Image-based Model Completion

Alan Brunton[1] Stefanie Wuhrer[1,2] Chang Shu[1]

[1]*Visual Information Technology Group*
National Research Council Canada
1200 Montreal Rd., Ottawa, Canada
firstname.lastname@nrc-cnrc.gc.ca

[2]*School of Computer Science*
Carleton University

Abstract

Geometric models created from range sensors are usually incomplete. Considerable effort has been made to fix this problem, ranging from manual repairing to geometric interpolation. We propose using multi-view stereo to complete such models. Our approach is practical and convenient because when scanning and object or environment one usually takes photographs to texture the resulting model. By using the incomplete scan data as a boundary condition, we use a variational multi-view approach to estimate the missing data.

1 Introduction

The wide use of laser range sensors and other modalities of 3D sensing device has produced increasingly detailed 3D models. However, one problem that has plagued 3D modeling since its beginning is the incompleteness of the models. Holes, due to a variety of reasons, are usually present in the models built from range scans. Some holes are caused by intrinsic limitations of the sensors. For example, a triangulation-based sensor requires that every reconstructed point to be visible from two different view points of the sensor. Others are the results of object self-occlusion, insufficient view coverage, and shallow grazing angles. Davis et al. [5] gave an account of holes and their causes in different situations.

Many methods have been proposed for hole-filling. The majority of them fill holes by interpolating the nearby geometry [5]. This method is only effective when the holes are small. Manual, interactive tools are often used to fix large holes. For keeping the surface smoothness the same as the nearby areas, interactive methods were proposed in a way that imitates image touch-up tools such as found in the Adobe Photoshop where users “cut and paste” surface geometry from nearby area [19].

In practice, we usually take colour digital images for recording the textures while digitizing objects and environment. These 2D images can also be used to improve the geometry of the 3D models. Image-based reconstruction techniques such as stereo [7], photogrammetry [1], and shape from shading [20], were used to augment the scan models.

In this paper, we propose a new image-based method for filling holes in the 3D models. The method is based on multi-view stereo. The key observation is that the boundaries of the holes are easily identifiable. This allows us to solve a constrained problem which is more manageable than the general multi-view stereo problem. We start with an initial filling of the hole using a geometric interpolation method. Then the initial filling surface is deformed to fit the image data using a surface-oriented multi-view stereo approach. This problem has a variational formulation – finding a surface such that it optimizes the photoconsistency energy function [11]. It has been solved previously by PDE-based method using either the level-set method [11] or mesh deformation [8]. In both cases, an initial, closed surface that enclose the object to be modeled is assumed and the algorithm evolves until it converges to a minimal energy solution. However, the problem is an ill-posed one and often the algorithm is trapped in local minima. Unlike the classical solutions, we make use of the known boundary of the surface and solve a boundary-value problem. This makes the solution easy to reach because the boundaries impose strong constraints. Furthermore, as we always maintain a smooth surface, occlusion and specularity, the two most prominent problems for stereo, are alleviated.

We demonstrate in this paper that 2D images can be used as geometry recording data complementary to the range data. Since we can take digital images with much more flexible viewpoint selections, we can record those areas of the object that are difficult for the range sensors to reach.

2 Related Work

Geometric methods for filling holes in a mesh model interpolate hole boundary or extrapolate surface geometry from the surrounding areas. Two types of representations are used. One works directly on the surface mesh. Barequet and Sharir [3] give an $O(n^3)$ algorithm for triangulating a 3D polygonal boundary which represents the boundary of a hole. Liepa [14] extends this method to include a surface fairing step. Dey and Goswami [6] use a Delaunay triangulation-based method, called Tight Cocone, in which tetrahedrons are labeled as in or out. The advantage of working directly with the surface mesh is that the rest of the surface is unchanged when the holes are filled. This class of algorithms usually only deal with holes with a single boundary.

The other type of representation is the volumetric representation, where the surface mesh is discretized into regular 3D grids or an octree structure either locally or globally. Davis et al. [5] diffuse the geometry from the hole boundary to the interior until the fronts meet. This method handles complex topological configurations such as holes with islands. However, it may change the existing mesh. Podolak and Rusinkiewicz [16] embed the incomplete mesh in an octree and use a graph cut method to decide the connections between pairs of the hole boundaries. It resolves difficult boundary topologies globally.

