Reconstruction of Objects with Jagged Edges through Rao-Blackwellized Fitting of Piecewise Smooth Subdivision Curves

Michael Kaess and Frank Dellaert
College of Computing, Georgia Institute of Technology, Atlanta, GA
{kaess,frank}@cc.gatech.edu

Abstract

In some applications objects are known to have non-smooth or "jagged" edges, which are not well approximated by smooth curves. We use subdivision curves as a simple but flexible curve representation, which allows tagging corners to model non-smooth features along otherwise smooth curves. A Markov chain Monte Carlo approach yields an approximate posterior distribution over tags, while Rao-Blackwellization allows us to integrate out the control point locations by an approximation. We apply this general methodology to multi-view reconstruction of piecewise smooth curves from multiple calibrated views in which the object has been segmented from the background. Results are shown for multiple images of two pot shards as would be encountered in archaeological applications.

1. Introduction

In this paper we investigate the 3D reconstruction of shards and other artifacts that are known to have "jagged edges". We exploit this prior knowledge about the object by formulating the problem as the reconstruction of piecewise smooth curves from multiple calibrated views. Existing 3D reconstruction methods rely on automatically extracted points and/or lines [11]. In the case of textured objects or in the presence of projected patterns, these methods can be used successfully to sparsely sample the 3D surface of an object. However, they fail to use or capture the fact that an object like a shard is delineated by a closed boundary curve. In this paper we compliment traditional 3D reconstruction methods by explicitly recovering these curves.

To model piecewise smooth curves we use tagged subdivision curves as the representation, the details of which are given in Section 2. This is inspired by the work of Hoppe [12], who successfully used tagged subdivision surfaces for fitting piecewise smooth surfaces to 3D point clouds. The curve fitting literature includes use of algebraic curves [13, 1], piecewise polynomials [6, 16, 17], point curves [2], and B-splines [5, 4, 7]. To the best of our knowledge, no prior work on fitting subdivision curves exists. Subdivision curves are simple to implement and provide a flexible way of representing curves of any type, including all kinds of B-splines and extending to functions without analytic representation. A comprehensive introduction to subdivision curves can be found in [19]. In [12], Hoppe introduces piecewise smooth subdivision *surfaces*, allowing to model sharp features such a creases and corners by tagging the corresponding control points. We use the same idea of tagging, and apply it to subdivision *curves* to represent piecewise smooth curves.

To infer the parameters of these curves from the data, we propose Rao-Blackwellized sampling, discussed below in Section 3. In our approach, MCMC sampling [9] is used to obtain a posterior distribution over the tags, while the control point locations are integrated out after a non-linear optimization step. In related work, Denison and Mallick [6, 16] propose fitting piecewise polynomials with an unknown number of knots using reversible jump Markov Chain Monte Carlo sampling [10]. Punskaya [17] extends this work to unknown model orders within each segment with applications in signal segmentation. DiMatteo [7] extends Denison's work for the special case of natural cubic B-splines, handling non-smooth curves by representing a corner with multiple knots. However, the knots cannot be at the same location, and therefore only approximate the corner. With our method corners can be represented exactly with a single control point. In addition, we are working with a much reduced sample space, as we directly solve for optimal control point locations and hence only sample over the boolean product space of corner tags.

We apply this general methodology to 3D reconstruction of piecewise smooth curves from multiple calibrated images, discussed below in Section 4. While much of the curve fitting literature is concerned with 1D curves for signal processing [16, 6, 7, 17], in computer vision it is more common to fit curves in 2D or 3D. For example, 2D curves are often fit to scattered point data [1] and image contours [5]. In multi-view fitting, for the special case of stereo cam-

eras, [4] describes reconstruction and tracking of 3D curves represented by 2D B-spline curves, that are either coupled through epipolar geometry constraints, or are coupled to a canonical frame model through affine transformations. More general multiple view curve reconstruction methods are described in [13] using algebraic curves, and in [2] for point curves with uncalibrated cameras.

A possible application of our work is the automatic reconstruction of archaeological artifacts. [14] describes manual restoration of archaeological objects. Shards are scanned with laser range finders, and the resulting 3D models can manually be manipulated in a VR environment. It is suggested to automate this process by classifying shards by their curvature and their texture.

2. Subdivision Curves

2.1. Subdivision Review

Below we briefly review subdivision curves. See [19] for a more detailed introduction.

