

Efficient Surface Reconstruction from Range Curves

Dragan Tubić, Patrick Hébert and Denis Laurendeau

Computer Vision and Systems Laboratory

University Laval, Québec, Canada

[tdragan,hebert,laurendeau]@gel.ulaval.ca

Abstract

This paper proposes an approach for surface reconstruction of free-form rigid objects from an arbitrary set of intersecting range curves. A strategy for updating the reconstructed surface during data acquisition is described as well. Geometric and color information is accumulated in a volumetric structure in which a vector field is built and updated. Moreover, the information that is needed for efficient curve registration is directly available in this vector field. This leads to a unified modeling approach combining surface reconstruction and curve registration. The algorithm implementing the approach is of linear complexity with respect to the number of input curves and makes it suitable for interactive modeling. A compression scheme based on a multiresolution decomposition of vector fields is introduced as well. Simulated data from a set of curvilinear patterns as well as data acquired with a hand-held range sensor are used to validate the approach.

1. Introduction

Surface reconstruction algorithms from range data can accept as input either range images [3, 10, 14] or unorganized sets of points [8, 12]. However, these algorithms fail when applied to arbitrary curves. For this reason, hand-held range sensors extracting profiles from well-focused laser sheets are constrained to scan the surface along a nearly regular scanning path thus allowing local surface images to be built [2, 7, 9, 12]. An alternative is to collect a very dense set of arbitrary profiles on the surface and then process them as a cloud of points. The structure of the collected data is then lost. A second type of hand-held sensor collects range images [11, 13] but most of these sensors actually collect a rigid set of curves and do not provide the same density of measurements along all directions. Our goal is to develop a new class of surface reconstruction algorithms from range curves. These algorithms are particularly suitable for hand-held sensors and can advantageously exploit various projected patterns such as a crosshair, circles or grids.

Surface curves contain more information about the mea-

sured surface than an unorganized set of points, namely tangents that can be exploited to improve the quality of the reconstructed surface. A local estimate of the surface can be obtained using the tangents of intersecting curves. Figure 1 illustrates the most simple case where a surface is reconstructed in the neighborhood of two intersecting curves. Although the error of the approximated surface increases with the distance from the intersection, the reconstructed surface still follows the shape of the curves faithfully. This cannot be achieved by reducing curves to unorganized sets of points, i.e. by fitting a plane in the neighborhood of a point.

Besides surface reconstruction, modeling from range data faces another challenge: pose refinement or registration. If registration is frequently used for conventional range sensors, it becomes a key aspect for hand-held mobile sensors. When obtained from a positioning system, the sensor's pose is submitted to error and it can be refined before integrating the measurements in a global reference frame. For range images of rigid objects ICP-based algorithms (Iterated Closest Points) have been developed [1] for registration. For simultaneous registration, this type of algorithm is of complexity $O(N^2)$ with respect to the number of range images. If the same algorithm is applied for registering curves, the complexity quickly becomes prohibitive since the number of curves required for completely reconstructing the surface of an object is at least an order of magnitude larger than the equivalent number of range images.

Volumetric structures have been exploited for reconstructing surfaces from range images or unordered sets of points [3, 7, 10, 12] in order to reduce computational complexity and provide incremental reconstruction. These volumetric approaches use an implicit representation of the surface i.e. a scalar signed distance field computed in a volumetric grid where the reconstructed surface is located at the zero crossings of the field. This paper demonstrates how such a volumetric representation can be built directly from a set of measured curves without any intermediate surface representation. The computational complexity with respect to the number of curves is linear and the reconstruction is incremental and order independent, thus relaxing constraints on sensor displacement. If one not only encodes the signed

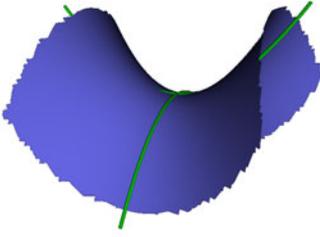


Figure 1. Reconstruction from two intersecting curves.

distance field in the volumetric structure but also the direction toward the nearest zero crossing, then matching for a control point can be obtained directly from the nearest voxel. We have developed this idea to provide near real-time registration of range images [15]. This paper explains how to provide the same linear complexity for surface curve registration by aligning them with the reconstructed surface.

The principle for intersecting curves has been presented in [16] as well as the algorithms and their behavior in the presence of noise. This paper builds on this work. The next sections describe the basic principle supporting surface reconstruction from curves including color information when collected at measured points. Section 2.5 describes curve registration in a vector field. Vector field compression is outlined in section 2.6. It is shown how the vector field can be used for an efficient compression of volumetric surface representation. The results presented in section 3 illustrate each aspect of the approach from simulated and real data.

2. Modeling using Range Curves

The reconstructed surface is represented implicitly as a vector field. Such a *volumetric* representation contains both the reconstructed surface and its corresponding matching information as direction and distance toward the reconstructed surface. While the surface is represented as the zero-crossing of the norm of the field, the distance and direction are represented by the field itself. In the following section, we describe how such an implicit representation of the surface can be created from a set of non-parallel, intersecting surface curves.

