Eurographics Symposium on Geometry Processing (2007)
Alexander Belyaev, Michael Garland (Editors)

Bayesian Surface Reconstruction via Iterative Scan Alignment to an Optimized Prototype

Qi-Xing Huang[1], Bart Adams[1,2], Michael Wand[1,3]

[1] Stanford University
[2] Katholieke Universiteit Leuven
[3] Max Planck Center for Visual Computing and Communication

Abstract

This paper introduces a novel technique for joint surface reconstruction and registration. Given a set of roughly aligned noisy point clouds, it outputs a noise-free and watertight solid model. The basic idea of the new technique is to reconstruct a prototype surface at increasing resolution levels, according to the registration accuracy obtained so far, and to register all parts with this surface. We derive a non-linear optimization problem from a Bayesian formulation of the joint estimation problem. The prototype surface is represented as a partition of unity implicit surface, which is constructed from piecewise quadratic functions defined on octree cells and blended together using B-spline basis functions, allowing the representation of objects with arbitrary topology with high accuracy. We apply the new technique to a set of standard data sets as well as especially challenging real-world cases. In practice, the novel prototype surface based joint reconstruction-registration algorithm avoids typical convergence problems in registering noisy range scans and substantially improves the accuracy of the final output.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Line and Curve Generation

1. Introduction

Recent improvements in laser rangefinder technology and digital geometry processing allow us to accurately digitize the shape and appearance of many physical objects. The pipeline of creating a 3D model from real-world data is typically comprised of three separate stages. In the acquisition stage, we obtain range scans of a real object by scanning it from multiple view points. Next, in the registration stage, we find the rigid body transformations that relate the coordinate system associated with each view point to the world coordinate system. These rigid body transformations unify the range scans into a single surface, and we use this surface in the final data reconstruction stage to generate a mesh model.

One may easily observe that the quality of the reconstructed models depends heavily on the accuracy of the data registration. State-of-the-art registration algorithms are able to align tens of range scans with very good quality [Neu97, HFG*06]. However, the registration problem becomes unstable as the number of scans reach the order of a few hundreds [KLMV05]. Moreover, in the presence of significant noise levels, standard methods often fail due to the absence of proper surface normals or the difficulty of defining a suitable distance field during registration.

In this paper, we introduce a novel joint registration and reconstruction technique for multiple range scans which overcomes above limitations. The proposed technique iteratively optimizes a prototype surface, which will eventually be the reconstructed surface, as well as the transformations that align the range scans. We start by computing a low resolution surface reconstruction from roughly aligned scans (which we expect as input) and use this surface as a prototype surface to align all range scans to. This improved alignment allows for a more detailed surface reconstruction, which in turn is then used to improve the alignment. This process is iterated until convergence.

The new formulation of simultaneous scan alignment and prototype surface optimization has several advantages compared to previous methods. Using the new approach, range scans are always aligned to a smooth prototype surface using a well-defined distance field. Other approaches typically rely on a distance field which is directly derived from the range scans, and as a result can not achieve the desired re-

© The Eurographics Association 2007.

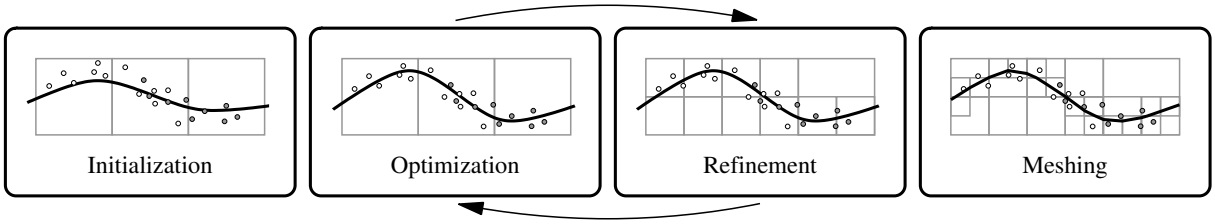

Figure 1: *The pipeline of our registration and reconstruction framework. The algorithm is initialized by computing a coarse prototype surface from the roughly aligned range scans. Next, the prototype surface control parameters and range scan alignments are improved by joint optimization. The prototype surface is refined and detail is introduced. Optimization and refinement are iterated until a desired accuracy is reached. Finally, the resulting prototype surface is meshed using a marching cubes algorithm.*

construction accuracy in the presence of strong noise artifacts. The proposed multi-resolution approach further increases the stability and efficiency for large and noisy input data and reduces the risk of getting stuck in local minima. In addition, the usage of the prototype surface avoids the difficulty of determining overlapping regions, because each data point is naturally associated to its closest point on the prototype. The ambiguities of determining the exact regions of partial overlap can introduce bias to the registration results that again reduce the obtained accuracy. Finally, our quadric-based prototype surface combines the strengths of implicit and explicit surface representations, such as fast point correspondence queries, high approximation power and flexibility in handling changing topology during the optimization process.

We apply our technique to a set of real-world data sets and show that our technique substantially increases the accuracy of the scan alignment and the quality of the reconstruction result in comparison to related state-of-the-art techniques.

1.1. Related work

Both registration and reconstruction algorithms have been studied considerably in the past. A complete survey of these two areas is beyond the scope of this paper and we refer to [BM92, Neu97, RL01, HH03, BR04, GMGP05, PHYH06, HFG*06] for state-of-the-art registration algorithms and to [CL96, ABK98, ACK01, OBA*03, NRDR05, HK06, KBH06, SLS*06] for recent advances in surface reconstruction.

Only few techniques exist that consider the registration and reconstruction problems together. Jin et al. [JDH*95] proposed a framework for iterative scan registration and mesh reconstruction from multiple input scans. The success of this method relies heavily on a successful mesh extraction, which in their method can be extremely difficult for large and noisy input scans, due to the necessary topology estimation step.

More recently, Tubic et al. [THL03] proposed a method to iteratively register the range scans to a reference surface that is defined by averaging the input scans' distance fields. A problem with this approach is that an average distance field is not well-defined in the presence of noise, and it is also not apparent how one can handle the cases where holes are present in the scans. Computation of the averaged distance field is also very expensive if this framework is used to process a large number of scans, or scans that have a large number of data points.

Both [JDH*95] and [THL03] still separate the registration and mesh reconstruction procedure. The nature of such optimization strategy is expectation minimization which can result in poor efficiency and convergence rate due to the possible different objectives of registration and reconstruction. In contrast, we overcome these problems by optimizing the objective function directly in terms of both the transformations and the shape of the prototype surface.

Two other related methods are presented in [MD97, LPW06]. Both methods fit a template surfaces to the input data. They need the topology as well as a rough initialization of the control parameters as input. In contrast, we compute the prototype surface automatically from the roughly aligned input data; our method does not require the user to specify a template surface.

Two algorithms for Bayesian surface reconstruction from range measurement are presented in [DTB06, JWB*06]. However, both of these techniques can only handle a single range scan. Moreover, to ease the formulation, these algorithms associate each data point with a corresponding point on the reference surface. Things are much more complicated in our setting since we have multiple range scans and the number of data points is huge such that we can not make direct associations.

Our prototype surface representation is similar to the multi-level partition of unity implicit definition of [OBA*03]. The surface is locally represented in each octree cell as a quadratic patch and patches are blended together using appropriate weight functions. This representation facilitates point to surface distance computation as well as obtaining curvature information used during the optimization process.

© The Eurographics Association 2007.