A number of authors have used 2D images to enhance the quality of the 3D models generated from range scans. Dias et al. [7] fuse stereo reconstruction with 3D points obtained from range sensors. Abdelhafiz et al. [1] and El-Hakim et al. [9] combine range image with photogrammetric reconstructions. The vision-based approach of Dias et al. needs points from range images to be close to the reconstructed stereo points. Therefore it does not work well with large holes. The photogrammetric approach needs a lot of manual interactions. Recently, Xu et al. [20] use 2D images captured for textures to fill holes in a shape-from-shading scheme. They learn the surface normal from the existing mesh geometry. However, the single-view approach to reconstruction has intrinsic limitations in handling non-Lambertian surface and difficult lighting conditions.

There is an extensive literature for multi-view stereo. Seitz et al. [18] give a recent survey. Our work is closely related to the surface-based approaches such as in [8, 10, 11], whereas the difference is that we explicitly use boundary conditions.

3 Initial Mesh

In this Section, we describe how to obtain an initial manifold that can be deformed by multi-view stereo. Given a triangular manifold S with partially missing data obtained from a laser range scanning device, we aim to fill the holes of S by an initial triangular mesh. Section 4 discusses how a sequence of calibrated images can then be used to efficiently deform the initial triangular mesh using multi-view stereo methods.

We first identify the boundaries of holes of S. Since S is a manifold, we can find the edges of S tracing a hole of S as edges of degree less than two, since every edge not adjacent to a hole of S has degree two. We compute an initial mesh for each loop of boundary edges of S separately. Filling a hole bounded by m edges with a triangulation that does not have self-intersections may require an exponential number of Steiner points in m [13]. Furthermore, the problem of deciding whether a non-self-intersecting triangulation filling a hole on the boundary of S exists is an NP-complete problem [2]. Hence, we do not require that the initial mesh avoids self-intersection.

To obtain a coarse initial mesh filling a hole of S bounded by m edges, we compute a triangulation of the boundary loop that does not add Steiner points and that minimizes the total area of the resulting triangulation. The approach used to find this triangulation was first proposed by Barequet and Sharir [3] and generalized to an arbitrary weighting scheme by Barequet et al. [2]. The approach proceeds by dynamic programming and takes $O(m^3)$ time and $O(m^2)$ space. Note that this approach yields a non-self-intersecting initial mesh in non-pathological cases.

Before multi-view stereo can be applied to the initial mesh, the initial mesh needs to be refined to have the same resolution as the mesh surrounding the hole. We refine the mesh using the approach by Chew [4] that adds Steiner points and computes the Delaunay triangulation of the added points. The technique is guaranteed to find a triangulation where all the angles are between 30^0 and 120^0 and where the edge lengths are at most twice as long as the edges of the mesh surrounding the hole. The running time of the algorithm is linear in the number of generated triangles.

For an illustration of the initial mesh, refer to Figure 1. Figure 1(a) shows the model of a chicken with a region synthetically removed to create a hole, Figure 1(b) shows a close view of the boundary of the hole, Figure 1(c) shows the filled hole without Steiner points, and Figure 1(d) shows the initial mesh used start the deformation.

