A New Variational Framework for Multiview Surface Reconstruction

Ben Semerjian

Urban Robotics, Inc., Portland, Oregon, USA

Abstract. The creation of surfaces from overlapping images taken from different vantages is a hard and important problem in computer vision. Recent developments fall primarily into two categories: the use of dense matching to produce point clouds from which surfaces are built, and the construction of surfaces from images directly. This paper presents a new method for surface reconstruction falling in the second category. First, a strongly motivated variational framework is built from the ground up based on a limiting case of photo-consistency. The framework includes a powerful new edge preserving smoothness term and exploits the input images exhaustively, directly yielding high quality surfaces instead of dealing with issues (such as noise or misalignment) after the fact. Numeric solution is accomplished with a combination of Gauss-Newton descent and the finite element method, yielding deep convergence in few iterates. The method is fast, robust, very insensitive to view/scene configurations, and produces state-of-the-art results in the Middlebury evaluation.

Keywords: Surface reconstruction, surface fairing, multiview stereo, Gauss-Newton, finite element method.

1 Introduction

One of the grandest problems in structure from motion concerns the creation of surfaces from images given known view extrinsics and intrinsics. This problem is important because it yields a dense and useful geometric representation of that which was photographed. The problem's complexity stems from many reasons: nonlinear relation between surface and pixel, discontinuities and folds in the scene, image noise, ambiguity in textureless regions, scaling and implementation difficulties, illumination changes, and scene changes, to name a few.

There has been much work done on the topic, boosted in part by advancements in computing power. One way to approach surface reconstruction is to first perform dense matching on pairs of images (*e.g.* [12][18]), then create a point cloud from the matches via triangulation (*e.g.* [15][31]), and finally create a surface from the point cloud (*e.g.* [19][29]). This methodology is popular for several reasons, including the availability of very fast and accurate dense matching algorithms, the fact that a point cloud is sometimes desired instead of a surface, processing speed, and relatively simple implementation due to the clear separation between steps. There are disadvantages too, most significant is

D. Fleet et al. (Eds.): ECCV 2014, Part VI, LNCS 8694, pp. 719–734, 2014.

that the output surfaces might lack accuracy and have excess noise; this is partly because these methods are based on pixel matching instead of surface generation (for example, planar correspondences do not imply planar surfaces).

The main other class of methods basically create surfaces directly from the images, a technique often called "multiview stereo", examples of which include [17][24]. Since these focus on building surfaces instead of matching pixels, they have the potential for higher quality output. Furthermore, they inherently handle multiview relations, which enables higher accuracy.

Fig. 1. Example of the proposed surface reconstruction in action: one of three handheld images (left) and surface output (right) rendered with Oren-Nayar shading [27]

The primary contribution of this work is a new method for surface reconstruction belonging to the second class described above, a sample application of which is shown in Figure 1. New ideas are combined with established concepts from multiview stereo, optical flow, and surface fairing. Perhaps unusual for a computer vision topic, numeric solution uses the finite element method with inspiration from continuum mechanics. The resulting surfaces are computed quickly and in an arbitrarily scalable manner, and since the formulation is continuous the range of depth (or disparity) has no effect on computation speed or memory. The resulting surfaces not only are accurate in an absolute sense, they also have smooth, accurate normals. A method of selecting high quality surfaces is also presented (enabled by the high accuracy of normals), yielding a means to avoid surface fusion and limits on the scale of output.

1.1 Related Work

The use of variational formulations to approach image matching problems (*e.g.* surface reconstruction, dense correspondence, or optical flow) is nothing new, and some of the strongest works on these topics go that route. In [5], a problem is built using the combination of data and smoothness to provide high accuracy optical flow. There, the data term penalizes for differences both in image intensity and in image gradient. This is extended in [4] to the case of "large displacement", extending the same data term with bias toward sparse features. Though indeed accurate and valid for large displacement, this will not work in the case of significant affine changes (such as rotation) since such will transform the image gradient, preventing it from being matched.

To deal with affine changes, [21] and [17] exploit surface normals to define local coordinate transforms, and use cross correlation oriented with those local transforms (potentially with normalization [23]) for the data term. This makes for not only a more flexible problem, but also a stronger one because the data term introduces a coupling over the surface due to its dependence on normal, forcing a higher level of consistency in the output (otherwise, the role of coupling rests entirely with the smoothness term). An issue with this approach is the need for selection of correlation window size: too small results in hampered robustness, too large smears things together. Despite many gains, the pointwise nature of [4] lending to simplicity and high accuracy is lost.

Regarding smoothing (or regularization), which is necessary to deal with image noise and textureless regions, [17] adds bending energy in the style of [20] to the minimization. This is fast, simple, and smooths effectively without biasing the solution toward minimum surface area, as a mean curvature approach would do. There are disadvantages though, as pointed out in [6][25], including poor numerical qualities and mesh-dependent behavior. They suggest principal curvature based smoothing with an elaborate curvature calculation, which works better and also does not induce surface area bias.

