MCMC-based Multiview Reconstruction of Piecewise Smooth Subdivision Curves with a Variable Number of Control Points

Michael Kaess, Rafal Zboinski, and Frank Dellaert

College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, USA
{kaess,rafal,frank}@cc.gatech.edu

Abstract. We investigate the automated reconstruction of piecewise smooth 3D curves, using subdivision curves as a simple but flexible curve representation. This representation allows tagging corners to model non-smooth features along otherwise smooth curves. We present a reversible jump Markov chain Monte Carlo approach which obtains an approximate posterior distribution over the number of control points and tags. In a Rao-Blackwellization scheme, we integrate out the control point locations, reducing the variance of the resulting sampler. We apply this general methodology to the reconstruction of piecewise smooth curves from multiple calibrated views, in which the object is segmented from the background using a Markov random field approach. 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 reconstruction of piecewise smooth 3D curves from multiple calibrated views. Among other applications, this is useful for the reconstruction of shards and other artifacts that are known to have "jagged edges". A motivating example of a broken pot-shard is shown in Figure 1. Such objects frequently show up in large museum collections and archaeological digs, and hence solving this problem would have important implications in preserving our cultural heritage. One possible application of our work is the automatic reconstruction of archaeological artifacts [1,2]. Apart from these applied uses, the problem of representing and performing inference in the space of piecewise smooth curves is of interest in its own right, and the methods developed here have potential application for other types of objects that have both continuous and discrete parameters that need to be optimized over.

We focus on the reconstruction of curves rather than surfaces. Existing 3D surface reconstruction methods rely on automatically extracted points and/or lines [3]. In the case of textured objects or when using structured light, these

To appear in ECCV 2004. © Springer-Verlag.

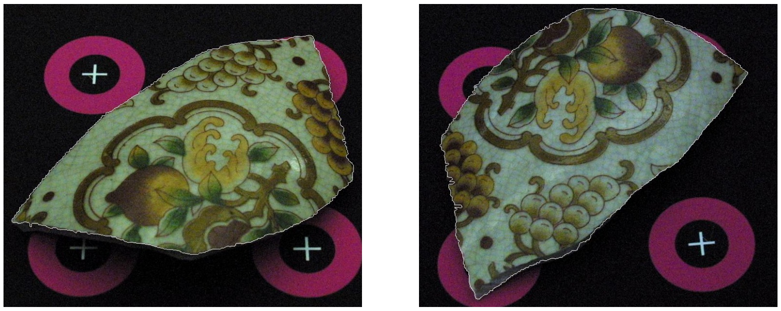


Fig. 1. Two automatically segmented pot-shard images. The boundaries derived from an MRF-based segmentation algorithm are shown in white.

methods can be used successfully to densely sample from the 3D surface of an object. However, these methods 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. This is inspired by the work of Hoppe [4], who successfully used tagged subdivision surfaces for fitting piecewise smooth surfaces to 3D point clouds. The curve fitting literature includes use of algebraic curves [5,6], piecewise polynomials [7,8,9], point curves [10], and B-splines [11,12,13].

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 [14]. In [4], Hoppe introduces piecewise smooth subdivision *surfaces*, allowing to model sharp features such as creases and corners by tagging the corresponding control points. We apply the tagging concept to subdivision *curves* to represent piecewise smooth curves. Earlier, we have presented this idea with some preliminary results in a workshop paper [15]. In this paper we significantly extend our approach to automatically determine the number of control points needed for a good fit. Furthermore, we replaced the manual segmentation process with an automatic MRF-based segmentation preprocessing step, obtaining a completely automated system.

To infer the parameters of these curves from the data, we propose Rao-Blackwellized sampling. In our approach, Markov chain Monte Carlo (MCMC) sampling [16] is used to obtain a posterior distribution over the discrete variables, while the continuous control point locations are integrated out after a non-linear optimization step. We also sample over the number of control points, using the framework of reversible jump (RJ) MCMC that was introduced by Green [17] and later described in a more easily accessible way as trans-dimensional MCMC [18]. In related work, Denison and Mallick [7,8] propose fitting piecewise polynomials with an unknown number of knots using RJMCMC sampling. Punskaya [9] extends this work to unknown models within each segment with applications in signal segmentation. DiMatteo [13] 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, multiple 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. While much of the curve fitting literature is concerned with 1D curves for signal processing [8,7,13,9], 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 [6] and image contours [11]. For the special case of stereo cameras, [12] 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 [5] using algebraic curves and in [10] for point curves with uncalibrated cameras.

