

Steady State Random Walks for Path Estimation

Antonio Robles-Kelly and Edwin R. Hancock

Department of Computer Science
University of York, York YO1 5DD, UK
{arobkell, erh}@minster.cs.york.ac.uk

Abstract. This paper describes a graph-spectral method for path estimation. Our aim is to find a maximum probability path through a lattice of pixel sites. We characterise the path recovery problem using a site transition matrix. A graph-spectral analysis of the transition matrix reveals how the maximum probability path can be located using an eigenvector of the associated normalised affinity matrix. We demonstrate the utility of the resulting method on the problem of recovering surface height from a field of surface normals.

1 Introduction

The recovery of maximum probability paths through a pixel lattice is one that arises throughout computer vision. This problem involves computing transition probabilities or costs associated with sites, and then searching for the maximum probability or minimum cost path. Of course, the underlying optimisation problem has exponential complexity, and hence exhaustive search is not a valid option. It is for this reason that optimisation methods such as dynamic programming [1], simulated annealing [2] and bayesian techniques [3] have been used to provide practical solutions to the problem. However, in this paper we aim to take a different approach and adopt a graph-spectral approach to the problem.

The idea underpinning graph-spectral methods is to abstract the problem in hand using a weighted graph. Here the nodes represent these basic image entities, and the weighted edges represent affinity relations between the entities. By computing the eigenvalues and eigenvectors of the weight matrix, it is possible to find groups or clusters of tokens. The graph-spectral method is in fact one of energy minimisation since the eigenvectors can be shown to be minimisers of a quadratic form. In fact, graph-spectral methods have recently proved highly effective in image processing and computer vision. Perhaps the best known method is that of Shi and Malik [4] which has shown how to locate image regions by recursively bisecting a weighted graph that represents the affinity of pairs of pixels. The method is based on the normalised cut. This is a measure of the relative weight of the edges connecting the two parts of the partition (the cut) to the weight assigned to the edges within the two parts of the bisection (the association). A relaxed solution to the bisection problem is found by locating the eigenvector associated with the second smallest eigenvalue of the Laplacian matrix (the degree matrix minus the affinity weight matrix).

Although it is convenient to work with the Laplacian, since it is positive and semi-definite, grouping and segmentation can also be performed using an edge-weight or affinity matrix. For instance, both Sarkar and Boyer [5] and Perona and Freeman [6] have developed matrix factorisation methods for line-segment grouping that use eigenvectors of an affinity matrix rather than the associated Laplacian. The Sarkar and Boyer [5] method can be understood as maximising the association (i.e. the total edge weight) of the clusters.

The methods described above all share the feature of using the eigenvectors of a Laplacian or an affinity matrix to define groups or clusters or objects. However, graph-spectral methods can also be used for path analysis tasks on graphs. For instance, it is well known that the path length distribution can be computed from the spectrum of eigenvalues of the adjacency matrix [7]. Ideas from spectral-graph theory have also been used to analyse the behaviour of random walks on graphs [8–10]. The observation underpinning this work is that random walks on a weighted graph can be represented as Markov chains in which the transition probabilities are computed from the normalised edge weights. The problem investigated is to compute the transition probability between pairs of pixel sites after a large number of time steps have elapsed. This study has led to a number of interesting findings. Of direct relevance to this paper is the fact that the steady state random walk on the graph is characterised by the leading eigenvector of the normalised edge-weight matrix. In addition, there are important relationships between the eigenvectors of the edge-weight matrix and other quantities related to random walks. These include the access time for a node (i.e. the expected number of time steps that must have elapsed before the node is visited) and the mixing rate (i.e. the rate at which the random walk converges to its steady state). The relationship between the leading eigenvector of the edge weight matrix and the steady state random walk has been exploited in a number of areas including routing theory and information retrieval [11, 12].

The advantage of graph-spectral methods is that they can be used to find approximate or relaxed solutions without the need for parallel iterative updates at the pixel site level. The method also obviates the need for complex search algorithms. However, although they have been applied to region segmentation and grouping problems, graph-spectral methods have not been applied to curve detection problems of the sort that arise in the determination of the optimal integration path.