2.1. Reconstruction from intersecting curves

The main idea behind our approach is to perform reconstruction by *approximating the surface in the neighborhood of the intersection points of the surface curves*. The reconstruction is based on a fundamental property of differential surfaces stating that all surface curves passing through some point have tangents located in a plane tangent to the surface at the same point [4]. The most relevant consequence of this property is that the tangent plane of a surface can be com-

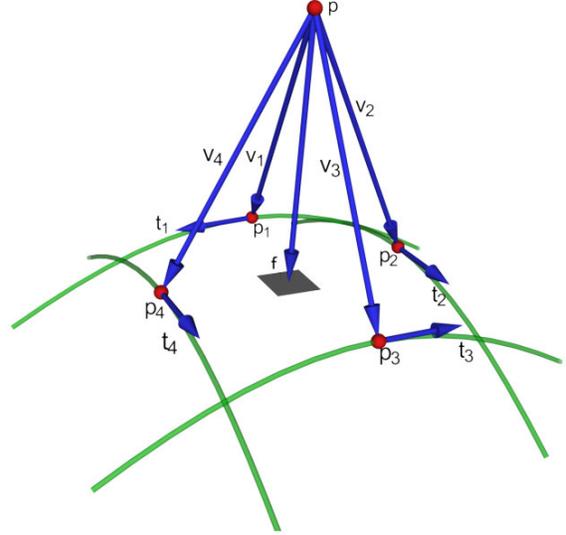


Figure 2. Reconstruction from multiple curves. The normal is obtained as a least-squares from the tangents t_i , $i = 1, 2, 3, 4$ at the closest points p_i of the four curves.

puted at the intersection point of at least two surface curves if their respective tangents are known at this point.

Formally, the reconstructed surface \hat{S} is represented as a vector field $f : R^3 \rightarrow R^3$ where $f(\mathbf{p})$ represents the direction and the distance from a point \mathbf{p} to the closest point \mathbf{p}_c on the surface \hat{S} , such that:

$$\mathbf{p} + \mathbf{f}(\mathbf{p}) = \mathbf{p}_c \in \hat{S}, \quad (1)$$

where

$$\mathbf{p}_c = \operatorname{argmin}_{\mathbf{q} \in \hat{S}} d(\mathbf{p}, \mathbf{q}), \quad (2)$$

and where d denotes a distance measure. In the next section, the distance measure will be defined to minimize the effect of the noise on the registration procedure, but for now, it can be assumed that the distance d is the Euclidean distance.

In practice, the field $f(\mathbf{p})$ is computed at points (voxels) on a regular lattice (volumetric grid) in the vicinity of a surface usually referred to as its envelope. The envelope encloses lattice points which are located at a distance to the surface being smaller than a predefined value ϵ . In this work, the envelope is defined for each curve and contains the set of lattice points closer than a predefined distance to the curve.

The approximation of the surface, i.e. the tangent plane at the closest point on the surface to some voxel \mathbf{p} can be obtained as a least-squares estimate of the normal using tangents at the closest points of all nearby curves; this is illustrated in Figure 2. This estimate corresponds to the "most perpendicular" vector to a set of tangents.

More formally, let $\alpha_1, \dots, \alpha_N$ be N surface curves passing within some predefined distance $\epsilon > 0$ from a point (voxel) \mathbf{p} and let t_1, \dots, t_N be their respective tangents at

the closest points to \mathbf{p} . Then, at the point \mathbf{p} , the normal on the surface is obtained as the vector $\mathbf{n} = [n_x, n_y, n_z]^T$ that minimizes the following expression:

$$\xi = \sum_{i=1}^N \langle \mathbf{t}_i, \mathbf{n} \rangle^2. \quad (3)$$

Taking the derivatives of ξ with respect to n_x, n_y and n_z and setting them equal to zero defines the following system of equations:

$$\frac{1}{N} \sum_{i=1}^N \mathbf{t}_i \mathbf{t}_i^T \mathbf{n} = \mathbf{C} \mathbf{n} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T. \quad (4)$$

The estimated value for \mathbf{n} is the eigenvector associated with the smallest eigenvalue of matrix \mathbf{C} which is just a covariance matrix of the tangents.

The distance toward the surface is obtained as the average value of the projected distance vectors $\mathbf{v}_i = \mathbf{p}_i - \mathbf{p}$ on the estimated normal, i.e.

$$d(\mathbf{p}, \hat{S}) = \sum_{i=1}^N \langle \mathbf{p}_i - \mathbf{p}, \mathbf{n} \rangle, \quad (5)$$

where \mathbf{p}_i is the closest point on the curve α_i . Finally the value of the field at point \mathbf{p} is:

$$\mathbf{f}(\mathbf{p}) = d(\mathbf{p}, S) \mathbf{n}. \quad (6)$$

In order to estimate the tangent plane to a surface at some point \mathbf{p} , at least two non-parallel tangents are needed to compute matrix \mathbf{C} at \mathbf{p} since a single tangent does not define a plane. This condition can be verified by analyzing the eigenvalues of matrix \mathbf{C} : if two eigenvalues are zero then only one tangent (or two or more parallel tangents) exists and the normal estimated at that point is not used.