2 Subdivision Curves

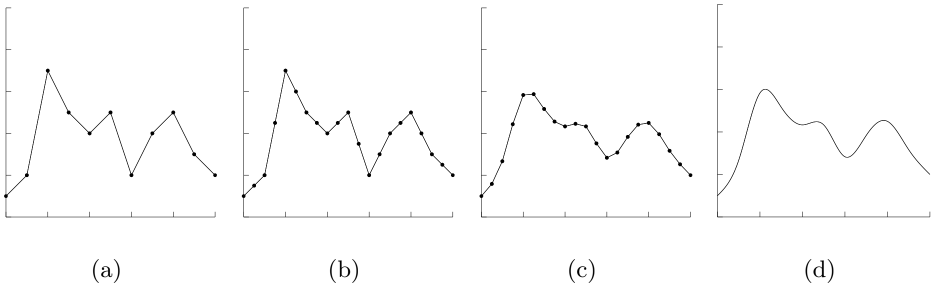


Fig. 2. The subdivision process: (a) initial control points, linearly interpolated; (b) the original mesh, subdivided by introducing average points; (c) the result after application of the averaging mask to the subdivided control points; (d) converged curve.

Here we briefly review subdivision curves. See [14] for more details. 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 2: a *splitting step* that introduces midpoints:

$$x_{2i}^{t+1} \triangleq x_i^t \qquad x_{2i+1}^{t+1} \triangleq \frac{1}{2}\left(x_i^t + x_{i+1}^t\right)$$

and an *averaging step* that computes weighted averages:

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

The type of the resulting curve depends on the *averaging mask* r. Subdivision can be used to create a wide range of functions [14], including uniform and non-uniform B-splines, and functions that have no analytic representation like Daubechies wavelets. For example, the mask for a cubic B-spline is $\frac{1}{4}(1,2,1)$.

As explained in [14], 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 increasingly 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 this mask is applied.

It is convenient for the exposition below to view the *entire* subdivision process as a 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 is 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 this case, the derivative equation (2) holds for each dimension separately.

3 Piecewise Smooth Subdivision Curves

There are a number of ways to represent sharp corners in otherwise smooth curves. One solution is to place multiple control points at the location of a corner, but consecutive subdivision creates superfluous points at this location. Furthermore, it places unwanted constraints on the adjacent curve segments. A commonly used method in connection with B-splines is extra knot insertion.

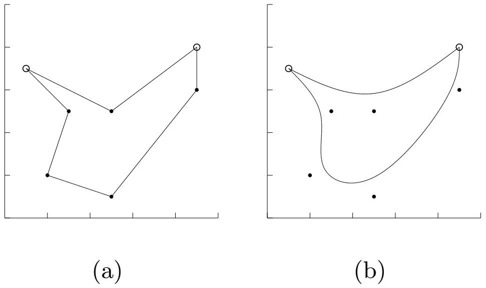


Fig. 3. A tagged 2D subdivision curve. Tagged points are drawn as circles. (a) the original control points, (b) the converged curve with non-smoothly interpolated points.

We employ a more general method here based on work of Hoppe [4] for subdivision surfaces. It allows us to "tag" control points, allowing different averaging masks to be used at these points. E.g., using the mask $(0, 1, 0)$ forces the interpolation of the control point and at this point introduces a discontinuity in the derivative, while retaining the smooth curve properties for all other points of the curve (Figure 3). The number of tagged control points does not increase during the subdivision process, and so the non-smoothness of the curve is restricted to the tagged control points, which will always be interpolated.

Below we will use the following notation to describe tagged 3D subdivision curves. The locations of the 3D control points are given by $\Theta \stackrel{\Delta}{=} \{x_0^0, x_1^0, ..., x_{n-1}^0\}$, 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 $T \stackrel{\Delta}{=} \{b_0, b_1, ..., b_{n-1}\}$.

4 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 5 this will be specialized to the problem of fitting from multiple, calibrated segmentations of an object.

