Discrete Representation of Top Points via Scale Space Tessellation

B. Platel[1], M. Fatih Demirci[2], A. Shokoufandeh[2], L.M.J. Florack[1],
F.M.W. Kanters[1], B.M. ter Haar Romeny[1], and S.J. Dickinson[3]

[1] Eindhoven University of Technology,
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
[2] Drexel University, 3141 Chestnut Street,
Philadelphia, PA 19104, United States of America
[3] University of Toronto, 6 King's College Rd.,
Toronto, Ontario, Canada M5S 3G4
b.platel@tue.nl, {mdemirci, ashokouf}@cs.drexel.edu
{l.m.j.florack, f.m.w.kanters, b.m.terhaarromeny}@tue.nl
sven@cs.toronto.edu

Abstract. In previous work, singular points (or top points) in the scale space representation of generic images have proven valuable for image matching. In this paper, we propose a construction that encodes the scale space description of top points in the form of a directed acyclic graph. This representation allows us to utilize graph matching algorithms for comparing images represented in terms of top point configurations instead of using solely the top points and their features in a point matching algorithm, as was done previously. The nodes of the graph represent the critical paths together with their top points. The edge set will capture the neighborhood distribution of vertices in scale space, and is constructed through a Delaunay triangulation scheme. We also will present a many-to-many matching algorithm for comparing such graph-based representations. This algorithm is based on a metric-tree representation of labelled graphs and their low-distortion embeddings into normed vector spaces via spherical encoding. This is a two-step transformation that reduces the matching problem to that of computing a distribution-based distance measure between two such embeddings. To evaluate the quality of our representation, two sets of experiments are considered. First, the stability of this representation under Gaussian noise of increasing magnitude is examined. In the second set of experiments, a series of recognition experiments is run on a small face database.

1 Introduction

Previous research has shown that top points (singular points in the scale space representation of generic images) have proven to be valuable sparse image descriptors that can be used for image reconstruction [6, 12] and image matching [7, 14]. In our previous work, images were compared using a point matching

R. Kimmel, N. Sochen, J. Weickert (Eds.): Scale-Space 2005, LNCS 3459, pp. 73–84, 2005.

scheme which took into account the positions, scales, and differential properties of corresponding top points [7, 6]. The underlying matching framework was based on the Earth Mover's Distance, a powerful, many-to-many point matching framework. However, treating the points as an unstructured collection ignores the salient group structure that may exist within a given scale or across scales. Grouping certain top points together explicitly encodes the neighborhood structure of a point, effectively enriching the information encoded at a point – information that can be exploited during both indexing [16] and matching [17].

In this paper, we take an unstructured set of top points and impose a neighborhood structure on them. Inspired by the work of Lifshitz and Pizer [10], we will encode the scale space structure of a set of top points in a *directed acyclic graph* (DAG). Specifically, we combine the position-based grouping of the top points provided by a Delaunay triangulation with the scale space ordering of the top points to yield a directed acyclic graph. This new representation allows us to utilize powerful graph matching algorithms to compare images represented in terms of top point configurations, rather than using point matching algorithms to compare sets of isolated top points. Specifically, we draw on our recent work in many-to-many graph matching [9, 2, 3], which reduces the matching problem to that of computing a distribution-based distance measure between embeddings of labelled graphs.

We describe our new construction by first elaborating on those basics of catastrophe theory required to introduce the concept of a top point. Next, we formally define a top point, and introduce a measure for its stability that will be used to prune unstable top points. Section 4 describes the construction of the DAG through a Delaunay triangulation scheme. Section 5 reviews our many-to-many DAG matching algorithm, which will be used to evaluate the construction. In the first experiment, we examine the stability of the construction under Gaussian noise of increasing magnitude applied to the original images. In the second experiment, we examine the invariance of the graph structure to within-class image deformation, which may include minor displacements of points both within and across scales.

2 Catastrophe Theory

Critical points are points at any fixed scale in which the gradient vanishes, i.e., $\nabla u = 0$. The study of how these critical points change as certain control parameters change is called *catastrophe theory.* A Morse critical point will move along a *critical path* when a control parameter is continuously varied. In principle, the single control parameter in the models of this article can be identified as the scale of the blurring filter. The only generic morsifications in Gaussian scale space are *creations* and *annihilations* of pairs of Morse hypersaddles of opposite Hessian signature[1] [1, 4]. An example of this is given in Fig. 1.

