

APPROXIMATE POINT-TO-SURFACE REGISTRATION WITH A SINGLE CHARACTERISTIC POINT

Darko Dimitrov, Christian Knauer*, Klaus Kriegel* and Fabian Stehn*
Institut für Informatik, Freie Universität Berlin, Berlin, Germany

Keywords: Point-to-surface registration, matching, medical navigation, approximation algorithms.

Abstract: We present approximation algorithms for point-to-surface registration problems which have applications in medical navigation systems. One of the central tasks of such a system is to determine a “good” mapping (the *registration transformation* or *registration* for short) of the coordinate system of the operation theatre onto the coordinate system of a 3D model M of a patient, generated from CR- or MRT scans. The registration φ is computed by matching a 3D point set P measured on the skin of the patient to the 3D model M . It is chosen from a class \mathcal{R} of *admissible transformations* (e.g., rigid motions) so that it approximately minimizes a suitable error function e (such as the directed Hausdorff or mean squared error distance) between $\varphi(P)$ and M , i.e., $\varphi = \arg \min_{\varphi' \in \mathcal{R}} e(\varphi'(P), M)$. A common technique to support the registration process is to determine either automatically or manually so-called *characteristic points* or *landmarks*, which are corresponding points on the model and in the point set. Since corresponding characteristic points are supposed to be mapped onto (or close to) each other, this reduces the number of degrees of freedom of the matching problem. We provide approximation algorithms which compute a rigid motion registration in the most difficult setting of only a single characteristic point.

1 INTRODUCTION

Most neurosurgical and an increasing number of otolaryngological operations currently require the support of *medical navigation systems*. The purpose of these systems is to provide an augmented image of the patient, e.g., the correct projection of the used instrument into a 3D model of the area of interest, such as the patient’s skull. To compute this projection, a good transformation has to be determined which maps the coordinate system of the operation field to the coordinate system of the model. Such transformations are called *registrations*, and their approximation is the central task we are investigating in this paper. Several approaches are known to solve this problem. Some of the methods currently used in practice are based on fiducial *landmarks*. Such landmarks are artificial markers, e.g., plastic cylinders containing a metal ball, which are attached to the skin of the patient and can automatically be detected in the model. In the beginning of the surgery these marker positions

are gauged with a traceable device. Then the registration is determined by mapping the measured points to the landmarks in the model.

Other solutions are based on geodesics and local geometry as in (Wang et al., 2000). A feature-based approach using thin-plate splines is presented in (Chui and Rangarajan, 2003) and applications in transcranial magnetic stimulation by point-to-surface registration using ICP are discussed in (Matthäus et al., 2006). These methods are either heuristics and therefore cannot provide guarantees on the quality of the result or are very sensitive to lost, misplaced or displaced landmarks.

In recent years algorithms have been developed which solve this registration problem by using so-called *characteristic points*, which are gauged along with an arbitrary set of points from the skin of the patient. Characteristic points are unique points in the 3D model with special anatomic properties, as the root of the nasal bone and give hints for the correct placement of the measured points.

The most general and difficult variant of rigid point-to-surface registration with characteristic points is the

*Supported by the German Research Foundation (DFG), grant KN 591/2-1.

scenario where only a single characteristic point can be measured in the operation field. These scenarios occur for example when the target area is only partly scanned or in case of an emergency operation when too much surface tissue is damaged.

Rigid motions in \mathbb{R}^3 have six degrees of freedom. The general strategy of our algorithms can be summarized as follows: first the translational component of the rigid motion is fixed by mapping the measured characteristic point onto the characteristic point on the surface. The remaining degrees of freedom are determined by iteratively choosing the direction \vec{d} of a rotation axis through the characteristic points from a set of allowed directions (two degrees) and the last degree is determined by rotating around this axis. The last part, the rotation around the axis, is computed by using an algorithm presented in (Dimitrov et al., 2006) (an outline of this algorithm is given later). After evaluating the quality of such a registration, the size of the set of allowed directions is reduced by excluding a neighborhood around \vec{d} based on the quality of the computed registration and a new direction is chosen. This process terminates after a certain constraint is fulfilled, see section 1.4.

1.1 Problem Description and Notation

Let $S \subset \mathbb{R}^3$ be a surface consisting of n triangles, representing the anatomic model of the patient, and let $P \subset \mathbb{R}^3$ be a point set consisting of k points measured from the patient (usually $k \ll n$). Furthermore let $S_c \subset S$ be a set of points on the surface (which will be called *characteristic points*). We think of points $s \in S_c$ as representing some characteristic anatomic feature of the patient (e.g., the root of the nasal bone). The corresponding set of characteristic points in the measured point set P is called $P_c \subset P$.