One issue common in these and other works is the fact that they rely on the computation of curvatures (or other second order quantities) on triangular meshes, which not only is fragile (individual mesh faces have no curvature) but as shown in [35] is guaranteed to suffer from at least one pathology no matter how elaborate. To make matters worse, these complicated quantities are typically minimized with gradient descent (*e.g.* [17][16][9]), which is sensible for simplicity but gives only linear convergence.

In this work, a notion of "infinitesimal patch" is introduced, giving pointwise illumination invariant error measurements as in [5] with the affine invariance and coupling of [17]. A scale invariant curvature-like smoothness term is used, whose magnitude is minimized (instead of its square) for edge and discontinuity preservation in the way second order total generalized variation [2] works. Discretization is accomplished using a second order finite element method [3], which represents the solution as a continuously differentiable function, implying continuous surface normals. The Gauss-Newton method [36] is used for numeric minimization, giving near second order convergence in few iterates.

2 Design of the Framework

Consider N overlapping images, whose greyscale content is notated as $B_i(\mathbf{u}_i)$ where $\mathbf{u}_i = (u_i, v_i)$ is a pixel coordinate on image i. Each image point $\mathbf{u}_i$ is the projection [14] of a 3D point $\mathbf{X}$ according to

$$w_i \mathbf{u}_i = K_i R_i (\mathbf{X} - \mathbf{c}_i) \tag{1}$$

where K_i, R_i, and $\mathbf{c}_i$ are respectively the intrinsics, rotation, and position of view i, and w_i is a quantity known as *depth*.

In this work, surface reconstruction will be posed as the problem of finding depth on one of these images, arbitrarily the first, as a function of pixel coordinate. That is, the goal is to find $w_0(\mathbf{u}_0)$. Such effectively defines 3D points as a function of image point as well, using the projection formula above.

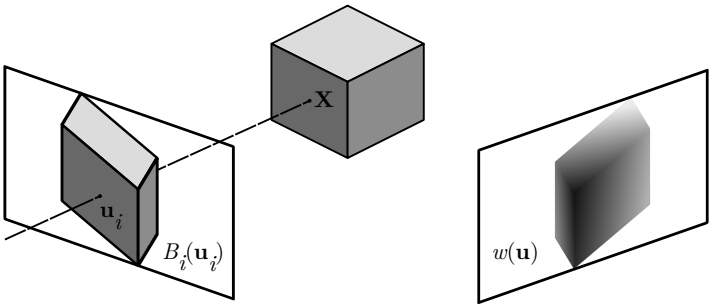

Fig. 2. A cube imaged by two views. Left frame: projected as a grey image onto view i, right frame: projected as depth image onto view 0.

This of course implies that only that which is viewable by the first view can be reconstructed; however one can create a depth image for every image available and reconstruct an arbitrarily large scene that way. As will be shown later, the fact that disparate surfaces are created with this strategy is not problematic (though they can be fused if desired, *e.g.* [29]), and the amount of focused effort the algorithm can put into the creation of a single depth image has benefit.

2.1 The Minimization Problem

Since the depth of image 0 will be the individual focus here, to simplify notation subscripts will be dropped for quantities referring to view 0. In other words, $B = B_0$, $w = w_0$, and $\mathbf{u} = \mathbf{u}_0$. The problem of finding the depth function $w(\mathbf{u})$ will then be posed as the minimization of the functional

$$\sum_{i>j\geq 0}^{N-1} \iint_{O_i \cap O_j} d\big(B_i, B_j, \mathbf{u}_i(w(\mathbf{u})), \mathbf{u}_j(w(\mathbf{u}))\big) + \alpha |\nabla B(\mathbf{u})| S\big(w(\mathbf{u})\big)\, \mathrm{d}u\mathrm{d}v \tag{2}$$

where d is a photo-consistency measure between images i and j, S is a smoothness function, α is a smoothness factor, and O_i is the subset of all points on image 0 which are viewed by image i (that is, the *overlap* between 0 and i).

Note that the summation, which appears outside of the integral, covers every combination of i and j once. This sets up interactions between every possible combination of views, and these interactions occur over the intersections of the overlap domains O_i and O_j. The smoothness term is also under the summation though it does not involve i or j, this is done so that the number of smoothness contributions equals the number of data combinations.

2.2 Photo-Consistency

Photo-consistency is the data of the minimization; it measures correspondence fitness based on image content. The function $d(B_i, B_j, \mathbf{u}_i, \mathbf{u}_j)$ therefore penalizes for mismatch between the given images at the given coordinates. Note that in (2) there is one term for every possible combination of overlapping images.

Two views. To ease the derivation of this function, two views (0 and i) will be considered first without any notion of 3D. A correspondence from image 0 to image i may then be represented generally by a piecewise smooth function $\mathbf{u}_i(\mathbf{u})$. Put into words, this functional representation takes as input a coordinate on image 0 and gives a coordinate on image i.

To build the photo-consistency function, consider a 3×3 patch of pixels centered around an arbitrary point $\mathbf{u}$ on image 0. The correspondence function $\mathbf{u}_i(\mathbf{u})$ defines not only the location of the corresponding patch on i, but also the local coordinate system which describes the shape of the patch. This is illustrated in Figure 3.

