# 1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (V.B)

Proceedings

# 1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition

June 18–20, 1996 San Francisco, California

*Sponsored by*

IEEE Computer Society Technical Committee
on Pattern Analysis and Machine Intelligence



IEEE Computer Society Press
Los Alamitos, California

Washington • Brussels • Tokyo

IEEE Computer Society Press
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1264



IEEE Computer Society Press Order Number PR07258
IEEE Order Plan Catalog Number 96CB35909
ISBN 0-8186-7258-7
IEEE Order Plan ISBN 0-8186-7259-5
Microfiche ISBN 0-8186-7260-9
ISSN 1063-6919

*Additional copies may be ordered from:*

IEEE Computer Society Press
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: +1-714-821-8380
Fax: +1-714-821-4641
Email: cs.books@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: +1-908-981-1393
Fax: +1-908-981-9667
misc.custserv@computer.org

IEEE Computer Society
13, Avenue de l'Aquilon
B-1200 Brussels
BELGIUM
Tel: +32-2-770-2198
Fax: +32-2-770-8505
euro.ofc@computr.org

IEEE Computer Society
Ooshima Building
2-19-1 Minami-Aoyama
Minato-ku, Tokyo 107
JAPAN
Tel: +81-3-3408-3118
Fax: +81-3-3408-3553
tokyo.ofc@computer.org

Editorial production by Penny Storms
Cover by Joseph Daigle / Studio Productions
Printed in the United States of America by KNI, Inc.

The Institute of Electrical and Electronics Engineers, Inc.

# SiteCity: A Semi-Automated Site Modelling System

Yuan Hsieh*
Digital Mapping Laboratory
School of Computer Science, Carnegie-Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213-3891
E-mail: ych@maps.cs.cmu.edu

## Abstract

*This paper presents SITECITY, a semi-automated building extraction system integrating photogrammetry, geometric constraints and image understanding algorithms. Existing automated building extraction systems produce mixed results and it is clear that human intervention is required to correct mistakes from fully automated systems. SITECITY gives human operators the ability to construct and manipulate three dimensional building objects using multiple images. Image understanding algorithms are integrated into SITECITY to assist users. The automated processes in SITECITY use user-delineated roof boundaries as cues, and attempt to locate the floor of a building and match the building object in other images. In addition, photogrammetric cues are used to assist automated processes. These automated processes are described and their performance is evaluated, illustrating that automated processes in SITECITY produce comparable performance to that of human subjects.*

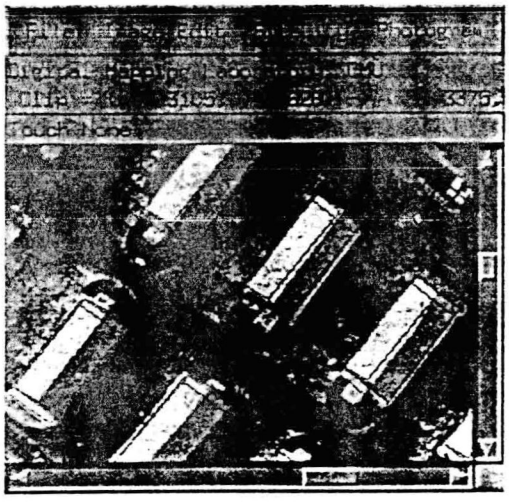
## 1 Introduction

Efficient and accurate delineation of man made features from digital aerial imagery has been a shared goal of the photogrammetry and computer vision communities for a number of years. The proliferation of digital photogrammetric workstations shows exciting possibilities for automating tedious and time consuming menial tasks. Among these tasks are automatic aerial triangulation, image mosaicing, and the generation of orthophotos and digital elevation maps (DEMs) [1; 2; 3; 4]. However, research in automated urban feature extraction has produced mixed results [5; 6; 7; 8; 9; 10; 11], and few systems [10; 9] produce 3D descriptions of extracted buildings in object space. In order to utilize these automated processes, an integrated system is needed that allows a user to correct and enhance the results of the automated systems.

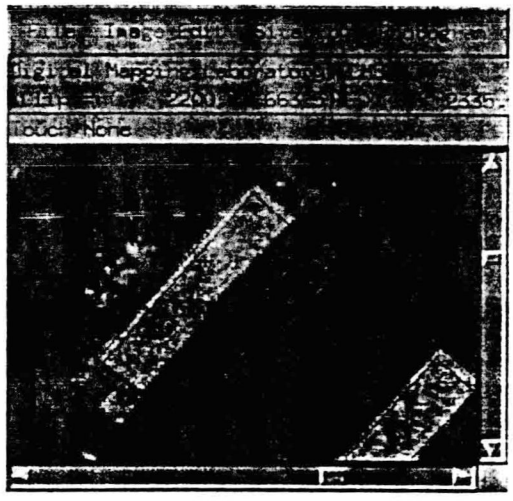
While most research has focused on fully automated feature extraction systems, there are some systems which attempt to integrate manual and automatic processes for extracting man-made features from aerial images. In [12], Mueller and Olson use a model based approach to detect and delineate rectangular and peak roof building models. In their system, users are required to specify a range of plausible parameters for the building models and approximate locations of the buildings in the image. The other system is the Cartographic Modeling Environment (CME) [13]. It uses model-based optimization techniques such as snakes [14] to perform feature extraction in an interactive environment.

