



Efficient High-Dimensional, Edge-Aware Filtering

Eduardo Simões Lopes Gastal

Federal University of Rio Grande do Sul

The word “filter” has origins in the use of felt to remove impurities from liquids. While filters are still important for brewing our morning coffee, the advent of computers and electronic systems has added new meaning to the word. In signal processing, the classic characterization of a filter is a system that selectively removes, scales, and/or amplifies a subset of frequencies from a signal. A band-pass filter, for example, isolates a band of frequencies by removing all others (as in telephony systems that isolate human speech as a superposition of sound waves oscillating between 300 and 3,400 Hz). Altogether, filters have widespread use in science and engineering, from telecommunications to chemistry to astronomy.

Filters are also central to many computer graphics applications. Antialiasing filters eliminate (some) unwanted artifacts in the image synthesis process by, intuitively, removing frequencies that are not representable in our finite and discrete computer monitors. In fact, the discrete representation of naturally continuous phenomena is a fundamental concept deeply related to filtering. This is why filters often appear as pixel operators in many image and video processing tasks, such as resizing and compression.

One inherent attribute of classic filters is that they are agnostic to the signal being filtered; they are linear and invariant,¹ completely defined by their response to an impulse (see Figure 1). In

practice, this is a problem for a vast number of image and video processing applications because linear and invariant filters often introduce visual artifacts around the edges of the objects in an image. Examples of this include the Gaussian filter, which removes noise but blurs edges, and enhancement filters, which increase detail but introduce halos around objects, as Figure 2 illustrates. These artifacts arise because such filters are agnostic to the concept of objects and the transition between objects (edges). Indeed, it is our visual system and brain that detect and ascribe significance to those elements. Ultimately, edges are indispensable to our understanding of the natural world through visual sensation, and image filters should take them into account in order to produce high-quality, artifact-free, beautifully formed images.

Filters that are edge-aware thus became fundamental building blocks for many applications, enabling high-quality state-of-the-art results in tasks such as upsampling, stylization, spatiotemporal filtering, detail manipulation, tone mapping, and recoloring. An edge-aware filter is nonlinear in the sense that its impulse response varies in space (or time) in a way that takes into account the underlying structure of the signal being filtered. Thus, edge-aware filters are not completely characterized by their frequency responses only, which complicates their design, analysis, and implementation.

My dissertation focused on efficient algorithms for performing edge-aware image and video filtering. The techniques we introduced leverage the known relationship between edge-aware filtering and high-dimensional spaces and are all capable of achieving real-time performance in high-resolution data.²⁻⁴ This is thanks to the optimal computational complexity of our algorithms: linear in both the number of pixels and the number of dimensions of the space in which the filters operate. This prop-

Editor's Note

Eduardo Simões Lopes Gastal received the 2016 ACM SIGGRAPH Outstanding Doctoral Dissertation Award and first place in the 2016 Brazilian Computer Society's Doctoral Dissertation Contest.

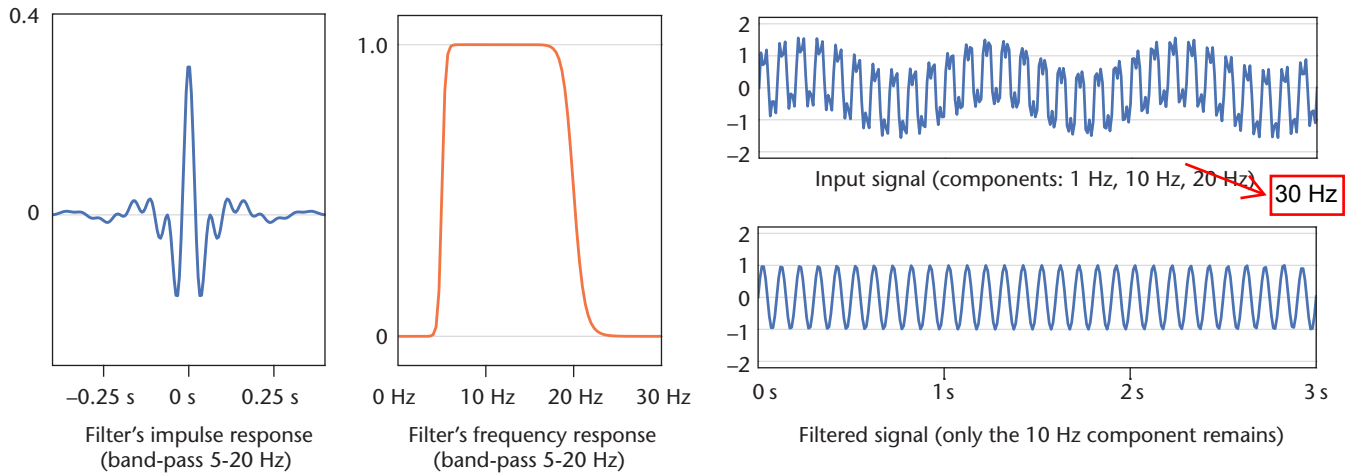


Figure 1. Classic filters. A filter is defined by its impulse response (also called the kernel) or, equivalently, by its frequency response. Filters are applied to a signal through convolution in the primal domain (time, space, and so on) or by point-wise multiplication in the frequency domain.

erty is especially important in light of the ever-increasing resolution of digitally captured images and because some applications involve spaces containing hundreds of dimensions (such as non-local-means denoising).

This article discusses some of the central ideas behind edge-aware and high-dimensional filtering and presents an intuitive overview of our algorithms and their properties. Figure 3 illustrates several applications of our filters.

High-Dimensional Filtering

Most edge-aware techniques perform filtering through an explicit or implicit linear combination of the colors in the input pixels. The most popular edge-aware operator in this class is arguably the bilateral filter,⁵ which works by weight averaging the pixel colors based on their Euclidean distances in 5D space (x, y, r, g, b). The first two dimensions of this space represent the pixels' positions in the 2D image lattice, and the last three dimensions represent their colors in the 3D RGB cube. Other color spaces may be used, such as the perceptual CIELAB, but the fundamental idea is the same: pixels with significantly different colors will be far apart in the 5D space (even if they are close together in the 2D image plane), and as such, the bilateral filter is designed to not mix their colors. This occurs since the weights used for the linear combinations are selected to be inversely proportional to the high-dimensional 5D distance between pixels, ultimately leading to an edge-preserving smoothing filter; small variations in the image are removed, but well-defined color discontinuities are preserved (see Figure 3f).