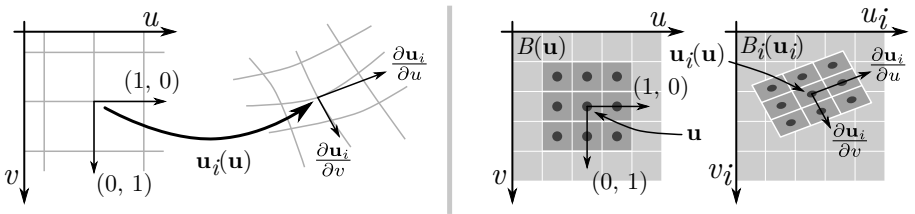

Fig. 3. Left: relation between coordinates on 0 and i at some arbitrary point, right: a patch on image 0 and a corresponding contorted patch on image i

Elaborating on "local coordinate system", the Jacobian of $\mathbf{u}_i(\mathbf{u})$ is:

$$J_i(\mathbf{u}) = \left[\frac{\partial \mathbf{u}_i}{\partial u} \;\; \frac{\partial \mathbf{u}_i}{\partial v}\right] = \begin{bmatrix} \frac{\partial u_i}{\partial u} & \frac{\partial u_i}{\partial v} \\ \frac{\partial v_i}{\partial u} & \frac{\partial v_i}{\partial v} \end{bmatrix} \tag{3}$$

where the columns may be seen as basis vectors on image i corresponding to the standard basis on image 0. These vectors define the shape of the patch.

Based on all this, a data term may be easily written as the magnitude of the (nine element) difference between the patches, with averages subtracted for brightness invariance. We can do much better though: by scaling the patch sizes by some factor Δs, dividing the patch differences by Δs, and taking the limit $\Delta s \to 0$, the data term simplifies to:

$$d(B_i, B_0, \mathbf{u}_i, \mathbf{u}_0) = |J_i^{\mathsf{T}} \nabla_i B_i - \nabla B| \quad ; \quad \nabla_i = \begin{pmatrix} \frac{\partial}{\partial u_i} \\ \frac{\partial}{\partial v_i} \end{pmatrix} \tag{4}$$

where the gradient operators are applied on the images in their own individual coordinates as shown. Note that the magnitude of the residual is used instead of the square; this makes numeric solution a little more complicated but ultimately yields better results.

This data term is in essence a difference in image gradients, but with the gradient of B_i contorted into the coordinates of view 0 using the Jacobian. In fact, the above could be derived more readily by writing the differences in gradients of 0 and i in the coordinates of 0, and then using the chain rule to change derivatives. The above derivation is interesting though, as it reveals that this sort of gradient matching is like patch matching, but with "infinitesimal patches". This strongly suggests that, under this formulation at least, another term penalizing for differences in raw color as in [4] is unnecessary.

This result gives the promised qualities: simple pointwise nature as in [4], illumination invariance, local affine invariance, and "built in" coupling between pixels due to the use of the Jacobian. One way to visualize the benefit of that last item is that a single mismatched point will unfavorably affect its neighbors due to distruption of the Jacobian.

Note that while this measure is pointwise on paper, in practice finite differences are used to differentiate images and a 3×3 sampling of pixels is still necessary. The limiting case derived above remains advantageous for several reasons, including the fixed size of finite differences (as opposed to chosen size of patch) and the lower number of residuals: two instead of (at least) nine.

Surface Parameterization and Multiple Views. Extension to multiple views is straightforward, accomplished by contorting the second term:

$$d(B_i, B_j, \mathbf{u}_i, \mathbf{u}_j) = |J_i^{\mathsf{T}} \nabla_i B_i - J_j^{\mathsf{T}} \nabla_j B_j| \tag{5}$$

where $J_0 = I$ is implied.

In order for this to make sense for surface reconstruction, the fact that the N abstract correspondence functions $\mathbf{u}_i(\mathbf{u})$ can be replaced with functions dependent on depth w instead is used. Writing projection equations (1) for views 0 and i separately and eliminating the 3D point yields this parameterization:

$$\mathbf{u}_i(w(\mathbf{u})) = \frac{1}{wr_i + t_{z,i}} \begin{pmatrix} wp_i + t_{x,i} \\ wq_i + t_{y,i} \end{pmatrix} \tag{6}$$

where the quantity in the denominator is the depth on view i, and the following definitions are made for compactness:

$$M_i = K_i R_i R_0^T K_0^{-1} \ , \quad \mathbf{t}_i = K_i R_i (\mathbf{c}_0 - \mathbf{c}_i) \ , \quad \begin{pmatrix} p_i \\ q_i \\ r_i \end{pmatrix} = M_i \mathbf{u} \ . \tag{7}$$

To emphasize, (5) was derived using pixel correspondences but for surface reconstruction is completely parameterized by the surface as represented by the depth function $w(\mathbf{u})$ via the relation (6). This implies that the Jacobian J_i must be written in terms of depth as well, which is possible by differentiating (6), involving the gradient of the depth function. This is in contrast with other approaches (*e.g.* [17]) that rely on the surface normal, a more complicated quantity, and one that is more difficult to involve in an optimization.