Definition 1.1 (optimal registration for a transformation class). *Given a triangulated surface S , a point set P and characteristic points on the model $S_c \subset S$ and in the point set $P_c \subset P$. A transformation t_{opt} is called optimal for a transformation class \mathcal{T} if*

$$t_{opt} \in \arg \min_{t \in \mathcal{T}} \left(\max \left(\vec{H}(t(P_c), S_c), \vec{H}(t(P), S) \right) \right)$$

Here, $\vec{H}(A, B)$ denotes the directed Hausdorff distance of a compact set $A \subset \mathbb{R}^3$ to a compact set $B \subset \mathbb{R}^3$. It is defined as

$$\vec{H}(A, B) := \max_{a \in A} \min_{b \in B} \|a - b\|,$$

where $\|a - b\|$ is the Euclidean distance of a and b in \mathbb{R}^3 .

We investigate a slightly modified problem by looking at scenarios where $S_c = \{s\}$ as well as $P_c = \{p\}$ consist of a single characteristic point and where the transformation class is restricted to the class of all rigid motions t which map p upon s . A solution minimizing the directed Hausdorff distance of P to S under this restriction is called *semioptimal*.

Definition 1.2 (semioptimal matching with a single characteristic point). *Let p and s be the characteristic points of P and S , respectively, and let $\mathcal{T}_s \subset \mathcal{T}$ be the set of rigid motions that map p onto s , i.e., $t(p) = s$. A matching t_{sopt} is called semioptimal if*

$$t_{sopt} = \arg \min_{t \in \mathcal{T}_s} \vec{H}(t(P), S).$$

1.2 Solving the Problem with Two Characteristic Points

A related matching problem where both P and S have two characteristic points was considered in (Dimitrov et al., 2006). Since the correspondence between two pairs of characteristic points is not enough to resolve all six degrees of freedom of rigid motions in \mathbb{R}^3 , the *semioptimal matching* with two characteristic points was considered, which has only one degree of freedom.

Definition 1.3 (semioptimal matching for two characteristic points). *Let p_1, p_2 and s_1, s_2 be the characteristic points of P and S , respectively, and let $\mathcal{T}_s \subset \mathcal{T}$ be the set of rigid motions which centrally align the line segment $\overline{p_1, p_2}$ with the line segment $\overline{s_1, s_2}$. A matching t_{sopt} is called semioptimal if*

$$t_{sopt} = \arg \min_{t \in \mathcal{T}_s} \vec{H}(t(P), S).$$

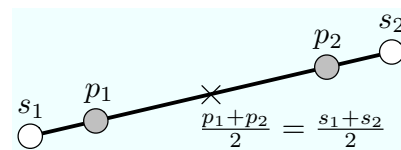


Figure 1: Points p_1, p_2, s_1, s_2 are centrally aligned, if all points lie on a line and the midpoints of the line segments $\overline{p_1, p_2}$ and $\overline{s_1, s_2}$ lie upon each other.

After centrally aligning p_1 to s_1 and p_2 to s_2 (see Figure 1) only the rotational part of t_{sopt} around the axis $\overline{s_1, s_2}$ has to be determined. An algorithm for computing such a semioptimal matching (Dimitrov et al., 2006) runs in $O(kn \log^2 kn)$ time. We refer to this algorithm as Alg_2 . It was shown that the quality of the semioptimal matching (compared to the optimal matching) depends on the relative spread of the characteristic points with respect to P . In our setting

this algorithm provides a 4-approximation to the optimum.

We restrict our attention to semi-optimal solutions because perturbation-based approximation schemes can be used to compute solutions that are arbitrary close to the optimum starting from a semi-optimal configuration (Dimitrov et al., 2006).

1.3 The General Strategy

In the following sections we present algorithms, which approximate the registration problem with a single characteristic point by sequentially fixing the degrees of freedom of the desired registration. The first three degrees, the translational part of the registration is determined by taking the vector difference of the characteristic points p and s . The remaining three degrees are computed in an iterative fashion. The remaining degrees of freedom can be described as determining the direction of a rotation axis through s and the rotation around this axis. We repeatedly choose an axis and determine the best rotation around this axis. By evaluating the quality of this registration we are able to exclude an area around the rotation axis from the parameter space and pick the next axis. For the last part, the rotation around an axis through the characteristic points, we introduce the notion of *virtual characteristic points*. Virtual characteristic points are auxiliary points in P and on S which extend the input for the one-point case to an input for Alg_2 . Given the characteristic point $p \in P$, we choose as the virtual characteristic point the furthest point $\hat{p} = \arg \max_{p' \in P} \|p - p'\|$ to p in P . For the virtual characteristic point in the model space we repeatedly choose points \hat{s} with distance $\|p - \hat{p}\|$ to s . The line segment s, \hat{s} is the axis around which P is rotated. This process is iterated until a certain quality constraint is fulfilled.

