

Superresolution Texture Maps for Multiview Reconstruction

Bastian Goldluecke and Daniel Cremers

Computer Science Department, University of Bonn, Germany

{bastian,dcremers}@cs.uni-bonn.de

<http://www-cvpr.iai.uni-bonn.de>

Abstract

We study the scenario of a multiview setting, where several calibrated views of a textured object with known surface geometry are available. The objective is to estimate a diffuse texture map as precisely as possible. A superresolution image formation model based on the camera properties leads to a total variation energy for the desired texture map, which can be recovered as the minimizer of the functional by solving the Euler-Lagrange equation on the surface. The PDE is transformed to planar texture space via an automatically created conformal atlas, where it can be solved using total variation deblurring. The proposed approach allows to recover a high-resolution, high-quality texture map even from lower-resolution photographs, which is of interest for a variety of image-based modeling applications.

1. Introduction

Recently, image-based methods in conjunction with laser range scans have emerged as a practical way to obtain high-quality 3D models, in particular for applications like digital preservation [3]. Increasing accuracy of image-based 3D reconstruction methods could soon make it feasible to even rely on image-based modeling alone [21]. Regardless of how geometry is obtained though, in most applications it is important to recover a high-resolution and accurate texture map for the model, in order to be able to render the model from novel viewpoints and in different lighting conditions. In multi-view settings, usually every patch of the surface is captured from several cameras. Therefore, using a suitable superresolution model, one should be able to recover the texture map in higher resolution than provided by the input images. However, this possibility has not yet been explored, with most existing methods fitting a local lighting model per-vertex or per-texel only, disregarding texel interdependencies. Thus, our paper aims at opening up the highly interesting area of superresolution models on surfaces.



Figure 1. Several input images for a multi-view reconstruction.

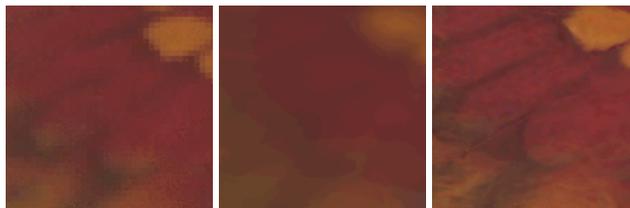


Figure 2. *Left:* Close-up of one of the low-resolution input images. *Center:* Rendered model with blurry texture initialized by weighted averaging of input images. *Right:* High-quality texture optimized with the proposed superresolution approach.

1.1. Image-based texture reconstruction

In image-based texture generation, one faces the problem of integrating several views of the same surface region into a single texture map. In most approaches to date, information from the cameras is blended per texel, using a suitable heuristic to weight contributions from different source images [3, 16, 24]. It is common to perform additional image registration before estimating surface properties, in order to correct for geometry errors [3, 8] or object motion [24]. Other algorithms circumvent texel-wise blending by selecting large regions which are assigned texture data from a single camera. However, those methods then face the problem of stitching together the different regions. Optimal seam locations can be estimated beforehand [15], in combination with multi-band blending at seam locations [1].

Although existing methods produce visually very pleasing results, there is one problem intrinsic to the approach of

selecting optimal cameras and blending the contributions. Namely, this technique does not scale favourably with the number of input cameras. Indeed, the result from blending becomes more blurred the more images are blended together, in particular if geometry estimates are not perfectly accurate. Thus, many algorithms end up throwing away most information in order to increase sharpness of the results. However, the fewer cameras contribute to a single patch, the more visible seams will arise at patch boundaries which have to be dealt with.

The proposed approach of texture superresolution delivers a solution to this dilemma. It is based on previous models employed in 2D superresolution [6, 10, 20, 22] for which there is a well-developed theory [2]. The main idea is to formulate an image formation process which explicitly models the downsampling by the cameras. A texture on the surface is then estimated to optimally fit all input images simultaneously, where a good fit is determined by minimization of an energy functional. The total variation is included as a regularizer in the functional. In comparison to the above methods, this approach has the following advantages:

- An arbitrary number of source images can be integrated in a natural way, globally reconstructing a spatially coherent texture map. No inherent scaling problems arise, in fact, we embrace having more cameras as this always leads to a more accurate solution.
- The model lends itself to an elegant and unified mathematical formulation. Suitable weighting factors for input image contributions arise naturally, and do not have to be imposed by heuristic assumptions.
- Neither visibility boundaries nor seams have to be treated explicitly, since the total variation norm in the energy already implies minimization of discontinuities.

Due to these unique advantages, the estimated texture maps are of excellent quality and contain more details than each single input image taken by itself, Fig. 2.