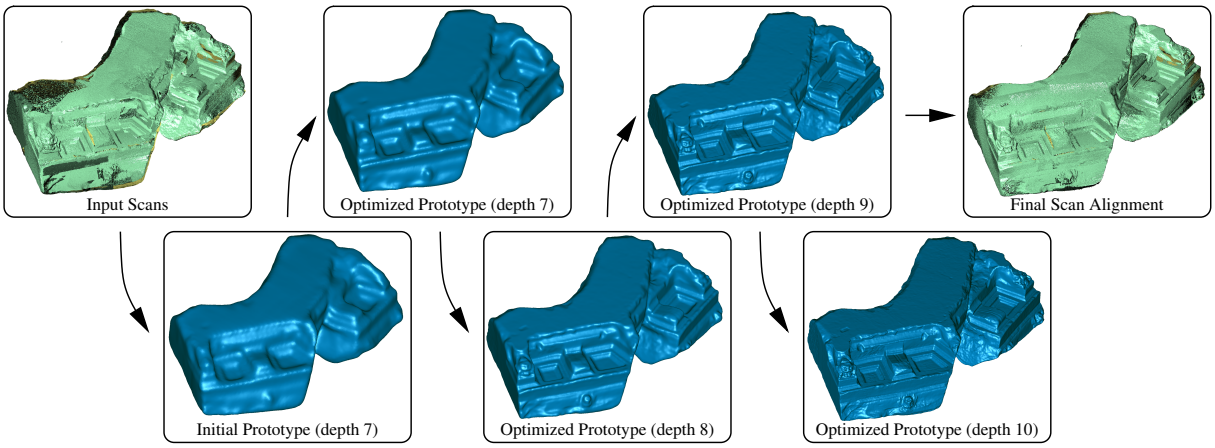

Figure 2: *Illustration of the different steps in our algorithm. The input is a roughly aligned set of range scans. From these range scans we compute a smooth initial prototype surface at octree depth 7. This prototype surface and the scan alignments are iteratively optimized and refined. In this example the final prototype is defined at depth 10. Note how detail is gradually introduced and also note that the final scan alignment has considerably improved compared to the input data.*

1.2. Overview

The pipeline of our approach is illustrated in Fig. 1. We begin by constructing a prototype surface from the input range scans. Then, we alternate between optimizing the prototype as well as poses of range scans and subdividing the surface, until its resolution is satisfactory or convergence is detected. Finally, we output the mesh generated by triangulating the prototype. Fig. 2 shows an example reconstruction obtained with the proposed algorithm.

The remainder of this paper is organized as follows. In Sec. 2, we introduce the Bayesian surface reconstruction framework, which is the foundation of our joint registration and reconstruction technique. Then, in Sec. 3, we describe the prototype surface definition and discuss how to initialize it from the input scan data and how to refine its representation to include more detail information. In Sec. 4, we discuss how to formulate the optimization problem, given our particular prototype surface definition. We summarize the surface reconstruction pipeline in Sec. 5. Results and comparisons with other approaches are given in Sec. 6. We end our paper in Sec. 7 with conclusions and a discussion of future work.

2. Bayesian Framework

The input of our algorithm is a set of N range scans S_i. Each such scan consists of N_i measurement points whose Cartesian coordinates in the scan's local coordinate system Σ_i are given by $\mathbf{s}_{ij}^0$. During the surface reconstruction process, we maintain and optimize a prototype surface $\mathcal{P}$, which is completely defined by its control vector Q. The specific form of the control vector depends on the choice of surface representation and could be something like the vector of control points for a B-spline surface or a vector of vertex positions for a triangular mesh. We will discuss its particular expression for our setting in Sec. 3, where we define the prototype surface in detail.

During registration, we want to find the relation between the coordinate system Σ_i of each scan S_i and the world coordinate system Σ. The relation is supposed to be a rigid body transformation α_i. In other words, the Cartesian coordinates of each data point are given in the world coordinate system Σ by $\mathbf{s}_{ij} = \alpha_i(\mathbf{s}_{ij}^0)$. In this paper, we fix the first scan during the reconstruction process, so that $\Sigma = \Sigma_1$.

The Bayesian surface reconstruction framework is described as follows. In probabilistic terms, we seek to find the most likely control vector Q for the prototype surface and the most likely rigid body transformation α_i for each scan given the measurements $\mathbf{s}_{ij}^0$. Hence, we want to maximize $P(Q, \{\alpha_i\}|\{\mathbf{s}_{ij}^0\})$. Using Bayes' rule, we can invert this probability as:

$$P(Q, \{\alpha_i\}|\{\mathbf{s}_{ij}^0\}) = \frac{P(\{\mathbf{s}_{ij}^0\}|Q, \{\alpha_i\})P(Q, \{\alpha_i\})}{P(\{\mathbf{s}_{ij}^0\})}. \quad (1)$$

We can ignore the denominator in this expression from now on as it is independent of both Q and $\{\alpha_i\}$. $P(\{\mathbf{s}_{ij}^0\}|Q, \{\alpha_i\})$ is the probabilistic model of the measurement formulation process, and $P(Q, \{\alpha_i\})$ is a prior probability distribution over the prototype surface and possible scan poses.

The process of reconstruction amounts to finding the control vector Q and the rigid body transformations $\{\alpha_i\}$ that maximize the posterior probability in Eq. 1. We adopt the standard technique of optimizing the negative logarithm of the likelihood:

$$\arg \min_{Q, \{\alpha_i\}} \left\{ -\log P(\{\mathbf{s}_{ij}^0\}|Q, \{\alpha_i\}) - \log P(Q, \{\alpha_i\}) \right\}. \quad (2)$$

To simplify Eq. 2, we make the standard assumption that

© The Eurographics Association 2007.

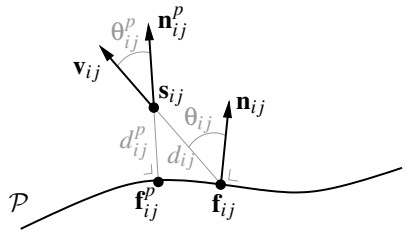

Figure 3: *Prototype surface $\mathcal{P}$ and an input sample $\mathbf{s}_{ij}$. The footpoint $\mathbf{f}_{ij}$ along the viewing ray $\mathbf{v}_{ij}$ is the most likely originating point on the surface. However, to ease computations, we approximate the likelihood using the distance and angle parameters derived from the orthogonally projected footpoint $\mathbf{f}^p_{ij}$.*

data points $\mathbf{s}^0_{ij}$ are sampled independently. Moreover, we assume that the distribution of the prototype surface and the distribution of scan poses are independent. As a further simplification, we assume that scan poses are distributed uniformly. We can now rewrite Eq. 2 as:

$$\arg\min_{Q,\{\alpha_i\}} \sum_{i=1}^{N}\sum_{j=1}^{N_i} \left\{ -\log P(\mathbf{s}^0_{ij}|Q,\alpha_i) \right\} - \log P(Q). \quad (3)$$

Thus, the argument to be minimized decomposes into two additive parts, namely the likelihood terms $-\log P(\mathbf{s}^0_{ij}|Q,\alpha_i)$, and the prior term $-\log P(Q)$. The prior term considers the smoothness of the prototype surface and will be discussed in Sec. 4.2 once the representation of the prototype surface is known.

The likelihood term formulates the probability of the measurement, given the prototype surface and viewing parameters. An accurate model for this problem tends to be very complicated [Cur97]. In this paper, we derive a model that takes the viewing parameters and local surface geometry into account, while allowing efficient optimization of Eq. 3.

The likelihood term for a sample $\mathbf{s}^0_{ij}$ is derived as follows (see also Fig. 3). Given the viewing direction $\mathbf{v}_{ij}$ of the (transformed) sample $\mathbf{s}_{ij}$, we can find the intersection of this ray with the prototype surface $\mathcal{P}$. Denote this intersection point as $\mathbf{f}_{ij}$ and the local surface normal as $\mathbf{n}_{ij}$. As is described in [DTB06], the deviation of a sample from its original surface point is roughly along the viewing direction. Thus, we assume that $\mathbf{s}_{ij}$ is sampled from $\mathbf{f}_{ij}$:

$$-\log P(\mathbf{s}^0_{ij}|Q,\alpha_i) = \sigma_{ij} d_{ij}^2, \quad (4)$$

where $d_{ij} = \|\mathbf{s}_{ij} - \mathbf{f}_{ij}\|$ and σ_{ij} is the variance which depends on the prototype's geometry and the viewing direction. In this paper, we model σ_{ij} to be proportional to the cosine of the angle between the viewing direction $\mathbf{v}_{ij}$ and the local surface normal $\mathbf{n}_{ij}$ at $\mathbf{f}_{ij}$:

$$\sigma_{ij} = (\cos\theta_{ij})^{\gamma}, \quad (5)$$

where γ is a user-specified parameter which controls the decay of the variance. The proposed likelihood model agrees with the intuition that variance is small if θ_{ij} is small and vice-versa.