Definition 1.4 (distance function). *Let $Alg_2(S, P, (s, \hat{s}), (p, \hat{p}))$ be the set of rigid motions computed by Alg_2 if \hat{p} and \hat{s} are added to the input as virtual characteristic points for P and S respectively. The distance function $\vec{H}_{sopt} : \mathbb{R}^3 \rightarrow \mathbb{R}$ is defined as*

$$\vec{H}_{sopt}(\hat{s}) := \min_{t \in Alg_2(S, P, (s, \hat{s}), (p, \hat{p}))} \vec{H}(t(P), S).$$

The term *quality of a transformation* and *quality of a virtual characteristic point* is defined dual to the term distance function: the quality is maximized, if the distance function is minimized and vice versa.

1.4 The Approximation Settings with a Single Characteristic Point

After fixing the translational part of the registration two tasks remain: finding a rotation axis and finding the right rotation around this axis. We call the set of allowed directions for the rotation axis the *search space*. For a characteristic point s a search space R can be represented as the set of virtual characteristic points $\hat{s} \in \mathbb{R}^3$ for S , where each direction is defined by the line segment through s and \hat{s} . For a search space R let $\epsilon_R = \min_{\hat{s} \in R} \vec{H}_{sopt}(\hat{s})$ be the quality of the best possible solution for the rotation around this axis as determined by Alg_2 .

We present approximation algorithms for the following two problems in two scenarios: In the first scenario the search space is given by the intersection of a sphere S_r with radius $r = \|p - \hat{p}\|$ centered in s with the surface S , in the second scenario the search space is given as the set of all points on S_r . In the first scenario we only consider registrations that map \hat{p} exactly into S where in the second scenario we also investigate transformations which map \hat{p} close to S .

Problem 1.1. *For an approximation parameter Δ , determine the set $Q \subset R$ of virtual characteristic points such that*

$$\max_{\hat{s} \in Q} \vec{H}_{sopt}(\hat{s}) \leq \epsilon_R + \Delta.$$

The second problem arises in applications, where an absolute upper bound for the quality of the registration is required:

Problem 1.2. *For an upper bound Δ for the quality of a matching, determine the set $Q \subset R$ of virtual characteristic points such that:*

$$\forall \hat{s} \in Q : \vec{H}_{sopt}(\hat{s}) \leq \Delta.$$

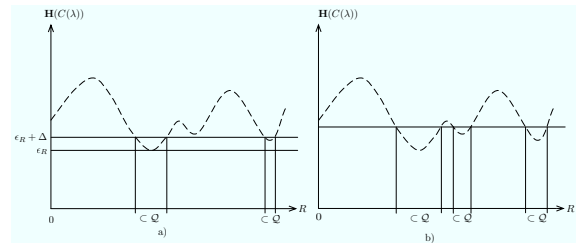


Figure 2: **a)** Illustration of Problem 1.1 **b)** Illustration of Problem 1.2.

Using the initial translation of P which maps p onto s , the computed set Q of valid directions for the rotation axis, and their corresponding rotation angles (computed by Alg_2) we report a dense representation of all rigid registrations which satisfy the properties stated above.

2 THE SEMIOPTIMAL SOLUTION FOR A SINGLE CHARACTERISTIC POINT

The following proposition gives a guaranty on the quality of the semioptimal matching with a single characteristic point.

Proposition 2.1. *Any semioptimal matching in the 1-point case is a 2-approximation of the optimal matching.*

Proof. Let ε_{opt} be the value of the optimal solution t_{opt} and let t_a be the translation that maps $t_{opt}(p)$ to $s = t_{sopt}(p)$. Since $\|t_{opt}(p) - s\| = \|t_{opt}(p) - t_{sopt}(p)\| \leq \varepsilon_{opt}$, the transformation $t_a \circ t_{opt}$ moves each point of P at most ε_{opt} far from its optimal position. Therefore,

$$\begin{aligned} \vec{H}(t_{sopt}(P), S) &\leq \vec{H}(t_a \circ t_{opt}(P), S) \\ &\leq \vec{H}(t_{opt}(P), S) + \varepsilon_{opt} \\ &\leq 2\varepsilon_{opt} \end{aligned}$$

□

Computing the semioptimal matching exactly. Let t be a translation that maps p to s , and let Γ be the set of all rotations around the point s . We are looking for the rotation $r_s = \arg \min_{r \in \Gamma} \vec{H}(r \circ t(P), S)$. We denote by B_j the ball with center $t(p)$ and radius $\|p - p_j\|$, for $p_j \in P \setminus \{p\}$. Let f_j denote the distance function between B_j and S . The function f_j is the lower envelope of the n distance functions between B_j and each triangle from S . Finding r_s corresponds to computing a minimum of the upper envelope f of the functions f_1, \dots, f_{k-1} .

