**| Short Paper |**

# Terrain Geometry from Monocular Image Sequences

Alexander McKenzie
Caltech
am@caltech.edu

Eugene Vendrovsky
Rhythm + Hues Studios
eugene@rhythm.com

Junyong Noh
KAIST
junyongnoh@kaist.ac.kr

Terrain reconstruction from images is an ill-posed, yet commonly desired Structure from Motion task when compositing visual effects into live-action photography. These surfaces are required for choreography of a scene, casting physically accurate shadows of CG elements, and occlusions. We present a novel framework for generating the geometry of landscapes from extremely noisy point cloud datasets obtained via limited resolution techniques, particularly optical flow based vision algorithms applied to live-action video plates. Our contribution is a new statistical approach to remove erroneous tracks ('outliers') by employing a unique combination of well established techniques–including Gaussian Mixture Models (GMMs) for robust parameter estimation and Radial Basis Functions (RBFs) for scattered data interpolation–to exploit the natural constraints of this problem. Our algorithm offsets the tremendously laborious task of modeling these landscapes by hand, automatically generating a visually consistent, camera position dependent, thin-shell surface mesh within seconds for a typical tracking shot.

Categories and Subject Descriptors: I.4 [**Image Processing and Computer Vision**]: Scene Analysis - Surface Fitting.

Additional Key Words and Phrases: Radial Basis Function, Gaussian Mixture Model.

## 1. INTRODUCTION

Building a 3D representation of a filming location is a crucial step in the visual effects

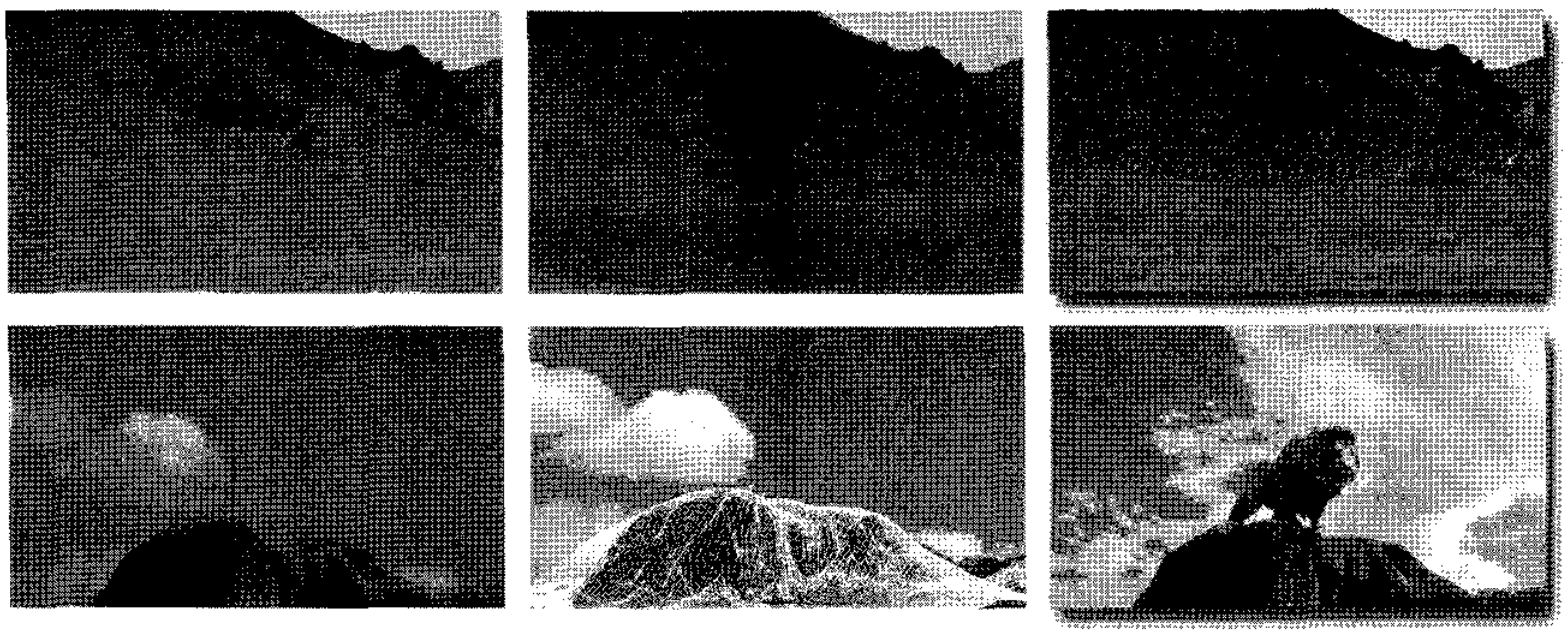

Fig. 1. Battle scenes from film "The Chronicles of Narnia: The Lion, the Witch, and the Wardrobe". In these visual effects sequences, the live-action photography plate (*Left*) is used as input to our novel, entirely automated landscape generation pipeline. An optimal flow tracking algorithm is applied in combination with a moving Radial Basis Function technique for noise filtering, in order to create a smooth surface representation (*Middle*) of the underlying landscape. Such terrain modeling is extremely useful in scene choreography and casting physically accurate shadows of CGI elements, as shown in the final composite (*Right*).