One potential weakness here is that since all of the gradients are in essence projected onto view 0, there will be some form of asymmetry in the photo-consistency measure. One possible means of alleviating this, still without introducing surface normal, is to project onto i and j separately:

$$d_{\mathrm{Sym}}(B_i, B_j, \mathbf{u}_i, \mathbf{u}_j) = |\nabla_i B_i - J_i^{-\mathsf{T}} J_j^{\mathsf{T}} \nabla_j B_j| + |J_j^{-\mathsf{T}} J_i^{\mathsf{T}} \nabla_i B_i - \nabla_j B_j| \ . \tag{8}$$

In practice this does not alter the surfaces significantly while adding significant complexity, it is therefore not considered.

2.3 Smoothness Function

As is well established in the study of differential geometry, quantities derived from curvature are high performing (though complicated) measures of surface quality, fairness, and noise [25][22].

It would seem natural then to add a curvature-derived quantity to the minimization here in order to keep the output of surface reconstruction fair and high-quality. Unfortunately, smoothness penalties derived from raw curvature are unsuitable because they will involve the scale of the 3D output. This is highly undesirable because scale is ambiguous in structure from motion problems [33].

To remedy this, [17] multiplies curvature with depth (canceling scale), however such is ad-hoc and not certain to work universally. In this work, the fact that curvature can be measured from spatial changes in surface normal is exploited.

For example, one of many definitions for the mean curvature of a surface is the divergence (with respect to 3D space) of the normal [13]. Following this, the smoothness term used here is based on the first derivatives of the normal, not against 3D space, but against image coordinates:

$$S(\mathbf{u}) = |\nabla \hat{\mathbf{n}}(\mathbf{u})| \ . \tag{9}$$

To emphasize, the unit normal is a dimensionless function of shape, and since differentiation is against pixel coordinate this quantity is free from physical scale. This fairness measure may be thought of as the curvature of the surface as *seen* by view 0.

Note that in (2) the magnitude of the gradient of B appears as a factor on S. This results in a texture/contrast invariant balance between data and smoothness; one might imagine that omitting such for more smoothness in areas with less contrast would make sense, however experiment shows that full contrast invariance works better. Furthermore, if the smoothness factor α is understood to have pixel coordinate units, the data and smoothness terms in (2) will both have the same units. This suggests a well-formed problem which will have very consistent behavior at multiple scales (the meaning of scale explained in the section on numeric solution), which is demonstrated in Figure 4.

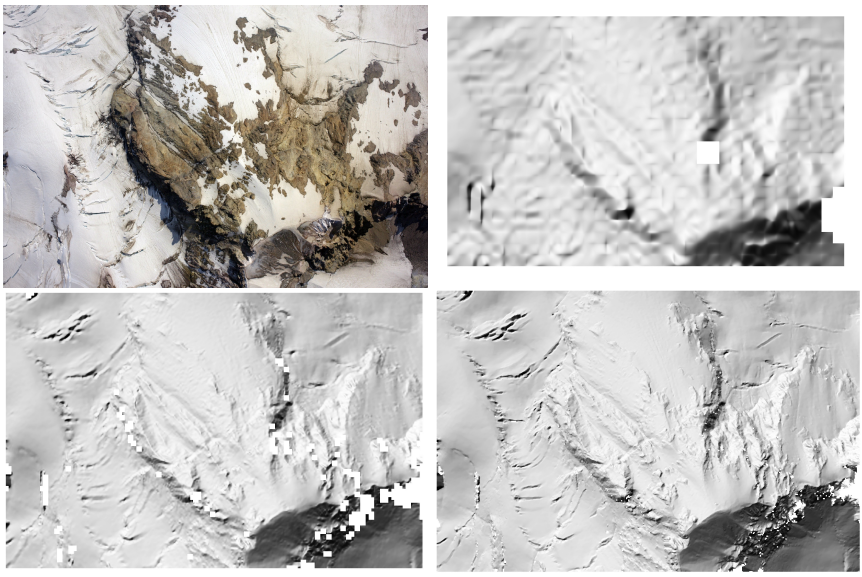

Fig. 4. From top left: a 2400×1596 aerial image of the peak and northeastern upper reaches of Mt. Hood taken with a nadir-looking wide angle lens; depth images at scales 64, 16, and 4, rendered with Oren-Nayar shading [27]. Sixteen other images were used to reconstruct these surfaces with $\alpha = 0.6$. Note that the coarse scale captures the overall shape of the terrain, while the fine scale sharply reveals every crevasse without sacrificing smoothness. These images feature a variety of reconstruction difficulties: high/low texture transitions, shadow noise, highly oblique surfaces, and sharp edges.

3 Numeric Solution

With the raw variational problem fully defined, numeric solution now will be described. It consists of three basic ingredients: a discretization reducing $w(\mathbf{u})$ to an interpolation on a regular 2D grid, a means of minimizing (2) at fixed scale and domain, and a coarse to fine domain management scheme.

3.1 Discretization

The finite element method is used to discretize the problem. This method is extremely popular in the study of continuum mechanics and other fields [28], but unfortunately has made few appearances in computer vision.