[1] The Hessian signature is the sign of the determinant evaluated at the location of the critical point.

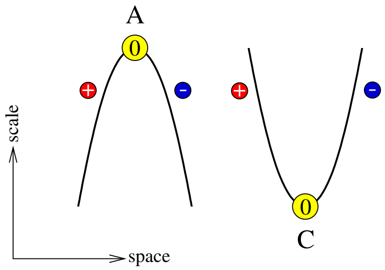


Fig. 1. The generic catastrophes in isotropic scale space. Left: an annihilation event. Right: a creation event. A positive charge ⊕ denotes an extremum, a negative charge ⊖ denotes a saddle, ⊚ indicates the singular point

The movement of critical points through scale, together with their annihilations and creations, forms *critical paths* in scale space. In this article, we will restrict ourselves to generic (non-symmetrical) 2D images, but the theory is easily adapted to higher dimensions. In the 2D case, the only generic morsification is an annihilation or creation where a saddle point and an extremum point meet. Critical paths in 2D therefore consist of an *extremum branch*, that describes the movement of an extremum through scale, and a *saddle branch*, that describes the movement of the saddle with which the extremum annihilates. Note that there is always one extremum branch continuing up to infinite scale [11]. In Fig. 2, the critical paths and their top points are shown for a picture of a face.

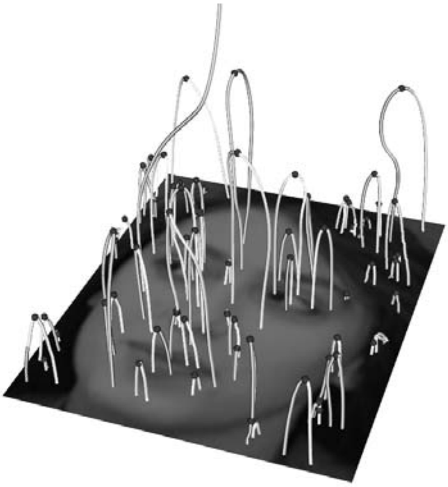


Fig. 2. Critical paths and top points of a face

3 Top Points

The points at which creation and annihilation events take place are often referred to as *top points*[2]. A top point is a critical point at which the determinant of the Hessian degenerates:

$$\begin{cases} \nabla u = 0 \\ \det(H) = 0. \end{cases} \tag{1}$$

An easy way to find these top points is by means of zero-crossings in scale space. This involves derivatives up to second order and yields sub-pixel results. Other, more elaborate methods, can be used to find or refine the top point positions. For details, the reader is referred to [4].

It is obvious that the positions of extrema at very fine scales are sensitive to noise. This, in most cases, is not a problem. Most of these extrema are blurred away at fine scales and won't affect our matching scheme at slightly coarser scales. However, problems do arise in areas in the image that consist of almost constant intensity. One can imagine that the positions of the extrema (and thus the critical paths and top points) are very sensitive to small perturbations in these areas. These unstable critical paths and top points can continue up to very high scales since there is no structure in the vicinity to interact with. To account for these instable top points, we need to have a measure of stability, so that we can either give unstable points a low weight in our matching scheme, or disregard them completely.

A top point is more stable in an area with a lot of structure. The amount of structure contained in a *spatial* area around a top point can be quantified by the *total (quadratic) variation* (TV) norm over that area:

$$TV(\Omega) \overset{\text{def}}{=} \frac{\sigma^2 \int_\Omega \|\nabla u(x)\|^2 dV}{\int_\Omega dV} \tag{2}$$

We calculate the TV norm in a circular area with radius $\lambda\sigma$ around a top point at position (x_c, t_c). Note that the size of the circle depends on the scale σ. The integration area of the TV norm Ω is defined by:

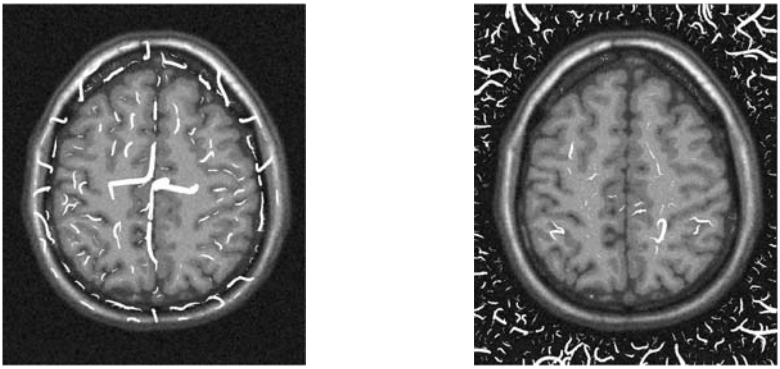
$$\Omega : \|x - x_c\|^2 \leq \lambda^2\sigma^2. \tag{3}$$

By using a spatial Taylor series around the considered top point, and taking into account that the first order spatial derivatives in this point are zero, we can simplify the TV-norm Eqn. (2) to what we refer to as the *differential TV-norm* by the following limiting procedure[14]:

$$tv \overset{\text{def}}{=} \lim_{\lambda \to 0} \frac{4}{\pi} \frac{1}{\lambda^4} TV(\lambda) = \sigma^4 \mathrm{Tr}(H^2) \tag{4}$$

The proportionality factor $\frac{4}{\pi}$ is irrelevant for our purposes. The normalization factor $\frac{1}{\lambda^4}$ is needed prior to evaluation of the limit since $TV(\lambda) = \mathcal{O}(\lambda^4)$. Eqn. (4)

[2] The terminology is reminiscent of the 1D case, in which only annihilations occur generically.



(a) Stable paths

(b) Unstable paths

Fig. 3. Spatial projection of critical paths of a MR brain scan image. The paths are filtered by thresholding the stability norm of their top points. Most instabilities occur in flat regions, as expected

has been referred to by Koenderink as *deviation from flatness*, which can indeed be seen to be the differential counterpart of Eqn. (2). It enables us to calculate a stability measure for a top point *locally* by using only its second order derivatives. This stability norm can be used to weigh the importance of top points in our matching scheme, or to remove any unstable top points by thresholding them on their stability value. The latter is demonstrated in Fig. 3.

4 Construction of the Graph

The goal of our construction is two-fold. First, we want to encode the neighborhood structure of a set of points, explicitly relating nearby points to each other in a way that is invariant to minor perturbations in point location. Moreover, when local neighborhood structure does indeed change, it is essential that such changes will not affect the encoded structure elsewhere in the graph (image). The Delaunay triangulation imposes a position-based neighborhood structure with exactly these properties [15]. It represents a triangulation of the points which is equivalent to the nerve of the cells in a Voronoi tessellation, i.e., that triangulation of the convex hull of the points in the diagram in which every circumcircle of a triangle is an empty circle [13]. The edge set of our resulting graph will be based on the edges of the triangulation. Our second goal is to capture the scale space ordering of the points to yield a directed acyclic graph, with coarser scale top points directed to nearby finer scale top points.

The first step in constructing our graph G is the detection of top points and critical paths using ScaleSpaceViz [5]. The root of G, denoted as v_1, will correspond to the single critical path that continues up to infinity; note that there is no top point associated with this critical path, but simply its position at the coarsest scale. All other nodes in G, denoted as $v_2, ..., v_n$, correspond to the detected top points and their corresponding critical paths. $v_2, ..., v_n$ are ordered in decreasing order of the scale at which they are detected, e.g., v_2 is detected at a coarser scale than v_n. As we build the Delaunay triangulation of the points, we will simultaneously construct the DAG. Beginning with the root, v_1, we have a singleton point in our Delaunay triangulation, and a corresponding single node in G. Next, at the scale corresponding to v_2, we project v_1's position down to v_2's level, and recompute the triangulation. In this case, the triangulation