The relationship between edge-aware filtering and high-dimensional spaces is made rigorous by the geometric interpretation of images as em-

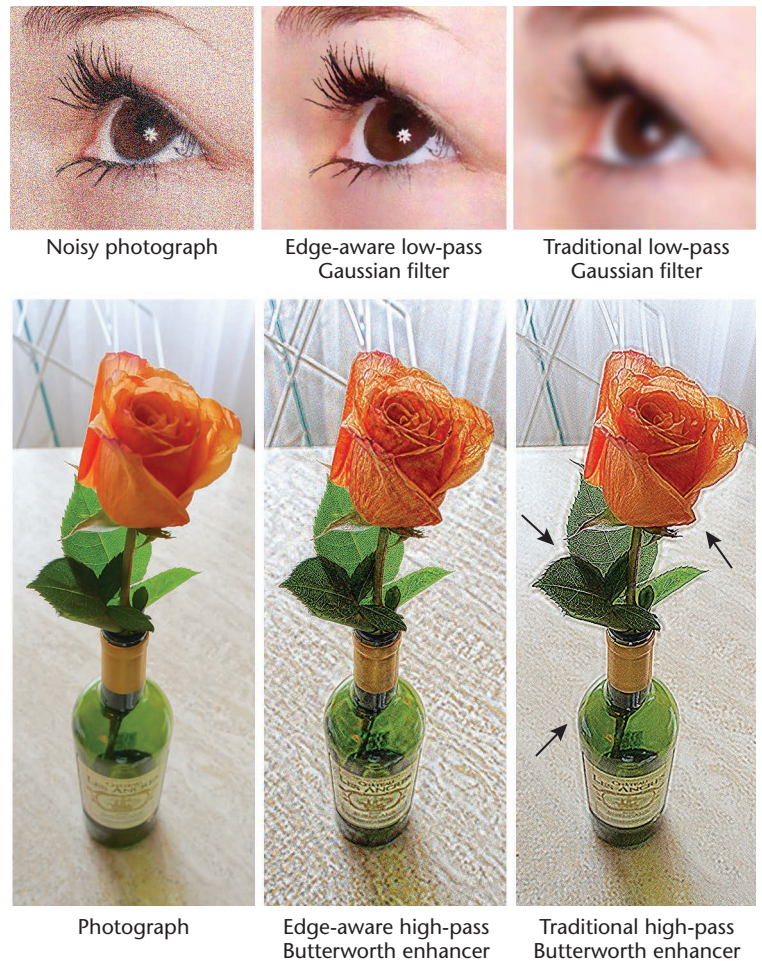


Figure 2. Denoising and detail enhancement. These are ideal tasks for edge-aware filters because traditional linear and invariant operators introduce artifacts such as blurring and halos (black arrows).

bedded surfaces.⁶ Figure 4 illustrates this point, showing a grayscale image visualized as a 2D surface embedded in 3D space. Note how the pixels

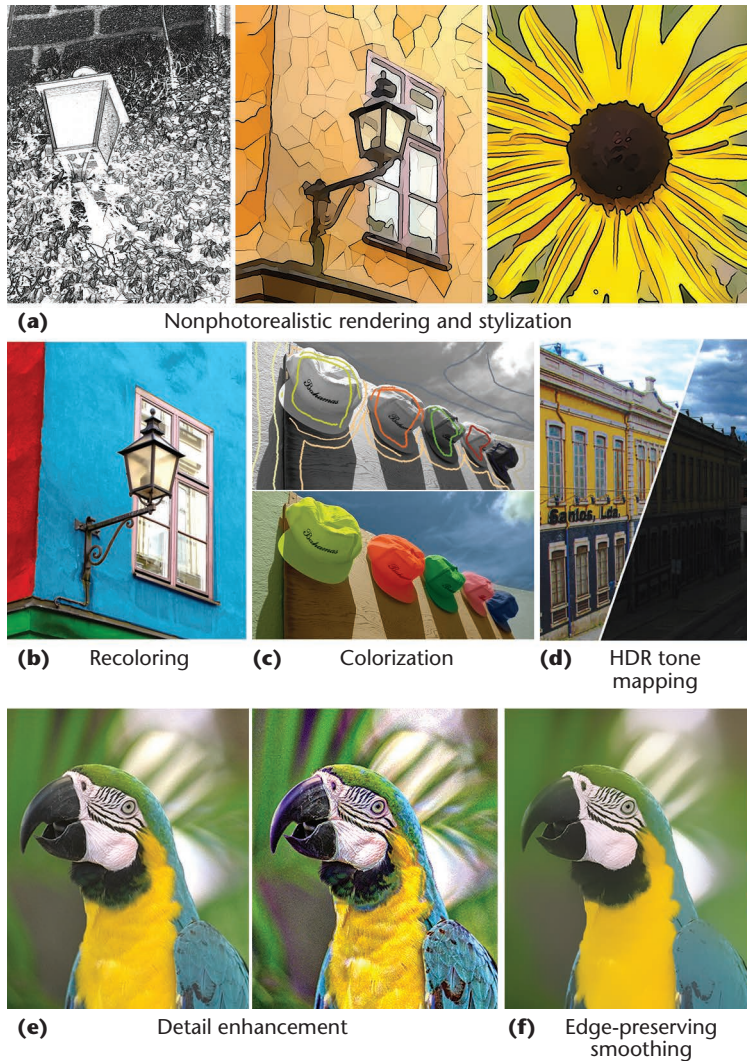


Figure 3. Real-time applications of our high-dimensional and edge-aware filters: (a) nonphotorealistic rendering and stylization, (b) recoloring, (c) colorization, (d) high dynamic range (HDR) tone mapping, (e) detail enhancement, and (f) edge-preserving smoothing.

marked by white circles in Figure 4a sit on top of the surface in Figure 4b and how well-defined edges in the image induce discontinuities in the surface. This is just one example of how to embed an image into an edge-aware high-dimensional space; other examples include the use of per-pixel infrared⁷ or neighborhood⁸ information for additional dimensionality (up to the hundreds), color videos seen as 3D manifolds (generalized surfaces) embedded in 6D space, and texture seen as 4D manifolds in 6D space.⁶ The take-home message is this: any filtering process that accounts for the high-dimensional distances between pixels will inevitably be edge-aware.