To determine the description complexity of f , it is necessary to apply the theory of *Davenport-Schinzel sequences* (Agarwal and Sharir, 1995). Because the detailed analysis is beyond the scope and space of this paper, we only mention the main facts. Since each function f_j is the lower envelope of n distance functions between a ball and a triangle, it can be described piecewise by trivariate polynomials of degree 4. The complexity of the lower envelope of n such polynomials is related to Davenport-Schinzel sequence whichs maximal length is bounded from above by $\tilde{O}(n^3)$, where \tilde{O} is standard O -notation that ignores the parameters that influence the constant of proportionality, see (Agarwal and Sharir, 1995, Theorem 7.17) for details. Thus, f is an upper envelope of $\tilde{O}(n^3 k)$ trivariate polynomials of degree 4, and its combinatorial complexity is $\tilde{O}(n^9 k^3)$. The envelope f can be computed in a randomized expected time $\tilde{O}(n^9 k^3)$, see (Agarwal and Sharir, 1995, Theorem 7.25) for details.

Moreover the time complexity presented above holds only under the assumption that there is an ap-

propriate computational model that is able to find the zeros of trivariate polynomials of degree 4 in a constant time. Since no analytical or any other kind of solution that needs a constant time to solve that problem is known, we therefore draw our attention to approximation algorithms.

3 APPROXIMATING THE REGISTRATION FOR A SINGLE CHARACTERISTIC POINT

In this section we present algorithms that convert the input for the one-point problem to instances of the two-point problem by selecting appropriate virtual characteristic points. These instances are then solved using algorithm Alg_2 .

The central task is to find a suitable position for the virtual characteristic point \hat{s} on S . Suitable in this context means, that under the restriction that p is mapped onto s and \hat{p} mapped to \hat{s} the distance function for Alg_2 is minimized. We show that the slope of the distance function in the parameter space with regard to the selected virtual characteristic point \hat{s} is bounded to lie within $[-1, 1]$ and how this fact can be used to exclude parts of the search space.

3.1 The Lipschitz Constant of the Distance Function

Lemma 3.1. *Let S, s, P, p be as above. For any two points $\hat{s}_1, \hat{s}_2 \in \mathbb{R}^3$ with $\|s - \hat{s}_1\| = \|s - \hat{s}_2\| = \|p - \hat{p}\|$ the following holds:*

$$\left| \vec{H}_{sopt}(\hat{s}_1) - \vec{H}_{sopt}(\hat{s}_2) \right| \leq \|\hat{s}_1 - \hat{s}_2\|.$$

Proof. Assume that $t \in A(S, P, (s, \hat{s}_1), (p, \hat{p}))$ is one of the transformations mapping p to s and \hat{p} to \hat{s}_1 and let ρ be a rotation around s mapping \hat{s}_1 to \hat{s}_2 . Since \hat{p} is a farthest point from p , we have for any point $p' \in P$:

$$\|\rho \circ t(p') - t(p')\| \leq \|\rho \circ t(\hat{p}) - t(\hat{p})\| = \|\hat{s}_1 - \hat{s}_2\|.$$

Consequently $\vec{H}_{sopt}(\hat{s}_2) \leq \vec{H}(\rho \circ t(P), S) \leq \vec{H}(t(P), S) + \|\hat{s}_1 - \hat{s}_2\| = \vec{H}_{sopt}(\hat{s}_1) + \|\hat{s}_1 - \hat{s}_2\|$. A symmetric argument provides that $\vec{H}_{sopt}(\hat{s}_1) \leq \vec{H}_{sopt}(\hat{s}_2) + \|\hat{s}_1 - \hat{s}_2\|$. □

Let S_r be the sphere centered in s with radius $r = \|p - \hat{p}\|$. Lemma 3.1 states that moving a virtual characteristic point \hat{s}_1 to a point \hat{s}_2 on S_r , changes the value of the distance function by at most $\|\hat{s}_1 - \hat{s}_2\|$.

This bound on the Lipschitz constant of the distance function can be used to exclude parts of the parameter space around a virtual characteristic point, because it describes by which amount the function value can change within the neighborhood around this point.

3.2 Approximation Strategies for One-dimensional Search Spaces

To illustrate the idea of our approximation techniques we first consider the case where the search space R is one-dimensional. We are interested in virtual characteristic points \hat{s} that are $\|p - \hat{p}\|$ close to s . Therefore, we choose this search space to be the intersection C of the triangulated surface S with S_r . This is the scenario where all virtual characteristic points $\hat{s} \in C$ map \hat{p} exactly onto the surface.