pipeline. This geometry is primarily required in the *compositing* of graphics and animation into a live-action background, where the bi-directional coupling is emphasized: computer generated characters should influence their environment by casting shadows, while conversely the filmed world should contribute to the appearance of CGI via partial occlusions and illumination conditions. Fig. 1 illustrates a few scenarios where we wish to restore the 3D information of a scene in order to convincingly import computer graphics into the image. Our algorithm is able to recover an appropriate terrain mesh, using as input just the original video sequence. Such an approach (acting on very limited input) is commonly demanded in practice, when high-quality geological survey data or LIDAR scans of the actual filming location prove difficult to obtain. Traditionally it is common in these circumstances for a skilled matchmoving artist to spend significant time to create and properly align a terrain, making any possible automation a very practical pursuit.

The high quality reconstruction of a surface from noisy point samples is a challenging problem that has received much attention in recent years, primarily due to advancements in automatic shape acquisition: data output from scanners and image based techniques is often presented as a large set of points. However unlike much of the previous work in this area (see Section 2), there is little attention for reconstructions from severely incomplete and noisy datasets, as we would expect to extract from the monocular, uncalibrated camera image sequences we consider here. These datasets are extremely unreliable due to triangulations in $\mathbb{R}^3$ that cannot be rejected solely on the basis of stereographic reprojection errors or illuminance information alone. Moreover, the dataset sampling is highly nonuniform, and outliers are arbitrary. Our task is to obtain a final surface that, while not necessarily accurate to the ground truth topography of the scene, is consistent with the input image sequence (no 'sliding' of geometry w.r.t. the images), and could thus be used directly to composite computer graphics, while maintaining acceptable runtime and memory requirements for use in a production environment.

In order to make this daunting problem more manageable, our key insight is to minimize the degrees of freedom by exploiting a number of unique natural constraints to obtain a robust solution. In particular, we claim that natural landscapes can be represented as a 2D "height function" with respect to the vertical axis because gravity combined with erosion tends to collapse any overhangs. By a similar argument we additionally enforce a smoothness constraint, as spectral analysis of these 2D functions show that the high frequency spectral modes decay very quickly. Starting with a noisy sampling generated by optical flow, we thereby propose a post filtering stage to remove outlying points. This is achieved by locally fitting a radial basis function to the neighborhood of a point to predict its error likelihood, and a Gaussian mixture model to automatically determine the appropriate error cutoff threshold. The final step is to use this cleaned dataset in a surface generation method that captures the many benefits of using RBFs as a scattered data interpolant for the remaining irregularly sampled dataset.

## 2. RELATED WORK

A rich history of Structure from Motion literature exists, and in particular our algorithm builds on the successful work of [Fitzgibbon and Zisserman 1998] for robust camera acquisition and 3D tracking. Surveyless uncalibrated camera tracking (*i.e.* no a priori knowledge of 3D locations in the scene) became feasible by capitalizing on achievements such as optical flow tracking [Barron et al. 1994], statistical analysis based on RANSAC [Forsyth and Ponce 2003; Hartley and Zisserman 2004], and structure from motion ideas employing epipolar geometry constraints of the observed images, as is clearly expressed in [Hartley and Zisserman 2004; Fitzgibbon and Zisserman 1998]. Elements from all these sources described above are present in the 3D camera tracking system we use in order to obtain the initial point cloud that serves as input data for the terrain generation algorithm described in Section 4.

This leads us to the adjacent and related field of surface reconstruction, which aims to generate a manifold watertight mesh with minimal geometric and topological noise, where input is in the form of a (comparatively) well sampled point set with few outliers, or outliers with a fairly regular distribution. The first very popular approach to tackle this problem is based on meshing, commonly employing Delaunay/Voronoi techniques to create a triangulation that interconnects the entire dataset, and requires a post-processing step to smooth the resultant geometry and minimize spurious oscillations. An excellent survey of this general approach to reconstruction can be found in [Cazals and Giesen 2004]. Alternate approaches are mesh-free methods. The most successful algorithms directly generate an approximated surface, often as the level set of an implicit function, for which one can speak of local and global schemes. A most simple local construction is to individually consider subsets of nearby points, estimate the tangent plane, and create an implicit function as the signed distance to the tangent plane of this subset of points [Hoppe et al. 1992]. Some alternate algorithms require point normals [Kazhdan et al. 2006], while others [Hornung and Kobbelt 2006; Schall et al. 2005] do not.

Global fitting strategies [Samozino et al. 2006; Kolluri et al. 2004] often generate an implicit function from the sum of polyharmonics (RBFs) centered at the point samples. Unfortunately, it is typical for radial basis functions to rely on a basis with global support, requiring the solution to a dense linear system, where the condition number of this matrix is generally coupled to the size of the dataset, and can thus cause significant instabilities for large problems. [Carr et al. 2001] addressed this challenge by introducing the fast multipole method. While the Multi-level Partition of Unity approach [Ohtake et al. 2003] also elegantly circumvents many issues plaguing RBFs, like most implicit function based shape representations, it is not capable of successfully handling objects with boundaries. This is a key motivation in the design of our new method. Our approach is similar to [Schall et al. 2005], who propose a point cloud filtering stage followed by a meshing stage [Amenta et al. 2001], however our algorithm incorporates RBFs for meshing: the tracked data is segmented into numerous objects of interest, decomposing the point set into several small problems for which numerical conditioning is quite acceptable, rendering the multipole method of [Carr et al. 2001] redundant for our needs.