Euclidean versus Geodesic Filters

Figures 4b and 4c also show that there are two distinct approaches for measuring distances in the high-dimensional space, which leads to the categorization of edge-aware filters as either Euclidean or geodesic. The main difference between the two groups is the filter behavior near strong discontinuities (edges) in the image. Note how the geodesic distance between the two bottom pixels in Figure 4c, indicated by white circles, is much larger than their Euclidean distance in Figure 4b. This is mainly because these two pixels lie on dark gray areas of the image and between them lies a vertical band of light gray pixels. Because the geodesic distance respects the surface’s structure, it is forced to “go over the hill” when connecting the two pixels. The Euclidean metric does not know about this underlying structure, so it simply connects the two pixels with a straight line in 3D space, avoiding the “hill” and computing a smaller distance. In the end, both metrics result in effective edge-

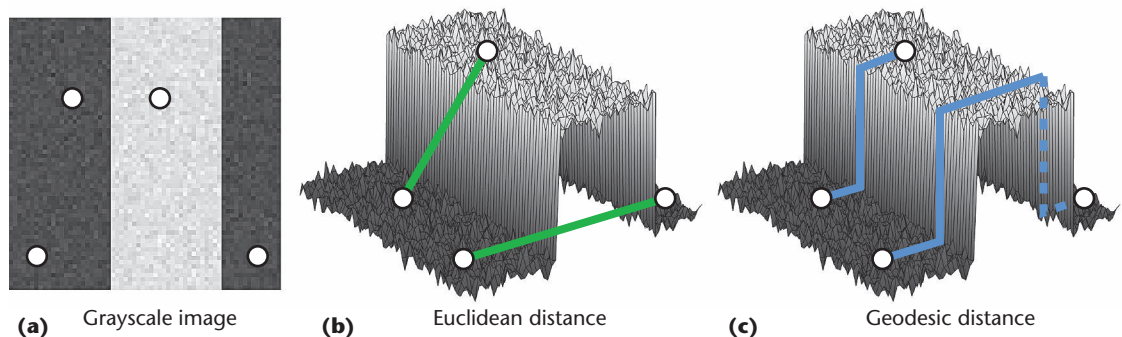


Figure 4. Embedding a 2D image into an edge-aware high-dimensional space. (a) The grayscale image can be interpreted as a surface embedded in 3D space—a height field. (b) The Euclidean distance is the smallest distance in 3D space: the length of a straight line (shown in green) between the high-dimensional points. (c) The geodesic distance is the length of a path (commonly the shortest) on top of the surface (shown in blue). For clarity, this illustration omits the small variations present in the blue line due to the noisy surface. Each distance metric provides best results for different types of applications.

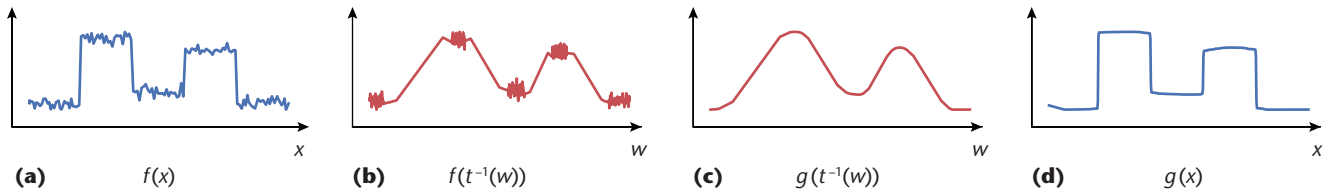


Figure 5. Edge-preserving smoothing using the domain transform: (a) input signal $f(x)$, (b) input signal after the warping defined by the domain transform $w = t(x)$, (c) signal after warping and filtering with a low-pass Gaussian filter, and (d) output signal $g(x)$ obtained after unwarping.

aware filters, but each provides a different filtering response and is best suited for particular types of applications. For example, the Euclidean response is better suited for recoloring disjoint elements in an image,⁹ whereas the geodesic response is best for adding color to grayscale images.¹⁰

The standard example of the use of the Euclidean metric is the bilateral filter I described earlier. For the geodesic metric, a good example is the Perona-Malik anisotropic diffusion.¹¹ These approaches have wide applicability in image processing and computer graphics, but they suffer from a high computational cost.¹² Their direct implementation is too slow for many practical uses, especially real-time computer graphics. As a result, several techniques have been proposed that either try to accelerate anisotropic diffusion or bilateral filtering or that introduce alternative ways of performing similar edge-aware filtering operations on images and videos. Although they clearly and significantly improve performance, at the time that I started my doctoral research the available solutions had various limitations, including only handling grayscale images,¹³ not being sufficiently fast for real-time applications,¹⁴ restricting filtering to certain scales,¹⁵ or possibly introducing artifacts by not using true high-dimensional distances.¹⁶

The goal of my dissertation was to address the main limitations of previous techniques as well as expand the spectrum of existing edge-aware filters, providing a simple and natural way to experiment with the design and composition of new filters. We succeeded in these tasks, introducing two novel approaches for efficient high-dimensional filtering: the domain transform for geodesic response and the adaptive manifolds for Euclidean response. With these frameworks, we proposed several edge-aware filters that addressed the main limitations of previous techniques, in addition to providing the fastest performance (on both CPUs and GPUs) for various real-world applications.