The intersection C consists of a set of curves and each curve consists of a sequence of circular patches (possibly closed), where each patch results from the intersection of the sphere S_r with a single triangle of the model. The task is to determine those parts on each curve that provide *good* virtual characteristic points. Good virtual characteristic points \hat{s} have the property that either $|\vec{H}_{sopt}(\hat{s}) - \epsilon_R| \leq \Delta$ (Problem 1.1) or that $\vec{H}_{sopt}(\hat{s})$ is below a given threshold (Problem 1.2). This is achieved by probing several points on the curve and, depending on the value of the distance function for these virtual characteristic points, exclude parts of the neighborhood around these points from R .

Let q_C be an endpoint of curve $C \in \mathcal{C}$ (if C is closed, we cut C open at an arbitrary position q_C) and parametrize each point on C by its arclength to q_C . Let $C(\lambda)$ denote the point on C with arclength λ from q_C . Recall that according to Lemma 3.1 moving a point \hat{s} by Δ_λ on S_r changes the quality of the registration by at most Δ_λ .

Corollary 3.1. *For two points $C(\lambda), C(\lambda + \Delta)$ on a curve $C \subset S_r$ with an arclength distance of Δ , the following holds (see Fig. 3 a):*

$$\left| \vec{H}_{sopt}(C(\lambda)) - \vec{H}_{sopt}(C(\lambda + \Delta)) \right| \leq \Delta$$

Depending on the quality of a probe point \hat{s} one can exclude parts of its neighborhood in the parameter space. This can be used to bound the value of the distance function between two probe points, as according to Lemma 3.1 the scope in any point of the graph lies within $[-1, 1]$.

Solving Problem 1.1: The Lipschitz constant of the distance function can be used to bound the number of samples needed to provide an absolute approximation

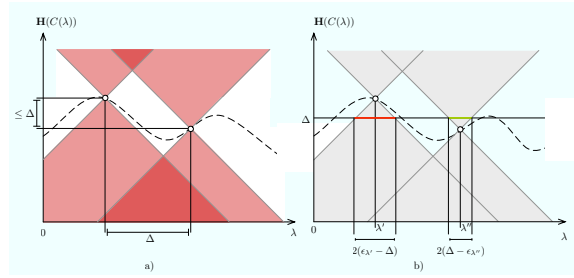


Figure 3: **a)** The quality of two points with arclength distance Δ differs by at most Δ , **b)** The ex- and inclusion areas for two probing positions with parameter λ' (left exclusion area) and λ'' (right inclusion area).

to $\epsilon_R = \min_{\hat{s} \in R} \vec{H}_{sopt}(\hat{s})$. The idea is to sample and test a set $Q \subset R$ of virtual characteristic points until ϵ_R is known to differ by at most Δ from the best quality of a sampled point. The number of samples needed to ensure that $\min_{\hat{s} \in Q} \vec{H}_{sopt}(\hat{s}) - \epsilon_R \leq \Delta$ is maximal if the distance function is constant on C . We have that:

$$\max_{\lambda' \in [\lambda, \lambda + \Delta]} \left| \vec{H}_{sopt}(C(\lambda')) - m_\lambda \right| \leq \Delta$$

where $m_\lambda = \min \left\{ \vec{H}_{sopt}(C(\lambda)), \vec{H}_{sopt}(C(\lambda + \Delta)) \right\}$.

Corollary 3.2. *Let the set of curves $C = \{C_0, \dots, C_i\}$ induced by the intersection of S and S_r with a total arclength of L be the search space R . Providing that $\min_{\hat{s} \in Q} (\vec{H}_{sopt}(\hat{s}) - \epsilon_R) \leq \Delta$ there is a probe set Q which size is bounded by $O(\frac{L}{\Delta})$.*

Solving Problem 1.2: The second section deals with determining all virtual characteristic points \hat{s} and their corresponding registrations which satisfy $\vec{H}_{sopt}(\hat{s}) \leq \Delta$ for a given quality Δ . As in the previously discussed problem the bound on the slope of the distance function allows us to exclude a neighborhood of a probe \hat{s} from the search space or to include a neighborhood to the solution, depending on the difference $\vec{H}_{sopt}(\hat{s}) - \Delta$. Let $C(\lambda)$ be a sample point on a curve C with parameter λ and distance value $\vec{H}_{sopt}(C(\lambda)) = \epsilon_\lambda$. The size of the in- or exclusion area depends on the difference of ϵ_λ and Δ (see also Fig. 3 b):

$\epsilon_\lambda - \Delta > 0$: The quality of the best registration if $C(\lambda)$ is chosen as the virtual characteristic point for S is above Δ . The next sample points \hat{s} on C with the property that $\vec{H}_{sopt}(\hat{s}) \leq \Delta$ have a distance to $C(\lambda)$ of at least $|\epsilon_\lambda - \Delta|$ in parameter space.