2 Graph Spectral Analysis

To cast the curve estimation problem in a graph-spectral setting we adopt an abstraction where the sites to be traversed are represented by a node-set V , the connectivity relations by an edge-set E and the edges have a weight function $W : E \rightarrow [0, 1]$. Here we aim to use the weight matrix W to define a Markov chain and to use the steady state random walk associated to this chain to find a path across the graph $G = (V, E)$. The elements of the weight matrix are computed using the energy or cost associated with the transitions between sites on the pixel lattice.

Suppose that $\mathcal{E}_{i,j}$ is the energy associated with the transitions between the sites with node-labels i and j , then the weight associated with the transition is $W_{i,j} = \exp[-\beta\mathcal{E}_{i,j}]$. Unfortunately, when computed in this way, the weight matrix W cannot be used directly as the transition probability matrix for the Markov chain since its rows do not sum to unity. To normalise the rows of the matrix we compute the degree of each node $\deg(i) = \sum_{j=1}^{|V|} W(i,j)$. With the diagonal degree matrix $D = \text{diag}(\deg(1), \deg(2), \dots, \deg(|V|))$ at hand, the transition probability matrix is given by $P = D^{-1}W$. The elements of the transition matrix are hence given by $P_{i,j} = \frac{1}{\deg(i)}W_{i,j}$. It is interesting to note that the transition matrix P is a row stochastic matrix. Moreover, it is related to the normalised symmetric positive definite matrix $\hat{W} = D^{-\frac{1}{2}}WD^{-\frac{1}{2}} = D^{\frac{1}{2}}PD^{-\frac{1}{2}}$, and as a result, we can write $P = D^{-\frac{1}{2}}\hat{W}D^{\frac{1}{2}}$. It is worth noting in passing that the matrix \hat{W} is related to the normalised Laplacian $L = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}} = I - \hat{W}$.

Our aim is to use the steady state random walk on the graph G as an estimate of the maximum probability path across the graph G . The walk commences at the pixel j_1 and proceeds via the sequence of pixel sites $\Gamma = \{j_1, j_2, j_3, \dots\}$. If the random walk can be represented by a Markov chain with transition matrix P , then the probability of visiting the pixel sites in the sequence above is

$$P_\Gamma = P(j_1) \prod_{l \in \Gamma} P_{j_{l+1}, j_l} = \prod_{l \in \Gamma} \frac{W_{j_{l+1}, j_l}}{\deg(l)}$$

Substituting for the path energy, we have that

$$P_\Gamma = \frac{\exp\left[-\beta \sum_{l \in \Gamma} \mathcal{E}_l\right]}{\prod_{l \in \Gamma} \deg(l)} = \frac{1}{Z_\Gamma} \exp[-\mathcal{E}_\Gamma]$$

where $\mathcal{E}_\Gamma = \beta \sum_{l \in \Gamma} \mathcal{E}_l$ and $Z_\Gamma = \prod_{l \in \Gamma} \deg(l)$. Hence, the integration path is a Markov chain with energy function \mathcal{E}_Γ and partition function Z_Γ . Further, let $Q_t(i)$ be the probability of visiting the pixel site indexed i after t -steps of the random walk and let $Q_t = (Q_t(1), Q_t(2), \dots)^T$ be the vector whose components are the probabilities of visiting the sites at time t . After t time steps we have that $Q_t = P^t Q_0$. If \hat{W}^t is the result of multiplying the symmetric positive definite matrix \hat{W} by itself t times, then $P^t = D^{-\frac{1}{2}}\hat{W}^t D^{\frac{1}{2}}$. To develop a spectral method for locating the steady state random walk, we turn to the spectral decomposition of the normalised affinity matrix $\hat{W} = D^{-\frac{1}{2}}WD^{-\frac{1}{2}} = \sum_{i=1}^N \lambda_i \phi_i \phi_i^T$ where the λ_i are the eigenvalues of \hat{W} and the ϕ_i are the corresponding eigenvectors. By constructing the matrix $\Phi = (\phi_1 | \phi_2 | \dots | \phi_N)$ with the eigenvectors of \hat{W} as columns and the matrix $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$ with the eigenvalues as diagonal elements, we can write the spectral decomposition in the more compact form $\hat{W} = \Phi \Lambda \Phi^T$. Since, the eigenvectors of \hat{W} are orthonormal, i.e. $\Phi \Phi^T = I$, we have that $\hat{W}^t = \Phi \Lambda^t \Phi^T$. Substituting the spectral expansion of the matrix \hat{W} into the expression for the state-vector of the random walk at time step t , we find