Domain Transform for Geodesic Filtering

The first part of my dissertation described the domain transform,² which is used to efficiently perform geodesic image and video filtering. The insight

behind this method is the use of a simple dimensionality reduction strategy: our idea was to map curves from the high-dimensional image surface to a new lower-dimensional 1D space in a way that preserves the geodesic distance between points on each curve. For this, we derived an isometric (distance-preserving) domain transformation $t(x)$ that maps image-space pixel coordinates to this new 1D space and takes into account user-provided filter parameters. Intuitively, the domain transform adaptively warps the input signal so that geodesic filtering can be efficiently performed in linear time in both the number of pixels and dimensionality of the original high-dimensional space.

Figure 5 illustrates the use of the domain transform in a simple 1D edge-preserving smoothing example. Note how small variations in the signal are removed, but sharp edges are preserved. In practice, there is no need to explicitly warp and unwarped the signal as shown in the illustration, and the entire operation is performed on the fly in a single step.

In our original publication,² we presented all the necessary steps for the derivation of $t(x)$, along with mathematical proofs of its properties. We also showed how to produce high-quality edge-aware image and video filtering by iterating axis-aligned 1D domain transforms. However, to produce artifact-free results, we must pay attention to how the filter is decomposed. For low-pass filters, we showed that one should cut in half the standard deviation of the kernel at each iteration, and two to three iterations are enough for convergence. All the application examples in Figures 3a–3d were computed using our domain transform.

Using the Domain Transform with Arbitrary Recursive Filters

One remarkable observation is that any classic (linear and invariant) filter can be made edge-aware by, before its evaluation, adaptively warping the input signal using the domain transform. In such a warped domain, the discrete pixels of an image or video become nonuniformly spaced. Filtering such a signal is challenging because the varying distance between samples must be taken

into account in order to compute a correct edge-aware result. The main difficulty is that standard algorithms for filtering—such as fast Fourier transforms (FFT), convolutions, and recursive filters—are not formulated with nonuniform sampling in mind.

One naive solution for filtering nonuniformly sampled signals is to perform a sample-rate conversion to bring the sampling rate to a constant value, followed by the computation with one of the standard filtering algorithms. However, this is impractical because it introduces severe overhead to the filtering process; depending on the distances between samples, sample-rate conversion to a constant rate could require large amounts of memory and time or introduce sampling errors. Another solution would be the nonuniform extension of the FFT,¹⁷ but its performance is still superlinear, $O(N \log N)$, in the number of pixels, and its implementation is somewhat complex.

During the initial development of my dissertation, we were able to solve the problem of filtering nonuniformly sampled signals in a restricted setting. We demonstrated efficient linear-time algorithms only for computing two simple filters in this warped, nonuniform domain: an iterated box filter and a recursive first-order decaying-exponential filter. These are both low-pass filters, useful to many image and video processing tasks.

Nevertheless, the full spectrum of digital filters available in the literature and software packages was still out of reach for efficient edge-aware filtering. For example, we could not accurately and efficiently filter a nonuniformly sampled signal with a Gaussian—a fundamental operator often considered the ideal smoothing filter because of its optimal locality in the time-frequency domain.¹⁸ (That is, the Gaussian does not introduce aliasing because of its frequency locality, and it does not introduce ringing because of its spatial locality and lack of negative lobes.)

We addressed this limitation a few years after developing the domain transform using a discrete-time mathematical formulation for applying any-order recursive digital filters to nonuniformly sampled signals.⁴ We used recursive filters because they have several advantages when compared with other filtering methods based on brute-force convolution, summed-area tables, and FFTs. For instance, they have linear-time complexity in the number of input samples, infinite impulse responses, and a relatively straightforward implementation. They can also approximate important filters such as the Gaussian and Gabor bases.

Together with the domain transform, our formu-

lation enabled, for the first time, a geodesic edge-aware evaluation of arbitrary recursive infinite impulse response digital filters (not only low-pass), allowing for practically unlimited control over the shape of the filtering kernel. Figure 6 shows a variety of filters we were able to design for the specific task of edge-aware detail manipulation, including Gaussian, Laplacian of Gaussian, and band-pass and high-pass Butterworth filters. Although previous approaches had performed edge-aware detail manipulation,¹⁹ they commonly worked by computing differences between the outputs of a fixed type of low-pass filter. In contrast, our approach allows the direct edge-aware computation of any band-pass and high-pass filter, all with real-time performance.

My dissertation discusses the desirable features of our discrete-time formulation, including that it preserves the stability of the original filters and is well-conditioned for low-pass, high-pass, and band-pass filters alike. Its cost is also linear in the number of samples and is not affected by the size of the filter support. Furthermore, the proposed method is general, working with any nonuniformly sampled signal and any recursive digital filter defined by a difference equation (that is, not necessarily edge-aware operations), as Figure 7 shows. Because our formulation uses the filter coefficients directly, it works “out of the box” with existing methodologies for digital filter design. For example, both MATLAB and the open source SciPy library provide routines that compute the coefficients of well-known filters such as Butterworth, Chebyshev, and Cauer. By providing a simple and natural way to experiment with the design and composition of new edge-aware digital filters, our method enables a great variety of new image and video effects.

Adaptive Manifolds for Euclidean Filtering

The second part of my dissertation described the adaptive manifolds³ used to efficiently perform Euclidean high-dimensional Gaussian image and video filtering. This operator is also known as a Gauss transform or bilateral filter, and it is intuitively a high-dimensional convolution of the image manifold (see Figure 4) with a Gaussian function.¹² As I mentioned earlier, naive implementations of such an important filter are too slow for many practical uses. With the adaptive manifolds, we demonstrated an efficient algorithm for Euclidean high-dimensional filtering, being the first with linear complexity in both the number of pixels and in the dimensionality of the space in which the filter operates.