In practice, determining the intersection point $\mathbf{f}_{ij}$ given the prototype surface and the sample point's viewing direction ray, is time-consuming and we approximate above derivation by using the footpoint $\mathbf{f}^p_{ij}$, obtained by orthogonally projecting $\mathbf{s}_{ij}$ onto the surface. This yields the angle $\theta^p_{ij} \approx \theta_{ij}$, distance $d^p_{ij} \approx d_{ij}\cos\theta_{ij}$ (see Fig. 3), and thus the final likelihood expression:

$$-\log P(\mathbf{s}^0_{ij}|Q,\alpha_i) = (d^p_{ij})^2 (\cos\theta^p_{ij})^{\gamma-2}. \quad (6)$$

Thus, a measurement point has a high probability if it lies close to the prototype surface and if the viewing (or scanning) direction is almost perpendicular to the surface. If no viewing information is available, we will set γ to 2, so that the dependence on viewing direction disappears. We refer to Sec. 6 for more discussion on the choice of γ.

3. Prototype Surface Definition

The prototype surface is defined as a collection of quadratic patches, defined on octree cells, which are blended together using appropriate weight functions to obtain a smooth implicit surface definition, similar to [OBA*03]. For efficiency reasons, we treat the prototype surface during optimization as the collection of quadratic patches restricted to their control cubes (i.e., we do not blend them together). We first discuss this representation and then discuss how these patches are initialized given the scan data points. Next, we show how patches can be combined using partition of unity weight functions to yield a smooth surface for reconstruction purposes. Finally, we describe how the prototype can be reinitialized after octree refinement.

3.1. Quadric Patches

The quadric patches are defined locally on octree cells. Given an octree of depth N, we call a subset S_c of its leaf nodes *control cubes*. Conceptually, these are the leaf nodes which intersect with the surface of the original model. We will show in Sec. 3.2 how these control cubes are obtained. We denote the control cubes by O_I, where $I = i,j,k$, $0 \le i,j,k \le 2^N - 1$ corresponds to the indices of the cell in the corresponding uniform grid.

In each control cube, we define a local coordinate system and a quadratic bivariate function which approximates the local geometry (see also Fig. 4). The local coordinate frame is given by the orthonormal vectors $(\mathbf{e}^1_I, \mathbf{e}^2_I, \mathbf{n}_I)$, where $\mathbf{n}_I$ is chosen to be the surface normal direction and $\mathbf{e}^{1,2}_I$ describe corresponding tangent vectors. The origin of the coordinate system is fixed at $\mathbf{o}_I$. We explain in Sec. 3.2 how the origin is determined. In this local frame, the quadric is expressed as the function:

$$z(x,y) = \frac{1}{2}(a_I x^2 + 2b_I xy + c_I y^2 + d_I). \quad (7)$$

© The Eurographics Association 2007.

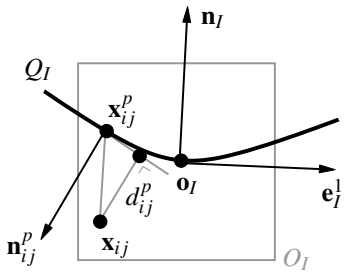

Figure 4: *Control quadric Q_I in the control cube O_I. The quadric is defined with respect to a local orthonormal frame at origin $\mathbf{o}_I$. The distance of a point $\mathbf{x}_{ij}$ to the quadric is approximated as the distance d_{ij}^p to the tangent plane at $\mathbf{x}_{ij}^p$ with normal $\mathbf{n}_{ij}^p$.*

We will not restrict the control quadric's center $\mathbf{o}_I + d_I\mathbf{n}_I$ to lie in its associated control cube. This offers more flexibility during the optimization process. As will be discussed in Sec. 3.4, the refinement step will reinitialize the quadrics and thus constrain them to the correct control cube.

As will be shown in Sec. 4, during the optimization process, we want to minimize the distance between the range scans and the prototype surface. Hence, we need an efficient way to compute point–to–quadric distances. Because computing these distances is at the core of our algorithm, we use following efficiently computable approximation (see also Fig. 4). Given a point $\mathbf{x}_{ij} = (x_{ij}, y_{ij}, z_{ij})$ in the local coordinate frame, we obtain its projection on the quadric along the z-axis: $\mathbf{x}_{ij}^p = (x_{ij}, y_{ij}, z(x_{ij}, y_{ij}))$. The (unnormalized) normal vector at this point to the quadric is given by $\mathbf{n}_{ij}^p = (a_I x_{ij} + b_I y_{ij}, b_I x_{ij} + c_I y_{ij}, -1)$. Finally, we approximate the distance to the quadratic surface by the distance to the tangent plane at $\mathbf{x}_{ij}^p$ (cf. [Tau94]):

$$d(\mathbf{x}_{ij}, Q_I) = \overline{\mathbf{n}}_{ij}^p \cdot (\mathbf{x}_{ij} - \mathbf{x}_{ij}^p) = \frac{z(x_{ij}, y_{ij}) - z_{ij}}{\|\mathbf{n}_{ij}^p\|}, \qquad (8)$$

with $\overline{\mathbf{n}}_{ij}^p = \mathbf{n}_{ij}^p / \|\mathbf{n}_{ij}^p\|$.

3.2. Initialization

We first determine the octree domain by computing a slightly enlarged bounding cube of the input data points. Given an initial depth $d_{\min}$, the octree is built by recursively inserting points. Leaf nodes which contain data points are marked as control cubes. For each cube O_I, we apply a range query to collect all the points within a distance to its center of at most 3 times the size of the leaf node. We use these data points to determine the control parabola Q_I. To ensure a stable fitting, we remove a control cube if the number of data points found is less than 6.

Given the data points, the reference frame $(\mathbf{e}_I^1, \mathbf{e}_I^2, \mathbf{n}_I)$ is estimated using the technique proposed in [HDD*92]. The weighted centroid of the data points (with respect to the cube's center) gives us the frame's origin $\mathbf{o}_I$. Note that the initialization is performed at a coarse octree level. And thus, even in the case of noisy and incomplete scan data, we are able to obtain consistently oriented frames.

We then optimally fit the proposed quadric to the data points. This amounts to determining the parameters a_I, b_I, c_I and d_I of Eq. 7 through minimization of the following quadratic function:

$$\sum_{i=0}^{k} w_i \left(\frac{1}{2}a_I x_i^2 + b_I x_i y_i + \frac{1}{2}c_I y_i^2 + d_I - z_i\right)^2, \qquad (9)$$

where w_i are the same weights as used for the centroid and reference frame computations. Note that we use the algebraic distance, instead of the approximative orthogonal distance as discussed in Sec. 3.1.

3.3. Implicit Surface

The above discussed collection of patches is used in our framework during the scan alignment and prototype optimization process. However, as can be easily seen, the union of these restricted quadratic patches, does not yield a continuous or smooth surface, necessary for reconstruction. Therefore, the final surface is defined as the zero level set $f^{-1}(0)$ of the implicit function f obtained by smoothly blending distance functions of overlapping quadrics:

$$f(\mathbf{x}) = \sum_I B_I(\mathbf{x}) d(\mathbf{x}, Q_I) = 0. \qquad (10)$$

The weights $B_I(\mathbf{x})$ are chosen to be the quadratic B-spline basis functions defined as a multiplication of uniform quadratic B-spline basis functions for each coordinate axis: $B_I(\mathbf{x}) = B_i(x)B_j(y)B_k(z)$. The distance function is the approximation derived in the previous section. The $B_I(\mathbf{x})$ are compactly supported by $[i-1, i+2] \times [j-1, j+2] \times [k-1, k+2]$ and form a partition of unity.

3.4. Refinement

After optimizing the control quadrics and the scan alignments, we refine the prototype surface by increasing the octree depth by one. This allows us to gradually include more detail in the prototype and to improve the scan alignments. The refinement process is very similar to the initialization described in Sec. 3.2. However, instead of fitting the new quadrics against the data points, we consider the implicit surface definition of Sec. 3.3 defined on the old octree as the surface to approximate.

Control cubes are defined in the new octree as those cells which intersect the zero level set as defined in Eq. 10. We can easily detect this by evaluating f at the cell's corners.

Next, we determine the local frame by projecting the center of the control cube onto the zero level set. This can be easily performed by walking in the gradient direction of the level set function. This yields the origin $\mathbf{o}_I$ of the local frame.

© The Eurographics Association 2007.

The z-axis is again defined by the local surface normal, now given by: $\mathbf{n}_I = \nabla f(\mathbf{o}_I)/\|\nabla f(\mathbf{o}_I)\|$. The tangent vectors $\mathbf{e}_I^{1,2}$ are again chosen to form an orthonormal basis with $\mathbf{n}_I$.