1.2. Contributions

The main contributions of this paper are the following. From a theoretical point of view, it introduces the first model for surface texture superresolution. To our knowledge, this is also the first time a superresolution model is solved on curved surfaces. While super-resolved textures were addressed previously, the authors either just perform a resampling on a higher-resolution grid and thus not actually solve a superresolution model [13, 19], or they neglect curvature and in effect perform 2D superresolution [7].

The theoretical derivation of the Euler-Lagrange equations is non-trivial, and also valid for the reconstruction of

other scalar fields with arbitrary point-wise error measures defined in image space. Thus, it will hopefully spark interest in future studies on surface PDEs for surface property reconstruction from multiple views.

In the next section, we introduce the image formation model. The optimal texture map can be recovered as the minimum of an energy functional, leading to a gradient descent approach to recover the texture. By computing a conformal atlas for the surface, one can formulate the Euler-Lagrange equation as a partial differential equation on texture space, Sect. 3. Details of the implementation are discussed in Sect. 4, followed by extensive experiments in Sect. 5.

2. Superresolution Texture Maps

In this section, we introduce the image formation model which leads to an energy functional whose minimum is attained at the unknown texture. Let $\mathcal{I}_1, \dots, \mathcal{I}_n : \Omega_i \rightarrow \mathbb{R}$ be the input images captured by cameras with known projections $\pi_1, \dots, \pi_n : \mathbb{R}^3 \rightarrow \mathbb{R}^2$. The cameras observe a known Lambertian surface $\Sigma \subset \mathbb{R}^3$, which is textured with the unknown texture map $T : \Sigma \rightarrow \mathbb{R}$. We will for now assume grayscale images and extend the model to color textures in Sect. 4.

2.1. Variational Formulation

The basic idea is to recover the unknown texture map as the minimizer of an energy functional, consisting of a data term and a regularization term,

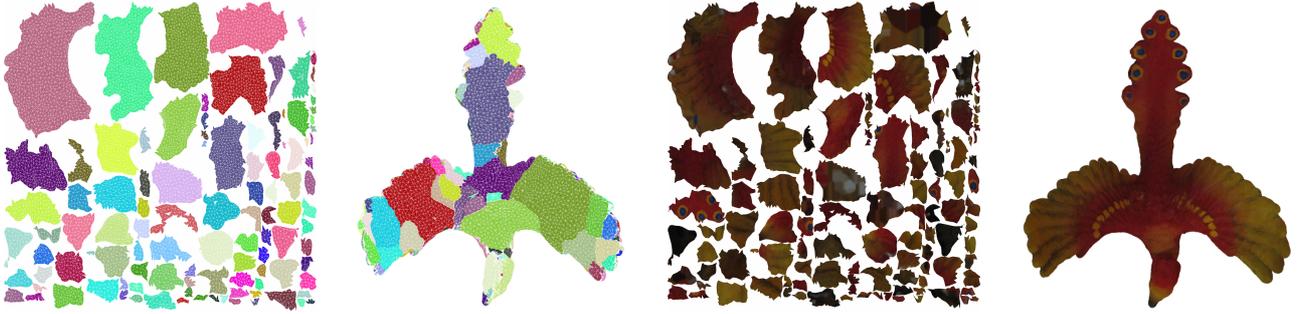
$$E(T) := E_{\text{data}}(T) + \sigma E_{\text{tv}}(T), \quad (1)$$

with $E_{\text{tv}}(T) := \int_{\Sigma} \|\nabla_{\Sigma} T\|_{\Sigma} \, ds$.

Here, $\sigma \geq 0$ is a parameter controlling the desired smoothness of the result. The differential operators on the surface and the norm on the tangent space are explained in detail in [17]. The total variation norm of the texture was chosen as the regularizer, because compared to alternatives, it is better suited to preserve a crisp texture with sharp high-resolution features,

The data term is based on the current state-of-the-art superresolution model [22], with the limitation that we currently do not take noise in the input images into account explicitly. The idea is that a real-world camera downsamples the input by integrating over the visible texels inside each sensor element. This integration process is modeled with a convolution kernel b , which can be derived from the properties of the camera [2]. The resulting data term is

$$E_{\text{data}}(T) := \sum_{i=1}^n \int_{\hat{S}_i} (b * (T \circ \beta_i) - \mathcal{I}_i)^2 \, dx. \quad (2)$$



(a) Chart domains $\text{dom}(\tau_j)$ in \mathbb{R}^2 forming the texture space \mathbb{T} . (b) Corresponding regions C_j on the surface. (c) Texture map $T = T \circ \tau_j$ on texture space. (d) Texture T mapped on surface.

Figure 3. Illustration of the charts and parametrization mappings. Charts are color-coded, i.e. different colors correspond to different charts.



Figure 4. Intensity-coded backprojection area element \mathcal{J}_i and corresponding input image. Some errors in the input geometry are clearly visible - no good texture can be expected in those regions.