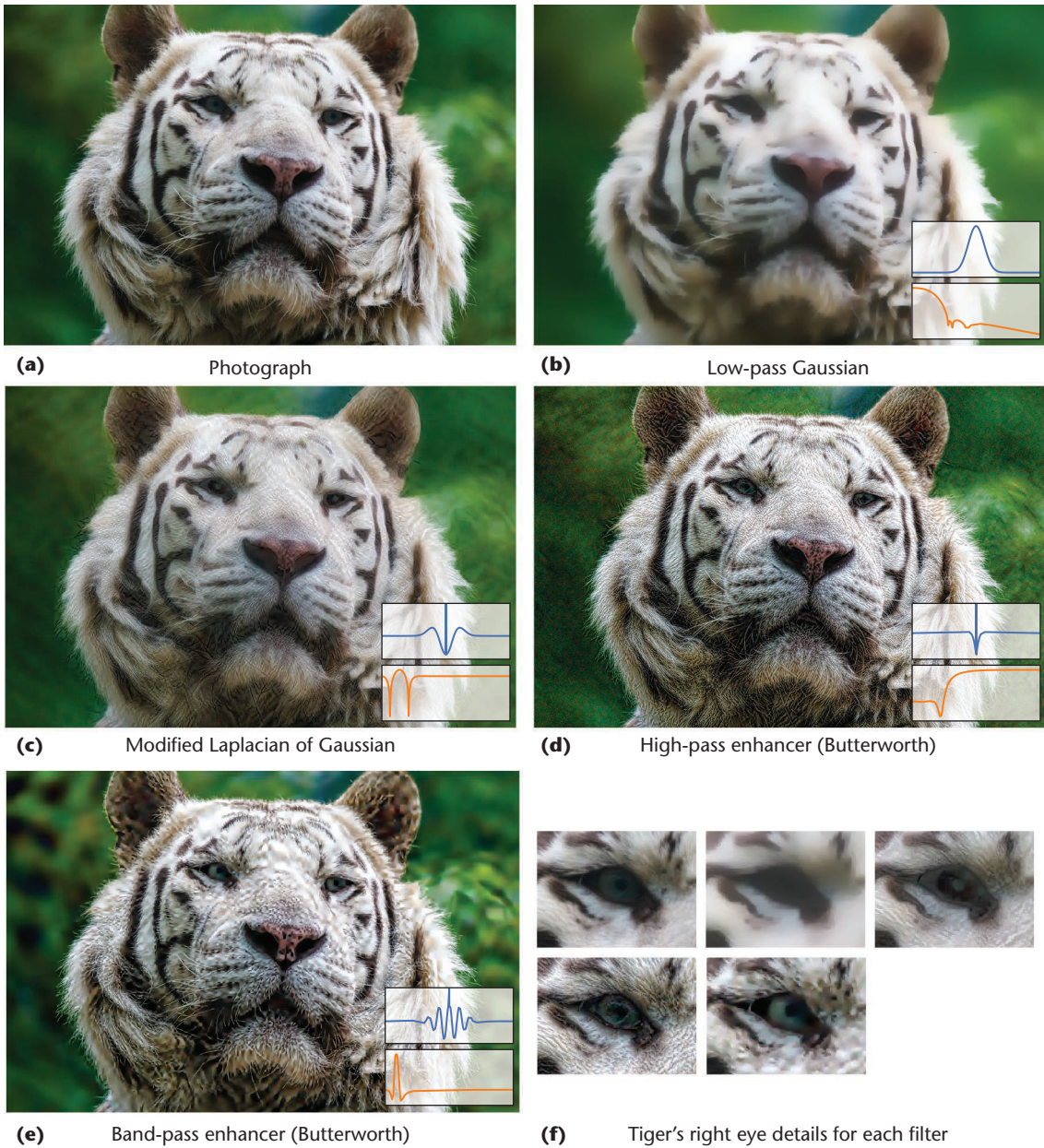


Figure 6: High-order recursive filters applied using our method: (a) initial photograph, (b) low-pass Gaussian, (c) modified Laplacian of Gaussian, (d) high-pass enhancer (Butterworth), and (e) band-pass enhancer (Butterworth). For these examples, nonuniform sampling positions are computed using our edge-aware domain transform. Thus, the resulting filters preserve the image structure and do not introduce visual artifacts such as halos around objects. The graphs in the insets show the filters' impulse response in blue and their frequency response (Bode magnitude plot) in orange. (f) The close-ups of the tiger's right eye show the detail for each filter, (a)–(e), respectively.

Before giving an overview of our algorithm, it is important to understand why Euclidean filters are fundamentally more difficult to compute than geodesic ones, especially on high-dimensional spaces. Note that convolution with a Gaussian implements a diffusion process—information is being propagated from each pixel to all others based on their distances in the high-dimensional space. In geodesic diffusion, information must flow through pathways on top of the image manifold, such as

the ones defined by the blue lines in Figure 4c. This means that the flow from one pixel to another must pass by all pixels on the path that connects them, a process which is simple to compute using iterative methods¹¹ or our domain transform. In fact, the domain transform implements geodesic diffusion efficiently because it reuses intermediate computations, similarly to how a box filter may be efficiently computed in constant time per pixel using a summed area table or moving average.²⁰

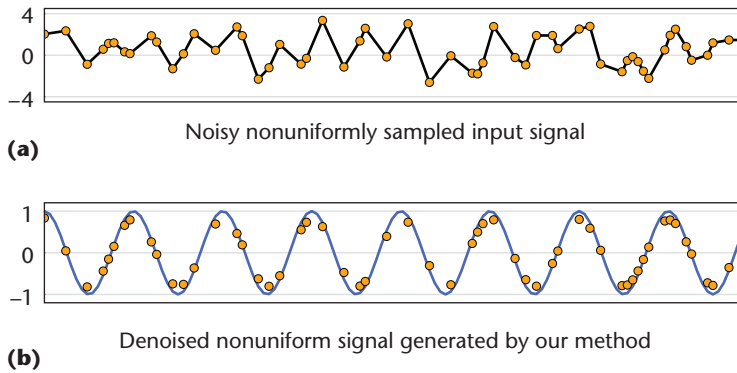


Figure 7. General approach to filter design. (a) A noisy nonuniformly sampled sinusoid can be filtered by an eighth-order band-pass Butterworth filter using our approach. (b) The filtered samples are superimposed on the original noiseless signal (in blue).

