_1})$ and two fixed parameters of the rotation (mapping the axis (p_1, p_2) to the axis $(\overline{s_1}, \overline{s_2})$). Thus, it remains to consider all rotations around the axis $(\overline{s_1}, \overline{s_2})$. First, we present an algorithm which reports (for any μ) all transformations $T \in \mathcal{M}_{\overline{S}}$ with $\mu(T) \leq \mu$.

Basic Algorithm (Outline)

1. Fix a rigid transformation $T_0 : \mathbb{R}^3 \to \mathbb{R}^3$ such that $T_0(p_1) = \overline{s_1}$, $T_0(p_2) = \overline{s_2}$. For all $p_i \in P \backslash \{p_1, p_2\}$ let $C_i = C(p_i)$ be the circle with the following properties (see figure 1):
   - the center of C_i is on the line defined by p_1 and p_2,
   - C_i lies in a plane perpendicular to $\overline{p_1, p_2}$, and
   - p_i is on C_i.

2. Consider the transformed circle $T_0(C_i)$ and let the point $p_i'(\alpha)$ rotate along this circle starting from $T_0(p_i)$, i.e., $p_i'(0) = T_0(p_i)$. Compute sets of intervals $I_i = \{\alpha \,|\, dist(p_i'(\alpha), S) \leq \mu\}$, for $i = 3, \ldots, k$.

3. Compute $I = \cap_{i=3}^{k} I_i$. For each $\alpha \in I$ let $R_\alpha(s, s')$ be the rotation around axis $\overline{s, s'}$ with angle α. Then $T_\alpha := R_\alpha(s, s') \circ T_0$ is a rigid transformation with $\mu(T_\alpha) \leq \mu$.

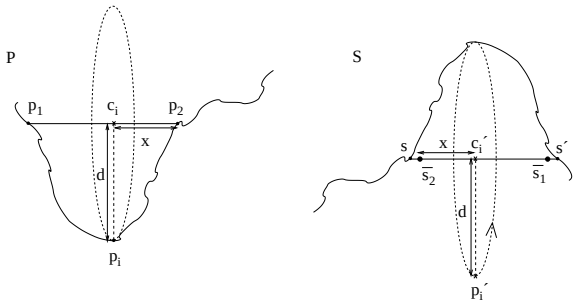

Figure 1: Corresponding points and the rotation of the point p_i'.

An obvious upper bound for the run time complexity of the algorithm is almost straightforward. To find the range of angles, for which the circle is at most μ near to a triangle, we compute the intersection of the circle and the Minkowski sum of the triangles and a ball with radius μ centered at the origin of the coordinate system. Computing such an intersection costs $O(1)$ time. In the worst case it should be repeated for all n triangles. To compute one particular interval I_i we sort and merge the $O(n)$ intervals, $O(1)$ contributed from each triangle, which cost $O(n \log n)$. Computing the intersection I can be done in $O(n)$ time in a sweep line manner. This altogether gives the following bound:

Lemma 3.1 *The run time complexity of the basic algorithm presented above is* $O(k\, n \log n)$.

The basic algorithm can be used as a decision algorithm answering the question whether for a given μ there is some matching T with $T(p_1) = s, T(p_2) = s'$ and $\mu(T) \leq \mu$ for all other points of P. Thus, using binary search one can approximate a semioptimal matching. However, it is also possible to compute the precise value μ of a semioptimal matching by the following modification of the basic algorithm. Instead of computing the interval sets $I_i = \{\alpha \mid \text{dist}(p_i'(\alpha), S) \leq \mu\}$, we compute the functions $f_i(\alpha) := \text{dist}(p_i'(\alpha), S)$. This function is the lower envelope of the distance functions of a rotating point to the surface triangles. Then, instead of computing $I = \cap_{i=3}^{k} I_i$, we compute the upper envelope f of all functions f_i. The minimum of f is the μ-value of a semioptimal matching.