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

$$P(n, \Theta_n, T_n|Z) \propto P(Z|n, \Theta_n, T_n)P(\Theta_n|n, T_n)P(T_n|n)P(n) \qquad (3)$$

over the possible number of control points n, the control point values Θ_n and the tag variables T_n. Here the control points $\Theta_n \in \mathbb{R}^{3n}$ are continuous and the tags

$T_n \in \{0,1\}^n$ are discrete. The *likelihood* $P(Z|n, \Theta_n, T_n)$ and *control polygon prior* $P(\Theta_n|n, T_n)$ are application specific and will be specified in Section 5. Choosing the *complexity prior* $P(n)$ to be uniform over a range of valid values allows us to find an unbiased distribution over the number of control points. Similarly, we use an uninformative *tagging prior* $P(T_n|n)$, but a binomial distribution $P(T_n|n) \propto p^c(1-p)^{n-c}$ over the number of active tags c could also be used.

4.1 Trans-Dimensional MCMC

Since the number of possible tag configurations is 2^n and hence exponential in n, we propose to use reversible jump Markov chain Monte Carlo (MCMC) sampling [18,17] to perform approximate inference. MCMC methods produce 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. The specific MCMC algorithm we use is the trans-dimensional MCMC algorithm from [18]:

1. Start with a random initial state $X^{(0)}$.
2. Propose a move type $m \in M$ with probability $j(m)$.
3. Generate a random sample u from the move-specific proposal density g_m. The move type m and random sample u determine how to move from the current state $X^{(r)}$ to the proposed state X'.
4. Calculate the corresponding reverse move (m', u').
5. Compute the acceptance ratio
$$a = \frac{\pi(X')}{\pi(X^{(r)})} \frac{j(m')}{j(m)} \frac{g_{m'}(u')}{g_m(u)} \left| \frac{\partial(x', u')}{\partial(x, u)} \right| \tag{4}$$
where the Jacobian factor corrects for the change in variables (see below).
6. Accept $X^{(r+1)} \leftarrow X'$ with probability $\min(a, 1)$, otherwise $X^{(r+1)} \leftarrow X^{(r)}$.

The generated sequence of states $\{X^{(r)}\}$ 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, one possible set of move types consists of "Up" and "Down" for inserting and deleting control points and "Modify" for flipping one or more of the tags. Care has to be taken to ensure that the move from (x, u) to (x', u') is reversible and therefore a diffeomorphism. One requirement is that the dimensions on both sides have to match. Note that in our case the Jacobian of the diffeomorphism $\left| \frac{\partial(x', u')}{\partial(x, u)} \right|$ is always 1 because we integrate out the continuous part of the space (see below).

4.2 Rao-Blackwellization

Sampling over the joint discrete-continuous space is expensive. A crucial element of our approach is to not sample from the joint posterior (3) but rather from the marginal distribution $P(n, T_n|Z)$ over the number of points n and the tags T:

$$\pi(X) \triangleq P(n, T_n|Z) = \int P(n, \Theta_n, T_n|Z) d\Theta_n \tag{5}$$

while performing the integration (5) above analytically.

From a sample over n and T of size N we can obtain a high quality approximation to the joint posterior as follows

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

with $\delta(.,.)$ being the Kronecker delta. Thus, (6) approximates the joint posterior $P(n, \Theta_n, T_n|Z)$ as a combination of discrete samples and continuous conditional densities $P(\Theta_n|Z, (n, T_n)^{(r)})$.

Integrating out the continuous part of the state space reduces the number of samples needed. The superiority of (6) over a density estimate based on joint samples is rooted in an application of the Rao-Blackwell theorem, which is why this technique is often referred to as *Rao-Blackwellization* [19,20,21]. Intuitively, the variance of (6) is lower because it uses exact conditional densities to approximate the continuous part of the state. As such, far fewer samples are needed to obtain a density estimate of similar quality.

Substituting the factorization of the posterior (3) in (5) we obtain

$$P(n, T_n|Z) = kP(n)P(T_n|n) \int P(Z|n, \Theta_n, T_n) P(\Theta_n|n, T_n) d\Theta_n$$

Assuming the conditional posterior $P(Z|n, \Theta_n, T_n)P(\Theta_n|n, T_n)$ is approximately normally distributed around the MAP estimate of the control points Θ_n^*

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

the integral can be approximated via Laplace's method, and we obtain the following target distribution over the number of control points n and the tags T_n:

$$P(n, T_n|Z) = kP(n)P(T_n|n)\sqrt{|2\pi\Sigma|} P(Z|n, \Theta_n^*, T_n) P(\Theta_n^*|n, T_n)$$

The MAP estimate Θ_n^* can be found by non-linear optimization (see below).

5 Multiple View Fitting

In this section we specialize the general methodology of Section 4 to reconstructing tagged 3D 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. For the results presented below, the object boundaries are segmented automatically using a Markov random field approach.