$$Q_t = D^{-\frac{1}{2}} \Phi \Lambda^t \Phi^T D^{\frac{1}{2}} Q_0 = \left\{ \sum_{i=1}^{|V|} \lambda_i^t D^{-\frac{1}{2}} \phi_i \phi_i^T D^{\frac{1}{2}} \right\} Q_0$$

The leading eigenvalue of \hat{W} is unity, i.e. $\lambda_* = 1$. Furthermore, from spectral-graph theory [9], we know that, provided that the graph G is not a bipartite graph, then the smallest eigenvalue $\lambda_{|V|}$ is greater than -1 . As a result, when the Markov chain approaches its steady state, i.e. $t \rightarrow \infty$, then all but the first term in the above series become negligible. Hence, the steady state random walk is given by $Q_s = \lim_{t \rightarrow \infty} Q_t = D^{\frac{1}{2}} \phi_* \phi_*^T D^{-\frac{1}{2}} Q_0$, where ϕ_* is the leading eigenvector of the normalised affinity matrix \hat{W} . This establishes that the leading eigenvector of the normalised affinity matrix \hat{W} determines the steady state of the random walk. It is also important to note that the equilibrium equation for the Markov process is $Q_s = P Q_s$, where Q_s is the vector of steady-state site visitation probabilities. Hence, since the leading eigenvalue of P is unity, then it follows that Q_s is the leading eigenvector of P . For a more complete proof of this result see the book by Varga [13] or the review of Lovasz [8].

We aim to visit the pixel sites on the lattice in the order of their steady-state state probabilities. Suppose that the initial state vector for the sites is uniform, i.e. $Q_0 = (\frac{1}{|V|}, \dots, \frac{1}{|V|})^T$. As a result the steady-state probability of visiting the pixel site i is

$$Q_s(i) = \frac{1}{|V|} \sum_{j=1}^{|V|} \sqrt{\frac{\deg(j)}{\deg(i)}} \phi_*(i) \phi_*(j) = \frac{1}{|V|} \frac{\phi_*(i)}{\sqrt{\deg(i)}} \sum_{j=1}^{|V|} \sqrt{\deg(j)} \phi_*(j)$$

Since the summation appearing above is the same for all pixel sites, the probability rank order is determined by the quantity $\hat{\phi}_*(i) = \frac{\phi_*(i)}{\sqrt{\deg(i)}}$.

3 Curvature Dependant Weights

The application vehicle used in this paper is the identification of an integration path that can be used to reconstruct a surface from a field of surface normals. The surface integration problem arises in shape-from-shading and shape-from-texture. Our aim is to reconstruct the height function for the surface S from a planar field of surface normals, under the assumption that the image of the surface is formed under orthographic projection. To realise this goal, we require an integration path. This path must traverse or connect the sites of the pixel lattice. By traversing the path, the relative surface height function can be reconstructed. This is done using the trapezium rule to increment the height using the distance travelled on the path and the known slant and tilt angles of the surface normals at different locations on the image plane. In the work reported here the path is one that optimises a graph-spectral criterion that penalises high curvature. To this end, we require a means of gauging the affinity of pixels based on an image plane approximation to the surface curvature. The path must minimise the change in surface normal direction or sectional curvature across the field of

surface normals. Suppose that \mathbf{N}_i is the surface normal at the point indexed i on the pixel lattice. We note that if the path between the locations i and j can be approximated by a circle of radius R on the surface, then the approximate sectional curvature is $|\hat{\kappa}_{i,j}| = \frac{1}{R_{i,j}}$. If the line connecting the pixel sites on the image plane is of length $s_{i,j}$, then the change in direction of the radius vector of the circle is $\theta_{i,j} = \arccos \mathbf{N}_i \cdot \mathbf{N}_j$, and as a result $\cos \theta_{i,j} = \mathbf{N}_i \cdot \mathbf{N}_j$. If the angle $\theta_{i,j}$ is small, then we can make the Maclaurin approximation $\cos \theta_{i,j} \simeq 1 - \frac{\theta_{i,j}^2}{2} = \mathbf{N}_i \cdot \mathbf{N}_j$. Moreover, the small angle approximation to the radius of curvature of the circle is $R_{i,j} = \frac{s_{i,j}}{\theta_{i,j}}$ and hence

$$\hat{\kappa}_{i,j}^2 = \frac{2(1 - \mathbf{N}_i \cdot \mathbf{N}_j)}{s_{i,j}^2} \quad (1)$$