At the intersection points of noiseless curves, all tangents are coplanar, hence one eigenvalue is always equal to zero and the tangents span a plane. In practice, all three eigenvalues are larger than zero since the surface is approximated in the neighborhood of the intersection points. Also, if the tangents are estimated from very noisy data, the three eigenvalues may have similar values, in which case the tangents span an ellipsoid and the estimate of the tangent plane is meaningless. To make sure that the estimated tangent plane is valid, an additional verification is made on the eigenvalues. Let λ_1, λ_2 and λ_3 be the three eigenvalues such that $\lambda_1 < \lambda_2 < \lambda_3$. Since matrix \mathbf{C} is normalized, the sum of eigenvalues is always equal to one i.e. $\lambda_1 + \lambda_2 + \lambda_3 = 1$. The eigenvalues have to satisfy the following two conditions: $\lambda_2 > 0.05$ and $\lambda_1 < 0.5\lambda_2$. While using these two thresholds does not affect the shape of the reconstructed surface, imposing them makes sure that the tangents used to compute \mathbf{C} are approximately coplanar; otherwise no surface is locally reconstructed.

A single curve influences the field only at points located inside its envelope with its influence dropping to zero outside the envelope. The computed field is thus discontinuous over the edges of the envelope and the reconstructed surface appears discontinuous as well. The solution to this problem is to weight the tangents using a continuous function of distance ideally dropping to zero at the edge of the envelope. Any decreasing monotonic function can be used for this purpose. In our experiments the following function proved to be useful:

$$\omega(d) = e^{-d^2/\sigma}. \quad (7)$$

The value of σ is chosen equal to $1/2\epsilon$ to make sure that ω is close to zero outside of the envelope.

The tangents can also be weighted in such a way to reduce influence of less-confident data, for example by measuring the angle between the surface (curve) normal and the measurement direction. The weighting value τ is defined as the cosine of the angle between the two directions. By taking into account τ and ω , the matrix defined in Eq. 4 becomes

$$\mathbf{C} = \frac{1}{\sum_{i=1}^N \tau_i \omega_i} \sum_{i=1}^N \tau_i \omega_i \mathbf{t}_i \mathbf{t}_i^T. \quad (8)$$

Finally, the reconstructed surface can be extracted from the vector field \mathbf{f} using a modified Marching Cubes algorithm. Actually, instead of searching zero-crossings of a scalar field between neighboring voxels, the modified algorithm searches for pairs of voxels whose values of the vector field are oriented in opposite directions.

2.2. Incremental reconstruction

For the reconstruction approach described above, it was assumed that all the data has been collected prior to reconstructing the surface. If the reconstruction has to be performed on-line, then the field needs to be updated incrementally by integrating a single curve at a time. However, the least-squares estimate of the surface normal and, consequently the vector field, cannot be computed incrementally. On the other hand, matrix \mathbf{C} computed at some voxel \mathbf{p} is obtained as a sum and can therefore be updated incrementally. Let $\mathbf{C}(\mathbf{p})$ be the matrix \mathbf{C} for voxel \mathbf{p} . Equation 8 can be rewritten as:

$$\mathbf{C}(\mathbf{p}) = \frac{1}{\sum_{i=1}^N \tau_i \omega_i} \sum_{i=1}^N \tau_i \omega_i \mathbf{C}_i(\mathbf{p}), \quad (9)$$

where $\mathbf{C}_i(\mathbf{p}) = \mathbf{t}_i \mathbf{t}_i^T$. During reconstruction, a matrix $\mathbf{C}(\mathbf{p})$ is attached to each voxel and is updated after acquiring each curve α_i by summing it with $\mathbf{C}_i(\mathbf{p})$. Matrix $\mathbf{C}_i(\mathbf{p})$ depends only on α_i and is computed using tangent \mathbf{t}_i at the point on the curve that is closest to \mathbf{p} . The value of the field $\mathbf{f}(\mathbf{p})$ is computed only during the extraction of the surface or during registration.

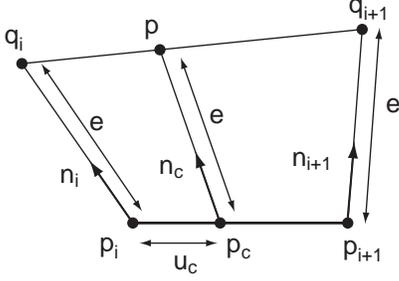


Figure 3. The distance in the direction of interpolated normals. The distance d between point p to the line segment $\overline{p_i p_{i+1}}$ is defined as the distance between p and the point p_c whose normal n_c passes through p .