To calculate an objective function, the 3D curve given by Θ_n and T_n is subdivided m times 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:

$$P(Z|n, \Theta_n, T_n) \propto e^{-\frac{1}{2\sigma^2} E(Z,n,\Theta_n,T_n)}$$

where σ^2 is a variance of the noise in the measurements. The error function E above is obtained as a sum of squared errors

$$E(Z, n, \Theta_n, T_n) = \sum_{c=1}^{C} \sum_{i=1}^{n2^m} \Delta(\Pi_c(p_i), Z_c)^2 \tag{7}$$

with one term for each of the $n2^m$ final subdivision curve points in each of the C images, explained in more detail below. The prior $P(n)$ on the number of control points, 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 $\Delta(\Pi_c(p_i), Z_c)$ 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 [22]. 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. In this way, we trade memory usage for computational speed.

The outlines of the shards were automatically segmented as foreground and background classes using a Markov random field (MRF) approach. To offset the analytical intractability associated with MRFs, we employed Gibbs sampling to approximate the posterior probability $P(Z|I)$ of the outlines Z given the input images I. The sampling is initialized by selecting the class with highest likelihood for each pixel. Consequently, Gibbs sampling requires only a few samples to achieve accurate segmentations.

The reprojection of the 3D curve in the images is done in the standard way [3]. 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

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

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 4 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.

6 Results

We illustrate our approach on two sets of pot-shard images, shown in Figure 4. The shards were placed on a calibration pattern, which allowed us to easily optimize for the camera calibration parameters. Six images for the first pot shard and four images for the second were used for curve fitting, all of which are taken from about the same angle from different sides of the shard. Two of those views are shown in columns (a) and (b) of Figure 4. The images in column (c) were not used for fitting and serve as verification views. They are informative, since they are taken at a much lower angle than all other images. The shard images were automatically segmented by an MRF approach, as described in Section 5. Figure 1 shows two such segmentations overlayed on the corresponding shard images.

The fitting process starts with a circular configuration of a small number of control points in a plane parallel to the ground plane, as shown in the first row of Figure 4. Each proposal for the Markov Chain consists of a random move as described in Section 4.1, followed by non-linear optimization of the control point locations. The move can either introduce a new control point, delete an existing one, or change the tag configuration by inverting each tag with probability $\frac{1}{n}$. For evaluation of the error function, the curve is subdivided four times before the limit positions of the points are calculated using the evaluation mask.

After only five iterations, as shown in the second row of Figure 4, the subdivision curve adapted pretty well to the boundary, but the number of control points is still too small and the tagging is wrong. After 250 iterations the number of control points and the tagging configuration both adapted to fit the shard boundary well. Note that we *sample* from the posterior distribution, and the number of control points and the tagging configuration never converge to a single value.

The algorithm finds a suitable number of control points independent of the initial number, as can be seen in Figure 5(a), where this number is plotted over time. Independent of the initial number of control points, the algorithm converges quickly to a “optimal” number that is influenced by the curve itself and the variance of the measurements.

The diagram in Figure 5(a) shows nicely the burn-in phase of the Markov chain. In what follows, the posterior distribution is determined by throwing away the first 500 of the 1500 samples to make the result independent of the initialization. 1500 samples seem sufficient, an evaluation of 9000 samples did not show a significant change in results.

One example of a marginal distribution that can be approximated from these samples is the number of control points that are needed to fit the curve well, as shown in Figure 5(b).