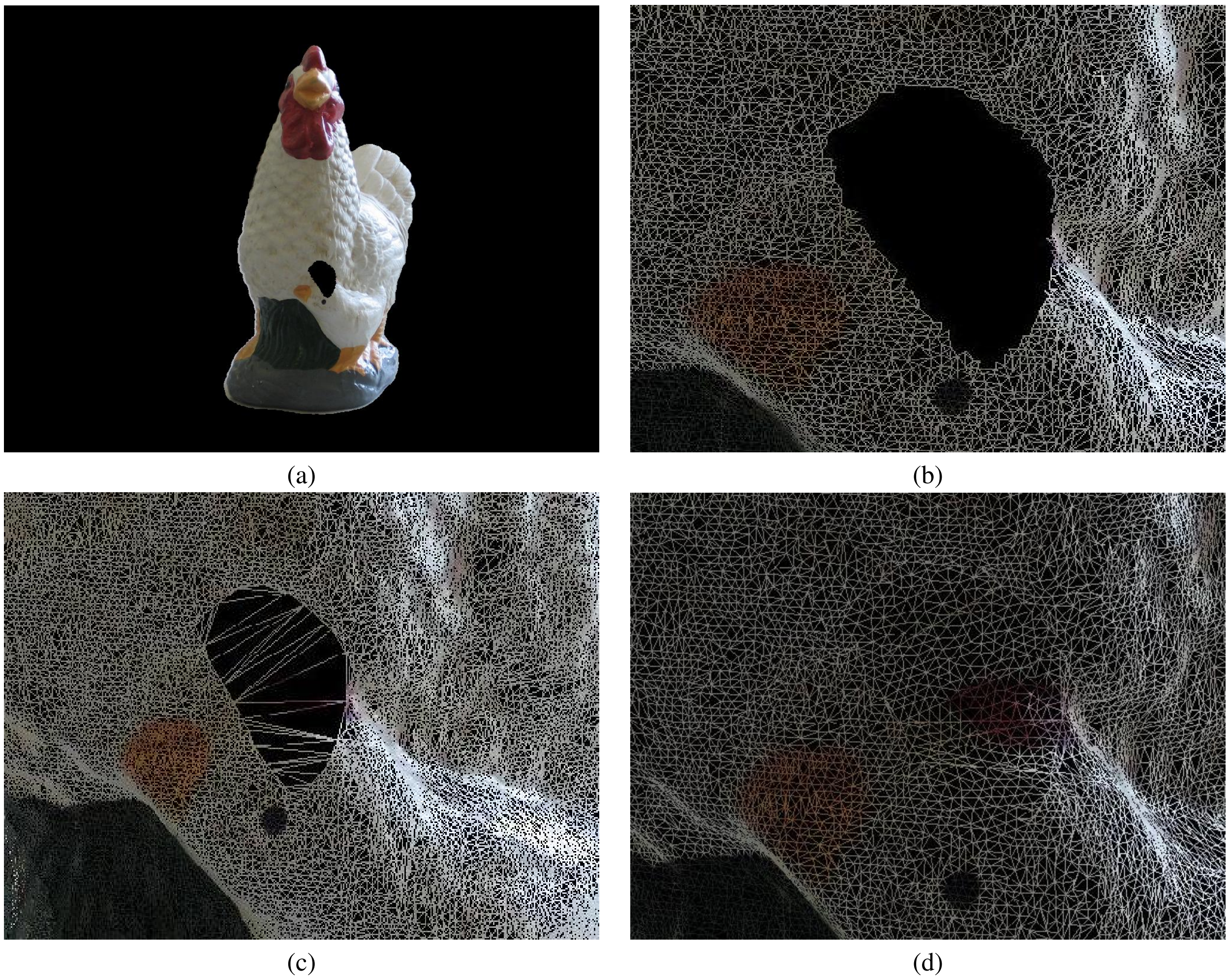


Figure 1. Illustration of the initial mesh.

4 Image-based Surface Deformation

This Section describes how we deform the initial mesh using a multi-view stereo method. The main contribution of this paper is using the hole boundary information given by the laser range data to formulate the multi-view stereo problem as a boundary value problem for constrained surface deformation under photoconsistency and Laplacian smoothing energy terms.

The input to the deformation algorithm is a watertight surface consisting of the scan S with H holes filled using initial meshes P_h for $h = 0, \ldots, H-1$ generated using the method described in Section 3, and a set of k pre-calibrated input images $I_1, \ldots, I_k$. For simplicity of notation, and without loss of generality, let us consider only a single missing surface P, which we deform to minimize the following energy function:

$$E(P) = \alpha \mathcal{M}(P) + \beta \mathcal{L}(P) \tag{1}$$

where $\mathcal{M}(P)$ is the matching cost of P for the set of input images, $\mathcal{L}(P)$ is the smoothing cost of P, and α and β are weights for the respective cost functions.