To briefly summarize, the method as applied here defines the unknown depth function $w(\mathbf{u})$ as a set of bicubic patches on a square grid of spacing σ. The grid intersection points are called *nodes*, and each node carries the value, gradient, and mixed second derivative of depth at that point, giving rise to a solution surface that is continuously differentiable by definition (*i.e.* without differencing).

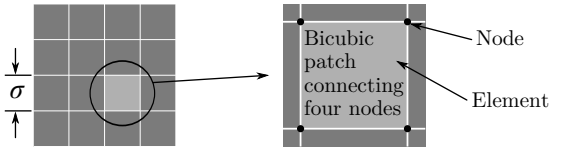

Fig. 5. Illustration of the finite element method discretization used

This representation is substituted into (2) and the integrals are evaluated with the midpoint method; all of the nodal values therefore form the discrete unknowns of the problem. Advantages of this approach include a clear and natural discretization of (2); more importantly, the smoothness of the solution implies continuous Jacobians in (5) and continuous normals in (9). A simpler discretization would require finite differences for these quantities, leading to messy implementations and much less effective smoothing.

3.2 Minimization

At some fixed spacing σ and set of domains O_i, the minimization of (2) is carried out using the Gauss-Newton method. That is, steps are taken by solving for a Newton step $\mathbf{p}$ to be applied to all nodes from the linear system

$$H\mathbf{p} = -\mathbf{g} \quad ; \quad H \approx J^{\mathsf{T}} J \ , \quad \mathbf{g} = J^{\mathsf{T}}\mathbf{r} \tag{10}$$

using a Hessian matrix H approximated as shown. Often, this is done by storing the Jacobian J and residuals $\mathbf{r}$ for the whole problem (this is different from the Jacobians of (5)), however in this case the residuals are too many and it is more practical to directly store the gradient $\mathbf{g}$ and approximate Hessian H.

The solution of one step is carried out using the conjugate gradient method [36], explicitly preconditioned with a Cholesky factorization [36] of the 4×4 diagonal blocks of H. This normally is not regarded as a very powerful preconditioner, however it is adequate for this problem because H tends to be relatively well-conditioned, because the blocks are moderate-sized, and also because the solution happens over multiple scales as in [8].

Unregulated steps work quite well (perhaps surprisingly), and step control (*e.g.* trust region, line search) is unnecessary. The minimization is allowed to run till the steps change the reprojections in amounts significantly smaller than one pixel.

The fact that both data (5) and smoothness (9) involve the magnitudes of vectors (as opposed to squares) makes the optimization potentially difficult; iterative re-weighting is used here to keep it simple. That is, squares of d and S are used for the computation of H and g, weighted by the reciprocals of their magnitudes. This is relatively simple to implement, and is surprisingly effective for data + smoothness type problems, for example the algorithm in [7] yields excellent results in just one iteration.

3.3 Coarse to Fine Domain Management

The minimization problem (2) is incomplete in that the overlap domains O_i are unknown. Since these are not differentiable objects and appear only fixed in (2), a set of heuristics are used to manage them outside of individual minimizations. This is done over multiple scales, coarse to fine, gradually refining both the solution $w(\mathbf{u})$ and the domains.

Initially, the spacing σ is set to a power of two larger than typical sparse point spacing but smaller than image dimensions, *e.g.* $\sigma = 128$. The input images are Gaussian blurred with standard deviation $0.12\sigma+0.2$. The initial domain is fitted to sparse points at this spacing.

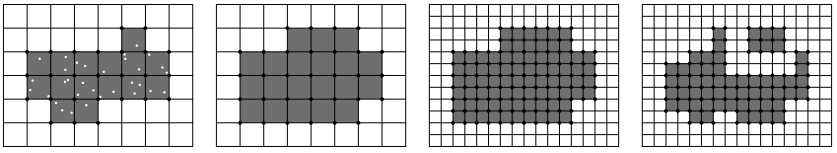

Fig. 6. Various domain operations. Left to right: initial domain with sparse points, expanded domain, domain at halved spacing, cleaned domain with topology change.

The solution process proceeds as follows: with scale and domains fixed, minimize (2) as described in the previous section, expand the domains by extrapolating, clean them to remove occlusions and non-converged points, and repeat a number of times, leading to convergence of both domain and depth. The spacing is then halved and the process repeated. Completion happens when (2) is minimized at $\sigma = 2$, or at some a coarser terminal spacing if lower resolution is considered adequate. Going below $\sigma = 2$ will add no benefit, since that is the point where the total number of degrees of freedom will equal the number of pixels. Behavior with decreasing scale is shown in Figure 4.

The expansion operation is an extrapolation of all of the boundary nodes using thin plates [1], the purpose of which is to enlarge the domains. The more complicated cleaning operation removes non-converged nodes (those tend to be

mismatches), elements covering adequately textured pixels with a normalized cross correlation [23] less than zero, spots on individual domains violating the visibility constraint [32], and elements that are suspected of lying on an occlusion boundary. The test for occlusion is similar to that in [11]: a small singular value of the Jacobian at a node is indicative.

