Proceedings of DETC'04
2004 ASME Design Engineering Technical Conferences
Salt Lake City, Utah, USA, September 28 - October 2, 2004

DRAFT DETC2004-57539

HAPTICALLY GUIDED FILTERING FOR REVERSE ENGINEERING

Kristin Potter
David Johnson
Elaine Cohen
School of Computing
University of Utah
Salt Lake City, Utah, 84112
kpotter@cs.utah.edu

ABSTRACT

Reverse engineering of mechanical systems often begins with large datasets produced from laser scanning of physical artifacts. Commonly it is necessary to remove noise and filter them; however, selecting noisy regions and preserving sharp edges on desired features is difficult using standard GUI interfaces. We demonstrate a haptic interface for marking and preserving features in noisy data and for performing local smoothing operations. The force-feedback provides a natural interface for these operations.

INTRODUCTION

Reverse engineering (RE) creates computer models from exemplar physical parts, usually so that new parts can be manufactured, or modifications can be made to existing systems. This is an increasingly important problem as expensive, complex mechanical systems are maintained past their expected lifetimes, yet the original plans, or even the companies that produced them, no longer exist.

Often, the existing physical part is scanned with a laser scanner to create a large set of point samples. This dataset can be represented as a depth image or height field, reflecting the regular nature of the samples. Some scanning technologies produce fairly noisy data; however, good quality surfaces can be extracted using filtering methods. Features may need to be selected to preserve sharp edges, but users of standard GUI interfaces have difficulty marking-up these large, noisy datasets.

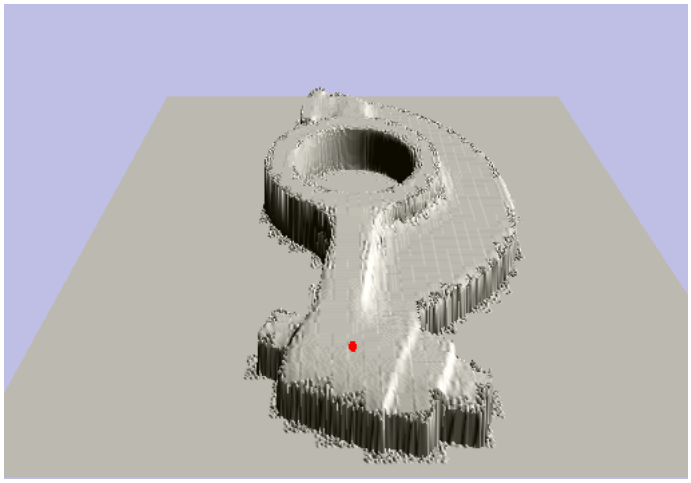

Figure 1. A SCREEN SHOT OF THE SYSTEM IN USE. THE RED POINT IS THE HAPTIC ENDPOINT.

This paper introduces a haptic interface and surface filtering algorithm for large height field datasets that allows haptically guided mark-up of feature edges for surface smoothing, and performs local, haptic-based smoothing of high noise regions. These two operations prepare noisy data for later stages in the RE pipeline, such as surface fitting or feature recognition.

We demonstrate the haptic interface and filtering on some sample depth field images. Our system can effectively deal with

datasets with millions of samples at haptic update rates greater than one kilohertz.

BACKGROUND

Our application relies on haptic rendering, haptic drawing on models, and filtering for reverse engineering. Previous work in these topics is presented here.

Haptic Rendering

Haptic rendering systems simulate the contact forces between virtual models and reflect those forces to a user with a haptic device such as the SensAble Phantom. This interaction is dependent on the geometric representation of the dataset and the haptic contact model.

Often, the geometric representation of the model used in a haptic rendering system is a triangulated mesh. This representation may lead to problems if inconsistencies in topology are present, effecting the forces that are haptically displayed to the user. Solutions to these issues treat the triangles contacted by the haptic endpoint as constraint planes [1, 2] which control the movement of the haptic endpoint across the model. Large models, however, may allow the user to cross many triangles in a single haptic update and the number of constraint planes may become overwhelming to the algorithm. Hierarchical bounding volumes may be used for high resolution models [3] allowing for local haptic calculations and achieving update rates nearly constant in time, however connectivity information is required.