SITECITY is a semi-automated three-dimensional site modeling system developed in the Digital Mapping Laboratory at Carnegie Mellon University. The philosophical and theoretical foundation that affected the design of SITECITY can be found in [15], in addition, a study of usability and validation of SITECITY is presented. A snap shot of SITECITY is shown in Figures 1 and 2. SITECITY uses rigorous photogrammetric principles and multiple images to accurately determine 3D locations of objects, such as buildings or roads in the scene. Photogrammetric methods supply cues to reduce the complexity of automated feature extraction tasks [16]. Automated processes such as model matching are used to reduce the number and complexity of manual measurements. A set of graphical user interface tools are provided for users to modify and create arbitrary objects and to supply cues, such as area of interest, to assist the automated processes.

This work was sponsored by the Advanced Research Projects Agency under Contracts DACA76-91-C-0014 and DACA76-95-C-0009 monitored by the U.S. Army Topographic Engineering Center, Fort Belvoir, VA. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Advanced Research Projects Agency, the U.S. Army Topographic Engineering Center, or the United States Government.
The Digital Mapping Laboratory's WWW Home Page may be found at: http://www.cs.cmu.edu/~MAPSLab
*Currently with Lockheed Martin Astronautic, Denver CO



(a) Image measurement window: fhrad1



(b) Image measurement window: fhrad3

Figure 1: **Snapshot of SiteCity: image measurement windows**

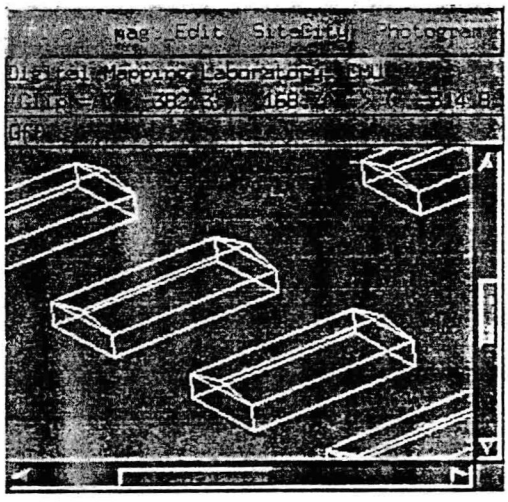


Figure 2: Snapshot of SiteCity: three dimensional display window

## 2 Automated Processes in SiteCity

Currently, SITECITY is designed for detection and delineation of three types of buildings, the peaked roof buildings, flat roof buildings and rectilinear flat roof buildings. Figure 3 shows the process flow for estimating 3D building object in SITECITY. In this scheme, a building object is generated and verified using a single image. The 3D position of the building is computed by epipolar matching of the building objects in image space. Other schemes are discussed in [15]. The initial roof polygon is measured by a user.

Three image understanding components are used to support building measurement tasks: a verification component, an object matching component and an edge estimation component [15].

**Verification Component:** The goal of the verification component is to verify the existence of a hypothesized building object in an image. The verification component used in SITECITY uses full 3D geometry to predict the visible building and shadow structures. When a 3D object is projected to an image, some of the edges might not be visible due to the viewing geometry. Therefore, before any verification is attempted, the hidden lines of the object in the image are first removed from consideration. The result is a two dimensional line drawing of the expected visible edges of the object in the image, and the problem reduces to matching this 2D line drawing to the image [17; 18; 19]. The verification component was rigorously evaluated and appeared to behave as expected [15].

**Edge Estimation Component:** Often, it is desirable to determine the terminating endpoint of an edge of known directed orientation $\theta$ and length $q$ in an image; the edge estimation component is used to achieve this task [17] by utilizing Hough transform [20].

**Object Matching Component:** The object matching component [17] is used to generate hypotheses for a known object in the image given a search region. The search region is defined by the automated processes that use this component. Given a projection of a 3D object to the image, the goal is to find instances of the projection of this object in the image, similar to the 2D object matching problem described in [18; 19].