2.3. Reconstruction of the surface color

In addition to 3D points measured on the surface of an object, some scanners can also provide color at the same points. In this case, the color of the final model can be reconstructed as well. To do so, the color for each voxel is incrementally computed in the same way as the covariance matrix. The color vector $s = [R, G, B]$ at voxel \mathbf{p} is computed as:

$$\mathbf{s} = \frac{1}{\sum_{i=1}^N \tau_i \omega_i} \sum_{i=1}^N \tau_i \omega_i \mathbf{s}_i, \quad (10)$$

where s_i is the color of the curve i at its closest point to the voxel \mathbf{v} . The vertices of the surface extracted from the field using the Marching Cubes are colored by interpolating the color at the two closest voxels. An example of color reconstruction is shown in Figure 12.

2.4. Defining the distance measure

As illustrated in Figure 4.a, points with high noise level tend to attract a large number of correspondences. This slows down the registration process and may reduce accuracy. To improve the robustness of matching, the *direction* toward the closest point on the line segment has to be corrected while taking into account two important constraints: i) the distance field has to be continuous ii) the position of measured 3D points should not be altered. The solution to this problem is to redefine the distance instead of using the Euclidean distance.

A curve α is represented as a set of line segments $\alpha = \{l_1, \dots, l_{N-1}\}$ between the measured points $\{\mathbf{p}_1 \dots \mathbf{p}_N\}$, with $l_i = \overline{p_i p_{i+1}}$. The tangent \mathbf{t}_i at each measured point \mathbf{p}_i is computed and filtered by averaging it with its neighbors. The tangents are then interpolated over the line segments.

The distance between a point \mathbf{p} and the curve α is always computed with respect to the closest line segment l_c . It is therefore sufficient to provide the distance from a point to a line segment. The set of all points for which a line segment l_c is the closest is called a *fundamental cell* associated with the *generator line segment* l_c . For the purpose of surface

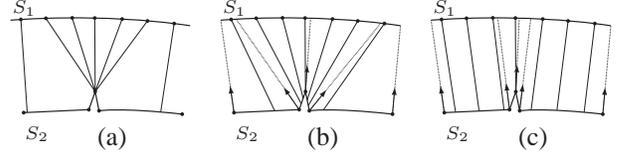


Figure 4. The effect of noise on the matching step of registration. a) Matching with a noisy curve (or surface) S_2 using Euclidean distance. Since the noisy point in this example is closest to S_1 it attracts most of the points. b) Matching using the distance defined in Eq. 11 with unfiltered normals. c) Matching in the direction of filtered normals. Matched points are evenly distributed over S_2 .

modeling, the field needs to be calculated only within a relatively small distance $\epsilon > 0$ from the measured curve, thus limiting the size of the cell to the set of points which are closer than ϵ . If d is the Euclidean distance then the fundamental cells correspond to the cells of the Voronoi diagram for a set of line segments.

The distance d between point \mathbf{p} and the line segment $\overline{p_i p_{i+1}}$ is defined as the distance between \mathbf{p} and the point \mathbf{p}_c whose normal \mathbf{n}_c passes through \mathbf{p} . The normal \mathbf{n}_c is defined as the vector that is perpendicular to the tangent at \mathbf{p}_c and contained in the plane $\overline{p_i p_{i+1} p}$. This is illustrated in Figure 3. More formally:

$$d(\mathbf{p}, \overline{p_i p_{i+1}}) = d(\mathbf{p}, \mathbf{l}(u_c)) = e, \quad 0 \leq u_c \leq 1, \quad (11)$$

such that

$$\mathbf{p} = \mathbf{l}(u_c) + e \cdot \mathbf{n}_c, \quad (12)$$

where

$$\mathbf{l}(u) = \mathbf{p}_i + u(\mathbf{p}_{i+1} - \mathbf{p}_i), \quad 0 \leq u \leq 1. \quad (13)$$

To obtain a closed-form solution for the distance between a point \mathbf{p} and a line segment $\overline{p_i p_{i+1}}$ according to the above definition, we note that, if the distance is d , then the point lies on the line segment whose end points are $\mathbf{q}_i = \mathbf{p}_i + e \cdot \mathbf{n}_i$ and $\mathbf{q}_{i+1} = \mathbf{p}_{i+1} + e \cdot \mathbf{n}_{i+1}$, as illustrated in Figure 3. This line segment is an *iso-segment* whose points are all located at a distance d from the generator line segment $\overline{p_i p_{i+1}}$. The distance is computed by making the area of the triangle $\mathbf{q}_i \mathbf{q}_{i+1} \mathbf{p}$ equal to zero, i.e. the cross-product of $\mathbf{q}_i - \mathbf{p}$ and $\mathbf{q}_{i+1} - \mathbf{p}$ is zero. This leads to a system of three quadratic equations with a single unknown d . Any of these equations can be used to solve for d after making sure that the chosen parameters do not vanish altogether.