$T \circ \beta_i$ denotes the visible texture intensity of the high-resolution input in the image plane. The backprojection mappings $\beta_i : S_i \rightarrow \Sigma$ assign the visible point on the surface to each point in the silhouettes $S_i := \pi_i(\Sigma) \subset \mathbb{R}^2$, Fig. 5. Actual integration takes place over the smaller subset $\hat{S}_i \subset S_i$ where all of the kernel covers only points within the silhouette.

2.2. Euler-Lagrange Equation

The next step is to establish necessary conditions for a minimum. Deriving the Euler-Lagrange equations of the functional leads to the following theorem, whose proof can be found in the appendix.

Theorem. *A texture which minimizes energy (1) also satisfies the partial differential equation*

$$\text{div}_\Sigma \left(\frac{\nabla_\Sigma T}{\|\nabla_\Sigma T\|_\Sigma} \right) + \sum_{i=1}^n \frac{v_i}{\sigma} ((\mathcal{J}_i \mathcal{D}_i) \circ \pi_i) = 0, \quad (3)$$

on the surface Σ .

Some new objects appear in the PDE: the difference images \mathcal{D}_i stem from the derivative of the data term, and are defined as

$$\mathcal{D}_i := \bar{b} * (b * (T \circ \beta_i) - \mathcal{I}_i), \quad (4)$$

with the mirrored kernel $\bar{b}(x) := b(-x)$. The functions $v_i : \Sigma \rightarrow \{0, 1\}$ indicate whether a surface point is visible in a source image,

$$v_i(s) := \begin{cases} 1 & \text{if } \pi_i(s) \in \hat{S}_i \text{ and } s = \beta_i \circ \pi_i(s), \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

Finally, \mathcal{J}_i is the inverse surface area element with respect to the backprojection,

$$\mathcal{J}_i(x, y) = \left\| \frac{\partial \beta_i}{\partial x} \times \frac{\partial \beta_i}{\partial y} \right\|^{-1}. \quad (6)$$

\mathcal{J}_i accounts for foreshortening of the surface in the input views, and is small in regions where the backprojection varies strongly, which is usually the case at silhouette boundaries or discontinuities of the backprojection due to self-occlusions, Fig. 4. As a consequence, we have the desirable property that in those regions where texture information from the image is unreliable, the input is assigned less weight. Note that we did not need any heuristic assumptions to arrive at this weighting scheme. It is rather a direct mathematical consequence of the variational formulation.

The main question arising now is how to solve a partial differential equation defined via derivative operators on the surface, which we will address in the next section.

3. Conformal Maps and Differential Operators

While it is possible and has some merits to solve surface PDEs based on an implicit representation [4], the goal here are high-resolution textures which exceed the feasible resolution of the voxel grids. Thus, a method based on parametrizing the surface [17, 23] is preferred for superresolution. The idea is to define the texture on the surface as a planar, 2D texture map. In order to achieve this, one needs a global parametrization of the surface. For the purposes here, it is defined as a collection of k charts (C_j, τ_j) with the following properties:

- The C_j are pairwise disjoint, open subsets of Σ , and the union of their closures $\cup_{j=1}^k \overline{C_j}$ covers Σ .
- Each $\tau_j : \text{dom}(\tau_j) \subset \mathbb{R}^2 \rightarrow C_j$ is a diffeomorphism, and the domains $\text{dom}(\tau_j)$ are pairwise disjoint.

The union $\mathbb{T} := \cup_{j=1}^k \text{dom}(\tau_j)$ of the chart domains is called the texture space. To simplify notation, the single mappings τ_j can be combined to form a global mapping $\tau : \mathbb{T} \rightarrow \Sigma$. Fig. 3 illustrates the concept. Now, the key idea for an elegant formulation of the required differential operators is to demand a last property:

- The parametrization is conformal, i.e. each τ_j is a conformal map. This means that their Jacobian matrix is everywhere a scalar λ , the conformal factor, times a rotation matrix.

Utilizing conformality enables us to explain the differential operators on the surface in terms of differentiation in \mathbb{R}^2 . Let $\lambda : \mathbb{T} \rightarrow \mathbb{R}$ be the mapping assigning the conformal factor to each point in texture space. Then the smoothness term of the Euler-Lagrange equation (3) can be expressed by pulling it back onto texture space [17]:

$$\text{div}_{\Sigma} \left(\frac{\nabla_{\Sigma} T}{\|\nabla_{\Sigma} T\|} \right) = \frac{1}{\lambda} \text{div} \left(\sqrt{\lambda} \frac{\nabla T}{\|\nabla T\|} \right), \quad (7)$$

where $\mathcal{T} := T \circ \tau$ is the texture map of the surface defined on the texture space \mathbb{T} , see Fig. 3c.