A subdivision curve is defined by repeatedly refining a vector of control points $\Theta_t = \left(x_0^t, x_1^t, ..., x_{n2^t-1}^t\right)$, where n is the initial number of control points and t the number of subdivisions performed. This refinement process can be separated into two steps as shown in Figure 1: a *splitting step* that introduces midpoints and therefore doubles the number of points:

$$\begin{aligned} x_{2i}^{t+1} &\triangleq x_i^t \\ x_{2i+1}^{t+1} &\triangleq \frac{1}{2}\left(x_i^t + x_{i+1}^t\right), \end{aligned}$$

and an *averaging step* that computes weighted averages according to the *averaging mask* $r = (..., r_{-1}, r_0, r_1, ...)$:

$$x_i^{t+1} \triangleq \sum_k r_k x_{i+k}^t$$

The type of the resulting curve depends on the mask r. Subdivision can be used to create a surprisingly wide range of functions [19], including linear interpolating curves, uniform and non-uniform B-splines, and even functions that have no analytic representation like Daubechies wavelets. For example, the averaging mask for a cubic B-spline is $\frac{1}{4}(1, 2, 1)$. Note that some of these curves interpolate the control points while others approximate them, e.g. all B-splines of second order and higher. We use *uniform* subdivision schemes where the same averaging mask is used everywhere along the curve, and generally use approximating rather than interpolating curves.

As explained in [19], the splitting and averaging steps can be combined into multiplication of the local control points with a *local subdivision matrix* L, e.g.

$$L = \frac{1}{8}\begin{pmatrix} 4 & 4 & 0 \\ 1 & 6 & 1 \\ 0 & 4 & 4 \end{pmatrix}$$

for cubic B-splines. Repeated application of this matrix to a control point and its immediate neighbors results in a sequence of more and more refined points, that converges to the limit value of the center point. Eigenvector analysis on the matrix L leads to an *evaluation mask* u that can be applied to a control point and its neighbors, resulting in the limit position for that control point. For example, the evaluation mask for cubic B-splines is

$$u = \frac{1}{6}(1, 4, 1)$$

The curve can be refined to the desired resolution before the evaluation mask is applied.

2.2. Matrix Notation and Derivatives

In what follows, it will be of use to view the *entire* subdivision process as one large matrix multiplication

$$C = S\Theta \tag{1}$$

where C is the final curve, the $n \times 1$ vector Θ represents the control points/polygon, and the *subdivision matrix* S combines all m subdivision steps and the application of the evaluation mask into one $n2^m \times n$ matrix. This can be done as both subdivision and evaluation steps are linear operations on the control points. The final curve C is a $n2^m \times 1$ vector that can be obtained by multiplying S with the control point vector Θ as in (1).

The *derivative* of the final curve C with respect to a change in the control points Θ, needed below to optimize over them, is simply a constant matrix:

$$\frac{\partial C}{\partial \Theta} = \frac{\partial (S\Theta)}{\partial \Theta} = S \tag{2}$$

While the above holds for 1D functions only, an n-dimensional subdivision curve can easily be defined by using n-dimensional control points, effectively representing each coordinate by a 1D subdivision curve. Our implementation is done in the functional language ML, and uses functors to remain independent of the dimensionality of the underlying space. Note that in that case, the derivative equation (2) holds for each dimension separately.

2.3. Piecewise Smooth Curves

There are different approaches to representing sharp corners in otherwise smooth curves. One obvious solution is to