The effect of choosing the distance defined in Eq. 11 is illustrated in Figure 4b and c. Filtering the tangents, and consequently the normals, affects the direction toward the closest point, and therefore the matching directions. This makes the matched points more evenly distributed over the reconstructed surface. It is important to note that only tangents on the curves are filtered, the measured 3D data remains unchanged.

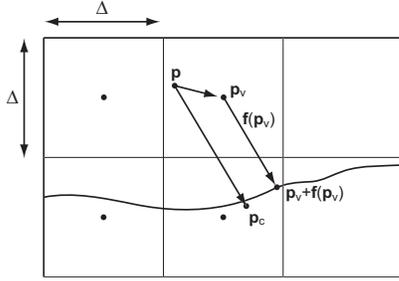


Figure 5. Matching a point p using the closest voxel centre p_v . The point p_c is obtained using Eq. 14.

2.5. Registration using vector fields

On one hand, unlike range images, surface curves cannot be registered with each other one pair at a time. In general, the number of intersections between two curves is insufficient to compute the rigid transformation needed to align the curves. For instance, two surface profiles intersect at a single point. On the other hand, registering all curves simultaneously has $O(N^2)$ complexity with respect to the number of curves and quickly becomes limiting due to the large number of curves that is needed to reconstruct an object. An efficient way to circumvent these problems is to register curves to the *reconstructed model*. Since the vector field contains all the information needed for matching, the computational complexity remains linear with respect to the number of curves. Matching a single control point is of $O(1)$ complexity.

Once the vector field is computed, registering a curve becomes straightforward: for a control point p on a curve, the corresponding point p_c on the reconstructed surface is approximated by the value of the vector field at the closest voxel p_v to the point p . An error introduced by field discretization can be corrected by linearly interpolating the value of the field at the voxel p_v (see Figure 5):

$$p_c = p + f(p_v) + \frac{f(p_v) \cdot (p - p_v)}{\|f(p_v)\|^2}. \quad (14)$$

The rigid transformation minimizing the sum of distances $(p_c - p)$ is applied to each curve. For registration, the model is first reconstructed from all available curves. Then, each curve is registered to this model one at a time. The vector field is then recomputed, and the whole procedure is repeated until no further improvement is possible.

In this work, all curve points are used as control points. Convergence is verified by measuring the displacement of points after applying the computed transformation; when it falls below a threshold the algorithm stops.

2.6. Compression of the vector field

The most simple software implementation of vector field representation is to allocate the memory for all grid points (voxels) in a volume. Since the vector field is computed only within an envelope of the surface, the number of used

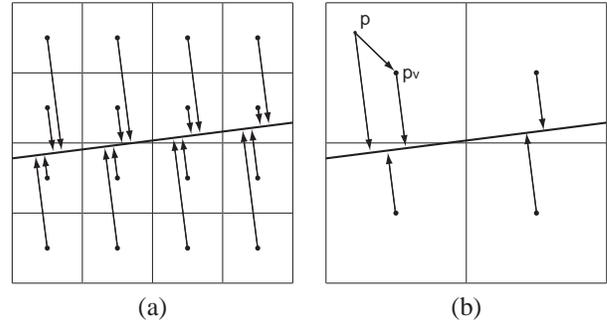


Figure 6. Principle of the compression of vector fields. In regions where the surface is planar or changes slowly (a), groups of voxels can be removed and replaced without loss by a smaller number of voxels at lower resolution (b).

voxels is very small with respect to the total number of voxels. This is a very inefficient approach since the largest part of allocated memory is used for unoccupied voxels. This problem could be solved by employing run-length encoding [3] or octrees. Even though these methods reduce the amount of memory, they are still inefficient since they encode only occupancy of the voxels without taking into account the shape of the represented surface. Furthermore, the compression rate depends significantly on the orientation and position of the object with respect to the volume.

The vector field representation offers a possibility to address these two problems related to run-length encoding and octrees. The compression is based on the fact that a vector field at any given voxel, together with the voxel position itself, defines a plane that corresponds to the tangent plane at the closest surface point. If the surface is planar in some region of the volume (see Figure 6a), all voxels define exactly the same plane. Therefore, the surface can be represented by a single voxel while all other voxels in the same region are redundant and can be removed (see Figure 6b). If the field is encoded in an octree, then groups of 8 voxels that represent a planar surface can be removed and replaced by a single voxel at lower resolution. The same procedure can be repeated for voxels at lower resolutions, thus further compressing the field. If the value of the field is required at some of the removed voxels, it can be recovered using Eq. 14. It should be noted that the field for the planar surface can be reduced to a single voxel without loss, *regardless of the position and the orientation of the surface*. If the geometry of the surface changes in some region of the volume, then it cannot be further compressed without loss. The value of the field at the voxel of lower resolution is chosen to minimize the error and if it is below some threshold, the higher resolution voxels are discarded. This type of compression corresponds to the wavelet based compression and multiresolution analysis [17] which was adapted for vector fields.

