Image Scale-Space from the Heat Kernel

Fan Zhang and Edwin R. Hancock

Department of Computer Science,
University of York, York, YO10 5DD, UK

Abstract. In this paper, we show how the heat-kernel can be used to construct a scale-space for image smoothing and edge detection. We commence from an affinity weight matrix computed by exponentiating the difference in pixel grey-scale and distance. From the weight matrix, we compute the graph Laplacian. Information flow across this weighted graph-structure with time is captured by the heat-equation, and the solution, i.e. the heat kernel, is found by exponentiating the Laplacian eigen-system with time. Our scale-space is constructed by varying the time parameter of the heat-kernel. The larger the time the greater the the amount of information flow across the graph. The method has the effect of smoothing within regions, but does not blur region boundaries. Moreover, the boundaries do not move with time and this overcomes one of the problems with Gaussian scale-space. We illustrate the effectiveness of the method for image smoothing and edge detection.

1 Introduction

Witkin was one of the first to formalise the multi-scale descriptions of images and signals in terms of scale-space filtering [13]. The technique plays an important role in low-level computer vision. The basic idea is to use convolutions with the Gaussian kernel to generate fine to coarse resolution image descriptions. Babaud [7], Yuille [17] and Hummel [6] have analysed and further developed the method. Broadly speaking this work has shown that there is considerable information to be gained from the analysis of changes in image structure over different scales. Moreover, the study of multi-scale and multi-resolution image processing has lead the development of a diverse family of algorithms. For instance, Gidas [4] has extended Geman and Geman's [3] stochastic image restoration method to the multi-scale case using the renormalisation group to relate the processing at different scales. Spann and Wilson [16] combined the spatial and frequency domain locality to segment images using multiresolution techniques.

The formal description of Witkin's idea is as follows. From the image $\mathcal{I}_0(x, y)$ a series of coarser scale images $\mathcal{I}(x, y, \sigma)$ are generated through convolution with a Gaussian kernel $G(x, y; \sigma)$ of scale σ. The convolution is

$$\mathcal{I}(x, y, \sigma) = \mathcal{I}_0(x, y) * G(x, y; \sigma) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \mathcal{I}_0(x-u, y-v) \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}} dudv \tag{1}$$

M. Lazo and A. Sanfeliu (Eds.): CIARP 2005, LNCS 3773, pp. 181–192, 2005.

As the scale σ is increased, the resolution becomes coarser. Since the Gaussian smoothing process is linear and isotropic, it has an equal blurring effect at all image locations. Hence, while the region interiors are smoothed their boundaries are blurred. Another problem with linear scale-space is that the boundary locations move and sometimes coalesce at coarse scales. As illustrated by Perona and Malik [12], in 2-D images there is additional problem that edge junctions, which contain much of the spatial information of edge maps, are destroyed.

Hummel [6] and Koenderink [8] pointed out that the family of images derived from the Gaussian convolution operation are solutions of the heat equation

$$\frac{\partial \mathcal{I}}{\partial t} = \Delta \mathcal{I} = \frac{\partial^2 \mathcal{I}}{\partial x^2} + \frac{\partial^2 \mathcal{I}}{\partial y^2} \tag{2}$$

with the initial condition $\mathcal{I}(x, y, 0) = \mathcal{I}_0(x, y)$. Based on this observation, Koenderink [8] stated two criteria for features to generate multi-scale descriptions. The first of these is causality, whereby any feature at a coarse level of resolution is required to possess a "cause" at a finer level of resolution, although the reverse need not be true. The second criterion is that of homogeneity and isotropy. According to this requirement the scale-space blurring is required to be spatially invariant. In [12], Perona and Malik suggested another definition of scale-space which breaks the isotropy criterion and works better than Gaussian blurring. Since Gaussian blurring is governed by the heat equation, the thermal conductivity in all directions is constant. As a result boundaries will be blurred. Perona and Malik's idea is to halt the heat-flow process at object boundaries. To do this they control the thermal conductivity $c(x, y, t)$ using the magnitude of the image gradient. When the gradient is large, which indicates the existence of a likely edge, the value c is small. When the gradient is small, on the other hand, the value of c is large. They generate a family of coarse resolution images which are the solutions of the anisotropic diffusion equation

$$\frac{\partial \mathcal{I}}{\partial t} = div((c(x, y, t)(\frac{\partial \mathcal{I}}{\partial x} + \frac{\partial \mathcal{I}}{\partial y}))) \tag{3}$$

where div is the divergence operator. The method is demonstrated to outperform Gaussian blurring, preserving boundary sharpness and location.