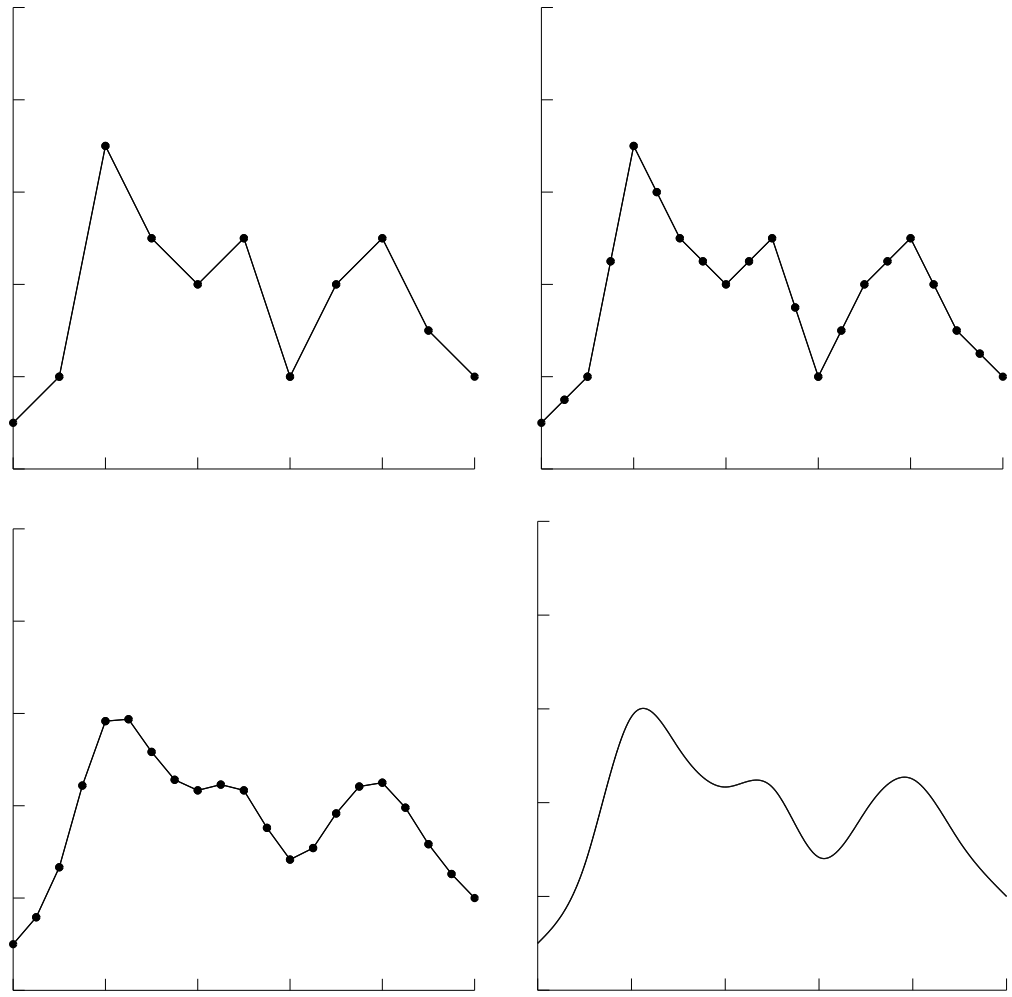


Figure 1. Different stages in the subdivision process are shown. The top left image shows the initial control points linearly interpolated. In the top right image, the original mesh is subdivided by introducing average points. The result after application of the averaging mask to the subdivided control points is shown in the bottom left. The last image shows the converged curve.

place multiple control points at the location of a corner. Besides unnecessary complexity of the subdivision curve by doubling the number of points at that location with each subdivision step, this places unwanted constraints on the adjacent curve segments. A commonly used method in connection with B-splines is the insertion of knots in a knot sequence, which defines the B-spline basis functions.

We employ a more general method here that works for any subdivision curve, based on work of Hoppe [12] for subdivision surfaces. It allows us to "tag" control points, allowing different averaging masks to be used at these points. In particular, using the mask $(0, 1, 0)$ forces interpolation of the control point and introduces a discontinuity in the derivative there, while retaining the smooth curve properties for all other points of the curve. An example is shown in Figure 2. The number of tagged control points does not increase during the subdivision process, and so the non-smoothness of the curve is restricted to exactly the tagged control points. These will be interpolated, regardless of the general averaging mask used.

Below we will use the following notation to describe tagged 3D subdivision curves. The locations of the control points in 3D are given by $\Theta \triangleq \{x_0^0, x_1^0, ..., x_{n-1}^0\}$, where n is the number of control points. For each original control point x_i^0, a boolean *tag* b_i indicates whether it is non-smoothly interpolated, i.e. whether there is a "corner" at control point i. The collection of tags b_i is written as $T \triangleq \{b_0, b_1, ..., b_{n-1}\}$.

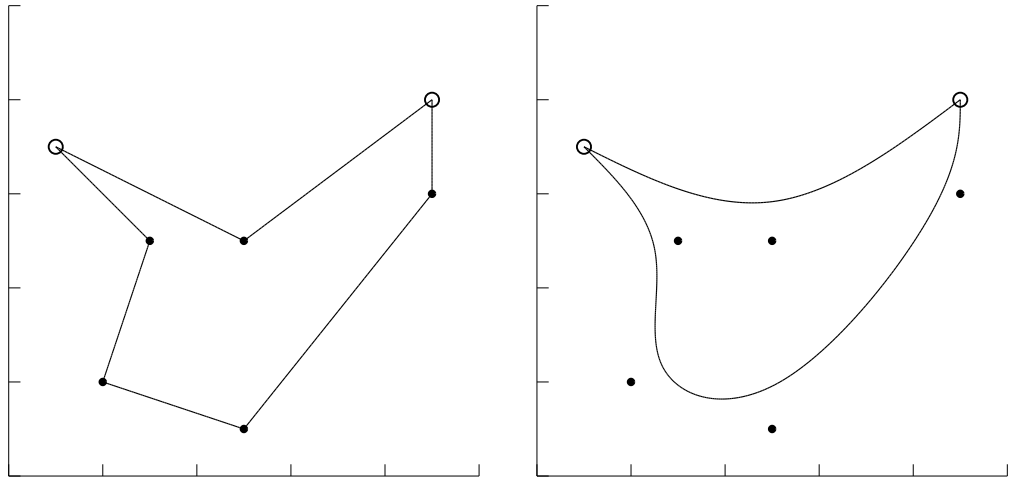


Figure 2. A tagged 2D subdivision curve is shown. Tagged control points are drawn as circles. The left image shows the original control point configuration, the right image the final converged curve with two non-smoothly interpolated control points.