4 Practical Usage

The methodology outlined here yields one surface for one image. Though potent, this is restrictive since one cannot expect that which is viewed by one image to cover all that is interesting in an arbitrary set of images. The obvious remedy is to separately output surfaces from different groups of overlapping images. Though this could mean redundant work, there is the significant benefit of scalability: an arbitrary number of surfaces can be processed, provided the number of overlapping images is controlled (in practice, approximately ten images per surface works well). Furthermore, the work is trivial to parallelize – individual processing units work on individual surfaces.

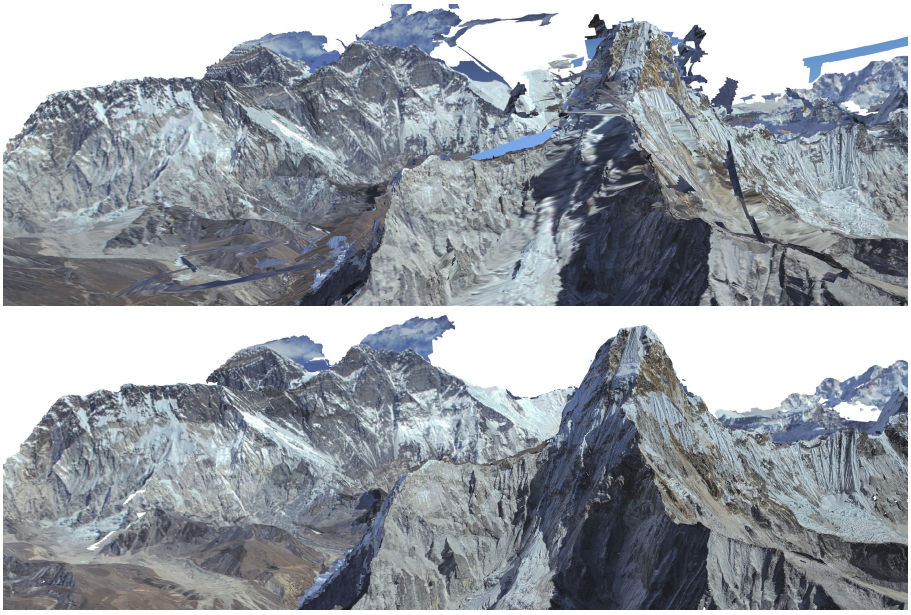

Fig. 7. Ama Dablam in the Himalayas; 639 depth images have been separately computed and outputted as meshes, taking less than an hour on 24 cores. Top: raw surfaces, bottom: after cutting (10 minute computation), with some permitted overlap. Helicopter photography by David Breashears, December 2010, www.glacierworks.com.

Such collections of surfaces can be fused into single surfaces [29][19]. Though this specific task is not arbitrarily scalable, it is very effective for the creation of mensurable, high quality surfaces.

For visual applications there is another option: the outputted surfaces can be "cut" so that surface points are kept only if best viewed by the view that generated them, and deleted otherwise. These cut surfaces can then be optimally meshed in two dimensions [10] (since they are individually represented as 2D functions $w(\mathbf{u})$), and then either "zippered" [34] or simply presented together without any fusion. An example of this is shown in Figure 7.

The criteria for surface cutting is based on the *surface resolving power* (derivation may be found in supplemental material):

$$\frac{\mathrm{d}Q}{\mathrm{d}A} = -\hat{\mathbf{n}} \cdot \left(\nabla_{\mathbf{X}} u \times \nabla_{\mathbf{X}} v\right) \tag{11}$$

where A is a physical area, Q is a projected pixel area, and $\mathbf{u}$ is the projection (1) of the 3D point $\mathbf{X}$ being evaluated. This may be interpreted as a "pixels per surface area" measure taking obliqueness into account, and the image which views a point best will score highest in surface resolving power. That point is deleted from all other images.

5 Results

Figure 8 shows the output of this work compared to ground truth and two other high performers in the Middlebury multiview stereo evaluation [30]. For this, a maximum of 15 views were used to output every individual depth image, and $\alpha = 0.2$. Generation of all 363 depth images took just under an hour on a 12 core machine; these were then fused together using Poisson reconstruction [19], taking an additional 10 minutes on one core.

In addition to being accurate, the result of this work clearly rivals others in terms of smoothness, sharpness, and visual quality; for example, the mouth of the stegosaurus is resolved, its scales are sharp, its toes are distinct, and there is

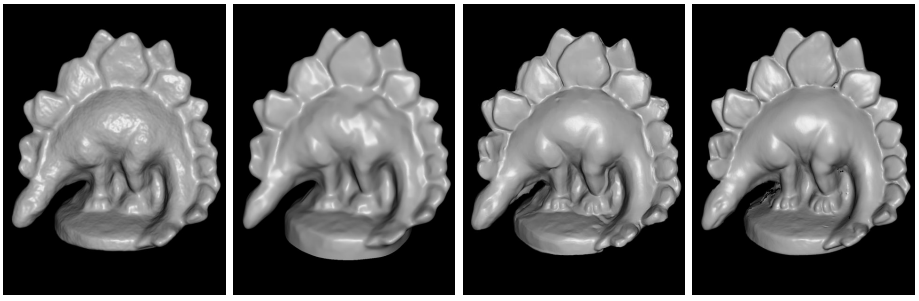

Fig. 8. Left to right: surface reconstructions of Furukawa2 [9], Shroers [29], this work, and ground truth for Middlebury's Dino dataset