$\epsilon_\lambda - \Delta \leq 0$: The point $C(\lambda)$ and all points \hat{s} in its $\Delta - \epsilon_\lambda$ neighbourhood have the property that $\vec{H}_{sopt}(\hat{s}) \leq \Delta$ and can therefore be included in the solution set.

Robustness. This approach is sensitive to noise on P , especially to the influence of noise on the distance $r = \|p - \hat{p}\|$, which determines the radius of the sphere S_r . A slight perturbation of p or \hat{p} could prevent S_r from intersecting S , which leaves the set of possible virtual characteristic points empty.

3.3 Approximation Strategies for Two-dimensional Search Spaces

In this section we describe variants of the approximation strategies of Section 3.2 and extend the search space R to the whole sphere S_r . By fixing a virtual characteristic point $\hat{s} \in S_r$ we determine an axis $l = \overline{s, \hat{s}}$ around which P is rotated to find the best semi-optimal solutions. Using this search space increases the robustness of our approach against noise on P .

Solving Problem 1.1: We want to determine a set $Q \subset R$ of virtual characteristic points, such that $\min_{q \in Q} \vec{H}_{sopt}(q) - \min_{q \in R} \vec{H}_{sopt}(q) \leq \Delta$. Such a probe set Q has the property that the search space S_r is completely contained in the union of all balls with radius Δ , centered in a sample point $q \in Q$. In other words any probe set $Q \subset S_r$ satisfying

$$\forall a \in S_r \exists \hat{s} \in Q \|a - \hat{s}\| \leq \Delta$$

is a valid probe set.

Corollary 3.3. *There is a non empty sample set Q which provides that $\min_{\hat{s} \in Q} \vec{H}_{sopt}(\hat{s}) - \epsilon_R \leq \Delta$ for $R = S_r$ whose size is bounded by $O\left(\frac{r^2}{\Delta^2}\right)$, for $r = \|p - \hat{p}\|$.*

Solving Problem 1.2: As in the previous section we want to compute the set $Q = \{\hat{s} \in S_r | \vec{H}_{sopt}(\hat{s}) \leq \Delta\}$. To this end, we sample points \hat{s} of S_r and depending on the value of $\vec{H}_{sopt}(\hat{s})$ exclude regions in the neighborhood of \hat{s} from the search space or add a region to the solution. Recall that $\vec{H}_{sopt}(\hat{s}_2) \geq \vec{H}_{sopt}(\hat{s}_1) - \Delta$ for any \hat{s}_2 with $\|\hat{s}_1 - \hat{s}_2\| \leq \Delta$ and if $\vec{H}_{sopt}(\hat{s}_1) - \Delta < 0$, all points \hat{s}_2 in the intersection of S_r with a ball centered at \hat{s}_1 with radius $\Delta - \vec{H}_{sopt}(\hat{s}_1)$ have the property that $\vec{H}_{sopt}(\hat{s}_2) \leq \Delta$ and can therefore be included in the solution set. These observations lead to Algorithm 1.

Algorithm 1: Computing the set of virtual characteristic points providing an absolute error registration

Data: The model S , its characteristic point $s \in S$, the set of measured points P , their characteristic point $p \in P$, an absolute error approximation value Δ .

Result: The set Q of virtual characteristic points realizing a distance function value of at most Δ .

```

// initializing the result set
1 Q := ∅;
// candidate probe set to Sr
2 M := Sr;
3 while M ≠ ∅ do
    // select a random point
4   ŝ := takeRandomPoint(M);
5   if  $\vec{H}_{sopt}(\hat{s}) - \Delta > 0$  then
        // exclude neighborhood
7     M := M \ (Sr ∩ Ball(ŝ,  $\vec{H}_{sopt}(\hat{s}) - \Delta$ ));
8   else
        // include neighborhood
        // compute intersection
10  I := Sr ∩ Ball(ŝ,  $\Delta - \vec{H}_{sopt}(\hat{s})$ );
        // remove I from search space
11  M := M \ I;
        // add I to solution
12  Q := Q ∪ I;
13 end
14 end
15 return Q ;

```

The function $Ball(c, r)$ (lines 7 and 10) computes a ball with radius r centered in c .

Algorithm 1 computes all points \hat{s} on S_r with $\vec{H}_{sopt}(\hat{s}) \leq \Delta$. The practicability of Algorithm 1 is quite limited, as it needs to maintain an arrangement of circles on a sphere which is by itself a challenging problem. The methods currently known to compute such arrangements are too time consuming to be used in a medical navigation system (Cazals and Loriot, 2007).

4 AN IMPLEMENTATION FOR TWO-DIMENSIONAL SEARCH SPACES

As the computation of the arrangement of circles on a sphere is complex and time consuming, we present a simple and efficient implementation of Algorithm 1 which uses quadtrees (de Berg et al., 1997) to approx-

imate the arrangement: The information about areas that are excluded from the search space is maintained not on S_r but on six quadtrees which are placed on each side of an axis-parallel cube surrounding S_r .