3. Rao-Blackwellized Curve Fitting

In this section we describe how the parameters of a tagged subdivision curve can be estimated from noisy measurements Z, irrespective of how those measurements were obtained. In Section 4 this will be specialized to fitting from multiple, calibrated segmentations of an object.

Because the measurements are noisy we take a probabilistic approach. Of interest is the posterior distribution

$$P(\Theta, T|Z) \propto P(Z|\Theta, T)P(\Theta|T)P(T) \tag{3}$$

over the possible control point values Θ and tag variables T, as defined in Section 2.3. Note that the control points $\Theta \in \mathbb{R}^{3n}$ are continuous and the tags $T \in \{0,1\}^n$ are discrete. The *likelihood* $P(Z|\Theta, T)$ and *control polygon prior* $P(\Theta|T)$ are application specific, and will be further specified in Section 4. For the *tagging prior* $P(T)$, one can either use a binomial distribution over the number of active tags c

$$P(T) \propto p^c(1-p)^{n-c}$$

or simply have an uninformative prior that does not favor any tagging configuration over any other.

Since the number of possible tag configurations is 2^n and hence exponential in n, we propose to use Markov chain Monte Carlo (MCMC) sampling to perform approximate inference, using the *Metropolis-Hastings* (MH) algorithm [9, 15]. This method produces an approximate sample from a target distribution $\pi(X)$, by simulating a Markov chain whose equilibrium distribution is $\pi(X)$. The algorithm starts from a random initial state $X^{(0)}$ and proposes probabilistically generated moves in the state space, which is equivalent to running a Markov chain. However, the MH algorithm modifies the chain to have the desired equilibrium distribution by rejecting some of the moves:

1. Start with a random initial state $X^{(0)}$.
2. Propose a move to a new state X' using an arbitrary but fixed *proposal density* $Q(X'; X^{(r)})$.
3. Compute the acceptance ratio
$$a = \frac{\pi(X')}{\pi(X^{(r)})} \frac{Q(X^{(r)}; X')}{Q(X'; X^{(r)})} \tag{4}$$
4. Accept the move to X' as $X^{(r+1)}$ with probability $\min(a, 1)$, otherwise $X^{(r+1)} = X^{(r)}$.

The sequence of states $\{X^{(r)}\}$ thus generated will be a sample from $\pi(X)$ if the sampler is run sufficiently long, and one discards the samples in the initial "burn-in" period of the sampler to avoid dependence on the chosen start state.

For fitting tagged subdivision curves, we propose to not sample from the joint posterior (3), but rather from the marginal distribution $P(T|Z)$ over the tags T, i.e.

$$\pi(X) \triangleq P(T|Z) = \int P(\Theta, T|Z) d\Theta \tag{5}$$

while performing the integration (5) above analytically. This will reduce the number of samples needed, as explained below. From a sample over T of size N we can obtain a high quality approximation to the joint posterior as follows

$$P(\Theta, T|Z) \approx \sum_{r=1}^{N} P(\Theta|T^{(r)}, Z)\delta(T, T^{(r)}) \tag{6}$$

with $\delta(.,.)$ the Kronecker delta. Thus, (6) approximates the joint posterior $P(\Theta, T|Z)$ as a combination of discrete samples and continuous conditional densities $P(\Theta|T^{(r)}, Z)$. The superiority of (6) over a density estimate based on joint samples is based on an application of the Rao-Blackwell theorem of mathematical probability, which is why integrating out of part of the state is often referred to as *Rao-Blackwellization* [8, 3, 18]. Intuitively, the variance of (6) is lower because it uses exact conditional densities to approximate the continuous part of the sate, conditioned on the discrete samples. As such, many fewer samples are needed to obtain a density estimate of similar quality. Any expectation calculated using (6) will have lower variance, as well, for example the mean value of a certain control point or the length of the curve.