Fig. 9. Left to right: one of five 2746×1832 images of a ceramic bull figurine taken with a telephoto lens, reconstruction from stereo correspondences using elas [12], reconstruction at $\alpha = 4$ and scale 4 using this work. Rendered with mean curvature colorization, best viewed in color.

nearly no noise to be found. Its specular characteristics (which reveal curvature and surface quality [26]) also match the ground truth most closely.

In Figure 9, comparison is made against elas, a high ranking stereo correspondence method [12]. Every surface point is triangulated from five densely matched points via reprojection error minimization [14]. While the surface is accurate in an absolute sense, the normals are almost random and curvature is not controlled; a surface such as this could not be cut using (11) due to inaccuracy of normal. It should be noted though that elas took less than 9 seconds, while the result of this work took 150 seconds.

Figure 7 demonstrates capability on a large physical scale. This shows what can be done with high accuracy surfaces foregoing fusion. The mountain in the center of the model is well resolved due to proximity with the camera; in contrast, the far content (up to 20 km away) serves as backdrop material, lacking fidelity without being noisy, much as it is in the source imagery.

Methods relying on fusion such as [19] or [29] would be unable to neatly mix far and near in a single reconstruction because these generally scale poorly with physical scale; methods such as [12] or [18] would also suffer because of the large disparities (a consequence of large depth ranges).

Figure 10 shows in detail and higher resolution one reconstruction from the image set in Figure 7. The seven images used were taken with a forward looking wide angle lens; all of the epipoles are in the image as shown in the figure. Though there is no stereo at an epipole and less than a pixel of parallax near Mt. Everest, the reconstruction is successful in making the most of what is available. Of particular note is the very consistent action of smoothing: the perceived curvatures are very even across all depth scales, and there is no increase in noise due to contrast changes, the epipoles, or distance from camera.

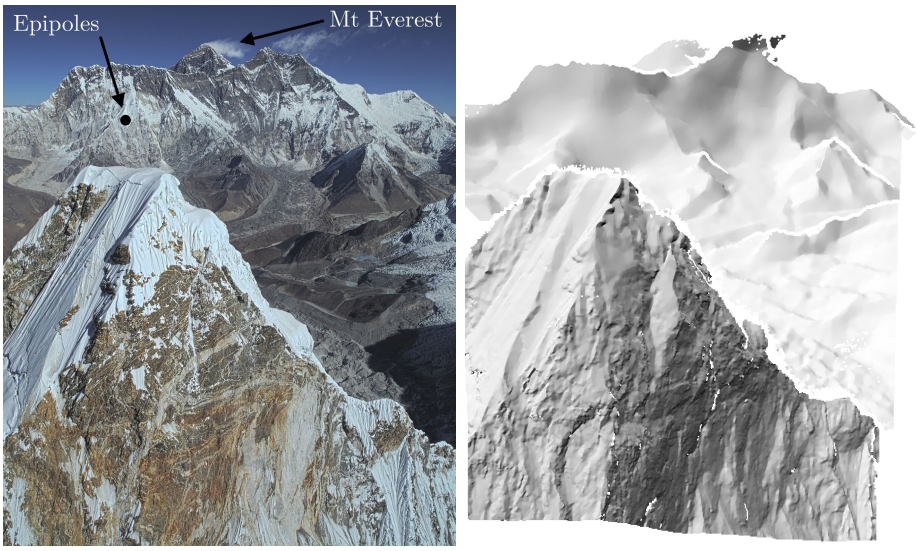

Fig. 10. Surface reconstruction of Ama Dablam (foreground peak) in the Himalayas using seven images and $\alpha = 0.3$, with Mt Everest 15 km away in the background. Helicopter photography by David Breashears, December 2010, www.glacierworks.com.

6 Conclusion

In this paper, a novel surface reconstruction method was built from scratch. With only a single tunable parameter controlling smoothness, it has been shown to output accurate, smooth, sharp, natural-looking surfaces. It is arbitrarily scalable and performs uniformly across many different kinds of images. A method for discarding surfaces that are better seen by other views was also given, reducing the need for fusion and allowing models of arbitrary size to be reconstructed.

Another notable aspect of this work is the use of the finite element method, which has unfortunately made few appearances in computer vision. Its application resulted in solutions with built-in differentiability, necessary for the smoothness term to work properly.

The highly effective smoothness function (9) could be used in other vision problems, such as shape from shading. Other possibilities for future work include the incorporation of sparse features as in [4], dense initialization with stereo correspondences, stronger occlusion handling, and refinement of view parameters alongside surfaces for very high accuracy output.

