On User-Interaction in 3D Reconstruction

David C. Schneider and Peter Eisert

Fraunhofer Heinrich Hertz Institute, Berlin, Germany†
Humboldt Universität zu Berlin, Berlin, Germany

Abstract

For many computer graphics tasks such as segmentation, tracking or retargeting, algorithms are designed to allow user interaction. For real-world applications, tools which give users some form of control are often preferred over fully automatic ones. In 3D reconstruction, however, algorithms are typically non-interactive and errors are corrected after reconstruction in generic 3D modeling software. In this contribution, we discuss fundamental algorithmic considerations regarding interactivity in 3D reconstruction and we outline an optimization-based reconstruction framework which allows the incorporation of interactive tools as energy terms.

1. Motivation

Many tasks in applied computer graphics are difficult to solve for a computer alone if the result is required to have production-grade quality. However, algorithms with some degree of artist control give rise to powerful tools which save hours of labour time and yield results suitable for professional media production. Some examples of successful semi-interactive approaches are (1) image segmentation and matting (e.g. Grabcut), where algorithms are initialized by "scribbles" and similar hints indicating background or foreground regions, (2) tracking, where users can interactively correct tracking paths [AV11] or (3) image and video retargeting [RGSS10], where users can define semantically important areas which must not be distorted by the algorithm. However, little work on semi-interactive tools has been published in the field of stereo and 3D reconstruction. This is surprising given the fact that many real-world scenes and objects are still challenging to reconstruct, especially if passive image-based methods are used rather than active scanners. The common pipeline for practitioners is to reconstruct the scene first and correct errors later using generic 3D modeling tools. We believe that interaction with the reconstruction algorithm itself is a more efficient approach for creating quality models—if an algorithm suitable for interactive correction and the appropriate tools are available. In this work, we discuss some fundamental algorithmic considerations regarding interactivity in stereo reconstruction, and we outline our ongoing research on interaction-friendly reconstruction algorithms and tools.

Many state-of-the-art stereo and multiview approaches are *local* in the sense that they reconstruct the 3D location and sometimes orientation of isolated image patches (e.g. [FP08] and many others). While this strategy is beneficial for parallelization, it requires a post-processing stage to generate a mesh: The reconstruction yields a point cloud with outliers which has to be filtered and meshed with appropriate algorithms such as Poisson meshing. Smoothness priors are often only considered at the meshing stage. Local reconstruction is difficult to combine with efficient interactive tools: As each patch is unaware of its neighbors, the correction of a single mismatched patch by the user will not affect its neighbors, although they are likely to be erroneous as well.

Therefore, we argue that a *global* reconstruction approach based on connected patches or meshes is beneficial for interactive tools. A connected reconstruction scheme also facilitates the incorporation—and interactive manipulation—of smoothness priors in the reconstruction process. Smoothness priors propagate information to neighboring patches, making user interaction regional and efficient.

2. Reconstruction framework

We use an image-based stereo setup with calibrated off-the-shelf SLR cameras which does not require projection of patterns into the scene. Narrow-baseline camera pairs are employed to compute high-quality depthmap-like meshes. For wider coverage, multiple pairs are used and depth-maps from several reconstructions are combined to obtain an over-

† This work was supported by the European FP7 project REACT (288369).

DOI: 10.2312/conf/EG2012/posters/013-014

Figure 1: *Reconstructions computed with our framework.*

all mesh. This allows us to keep the camera count low in contrast to volumetric approaches.

We formulate 3D reconstruction as a problem of estimating the parameters of a global warp function which maps one stereo image onto the other one. The stereo images are assumed to be single-channel and denoted as $\mathcal{I}$ and $\mathcal{J}$. The warp function $\mathcal{W}:(\mathbb{R}^2,\mathbb{R}^n)\rightarrow\mathbb{R}^2$ is parametrized with an n-dimensional vector $\mathbf{v}$. For a pixel at coordinates $\mathbf{x}$ the matching error is given by:

$$\mathcal{E}(\mathbf{x},\mathbf{v}) = \mathcal{I}(\mathbf{x}) - \mathcal{J}(\mathcal{W}(\mathbf{x},\mathbf{v})) \tag{1}$$