Substituting (3) in (5) we obtain

$$P(T|Z) = kP(T) \int P(Z|\Theta, T)P(\Theta|T)d\Theta$$

Under the assumption that the conditional posterior $P(Z|\Theta, T)P(\Theta|T)$ is approximately normally distributed around the MAP estimate of the control points Θ^*

$$P(\Theta|Z, T) \approx \frac{1}{\sqrt{|2\pi\Sigma|}} e^{-\frac{1}{2}|\Theta-\Theta^*|_\Sigma}$$

the integral can be approximated via Laplace's method and we obtain the following target distribution over the tags T:

$$P(T|Z) = kP(T)\sqrt{|2\pi\Sigma|}P(Z|\Theta^*, T)P(\Theta^*|T)$$

The MAP estimate Θ^* can be found by non-linear optimization, as explained in the section below.

4. Multiple View Fitting

In this section we specialize the general methodology of Section 3 to reconstructing tagged subdivision curves from multiple 2D views of a "jagged" object. We assume here that (a) the images are calibrated, and (b) the measurements Z are the object boundaries in the 2D images, i.e. the object has been segmented out from the background. The curve given by Θ and T is subdivided in m steps to the desired resolution, evaluated, and the resulting points $\{p_i\}_1^{n2^m}$ are projected into each view. We then use the following form for the likelihood $P(Z|\Theta, T)$:

$$P(Z|\Theta^*, T) \propto e^{-\beta E(Z,\Theta^*,T)}$$

where β is a constant, and the error function E is obtained as a sum of squared errors $\Delta_{ci}(.)$

$$E(Z, \Theta, T) = \sum_{c=1}^{C} \sum_{i=1}^{n2^m} \Delta_{ci}(\Pi_c(p_i), Z_c)^2 \qquad (7)$$

with one term for each of the $n2^m$ final subdivision curve points in each of the C images, explained in more detail below. Both the prior $P(T)$ on the tag configuration and the conditional control polygon prior are taken to be uniform in all the results reported below.

Each error term $\Delta_{ci}(\Pi_c(p), Z_c)$ in (7) determines the distance from the projection $\Pi_c(p_i)$ of a point p_i into view c, to the nearest point on the object boundary Z_c. In order to speed up this common calculation, we pre-calculate a lookup table for Δ by means of the well known distance transform, to obtain a Chamfer image [20]. Each pixel in a Chamfer image contains the distance from this pixel to the nearest point on the segmented curve Z_c in view c. Calculating the Chamfer images has to be done only once and runs in linear time, by two passes over the image. In this way, we trade memory usage for computational speed.

The reprojection of the 3D curve in the images is done in the standard way [11]. We assume the cameras are described using a single radial distortion parameter κ, focal lengths f_x and f_y, principal point (p_x, p_y), and skew s. The pose of a camera is given by a rotation R and translation t. The projection of a 3D point X into an image is then given by

$$\Pi(X) = D(K[R|t], \kappa, X)$$

where K is the 3×3 calibration matrix

$$K = \begin{pmatrix} f_x & s & p_x \\ & f_y & p_y \\ & & 1 \end{pmatrix}$$

and D is a function that models the radial distortion.