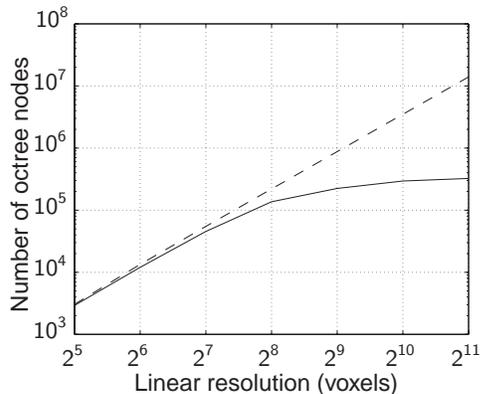


Figure 7. Amount of memory (number of octree nodes) for uncompressed and compressed octree representation of the field (log scale). While the amount of memory for uncompressed octree (dashed curve) grows exponentially, the amount of memory for the compressed field (solid curve) is bounded.

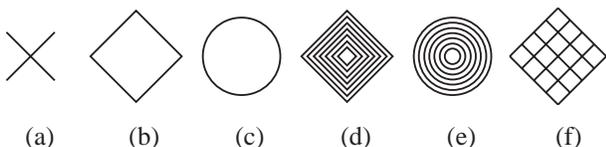


Figure 8. Light patterns used in experiments.

For every signal there is a maximum required sampling frequency (determined by the Shannon theorem). Increasing the sampling frequency over this value does not add any new information about the signal. The same argument applies to vector fields. Therefore, the most important aspect of compression is to remove this redundant information and prevent unnecessary exponential growth of the memory requirements when resolution is increased. As shown in Figure 7, there is an upper bound on the amount of required memory for a compressed field as the resolution increases. This upper bound depends on the geometry of the object.

Locally, the vector field for a single surface might require different sampling resolutions. For example, the vector field around flat regions of a surface can be sampled with lower resolution than the regions where the surface deforms rapidly. This adaptive sampling is also provided with the proposed compression scheme since high-resolution voxels are discarded only in regions where the resulting error is lower than a threshold.

3. Results

An example of reconstruction from multiple curves, using real range data is shown in Figure 9b. The data for this model has been acquired using the 3D sensor described in [6] which acquires surface curves by measuring distances to a cross-hair laser pattern (see Figure 8a) projected on the surface of an object. In Figure 9, one will note that the algorithm performs well even where the density of the curves

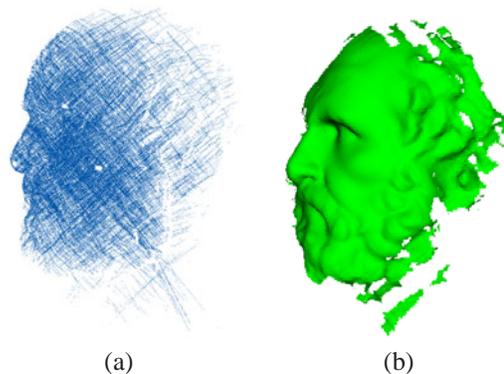


Figure 9. Reconstruction from real range data. a) Raw range data (cross-hair pattern). b) Reconstructed surface.

vary (right-hand side of the figure). However, if the distance between the curves becomes larger than the size of the envelope, holes appear in the reconstructed model. As expected, this is due to the validation test in section 2.1.

The quality of registration has been assessed by registering synthetic curves with varying levels of noise, outliers and initial registration errors. For that purpose a highly accurate model of the head measured using a range sensor with $25\mu m$ accuracy was used to generate synthetic data. The curves generated using the 6 laser patterns in Figure 8, have been randomly rotated for up to 4 degrees and translated for up to $4mm$ in a working volume of $200mm$. Noise was simulated by translating each point in the direction of the laser projector for up to $4mm$, while outliers were generated by randomly choosing 1% of the total number of points and displacing them for a random distance smaller than $10mm$. Values for the curve displacement and noise levels were chosen smaller than the size of the envelope ($4mm$). Otherwise, some curves are located outside of the envelope and cannot be matched and registered. The size of the model is approximately $20cm$ while the size of a voxel is $1mm$.

Color encoded residual errors are shown in Figure 10. The average error for all six patterns was below $0.32mm$ with a variance below $0.12mm$. This error level is sufficiently small to yield a reconstructed surface without visible artefacts. On the other hand, the maximum residual error in some cases was as high as $9mm$. However, since the variance and average error are low, the number of curves showing such a high error is small. As illustrated in Figure 10 the curves with large errors are located at the boundary of the scanned area where the density of curves is low. In these regions, the field could not be computed, and curves are only partially matched against the reconstructed surface, thus leading to unreliable registration. Among 450 curves, only three curves showed an error larger than $1mm$. This problem can be solved by rejecting curves that are partially matched.