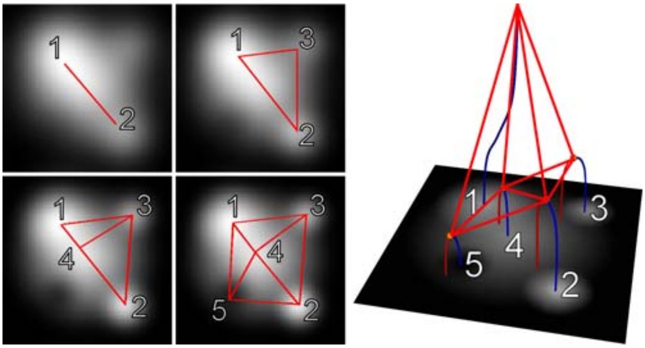


Fig. 4. Visualization of the DAG construction algorithm. Left: the Delaunay triangulations at the scales of the nodes. Right: the resulting DAG (edge directions not shown)

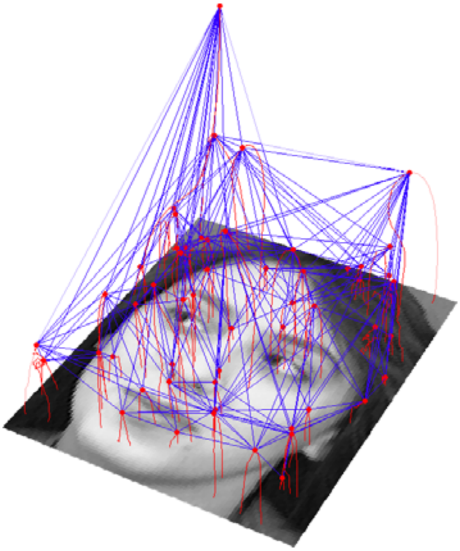


Fig. 5. The DAG obtained from applying Algorithm 1 to the critical paths and top points of the face in Fig. 2

yields an edge between v_1 and v_2. Each new edge in the triangulation yields a new edge in G, directed from coarser top points to finer top points; in this case, we add a directed edge in G from v_1 to v_2. We continue this process with each new top point, first projecting all previous top points to the new point's level, recomputing the triangulation, and using the triangulation to define new directed edges in G. A summary of this procedure is presented in Algorithm 1.

The construction is illustrated for a simple image in Fig. 4. In the top two frames in the left figure, we show the transition in the triangulation from v_2 (point 2) to v_3 (point 3); the root is shown as point 1. In the upper right frame, the triangulation consists of three edges; correspondingly, G has three edges: $(1, 2), (1, 3), (2, 3)$, where (x, y) denotes an edge directed from node x to node y. In the lower left figure, point 4 is added to the triangulation, and the triangulation recomputed; correspondingly, we add edges $(1, 4), (2, 4), (3, 4)$ to G (note that $(1, 2)$ is no longer in the triangulation, but remains in G). Finally, in the lower right frame, point 5 is added, and the triangulation recomputed. The new edges in the triangulation yield new edges in G: (2,5),(4,5),(1,5). The right side of Fig. 4 illustrates the resulting graph (note that the directions of the edges are not shown). Fig. 5 is the result of applying this construction to the face of Fig. 2.

Algorithm 1 Top point graph construction procedure

```
Detect the critical paths.
Extract the top points from the critical paths.
Label the extremum path continuing up to infinity as $v_1$.
Label the rest of the nodes (critical paths, together with their top points) according
  to the scale of their top points from high scale to low as $v_2, ..., v_n$.
For $i = 2$ to $n$ evaluate node $v_i$:
    Project the previous extrema into the scale of the considered node $v_i$.
    Calculate the 2D Delaunay triangulation of all the extrema at that scale.
    All connections to $v_i$ in the Delaunay triangulation are stored as directed edges
  in $G$.
```

5 Experiments

To evaluate our construction, we explore the invariance of the construction to two types of perturbations. The first is the sensitivity of the construction to noise in the image, while the second is within-class deformation resulting in displacements of top points both within and across scales. We conduct our experiments using a subset of the Olivetti Research Laboratory face database. The database consists of faces of 20 people with 10 faces per person, for a total of 200 images; each image in the database is 112×92 pixels. The face images are in frontal view and differ by various factors such as gender, facial expression, hair style, and presence or absence of glasses. A representative view of each face is shown in Fig. 6. Invariance of a graph to noise or within-class deformation requires a measure of graph distance, so that the *distance* between the original and perturbed graphs

Fig. 6. Sample faces from 20 people

can be computed. For the experiments reported in this paper, we compute this distance using our many-to-many graph matching algorithm, which we briefly describe in the next subsection. Note that we have developed a general algorithm that is in no way specifically designed for face recognition. Therefore we have not compared our method to state-of-the-art face recognition algorithms. We present this experiment only as a proof of concept.