The grid structure of height field datasets allows fast haptic rendering of massive models by localizing all haptic calculations to local patches on the grid, which can be represented as triangles [4] or bilinear patches [5].

The interaction between the haptic endpoint and the model can be simulated as a massless point moving along the object [2], a sphere proxy contacting the model [1], two polygonal models coming in contact with each other [6], or a ray that simulates the contact of the tip and side of the haptic endpoint to the model [7]. The latter two approaches simulate more complicated interactions between the haptic device and the model. The massless point or *god-object* and the sphere proxy techniques simulate a single position constrained to the surface of the model representing the virtual position of the haptic endpoint on the model. This constrained position is not allowed to penetrate the surface and enables the systems to haptically render infinitely thin objects such as planes and polygons. The forces returned to the user are directly proportional to the distance between the haptic endpoint and the virtual surface position.

The haptic rendering model used in this work is presented in [5] and is closely related to work by [4]. Both approaches use two modes of operation, a collision detection mode, for when the haptic endpoint is above the height field, and a mode which tracks the haptic endpoint while it is in contact with the model. This tracking stage finds a local minimum distance from the haptic endpoint to the tracked proxy point and applies forces.

Haptic Drawing

Our application allows the user to select regions of interest by drawing on the model using the haptic interface. Haptic painting was introduced for sculptured models in [8], and extended to polygonal models in [9]. The haptic interface in each system provided natural contact cues with the models and intuitive positioning of the brush. These characteristics are useful for our task, trying to select feature edges in the presence of noisy data.

Filtering for Reverse Engineering

Filtering of images and surfaces is a heavily researched topic. A related topic is surface reconstruction from scattered points [10].The filtering methods used herein are relatively simple to provide fast feedback with the haptic device.

Maintaining sharp edges is an important problem in model reconstruction and surface filtering. Automatic methods exists to infer feature edges given a threshold [11]; however, very noisy data causes problems for them. Another automatic technique for sharp corner detection was used in the context of geometry compression [12]. Marking edges for desired sharpness was done in [13], but has mostly been done in the context of fitting subdivision surfaces, where the initial mesh is small.

Filtering in the presence of sharp features is called anisotropic filtering. Local principal component analysis of the surface detected the features in [14], and anisotropic basis functions preserved those features. Anisotropic filtering on surface normals has yielded good results in the presence of noisy data, as well as feature preservation, see [15].

APPROACH

The main components of a haptically assisted filtering system are the haptic contact and force feedback models, the edge marking and global filtering algorithm, and local, haptically-guided smoothing.

The haptic model uses bilinear patches to represent the height field data and performs at haptic update rates (at least one kilohertz) independent of the size of the dataset. Algorithms are used for for fast collision detection between the haptic endpoint and the model, closest point tracking to follow the haptic endpoint while in collision, and penetration calculations to determine feedback forces. These penetration calculations are sensitive to variations in the data, and thus high frequency noise is echoed to the user through the haptic endpoint which is distracting and limiting to the user.

Noise present in the scanned data must be removed in order to gain a good quality surface for further RE applications.

The reduction or removal of noise can be done through global filtering of the dataset. However, applying a uniform blurring filter across the whole dataset will smooth sharp edges, thus result in a model with deteriorated quality. To maintain sharp edges, the user haptically marks edges and any other region not to be filtered. The selected out regions may then be filtered independently if desired.

User selected regions maintains sharp edges while removing global noise. However, the regions tend to be very noisy, making it hard to precisely select the areas of interest. A local filtering method is provided which is haptically driven, allowing the user to use the haptic endpoint as a smoothing tool the strength of which is proportional to the penetration depth.

ALGORITHM

The biggest challenge faced in this system is maintaining the haptic update rates which are much faster than visual updates, about 1000 frames per second compared to 30 frames per second for visual. The haptic system without filtering has update rates fast enough to allow the addition of local filtering, but global filtering cannot be completed within haptic rates, and thus the haptic interface is suspended while global filtering takes place.