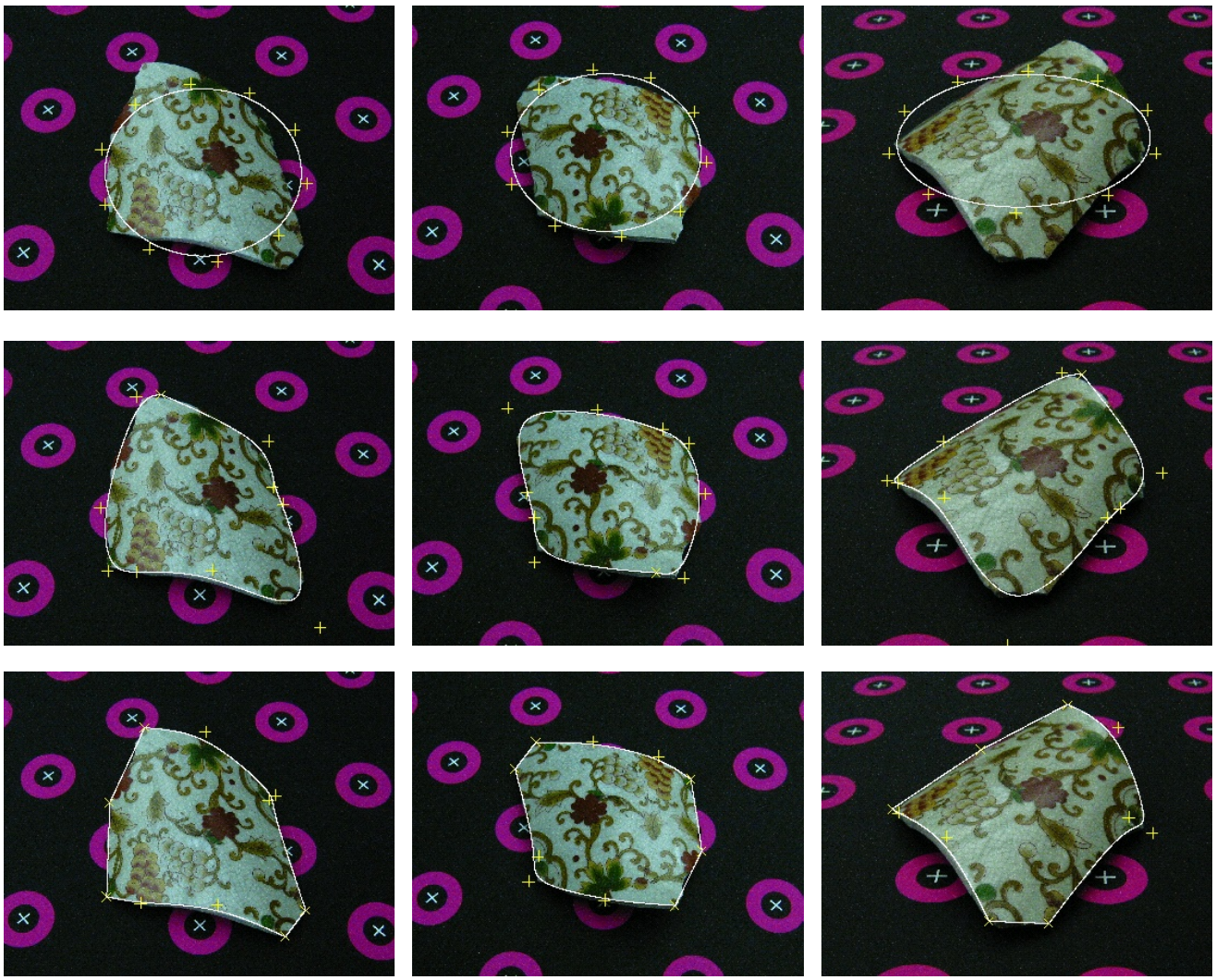
To minimize (7) given a specific tag configuration T, we use Levenberg-Marquardt non-linear optimization. To obtain the derivative $\frac{\partial E}{\partial \Theta_0}$ of the error function E with respect to the original control points Θ_0, we apply the chain rule to combine the derivatives of the camera projections and the derivative S from equation 2 on page 2 of the subdivision curve with respect to the original control points. Implementing the chain rule involves a pointwise multiplication of the projected curve derivatives with the Chamfer image gradients, which are estimated by convolution with a Gaussian derivative mask.

5. Results

Pictures of two pot shards were taken (see Figures 3 and 4) and calibrated with standard computer vision methods, using a calibration grid, that is partially visible in the images. For the first pot shard, six of the images were used for curve fitting, two of which are displayed in columns a and b of Figure 3. The image shown in the third column is not used for fitting and serves as a verification view. It is especially suited for that purpose, since it is taken at a much lower angle than the other six images.

(a) (b) (c)

Figure 3. Results are shown for the first pot shard. Projections of the control points are drawn as yellow '$+$' for untagged and yellow '$\times$' for tagged ones, and the corresponding subdivision curve is drawn in white. Six views are used for the fitting process, two of which are shown in columns (a) and (b). The third column (c) shows a view that is not used for fitting, and is taken from a lower angle than the other six images.
The first row shows the projections of the initial configuration of the control points, and the corresponding 3D subdivision curve (error: $3.7 \cdot 10^5$). The second row shows results after two iterations (error: $1.9 \cdot 10^3$). In the bottom row, the subdivision curve after 16 iterations fits the boundary of the shard pretty well (error: $7.7 \cdot 10^2$).



(a) (b) (c)

Figure 4. Results are shown for the second pot shard. Again, projections of the control points are drawn as yellow '$+$' for untagged and yellow '$\times$' for tagged ones, and the corresponding subdivision curve is drawn in white. This time four views are used for the fitting process, two of which are shown in columns (a) and (b). The third column (c) shows a view that is not used for fitting, and is taken from a lower angle than the other four images.
The first row shows the projections of the initial configuration of the control points, and the corresponding 3D subdivision curve (error: $2.4 \cdot 10^5$). The second row shows results after two iterations (error: $3.7 \cdot 10^3$). In the bottom row, the subdivision curve after 30 iterations fits the boundary of the shard pretty well (error: $1.1 \cdot 10^3$).