To determine the description complexity of f it is necessary to apply the theory of Davenport-Schinzel sequences, see (Sharir and Agarwal, 1995). Because the detailed analyzes is beyond the scope and space of this paper, we only mention the main facts. Each function f_i is the lower envelope of n distance functions between a point on a circle and a triangle. So, the total number of functions contributing to f is $O(kn)$. The distance function between a point on a circle and a triangle can be described piecewise by polynomials of degree 4. The envelope f of $O(kn)$

such polynomials is related to a $(kn, 4)$-*Davenport-Schinzel sequence*, whichs maximal length is bounded from above by $O(kn\, h(kn))$, where $h(kn)$ is a very slowly growing sublogarithmic function. In divide-and-conquer manner, see (Sharir and Agarwal, 1995) for details, we can compute the envelope f in time $O(kn\, h(kn) \log kn)$, which is also the upper bound for time complexity of the semioptimal matching.

Lemma 3.2 *The run time complexity of the algorithm for computing a semioptimal matching is* $O(kn\, h(kn) \log kn)$.

3.1.1 Improving the Run Time Complexity

Both run time bounds above are dominated by the fact, that for a point rotating on a circle C_i the distance to all triangles of the surface is taken into account. However, for computing the intervals I_i on a circle that are close to the surface it is sufficient to consider only the triangles that are close to the circle. In reality these triangles form only a small subset of the whole surface. The same applies for the computation of the distance functions $f_i : [0, 2\pi] \longrightarrow \mathbb{R}$ that can be restricted to the range of angles where the corresponding points are close to the surface.

Thus, we aim at the construction of a data structure supporting queries for triangles close to a given point. We adapt a well known approach subdividing the bounding box of the surface S by equidistant planes parallel to the xy-,xz- and yz-plane. In a first step we compute for each subbox the list of all triangles intersecting that box.

In the main procedure the distance between a point and the surface is computed by point location in the grid of subboxes and checking only the triangles associated with the located subbox or with its neighboring subboxes. Using an $m \times m \times m$ grid the expected number of triangles to be checked is reduced from n to $\frac{27n}{m^3}$. It is clear that such an average argument does not help in a worst case analysis, and indeed, one can construct surfaces with very large triangle lists in many subboxes. However, accepting some reasonable assumptions on the surface representation it will be possible to prove some useful upper bounds. Since this analysis is technically involved (and not in the main focus of a computer vision conference) we will restrict ourself to sketch the main ideas needed for that proof.

We consider a triangulated surface $\mathcal{S}$ given by n triangles and denote the minimal and maximal lengths of triangle edges by l and L. We will call $\mathcal{S}$ *a well represented surface* with respect to global constants (independent of n) $c_1 \geq 1, c_2 > 0, c_3 \geq 1, c_4 \in \mathbb{N}, 0 < \alpha \leq \pi/3$, if the following properties hold:

1. *Shape property:* The three side lengths of the bounding box B of $\mathcal{S}$ differ at most by the factor c_1 and the projection of $\mathcal{S}$ to at least one of the three side planes of B covers at least a c_2 fraction of the side plane area.
2. *Edge ratio:* $\frac{L}{l} \leq c_3$
3. *Stabbing density*: Any line segment of length l in the space intersects at most c_4 triangles.
4. *Fatness:* All triangles are α-fat, i.e. all angles are at least α.

We would like to stress that most surfaces encountered in the practice satisfy above properties.

Theorem 3.3 *If $\mathcal{S}$ a well represented surface is subdivided into a $\sqrt{n} \times \sqrt{n} \times \sqrt{n}$ grid of subboxes then any subbox is intersected by at most a constant number of triangles. This constant depends only on the constants in the definition above*

Since any circle in space intersects at most $O(\sqrt{n})$ subboxes we obtain a sublinear runtime for both, the basic algorithm and the algorithm for semioptimal matching.

Lemma 3.4 *If $\mathcal{S}$ is a well represented surface subdivided into a $\sqrt{n} \times \sqrt{n} \times \sqrt{n}$ grid of subboxes then the run times complexity of the basic and semioptimal matching algorithms are $O(k\sqrt{n} \log n)$ and $O(k\sqrt{n}\, h(k\sqrt{n}) \log kn)$, respectively. The preprocessing time is $O(n^{1.5})$.*

We remark that the properties *edge ratio* and *fatness* can be relaxed in such a way that they can be violated by $O(\sqrt{n})$ triangles. In that case theorem 3.3 holds only for the number of nonviolating triangles, but we obtain the same run time bound as above.

Finally, we remark that the $\sqrt{n} \times \sqrt{n} \times \sqrt{n}$ grid size was chosen particularly with regard to best possible results in the theoretical run time analysis. In practical experiments already $20 \times 20 \times 20$ grids proved to be very useful.