In summary, in order to compute the desired texture map \mathcal{T} , we need to solve the following

Superresolution problem: Find a texture map $\mathcal{T} : \mathbb{T} \rightarrow \mathbb{R}$ which satisfies the Euler-Lagrange equation

$$\frac{1}{\lambda} \text{div} \left(\sqrt{\lambda} \frac{\nabla T}{\|\nabla T\|} \right) + \sum_{i=1}^n \frac{v_i}{\sigma} ((\mathcal{J}_i \mathcal{D}_i) \circ \phi_i) = 0 \quad (8)$$

on the chart domains $\text{dom}(\tau_j)$, $j = 1, \dots, k$. Here, $\phi_i := \pi_i \circ \tau$ are the mappings from texture space into the image planes, λ assigns the conformal factor of the parametrization to each point in \mathbb{T} , and $v_i := v_i \circ \tau$ indicates visibility of a texel in image i . \mathcal{D}_i and \mathcal{J}_i are defined according to Eqns. (4) and (6), respectively.

Note that the surface is closed and technically, there are no boundary conditions. However, because a global parametrization usually requires several charts, \mathbb{T} is made up of several disjoint domains. Thus, the domain boundaries have to be handled correctly. After discretization, the chart neighbourhood relationships will impose a certain topology of the computation grid resembling periodic boundary conditions. The next section will detail the implementation of the PDE.

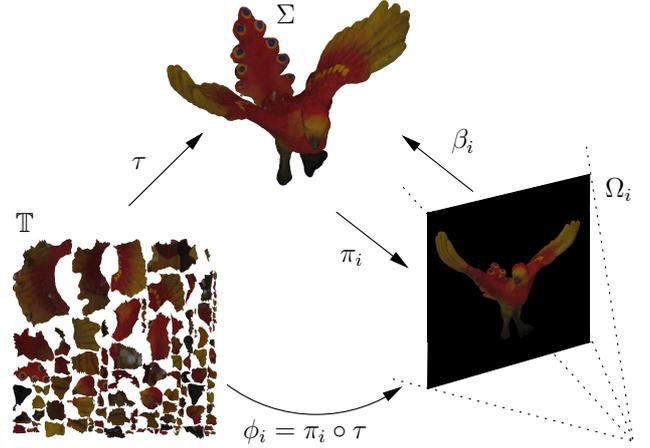


Figure 5. The various mappings connecting texture space \mathbb{T} , the surface Σ and the image planes Ω_i .

4. PDE-based Energy Minimization

Since the implementation of the superresolution model is very involved, we will give a detailed description of the necessary steps in this section. The first step will be to construct a conformal atlas. Once the mappings from the surface to texture space are established, the texture space needs to be discretized and the topology of the computation grid initialized using the chart neighbourhood relationships. On the grid, we employ a straightforward gradient descent scheme to solve the Euler-Lagrange equation, for which it is necessary to compute the total variation and data term. Finally, we describe the extension of the scheme to color textures and discuss runtime and memory requirements, as well as parameter settings.

4.1. Conformal Atlas and Computation Grid

Numerous methods for surface parametrization in general are available [9]. However, when one needs a conformal parametrization, the options are more limited [12, 18, 25]. Our algorithm to compute a conformal atlas is a straightforward implementation of [18]. An even coverage of the model with texels is desirable, so charts are split up if the ratio of largest to smallest conformal factor is greater than $3/2$. Chart domains are scaled so that the average conformal factor equals a common constant. The algorithm is fully automatic and has the desirable property that chart boundaries tend to coincide with high-curvature edges on the surface.

For discretization, the texture space is subdivided into a grid of texels. The grid needs to admit a flexible topology, since only texels in the interior of chart domains are connected to their direct neighbours. On the boundary of charts, neighbourhood is established according to the correct relationships on the surface. To achieve this, we take

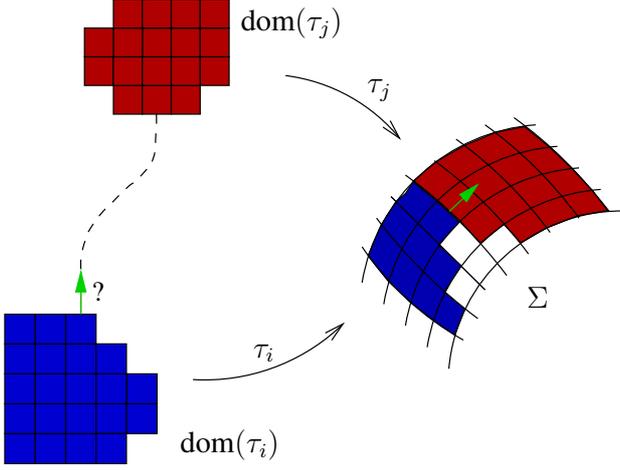


Figure 6. Boundary texel neighbour connections on the computation grid are established by searching in normal direction on the surface.

the outer normals of boundary texels in texture space and transform them up onto the surface. Then, we search for the closest texel of the neighboring chart in this direction, and assign that one as a neighbour, Fig. 6. The discretization of differential operators in boundary texels is then performed similar to periodic boundary conditions in the 2D case.