Very recent commercial softwares (*e.g.* [2d3]) have introduced terrain generation into their matchmoving pipeline, however the methods employed are still rudimentary. For example, it is observed in several matchmoving packages that a terrain is created by applying a Delaunay triangulation to the 2D projection of the tracked point cloud onto the base plane. The mesh resolution and shape is thus directly tied to the number and distribution of tracked points, without regard to possible outliers. Similarly, most published works in photogrammetry (for land surveying) focus on out-of-core algorithms for rendering and managing massive datasets (*e.g.* [Isenburg et al. 2006; Lindstrom and Pascucci 2002]); not directly relevant in this context.

What follows is a brief mathematical formulation of the approximation problem, and an outline for a succinct and robust algorithm to clean datasets obtained by vision algorithms for use in matchmoving applications.

## 3. BACKGROUND

Given a set of $n$ samples from a real valued 2D "height" function $f(x)$ with values $\{f_1, ..., f_n\} \subset \mathbb{R}$ at the locations $\{x_1, ..., x_n\} \subset \mathbb{R}^2$, we wish to find a function $\tilde{f}(x)$ that best approximates $f(x)$. In other words, the optimal function $\tilde{f}(x)$ minimizes the least squares error objective

$$E(\tilde{f}) = \sum_{i=1}^{n} (f_i - \tilde{f}(x_i))^2. \tag{1}$$

Using the RBF method, one selects a function $\tilde{f}(x)$ defined by a linear combination of the basis function $\phi : \mathbb{R}^+ \rightarrow \mathbb{R}$ in a way that is expressed by

$$\tilde{f}(x) = \sum_{i=1}^{n} w_i \phi(|x - x_i|), \tag{2}$$

where $|\cdot|$ denotes the standard $L^2$ Euclidean norm, and $\{w_1, ..., w_n\} \subset \mathbb{R}$ a set of unknown weights to be solved for such that all $\tilde{f}(x_i) = f_i$. [Carr et al. 2001] suggests various appropriate possibilities of the basis function $\phi$, and we use the thin-plate spline $\phi(x) = x^2 \log x$ that is very effective for our problem, although even a most trivial function such as $\phi(x) = x$ can also be adequate for fitting the smooth bivariate terrain height.

## 4. ALGORITHMIC APPROACH

Interpolation schemes are conceived to minimize an appropriate error objective function, and one such approach is the RBF method. Because our input samples are unreliable, our contribution is a new method of pre-processing the point cloud that results from tracking a video sequence, possibly using the techniques referenced in Section 2. We describe this filtering algorithm to clean the noisy point cloud, and finally propose a useful approach to generating the final surface mesh. The filtering algorithm is a local approach that can be outlined as follows: (1) locally fit a 2D radial basis function $h_i$ to the neighborhood of a vertex $i$ in the initial point cloud $\mathcal{P}$, (2) quantify the error between the actual height of point $i$ and the estimate based on our local reconstruction $h_i$, (3) apply a Gaussian mixture model to estimate a dataset dependent error tolerance, (4) cull erroneous points from the dataset. Finally a RBF interpolation of the cleaned dataset is used to generate a quad mesh of the terrain.

**Position Dependent Logarithmic Sampling:** In many situations, a user may wish to specify an appropriate camera hither and yon distance, in which case any points outside of this range are culled. More generally, we emphasize points near to the camera because the visual detail is most important there, and also because parallax is more predominant closer to the camera, leading to accurate optical flow tracks. This is achieved by a logarithmic sampling process: a vertex $i \in \mathcal{P}$ is sampled with a probability of $1/\log d_i$ where $d_i$ is the *minimum* distance of $i$ to the camera position, which can be expressed formally as $d_i = \min_m |i - c_m|$, where $c_m$ denotes the camera position vector at frame $m$ of the image sequence. In other words, we determine the minimum distance between the tracked vertex $i$ and the camera position over the entire video, necessary because the camera needn't remain static. We now update $\mathcal{P}$ to reflect this subsampling of the initial point set.