3.2 The Approximation Problem

There are two groups of standard approaches for approximating an optimal solution starting from some initial solution. The first group consists of heuristic approaches, including simulated annealing (van Laarhoven and Aarts, 1987) and ICP variants (Rusinkiewicz and Levoy, 2001), (Mitra et al., 2004). These methods have proved to be useful in many practical situations, but, it is hard to prove something about the quality of the approximation in the worst case. The algorithm from the second group are based on a method capable to compute initial solutions that are provably close to the optimal solution, and on a dense discretization pattern around the initial solution.

Our approach belongs to the second group. First we compute a semioptimal matching T as a initial solution, such that $\epsilon(T) \leq C \cdot \epsilon_{opt}$, where C is a constant.

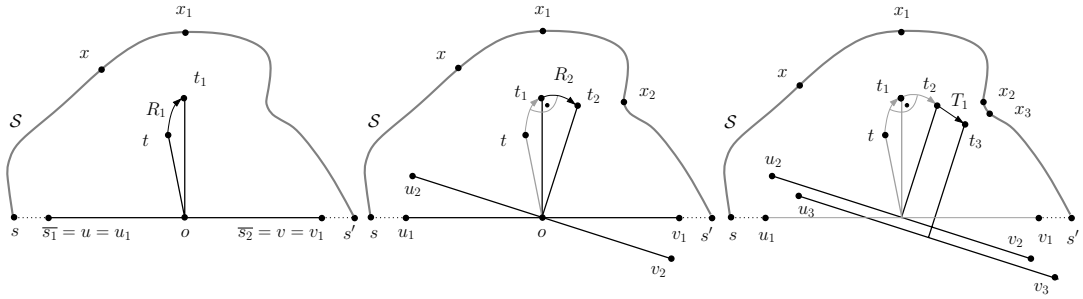

Figure 2: The positions of the points t, $\overline{s_1}$ and $\overline{s_2}$ after applying the transformations R_1, R_2, T_1. Points x, x_1, x_2, x_3 are the nearest points on the surface $\mathcal{S}$ of the points t, t_1, t_2, t_3 correspondingly.

Then, for any given $\lambda > 1$, we construct a grid-like pattern around the initial solution such that at least one of the grid points represents a λ-approximation of the optimal matching. The density of the grid (and consequently the size of the grid pattern) will depending only on λ and on $\epsilon(T)$.

A common key problem of many approximation problems focuses on the fact the the value of an optimal solution is unknown. Here we are able to derive upper and lower bounds for ϵ_{opt} from the results of some semioptimal matchings. Since each pair (s, s') of characteristic points on S could constitute the approximated destination of (p_1, p_2) we apply this procedure for each such pair. More precisely, we compute the semioptimal matching $T_{s,s'}$ mapping (p_1, p_2) onto the point pair $(\overline{s_1}, \overline{s_2})$, where $(\overline{s_1}, \overline{s_2})$ forms the same line and has the same center as (s, s'), but $||\overline{s_1} - \overline{s_2}|| = ||p_1 - p_2||$. We denote the qualities of these semioptimal matchings by $\delta_{s,s'} = \epsilon(T_{s,s'})$ and the best value by $\delta = \min\{\delta_{s,s'} \mid s, s' \in S_c\}$. Furthermore, we introduce the radius r_P and the relative radius ρ_P of the point set P with respect to the center of the characteristic points as follows:

$$\begin{aligned} r_P &= \max_{p \in P} ||\tfrac{p_1+p_2}{2} - p||, \\ \rho_P &= \tfrac{r_P}{\frac{||p_1-p_2||}{2}} = \tfrac{2\, r_P}{||p_1-p_2||}. \end{aligned} \tag{1}$$

Obviously, $\rho_P \geq 1$.

Proposition 3.5 $\delta \geq \epsilon_{opt} \geq \frac{\delta}{\rho_P + 2}$.