Given the local frame, the shape of the control quadric is obtained by minimizing following quadratic function:

$$\sum_{J\in N(I)} (\frac{1}{2}a_I x_{\mathbf{o}_J}^2 + b_I x_{\mathbf{o}_J} y_{\mathbf{o}_J} + \frac{1}{2}c_I y_{\mathbf{o}_J}^2 - z_{\mathbf{o}_J})^2, \qquad (11)$$

where the summation is over the neighbors of control cube I and $\mathbf{o}_J = (x_{\mathbf{o}_J}, y_{\mathbf{o}_J}, z_{\mathbf{o}_J})$ is the frame center of the neighboring cube, defined in the local frame of cube I. Note that we set d_I to zero, and hence force the quadric to pass through $\mathbf{o}_I$, which indeed lies on the zero level set.

4. Optimization

Given the specific prototype surface representation of Sec. 3, we can now refine the optimization problem. We first discuss the likelihood term for a measurement point. Next, we derive a suitable prior term for the prototype surface. Finally, we give the resulting optimization problem.

4.1. Likelihood Term

Recall that the likelihood term describes the probability of a measurement point $\mathbf{s}_{ij}^0$, given the prototype surface and corresponding scan transformation parameters. We derived following expression in Sec. 2:

$$-\log P(\mathbf{s}_{ij}^0|Q,\alpha_i) = (d_{ij}^p)^2(\cos\theta_{ij}^p)^{\gamma-2}. \qquad (12)$$

Again, as in Fig. 4, we approximate the normal at the closest point on the prototype surface by the (normalized) normal of the tangent plane $\bar{\mathbf{n}}_{ij}^p$ found by projecting the transformed measurement point $\mathbf{x}_{ij}$ (expressed in the local coordinate frame) along the z-axis onto the closest quadric $Q_{I_{ij}}$. Similarly, the distance d_{ij}^p is approximated by the distance to this tangent plane. This yields for the likelihood term:

$$-\log P(\mathbf{s}_{ij}^0|Q,\alpha_i) = d(\mathbf{x}_{ij},Q_{I_{ij}})^2(\bar{\mathbf{n}}_{ij}^p\cdot\mathbf{v}_{ij})^{\gamma-2} = \Phi_{ij}^2. \quad (13)$$

The nearest quadric query is accomplished using a similar data structure as the d^2-tree [LPZ03]. Each octree node stores the nearest control quadric to its center and data points simply use the quadric associated to the octree cell they lie in. We initialize this data structure by assigning each quadric to the corresponding control cube and by propagating this information to neighboring octree cells using a fast sweeping algorithm [hRT02].

4.2. Prior Term

The prior term constrains the prototype surface. In this paper, the prior term consists of two parts, the smoothness T_s and consistency T_c:

$$-\log P(Q) = \lambda_1/|S_c|T_s + \lambda_2/|S_c|T_c, \qquad (14)$$

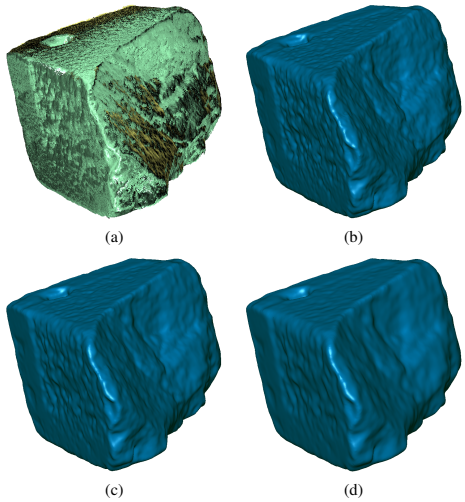

Figure 5: *Different reconstructions for the O088 model. The prior term consists of a smoothness and a consistency term. The smoothness is varied by λ_1 and the consistency by λ_2. By changing these weights, we can obtain a different smoothness of the reconstructed model. (a) The input consisting of 15 scans. (b) $\lambda_1 = 1.8e5$, $\lambda_2 = 1.8e4$. (c) $\lambda_1 = 1.8e6$, $\lambda_2 = 1.8e5$. (d) $\lambda_1 = 1.8e7$, $\lambda_2 = 1.8e6$.*

with λ_1 and λ_2 scaling parameters to trade-off both contributions. We refer to Fig. 5 for an illustration of the effect of varying these parameters. The scaling by $1/|S_c|$ is necessary, because after each refinement the number of control quadrics $|S_c|$ will increase as well as the contribution of the prior term if no such scaling is done. This would result in incorrectly scaling down the likelihood term after each refinement step.

The smoothness term T_s approximates the integral of the squared sum of principal curvatures over the surface, which is given by

$$T_s = \frac{1}{2}\sum_{I\in S_c}(a_I^2 + 2b_I^2 + c_I^2) = \sum_{I\in S_c}\Psi_I^T\Psi_I, \qquad (15)$$

where $\Psi_I = (\frac{1}{\sqrt{2}}a_I, b_I, \frac{1}{\sqrt{2}}c_I)$ is introduced to simplify further discussion. The summation is over all control cubes. The consistency term will prevent the trivial smooth solution, i.e., avoid incorrectly reconstructing flat surfaces.

The consistency term T_c is introduced to force neighboring quadrics to join nicely. More specifically, we want the origin of each quadric to lie as close as possible to quadrics in neighboring control cubes:

$$T_c = \sum_{I\in S_c}\sum_{J\in N(I)} w_{IJ}d^2(\mathbf{o}_I + d_I\mathbf{n}_I, Q_J) = \sum_{I\in S_c}\sum_{J\in N(I)} w_{IJ}\Psi_{IJ}^2. \qquad (16)$$

Here, the outer sum is over all control quadrics and the inner sum is over all neighboring control quadrics. More precisely, we say that two control quadrics Q_I and $Q_{I'}$ are adjacent if $|I-I'| = |i-i'|+|j-j'|+|k-k'| \le 2$. The weight func-

© The Eurographics Association 2007.

tions w_{IJ} are introduced to deal with sharp edges or thin features. Here we use a simple heuristic by choosing $w_{IJ} = 1$ if $\mathbf{n}_I \cdot \mathbf{n}_J > 0$ and $w_{IJ} = \varepsilon$ otherwise. In this paper, we set $\varepsilon = 0.01$ for all examples.

4.3. Objective Function

Substituting Eq. 13, 15 and 16 into Eq. 3 yields following energy function:

$$E = \sum_{i=1}^{N} \sum_{j=1}^{N_i} \Phi_{ij}^2 + \frac{1}{|S_c|} \sum_{I \in S_c} (\lambda_1 \Psi_I^T \Psi_I + \lambda_2 \sum_{J \in N(I)} w_{IJ} \Psi_{IJ}^2). \tag{17}$$

The goal is to obtain the control quadric parameters and and range scan rigid transformations which minimize above energy function. The energy function is non-linear in the parametrization and we propose to use preconditioned non-linear conjugate gradient, because it is both fast, stable and memory efficient. Another possibility would be to use a Gauss-Newton method. However, memory storage and computation time become a bottleneck for large-scale problems, e.g., we are typically dealing with more than 200k control quadrics.

We refer to Appendix A for a detailed discussion of our parametrization of the unknowns Q and α_i and to Appendix B for a detailed derivation of the preconditioners used in the conjugate gradient algorithm.

5. Summary

To summarize, our surface reconstruction pipeline (as depicted in Fig. 1 and illustrated in Fig. 2) starts from a roughly aligned set of scans and initializes the prototype surface as described in Sec. 3.2. Next, we optimize the prototype's shape and the scan alignments by minimizing the energy function which is summarized in Sec. 4.3. After optimization, we refine the prototype by subdividing the octree one more level and by recomputing the control cubes and quadrics as discussed in Sec. 3.4. This refinement yields suboptimal control quadrics and we iterate and invoke a new optimization which in turn yields optimal quadrics and rigid scan transforms for this octree level. This alternation of optimization and refinement is repeated until convergence. During this process we prune outliers by deleting sample points which lie at a threshold distance away from the prototype (we set this distance to 4 times the width of the smallest octree cubes). Finally, the algorithm outputs a triangular mesh from the implicit surface defined in Sec. 3.3 using the octree version [WG92, WKE99] of the marching cubes algorithm [LC87].