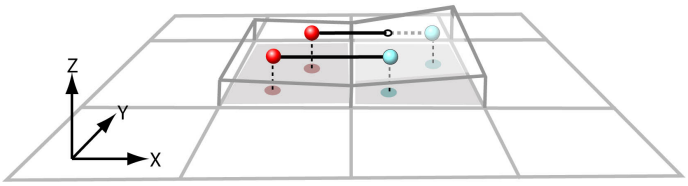

Figure 2. THE PROJECTION OF HAPTIC RAY ENDPOINTS ONTO THE XY PLANE AND TWO POSSIBLE INTERSECTION SCENARIOS.

Haptic Rendering

The main duty of the haptic rendering algorithm is to determine feedback forces by maintaining information about the interaction of the haptic endpoint and the model. Specifically, the haptic endpoint is allowed to move freely until it intersects with the model. A local point on the surface closest to the haptic endpoint is found (the proxy point), and the relationship between the penetration direction and the surface normal at the closest point determines the forces to be applied.

The approach used to track haptic movement is a historical approach, meaning the previous and current haptic positions are used in the calculations. The two haptic points form a ray known as the *haptic ray*. For each movement of the haptic endpoint the intersection between the haptic ray and the model is approximated. This approximation first projects the ray onto the XY plane as shown in Figure 2 (we assume that the Z coordinate is the height data). Starting at the previous haptic position, the maximum height of each bilinear patch in the path of the haptic ray is compared to the minimum height of the ray (that is the minimum height of the previous or current haptic positions). If the minimum height of the ray is less than the maximum height of the patch, then an intersection is possible as demonstrated in Figure 2. The first patch to pass this check is used as the starting patch for the local closest point search. This approach will detect false intersections, with the only consequence being that some rays that do not intersect the model will go through the consecutive steps of the algorithm before the ray is found not to intersect the model.

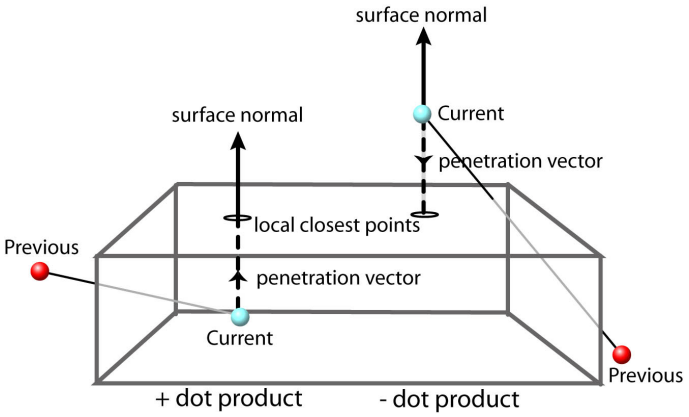

Figure 3. THE DOT PRODUCT OF THE PENETRATION VECTOR AND THE SURFACE NORMAL DETERMINES IF FORCES SHOULD BE APPLIED.

Once it is determined that the haptic ray intersects the model, the search begins for the local closest point. This is the point on the surface of the model that minimizes the distance between the surface and the haptic endpoint. Newton's method is used to solve for the local closest point and may return positions outside of the current patch. In this case, the search must move to the next bilinear patch determined by the parametric values of the out-of-bounds point. Concavities in the model may cause an infinite loop when moving to the next bilinear patch and thus the point bounded to the edges and vertices of the current patch and closest to the haptic endpoint is used to determine a neighborhood of patches to search. The search neighborhood consists of the patch or patches that share the edge or vertex location of the bounded closest point. Next, for each patch in the search neighborhood, a bounded closest point is found. The distances from each of the neighborhood bounded points and the bounded point of the current patch to the haptic endpoint are compared,

the shortest of which is kept. If the point with shortest distance is on a neighborhood patch, the local closest point search moves to that patch. If it is on the current patch, the point is kept as the local closest point.

To determine if forces should be applied, the haptic endpoint must be found to be in contact with the model. This test consists of comparing the dot product of the penetration vector and surface normal to zero. The penetration vector is the vector from the previous to the current haptic endpoint. The surface normal is the normal to the surface at the local closest point. If the local closest point lies on an edge or vertex, the normals from all sharing patches are averaged as the surface normal. Positive dot products produce forces while negative ones do not as shown in Figure 3.

Force calculations are dependent on the penetration vector. Forces are applied from the proxy position (the local or bounded closest point from the previous step) in the direction and proportional magnitude of the penetration vector.