We take our matching cost from Pons et al. [17]. The matching cost of P is the sum over all pairs of input images of a pair-wise matching term:

$$\mathcal{M}(P) = \sum_i \sum_{j \neq i} \mathcal{M}_{ij}(P). \tag{2}$$

The matching term for images I_i and I_j is computed by reprojecting the matching image I_j into camera i and measuring the dissimilarity of the two. That is, for the surface P

$$\mathcal{M}_{ij}(P) = M|_{\Omega_i \cap \Pi_i(P_j)} \left(I_i, I_j \circ \Pi_j \circ \Pi_{i,P}^{-1} \right) \tag{3}$$

where M is a dissimilarity measure, Ω_i is the domain of I_i, P_j is the portion of P that is visible in I_j, Π_i is the perspective projection calibrated for camera i, and $\Pi_{i,P}^{-1}$ is the reprojection from camera i onto P. The final term, $I_j \circ \Pi_j \circ \Pi_{i,P}^{-1}$, is the image predicted by reprojecting I_j into camera i via the surface P. Henceforth, we will denote the predicted reprojection of the matching image by $\tilde{I}$. We use as our dissimilarity measure the sum of the squared differences over a small window surrounding each pixel location. Pons et al. derive the gradient $\nabla\mathcal{M}_{ij}(P)(\mathbf{x})$ of the matching term for a location $\mathbf{x}$ on P as

$$\nabla\mathcal{M}_{ij} = -\delta_{P_i \cap P_j}(\mathbf{x}) \left[\partial_2 M(\mathbf{x}_i) \, DI_j(\mathbf{x}_j) \, D\Pi_j(\mathbf{x}) \frac{\mathbf{d}_i}{z_i^3}\right] \mathbf{N} \tag{4}$$

where $\partial_2 M(\mathbf{x}_i)$ is the derivative of the dissimilarity measure with respect to its second argument, DI_j and $D\Pi_j$ indicate the Jacobian matrices of the respective functions, $\mathbf{d}_i$ and z_i are the displacement and depth relative to camera i, and $\mathbf{N}$ is the outward surface normal. The Kronecker delta $\delta_{P_i \cap P_j}$ maintains that the gradient is zero in regions not visible from both cameras. From (2) we have $\nabla\mathcal{M}(P)(\mathbf{x}) = \sum_i \sum_{j \neq i} \nabla\mathcal{M}_{ij}(P)(\mathbf{x})$.

While minimizing the matching cost, we also apply Laplacian smoothing to P to encourage a smooth deformation of the surface patch. Laplacian smoothing aims to find a deformation of the surface to satisfy $LX = \mathbf{0}$, where X is a $n \times 3$ matrix containing the 3 coordinates of each of the n vertices of P and L is an $n \times n$ matrix with elements

$$L_{i,j} = \begin{cases} 1, \; if\; i = j \\ -\omega_{i,j}, \; if\; \mathbf{x}_i \; is\; a\; neighbor\; of\; \mathbf{x}_j \\ 0, \; otherwise \end{cases} .$$

The weight $\omega_{i,j}$ for pairs of neighboring vertices in the mesh can be set in different ways. In our implementation, we set $\omega_{i,j}$ as a uniform weight that only depends on the degree of $\mathbf{x}_i$. We achieve Laplacian smoothing by defining a Laplacian cost as

$$\mathcal{L}(P) = trace((LX)(LX)^T).$$

The gradient $\nabla\mathcal{L}(P)$ of this term with respect to all vertices on P is

$$\nabla\mathcal{L}(P) = 2L^T(LX).$$

During minimization, we restrict the boundary vertices of P not to move. Since we know the derivative of the cost function in closed form, we can formulate the problem as a boundary value problem and solve it using a quasi-Newton method. The method we use to minimize our energy function is the *limited-memory Broyden-Fletcher-Goldfarb-Shanno* (LSBFGS) scheme [15].

We efficiently implemented the calculation of the predicted image, the matching cost, and the matching gradient using GPU programming in DirectX's HLSL (High Level Shading Language). Our implementation scales well with the size of the hole, and with some minor modifications could be made virtually insensitive to the number of holes in the mesh. The main bottleneck is rendering the boundary mesh (i.e. the scan data which are not missing) $O(k^2)$ times per LSBFGS iteration, which could be further optimized by storing the depth relative to each image (at the cost of higher texture requirements). Another optimization would be to only match images from nearby viewpoints. We achieve nearly minimal data transfer from graphics memory to main memory by using reduction-summation operations to sum the matching cost over each $\Omega_i \cap \Pi_i(P_j)$ on the GPU, and therefore only have to read back a single floating point value for the matching cost of a given image pair. This gives an efficient interface between computing the matching cost and gradient, and a FORTRAN LSBFGS solver [21]. There remains, however, some bottleneck in transfering the input images from main memory to texture memory, which could be alleviated at the cost of greater texture memory needs. In its current form, each LSBFGS iteration takes about one minute for the model used in the experiments for this paper.