Recently, there has been considerable interest in the use of graph-spectral methods for image segmentation. The pioneering work here was done by Shi and Malik [15]. The idea is to characterise the similarity of image pixels using a weight matrix which is computed by exponentiating the difference in pixel brightness. From the weight matrix the Laplacian matrix (the degree matrix minus the weight matrix) of the associated weighted graph is computed. The bi-partition of the graph that minimises the normalised cut is located using the Fiedler eigen-vector of the Laplacian.

This paper aims to exploit the close relationship between the heat-kernel and the Laplacian eigensystem to develop a graph-spectral method for scale-space image representation. Our method is motivated by the heat kernel on graphs [9] which is based on the heat equation for discrete structures, recently proposed in

the machine learning domain. According to the heat-equation, the Laplacian determines the rate of heat-flow across the weighted graph with time. The solution to the heat equation, i.e. the heat-kernel, is found by exponentiating the Laplacian eigensystem with time. We exploit this property to develop a scale-space representation from the affinity weight matrix. According to our representation, time plays the role of scale. By varying time we control the amount of blurring resulting from heat-flow.

2 Heat Kernels on Graphs

To commence, suppose that the graph under study is denoted by $G = (V, E, W)$ where V is the set of nodes, $E \subseteq V \times V$ is the set of edges and $W : E \rightarrow [0, 1]$ is the weight function. Since we wish to adopt a graph-spectral approach we introduce the adjacency matrix A for the graph where the elements are

$$A(u, v) = \begin{cases} W(u, v) & \text{if } (u, v) \in E \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

We also construct the diagonal degree matrix D, whose elements are given by $D(u, u) = deg(u) = \sum_{v \in V} A(u, v)$. ¿From the degree matrix and the adjacency matrix we construct the Laplacian matrix $L = D - A$, i.e. the degree matrix minus the adjacency matrix. The spectral decomposition of the Laplacian matrix is $L = \Phi \Lambda \Phi^T$ where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_{|V|})$ is the diagonal matrix with the decreasingly ordered eigenvalues $(0 = \lambda_1 < \lambda_2 \leq \lambda_3 ...)$ as elements and $\Phi = (\phi_1 | \phi_2 | .... | \phi_{|V|})$ is the matrix with the correspondingly ordered eigenvectors as columns. Since L is symmetric and positive semi-definite, the eigenvalues of the Laplacian are all positive. The eigenvector ϕ_2 associated with the smallest non-zero eigenvalue λ_2 is referred to as the Fiedler-vector. We are interested in the heat equation associated with the Laplacian and with the accompanying initial conditions $h_0 = \mathcal{I}$ where $\mathcal{I}$ is the identity matrix, i.e.

$$\frac{\partial h_t}{\partial t} = -Lh_t \tag{5}$$

where h_t is the heat kernel and t is time. The heat kernel can hence be viewed as describing the flow of heat across the edges of the graph with time. The rate of flow is determined by the Laplacian of the graph. The solution to the heat equation is found by exponentiating the Laplacian eigen-spectrum, i.e.

$$h_t = \exp[-tL] = \Phi \exp[-t\Lambda]\Phi^T. \tag{6}$$

The heat kernel is a $|V| \times |V|$ matrix, and for the nodes u and v of the graph G the resulting element is

$$h_t(u, v) = \sum_{i=1}^{|V|} \exp[-\lambda_i t]\phi_i(u)\phi_i(v) \tag{7}$$

When t tends to zero, then $h_t \simeq I - Lt$, i.e. the kernel depends on the local connectivity structure or topology of the graph. If, on the other hand, t is large, then $h_t \simeq \exp[-\lambda_2]\phi_2\phi_2^T$, where λ_2 is the smallest non-zero eigenvalue and ϕ_2 is the associated eigenvector, i.e. the Fiedler vector. Hence, the large time behavior is governed by the global structure of the graph.

3 Graph Scale-Space

The heat kernel matrix h_t is real valued and from the well-known 'kernel trick' [14] can be interpreted as a inner-product or Gram matrix. As a result the nodes of the graph can be viewed as residing in a possibly infinite dimensional Hilbert space. In other words, $h_t(u, v)$ efficiently characterizes the similarity between the nodes u and v. The Laplacian L encodes the local structure of a graph and dominates the heat-kernel at small time, but as time t increases then the global structure emerges in h_t.