To compute the elements of the transition probability matrix we associate to the pair of pixels a cost or energy that is equal to the square of the product of the distance between the sites and sectional curvature of the connecting path. Hence, the transition weight matrix has elements

$$W_{i,j} = \exp \left[-\beta \hat{\kappa}_{i,j}^2 l_{i,j}^2 \right] = \exp \left[-2\beta(1 - \mathbf{N}_i \cdot \mathbf{N}_j) \right] \quad (2)$$

With this definition of the weight matrix, we can also view the recovery of the graph-spectral integration path as one of energy minimisation. The leading eigenvector of the matrix \hat{W} satisfies the condition $\phi_* = \arg \max_{\phi} \phi^T \hat{W} \phi = \arg \max_{\phi} \phi^T D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \phi$. We can make the relationship to the raw field of surface normals more explicit by introducing the matrix $F = (\mathbf{N}_1 | \mathbf{N}_2 | \dots | \mathbf{N}_{|V|})$ with the surface normals as columns. When the constant β is small, then making use of the Maclaurin expansion of the exponential weighting function we can write $W = \mathbf{e}\mathbf{e}^T - \beta(\mathbf{e}\mathbf{e}^T - F^T F)$ where $\mathbf{e} = (1, 1, \dots, 1)$ is an all-ones vector of length $|V|$. Using this approximation it is a straightforward matter to show that the path is the one that satisfies the condition

$$\phi_* = \arg \max_{\phi} \phi^T F^T F \phi = \arg \min_{\phi} \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} \mathbf{N}_i \cdot \mathbf{N}_j \phi(i) \phi(j)$$

Hence, the integration path will minimise the change in surface normal direction.

Our aim is to use the sequence of pixel sites given by the rank-order of the eigenvector coefficients to define a serial ordering for the sites on the pixel lattice. If we visit the sites of the pixel lattice in the order defined by the magnitudes of the coefficients of the leading eigenvector of the normalised affinity matrix, then the path is the steady state of the Markov chain. In this paper, we aim to exploit this property to locate a connected path on the sites of the pixel lattice, and to use this path for surface integration and height recovery. Unfortunately, the path followed by the steady state random walk is not edge-connected. Hence, we need a means of placing the pixel sites in an order in which neighbourhood connectivity constraints are preserved using the elements of the leading eigenvector ϕ_* .

To do this we commence from the pixel site associated with the largest component of ϕ_* , i.e. the largest site probability. We then sort the elements of the scaled leading eigenvector such that they are both in the decreasing magnitude order of the coefficients of the eigenvector, and satisfy edge connectivity constraints on the graph. The procedure is a recursive one that proceeds as follows. At each iteration, we maintain a list of pixel sites visited. At iteration k let the list of pixel sites be denoted by \mathcal{L}_k . Initially, $\mathcal{L}_0 = j_0$ where $j_0 = \arg \max_j \phi_*(j)$, i.e. j_0 is the component of ϕ_* with the largest magnitude. Next, we search through the set of first neighbours $\mathcal{N}_{j_0} = \{k | (j_0, k) \in E\}$ of j_0 to find the pixel site associated with the largest remaining component of ϕ_* . The second element in the list is $j_1 = \arg \max_{l \in \mathcal{N}_{j_0}} \phi_*(l)$. The pixel site index j_1 is appended to the list of pixel sites visited and the result is \mathcal{L}_1 . In the k th (general) step of the algorithm we are at the pixel site indexed j_k and the list of pixel sites visited by the path so far is \mathcal{L}_k . We search through those first-neighbours of j_k that have not already been traversed by the path. The set of pixel sites is $C_k = \{l | l \in \mathcal{N}_{j_k} \wedge l \notin \mathcal{L}_k\}$. The next site to be appended to the path list is therefore $j_{k+1} = \arg \max_{l \in C_k} \phi_*(l)$. This process is repeated until no further moves can be made. This occurs when $C_k = \emptyset$ and we denote the index of the termination of the path by T . The serial ordering of the pixel sites that results from this edge-based sorting is the integration path $\Gamma = \mathcal{L}_T$.