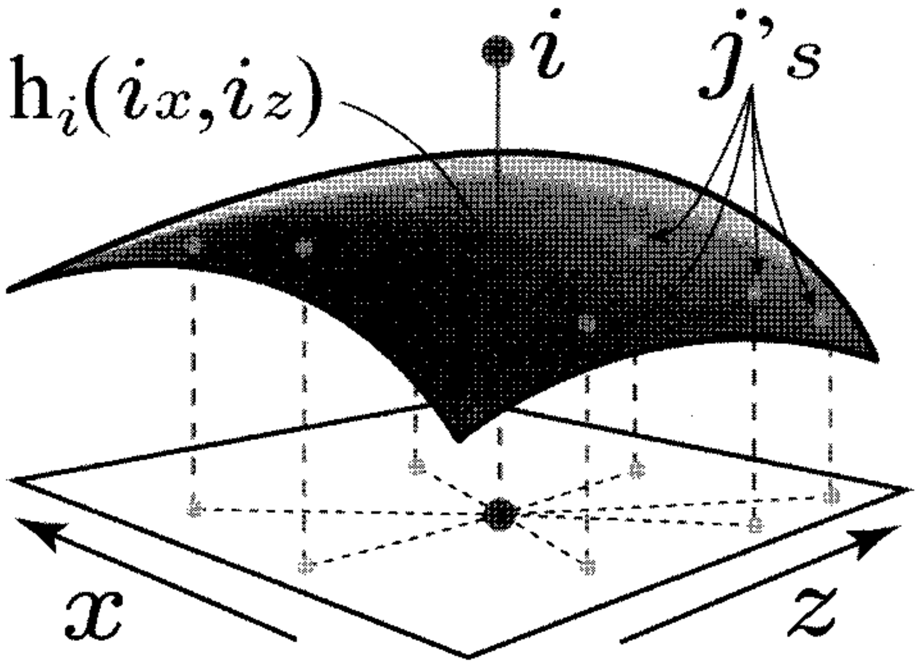

Fig. 2. Quantification of local errors: the accuracy of vertex $i$ is estimated by creating a radial basis function $h_i$ from the neighboring points $j \in N(i)$ and considering the deviation $\delta_i = (i_y - h_i(i_x, i_z))$. The neighborhood $N(i)$ simply contains all points $j$ within some $\varepsilon$-ball from vertex $i$ in the $xz$-plane.

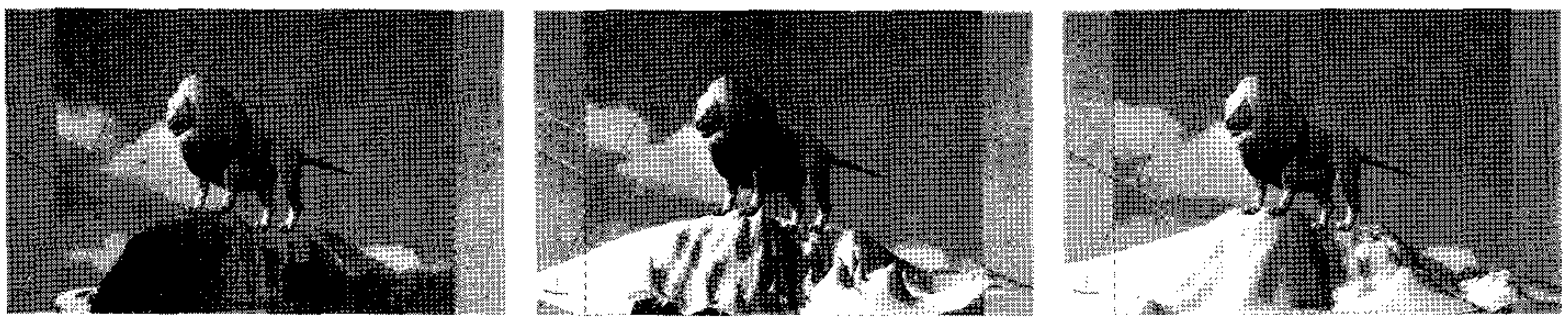

Fig. 3. We contrast the surface reconstruction of the mountain top terrain (*Left*, also see Figure 1): it is first performed without noise filtering of $\mathcal{P}$ (*Middle*), and once again using our RBF/ GMM based filtering algorithm (*Right*). Notice our method helps to reduce spurious high frequency modes in the resultant surface.

**Quantification of Error:** It is necessary to evaluate the remaining points on an error criterion; errors are first estimated by employing a technique based on local fitting of radial basis functions. Each vertex $i \in \mathcal{P}$ is considered in turn by generating a RBF $h_i(x, z)$–see Section 3–that intersects all the neighboring points $j \in N(i)$ as smoothly as possible, in order to see how closely the neighborhood $N(i)$ is able to predict the true surface elevation at vertex $i$. The neighborhood $N(i)$ is defined as the set of closest vertices $j$ to $i$ in the *xz*-base plane, as depicted. We now evaluate $i^*_y = h_i(i_x, i_z)$ and quantify the error between the height of i and our local prediction based on interpolation of $N(i)$ by considering $\delta_i = (i_y - i^*_y)$, please refer to Fig. 2 to clarify this construction. Note that a non-standard xz-base plane can be specified (*e.g.*, when modeling a mountain cliff) by applying a rotation to $\mathcal{P}$ ahead of the aforementioned computations.