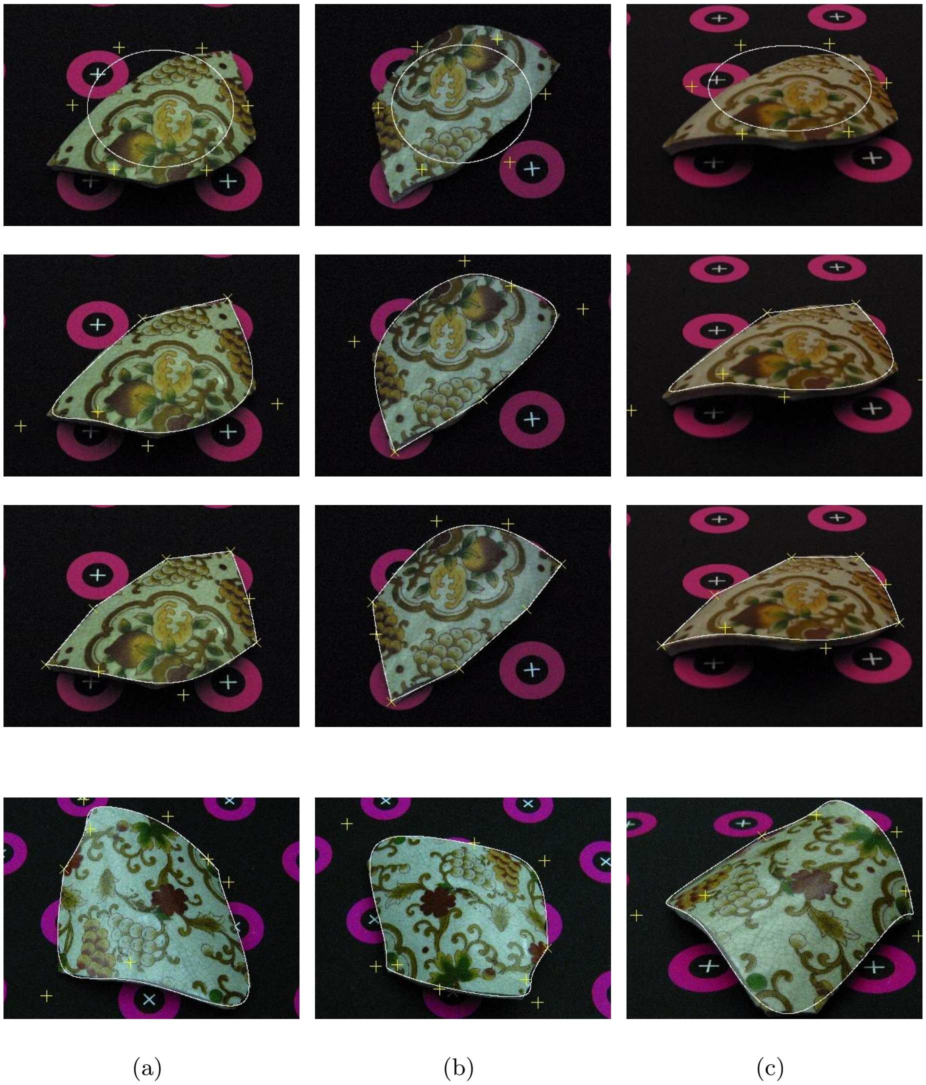
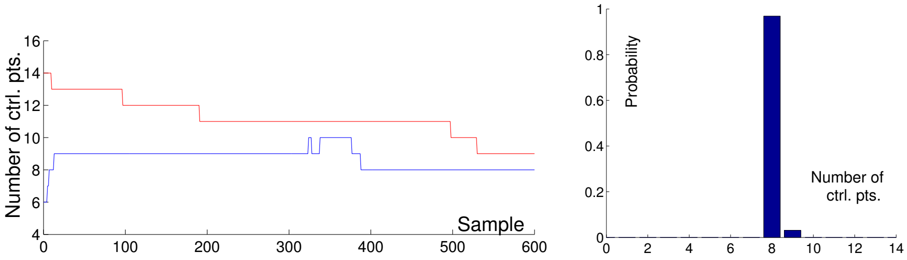


Fig. 4. Results are shown for two pot shards. Projections of the control points are drawn as yellow '+' for untagged and yellow '×' for tagged ones, and the corresponding subdivision curve is drawn in white. Six views are used for the fitting process of the first shard, while only four are used for the second shard. In both cases, two of those 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 all other images.
The first row shows the projections of the initial control points and the corresponding 3D subdivision curve on images of the first shard. The second row shows results after five iterations (error: $9.8 \cdot 10^2$). The third row is sample number 250 (error: $4.3 \cdot 10^2$). The last row is the result for the second shard after 250 samples.



(a) (b)

Fig. 5. (a) Number of control points during the burn-in phase. The lower curve starts with six control points, the upper one with 14. (b) Probability over different numbers of control points after burn-in phase.

7 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 number of control points and the tag configurations was sampled over.

This method was successfully applied to 3D curve reconstruction from multiple images, as illustrated in the results section. These results were obtained in automated fashion, using an MRF-based segmentation approach to automatically segment the images with a known calibration background. Then, starting from an initial circular distribution of the control points, the algorithm approximated the object boundary well within a small number of sampling steps.