From the standpoint of heat diffusion, the heat kernel h_t is the solution of the heat equation (5). As pointed out in [10], for an equally weighted graph the heat kernel h_t is the counterpart of the Gaussian kernel for discrete spaces $R^{|V|}$ with variance $\sigma^2 = 2t$. The value of $h_t(u, v)$ decays exponentially with the distance or weight of edge $W(u, v)$. It is useful to consider the following picture of the heat diffusion process on graphs. Suppose we inject a unit amount of heat at the vertex k of a graph, and allow the heat diffuse through the edges of the graph. The rate of diffusion over the edge $E(u, v)$ is determined by the edge weight $W(u, v)$. At time t, the heat kernel value of $h_t(k, v)$ is the amount of heat accumulated at vertex v.

Following recent work on graph-spectral methods for image segmentation [15] [11] we abstract images using the graph $G = (V, E, W)$ where the vertices of G are the pixels of the image, and an edge is formed between each pair of vertices. We denote the pixel intensities of the image as a column vector $\mathcal{I}_0$. The weight of each edge, $W(u, v)$, is a function characterizing the relationship between the pixels u and v. We would like to generate a family of coarser resolution images from $\mathcal{I}_0$ using heat flow on the graph G. To do this we inject at each vertex an amount of heat energy equal to the intensity of the associated pixel. The heat at each vertex diffuses through the graph edges as time t progresses. The edge weight plays the role of thermal conductivity. If two pixels belong to the same region, then the associated edge weight is large. As a result heat can flow easily between them. On the other hand, if two pixels belong to different regions, then the associated edge weight is very small, and hence it is difficult for heat to flow from one region to another. We wish to minimize the influence of one region on another. This behaviour is of course captured by the standard weight matrix

$$W(u,v) = \begin{cases} e^{-\frac{|\mathcal{I}_0(u)-\mathcal{I}_0(v)|^2}{\sigma^2}} & \text{if } \|X(u) - X(v)\| \leq r \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $\mathcal{I}_0(u)$ and $X(u)$ are the intensity and location of the pixel u respectively. This heat evolution model is similar to the graph heat kernel described in

Section 2, except that the initial heat residing at each vertex is determined by the pixel intensities. Since we wish to find the heat at each node of the graph at time t, the heat diffusion here is still controlled by the graph Laplacian. So the evolution of the image intensity $\mathcal{I}_0$ follows the equation

$$\frac{\partial \mathcal{I}_t}{\partial t} = -L\mathcal{I}_0. \tag{9}$$

The solution of the above equation is $\mathcal{I}_t = e^{-tL}\mathcal{I}_0 = h_t\mathcal{I}_0$. So the intensity of pixel v at time t is

$$\mathcal{I}_t(v) = \sum_{u=1}^{|V|} \mathcal{I}_0(u) \times h_t(u, v) \tag{10}$$

Since each row u of the heat kernel h_t satisfies the conditions $0 \leq h_t(u, v) \leq 1 \forall v$ and $\sum_{v=1}^{|V|} h_t(u, v) = 1$, the total intensity of the image at all scales (times) is preserved.

3.1 Lazy Random Walk View of Graph Scale-Space

Our proposed graph scale-space also has an explanation from the viewpoint of the continuous time lazy random walk. Consider a lazy random walk with transition matrix $T = (1 - \alpha)I + \alpha D^{-1}A$ which migrates between different nodes with probability α and remains static at a node with probability $1 - \alpha$. In the continuous time limit, i.e. $N \to \infty$, let $t = \alpha N$, then

$$\lim_{N\to\infty} T^N = \lim_{N\to\infty} ((1 - \alpha)I + \alpha D^{-1}A)^N \tag{11}$$

$$= e^{-tD^{-1}L} \tag{12}$$

Let p_t be the vector whose element $p_t(i)$ is the probability of visiting node i of the graph under the random walk. The probability vector evolves under the equation $\frac{\partial p_t}{\partial t} = -Lp_t$, which has the solution $p_t = e^{-tL}p_0$. As a result

$$p_t = h_t p_0 \tag{13}$$

As a result, the heat kernel is the continuous time limit of the lazy random walk. If we normalize the image intensity vector $\mathcal{I}_0$ and consider it as the initial probability distribution of the associated graph, then the intensity or probability of each node at time t is given by (13).