4.2. Gradient Descent

The descent direction for the gradient descent is given by the Euler-Lagrange equation (8). Because of the complexity of the data term, its appropriate implementation is crucial for the methods overall efficiency.

Data Term. The visibility indicator v_i and the mapping ϕ_i which maps texture to image space can easily be pre-computed for each texel. In order to pre-compute \mathcal{J}_i for each camera, we need to obtain the backprojection mapping β_i . An efficient way to do this is by rendering the surface into each camera via OpenGL and then reading the Z-buffer to get the exact location for the backprojection of each pixel. \mathcal{J}_i is then obtained numerically via Eq. (6) using central differences.

The remaining terms to evaluate in each iteration are now the difference images

$$\mathcal{D}_i = \bar{b} * (b * (T \circ \beta_i) - \mathcal{I}_i). \quad (9)$$

Note that $T \circ \beta_i$ is simply a rendering of the surface in the i th image using the current texture map and a pinhole camera model, which can efficiently be performed with OpenGL. A high-resolution rendering is required, which is then convolved with b and subsampled to the resolution of the input images, following the imaging model of the camera. After that, the difference to \mathcal{I}_i is computed and the result convolved again with \bar{b} . All those operations can be implemented on the GPU for significant speedup.

Total Variation Term. Compared to the data term, the regularization term does not take much computation time to evaluate. However, care must be taken with the discretization of the operators. The scheme from [26] offers improved rotation invariance, with the diffusion tensor D set to the (isotropic) regularized TV flow,

$$D = \frac{\sqrt{\lambda}}{\max(\epsilon, \|\nabla \mathcal{T}\|)} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (10)$$

where $\epsilon > 0$ is a small regularization parameter. Note that adjusting the diffusion tensor allows to compare the TV flow to alternative regularizers like coherence-enhancing diffusion. In practice, however, when optimal parameters were used, only little difference can be observed in the resulting texture maps.

4.3. Extension to Color Textures

Probably the most elegant way to extend our model to color textures is to use the multidimensional total variation norm [5] in the energy functional. If $\mathbf{T} : \Sigma \rightarrow \mathbb{R}^m$ is a texture with m channels, then its TV norm is defined as

$$\text{TV}(\mathbf{T}) := \sqrt{\sum_{i=1}^m \text{TV}(T_i)}. \quad (11)$$

The Euler-Lagrange equations for this norm are [5]

$$\frac{\text{TV}(T_i)}{\text{TV}(\mathbf{T})} \cdot \frac{1}{\lambda} \text{div} \left(\sqrt{\lambda} \frac{\nabla \mathcal{T}}{\|\nabla \mathcal{T}\|} \right) = 0, \quad (12)$$

which must be satisfied for each texture component. The data term from (8) is simply added to the regularization term for each color channel separately in order to get the final Euler-Lagrange equations for our model. Note that the total variation of each channel can be computed in texture space as well by transforming the integral:

$$\begin{aligned} \text{TV}(T_i) &= \int_{\Sigma} \|\nabla_{\Sigma} T_i\|_{\Sigma} \, ds = \int_{\mathbb{T}} \|D\tau^{-1}(\nabla_{\Sigma} T_i \circ \tau)\| \lambda \, dx \\ &= \int_{\mathbb{T}} \sqrt{\lambda} \|\nabla T_i\| \, dx. \end{aligned} \quad (13)$$

A conformal map scales tangent vectors uniformly, so lengths scale with the square root of the area element, i.e. the conformal factor.

5. Experiments

After a description of the basic setup of all experiments, the merits of our approach are demonstrated on several synthetic datasets before turning to a real challenge: a multi-view dataset of a bird with approximate geometry estimated by a 3D reconstruction algorithm.