**Rejection of Outliers:** For all vertices $i \in \mathcal{P}$ we compute $\delta_i$, which defines a onedimensional signal of errors. It has been observed through empirical testing that such signal typically follows a Gaussian distribution with mean $\mu \approx 0$ and relatively low variance $\sigma^2$. A Gaussian Mixture Model (GMM) code is invoked [Bouman 2005], based on the classic EM algorithm, to estimate these parameters $\mu$ and $\sigma$ for the point cloud under consideration, and finally exclude any vertices that lie further than $2\sigma$ from $\mu$. The remaining points are considered valid optical flow triangulations in 3D, and are used to compute a *global* RBF function $\mathcal{R}$ that defines our final landscape. One possible caveat in this setup is in the presence of *systematic* reprojection errors. Frequently not one, but two Gaussians are automatically detected by the GMM system–the Gaussian with larger variance almost always corresponds to the systematically erroneous data, hence we neglect the entire distribution from consideration, and we proceed with the alternate remaining Gaussian as above, by removing points further than $2\sigma$ away. Once again, $\mathcal{P}$ is updated, now to exclude outliers. Figure 3 demonstrates how this outlier rejection improves the reconstruction quality.

**Surface Alignment:** A high resolution quad mesh is created, with principle axes aligned with the principle axes of the dataset $\mathcal{P}$ (expressed as a matrix $\mathbf{X}$) through use of Principle Component Analysis (PCA). We compute the covariance matrix $\mathbf{C} = (\mathbf{X} - \overline{\mathbf{X}})(\mathbf{X} - \overline{\mathbf{X}})^T$ where $\overline{\mathbf{X}}$ denotes the mean position, and apply an eigen decomposition $\mathbf{D} = \mathbf{V}\mathbf{C}\mathbf{V}^T$ to extract the principle direction vectors $\mathbf{V}$. This matrix can be thought of as a rotation matrix to be applied to $\mathbf{X}$ for alignment. The vertices of the mesh are now modulated in the *y*-axis according to the global RBF function that represents our elevation, in order to obtain a polygonal representation of the surface. Applying the inverse rotation $\mathbf{V}^{-1} \equiv \mathbf{V}^{\mathrm{T}}$ completes the alignment.

**Point Set Clustering:** In generating our final global RBF terrain field, numerical instabilities (inversion of a singular matrix) restrict the size of our problem domain to a cloud of approximately $|\mathcal{P}| = 500$, which might seem overly restrictive for longer tracking shots. In practice, however, this is not the case, as it usually desirable to generate a separate mesh for each object (hill, boulder, etc) in the scene. To achieve this, we integrate an automated clustering process into the framework, by the invocation of the GMM code a second time (as illustrated in Figure 4), now for the purpose of partitioning the cloud $\mathcal{P}$ into several disjoint sub-clusters $\mathcal{P}_k \subset \mathcal{P}$ such that $\cup_k \mathcal{P}_k = \mathcal{P}$ and $\cap_k \mathcal{P}_k = 0$. As we wish to cluster the data on the xz-plane, we feed the xz-components of $\mathcal{P}$ to the GMM algorithm, which is now given the additional degree of freedom to determine for itself the optimum number of Gaussians to use in representing the data, assuming some reasonable upper threshold (by default we use 10; consult [Bouman 2005] for details). For each extracted sub-cluster of points, we perform the filtering algorithm outlined above to clean each set $\mathcal{P}_k$ individually and then generate one RBF $\mathcal{R}_k$ per cluster. If we *do* wish to obtain a single mesh of the complete scene, a 'global' RBF $\mathcal{R}$ can be created by taking weighted contributions from all of the $\mathcal{R}_k$ functions, where the weight is dictated by the probabilistic distance from the evaluation point of the RBF to the centers of each cluster. An alternate approach is the multipole method [Carr et al. 2001] to bypass the numerical instabilities and directly generate a RBF from the whole dataset, although in reality this is usually unnecessary.

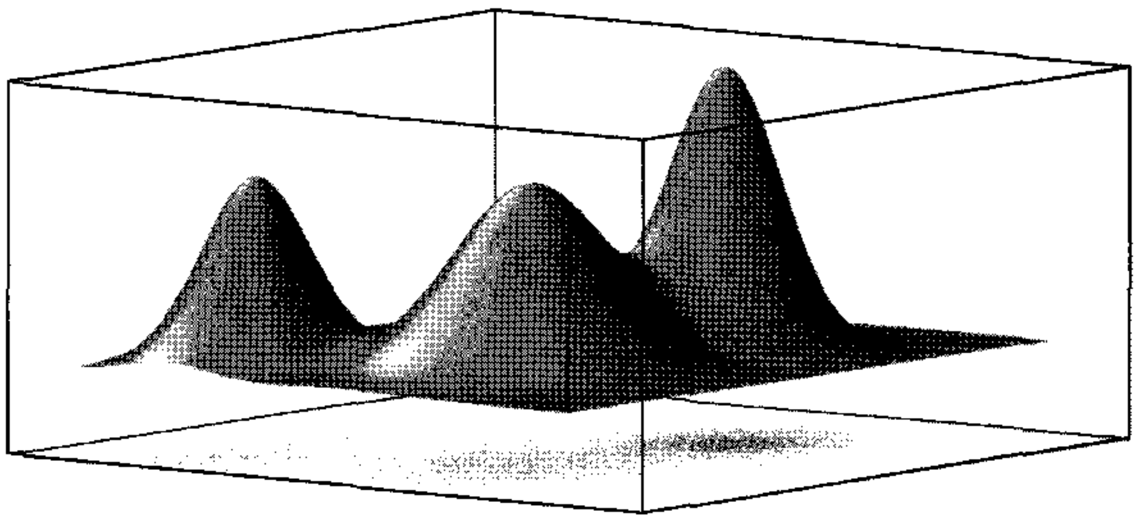

Fig. 4. A Gaussian Mixture Model algorithm is used to estimate the number of normal distributions (and their parameters $\mu$ and $\sigma$) of a scattered point set (as shown on the base plane).