It would be of interest to more closely examine the quality of the Gaussian assumption made in the Rao-Blackwellization step. One way to validate this assumption is by MCMC sampling over the control point locations for a given tag configuration. Also, on the image processing side, there are some problems with double shard boundaries and occluding contours that need to be resolved in order to create a robust, automated system.

We would like to connect our 3D curve reconstruction methods with a complete "broken-pot" reconstruction such as described in [2]. 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 [1]. Finally, it is our hope that the general methodology described here can be successfully applied in other discrete-continuous reconstruction settings.

Acknowledgments

We would like to thank Peter Presti from IMTC for providing the shard images and Zia Khan for valuable discussions on reversible jump MCMC.

References

1. Kanaya, I., Chen, Q., Kanemoto, Y., Chihara, K.: Three-dimensional modeling for virtual relic restoration. In: MultiMedia IEEE. Volume 7. (2000) 42–44
2. Cooper, D., Willis, A., Andrews, S., Baker, J., Cao, Y., Han, D., Kang, K., Kong, W., Leymarie, F., Orriols, X., Velipasalar, S., Vote, E., Joukowsky, M., Kimia, B., Laidlaw, D., Mumford, D.: Bayesian virtual pot-assembly from fragments as problems in perceptual-grouping and geometric-learning. In: Intl. Conf. on Pattern Recognition (ICPR), IEEE Computer Society Press (2002) 11–15
3. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000)
4. Hoppe, H., DeRose, T., Duchamp, T., Halstead, M., Jin, H., McDonald, J., Schweitzer, J., Stuetzle, W.: Piecewise smooth surface reconstruction. Computer Graphics **28** (1994) 295–302
5. Kaminski, J., Fryers, M., Shashua, A., Teicher, M.: Multiple view geometry of non-planar algebraic curves. In: ICCV. Volume 2. (2001) 181–186
6. Bajaj, C., Xu, G.: Data fitting with cubic A-splines. In: Comp. Graph. Int. (1994)
7. Denison, D.G.T., Mallick, B.K., Smith, A.F.M.: Automatic bayesian curve fitting. Journal of the Royal Statistical Society, Series B **60** (1998) 333–350
8. Mallick, B.K.: Bayesian curve estimation by polynomials of random order. J. Statist. Plan. Inform. **70** (1997) 91–109
9. Punskaya, E., Andrieu, C., Doucet, A., Fitzgerald, W.: Bayesian curve fitting using MCMC with applications to signal segmentation. IEEE Transactions on Signal Processing **50** (2002) 747–758
10. Berthilsson, R., Åström, K., Heyden, A.: Reconstruction of curves in R^3, using factorization and bundle adjustment. In: ICCV. Volume 1. (1999) 674–679
11. Cham, T.J., Cipolla, R.: Automated B-spline curve representation incorporating MDL and error-minimizing control point insertion strategies. PAMI **21** (1999) 49–53
12. Cham, T.J., Cipolla, R.: Stereo coupled active contours. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (1997) 1094–1099
13. DiMatteo, I., Genovese, C.R., Kass, R.E.: Bayesian curve fitting with free-knot splines. In: Biometrika. (2001) 1055–1071
14. Stollnitz, E., DeRose, T., Salesin, D.: Wavelets for computer graphics: theory and applications. Morgan Kaufmann (1996)
15. Kaess, M., Dellaert, F.: Reconstruction of objects with jagged edges through rao-blackwellized fitting of piecewise smooth subdivision curves. In: Workshop on Higher Level Knowledge in Computer Vision at ICCV. (2003)
16. Gilks, W., Richardson, S., Spiegelhalter, D., eds.: Markov chain Monte Carlo in practice. Chapman and Hall (1996)
17. Green, P.: Reversible jump Markov chain Monte Carlo computation and bayesian model determination. Biometrika **82** (1995) 711–732
18. Green, P.: Trans-dimensional Markov chain Monte Carlo. Chapter in Highly Structured Stochastic Systems (2003)
19. Gelfand, A., Smith, A.: Sampling-based approaches to calculating marginal densities. J. Am. Statistical Association **85** (1990) 398–409
20. Casella, G., Robert, C.: Rao-Blackwellisation of sampling schemes. Biometrika **83** (1996) 81–94
21. Robert, C., Casella, G.: Monte Carlo Statistical Methods. Springer (1999)
22. Thiel, E., Montanvert, A.: Chamfer masks: Discrete distance functions, geometrical properties and optimization. In: Pattern Recognition. (1992) 244–247