5 Experimental Results

We tested the algorithm using the scanned surface of the chicken shown in Figure 2. The model was scanned using a ShapeGrabber laser range scanner. We obtained 11 images of the model using a Canon Powershot A520 4 mega-pixel camera. We calibrated the images by manually selecting features on the model and in the images and computing the projection matrix using the direct linear transform (DLT) algorithm [12, Chapter 6].

In our experiments, we set $\alpha = 10^{-6}$ and $\beta = 1$ when minimizing the energy function. These values were set empirically through experimentation.

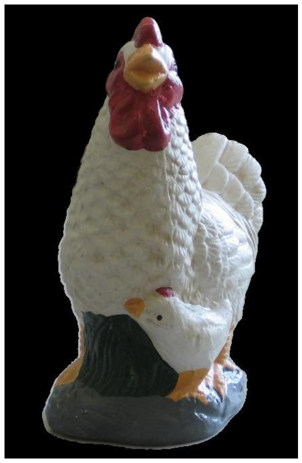


Figure 2. Scanned model.

The scanned model consists of 194491 vertices and

Mesh	Min Error	Max Error	Mean Error
Initial mesh 1	0.0031	1.9346	0.2302
Deformation 1	0.0021	1.8151	0.2256
Initial mesh 2	0.0024	1.1857	0.1645
Deformation 2	0.0007	0.8930	0.1588

Table 1. Errors of the initial meshes and the deformed surfaces with respect to the ground truth, respectively.

302930 triangles. We added artificial holes to the model in order to compare the result of our algorithm to the true surface. The first hole that was added is shown in Figure 1(a) and (b). The original model before the addition of the synthetic hole is shown in Figure 3(a). The result of our deformation algorithm on the artificial hole is shown in Figure 3(b). Note that the regularity of the mesh is preserved due to the smoothing factor while the mesh is deformed.

The second hole that was added is shown in Figure 4(b). The original model before the addition of the synthetic hole is shown in Figure 4(a) and (c). The result of our deformation algorithm on the artificial hole is shown in Figure 4(d).

Table 1 shows the errors of mesh patches with respect to the scanned model. The error between a vertex v of a mesh $\mathcal{M}$ and the scanned surface is computed as the squared distance between v and v's nearest neighbour in the scan. We report the minimum error, the maximum error, and the mean error over all vertices in $\mathcal{M}$. The meshes used to measure the error are the initial meshes and the results of our deformation algorithm for the two holes. Table 1 shows that the error decreases during the deformation. This is true in spite of the fact that the model is slightly specular.

6 Conclusions and Future Work

Traditional surface based stereo algorithms assume the surface is closed. Since the problem is ill-posed, the optimization problem can be hard to solve – often the algorithm is trapped in local minima. In our case, since we know the boundary of the surface, we can solve a boundary-value problem. Since this is a much better conditioned problem, the method will be more robust to conditions such as noise and specularity that plague the general stereo algorithms.

Future work on this project includes improving the calibration using a non-linear method, which should help both the matching cost and its gradient. Future work also includes more advanced methods for creating an initial mesh, which are more robust to complex hole boundaries. Further, there are several areas where algorithmic and implementational optimizations can be made.

Acknowledgments

The authors would like to thank Dr. Patrick Morin of the School of Computer Science, Carleton University, for the use of the laser range scanner, and Dr. Sabry El-Hakim of VIT-NRC for his help with the calibration software.

References

[1] A. Abdelhafiz, B. Riedel, and W. Niemeier. Towards a 3D true colored space by the fusion of laser scanner point cloud and digital photos. In *Proceedings of the ISPRS Working Group V/4 Workshop (3D-ARCH 2005)*, 2005.

[2] Gill Barequet, Matthew Dickerson, and David Eppstein. On triangulating three-dimensional polygons. In *Symposium on Computational Geometry*, pages 38–47, 1996.