3.2 Approximate Schemes to Estimate the Heat Kernel

Since in practice the number of image pixels is large, it is time consuming and demanding on memory space to calculate the heat kernel through finding all the eigenvalues and eigenvectors of the graph Laplacian matrix. However, we can perform a McLaurin expansion [2] on the heat-kernel to re-express it as a polynomial in t with the result

$$h_t = e^{-tL} = I - tL + \frac{t^2}{2!}L^2 - \frac{t^3}{3!}L^3 + \cdots \tag{14}$$

Hence, we can approximate h_t using the leading terms. However, in practice the graph Laplacian matrix L of an image is too large and this still proves computationally restrictive. To overcome this problem, we explore the following two simplification schemes:

Scheme 1: Since the graph Laplacian L is positive semi-definite, its eigenvalues are all positive (with the exception of one that is zero for a connected graph). Further, since $h_t = \Phi \exp[-t\Lambda]\Phi^T$, then only the largest few eigenvalues and corresponding eigenvectors of L make a significant contribution to h_t. As a result, since the graphs we use here are only locally connected and as result L is very sparse, then we can use the Lanczos algorithm [5] to find the leading few eigenvalues and eigenvectors. If we select the largest d eigenvalues, then

$$h_t \approx \Phi_d \exp[-t\Lambda_d]\Phi_d^T \tag{15}$$

where Φ_d is a $n \times d$ matrix with the first d columns of Φ and Λ_d is a $d \times d$ diagonal matrix containing the first d eigenvalues.

Scheme 2: An alternative is to restrict our attention to pixels that are close to one-another. We can then use a smaller $n_1 \times n_2$ window of each pixel to construct a smaller graph. As a result we can use the heat kernel of this smaller graph to calculate the intensity of the pixel at time t. This method has a high degree of potential parallelism and so could be implemented on a multi-processor machine.

We have used both of the above schemes in our experiments, and both give good performance. Moreover, unlike Perona and Malik, our method does not need iteration to compute the brightness of the image at different scales.

3.3 Properties of Graph Scale-Space

Anisotropic diffusion is based on the heat conduction equation for a two dimensional continuous function, and locates the solution using an iterative numerical scheme for discrete images. Our scale-space construction commences with the discrete image data and is derived from the diffusions (or lazy random walks) on the discrete image structures. Exact solutions are found using the graph spectrum. In our method the total pixel weight is invariant to the diffusion time t, while the weight or gradient in the numerical scheme for anisotropic diffusion needs to be updated at each step.

Our graph-based scale-space representation overcomes the drawbacks of the Gaussian scale-space outlined in Section 1 since it is anisotropic due to the difference in edge weights. Moreover, it has the following characteristics which are similar to the method of Perona and Malik:

1. Causality: The graph scale-space representation has no spurious details generated through the fine to coarse scale sampling. Witkin [13] and Koenderink [8] pointed out that any candidate multi-scale description must satisfy this criteria.
2. Object boundaries maintain sharpness at coarse scales and the locations of the boundaries do not shift through the scale-space.
3. Region smoothing is preferred over boundary smoothing.

4. The total intensity of the image does not change, and the contrast of different regions is maintained at different scales
5. The new scale-space can be efficiently calculated without iteration using approximation schemes.

4 Experiments

In this section we present the results of applying our method to synthetic and real world data, and provide some experimental evaluation. We have also compared our method with Gaussian smoothing and Perona and Malik's algorithm (Anisotropic diffusion). In the following, Figures 1 and 4 used simplification scheme 1 and the remaining results used scheme 2. In all our experiments we set $r = 1$.

We first constructed a synthetic image and generated a sequence of blurred images with different amounts of added noise. Then, both anisotropic diffusion and the heat kernel filter were applied to the sequence. The scale spaces of a sample image from the sequence are shown in Figure 1. To compare the two methods, we counted the number of error pixels of each image of the sequence at different scales or time. The image statistics are plotted in Figure 2. When the amount of noise is small, anisotropic diffusion works a little better than the heat kernel method. However, when the amount of added noise becomes larger, then the heat kernel method clearly outperforms anisotropic diffusion.