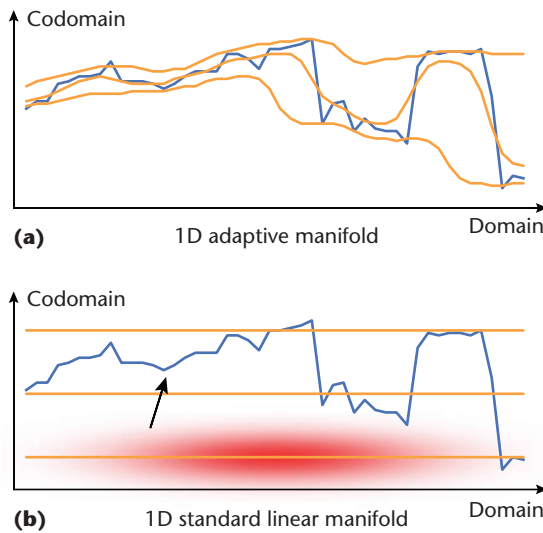


Figure 8. Adaptive sampling versus standard linear sampling, applied to a 1D signal (in blue). (a) Our adaptive manifolds adjust themselves to the signal in high-dimensional space, creating pathways to exchange information among pixels with similar colors. (b) Standard linear quantization under-represents certain regions of the signal (see black arrow) and performs unnecessary work in other regions (in red).

In Euclidean diffusion, however, information is free to flow unrestricted throughout space. The green lines in Figure 4b exemplify the Euclidean pathways for high-dimensional information flow for two pairs of pixels. Note how the paths do not follow the image manifold’s structure, which means that it is not directly possible to reuse intermediate computations. Luckily, a common strategy exists that enables the reuse of intermediate computations and improves the performance of Euclidean diffusion: sampling the high-dimensional space at a specially designed set containing M sampling points and using them to

interpolate the filter’s response at all N original input pixels.

This approach shows performance gains when M is much smaller than N^{21} and/or the M sampling points are distributed in a structured way on top of some flats (generalized planes) embedded in the high-dimensional space,^{13,14} which we call a structured set of points. In this case, using classic linear-time filtering approaches and the separability of the Gaussian function, the filter’s response for the M sampling points can be computed in $O(dN + dM)$ time instead of $O(dNM)$, where d is the dimensionality of the space.

Previous approaches built a structured set of points by distributing them on top of oriented flats, either axis-aligned^{13,21} or with a few discrete orientations.¹⁴ These flats define a tessellation of the space into cells given by hyperrectangles or simplices, respectively. To compute the filter’s response for the original pixels, we then perform multilinear or barycentric interpolation. Because the signal being filtered is rarely a linear manifold, as the dimensionality of the space increases, we need more flats to enclose the original pixels into cells (required for interpolation). Furthermore, many of the defined cells will not contain any pixels, and thus any work performed on them is wasted. This can be avoided using data structures such as hash tables,¹⁴ but this makes the algorithm’s cost quadratic in the dimensionality d , and its implementation less parallelizable, especially on GPUs.

The distinctive characteristic of our adaptive-manifold approach is that it computes a structured set of sampling points adapted to the signal. This is done by distributing the sampling points on top of nonlinear manifolds defined specifically to each signal and constructed considering the standard deviations of the high-dimensional Gaussian. To obtain a good approximation for Euclidean diffusion, my dissertation shows that it is sufficient for such manifolds to be only approximately linear in all local neighborhoods—that is, the manifolds should be smooth surfaces, as Figure 8 illustrates, and the degree of smoothness is automatically computed from the parameters of the filter.

Euclidean filtering using the adaptive manifolds becomes an efficient operation because the Gaussian response for all sampling points lying on top of the manifolds may be computed using our domain transform. In other words, we perform high-dimensional Euclidean diffusion through linear-time geodesic steps on top of each nonlinear manifold. Furthermore, the final response for all pixels is computed using a normalized convolution interpolator,²² which does not require en-

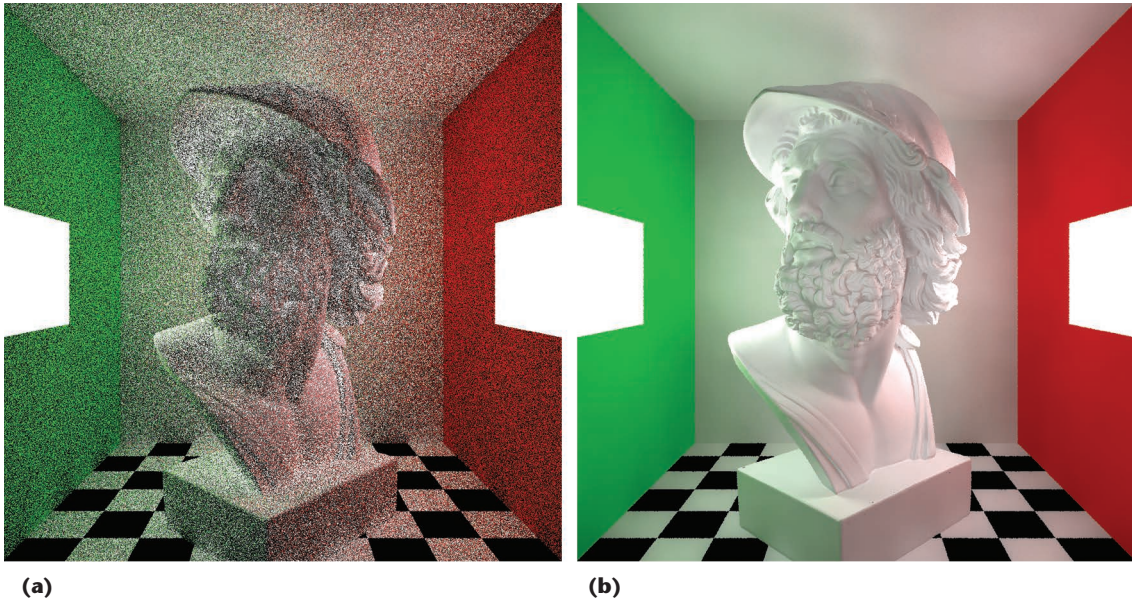


Figure 9. Example of Euclidean filtering of a noisy undersampled image generated using path tracing: (a) noisy undersampled rendering (one sample per pixel for indirect illumination) and (b) indirect illumination denoised using our adaptive manifolds in 8D geometric space. Note the smoothness of the shading.

closing pixels into cells to perform multilinear or barycentric interpolation. This is also a key factor for our method’s performance.