Proof The upper bound for ϵ_{opt} is trivial, because the best semioptimal matching cannot be better than the global optimum.
To prove the lower bound, let T_{opt} denote the optimal matching. According to the definition there are two characteristic points $s, s' \in S_c$ such that $||T_{opt}(p_1) - s|| \leq \epsilon_{opt}$ and $||T_{opt}(p_2) - s'|| \leq \epsilon_{opt}$. We consider the semioptimal matching $T_{s,s'}$ and the additional transformation T_{add} transforming $T_{s,s'}$ into T_{opt}. Obviously one can decompose T_{add} into two rotations and one translation $T_{add} = T \circ R_2 \circ R_1$, where

- the first rotation R_1 is around the axis spanned by s and s',
- the second rotation R_2 is around an axis through the center o of the segment ss' and orthogonal to this segment,
- and T is a unique, final translation.

We have $T_{opt} = T_1 \circ R_2 \circ R_1 \circ T_{s,s'}$ and $\delta = \epsilon(T_{s,s'}) \leq \epsilon(R_1 \circ T_{s,s'})$ because $T_{s,s'}$ was semioptimal and R_1 is a rotation around (s, s'). Thus, there is a point $p \in P$ such that the distance of $(R_1 \circ T_{s,s'})(p)$ to the surface S is at least δ. To avoid double indices let us introduce the following notations, see figure 2 for illustration: $u := s_1$, $v := s_2$ and $t := T_{s,s'}(p)$. Furthermore, we set $u_1 := R_1(u)$, $u_2 := R_2(u_1)$, $u_3 := T(u_2)$ and analog notations for v and t. Finally, let x_1, x_2, and x_3 be the closest points on the surface to t_1, t_2, and t_3. From the observations above we know that

$$\delta \leq \delta_{s,s'} \leq \|t_1 - x_1\|. \tag{2}$$

Since x_1 is the closest point on S to t_1, we have $\|t_1 - x_1\| \leq \|t_1 - x_3\|$ and applying the triangle inequality we obtain:

$$\|t_1 - x_1\| \leq \|t_1 - x_3\| \leq \|t_1 - t_2\| + \|t_2 - t_3\| + \|t_3 - x_3\|. \tag{3}$$

Now we estimate the three terms on the right side of the inequality above. Since t_3 is the image of a point $p \in P$ under T_{opt} we get

$$\|t_3 - x_3\| \leq \epsilon_{opt}. \tag{4}$$

Then distance between t_1 and $t_2 = R_2(t_1)$ depends on the distance d of t_1 to the rotation axis and the angle of the rotation. Clearly, d is bounded from above by the distance to the center o of the segment (s, s') and the angle of rotation is the same as for turning u_1 to u_2. Thus, $\frac{\|t_1 - t_2\|}{\|u_1 - u_2\|} = \frac{d}{\|u_1 - o\|} \leq \frac{\|t_1 - o\|}{\|u_1 - o\|} \leq \rho_P$ and

$$\|t_1 - t_2\| \leq \rho_P \|u_1 - u_2\|. \tag{5}$$

We remark that the translation T applied to u_2 and v_2 increases the distance between u_2 and s, or the distance between v_2 and s' and thus

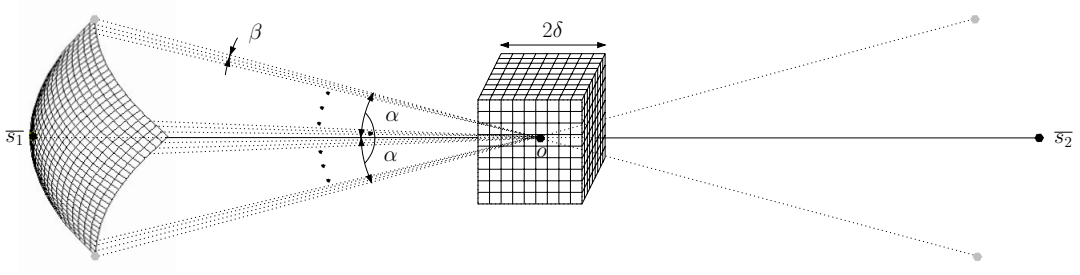

Figure 3: Two grids around the points s' and o.

$$\begin{aligned} \|u_1 - u_2\| &\leq \|s - u_2\| = \|s' - v_2\| \\ &\leq \max\left(\|s - u_3\|, \|s' - v_3\|\right) \\ &\leq \epsilon_{opt}. \end{aligned} \tag{6}$$