5.1 Overview of Matching Algorithm

The matching algorithm is based on the metric-tree representation of labelled graphs and their low-distortion embeddings into normed vector spaces via spherical coding [3, 9]. The advantage of this embedding technique is that it prescribes a single vector space into which both graphs are embedded. This two-step transformation reduces the many-to-many matching problem to that of computing a distribution-based distance measure between two such embeddings. To compute the distance between two sets of weighted vectors, we use a variation of Earth Mover's Distance under transformation sets. For two given graphs, the algorithm provides an overall similarity (distance) measure.

Fig. 7 presents an overview of the approach. For a given face, we first create its DAG according to Section 4 (Transition 1), and embed each vertex of the DAG into a vector space of prescribed dimensionality using a deterministic spherical coding (Transition 2). Finally (Transition 3), we compute the distance between the two distributions by the modified Earth Mover's Distance under transformation. The dimension of the target space in Transition 1 has a direct effect on the quality of the embedding. Specifically, as the dimensionality of the target space increases, the quality of the embedding will improve. Still, there exists an asymptotic bound beyond which increasing the dimensionality will no longer improve the quality of the embedding. Details on the many-to-many matching algorithm can be found in [3].

5.2 Graph Stability Under Additive Noise

To test the robustness of our graph construction, we first examine the stability of our graphs under additive Gaussian noise at different signal levels applied to the original face images. For this experiment, the database consists of the original 200 unperturbed images, while the query set consists of noise-perturbed

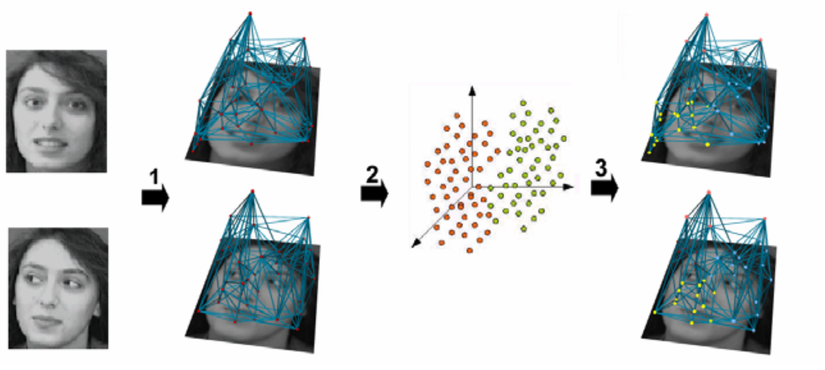


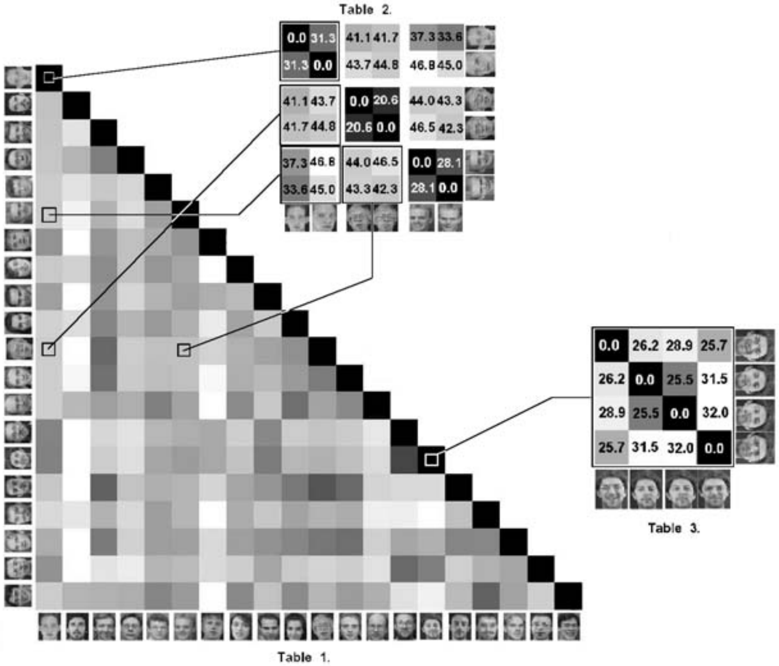
Fig. 7. Computing similarity between two given faces. (Matched point clusters are shaded with the same color.) See text