4.1 The Implementation

The implementation proceeds round based and each round consists of the following steps: Consider the smallest axis parallel cube B_r which contains S_r . First a point \hat{s} is selected from a side of B_r by an heuristic described later. Then this point is projected down onto a point \hat{s}' on S_r . For \hat{s}' we call Alg_2 and compute the distance value $\vec{H}_{sopt}(\hat{s}')$ and by taking the difference of $\vec{H}_{sopt}(\hat{s}') - \Delta$ we determine the radius of the in-/ exclusion balls around \hat{s}' . Finally this ball is projected back onto B_r and the quadtrees which maintain the in- and excluded areas on B_r are refined to approximate the projected ball.

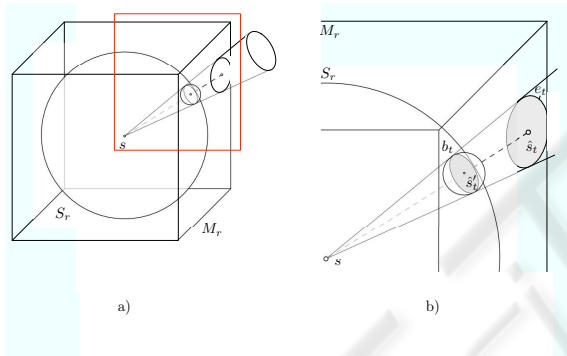


Figure 4: The ball b_t with radius $|\vec{H}_{sopt}(\hat{s}') - \Delta|$ intersects the sphere S_r in a circle on which is then projected onto B_r .

In Detail. In each round of the algorithm we determine a facet t of a quadtree and its center \hat{s}_t . The probe point \hat{s}'_t for this position is computed by intersecting S_r with the ray starting in s passing through \hat{s}_t . The difference of $\vec{H}_{sopt}(\hat{s}'_t) - \Delta$ defines the radius of an inclusion (in case of $\vec{H}_{sopt}(\hat{s}'_t) - \Delta \leq 0$) or an exclusion (in case of $\vec{H}_{sopt}(\hat{s}'_t) - \Delta > 0$) ball b_t centered in \hat{s}'_t . All sample points \hat{s} in the intersection of b_t with S_r either fulfill $\vec{H}_{sopt}(\hat{s}) \leq \Delta$ and can therefore be added to the solution set or can be discarded otherwise. This information has to be propagated onto the sides of B_r in order to adjust the quadtrees on its sides. To this end we determine the intersection e_t of B_r with the cone, whose apex is s and that touches the border of the intersection of b_t with S_r (see Fig. 4). All facets of the quadtrees intersected by e_t are subdivided until they are either not intersected by e_t anymore or have

an area below a reasonable threshold.

Note that the projection of points from B_r to S_r is not distance preserving: two points on a side and close to a corner of B_r have a larger distance to each other after being projected onto S_r than two points that lie closer to a midpoint of a side of B_r . This effect is compensated by the backward projection of b_t onto the sides, as the area of e_t depends on the distance of \hat{s}_t to the closest corner of B_r .

Figure 5 shows screen shots of the implementation at the moment where the first inclusion area was found.

A Heuristic for Choosing the next Sample Point.

To determine which facet to choose for the next sample point we introduce a max-area heuristic. Each facet of the quadtree has a priority and in addition is labeled either as included, excluded or unknown. If the label of a facet is unknown, the priority is set to the area of the facet and set to $-\infty$ otherwise. Each quadtree is initialized with one facet, a side of B_r , labeled unknown. All facets of the six quadtrees with label unknown are furthermore stored in a single max priority queue (the order of facets with the same priority is arbitrary). Facets that are covered by e_t are labeled either included or excluded, depending on whether e_t is an in- or exclusion area. Facets labeled included are added to the solution set. In each round the facet with the highest priority is chosen, as we expect the area of sample points with quality to be small and accordingly the exclusion areas to be large. Facets that are completely covered by e_t are labeled corresponding to the sign of the difference $\vec{H}_{sopt}(\hat{s}'_t) - \Delta$. All new facets that are not intersected by e_t are labeled unknown and are inserted into the priority queue, according to their area.

The algorithm terminates after either all leaves are labeled or under the given threshold or a certain number of rounds is reached.

4.2 Evaluation

We implemented the algorithm as described in Section 4.1 and evaluated the performance on a intel-core 2 duo computer with 2GB central memory. The computations on a model with about three thousand triangles and a point set P consisting of 8 points, both scaled to fit into the unit cube, took on average 25.31 seconds.