Our surface height recovery algorithm proceeds along the sequence of pixel-sites defined by the order of the coefficients. As we move from pixel-site to pixel-site defined by this path we increment the surface-height function. The trigonometry of the height incrementation process is as follows. At step n of the algorithm we make a transition from the pixel with path-index j_{n-1} to the pixel with path-index j_n . The distance between the pixel-centres associated with this transition is d_n . This distance together with the surface normals $\mathbf{N}_{j_n} = [N_{j_n}(x), N_{j_n}(y), N_{j_n}(z)]^T$ and $\mathbf{N}_{j_{n-1}} = [N_{j_{n-1}}(x), N_{j_{n-1}}(y), N_{j_{n-1}}(z)]^T$ at the two pixel-sites may be used to compute the change in surface height associated with the transition. The height increment is given by

$$h_n = \frac{d_n}{2} \left\{ \frac{N_{j_n}(x)}{N_{j_n}(y)} + \frac{N_{j_{n-1}}(x)}{N_{j_{n-1}}(y)} \right\} \quad (3)$$

If the height-function is initialised by setting $z_{j_0} = 0$, then the centre-height for the pixel with path-index j_n is $z_{j_{n+1}} = z_{j_n} + h_n$.

4 Experiments

We commence with some experiments on synthetic data. The aim here is to determine the accuracy of the surface reconstruction method. To this end, we have generated synthetic surfaces. From the surfaces, we have computed the field of surface normal directions. We have then applied the graph-spectral method to the field of surface normals to recover an estimate of the surface height.

In Figure 1, we show the results obtained for a series of different surfaces. In the top row we show the original synthetic surface. The second row shows the

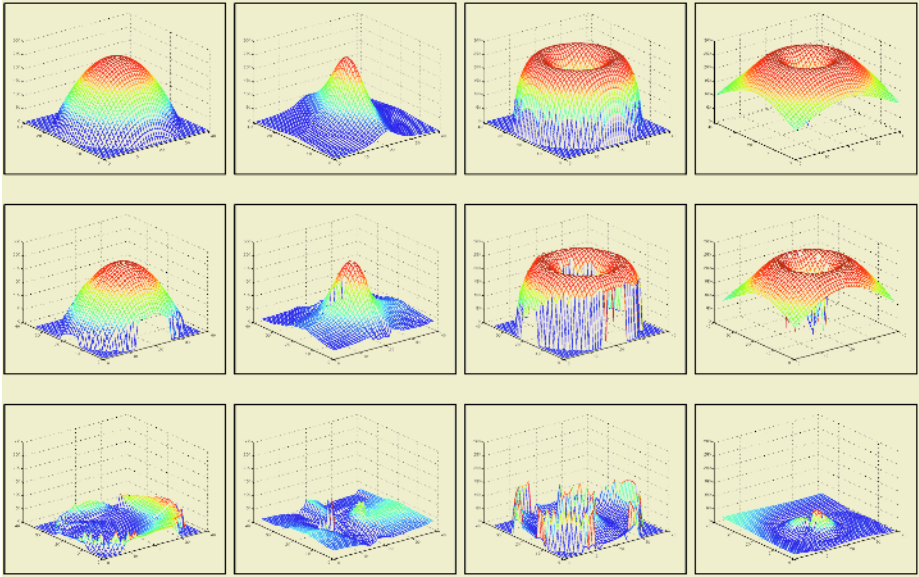


Fig. 1. Top row: Artificially generated data; Second row: Reconstructed surface; Bottom row: Error plot.

surface reconstructed from the field of surface normals. The bottom row shows the absolute error between the ground-truth and reconstructed surface height. From left-to-right, the surfaces studied are a dome, a sharp ridge, a torus and a volcano. In all four cases the surface reconstructions are qualitatively good. For the dome the height errors are greater at the edges of the surface where the slope is largest. In the case of the ridge, there are errors at the crest. For the volcano, there are some problems with the recovery of the correct depth of the “caldera”, i.e. the depression in the centre. For the reconstructed surfaces, the mean-squared errors are 5.6% for the dome, 10.8% for the ridge, 7.8% for the torus and 4.7% for the volcano. Hence, the method seems to have greater difficulty for surfaces containing sharp creases.