In the dissertation, we present a derivation for the equations that define Euclidean filtering using our adaptive manifolds, providing a theoretical justification for the technique and for its properties. The denoising example in Figure 2 was computed with our adaptive manifolds in 27D non-local-means space, where the coordinates of each pixel is given by a vector containing the colors of all its neighbors in a 3×3 window. Figure 9 shows a similar example for an image-synthesis task, where the 8D space considers geometric information (3D normals, 3D world positions, and 2D imagespace coordinates).

Past, Present, and Future

We are happy to see that our filters and algorithms have been well received by the image and video processing, computer graphics, computer vision, and computational photography communities. A good indication of this interest is the fact that third-party implementations of the domain transform and adaptive manifolds have been included in the open source computer vision library (OpenCV), in addition to other implementations available throughout the Web. The official implementations of our algorithms may be found on the homepages of our three publications (<http://inf.ufrgs.br/~eslgastal>). We also suggest that readers watch the videos available on these homepages for intuitive descriptions and several examples of applications of the developed techniques.

One of the first uses of our domain transform was by Manuel Lang and his colleagues,²³ where they employed and extended our filters for the purpose of introducing temporal consistency in a variety of optimization-driven image-based problems, such as optic flow, disparity estimation, and visual saliency computation. They extended our iterative axis-aligned framework to filter video frames along the temporal motion paths of points in the scene and were able to achieve large gains in efficiency in terms of both memory and time. This allowed for the processing of “entire shots at once, taking advantage of supporting information that exists across far away frames.”

In image synthesis, our adaptive manifolds have been extended to support the filtering of an arbitrary number of samples per pixel,²⁴ as commonly generated by rendering software. A fast GPU implementation of this technique reconstructs high-quality global illumination at interactive rates and also supports important effects such as shallow depth of field.

In computational photography, our filters have been used for real-time cross-frame filtering and edge-aware upsampling in digital cameras.²⁵ These operations reduce temporal artifacts, in addition to speeding up computation, allowing for the real-time preview of several computational photography techniques directly in the camera’s viewfinder. Similarly, synthetic defocus effects may be generated directly in smartphone cameras,²⁶ where our domain transform has been used to remove blocking artifacts introduced by other techniques in the imaging pipeline.

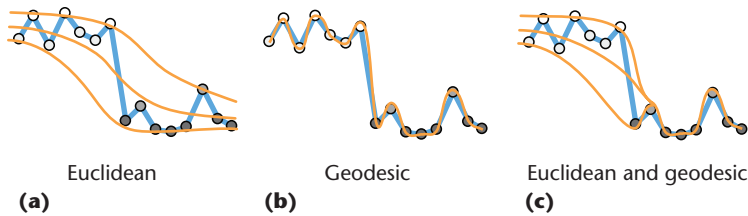


Figure 10. Adaptive manifolds allow for the definition of hybrid Euclidean-geodesic filters. (a) Our original manifolds (in orange) are defined for Euclidean filtering. (b) If the manifolds are forced to exactly follow the signal for a single dimension of the space, we obtain a hybrid Euclidean-geodesic filter. (c) We can even mix manifold types in a single dimension, thus simultaneously computing both Euclidean and geodesic diffusion.

In computer vision, our filters have been used for fast cost-aggregation in local stereo matching,²⁷ a fundamental step in estimating the disparity between two photographs of the same scene. More recently, the domain transform has been used as a layer inside a convolutional neural network (CNN), with the goal of performing dense semantic segmentation of the objects in an image.²⁸ That original work by Liang-Chieh Chen and his colleagues touches on two fundamental topics related to edge-aware filtering, which I envision will play a central role in the future of the field. First, the authors show that it is possible to back-propagate errors through the domain transform, thereby allowing for the joint training of the whole semantic-segmentation-edge-aware network (their domain transform code, using the Caffe framework, is available at <https://bitbucket.org/aquariusjay/deeplab-public-ver2>). This joint training is key because deep CNNs seem to be a good model for computation in the human visual system, and the sensing of edges is essential for the visual understanding of natural scenes. Therefore, the integration of edge-aware operators into deep learning networks is an interesting and fruitful direction for future investigation.

What is an edge, and which edges should be taken into account by the filter? This is the second fundamental topic—a subjective, and perhaps philosophical question we’ve been ignoring since the beginning of this text—which has been partially addressed by Chen and his colleagues.²⁸ Their work integrates into the domain transform an edge estimator obtained from intermediate layers of the DeepLab CNN, a network designed specifically for segmentation and trained on examples given by humans. This is a significant improvement over our original formulation of the domain transform, in which we use a local edge estimator given by the gradient magnitude functional (a popular estimator, but not robust, to noise for example). Other works also extend the

domain transform through different edge detectors,^{27,29} and a whole spectrum of possibilities are still available for future research.

Our original edge estimator results from embedding the image surface into a high-dimensional space using direct color information, as exemplified in Figure 4 for a grayscale image. However, as exemplified in the literature,^{27–29} our domain transform may be used even in situations where one does not have an explicit embedding. That is, we can work directly and efficiently with just the pairwise distances between neighboring pixels, in a pure geodesic metric space. Whether an efficient algorithm exists for the analogous pure Euclidean metric space is an open problem—our adaptive manifolds and other methods rely on high-dimensional coordinates given by an explicit embedding of the image surface.

A related fact is that our adaptive-manifold formulation allows for the definition of hybrid Euclidean-geodesic filters. To the best of our knowledge, our work is the first to even consider such a filter. The main idea is to consider distances as Euclidean for some dimensions of the space and as geodesic for others. We can push this further by performing Euclidean diffusion in a subset of a single dimension and geodesic diffusion on the complementing subset, with a smooth user-controllable transition between the two. All these operations are performed through the design of task-specific adaptive manifolds (see Figure 10), allowing for a flexible framework that we believe may allow for a variety of new applications.