The same argument yields $\|u_2 - u_3\| \leq \epsilon_{opt}$ and since T is a translation

$$\|t_2 - t_3\| = \|u_2 - u_3\| \leq \epsilon_{opt}. \tag{7}$$

It remains to merge (2) with (3) and to plug in (4), (5), (6) and (7) into the right hand side of (3):

$$\begin{aligned} \delta &\leq \|t_1 - x_1\| \\ &\leq \|t_1 - t_2\| + \|t_2 - t_3\| + \|t_3 - x_3\| \\ &\leq \rho_P \cdot \epsilon_{opt} + \epsilon_{opt} + \epsilon_{opt} = (2 + \rho_P)\epsilon_{opt}. \end{aligned}$$

It follows that $\epsilon_{opt} \geq \dfrac{\delta}{\rho_P + 2}$. □

Given an approximation factor $\lambda > 1$ we try to improve the best value δ obtained so far by small changes of the predefined matching positions $\overline{s_1}, \overline{s_2}$. The bounds above can be used to design a grid based set of perturbed matching positions which is dense enough to include a λ-approximation of an optimal matching. Let us decompose the transformation, which leads from one of the semioptimal matchings $T_{\overline{s_1},\overline{s_2}}$ to the optimal matching, into the rotational and translational component. A spherical squared grid centered at the point $\overline{s_1}$ determines the rotational component, and a cubic grid centered at the point o determines the translation component, see figure 3. The total size of the spherical grid (the angle α), and the size of the cubic grid depend on δ, namely, $\alpha = \frac{2\delta}{\|p_1 - p_2\|} = \frac{\rho_P}{r_P}$ and the spherical grid size is 2δ. This choice ensures that the optimal matching is within the range of that scheme. The subdivision of the spherical grid is defined by an angle $\beta = \frac{(\lambda-1)\delta\, r_P}{(\rho_P+2)}$ and the cells of the cubic grid have side length $\frac{\sqrt{3}(\lambda-1)\delta}{3(\rho_P+2)}$. It can be verified easily that such an approximation scheme is fine enough to provide for a semioptimal matching that is at most $(\lambda - 1)\epsilon_{opt}$ far from its optimal matching position.

Consequently, the number of the grid combinations, defining possible matching positions $(\overline{s_1}, \overline{s_2})$, is $(\frac{16\sqrt{3}\rho_P(\rho_P+1)}{\lambda-1} + 1)^2 (\frac{\sqrt{3}(\rho_P+1)}{\lambda-1} + 1)^3$. This implies the following estimation of the total run time:

Lemma 3.6 *The run time complexity of the λ-approximation presented above is* $O(n_c^2\, k\, n\, h(kn)\, \log(kn) \frac{\rho_P^5}{(\lambda-1)^5})$.

We remark that the factor n in this formula can be improved in the same way as discussed in the analysis of the semioptimal matching in the section 3.1.1. Moreover, in the applications the ratio ρ_P can be regarded as a small constant.

4 EXPERIMENTAL RESULTS

We have implemented a variant of the semioptimal matching algorithm with the subdivided bounding box data structure as described in section 3.1. The maximal size of the subbox structure was set to $20 \times 20 \times 20$, which compared with the theoretically optimal one $(\sqrt{n} \times \sqrt{n} \times \sqrt{n})$ decreases the preprocessing time (creating and initializing of the subbox data structure), but increases the matching time. Usually, in the medical navigation applications, there is enough time for preprocessing, so one can build a finer subbox structure to increase the matching time. Time efficiency, preciseness and robustness of the algorithm were verified on real and synthetic data. The surfaces $\mathcal{S}$ were reconstructed by scans acquired by MR, CT and range scanner devices, or were generated synthetically. As it is a case with most surfaces encountered in the practice, our test surfaces are also well defined in a sense described in 3.1.1. An example of the *head* test model reconstructed by MR scan data is given in figure 4. The set P was generated in the following manner: first, some very small subset P_s of points from the surface $\mathcal{S}$ was randomly selected, s.t. two of the points were characteristic points and the rest some arbitrary points from the surface $\mathcal{S}$. Then, the points from P_s were randomly perturbed