## 5. RESULTS

Generating a RBF requires the solution to a dense linear system, which is performed by an efficient Cholesky Factorization. Our local RBF method typically requires us to solve *hundreds* of such systems, yet our solution is very fast because we enforce a local support, *i.e.* $|N(i)| \leq 25$ is already sufficient to provide impressive results. As highlighted by the supplemental video in addition to Figure 3, the filtering algorithm is a highly effective intermediate step for removing spurious oscillations in the final terrain geometry. Figures 1, 5, and 6 all present various situations in which our approach was successful in generating a terrain geometry that is very consistent with

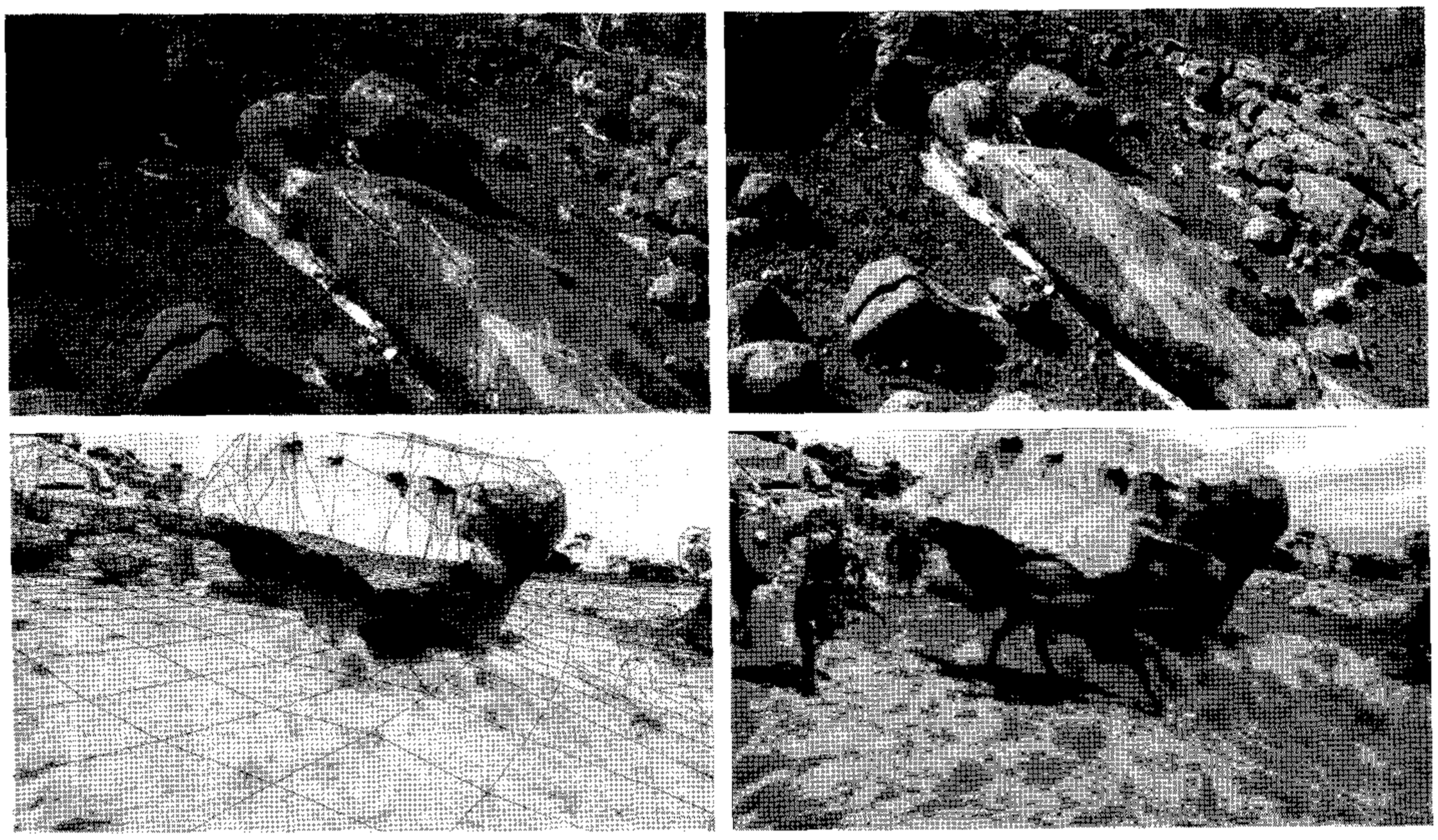

Fig. 5. Additional terrain geometry created with our system for "The Lion, the Witch, and the Wardrobe". Landscape meshes are created from the original plates (*Left*) and used to composite characters and shadows into the scene (*Right*). Note that for distinct terrain features in a scene, the filtering algorithm is applied on the distinct objects independently (*Lower Left*).

the input sequence. These examples took between 0.5 to 2 minutes of computation on a 2Ghz Pentium 3 with 512MB RAM.