References

1. Bookstein, F.L.: Principal warps: Thin-plate splines and the decomposition of deformations. IEEE Transactions on Pattern Analysis and Machine Intelligence 11(6), 567–585 (1989)

2. Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. SIAM Journal on Imaging Sciences 3(3), 492–526 (2010)
3. Brenner, S.C., Scott, R.: The mathematical theory of finite element methods, vol. 15. Springer (2008)
4. Brox, T., Bregler, C., Malik, J.: Large displacement optical flow. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 41–48. IEEE (2009)
5. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
6. Desbrun, M., Meyer, M., Schröder, P., Barr, A.H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pp. 317–324. ACM Press/Addison-Wesley Publishing Co. (1999)
7. Farbman, Z., Fattal, R., Lischinski, D., Szeliski, R.: Edge-preserving decompositions for multi-scale tone and detail manipulation. In: ACM Transactions on Graphics (TOG), vol. 27, p. 67. ACM (2008)
8. Fattal, R., Lischinski, D., Werman, M.: Gradient domain high dynamic range compression. In: ACM Transactions on Graphics (TOG), vol. 21, pp. 249–256. ACM (2002)
9. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(8), 1362–1376 (2010)
10. Garland, M., Heckbert, P.S.: Fast polygonal approximation of terrains and height fields. School of Computer Science, Carnegie Mellon University (1995)
11. Gay-Bellile, V., Bartoli, A., Sayd, P.: Direct estimation of nonrigid registrations with image-based self-occlusion reasoning. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(1), 87–104 (2010)
12. Geiger, A., Roser, M., Urtasun, R.: Efficient large-scale stereo matching. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part I. LNCS, vol. 6492, pp. 25–38. Springer, Heidelberg (2011)
13. Goldman, R.: Curvature formulas for implicit curves and surfaces. Computer Aided Geometric Design 22(7), 632–658 (2005)
14. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press (2003)
15. Hartley, R.I., Sturm, P.: Triangulation. Computer vision and image understanding 68(2), 146–157 (1997)
16. Hernández, C., Vogiatzis, G., Cipolla, R.: Multiview photometric stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(3), 548–554 (2008)
17. Hiep, V.H., Keriven, R., Labatut, P., Pons, J.P.: Towards high-resolution large-scale multi-view stereo. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 1430–1437. IEEE (2009)
18. Hirschmuller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 2, pp. 807–814. IEEE (2005)
19. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the Fourth Eurographics Symposium on Geometry Processing (2006)
20. Kobbelt, L., Campagna, S., Vorsatz, J., Seidel, H.P.: Interactive multi-resolution modeling on arbitrary meshes. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, pp. 105–114. ACM (1998)

21. Kolev, K., Klodt, M., Brox, T., Cremers, D.: Continuous global optimization in multiview 3D reconstruction. International Journal of Computer Vision 84(1), 80–96 (2009)
22. Lee, C.H., Varshney, A., Jacobs, D.W.: Mesh saliency. ACM Transactions on Graphics (TOG) 24, 659–666 (2005)
23. Lewis, J.: Fast normalized cross-correlation. Vision Interface 10, 120–123 (1995)
24. Liu, Y., Cao, X., Dai, Q., Xu, W.: Continuous depth estimation for multi-view stereo. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 2121–2128. IEEE (2009)
25. Meyer, M., Desbrun, M., Schröder, P., Barr, A.H.: Discrete differential-geometry operators for triangulated 2-manifolds. In: Visualization and Mathematics III, pp. 35–57. Springer (2003)
26. Nordström, M., Järvstråt, N.: An appearance-based measure of surface defects. International Journal of Material Forming 2(2), 83–91 (2009)
27. Oren, M., Nayar, S.K.: Generalization of lambert's reflectance model. In: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, pp. 239–246. ACM (1994)
28. Reddy, J.N., Gartling, D.K.: The finite element method in heat transfer and fluid dynamics. CRC Press (2010)
29. Schroers, C., Zimmer, H., Valgaerts, L., Bruhn, A., Demetz, O., Weickert, J.: Anisotropic range image integration. In: Pinz, A., Pock, T., Bischof, H., Leberl, F. (eds.) DAGM/OAGM 2012. LNCS, vol. 7476, pp. 73–82. Springer, Heidelberg (2012)
30. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 519–528. IEEE (2006)
31. Stewenius, H., Schaffalitzky, F., Nister, D.: How hard is 3-view triangulation really? In: Tenth IEEE International Conference on Computer Vision, ICCV 2005, vol. 1, pp. 686–693. IEEE (2005)
32. Sun, J., Li, Y., Kang, S.B., Shum, H.Y.: Symmetric stereo matching for occlusion handling. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 2, pp. 399–406. IEEE (2005)
33. Szeliski, R., Kang, S.B.: Shape ambiguities in structure from motion. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(5), 506–512 (1997)
34. Turk, G., Levoy, M.: Zippered polygon meshes from range images. In: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, pp. 311–318. ACM (1994)
35. Wardetzky, M., Mathur, S., Kälberer, F., Grinspun, E.: Discrete laplace operators: no free lunch. In: Symposium on Geometry Processing, pp. 33–37 (2007)
36. Wright, S., Nocedal, J.: Numerical optimization, vol. 2. Springer, New York (1999)