[3] Gill Barequet and Micha Sharir. Filling gaps in the boundary of a polyhedron. *Computer Aided Geometric Design*, 12(2):207–229, 1995.

[4] L. Paul Chew. Guaranteed-quality mesh generation for curved surfaces. In *Proceedings of the Nineth Annual Symposium on Computational Geometry*, pages 274–280, 1993.

[5] J. Davis, S. R. Marschner, M. Garr, and M. Levoy. Filling holes in complex surfaces using volumetric diffusion. In *First International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT'02)*, pages 438–433, 2002.

[6] Tamal K. Dey and Samrat Goswami. Tight cocone: A water-tight surface reconstructor. In *ACM Symposium on Solid Modeling and Applications 2003*, pages 127–134, 2003.

[7] P. Dias, V. Sequeira, F. Vaz, and V. Gonçalves. Registration and fusion of intensity and range data for 3D modelling of real world scenes. In *Fourth International Conference on 3-D Digital Imaging and Modeling (3DIM'03)*, 2003.

[8] Ye Duan, Liu Yang, Hong Qin, and Dimitris Samaras. Shape reconstruction from 3D and 2D data using PDE-based deformable surfaces. In *ECCV (3)*, pages 238–251, 2004.

[9] Sabry F. El-Hakim, J.-Angelo Beraldin, Michel Picard, and Antonio Vettore. Effective 3D modeling of heritage sites. In *Fourth International Conference on 3-D Digital Imaging and Modeling (3DIM'03)*, pages 302–309, 2003.

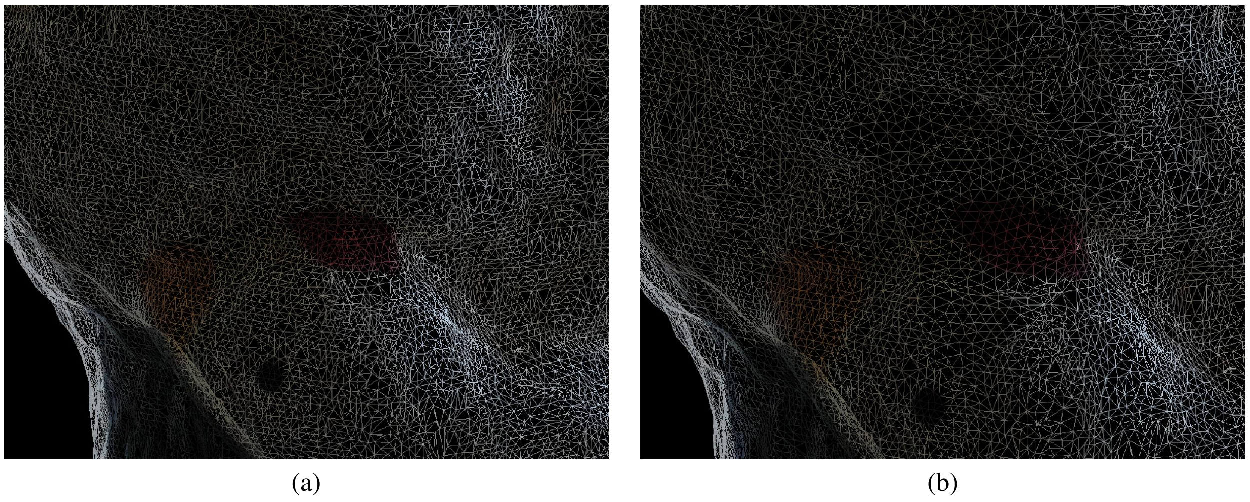


Figure 3. Deformation of the first hole in the model of a chicken compared to the original mesh.

[10] Carlos H. Esteban and Francis Schmitt. A snake approach for high quality image-based 3D object modeling. In *2nd IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, pages 241–248, Nice, France, 2003.

[11] Olivier Faugeras and Renaud Keriven. Variational principles, surface evolution, PDE's, level set methods and the stereo problem. *IEEE Transactions on Image Processing*, 7(3):336–344, 1998.

[12] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[13] Joel Hass, Jack Snoeyink, and William P. Thurston. The size of spanning disks for polygonal curves.

[14] P. Liepa. Filling holes in meshes. In *Eurographics Symposium on Geometry Processing*, pages 200–205, 2003.