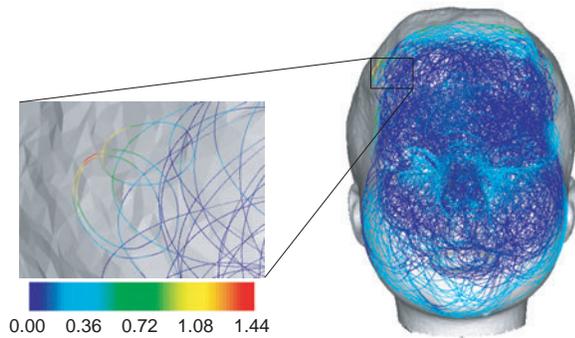


Figure 10. Color encoded distribution of residual registration errors (mm). Curves on the border of the scanned area are only partially matched and therefore unreliably registered. Only three curves have a registration error larger than 1mm.

Registering real range data of the same head model obtained with the hand-held sensor described in [6] leads to similar results. Initial average registration errors were $0.35mm$ and with a variance of $0.09mm^2$. After registration, these errors were reduced to $0.26mm$ (average) and $0.07mm^2$ (variance), which is in conformity with results obtained with synthetic data. The residual registration error was evaluated using the reference head model measured with $25\mu m$ accuracy.

The registration algorithm converges quickly, typically in less than 30 iterations. No important differences with respect to convergence have been observed for different scanning patterns. A more detailed analysis of convergence as well as a detailed analysis of performance of registration in the presence of noise and varying levels of initial registration errors are presented in [16].

In planar or rotationally symmetric regions, the objective function of an ICP algorithm (sum of distances toward the closest points) reaches a minimum in a flat region, which causes the registration to converge towards the closest minimum. For example, a curve measured in a planar region of the surface can be translated and rotated arbitrarily in the same plane without increasing the distance to the surface and without affecting the geometry of the reconstructed surface. This "sliding" effect can be measured by comparing the distance of registered points to *their exact position and the distance to the closest point* on the reference model. As expected, the distance to the exact positions is higher in relatively flat regions of the surface (see Figure 11).

Even though the sliding effect does not affect the geometry of the object, it becomes very important when the colors have to be reconstructed as well. As illustrated in Figure 12, although the initial registration errors were large, the reconstructed geometry is well preserved, but the colors may become erroneous.

Using a hand-held sensor [6], the presented algorithms

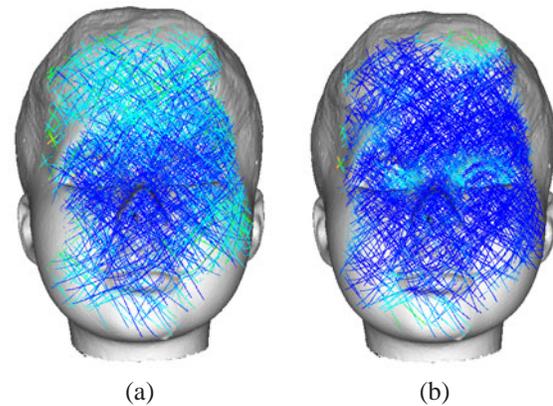


Figure 11. a) Residual registration error (brighter color indicates larger error) measured as the distance between the corrected positions of points and their exact position. The regions of larger errors are featureless regions such as forehead and cheeks. b) Residual registration error measured as the distance towards the closest point on the reference surface. Errors are uniformly distributed over the surface.

were exploited to create an interactive modeling system. The incremental reconstruction from curves having approximately 250 points, each is performed in real-time at the frame rate of the sensor (30fps). Thereafter, the registration algorithm is run. The total registration and reconstruction time depends mostly on the size of the envelope: for 450 curves it is 55 sec for $\epsilon = 3$ mm and 140 sec for $\epsilon = 5$ mm. Another factor that affects the execution time is initial registration errors. Smaller errors require less iterations thus reducing the number of field recomputations which is the most computationally intensive operation. An incremental variant of the proposed registration algorithm can be used to reduce modeling time. The registration can proceed online as soon as an initial field is created from a relatively small number of curves (typically one hundred). This way, the registration can be performed at the frame rate of the sensor despite a somewhat reduced quality.

4. Conclusion

A volumetric approach for modeling from arbitrary intersecting curves measured on the surface of a free-form object is proposed. An implicit representation of the surface is created incrementally from a set of unordered curves without any intermediate surface representations. All steps of the modeling procedure, including matching, are of linear complexity, making this approach particularly well-suited for interactive modeling and hand-held sensors. Robustness of the algorithm with respect to noise and outliers is improved by redefining the distance measure. The proposed approach allows easy integration of the color in the reconstruction procedure.