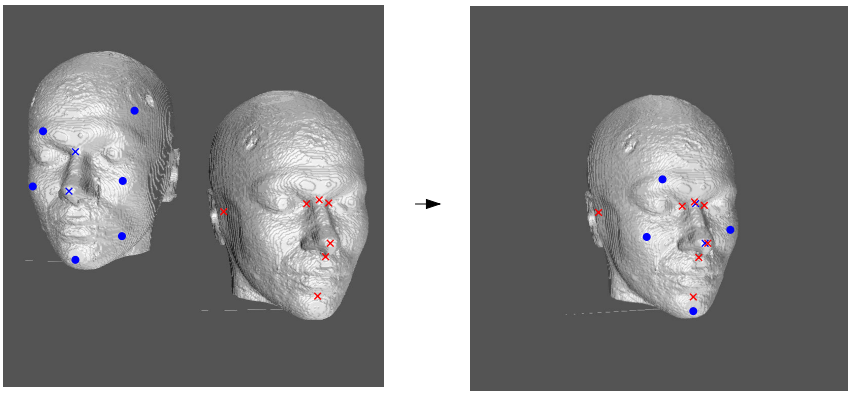

Figure 4: Registration of the head model. The characteristic points are denoted with crosses, and the non-characteristic ones with circles.

Table 1: Some relevant parameters of the tested models. The last parameter shows the size of the subcube structure build around the models.

model	# of faces	$\lvert s_c \rvert$	avg. edge length	diameter	# of subboxes
head	358784	10	0.988736	330.193	5120
bunny	69451	10	0.001471	0.25025	6400
cow	5804	10	0.021878	1.27111	1820
cat	671	6	0.014992	0.27041	3520

forming point set $\tilde{P}_s$. Finally, point set $\tilde{P}_s$ was transformed with a random rigid transformation T_r, i.e. $P = \{T_r(\tilde{p}_i) \,|\, \tilde{p}_i \in \tilde{P}_s\}$.

The algorithm was tested both on noiseless and noisy data. The tests were performed on P3, 1.27 GHz, 512 MB RAM - PC. To estimate the quality of the registration we used the following 3 measures:

- $measure1 = \max_i \frac{\|\tilde{p}_i - T_{out}(p_i)\|}{diameter(\mathcal{S})} = \max_i \frac{\|\tilde{p}_i - T_{out}(T_r(\tilde{p}_i))\|}{diameter(\mathcal{S})}$,
- $measure2 = \frac{\vec{H}(T_{out}(P)\mathcal{S})}{diameter(\mathcal{S})}$,
- $measure3 = \max_{i,j,i\neq j}(|1 - a_{ii}|, |0 - a_{ij}|)$, $a_{ij}, a_{ii} \in M_{out} \times M_r$.

where T_{out} is the output transformation given by our algorithm, which matches the point set P with the surface $\mathcal{S}$, and M_{out} and M_r are the corresponding matrices of the rigid transformations T_{out} and T_r respectively.

The results (see tables 2, 3) show that the implementation of the semioptimal algorithm is fast, robust and gives accurate matchings. Eventual further improvements of the accuracy can be achieved with the λ-approximation algorithm described in the section 3.2. Its implementation is one of the tasks for the future. Alternatively, one can use another heuristical approaches to improve the matching, although, as it was previously mentioned, there is no guarantee what kind of approximation of the optimal matching will be achieved.

5 CONCLUSION AND FUTURE WORK

We have given an efficient algorithms for matching a point set and a surface, both with characteristic points. The first nontrivial case, when the point set has two characteristic points, was considered. We design an algorithm which assumes that the matching positions of both characteristic points are known. Based on it, we show how a λ-approximation of the optimal matching can be computed. An implementation of this algorithm was tested on real and synthetic data, demonstrating its efficiency, robustness and accuracy. We would like to stress that with slight modification of the algorithms and analysis, we can extend our results and match arbitrary point sets and shapes with characteristic points in 3 space. The analysis shows that the presented algorithm is competitive in complexity with the solutions we have for solving the approximate matching problem when 3 characteristic point are known.

Further work will consist in implementation of the grid scheme needed for the λ-approximation algorithm, and designing and implementing a solution of the matching problem when the point set contains only one characteristic point.

REFERENCES

Alt, H. and Guibas, L. (1999). Discrete geometric shapes: Matching, interpolation, and approximation. In Sack, J.-R. and Urrutia, J., editors, *Handbook of Computational Geometry*, pages 121 – 153. Elsevier Science Publishers B.V. North-Holland, Amsterdam.