versions of the database images. Specifically, for each of the 200 images in the database, we create a set of query image by adding 1%, 2%, 4%, 8%, and 16% of Gaussian noise. Next, we compute the similarity between each query (perturbed database image) and each image in the database, and score the trial as correct if its distance to the face from which it was perturbed is minimal across all database images. This amounts to 40,000 similarity measurements for each noise level, for a total of 200,000 similarity measurements. Our results show that the recognition rate decreases down to 96.5%, 93%, 87%, 83.5%, and 74% for 1%, 2%, 4%, 8%, and 16% of Gaussian noise, respectively. These results indicate a graceful degradation of graph structure with increasing noise.

5.3 Graph Stability Under Within-Class Variation

To test the stability of the graph construction to within-class variation (e.g., different views of the same face), we first group the faces in the database by individual; these will represent our categories. Next, we remove the first image (face) from each group and compare it (the query) to all remaining database images. The image is then put back in the database, and the procedure is repeated with the second image from each group, etc., until all 10 face images of each of the 20 individuals have been used as a query. If the graph representation is invariant to within-class deformation, resulting from different viewpoints, illumination conditions, presence/absence of glasses, etc., then a query from one individual should match closest to another image from the same individual, rather than an image from another individual. The results are summarized in Table 1, Fig. 8.

The magnitudes of the distances are denoted by shades of gray, with black and white representing the smallest and largest distances, respectively. Due to symmetry, only the lower half of distance matrix is presented. Intra-object distances, shown along the main diagonal, are very close to zero.



0.0	31.3	41.1	41.7	37.3	33.6
31.3	0.0	43.7	44.8	46.8	45.0
41.1	43.7	0.0	20.6	44.0	43.3
41.7	44.8	20.6	0.0	46.5	42.3
37.3	46.8	44.0	46.5	0.0	28.1
33.6	45.0	43.3	42.3	28.1	0.0

0.0	26.2	28.9	25.7
26.2	0.0	25.5	31.5
28.9	25.5	0.0	32.0
25.7	31.5	32.0	0.0

Fig. 8. Table 1: Matching results of 20 people. The rows represent the queries and the columns represent the database faces (query and database sets are non-intersecting). Each row represents the matching results for the set of 10 query faces corresponding to a single individual matched against the entire database. The intensity of the table entries indicates matching results, with black representing maximum similarity between two faces and white representing minimum similarity. Table 2: Subset of the matching results with the pairwise distances shown. Table 3: Effect of presence or absence of glasses in the matching for the same person. The results clearly indicate that the graph perturbation due to within-class deformation, including facial expression changes, illumination change, and the presence/absence of glasses is small compared to the graph distance between different classes

To better understand the differences in the recognition rates for different people, we randomly selected a subset of the matching results among three people in the database, as shown in Table 2, Fig. 8. Here, the (i, j)-th entry shows the actual distance between face i and face j. It is important to note that the distance between two faces of the same person is smaller than that of different people, as is the case for all query faces. In our experiments, one of our objectives was to see how various factors, such as the presence or absence of glasses, affects the matching results for a single person. Accordingly, we took a set of images from the database of one person, half with the same factor, and computed the distances between each image pair. Our results show that images with the same factors are more similar to each other than to others. Table 3 of Fig. 8 presents a subset of our results. As can be seen from the table, images of the same person with glasses are more similar than those of the same person with and without glasses. Still, in terms of categorical matching, the closest face always belongs to the same person. Although these results are encouraging, further evaluation on a larger database needs to be investigated to be more conclusive.