Haptically Driven Filtering

The haptic interface is a natural three dimensional interface. The freedom of movement allowed by the interface device is intuitive, enhancing interaction with the model. The advantages of haptic rendering lend themselves to the manipulation of height field models.

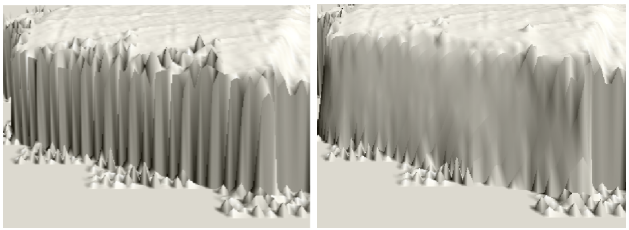

Figure 4. DATASET BEFORE (LEFT) AND AFTER (RIGHT) HAPTICALLY GUIDED LOCAL SMOOTHING.

Haptically driven smoothing allows the user to select the location and control the amount of smoothing. The haptic endpoint is used as a smoothing tool, the areas of the model touched by the haptic endpoint are smoothed based on the amount of pressure the user exerts on the model. The results of using the haptic filtering tool can be seen in Figure 4. While the steep edges of the model and the noise at the very top of the edge is smoothed, the noise on the floor of the model is left because it will be filtered out during the global filtering stage. Only the noise that causes the haptic forces during edge selection to be erratic must be filtered during this step.

The implementation of the haptically driven filtering is an add-on to the haptic contact model. When in the local filtering mode, applying forces implies filtering. The calculated strength of the force feedback vector is used to linearly blend between the calculated results of two smoothing filters, a box filter and a Gaussian filter.

1	1	1
1	1	1
1	1	1

1	2	1
2	4	2
1	2	1

Figure 5. FILTER MASKS OF THE BOX (LEFT) AND GAUSSIAN (RIGHT).

To filter the data, the lower left data value of the bilinear patch containing the proxy point is selected. The eight data values surrounding the chosen point are use to fill the filtering window. Weights, determined by the filter used, are applied to each cell of the filtering window. The final value of the filter is calculated by summing the weighted data values and dividing by the sum of the weights of the filtering mask.

Three filters are used in this system, a box filter, a median filter, and a Gaussian filter. The masks of the Gaussian and box filters can be seen in Figure 5. These filters are both averaging or low pass filters which replace the value of the center data point with the average value of the neighborhood [16]. The box filter is a uniform averaging filter in which all values in the neighborhood receive the same weight. The Gaussian filter gives more weight to the center data value, and the least weight to the data values diagonal from the center cell since these cells have the greatest distance from the center data value. Both of these filters perform similarly, however the Gaussian filter tends to maintain some noise since the center pixel is weighted most heavily. The median filter is a filtering strategy that chooses the median value of the filter window as the resulting value. This filter is more likely to preserve edges than the averaging filters. Figure 6 shows the original dataset and a single, global iteration of each of the filters.

Edge Selection

Once the high frequency noise is locally smoothed using the haptic smoothing tool, the user can use the haptic device to mark

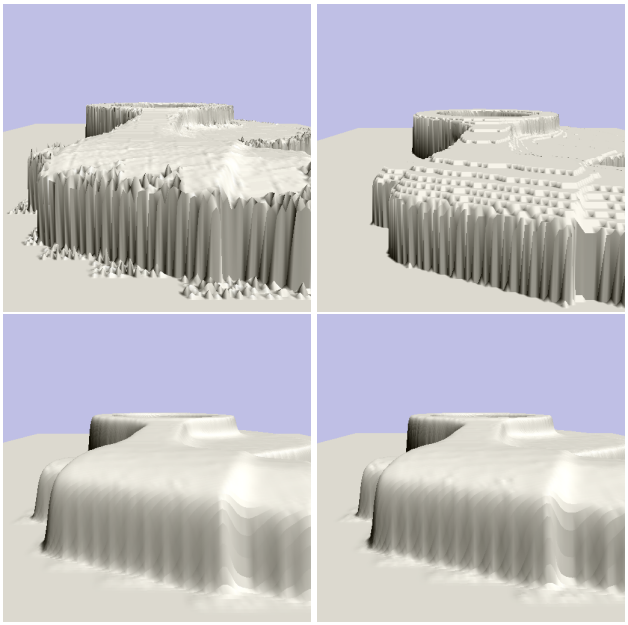

Figure 6. THE ORIGINAL DATASET (TOP LEFT) AND A SINGLE, GLOBAL ITERATION OF THE MEDIAN (TOP RIGHT), AVERAGE (BOTTOM LEFT) AND GAUSSIAN (BOTTOM RIGHT) FILTERS.