Besl, P. J. and McKay, N. D. (1992). A method for registration of 3-d shapes. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 14(2):239–256.

Cohen-Steiner, D. and Morvan, J.-M. (2003). Restricted delaunay triangulations and normal cycle. In *Proceedings of the 19th Annual ACM Symposium on Computational Geometry*, pages 237–246.

Delingette, H., Hebert, M., and Ikeuchi, K. (1993). A spherical representation for the recognition of curved objects. *International Conference on Computer Vision*, pages 103–112.

Dimitrov, D., Knauer, C., and Kriegel, K. (2005). Matching surfaces with characteristic points. In *Proceedings of the 21th European Workshop on Computational Geometry (EWCG), Eindhoven, Netherlands*.

Hagedoorn, M. (2000). *Pattern matching using similarity measures*. PhD thesis, Utrecht University.

Hoffmann, F., Kriegel, K., Schönherr, S., and Wenk, C. (1999). A simple and robust algorithm for landmark registration in computer assisted neurosurgery. Technical report, Freie Universität Berlin.

Johnson, A. and Hebert, M. (1997). Surface registration by matching oriented points. *International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, pages 121–128.

Kim, S.-J., Jeong, W.-K., and Kim, C.-H. (1999). Lod generation with discrete curvature error metric. In *2nd Korea Israel Bi-National Conference on Geometrical Modeling and Computer Graphics in the WWW Era*, pages 97–104.

Mitra, N. J., Gelfand, N., Pottmann, H., and Guibas, L. (2004). Registration of point cloud data from a geometric optimization perspective. In *Symposium on Geometry Processing*, pages 23–32.

Petitjean, S. (2002). A survey of methods for recovering quadrics in triangle meshes. *ACM Comput. Surv.*, 34(2):211–262.

Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the icp algorithm. In *Third International Conference on 3D Digital Imaging and Modeling*, pages 145–152.

Sharir, M. and Agarwal, P. (1995). *Davenport-Schinzel sequences and their geometric applications*. Cambridge University Press.

Sun, Y. and Abidi, M. A. (2001). Surface matching by 3d point's fingerprint. In *IEEE Int. Conf. On Computer Vision, vol. II*, pages 263–269.

van Laarhoven, P.-J.-M. and Aarts, E.-H.-L. (1987). *Simulated Annealing: Theory and Applications*. D.Reidel Publishing Company, Kluwer.

Wang, Y., Peterson, B. S., and Staib, L. H. (2000). Shape-based 3d surface correspondence using geodesics and local geometry. *Computer Vision and Pattern Recognition*, 2:2644–2651.

Yamany, S. M. and Farag, A. A. (2002). Surfacing signatures: An orientation independent free-form surface representation scheme for the purpose of objects registration and matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24(8)*, pages 1105–1120.

Zhang, D. (1999). *Harmonic Shape Images: A 3D Free-form Surface Representations and Its Applications in Surface Matching*. PhD thesis, Carnegie Mellon University.

Zhang, D. and Hebert, M. (1999). Harmonic maps and their applications in surface matching. In *Computer Vision and Pattern Recognition*, volume 2, page 2524.

Table 2: Experimental results on noiseless data. The values in the table are the average of the results of 100 runs of the algorithm, each time with a newly generated point set P.

model	preproc. [s]	matching [s]	# of calls of the basic algorithm
head	11.433	21.556	6
bunny	2.433	4.556	7
cow	0.201	1.771	5
cat	0.04	0.951	2
	measure1	measure2	measure3
head	0.000748	0.000103	0.0000037
bunny	0.004483	0.000203	0.0000663
cow	0.005954	0.001869	0.0032076
cat	0.009801	0.000899	0.0028853

Table 3: Experimental results on noisy data of the *head* model. Random noise, between 0 and *noise factor*average edge length*, was added to the points from P. The values in the table are the average of the results of 100 runs of the algorithm, each time with a newly generated point set P.

noise factor	preproc. [s]	matching [s]	# of calls of the basic algorithm
0	11.433	21.556	6
1	11.386	21.281	6
5	11.425	22.771	5
10	11.375	22.609	3
	measure1	measure2	measure3
0	0.000748	0.000103	0.0000037
1	0.008512	0.000703	0.0000067
5	0.015954	0.001869	0.0032076
10	0.036983	0.210899	0.0048853