As a variant of ICP, the proposed registration algorithm

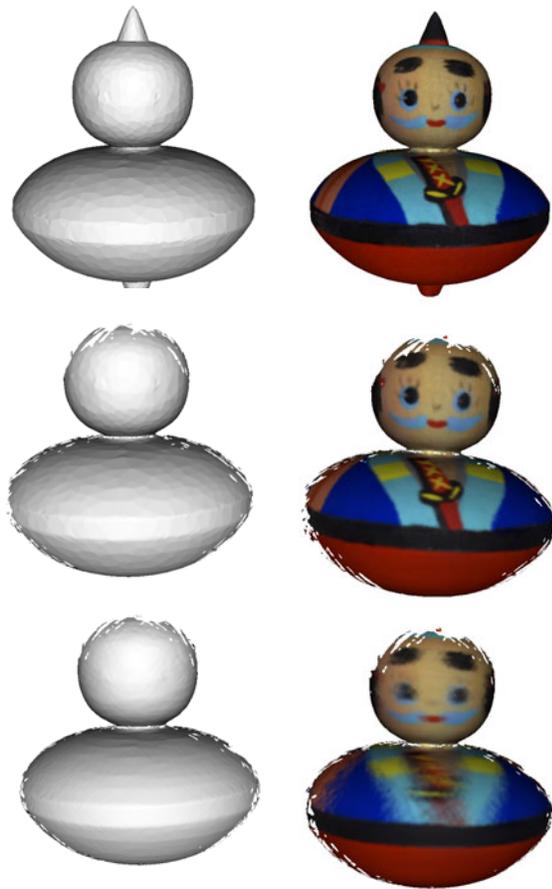


Figure 12. Example of reconstruction of color models. Top row: original model. Middle: reconstruction from curves without registration errors. Bottom row: Reconstruction from registered data with large initial registration errors. Sliding effect (see text) does not affect the geometry of the reconstructed model (left-hand side) but it renders reconstructed color blurred (bottom-right image).

fails to recover the exact position of curves in rotationally symmetric or flat regions of the surface. This makes the surface color reconstruction unreliable whenever the initial registration errors are large. We believe that constraining the registration using color information should solve this problem. Providing an algorithm similar to [5] for curve registration will be a starting point of our future work.

5. Acknowledgments

This work has been supported by FQRNT (Fonds Québécois de la recherche sur la nature et les technologies) and NSERC (Natural Sciences and Engineering Research Council of Canada). The authors would like to thank Guy Godin from VIT-NRCC for valuable comments as well as for range data used for color reconstruction.

References

- [1] P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE Transactions PAMI*, 14(2):239–256, February 1992.
- [2] S. J. Cunningham and A. J. Stoddart. Self-calibrating surface reconstruction for the modelmaker. In *BMVC'98 proc.*, volume 2, pages 790–799, 1998.
- [3] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH '96 Proceedings*, pages 303–312, 1996.
- [4] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.
- [5] G. Godin, D. Laurendeau, and R. Bergevin. A method for the registration of attributed range images. In *Proceedings 3DIM*, pages 179–186, 2001.
- [6] P. Hébert. A self-referenced hand-held range sensor. In *Proceedings of 3DIM*, pages 5–11, May 2001.
- [7] A. Hilton and J. Illingworth. Geometric fusion for a hand-held 3d sensor. *Machine vision and applications*, 12:44–51, 2000.
- [8] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *SIGGRAPH '92 Proceedings*, volume 26, pages 71–78, July 1992.
- [9] J. Kofman and G. K. Knopf. Registration and integration of narrow and spatiotemporally-dense range views. In *Proceedings of SPIE, Vision Geometry VII*, volume 3454, pages 99–109, July 1998.
- [10] T. Masuda. Object shape modeling from multiple range images by matching signed distance fields. In *Proceedings of 3DPVT*, pages 439–448, June 2002.
- [11] M. Proesmans, L. V. Gool, and F. Defoort. Reading between the lines - a method for extracting dynamic 3d with texture. In *Proceedings of ICCV 1998*, 1998.
- [12] G. Roth and E. Wibowo. An efficient volumetric method for building closed triangular meshes from 3-d image and point data. In *Graphics Interface*, pages 173–180, May 1997.
- [13] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3d model acquisition. In *proceedings of SIGGRAPH 2002*, 2002.
- [14] M. Soucy and D. Laurendeau. A general surface approach to the integration of a set of range views. *IEEE Transactions PAMI*, 17(4):344–358, 1995.
- [15] D. Tubić, P. Hébert, and D. Laurendeau. A volumetric approach for interactive 3d modeling. In *Proceedings of 3DPVT*, volume 1, pages 150–158, June 2002.
- [16] D. Tubić, P. Hébert, and D. Laurendeau. 3d surface modeling from curves. In *Proceedings of CVPR 2003*, pages 842–849, 2003.
- [17] T. Udeshi, R. Hudson, and M. E. Papka. Seamless multiresolution isosurfaces using wavelets. *Argonne National Laboratory, ANL/MCS-P801-0300*, 2000.