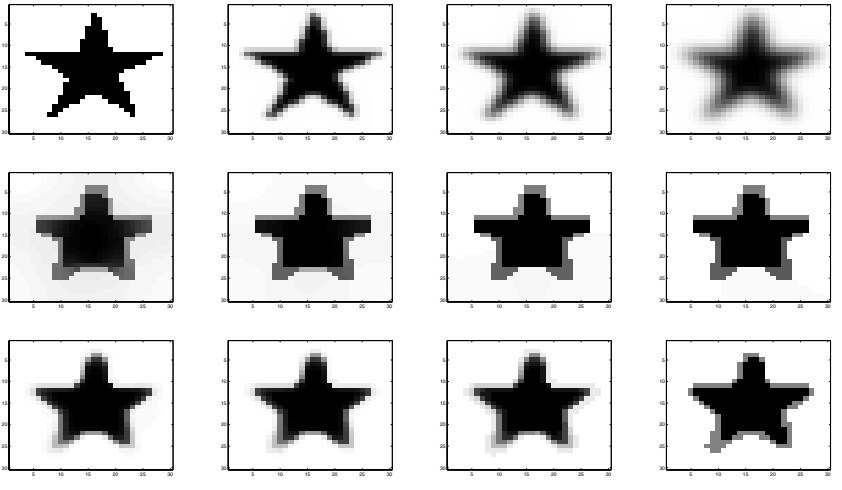

Fig. 1. Row 1: Four synthetic blurred images with noise 0, 0.1, 0.2 and 0.4. Row 2: Smoothing the last image of row 1 using anisotropic diffusion with $\lambda = 0.20$, $K = 0.1$, and 50, 100, 300, 800 iterations. Row 2: Smoothing the same image of row 2 using graph heat kernel with $\sigma = 0.05$, t=500, 1000, 3000, 5000.

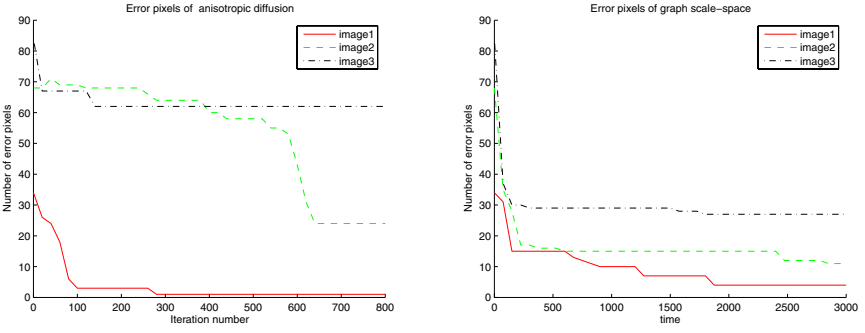

Fig. 2. Error pixels comparison of the the anisotropic diffusion (left) and graph scale space (left)

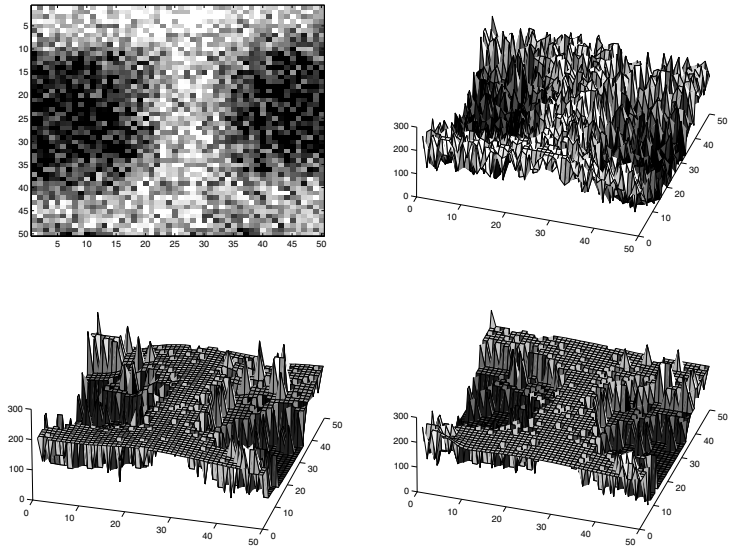

Fig. 3. (a) Synthetic heavily noisy image. (b) 3D brightness of original image. (c) 3D brightness using anisotropic diffusion after 350 iterations. (d) 3D brightness after using Graph heat kernel, window size: 13×13; $\sigma = 0.1$; $t = 25$.