6. Results and Discussion

We conducted a series of experiments to evaluate our approach on different synthetic and real-world data sets. All results are obtained on a 3GHz Intel CPU with 2GB of memory. Table 1 summarizes the statistics of the input data, the parameters used, the properties of the resulting model and the total reconstruction time. To obtain a rough alignment of the input scans, we used the multi-view global matching method of [HFG*06]. Due to lack of features in the O005 model, 4 scans were matched manually and the poses of all 119 scans were then computed using the local registration method of [Neu97]. If we are not given viewing directions per sample point or when scanning was performed under difficult circumstances such as for the archaeological pieces, we set $\gamma = 2$, so that the dependence on viewing direction in Eq. 13 disappears, and likelihood only varies with distance to the surface. For the the data where viewing directions are available, experiments showed that $\gamma = 4$ is a good choice.

	BUNNY	DRAGON	O088	O209	O005
#scans	10	71	15	27	119
#points	0.36M	1.8M	5M	11M	59M
λ_1	6e5	4.5e5	1.8e5	1.4e6	3.8e6
λ_2	6e1	4.5e1	1.8e4	7e3	1.9e4
γ	4	4	2	2	2
$d_{\min}$	8	8	7	8	7
$d_{\max}$	9	9	8	9	10
#quadrics	600k	450k	180k	715k	1891k
#triangles	1624k	1124k	331k	1237k	3541k
time	16m20s	29m21s	15m45s	45m11s	6h21m

Table 1: *Statistics and parameters for the different examples shown in this paper.*

Fig. 2 shows the different steps in the reconstruction of the O005 model. This model, as well as the O088 and O209 models, are from an archaeological data set, which are particularly challenging, because of the incomplete and noisy scans. The input consists of 119 roughly aligned scans and a total of 59M data points. From this initial point cloud, we compute the prototype surface at octree depth 7. Then we iteratively optimize and refine until octree depth 10 is reached. The final output of the algorithm is the reconstructed triangle mesh (consisting of 3541k triangles) and the final poses of the range scans. We will give for this example a detailed reconstruction quality comparison with the state-of-the-art mesh reconstruction method of [KBH06] below.

Fig. 5 shows different reconstruction results of the O088 model for varying coefficients of the prior term. As discussed in Sec. 4.2, our prior term is composed of two parts: the smoothness and consistency term. The contribution of both with respect to the likelihood term can be varied by adjusting the λ_1 and λ_2 parameters. Fig. 5 illustrates the effect for different choices for these parameters. As can be seen in the figure, by increasing these parameters, the final reconstructed object becomes smoother.

6.1. Comparison to Prior Work

As our approach jointly aligns range scans and reconstructs triangular meshes, we present comparisons to state-of-the-art registration and mesh reconstruction algorithms.

© The Eurographics Association 2007.

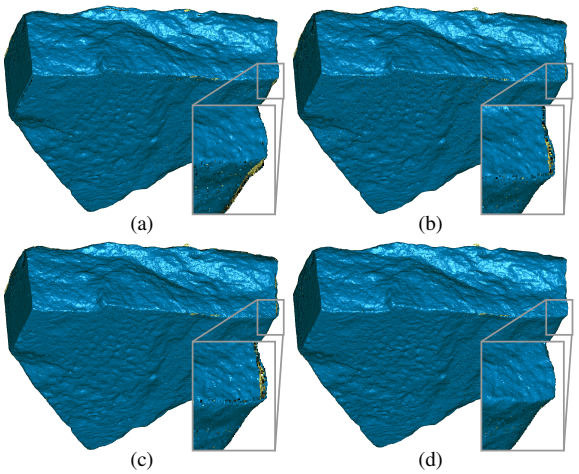

Figure 6: *Our approach exhibits better performance than standard registration algorithms in aligning the O209 model. (a) The input obtained from multi-view global matching. (b) Registration result of [Neu97]. (c) Registration result of [THL03]. (d) Registration result using our approach.*

Registration Quality Fig. 6 compares our approach with standard registration algorithms. The input data are the 27 scans of the O209 model. We compare our method with the multi-view version of the ICP method [BM92], which is described in [Neu97], where registration is performed by minimizing the sum of pairwise distances between overlapping scans, and the method of [THL03], which aligns scans to an implicit surface defined by averaging distance fields. We also applied the techniques described in [RL01] to improve the performance of these two algorithms. Because the input scans of the O209 model are very noisy and bumpy, both methods get stuck in local minima, resulting in poor alignment quality. However, as we align scans to a smooth prototype surface with a well-defined distance field, we obtain improved registration quality as can be seen on the figure.

Reconstruction Quality We compare our surface reconstruction quality, for a different number of examples, with the Poisson Surface Reconstruction (PSR) method of [KBH06]. The surface normals used in the PSR method were obtained using the technique presented in [HDD*92]. We also tried other surface reconstruction algorithms such as the ones presented in [ACK01, OBA*03], however these methods failed due to the large input size and high noise levels. To make a fair comparison, we take the alignment obtained by our approach as the input for the PSR algorithm. Fig. 7 shows the result for the dragon model. Both algorithms produce similar quality results, although at different octree depths (depth 9 for ours compared to depth 10 for PSR).

However, when the input scans are noisy, we found that our method is more accurate and stable than PSR. This is illustrated in Fig. 8 for synthetic data. We generated 10 virtual range scans from the Stanford bunny model, and added

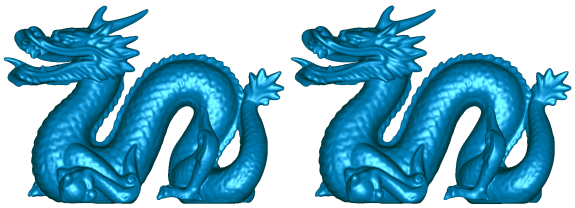

Figure 7: *Reconstruction of the dragon model. Left: Result of our method at octree depth 9. Right: Result of [KBH06] at octree depth 10.*

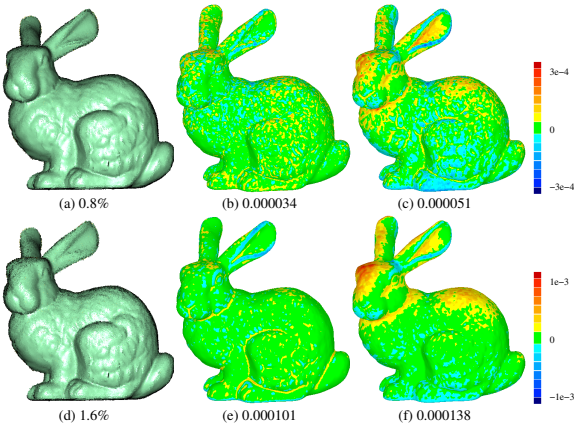

Figure 8: *Comparison of the reconstruction quality of our method (middle column) with the method presented in [KBH06] (right column). The top row shows the result for noise level* 0.8%*, the bottom row for noise level* 1.6%*. The standard deviation is color coded on the resulting geometry (green is zero standard deviation, and red and dark blue are high deviations).*

different amounts of noise. In the top row of the figure, we show the standard deviation for a noise level of 0.8%, while in the bottom row we increase the noise to 1.6%. As can be seen, both methods reconstruct roughly the same geometry. However, our reconstruction is closer to the original model as indicated by the deviations. Moreover, when increasing the noise level up to 3.2%, our method is still able to reconstruct the bunny, but PSR fails because it is hard to estimate good normals from the noisy input point cloud.

Fig. 9, finally, compares our method to PSR for real-world data that has significant noise, outliers and holes. As indicated by the arrows on the figure, our method has better reconstruction quality near corners and in noisy regions.

6.2. Discussion

The most time consuming part of our algorithm is the computation of the derivatives of the energy function as described in Appendix B. Our current implementation allows to process 0.8M data points on average per second. Experiments showed that our approach is computationally more expensive than the PSR algorithm [KBH06]. For the O005 model of 59M points as input, our approach took 6 hours to

© The Eurographics Association 2007.