We have repeated these experiments under conditions of controlled noise. To do this we have added random measurement errors to the surface height. The measurement errors have been sampled from a Gaussian distribution with zero mean and variance $\sigma = 1$. In Figure 2, we show the result of reconstructing the surfaces shown in Figure 1 when noise has been added. In the left-hand column of the figure we show the field of surface normals for the noise-free surface. In the second column, we show the field of surface normals for the noise-corrupted surface. In the third column, we show the reconstructed height-function obtained from the noisy surface normals. The fourth, i.e. rightmost, column shows the difference between the height of the surface reconstructed from the noisy surface normals and the ground-truth height function. In the case of all four surfaces, the gross structure is maintained. However, the recovered height is clearly noisy.

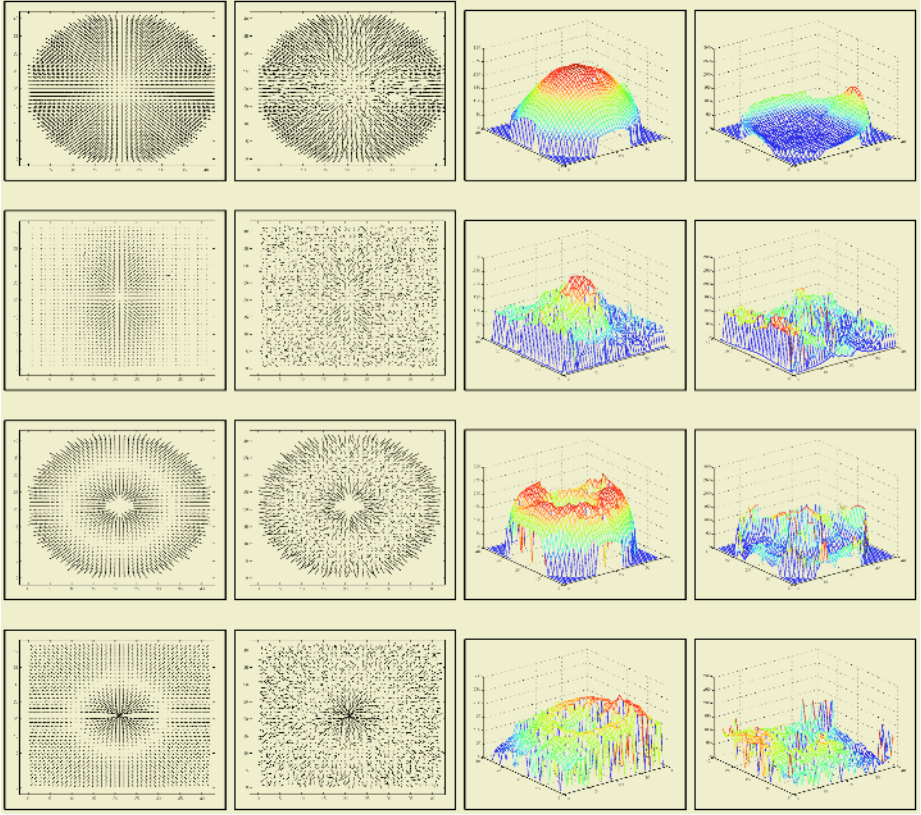


Fig. 2. Left-hand column: Needle-map without added noise; Second Column: Needle-map with Gaussian noise added; Third column: Reconstructed surface; Fourth column: Error plot.

The height difference plots are relatively unstructured. These are important observations. They mean that our graph-spectral method simply transfers errors in surface normal direction into errors in height, without producing structural noise artefacts.