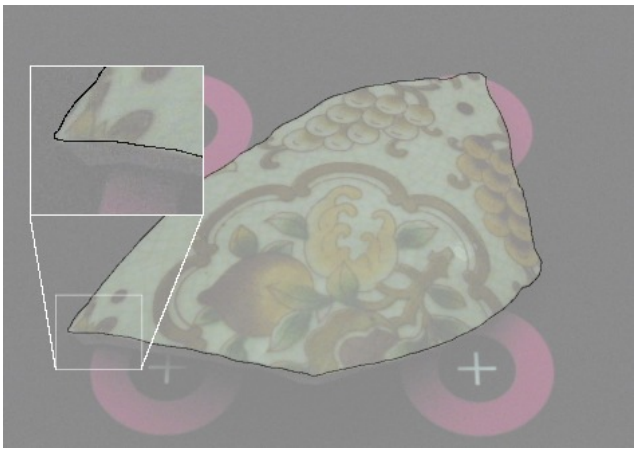


Figure 5. One of the hand-marked object boundaries in 2D.

The outlines of the shards were marked by hand as shown in Figure 5. Fitting starts with a circular distribution of a fixed number of control points in a plane parallel to the ground plane as initial guess, as shown in the first row of Figures 3 and 4. Each proposition for the Markov Chain consists of a random change in the tag configuration, by inverting each tag with some probability, followed by non-linear optimization of the control point locations. For evaluation of the error function, the curve is subdivided four times. Starting with nine control points for the first shard, this results in 144 points, for which the final function values are calculated with the evaluation mask.

After only two iterations, as shown in the second row of Figures 3 and 4, the subdivision curve adapted pretty well to the boundary, but the tagging process is only at the beginning, and some corners are not represented well. After 16 iterations for the first, and 30 for the second, the tagging configuration also adapted to fit the shard boundary. Note that we *sample* from the posterior distribution, and the tags never converge to a single combination.

In what follows, the posterior distribution is determined by throwing away the first 500 of 1500 samples to make the result independent of the initialization. Evaluation of 5000 samples did not show a significant difference in the values. One example of an interesting marginal distribution that can be approximated from these samples is the probability for each control point of being tagged, as shown in Figure 6. Four control points are clearly above random, suggesting sharp corners. Looking at the real pot shard, there are four obvious corners, and at least one more could be represented non-smoothly. Figure 7 shows the distribution over the number of tagged control points as another example for a marginal distribution. The average number of non-smoothly interpolated features of the boundary is just above

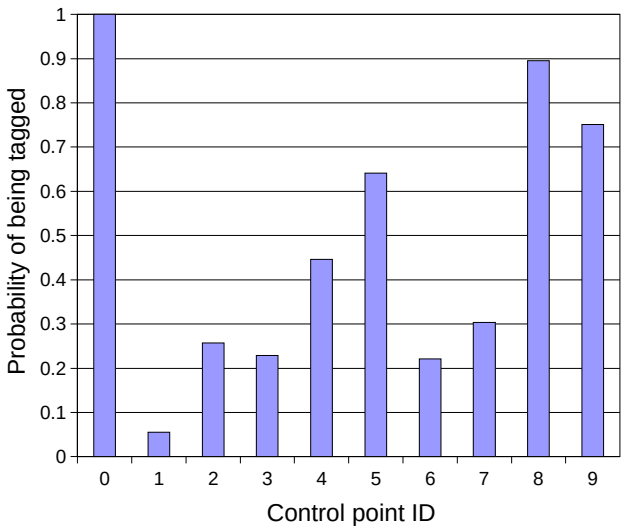


Figure 6. The probability of being tagged in the posterior distribution of the second pot shard is shown for each control point, as an example of an interesting marginal distribution that can be approximated from the samples. Clearly, control points 0, 5, 8 and 9 should be tagged, while all others, with the possible exception of control point 4 should not be tagged.

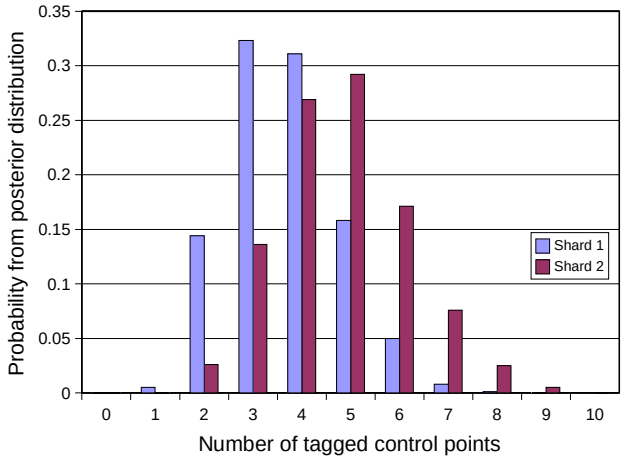


Figure 7. The distribution over the number of tagged control points in the posterior is shown for both pot shards, suggesting that the optimal number is near three for the first, and near five for the second shard.

three for the first shard and below five for the second.