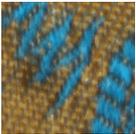
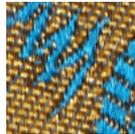
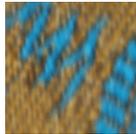
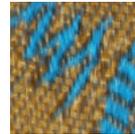
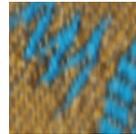
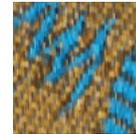
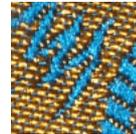
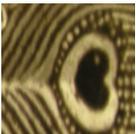
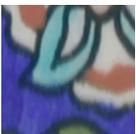
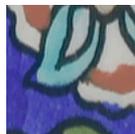
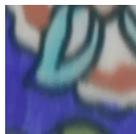
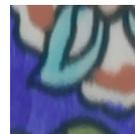
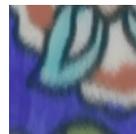
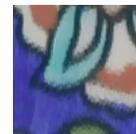
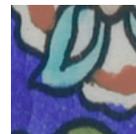
512 × 512		341 × 341		256 × 256		Ground truth
Initialization	Result	Initialization	Result	Initialization	Result	
						
6.35	4.56	8.01	6.07	8.08	6.23	
						
25.08	17.81	25.84	22.48	25.33	22.56	
						
5.08	3.33	7.45	5.63	7.62	5.66	
						
4.17	2.94	4.76	3.04	5.03	3.23	

Table 1. Superresolution results from full resolution input images are visually almost indistinguishable from ground truth. Furthermore, superresolution reconstructions from downsampled images are on average of better quality than the texel-wise initialization from four times the input image resolution. Mean squared error ϵ for a color range of $[0, 255]$ is shown below the image close-ups.

5.1. General Setup

A tough choice for our algorithm is the convolution kernel b modelling the cameras. Following the advice in [2], a Gauss kernel with standard deviation equal to half the pixel diameter leads to good simulation of the real-world situation.

The remaining free parameter σ influences the smoothness of the result. As we aim at correct textures, it should be as low as possible. However, since deconvolution problems are notoriously ill-posed, it is useful to start the minimization process with a somewhat larger σ and larger time-steps to speed up convergence and get a good approximation quickly [27]. Thus, we perform 100 iterations with $\sigma = 1.0$, followed by another 100 iterations with $\sigma = 0.1$. The time step is fixed and set to 0.02. Throughout all experiments the same parameters were used. At the given resolutions, the algorithm takes about 3 hours until convergence, running on a 2.8 GHz Core 2 Duo processor with CUDA enhancements running on a GeForce GTX. Main memory required is around 4 GByte.

5.2. Synthetic Data

In order to evaluate the algorithm numerically and to make meaningful comparisons of texture estimation errors, we first run it on synthetic test scenes. For complete knowledge of all mappings and easy comparison with ground truth we chose a torus as a test object, with a single texture wrapped around it using the standard parametrization. Images of the torus were raytraced from 48 camera locations, distributed evenly around it on six different height levels. Image resolution is 512×512 pixels, while the original and reconstructed texture is of size 1024×1024 texels.

For initialization, each texel is set to the weighted average color of its projections in all views where it is visible. Weights are given by the inverse backprojection area element, Eq. (6). As can be seen in Table 1, this common pointwise approach to texture estimation leads to slightly blurry results in which many of the original details are lost. Naturally, the results will be even worse if the geometry is less exact than in this ideal example.

To investigate how the result depends on input image resolution, the superresolution algorithm is applied to downsampled as well as the original images. In each case, the mean squared error ϵ between reconstructed and original