## 6. CONCLUSION AND FUTURE WORK

We have presented a new method for robust outlier detection in noisy point cloud datasets obtained via optical flow of an input image sequence. The algorithm used locally fit radial basis functions to estimate errors, and a Gaussian mixture model mechanism to determine an appropriate cutoff threshold. The GMM was also used for automated clustering of points. The resulting "clean" point samples were used in a geometry reconstruction step to obtain a final three-dimensional representation of the photographed terrain. Our approach was successfully employed in several feature films, most notably 'The Chronicles of Narnia', and the recent film 'The Kingdom'.

Future work includes extending the approach to handle fully 3-dimensional terrain, complete with overhangs and other complicated geometries. Such an improvement requires modification to our quantification of errors, reminiscent of [Hoppe et al. 1992], which may be outlined as follows: (1) locally fit a 3D radial basis function $\mathcal{R}_p$ to the neighborhood of a point $p$, (2) estimate the tangent plane of $\mathcal{R}_p$ to obtain the orthogonal normal vector $\hat{n}_p$, (3) calculate the error as the distance of $p$ to $\mathcal{R}_p$ in the normal direction $\hat{n}_p$. The GMM error tolerance procedure can then be used as previously described. Another open question raised by our work is the optimality of modeling errors in extracted point clouds through a Gaussian distribution. Although this seems to be highly effective in practice, we cannot positively claim it to be ideal in all situations.

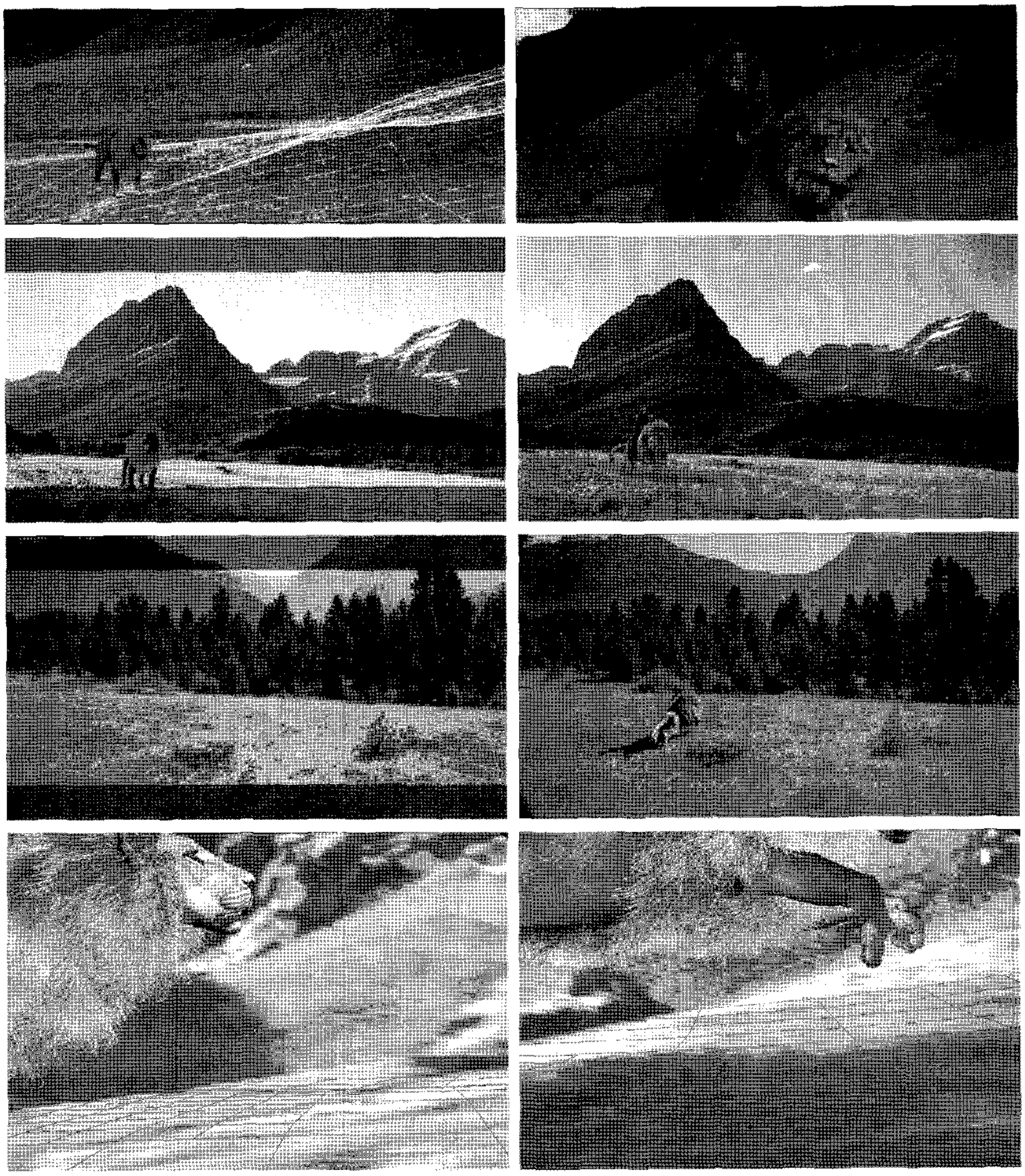

Fig. 6. Various additional scenes from the film reconstructed using our algorithm (*Left*), for use in final compositing of CGI into the frame (*Right*).