6. Conclusion

We investigated modeling piecewise smooth curves with tagged subdivision curves, that provide a simple and flexible representation. The parameters of these curves were determined by a Rao-Blackwellized sampler, in which the optimal locations of the control points were integrated out and determined by non-linear optimization, and only the distribution over the tag configurations was sampled over.

This method was successfully applied to 3D reconstruction from multiple images, and results were obtained for boundary reconstruction of two pot shards from multiple images. Starting from an initial circular distribution of the control points, the algorithm approximated the object boundary well within a few sampling steps.

As a next step, automatic segmentation of the object boundary in the images needs to be added. Especially since a suitable background can be chosen in the archaeological application, this should not be too hard of a problem. More critical are partially occluded boundaries, and the fact that a shard has two such boundaries, one around each of the inner and outer surface.

Furthermore, the quality of the Gaussian assumption for Rao-Blackwellization warrants some examination. A possible approach currently under investigation is MCMC sampling over the control point locations for a given tag configuration.

A future extension of this work is to allow the algorithm to add or delete control points, to find the optimal number automatically, as is done in [6] for fitting splines to curves with reversible jump MCMC. As mentioned before, comparing features of boundaries from different shards could be used for reconstruction of archaeological artifacts, possibly in connection with other features, like texture and surface curvature, as suggested in [14].

Acknowledgments

We would like to thank Peter Presti from IMTC for providing the shard images.

References

[1] C. Bajaj and G. Xu. Data fitting with cubic A-splines. In *Computer Graphics Int.*, 27 –1 1994.

[2] R. Berthilsson, K. Åström, and A. Heyden. Reconstruction of curves in $\mathbb{R}^3$, using factorization and bundle adjustment. In *Int. Conf. on Computer Vision (ICCV)*, volume 1, pages 674–679, 1999.

[3] G. Casella and C. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, March 1996.

[4] T.-J. Cham and R. Cipolla. Stereo coupled active contours. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1094–1099, 1997.

[5] T.-J. Cham and R. Cipolla. Automated B-spline curve representation incorporating MDL and error-minimizing control point insertion strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(1):49–53, 1999.

[6] D. G. T. Denison, B. K. Mallick, and A. F. M. Smith. Automatic bayesian curve fitting. *Journal of the Royal Statistical Society, Series B*, 60(2):333–350, 1998.

[7] I. DiMatteo, C. R. Genovese, and R. E. Kass. Bayesian curve fitting with free-knot splines. In *Biometrika*, pages 1055–1071, 2001.

[8] A. Gelfand and A. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409, June 1990.

[9] W. Gilks, S. Richardson, and D. Spiegelhalter, editors. *Markov chain Monte Carlo in practice*. Chapman and Hall, 1996.

[10] P. Green. Reversible jump Markov chain Monte Carlo computation and bayesian model determination. *Biometrika*, 82:711–732, 1995.

[11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[12] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction. *Computer Graphics*, 28(Annual Conference Series):295–302, 1994.

[13] J. Kaminski, M. Fryers, A. Shashua, and M. Teicher. Multiple view geometry of non-planar algebraic curves. In *Int. Conf. on Computer Vision (ICCV)*, volume 2, pages 181–186, 2001.

[14] I. Kanaya, Q. Chen, Y. Kanemoto, and K. Chihara. Three-dimensional modeling for virtual relic restoration. In *MultiMedia IEEE*, volume 7, pages 42–44, 2000.

[15] D. MacKay. Introduction to Monte Carlo methods. In M.I.Jordan, editor, *Learning in graphical models*, pages 175–204. MIT Press, 1999.

[16] B. K. Mallick. Bayesian curve estimation by polynomials of random order. *J. Statist. Plan. Inform.*, 70:91–109, 1997.

[17] E. Punskaya, C. Andrieu, A. Doucet, and W. Fitzgerald. Bayesian curve fitting using MCMC with applications to signal segmentation. *IEEE Transactions on Signal Processing*, 50:747–758, March 2002.

[18] C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 1999.

[19] E. Stollnitz, T. DeRose, and D. Salesin. *Wavelets for computer graphics: theory and applications*. Morgan Kaufmann, 1996.

[20] E. Thiel and A. Montanvert. Chamfer masks: Discrete distance functions, geometrical properties and optimization. In *Pattern Recognition*, pages 244–247, August 1992.