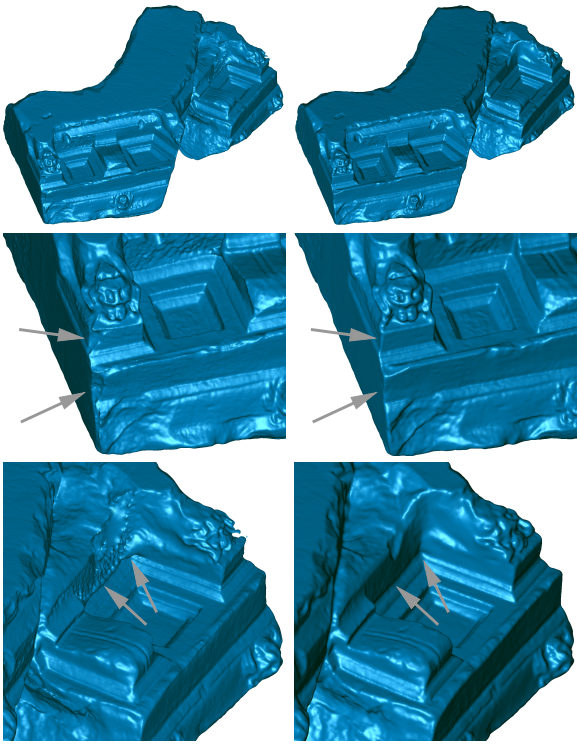

Figure 9: *Comparison of our method (right column) with [KBH06] (left column) for the O005 model. As indicated by the arrows in the two closeups, our method better resolves noisy and missing input data.*

reconstruct a mesh of 3.5M triangles, while the PSR algorithm completed in 1 hour.

When the input data is clean and complete, both PSR and our approach can reconstruct the geometry accurately. However, when measurement errors become more pronounced, our approach becomes superior, mainly because we consider this error in the optimization and try to control the final shape by the priors. In this case, the additional computation time can be justified. Also, when the original model is complicated, standard registration algorithms have difficulties aligning the range scans and we found in practice that our approach yields better results than standard two-stage algorithms which first do registration and then reconstruction.

A limitation of our approach is that we assume a single noise level of the original model. This can result in good detailed reconstructions for one part of the model, but can at the same time smooth out too much detail in other parts. Adapting the smoothing parameters locally should prevent this problem.

7. Conclusions and Future work

We have presented a Bayesian approach for joint surface reconstruction and registration. It takes a set of coarsely aligned range scans as input and produces a watertight solid surface as well as a high-accuracy alignment of the original partial scans. In comparison to previous work, the new approach improves substantially on the accuracy of the reconstruction and alignment results.

There are some opportunities for future research. First of all, one could try to preserve sharp features or preserve user specified topology information during the meshing process. As our goal is to reconstruct high quality 3D models from large data sets, it would be very useful to devise an out-of-core implementation that allows us to handle gigantic data sets such as those from the Digital Michelangelo Project [LPC*00].

Acknowledgments The authors acknowledge the support of the Max Planck Center for Visual Computing and Communication. The second author is funded as a postdoctoral researcher by the Fund for Scientific Research, Flanders (F.W.O.-Vlaanderen) and by a Fellowship of the Belgian American Educational Foundation. We would also like to thank the anonymous reviewers for their valuable comments. The bunny and dragon models are courtesy of The Stanford 3D Scanning Repository. We thank Michael Hofer, Barbara Thuswaldner, Robert Kalasek and Marina Doering from TU Vienna and Hilke Thuer and Friedrich Krinzinger from the Austrian Academy of Sciences for providing us the O005, O088 and O209 models.

References

[ABK98] AMENTA N., BERN M., KAMVYSSELIS M.: A new voronoi-based surface reconstruction algorithm. In *SIGGRAPH '98* (1998), pp. 415–421.

[ACK01] AMENTA N., CHOI S., KOLLURI R. K.: The power crust. In *SMA '01: Proceedings of the sixth ACM symposium on Solid modeling and applications* (2001), pp. 249–266.

[BM92] BESL P. J., MCKAY N. D.: A method for registration of 3-d shapes. *PAMI. 14*, 2 (1992), 239–256.

[BR04] BROWN B., RUSINKIEWICZ S.: Non-rigid range-scan alignment using thin-plate splines. In *Symposium on 3D Data Processing, Visualization, and Transmission* (2004).

[CL96] CURLESS B., LEVOY M.: A volumetric method for building complex models from range images. In *SIGGRAPH '96* (1996), pp. 303–312.

[Cur97] CURLESS B.: *New Methods for Surface Reconstruction from Range Images*. PhD thesis, Stanford University, 1997.

[DTB06] DIEBEL J. R., THRUN S., BRÜNIG M.: A bayesian method for probable surface reconstruction and decimation. *ACM Trans. Graph. 25*, 1 (2006), 39–59.

[GMGP05] GELFAND N., MITRA N. J., GUIBAS L. J., POTTMANN H.: Robust global registration. In *Symposium on Geometry Processing* (2005), pp. 197–206.

[HDD*92] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. In *SIGGRAPH '92* (1992), pp. 71–78.

[HFG*06] HUANG Q.-X., FLÖRY S., GELFAND N., HOFER M.,

© The Eurographics Association 2007.

POTTMANN H.: Reassembling fractured objects by geometric matching. In *SIGGRAPH '06* (2006), pp. 569–578.

[HH03] HUBER D., HEBERT M.: Fully automatic registration of multiple 3d data sets. *Image and Vision Computing 21*, 7 (2003), 637–650.

[HK06] HORNUNG A., KOBBELT L.: Robust reconstruction of watertight 3d models from non-uniformly sampled point clouds without normal information. In *SGP '06* (2006), pp. 41–50.

[hRT02] HSI RICHARD TSAI Y.: Rapid and accurate computation of the distance function using grids. *J. Comput. Phys. 178*, 1 (2002), 175–195.

[JDH*95] JIN H., DUCHAMP T., HOPPE H., MCDONALD J. A., PULLI K., STUETZLE W.: Surface reconstruction from misregistered data. In *Proc. SPIE* (Aug. 1995), Melter R. A., Wu A. Y., Bookstein F. L., Green W. D., (Eds.), vol. 2573, pp. 324–328.

[JWB*06] JENKE P., WAND M., BOKELOH M., SCHILLING A., STRASSER W.: Bayesian point cloud reconstruction. In *Proc. Eurographics 2006* (2006).

[KBH06] KAZHDAN M., BOLITHO M., HOPPE H.: Poisson surface reconstruction. In *SGP '06* (2006), pp. 61–70.

[KLMV05] KRISHNAN S., LEE P. Y., MOORE J. B., VENKATASUBRAMANIAN S.: Global registration of multiple 3d point sets via optimization-on-a-manifold. In *SGP'05* (2005), pp. 187–196.

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH '87* (1987), pp. 163–169.

[LPC*00] LEVOY M., PULLI K., CURLESS B., RUSINKIEWICZ S., KOLLER D., PEREIRA L., GINZTON M., ANDERSON S., DAVIS J., GINSBERG J., SHADE J., FULK D.: The digital michelangelo project: 3d scanning of large statues. In *SIGGRAPH '00* (2000), pp. 131–144.

[LPW06] LIU Y., POTTMANN H., WANG W.: Constrained 3d shape reconstruction using a combination of surface fitting and registration. *CAD 38*, 6 (2006), 572–583.

[LPZ03] LEOPOLDSEDER S., POTTMANN H., ZHAO H.: *The d^2-tree: A hierarchical representation of the squared distance function*. Tech. Rep. 101, Institute of Geometry, March 2003.

[MD97] MONTAGNAT J., DELINGETTE H.: A hybrid framework for surface registration and deformable models. In *CVPR '97* (1997), p. 1041.

[Neu97] NEUGEBAUER P. J.: Reconstruction of real-world objects via simultaneous registration and robust combination of multiple range images. *IJSM 3*, 1-2 (1997), 71–90.

[NRDR05] NEHAB D., RUSINKIEWICZ S., DAVIS J., RAMAMOORTHI R.: Efficiently combining positions and normals for precise 3d geometry. *ACM Trans. Graph. 24*, 3 (2005), 536–543.

[OBA*03] OHTAKE Y., BELYAEV A., ALEXA M., TURK G., SEIDEL H.-P.: Multi-level partition of unity implicits. In *SIGGRAPH '03* (2003), pp. 463–470.

[PHYH06] POTTMANN H., HUANG Q.-X., YANG Y.-L., HU S.-M.: Geometry and convergence analysis of algorithms for registration of 3d shapes. *IJCV 67*, 3 (2006), 277–296.

[RL01] RUSINKIEWICZ S., LEVOY M.: Efficient variants of the ICP algorithm. In *3DIM'01* (2001), pp. 145–152.

[She94] SHEWCHUK J. R.: *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. Tech. rep., 1994.