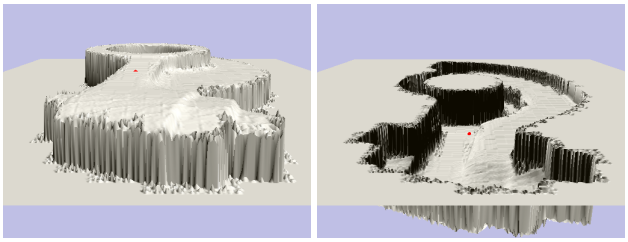

Figure 7. THE HEIGHTS OF THE DATA POINTS CAN BE INVERTED TO EASE IN CONVEX EDGE SELECTION.

edges and other areas that should remain untouched by the global filtering step. The local filtering enables the user to easily select edges without the disturbance of high frequency noise. In addition, an option to invert the height is included to help the user select convex edges. The inverted height option is a simple way to constrain the haptic endpoint to a concave corner, making the selection of convex edges easier. Figure 7 shows the results of the height inversion option.

Global Filtering

The final step in noise removal is the global filtering of the dataset. The box, Gaussian and median filters are used in the same manner as described above, however the entire dataset is modified rather than only user selected regions. To preserve edges, areas marked by the user in the edge selection stage are not filtered. Figure 8 demonstrates the results of marking and edge path and globally filtering using each of the filters described above. While the edges are not as smooth as the edges filtered globally, they keep their sharp edge. Interestingly, the edge selection and local filtering removed the step artifacts resulting from the median filter.

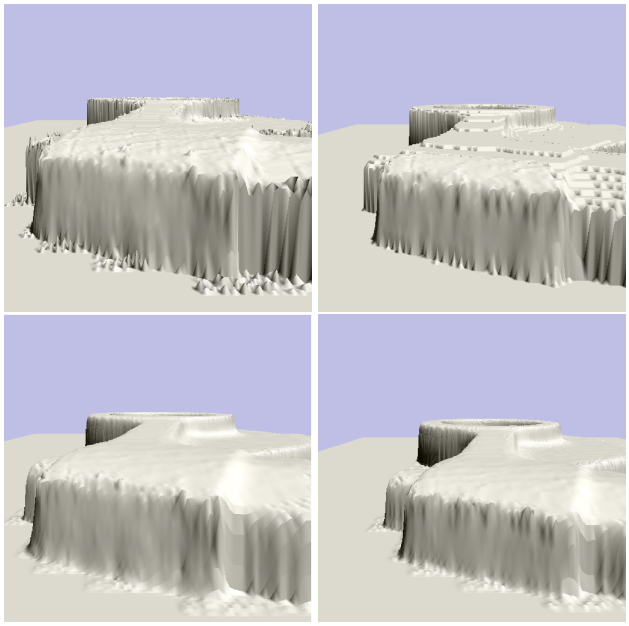

Figure 8. THE DATASET AFTER LOCAL FILTERING AND EDGE SELECTION WITHOUT GLOBAL FILTERING (TOP LEFT), WITH MEDIAN FILTERING (TOP RIGHT), AVERAGE FILTERING (BOTTOM LEFT) AND GAUSSIAN FILTERING (BOTTOM RIGHT).

RESULTS

The results of haptic local filtering, edge selection and global filtering or HEG process, can be seen in Figure 8. In comparison to global filtering without edge selection as seen in Figure 6, the HEG datasets are less smooth. However, the HEG models have sharper edges and will recover a better quality surface. The median filter performed the best edge preservation in comparison to the other filters, however the step artifacts reveal that the best use of the median filter is in areas of high discontinuity. Overall, the addition of the HEG filtering process quickly improves the quality of the dataset by maintaining sharp edges.