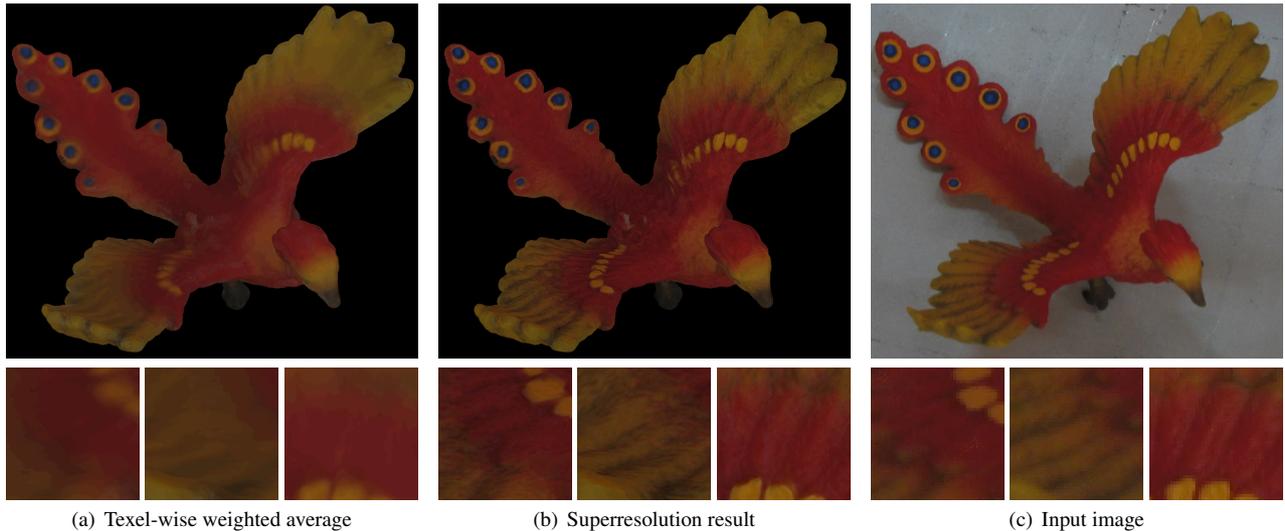


Figure 7. Results from a real-world multiview dataset. The 3D model is rendered with the texture from texel-wise initialization (a) and the texture resulting from the proposed superresolution algorithm (b). Our result has more visible details than an input image taken from the same viewpoint (c). The bottom row shows some close-ups, see also Fig. 2 for a higher zoom factor, and Fig. 3 for a different viewpoint and the complete texture atlas.

texture map was recorded. Error values and close-ups of the resulting texture maps are shown in Table 1.

If the input images are scaled down by a factor of 1.5, our method still produces textures which are as good as the texel-wise initialization in the high-resolution case. This clearly demonstrates the viability of the superresolution approach.

5.3. Real-world Images

Finally, we perform some experiments on real-world datasets. From 36 images of a bird at resolution 768×584 , Fig. 1, a 3D reconstruction was obtained using an implementation of the algorithm in [14]. After creating a conformal atlas, a texture map was computed using the proposed superresolution algorithm, with the same texel-wise initialization as for the synthetic data sets.

The data is challenging for a texture reconstruction method, since the geometry is imperfect, and weighted averaging of color information over cameras leads to a blurry initialization. Notably, however, the optimized texture estimated via superresolution exhibits even more details than an individual original input image, Fig. 7. This is something no previous method for texture generation could hope for. The only visible artifacts result from clearly incorrect 3D geometry, while no seams due to visibility boundaries can be observed.

6. Conclusion

We proposed the first approach to superresolution texture reconstruction from multi-view images which solves a

superresolution model on curved surfaces. The image formation model leads to an energy with the desired texture map as the minimizer, whose Euler-Lagrange equation is a PDE on the surface. Using a conformal atlas for the known surface geometry, the equation can be solved via total variation deblurring on planar 2D texture space. The method produces high-fidelity texture maps, which approach the point-wise reconstruction quality from much higher resolution input. The visible texture detail can surpass the detail of individual input images. A current limitation is that no additional image registration is performed, so geometry and camera calibration needs to be accurate. In a followup work [11], we therefore focus on a generalized model for joint texture and geometry reconstruction.

References

- [1] C. Allne, J.-P. Pons, and R. Keriven. Seamless image-based texture atlases using multi-band blending. In *19th International Conference on Pattern Recognition*, 2008.
- [2] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *PAMI*, 24(9):1167–1183, 2002.
- [3] F. Bernardini, I. Martin, and H. Rushmeier. High-quality texture reconstruction from multiple scans. *IEEE Transactions on Visualization and Computer Graphics*, 7(4):318–332, 2001.
- [4] M. Bertalmio, G. Sapiro, L. T. Cheng, and S. Osher. Variational problems and PDEs on implicit surfaces. In *Proceedings of the IEEE Workshop on Variational and Level Set Methods (VLSM'01)*, pages 186–193, 2001.
- [5] P. Blomgren and T. F. Chan. Color TV: Total variation methods for restoration of vector-valued images. *IEEE Trans. Image Processing*, 7:304–309, 1998.