Other exciting opportunities exist for research and development relating to edge-aware operators, from their efficient use in nonstructured data, such as meshes and point clouds, to their use in fundamental problems such as sampling and data representation. The perceptual nature of high-dimensional and edge-aware filters is an essential property for handling visual data, and I expect they will continue their tradition as fundamental building blocks for several new and exciting applications. We hope that our filters and algorithms persist as valuable tools for the computer graphics and scientific communities. ❏

Acknowledgments

I thank my adviser Manuel M. Oliveira. My dissertation and the writing of this work were sponsored by CAPES and CNPq-Brazil (fellowships and grants 158666/2010-0, 557814/2010-3, 142889/2010-5, 308936/2010-8, 482271/2012-4, 403591/2014-7, and 150550/2015-4).

References

1. J. G. Proakis and D. K. Manolakis, *Digital Signal Processing*, 4th ed., Pearson, 2006.
2. E.S.L. Gastal and M.M. Oliveira, "Domain Transform for Edge-Aware Image and Video Processing," *ACM Trans. Graphics*, vol. 30, no. 4, 2011, article no. 69.
3. E.S.L. Gastal and M.M. Oliveira, "Adaptive Manifolds for Real-Time High-Dimensional Filtering," *ACM Trans. Graphics*, vol. 31, no. 4, 2012, article no. 33.
4. E.S.L. Gastal and M.M. Oliveira, "High-Order Recursive Filtering of Non-uniformly Sampled Signals for Image and Video Processing," *Computer Graphics Forum*, vol. 34, no. 2, 2015, pp. 81–93.
5. C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," *Proc. 6th Int'l Conf. Computer Vision (ICCV)*, 1998, pp. 839–846.
6. N. Sochen, R. Kimmel, and R. Malladi, "A General Framework for Low Level Vision," *IEEE Trans. Image Processing*, vol. 7, no. 3, 1998, pp. 310–318.
7. S. Zhuo et al., "Enhancing Low Light Images Using Near Infrared Flash Images," *Proc. 17th IEEE Int'l Conf. Image Processing (ICIP)*, 2010, pp. 2537–2540.
8. A. Buades, B. Coll, and J. Morel, "A Non-local Algorithm for Image Denoising," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2005, pp. 60–65.
9. J. Chen, S. Paris, and F. Durand, "Real-Time Edge-Aware Image Processing with the Bilateral Grid," *ACM Trans. Graphics*, vol. 26, no. 3, 2007, article no. 103.
10. A. Levin, D. Lischinski, and Y. Weiss, "Colorization Using Optimization," *ACM Trans. Graphics*, vol. 23, no. 3, 2004, pp. 689–694.
11. P. Perona and J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, 1990, pp. 629–639.
12. S. Paris et al., "A Gentle Introduction to Bilateral Filtering and Its Applications," *ACM SIGGRAPH 2008 Classes*, ACM, 2008, article no. 1.
13. Q. Yang, K.H. Tan, and N. Ahuja, "Real-Time O(1) Bilateral Filtering," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 557–564.
14. A. Adams, J. Baek, and M.A. Davis, "Fast High-Dimensional Filtering Using the Permutohedral Lattice," *Computer Graphics Forum*, vol. 29, no. 2, 2010, pp. 753–762.
15. R. Fattal, "Edge-Avoiding Wavelets and Their Applications," *ACM Trans. Graphics*, vol. 28, no. 3, 2009, article no. 22.
16. K. He, J. Sun, and X. Tang, "Guided Image Filtering," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, 2013, pp. 1397–1409.
17. F. Marvasti, *Nonuniform Sampling: Theory and Practice*, vol. 1, Springer, 2001.
18. D. Marr and E. Hildreth, "Theory of Edge Detection," *Proc. Royal Society of London B: Biological Sciences*, vol. 207, no. 1167, 1980, pp. 187–217.
19. Z. Farbman et al., "Edge-Preserving Decompositions for Multi-scale Tone and Detail Manipulation," *ACM Trans. Graphics*, vol. 27, no. 3, 2008, article no. 67.
20. E. Dougherty, *Digital Image Processing Methods*, CRC Press, 1994.
21. S. Paris and F. Durand, "A Fast Approximation of the Bilateral Filter Using a Signal Processing Approach," *Int'l J. Computer Vision*, vol. 81, no. 1, 2009, pp. 24–52.
22. H. Knutsson and C.-F. Westin, "Normalized and Differential Convolution: Methods for Interpolation and Filtering of Incomplete and Uncertain Data," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 1993, pp. 515–523.
23. M. Lang et al., "Practical Temporal Consistency for Image-Based Graphics Applications," *ACM Trans. Graphics*, vol. 31, no. 4, 2012, article no. 34.
24. P. Bauszat et al., "Sample-Based Manifold Filtering for Interactive Global Illumination and Depth of Field," *Computer Graphics Forum*, vol. 34, no. 1, 2015, pp. 265–276.
25. J. Baek et al., "WYSIWYG Computational Photography via Viewfinder Editing," *ACM Trans. Graphics*, vol. 32, no. 6, 2013, article no. 198.
26. J.T. Barron et al., "Fast Bilateral-Space Stereo for Synthetic Defocus," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4466–4474.
27. C.C. Pham and J.W. Jeon, "Domain Transformation-Based Efficient Cost Aggregation for Local Stereo Matching," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 23, no. 7, 2013, pp. 1119–1130.
28. L.-C. Chen et al., "Semantic Image Segmentation with Task-Specific Edge Detection Using CNNs and a Discriminatively Trained Domain Transform," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4545–4554.
29. Q. Yang, "Semantic Filtering," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4517–4526.

Eduardo Simões Lopes Gastal is a postdoctoral researcher in the Institute of Informatics at the Federal University of Rio Grande do Sul (UFRGS), Brazil. His research interests include image processing, computer graphics, and computer vision. Gastal has a PhD in computer science from UFRGS. Contact him at eslgastal@inf.ufrgs.br.

Contact department editor Jim Foley at foley@cc.gatech.edu.

myCS

Read your subscriptions through the myCS publications portal at <http://mycs.computer.org>.