The original haptic system performs at update rates greater than needed for haptic rendering. The addition of the haptically driven filtering slows the update slightly, however the rates re-

Method	Update Rate
Without Filtering	1002 kHz
With Filtering	983 kHz

Table 1. HAPTIC UPDATE RATES WITH AND WITHOUT LOCAL FILTERING.

main at above haptic speed. A comparison of the update rates with and without local filtering can be seen in Table 1. The largest cost of the haptically driven local filtering is the visual update of the modified data. The visual and haptic displays are separated into two distinct threads. Haptic modifications of the data trigger the regeneration of openGL display lists which are used to visually display large scale models at interactive rates. This causes a pause each time the display list is regenerated, however the display rates immediately return to interactive. The global filtering however is not an interactive step, and thus both visual and haptic updates are paused while the global filtering takes place.

CONCLUSION

The haptically assisted filtering system presented here provides the user with a tool to modify and extract a good quality surface from a scanned dataset. In addition to haptically exploring the height field dataset, multiple filtering modes allow the manipulation of the dataset to fit the user's needs.

The current implementation is structured around high frequency noise removal while maintaining the edge features of scanned data. Depending on the characteristics of the dataset and noise, other filtering techniques may be useful. Also, additional tools for more advanced surface manipulation can easily be embedded.

ACKNOWLEDGMENT

This work was supported in part by ARO DAAD19-01-1-0013 and NSF DMI-9978603. All opinions, findings, conclusions or recommendations expressed in this document are those of the author and do not necessarily reflect the views of the sponsoring agencies.

REFERENCES

[1] Diego C. Ruspini, Krasimir Kolarov, and Oussama Khatib. The haptic display of complex graphical environments. In *SIGGRAPH 97 Conference Proceedings*, pages 345–352, 1997.

[2] C.B. Zilles and J. Kenneth Salisbury. A constraint-based god-object method for haptic display. In *IEEE/RSJ International Conference on Intellegent Robots and Systems, Human Robot Interaction, and Cooperative Robots*, volume 3, pages 146–151, 1995.

[3] C.-H. Ho, C. Basdogan, and M. A Srinivasan. Efficient point-based rendering techniques for haptic display of virtual objects. *Presence*, 8(5):477–491, 1999.

[4] Sean P. Walker and J. Kenneth Salisbury. Large haptic topographic maps:marsview and the proxy graph algorithm. In *Proceedings of the 2003 symposium on Interactive 3D graphics*, pages 83–92, 2003.

[5] Kristin Potter, David Johnson, and Elaine Cohen. Height field haptics. In *TO APPEAR: 12th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*.

[6] D. Johnson and P. Willemsen. Six degree-of-freedon haptic rendering of complex polygonal models. In *Haptics Symposium 2003*. IEEE, March 2003.

[7] C. Basdogan, C.-H. Ho, and M.A. Srinivasan. A ray-based haptic rendering technique for displaying shape and texture of 3d objects in virtual environments. In *Proc. ASME Dynamic Systems and Control Division, DSC*, volume 61, pages 77 – 84, 1997.

[8] David Johnson, Thomas V. Thompson II, Matthew Kaplan, Donald D. Nelson, and Elaine Cohen. Painting textures with a haptic interface. In *VR*, pages 282–285, 1999.

[9] Arthur D. Gregory, Stephen A. Ehmann, and Ming C. Lin. inTouch: Interactive multiresolution modeling and 3d painting with a haptic interface. In *VR*, pages 45–, 2000.

[10] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 26(2):71–78, 1992.

[11] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Mesh optimization. *Computer Graphics*, 27(Annual Conference Series):19–26, 1993.

[12] Marco Attene, Bianca Falcidieno, Jarek Rossignac, and Michela Spagnuolo. Edge-sharpener: Recovering sharp features in triangulations of non-adaptively re-meshed surfaces.

[13] T DeRose, M. Kass, and T. Truong. Subdivision surfaces in character animation. *Computer Graphics*, 32(Annual Conference Series), 1998.

[14] Huong Quynh Dinh, Greg Turk, and Greg Slabaugh. Reconstructing surfaces using anisotropic basis functions. In *International Conference on Computer Vision (ICCV) 2001*, pages 606–613.

[15] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher. Geometric surface smoothing via anisotropic diffusion of normals, 2002.

[16] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, 2002.