- [6] D. Capel and A. Zisserman. Super-resolution from multiple views using learnt image models. In *Proc. CVPR*, volume 2, pages 627–634, 2001.
- [7] P. Cheeseman, B. Kanefsky, R. Kraft, and J. Stutz. Super-resolved surface reconstruction from multiple images. Technical Report FIA9412, NASA Ames Research Center, 1994.
- [8] M. Eisemann, B. D. Decker, M. Magnor, P. Bekaert, E. de Aguiar, N. Ahmed, C. Theobalt, and A. Sellent. Floating Textures. *Computer Graphics Forum (Proc. Eurographics EG'08)*, 27(2):409–418, 2008.
- [9] M. S. Floater and K. Hormann. Surface parameterization: a tutorial and survey. In *Advances in Multiresolution for Geometric Modelling*, Mathematics and Visualization, pages 157–168. Springer, 2006.
- [10] R. Fransens, C. Strecha, and L. van Gool. Optical flow based super-resolution: A probabilistic approach. *Computer Vision and Image Understanding*, 106(1):106–115, 2007.
- [11] B. Goldluecke and D. Cremers. A superresolution framework for high-accuracy multiview reconstruction. In *Pattern Recognition (Proc. DAGM)*, 2009. *accepted*.
- [12] X. Gu and S.-T. Yau. Global conformal surface parameterization. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, volume 43, pages 127–137, 2003.
- [13] R. Koch, M. Pollefeys, and L. V. Gool. Multi viewpoint stereo from uncalibrated video sequences. In *Proc. ECCV*, volume 1406 of *Lecture Notes In Computer Science*, pages 55–63, 1998.
- [14] K. Kolev and D. Cremers. Integration of multiview stereo and silhouettes via convex functionals on convex domains. In *Proc. ECCV*, volume 5302 of *Lecture Notes In Computer Science*, pages 752–765, 2008.
- [15] V. Lempitsky and D. Ivanov. Seamless mosaicing of image-based texture maps. In *Proc. CVPR*, volume 1, pages 1–6, 2007.
- [16] H. Lensch, W. Heidrich, and H. P. Seidel. A silhouette-based algorithm for texture registration and stitching. *Graphical Models*, 63(4):245–262, 2001.
- [17] L. M. Lui, Y. Wang, and T. F. Chan. Solving PDEs on manifold using global conformal parameterization. In *Proc. VLSSM*, pages 309–319, 2005.
- [18] B. Lvy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on graphics (SIGGRAPH)*, 21(3):362–371, 2003.
- [19] K. Nakamura, H. Saito, and S. Ozawa. Generation of 3d model with super resolved texture from image sequence. In *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, pages 1406–1411, 2000.
- [20] T. Schoenemann and D. Cremers. High resolution motion layer decomposition using dual-space graph cuts. In *Proc. CVPR*, pages 1–7, 2008.
- [21] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. CVPR*, pages 519–528, 2006.
- [22] F. Sroubek, G. Cristobal, and J. Flusser. A unified approach to superresolution and multichannel blind deconvolu-

tion. *IEEE Transactions on Image Processing*, 16(9):2322–2332, 2007.

- [23] J. Stam. Flows on surfaces of arbitrary topology. *ACM Transactions on Graphics (SIGGRAPH)*, 22(3):724–731, 2003.
- [24] C. Theobalt, N. Ahmed, H. Lensch, M. Magnor, and H. P. Seidel. Seeing people in different light-joint shape, motion, and reflectance capture. *IEEE Transactions on Visualization and Computer Graphics*, 13(4):663–674, 2007.
- [25] Y. Wang, X. Gu, K. Hayashi, T. F. Chan, P. Thompson, and S.-T. Yau. Surface parameterization using Riemann surface structure. In *Proceedings of ICCV*, volume 2, pages 1061–1066, 2005.
- [26] J. Weickert and H. Schar. A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance. *Journal of Visual Communication and Image Representation*, 13(1–2):103–118, 2002.
- [27] M. Welk, D. Theis, T. Brox, and J. Weickert. PDE-based deconvolution with forward-backward diffusivities and diffusion tensors. volume 3459 of *Lecture Notes in Computer Science*, pages 585–597. Springer, 2005.

Appendix: Proof of Theorem

This section deals with the derivation of the Euler-Lagrange equation for the energy functional. For this, let δT be a variation of the texture. In [17], the directional derivative of the regularization term in direction δT was found to be

$$\left. \frac{d}{dT} E_{\text{tv}}(T) \right|_{\delta T} = \int_{\Sigma} \operatorname{div}_{\Sigma} \left(\frac{\nabla_{\Sigma} T}{\|\nabla_{\Sigma} T\|_{\Sigma}} \right) \cdot \delta T \, ds. \quad (14)$$

A precise definition of differential operators appearing below the integral is beyond the scope of this work, the reader is referred to the above reference.

The data term can first be treated in image space with respect to the projected variation $\delta T \circ \beta_i$. Following the arguments of the 2D case [27], one obtains

$$\left. \frac{d}{dT} E_{\text{data}}(T) \right|_{\delta T} = \sum_{i=1}^n \int_{\hat{\mathcal{S}}_i} \mathcal{D}_i \cdot (\delta T \circ \beta_i) \, dx, \quad (15)$$

For the derivation of the final Euler-Lagrange equation, however, it is a problem that the integration in the data term takes place over different domains. Fortunately, by means of the visibility indicator functions $v_i : \Sigma \rightarrow \{0, 1\}$ from (5) and backprojection area elements \mathcal{J}_i from (6), the expressions in (15) can be transformed to surface integrals:

$$\left. \frac{d}{dT} E_{\text{data}}(T) \right|_{\delta T} = \sum_{i=1}^n \int_{\Sigma} v_i ((\mathcal{J}_i \mathcal{D}_i) \circ \pi_i) \cdot \delta T \, ds, \quad (16)$$

At a local minimum, the directional derivative vanishes for every displacement δT . Putting (14) and (16) together yields the desired Euler-Lagrange equation. \square