[15] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 1999.

[16] Joshua Podolak and Szymon Rusinkiewicz. Atomic volumes for mesh completion. In *Eurographics Symposium on Geometry Processing*, 2005.

[17] J-P. Pons, R. Keriven, and O. Faugeras. *Modelling Non-Rigid Dynamic Scenes from Multi-View Image Sequences, in Handbook of Mathematical Models in Computer Vision, N. Paragios, Y. Chen and O. Faugeras, Eds*, 2006.

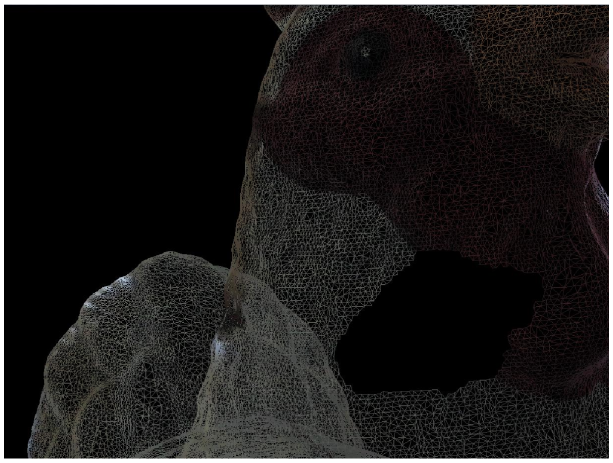
[18] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. Computer Vision and Pattern Recognition (CVPR 2006)*, 2006.

[19] Andrei Sharf, Marc Alexa, and Daniel Cohen-Or. Context-based surface completion. *ACM Trans. Graph. (SIGGRAPH'04)*, 23(3):878–887, 2004.

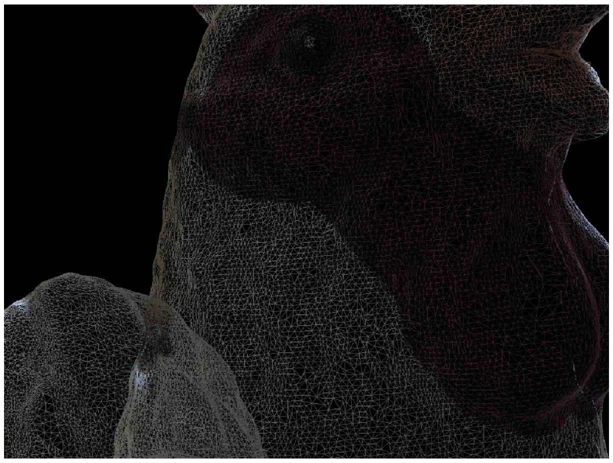
[20] Songhua Xu, Athinodoros Georghiades, Holly Rushmeier, Julie Dorsey, and Leonard McMillan. Image guided geometry inference. In *Third International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT'06)*, 2006.

[21] Ciyou Zhu, Richard H. Byrd, Peihuaung Lu, and Jorge Nocedal. Algorithm 778: L-BFGS-B, Fortran subroutines for large-scale bound constrained optimization. *ACM Trans. Mathematical Software*, 23(4):550–560, 1997.

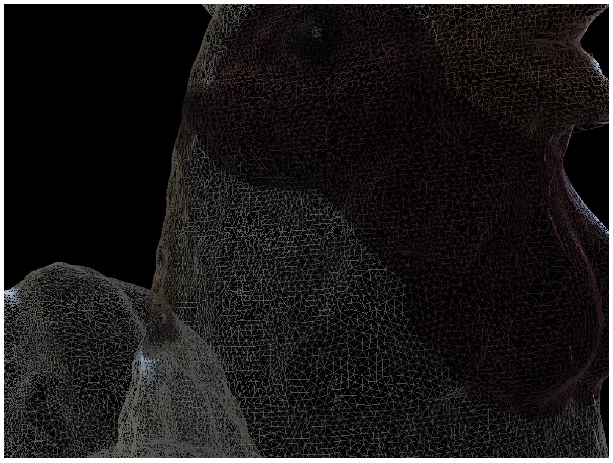
(a)



(b)



(c)



(d)

Figure 4. Deformation of the first hole in the model of a chicken compared to the original mesh.