In Figure 3 we illustrate the effect of the heat kernel method on another synthetic image. Panel (a) shows the original image, which is subject to considerable noise. Panel (b) displays the grey-scale values as an elevation plot. The second row shows the smoothed images after applying anisotropic diffusion and graph heat kernel filter. Here both methods recovered the main image structure, and the boundaries are well preserved. Comparing the two elevation plots it is clear that the heat kernel filter preserves the original structure a little better.

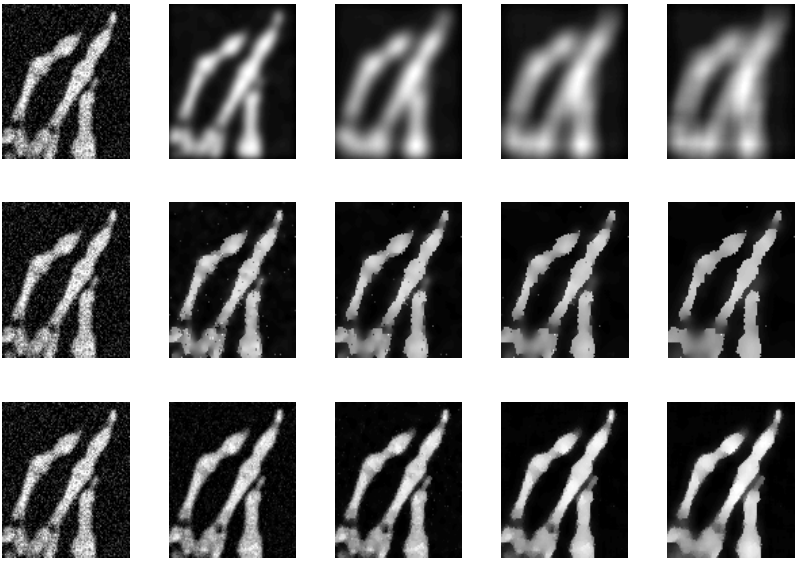

Fig. 4. CT hand with noise (image size: 94 × 110). Row 1: Linear Gaussian scale-space, $\sigma = 0, 2, 4, 8, 16$. Row 2: Anisotropic scale-space, 0, 10, 20, 30, 50 iterations. Row 3: Graph scale-space, $\sigma = 0.1$; $t = 0, 0.1, 1, 5, 15$. All scale parameters increase from left to right.

In Figure 4 we show the result of applying the method to a CT scan of a hand. The top row shows the result of Gaussian filtering. Here the different images are for different widths of the Gaussian filter. The middle row shows the result of anisotropic diffusion and the bottom row shows the result of heat-kernel smoothing. Here the different images are for different values of scale or t. In the case of the Gaussian scale-space, the main effect of increased filter width is increased blurring. When the heat-kernel method is used, then the effect is to smooth noise while preserving fine image and boundary detail.

Another synthetic example is shown in Figure 5. Here the test image is a picture of a house, with 10% Gaussian noise added. The top two rows show the smoothed image obtained using a Gaussian filter and the resulting edge map detected using Canny's method [1]. The middle two rows and bottom two rows are the results of applying the anisotropic diffusion and heat-kernel filter respectively. In the first column, we show the original image and its edge-map. The remaining columns show the results obtained with increasing values of scale or time t. From the top two rows, although the results obtained with the Gaussian filter control the effects of noise, the edge-structures are badly eroded. For comparison, the effect of anisotropic diffusion and the heat kernel filter are to smooth away the noise present in the original image and edge-map, while preserving edge-structure. Comparing the edges detected, the heat kernel filter preserves the edges and junctions best.