[SLS*06] SHARF A., LEWINER T., SHAMIR A., KOBBELT L., COHEN-OR D.: Competing fronts for coarse–to–fine surface reconstruction. In *Eurographics* (2006), pp. 389–398.

[Tau94] TAUBIN G.: Distance approximations for rasterizing implicit curves. *ACM Trans. Graph. 13*, 1 (1994), 3–42.

[THL03] TUBIC D., HÉBERT P., LAURENDEAU D.: A volumetric approach for interactive 3d modeling. *Comput. Vis. Image Underst. 92*, 1 (2003), 56–77.

[WG92] WILHELMS J., GELDER A. V.: Octrees for faster isosurface generation. *ACM Trans. Graph. 11*, 3 (1992), 201–227.

[WKE99] WESTERMANN R., KOBBELT L., ERTL T.: Real-time exploration of regular volume data by adaptive reconstruction of isosurfaces. *The Visual Computer 15*, 2 (1999), 100–111.

Appendix A: Parametrization

This section describes the parametrization of the unknowns in the optimization problem of Eq. 17. We also give expressions for the first order derivatives, which are used to build the final system of equations.

Quadric Parametrization

We parametrize each control quadric Q_I near initial values (which are obtained from the previous iteration step or in the beginning after initialization). Each quadric has 6 degrees of freedom. Basically, we allow the normal direction $\mathbf{n}_I$ of the local frame to move in the plane spanned by $\mathbf{e}_I^1$ and $\mathbf{e}_I^2$. This yields two degrees of freedom, given by t_I^1 and t_I^2:

$$\mathbf{n}_I = \frac{\mathbf{n}_I^0 + \mathbf{e}_I^{1\,0} t_I^1 + \mathbf{e}_I^{2\,0} t_I^2}{\|\mathbf{n}_I^0 + \mathbf{e}_I^{1\,0} t_I^1 + \mathbf{e}_I^{2\,0} t_I^2\|}, \tag{18}$$

where $(\mathbf{e}_I^{1\,0}, \mathbf{e}_I^{2\,0}, \mathbf{n}_I^0)$ is the initial frame before optimization. The new tangent vectors $\mathbf{e}_I^1$ and $\mathbf{e}_I^2$ are easily obtained from $\mathbf{n}_I$ (and hence also parametrized by t_I^1 and t_I^2):

$$\mathbf{e}_I^1 = \frac{\mathbf{e}_I^{1\,0} - \mathbf{n}_I \mathbf{n}_I^T \mathbf{e}_I^{1\,0}}{\|\mathbf{e}_I^{1\,0} - \mathbf{n}_I \mathbf{n}_I^T \mathbf{e}_I^{1\,0}\|}, \quad \mathbf{e}_I^2 = \mathbf{n}_I \times \mathbf{e}_I^1. \tag{19}$$

Our optimization solver requires derivatives with respect to the unknowns. This yields (with $1 \le k \le 2$):

$$\frac{\partial \mathbf{n}_I}{\partial t_I^k} = \frac{(I_3 - \mathbf{n}_I \mathbf{n}_I^T)\mathbf{e}_I^{k\,0}}{\|\mathbf{n}_I^0 + \mathbf{e}_I^{1\,0} t_I^1 + \mathbf{e}_I^{2\,0} t_I^2\|}, \tag{20}$$

$$\frac{\partial \mathbf{e}_I^1}{\partial t_I^k} = \frac{(\mathbf{e}_I^1 \mathbf{e}_I^{1\,T} - I_3)(\mathbf{n}_I^T \mathbf{e}_I^{1\,0} + \mathbf{n}_I \mathbf{e}_I^{1\,0\,T})}{\|\mathbf{e}_I^{1\,0} - < \mathbf{e}_I^{1\,0}, \mathbf{n}_I > \mathbf{n}_I\|} \frac{\partial \mathbf{n}_I}{\partial t_I^k}, \tag{21}$$

$$\frac{\partial \mathbf{e}_I^2}{\partial t_I^k} = -\mathbf{e}_I^1 \times \frac{\partial \mathbf{n}_I}{\partial t_I^k} + \mathbf{n}_I \times \frac{\partial \mathbf{e}_I^1}{\partial t_I^k}. \tag{22}$$

The 4 remaining degrees of freedom define the quadric in its local coordinate frame:

$$a_I = a_I^0 + t_I^3,\ b_I = b_I^0 + t_I^4,\ c_I = c_I^0 + t_I^5,\ d_I = d_I^0 + t_I^6. \tag{23}$$

© The Eurographics Association 2007.

Here, the initial quadric is defined by a_I^0, b_I^0, c_I^0 and d_I^0. The derivatives are trivial.

Rigid Transform Parametrization

We parametrize $\alpha_i = (R_i, T_i)$ around an initial transformation $\alpha_i^0 = (R_i^0, T_i^0)$. The rotation R_i is parametrized using standard lie-group parametrization. This gives, together with its derivatives ($1 \le k \le 3$):

$$R_i(\mathbf{c}_i) = exp(\mathbf{c}_i\times)\cdot R_i^0, \quad \frac{\partial R_i}{\partial c_{ik}} = E_k R_i, \tag{24}$$

where $\mathbf{c}_i = (c_{i1}, c_{i2}, c_{i3})$ are the parameters and E_1, E_2, E_3 is the basis of 3×3 antisymmetric matrices given by:

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

The parametrization of T_i and its derivatives are given by

$$T_i(\bar{\mathbf{c}}_i) = T_i^0 + \bar{\mathbf{c}}_i, \quad \frac{\partial T_i}{\partial \bar{\mathbf{c}}_i} = I_3. \tag{25}$$

This yields a total of 6 unknowns for each rigid body transformation.

Because the first range scan is fixed, we finally have $6(|S_c|+N-1)$ unknowns to solve for. Recall that N is the number of scans and $|S_c|$ is the number of control quadrics.

Appendix B: Preconditioned Non-Linear Conjugate Gradient

To optimize the derived energy function of Eq. 17, we use a preconditioned non-linear conjugate gradient (cg) algorithm with Polak-Ribière update. The algorithm is described in detail in [She94]. Here, we limit the discussion to the construction of the preconditioner.

For the non-linear cg algorithm, we use an approximation H_{GN} of the Hessian to precondition the gradient ∇E, used for line-searching. As will be shown below, H_{GN} has following block structure:

$$H_{GN} = \begin{pmatrix} A_{6|S_c|\times 6|S_c|} & B_{6|S_c|\times 6(N-1)} \\ B^T_{6|S_c|\times 6(N-1)} & C_{6(N-1)\times 6(N-1)} \end{pmatrix}. \tag{26}$$

Then, we construct the preconditioner as:

$$M = \begin{pmatrix} diag(A_{6|S_c|\times 6|S_c|}) & 0 \\ 0 & C_{6(N-1)\times 6(N-1)} \end{pmatrix}. \tag{27}$$

Note that, $C_{6(N-1)\times 6(N-1)}$ consists of $(N-1)$ 6×6 diagonal blocks, so the inverse of M can be computed efficiently. We also tried other preconditioning matrices and found that M exhibits the best tradeoff between memory concern and convergence rate of the optimization.

The line search is accomplished using a Gauss-Newton method. We found in practice that this strategy is extremely efficient. For all the models we have tested, we only needed 2-3 Gauss-Newton iterations on average to finish a line search.

The gradient ∇E of the energy and Gauss-Newton approximation H_{GN} of its Hessian are computed by using the chain rule. To simplify our discussion, we denote the derivative of a function f in terms of the parameters as ∇f. If f is a vector of size m, then ∇f is a matrix of size $6(|S_c|+N-1)\times m$.

Our goal here is to represent ∇E and H_{GN} in terms of $\nabla a_I, \nabla b_I, \nabla c_I, \nabla d_I, \nabla \mathbf{n}_I, \nabla \mathbf{e}^1, \nabla \mathbf{e}^2, \nabla \mathbf{c}_i, \nabla \bar{\mathbf{c}}_i$ which are given directly by their parameterizations.