Estimating $\mathbf{v}$ amounts to computing

$$\arg\min_{\mathbf{v}} \sum_{\mathbf{x}} \phi(\mathcal{E}(\mathbf{x},\mathbf{v})) + \lambda_s \mathcal{S}(\mathbf{v}) \tag{2}$$

where ϕ is a robust norm-like function (e.g. Huber's), and $\mathcal{S}$ is a smoothness term weighted by λ_s. This energy can be minimized with a variant of the Gauss-Newton algorithm for norm-like functions. This requires the gradient of $\mathcal{E}$, which is $\nabla\mathcal{E}_{\mathbf{x}} = -\nabla\mathcal{J}^T\mathbf{J}_{\mathcal{W}}$. Here, $\mathbf{J}_{\mathcal{W}}$ denotes the Jacobian of the warp function. Note that if $\mathcal{W}$ is linear then $\mathbf{J}_{\mathcal{W}}$ is constant and only the image gradient $\nabla\mathcal{J}$ varies in each Gauss-Newton iteration. This estimation framework is generic until a warp function is specified. For 3D reconstruction, we need the warp $\mathcal{W}$ to describe or approximate the geometric relation between two stereo images. As speed is essential for an interactive system, $\mathcal{W}$ should be linear in $\mathbf{v}$ as discussed above. Also, $\mathcal{W}$ should be parametrized such that we can formulate an efficient smoothness term as well as influence terms for interaction.

The warp function we use describes the scene with a triangle mesh. By fixing the projection of the mesh in image $\mathcal{I}$ the warp can be parametrized with a depth value per vertex. However, the relationship between two projections of a 3D triangle is nonlinear. Therefore, we approximate the homographies which describe the relation between projected triangles by affinities. This yields a warp function which is closely related to the well-known mesh-based approach to 2D image warping, where an image deforms along with a control mesh in the image plane. To account for epipolar geometry, we constrain the vertex displacement of the control mesh, which deforms image $\mathcal{J}$, to the epipolar lines corresponding with the vertices of the (fixed) mesh in image $\mathcal{I}$. The parametrization $\mathbf{v}$ of our warp is a displacement of each vertex along an epipolar line. The use of a mesh makes it easy to formulate the smoothness term $\mathcal{S}(\mathbf{v})$: Denoting by $\mathbf{L}$ the mesh cotangent Laplacian, smoothness is controlled by $\mathcal{S}(\mathbf{v}) = \phi(\mathbf{L}\mathbf{v})$. Again, we use a robust norm-like function ϕ to allow discontinuities. Examples of reconstruction results are shown in figure 1.

3. Interactive tools

The framework sketched above facilitates the formulation of several kinds of user interaction with the algorithm. To initialize the nonlinear optimization, we use feature correspondences which are filtered for epipolar consistency. We then run Gauss-Newton until convergence before allowing interactive correction. The following tools are provided:

(1) The user can correct correspondence in areas where the algorithm has converged to a false match. The correction is added to equation (2) as energy term $\lambda_c \left\|\mathrm{diag}\left(\bar{\theta}_1 \ldots \bar{\theta}_n\right)\mathbf{v}\right\|^2$ where $\bar{\theta}_i$ is the target disparity for vertex i or zero of there is no correction for the vertex. Due to the smoothness term nearby vertices follow the corrected one, which allows the correction of a region with a single click. The energy formulation has the additional advantage that the correction does not have to be pixel-precise: Since the correction term is not a hard constraint the optimization is able to converge to a better state nearby.
(2) The user can influence the smoothness of a region with a smoothness brush. Smoothing is implemented by up- or down-scaling individual rows of the Laplacian $\mathbf{L}$ in the smoothness term. Thereby, vertices corresponding with scaled rows are penalized more (smoothing) or less (roughening) for deviating from their neighbors.
(3) Each off-diagonal element in the Laplacian corresponds with an edge of the mesh. By setting these elements to zero and re-normalizing the Laplacian row the user can break ties between vertices in regions where the smoothness term acts too strongly, e.g. at discontinuities.
(4) To create the control meshes we use a quality mesh generation algorithm which allows the inclusion of predefined edges as constraints. This allows various forms of interaction: For example, a user-annotated straight line can be included as a sequence of edges in the mesh. By adding an additional cost term for these edges, the line can be enforced the remain straight under the warp.

References

[AV11] AMBERG B., VETTER T.: GraphTrack: Faster than realtime tracking in videos. In *CVPR* (Colorado Springs, 2011). 1

[FP08] FURUKAWA Y., PONCE J.: Accurate, Dense, and Robust Multi-View Stereopsis. *TPAMI 1* (2008), 1–14. 1

[RGSS10] RUBINSTEIN M., GUTIERREZ D., SORKINE O., SHAMIR A.: A comparative study of image retargeting. *ACM Transactions on Graphics 29*, 5 (2010), 160:1–160:10. 1