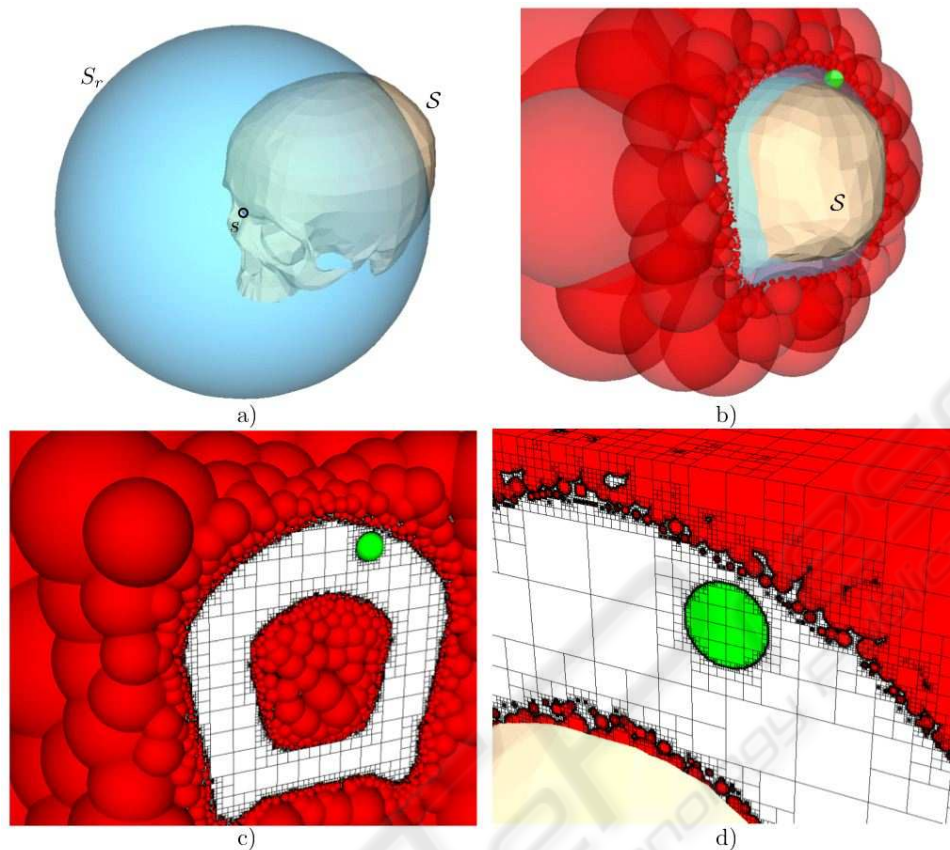


Figure 5: Screen shots of the implementation in the moment where the first inclusion area is found: **a)** the model S , its characteristic point s and the sphere S_r . **b)** the back of the model and the exclusion (dark gray/red) and inclusion (light gray/green) balls, some exclusion areas are hidden by the model **c)** the projected error balls, parts of the surrounding cube B_r with the quadtree structure on its sides **d)** a part of the model in the lower left corner and the quadtree refinement with included (light gray/green) and excluded areas (dark gray/red).

REFERENCES

- Agarwal, K. P. and Sharir, M. (1995). Davenport–schinzel sequences and their geometric applications. Technical report, Durham, NC, USA.
- Cazals, F. and Lorient, S. (2007). Computing the exact arrangement of circles on a sphere, with applications in structural biology: video. In *SCG '07: Proc. 23rd Annu. ACM Sympos. on Comput. Geom.*, pages 119–120.
- Chui, H. and Rangarajan, A. (2003). A new point matching algorithm for non-rigid registration. In *Computer Vision and Image Understanding archive*, vol. 89, pages 114–141.
- de Berg, M., van Kreveld, M., Overmars, M., and Schwarzkopf, O. (1997). *Computational Geometry: Algorithms and Applications*. Springer-Verlag.
- Dimitrov, D., Knauer, C., and Kriegel, K. (2006). Registration of 3d - patterns and shapes with characteristic points. In *Proc. of International Conference on Computer Vision Theory and Applications - VISAPP 2006*, pages 393–400.
- Dimitrov, D., Knauer, C., Kriegel, K., and Stehn, F. (2007). Approximation algorithms for a point-to-surface registration problem in medical navigation. In *Proc. Frontiers in Algorithmics Workshop - FAW 2007*, pages 26–37.
- Matthäus, L., Giese, A., Trillenber, P., Wertheimer, D., and Schweikard, A. (2006). Solving the positioning problem in TMS. In *GMS CURAC*, vol. 1.
- Wang, Y., Peterson, B., and Staib, L. (2000). Shape-based 3D surface correspondence using geodesics and local geometry. In *Comp. Vision Pattern Recog.*, vol. 2, pages 644–651.