We can first compute ∇E and H_{GN} as

$$\begin{aligned} \nabla E &= 2\sum_{i=1}^{N}\sum_{j=1}^{N_i} \nabla\Phi_{ij}\Phi_{ij} + 2\tfrac{1}{|S_c|}\sum_{I\in S_c}(\lambda_1 \nabla\Psi_I\Psi_I \\ &+ \lambda_2 \sum_{J\in N(I)} w_{IJ}\nabla\Psi_{IJ}\Psi_{IJ}), \end{aligned} \tag{28}$$

and

$$\begin{aligned} H_{GN} &= \sum_{i=1}^{N}\sum_{j=1}^{N_i} \nabla\Phi_{ij}\nabla\Phi_{ij}^T + \tfrac{1}{|S_c|}\sum_{I\in S_c}(\lambda_1 \nabla\Psi_I\nabla\Psi_I^T \\ &+ \lambda_2 \sum_{J\in N(I)} w_{IJ}\nabla\Psi_{IJ}\nabla\Psi_{IJ}^T). \end{aligned} \tag{29}$$

As $\nabla\Psi_I$ is trivial, we only discuss how to compute $\nabla\Phi_{ij}$ and $\nabla\Psi_{IJ}$ in the following. Applying the chain rule, we can expand $\nabla\Phi_I$ as

$$\begin{aligned} \nabla\Phi_I &= \nabla d(\mathbf{s}_{ij}, Q_{I_{ij}})\cdot(\bar{\mathbf{n}}_{ij}^p\cdot\mathbf{v}_{ij})^{\frac{\gamma-2}{2}} \\ &+ \tfrac{\gamma-2}{2} d(\mathbf{s}_{ij}, Q_{I_{ij}})\cdot(\bar{\mathbf{n}}_{ij}^p\cdot\nabla\mathbf{v}_{ij} + \mathbf{v}_{ij}\cdot\nabla\bar{\mathbf{n}}_{ij}^p) \end{aligned} \tag{30}$$

$\nabla\mathbf{v}_{ij}$ and $\nabla\bar{\mathbf{n}}_{ij}^p$ can be calculated easily as

$$\nabla\mathbf{v}_{ij} = -(\mathbf{v}_{ij}\times)\nabla\mathbf{c}_i, \tag{31}$$

$$\begin{aligned} \nabla\bar{\mathbf{n}}_{ij}^p = ((1+\mathbf{n}_{ij}^{p\,y}\mathbf{n}_{ij}^{p\,y})\nabla\mathbf{n}_{ij}^{p\,x} - \mathbf{n}_{ij}^{p\,x}\mathbf{n}_{ij}^{p\,y}\nabla\mathbf{n}_{ij}^{p\,y}, \\ (1+\mathbf{n}_{ij}^{p\,x}\mathbf{n}_{ij}^{p\,x})\nabla\mathbf{n}_{ij}^{p\,y} - \mathbf{n}_{ij}^{p\,x}\mathbf{n}_{ij}^{p\,y}\nabla\mathbf{n}_{ij}^{p\,x}, \\ \mathbf{n}_{ij}^{p\,x}\nabla\mathbf{n}_{ij}^{p\,x} + \mathbf{n}_{ij}^{p\,y}\nabla\mathbf{n}_{ij}^{p\,x})/\|\mathbf{n}_{ij}^p\|^3. \end{aligned} \tag{32}$$

where

$$\begin{aligned} \nabla\mathbf{n}_{ij}^{p\,x} &= -(a_{I_{ij}}\nabla x_{ij} + x_{ij}\nabla a_{I_{ij}} + b_{I_{ij}}\nabla y_{ij} + y_{ij}\nabla b_{I_{ij}}) \\ \nabla\mathbf{n}_{ij}^{p\,y} &= -(b_{I_{ij}}\nabla x_{ij} + x_{ij}\nabla b_{I_{ij}} + c_{I_{ij}}\nabla y_{ij} + y_{ij}\nabla c_{I_{ij}}) \end{aligned}$$

Given a point $\mathbf{x} = (x, y, z)$ expressed in the local coordinate system of its nearest control quadratics Q_I. We can compute $\nabla d(\mathbf{x}, Q_I)$ as

$$\begin{aligned} \nabla d(\mathbf{x}, Q_I) = \tfrac{1}{\|\mathbf{n}^p\|}(\nabla d_I - \nabla z - (\mathbf{n}^{p\,x} - \tfrac{\mathbf{n}^{p\,x}a_I + \mathbf{n}^{p\,y}b_I}{\|\mathbf{n}^p\|}d(\mathbf{x}, Q_I))\nabla x \\ -(\mathbf{n}^{p\,y} - \tfrac{\mathbf{n}^{p\,x}b_I + \mathbf{n}^{p\,y}c_I}{\|\mathbf{n}^p\|}d(\mathbf{x}, Q_I))\nabla y \\ -(\tfrac{1}{2}x^2 + \tfrac{(\mathbf{n}^{p\,x}a_I + \mathbf{n}^{p\,y}b_I)x}{\|\mathbf{n}^p\|}d(\mathbf{x}, Q_I)))\nabla a_I \\ -(xy + \tfrac{(\mathbf{n}^{p\,x}a_I + \mathbf{n}^{p\,y}b_I)y + (\mathbf{n}^{p\,x}b_I + \mathbf{n}^{p\,y}c_I)x}{\|\mathbf{n}^p\|}d(\mathbf{x}, Q_I)))\nabla b_I \\ -(\tfrac{1}{2}y^2 + \tfrac{(\mathbf{n}^{p\,x}b_I + \mathbf{n}^{p\,y}c_I)y}{\|\mathbf{n}^p\|}d(\mathbf{x}, Q_I)))\nabla c_I) \end{aligned} \tag{33}$$

Recall that the local coordinates of $\mathbf{s}_{ij}$ in $Q_{I_{ij}}$ are given by (x_{ij}, y_{ij}, z_{ij}). Suppose $\mathbf{o}_I$ in Q_J is expressed as (x_{IJ}, y_{IJ}, z_{IJ}). To compute $\nabla d(\mathbf{s}_{ij}, Q_{I_{ij}})$ and $\nabla d(\mathbf{o}_I + d_i\mathbf{n}_I, Q_J)$, we only need to compute

$$\begin{aligned} \nabla z_{ij} &= (\mathbf{s}_{ij} - \mathbf{o}_{I_{ij}})\cdot\nabla\mathbf{n}_{I_{ij}} + \mathbf{n}_{I_{ij}}\cdot\nabla\bar{\mathbf{c}}_i + (\mathbf{s}_{ij}\times\mathbf{n}_{I_{ij}})\cdot\nabla\mathbf{c}_i \\ \nabla x_{ij} &= (\mathbf{s}_{ij} - \mathbf{o}_{I_{ij}})\cdot\nabla\mathbf{e}^1_{I_{ij}} + \mathbf{e}^1_{I_{ij}}\cdot\nabla\bar{\mathbf{c}}_i + (\mathbf{s}_{ij}\times\mathbf{e}^1_{I_{ij}})\cdot\nabla\mathbf{c}_i \\ \nabla y_{ij} &= (\mathbf{s}_{ij} - \mathbf{o}_{I_{ij}})\cdot\nabla\mathbf{e}^2_{I_{ij}} + \mathbf{e}^2_{I_{ij}}\cdot\nabla\bar{\mathbf{c}}_i + (\mathbf{s}_{ij}\times\mathbf{e}^2_{I_{ij}})\cdot\nabla\mathbf{c}_i \end{aligned} \tag{34}$$

and

$$\begin{aligned} \nabla z_{IJ} &= <\mathbf{n}_I, \mathbf{n}_J>\nabla d_I + d_I(\mathbf{n}_J\cdot\nabla\mathbf{n}_I) - d_J \\ &+ (\mathbf{o}_I - \mathbf{o}_J + d_I\mathbf{n}_I)\cdot\nabla\mathbf{n}_J \\ \nabla x_{IJ} &= <\mathbf{n}_I, \mathbf{e}^1_J>\nabla d_I + d_I(\mathbf{e}^1_J\cdot\nabla\mathbf{n}_I) \\ &+ (\mathbf{o}_I - \mathbf{o}_J + d_I\mathbf{n}_I)\cdot\nabla\mathbf{e}^1_J \\ \nabla y_{IJ} &= <\mathbf{n}_I, \mathbf{e}^w_J>\nabla d_I + d_I(\mathbf{e}^2_J\cdot\nabla\mathbf{n}_I) \\ &+ (\mathbf{o}_I - \mathbf{o}_J + d_I\mathbf{n}_I)\cdot\nabla\mathbf{e}^2_J \end{aligned} \tag{35}$$

© The Eurographics Association 2007.