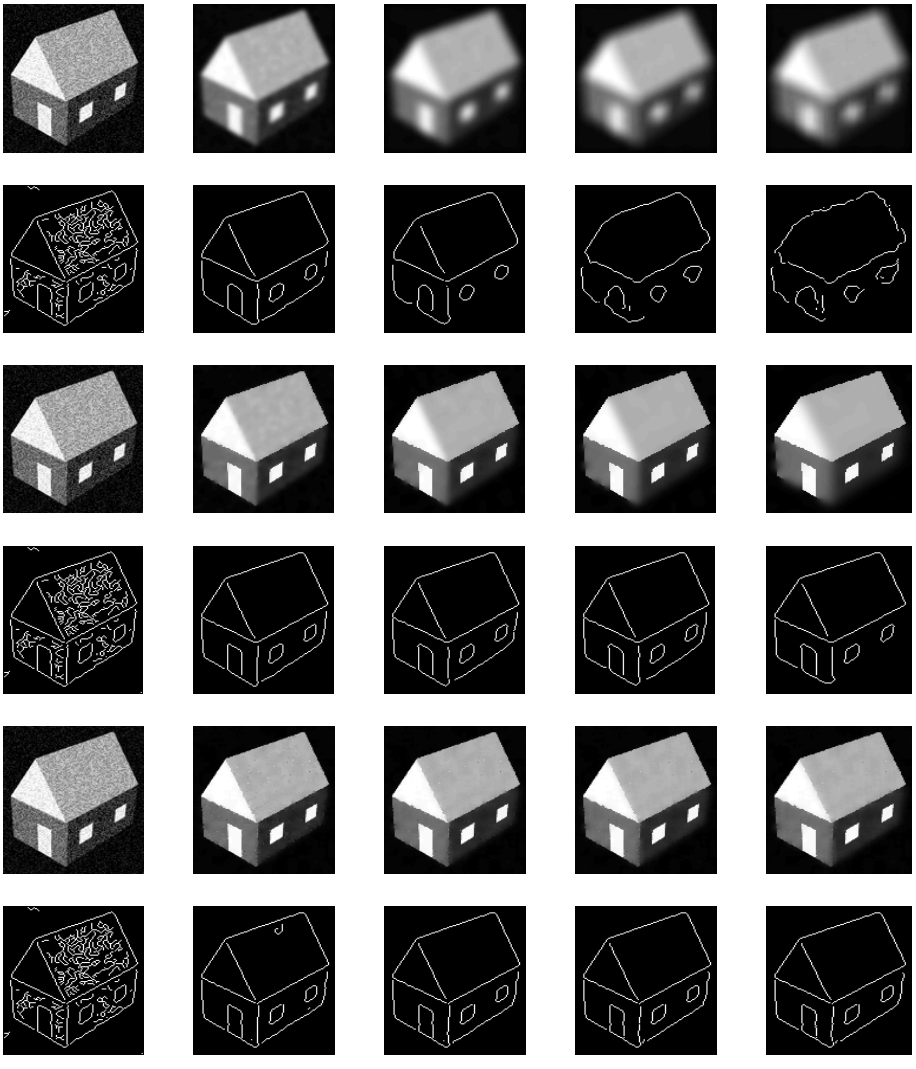

Fig. 5. Synthetic house with 10% noise (image size: 120 × 141). Row 1: Gaussian scale-space, $\sigma = 0, 2, 4, 8, 16$. Row 2: Edges detected using Canny detector [1] with Gaussian kernel variance as row 1. Edges are distorted and the junctions disappear. Row 3: Anisotropic scale-space, 0, 10, 20, 30, 40 iterations. Row 4: Edges using the same parameters of row 3. Row 5: Graph scale-space, window size: 11 × 11; $\sigma = 0.08$; $t = 0, 2, 5, 10, 15$. Row 6: Edges using the same parameters of row 5. The object shapes, boundaries and edge junctions are all preserved. All scale parameters increase from left to right.