We have also applied our surface recovery method to needle-maps extracted from real-world data using the shape-from-shading algorithm of Worthington and Hancock [14]. In the columns of Figure 3 we show, from left-to-right, the raw image, two views of the reconstructed surface and the integration path. In each case the integration path seems to follow the height contours on the surface, and both the overall geometry and the surface detail of the objects is well reproduced.

5 Conclusions

In this paper, we have demonstrated how steady state random walks can be used for path estimation on pixel lattices. We have illustrated the utility of the

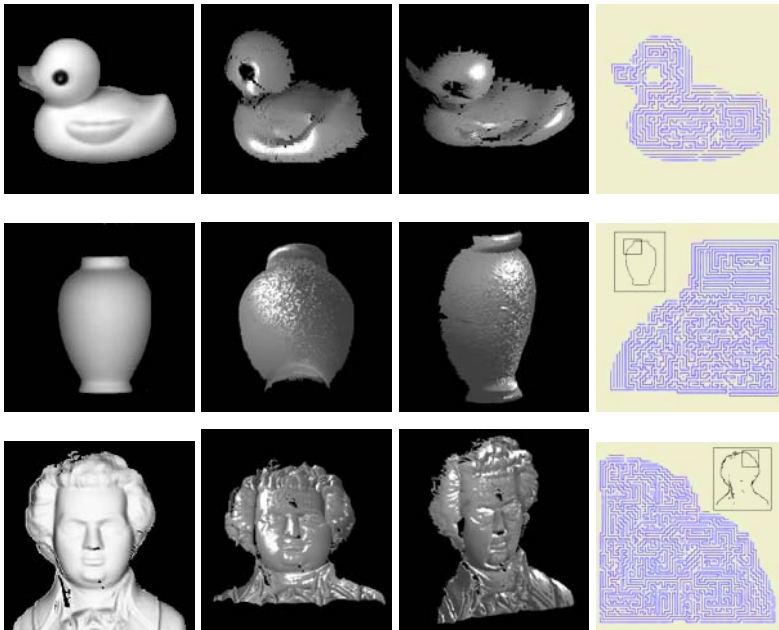


Fig. 3. Results on real-world imagery.

method for purposes of surface integration from fields of surface normals. Our future plans are to develop a more sophisticated model. In this paper, we have sought the path that is the steady state random walk of a Markov chain on a graph. This is a type of diffusion process. A more principled approach may be to pose the recovery of the integration path as the solution of a stochastic differential equation. It may also be interesting to investigate whether the idea of recovering a path using graph-spectral methods can be applied to other 2D curve enhancement problems.

References

1. L. J. Hubert and R. G. Golledge. Matrix reorganization and dynamic programming: Applications to paired comparisons and unidimensional seriation. *Psychometrika*, 46:429–441, 1981.
2. R. Burkard, V. Deinko, and G. Woeginger. The traveling salesman and the pq-tree. *Mathematics of Operations Research*, 23(3):613–623, 1998.
3. A.L. Yuille, M. Ferraro, and T. Zhang. Image warping for shape recovery and recognition. *Computer Vision and Image Understanding*, 72(3):351–359, 1998.
4. J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
5. S. Sarkar and K. L. Boyer. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. *Computer Vision and Image Understanding*, 71(1):110–136, 1998.

6. P. Perona and W. T. Freeman. Factorization approach to grouping. In *Proc. ECCV*, pages 655–670, 1998.
7. N. L. Biggs. *Algebraic Graph Theory*. Cambridge University Press, 1993.
8. L. Lovász. Random walks on graphs: a survey. *Bolyai Society Mathematical Studies*, 2(2):1–46, 1993.
9. Fan R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.
10. D. Cvetković, M. Doob, and H. Sachs. *Spectra of Graphs: Theory and Application*. Academic Press, 1980.
11. Y. Azar, A. Fiat, A. R. Karlin, F. McSherry, and J. Saia. Spectral analysis of data. In *ACM Symposium on Theory of Computing*, pages 619–626, 2000.
12. J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc. ACM-SIAM symposium on discrete algorithms*, pages 668–677, 1998.
13. R. S. Varga. *Matrix Iterative Analysis*. Springer, second edition, 2000.
14. P. L. Worthington and E. R. Hancock. New constraints on data-closeness and needle map consistency for shape-from-shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(12):1250–1267, 1999.