## REFERENCES

2D3. Boujou 4. http://www.2d3.com.

AMENTA, N., CHOI, S., AND KOLLURI, R. 2001. The Power Crust, Unions of Balls, and the Medial Axis Transform. In *Computational Geometry: Theory and Applications.* 127–153.

BARRON, J. L., FLEET, D. J., AND BEAUCHEMIN, S. 1994. Performance of optical flow techniques. In *International Journal of Computer Vision.* 43–77.

BOUMAN, C. A. 2005. Cluster: An Unsupervised Algorithm for Modeling Gaussian Mixtures. Tech. rep., Purdue.

CARR, J. C., BEATSON, R. K., CHERRIE, J. B., MITCHELL, T. J., FRIGHT, W. R., MCCALLUM, B. C., AND EVANS, T. R. 2001. Reconstruction and Representation of 3D Objects with Radial Basis Functions. In *ACM Siggraph.* 67–76.

CAZALS, F. AND GIESEN, J. 2004. Delaunay Triangulation Based Surface Reconstruction: Ideas and Algorithms. Tech. rep., INRIA Sophia-Antipoles.

Fitzgibbon, A. W. and ZISSERMAN, A. 1998. Automatic Camera Recovery for Closed or Open Image Sequences. In *European Conference on Computer Vision.* 311–326.

FORSYTH, D. A. AND PONCE, J. 2003. *Computer Vision, a modern approach*. Prentice Hall.

HARTLEY, R. I. AND ZISSERMAN, A. 2004. *Multiple View Geometry in Computer Vision.* Cambridge University Press.

HOPPE, H., DEROSE, T., DUCHAMP, T., MCDONALD, J., AND STUETZLE, W. 1992. Surface Reconstruction from Unorganized Points, *Computer Graphics 26.*

HORNUNG, A. AND KOBBELT, L. 2006. Robust Reconstruction of Watertight 3D Models from Non-uniformly Sampled Point Clouds Without Normal Information. In *Symposium on Geometry Processing.* 41–50.

ISENBURG, M., LIU, Y., SHEWCHUK, J., AND SNOEYINK, J. 2006. Streaming Computation of Delaunay Triangulations. In *ACM Siggraph.* 1049–1056.

KAZHDAN, M., BOLITHO, M., AND HOPPE, H. 2006. Poisson Surface Reconstruction. In *Symposium on Geometry Processing.* 61–70.

KOLLURI, R., SHEWCHUK, J., AND O'BRIEN, J. 2004. Spectral Surface Reconstruction from Noisy Point Clouds. In *Symposium on Geometry Processing.* 11–21.

LINDSTROM, P. AND PASCUCCI, V. 2002. Terrain Simplification Simplified: A General Framework for View-Dependent Out-of-Core Visualization. *IEEE TVCG 8*, 3, 239–254.

OHTAKE, Y., BELYAEV, A., ALEXA, M., TURK, G., AND SEIDEL, H. 2003. Multi-level Partition of Unity Implicits. In *ACM Siggraph.* 463–470.

SAMOZINO, M., ALEXA, M., ALLIEZ, P., AND YVINEC, M. 2006. Reconstruction with Voronoi Centered Radial Basis Functions. In *Symposium on Geometry Processing.* 51–60.

SCHALL, O., BELYAEV, A., AND SEIDEL, H.-P. 2005. Robust filtering of noisy scattered point data. In *IEEE/Eurographics Symposium on Point-Based Graphics.* 71–77.

**Alexander McKenzie** received his undergraduate degree in Computer Science from University College London in 2005 and is currently pursuing a graduate degree at the California Institute of Technology. His main research interests are discrete differential geometry and its application to dynamical systems. In 2005, Alex was the recipient of the SET "Computational Science Student of the Year" award in the UK, and has recently worked as a research intern at Rhythm & Hues Studios in California, where this project was undertaken.

**Eugene Vendrovsky** received MS in Electrical Engineering from Moscow Institute of Electronics and Mathematics in 1977 and PhD in Computer Science from Academy of Sciences of the USSR in 1986. He joined Rhythm and Hues Studios in 1993 and currently as a Principal Graphics Scientist his prime responsibilities is Research and Development in the fields of Computer Vision, Image Processing and Physical Simulation.

**Junyong Noh** is an Assistant Professor in the Graduate School of Culture Technology at the Korea Advanced Institute of Science and Technology (KAIST). He earned his computer science Ph.D. from the University of Southern California (USC) in 2002 where his focus was on facial modeling and animation. His research relates to human facial modeling/animation, character animation, fluid simulation, video visualization, and interactive media. Prior to his academic career, he was a graphics scientist at a Hollywood visual effects company, Rhythm and Hues Studios.