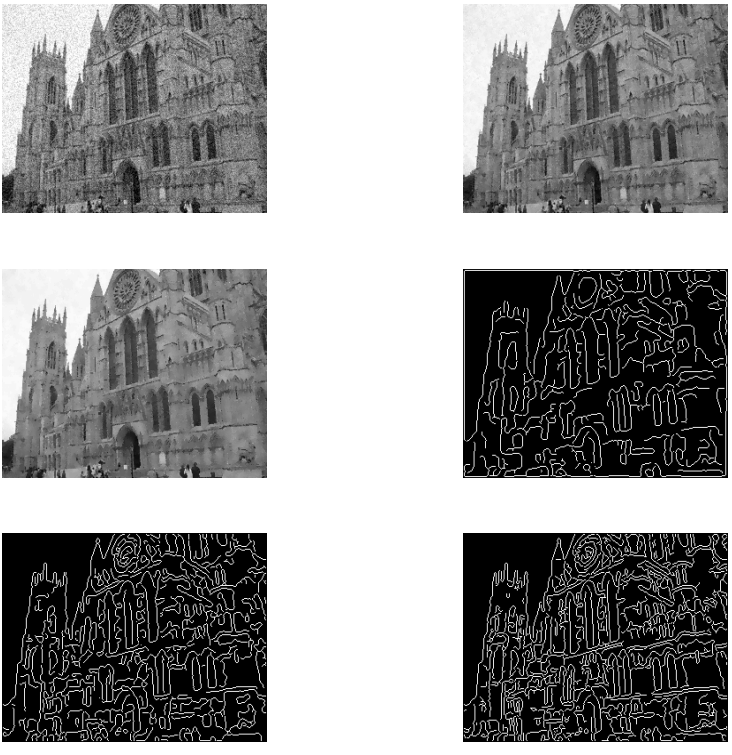

Fig. 6. York Minster with 15% noise (image size: 350 × 262). (a) original picture. (b) and (c) the results after using graph kernel with window size: 15 × 15; $\sigma = 0.06$; $t = 2, 5$. (d) edges detected using Gaussian kernel. (e) edges detected using anisotropic diffusion. (f) edges detected using graph heat kernel.

A complex real-world example is shown in Figure 6. Subfigure (a) shows the original image, and (b) and (c) show the results of applying the heat-kernel to the original image. Subfigure (f) shows the result of applying edge detection to the smoothed image in (b). For comparison subfigures (d) and (e) show the results of applying edge detection to the output of Gaussian filtering and anisotropic diffusion respectively. The main feature to note here is that the heat kernel best preserves the fine detail of the image.

5 Conclusions and Future Work

In this paper we have shown how the heat kernel can be used to smooth images without loss of fine boundary detail. Our approach is a graph-spectral one. We commence from an affinity matrix which measures the similarity of pixel grey-scale values. From the affinity matrix we compute the Laplacian, and the spectrum of the Laplacian matrix is used to estimate the elements of the

heat-kernel. Experiments show that the boundaries and regions extracted from the smoothed images preserve fine detail.

Our future plans are to extend the method to the processing of vector fields. In particular we are interested in how the method can be used to segment structures from tensor MRI imagery.

References

1. J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. and Machine Intell.*, 8(6):679 – 698, 1986.
2. F.R.K. Chung and S.-T. Yau. Discrete green's functions. In *J. Combin. Theory Ser.*, pages 191–214, 2000.
3. S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. and Machine Intell.*, 6(6):721–741, 1984.
4. Basilis Gidas. A renormalization group approach to image processing problems. *IEEE Trans. Pattern Anal. and Machine Intell.*, 11(2):164–180, 1989.
5. G.H. Golub and C.F. Van Loan. *Matrix Computations*. John Hopkins Press, 1989.
6. R. Hummel. The scale-space formulation of pyramid data structures. *Parallel Computer Vision, Ed. by L. Uhr, Academic Press*, 1987.
7. J. Babaud, A. Witkin, M. Baudin, and R. Duda. Uniqueness of the gaussian kernel for scale-space filtering. *IEEE Trans. Pattern Anal. and Machine Intell.*, 8(1):26–33, 1986.
8. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.
9. R. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. *19th Intl. Conf. on Machine Learning (ICML) [ICM02].*, 2002.
10. M. Lozano and F. Escolano. A significant improvement of softassign with diffusion kernels. In *SSPR and SPR 2004, LNCS 3138*, pages 76–84, 2004.
11. M.Meila and J.Shi. A random walks view of spectral segmentation. *In proceedings of AI and STATISTICS (AISTATS)*, 2001.
12. P. Perona and J. Malik. Scale-space and edge dectection using anisotropic diffusion. *IEEE Trans. Pattern Anal. and Machine Intell.*, 12(7):629–639, 1990.
13. Andrew P.Witkin. Scale-space filtering. In *8th Int. Joint Conf. on Artificial Intelligence, Karlsruhe, Germany*, pages 1019–1021, 1983.
14. M. Scholkopf and A. Smola. *Learning with kernels*. MIT Press, 2001.
15. J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
16. R. Wilson and M. Spann. Finite prolate spheroidal sequences and their application ii: Image feature description and segmentation. *IEEE Trans. Pattern Anal. and Machine Intell.*, 10(2):193–203, 1988.
17. A. Yuille and T. Poggio. Scaling theorems for zero crossings. *IEEE Trans. Pattern Anal. and Machine Intell.*, 8(1):15–25, 1986.