6 Conclusions

Imposing neighborhood structure on a set of points yields a graph, for which powerful indexing and matching algorithms exist. In this paper, we present a method for imposing neighborhood structure on a set of scale space top points. Drawing on the Delaunay triangulation of a set of points, we generate a graph whose edges are directed from top points at coarser scales to nearby top points at finer scales. The resulting construction is stable to noise, and within-class variability, as reflected in a set of directed acyclic graph matching experiments.

Acknowledgments

This work is part of the DSSCV project supported by the IST Program of the European Union (IST-2001-35443). Ali Shokoufandeh gratefully acknowledges the partial support provided by grants from National Science Foundation (NSF/EIA 02-05178), and the Office of Naval Research (ONR-N000140410363). Sven Dickinson gratefully acknowledges the support of NSERC, IRIS, PREA, and CITO.

References

1. J. Damon. Local Morse theory for solutions to the heat equation and Gaussian blurring. *Journal of Differential Equations*, 115(2):368–401, January 1995.
2. M. Fatih Demirci, A. Shokoufandeh, S. Dickinson, Y. Keselman, and L. Bretzner. Many-to-many matching of scale-space feature hierarchies using metric embedding. In *Proceedings, Scale Space Methods in Computer Vision, 4th International Conference*, pages 17–32, June 2003.
3. M. Fatih Demirci, A. Shokoufandeh, S. Dickinson, Y. Keselman, and L. Bretzner. Many-to-many feature matching using spherical coding of directed graphs. In *Proceedings, 8th European Conference on Computer Vision*, pages 332–335, May 2004.
4. L. Florack and A. Kuijper. The topological structure of scale-space images. *J. Math. Imaging Vis.*, 12(1):65–79, February 2000.
5. F.M.W. Kanters. Scalespaceviz. World Wide Web, `http://www.bmi2.bmt.tue.nl/image-analysis/people/FKanters/Software/ScaleSpaceViz.html`, 2004.
6. F.M.W. Kanters, L.M.J. Florack, B. Platel, and B.M. ter Haar Romeny. Image reconstruction from multiscale critical points. In *Proceedings of the 4th international conference on Scale Space Methods in Computer Vision (Isle of Skye, UK, June 2003)*, pages 464–478.

7. F.M.W. Kanters, B. Platel, L.M.J. Florack, and B.M. ter Haar Romeny. Content based image retrieval using multiscale top points. In *Proceedings of the 4th international conference on Scale Space Methods in Computer Vision (Isle of Skye, UK, June 2003)*, pages 33–43.
8. M. Kerckhove, editor. *Scale-Space and Morphology in Computer Vision: Proceedings of the Third International Conference, Scale-Space 2001, Vancouver, Canada*, volume 2106 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, July 2001.
9. Y. Keselman, A. Shokoufandeh, M. F. Demirci, and S. Dickinson. Many-to-many graph matching via low-distortion embedding. In *Proceedings, Computer Vision and Pattern Recognition*, pages 850–857, 2003.
10. L. M. Lifshitz and S. M. Pizer. A multiresolution hierarchical approach to image segmentation based on intensity extrema. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):529–541, 1990.
11. M. Loog, J. J. Duistermaat, and L. M. J. Florack. On the behavior of spatial critical points under Gaussian blurring. a folklore theorem and scale-space constraints. In Kerckhove [8], pages 183–192.
12. M. Nielsen and M. Lillholm. What do features tell about images? In Kerckhove [8], pages 39–50.
13. A. Okabe and B. Boots. *Spatial Tesselations: Concepts and Applications of Voronoi Diagrams*. New York, 1992.
14. B. Platel, F.M.W. Kanters, L.M.J. Florack, and E.G. Balmachnova. Using multiscale top points in image matching. In *Proceedings of the 11th International Conference on Image Processing (Singapore, October 2004)*.
15. F. Preparata and M. Shamos. *Computational Geometry*. Springer-Verlag, New York, NY, 1985.
16. A. Shokoufandeh, S. Dickinson, K. Siddiqi, and S. Zucker. Indexing using a spectral encoding of topological structure. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 491–497, Fort Collins, CO, June 1999.
17. K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 30:1–24, 1999.