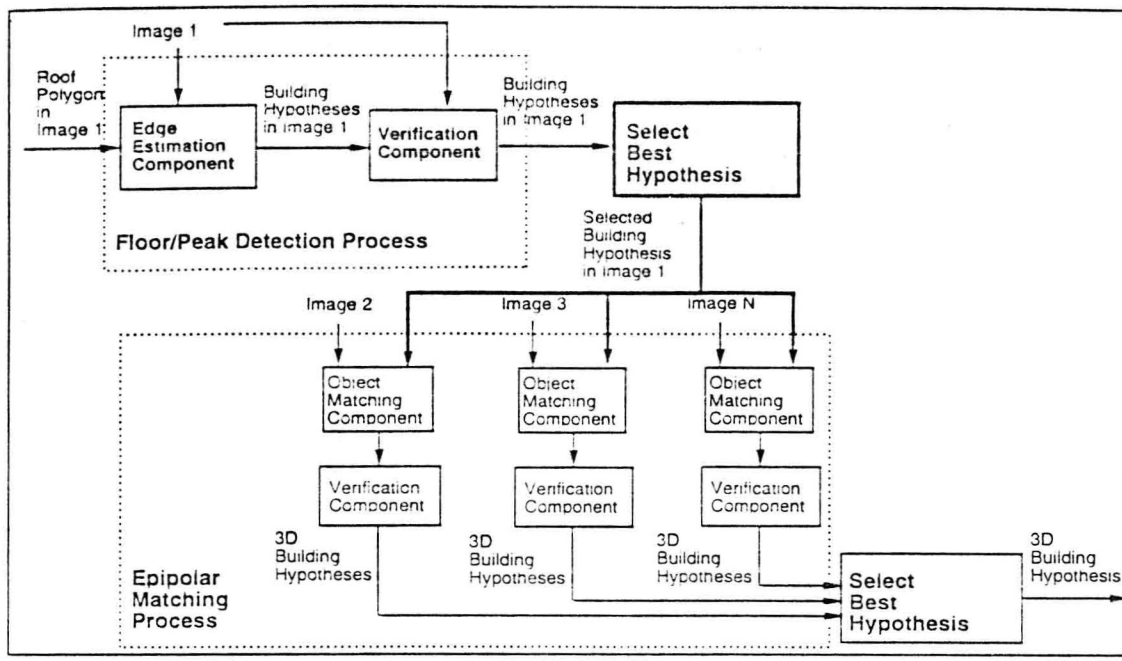



Figure 3: **Process flow for single image building detection (SIBD)**

## 2.1 Automated Processes in SiteCity

The three components described in the previous section can be composed in a variety of ways to perform different tasks [17].

**Estimate Peak Edge of a Peak Roof Building:** For a peak roof building model, once the user measures the rectangular outline for the roof, the first task is to estimate the position of the peak edge. The expected edge length and orientation can be determined by the user defined roof outline, and the search space can be constrained by the vertical vanish lines. Edge estimation component uses these information to determine plausible peak edge hypotheses. To evaluate the edge hypotheses, the verification component is used.

**Estimate Building Height:** In [9], the height of a building is measured by using an imperfect sequence finding technique [21] to locate vertical edges. Another clue in the image that allows us to determine the building height is the location of the visible floor edges, where the walls of the building meet the ground. The orientation, length and location of the floor edges are constrained by the camera geometry and the delineated roof structure, and edge estimation component is invoked to determine plausible floor location of a building. Each floor edge hypothesis produces a hypothesized building model, and the verification component is used to select the best model.

**Epipolar Matching:** Due to errors in the DEM, and the uncertainties of the camera parameters, the projection of a 3D object onto another image might not be correct. To achieve a good 3D measurement, it is often necessary to locate the projected object in several images. If the location of the object is known in one image, then the search space in another image is defined by the epipolar line and the precision of the epipolar line. Given the object model in the image and the search space, the object matching component is used to generate a set of hypotheses. The verification component is used to select the best hypothesis.

**Automatic Copying:** Frequently, many objects in the scene might be identical. Suburban scenes often have repetitive instances of the same buildings. Two objects are identical if they share the same shape, size and orientation. The object matching component can be used to reduce the effort of copying an object to a different part of the image. Since the object matching component requires the determination of the search area and the object, we devised three cues to specify the search region: point, line and areal cues. These cues are combined with the known object to form a search region [17].

After processing by the object matching component, a number of hypothesized object locations are determined. Most of these hypotheses are erroneous and the verification component is used to compute the confidence of each hypothesis. With a point cue, the hypothesis with the best confidence is selected. For the line and area cues, the best set of non-overlaping hypotheses are selected.

## 3 Measurement Accuracy and Geometric Constraints in SiteCity

Due to uncertainty in the imaging parameters and the image measurements, the calculated 3D position of the measured points will also be uncertain. In order to assess the quality of these measurements, a least-squares optimization with error propagation [22] is implemented to quantify the precision of all 3D measurements. The least-squares approach requires as input an estimate of the initial measurement precision.

There have been numerous attempts to quantify image measurement precision [23; 24; 25; 26]. These reports show that the measurement precision on a digital image depends on Signal to Noise Ratio (SNR), quantization level and pixel size of the digital image. However, since an end user often does not have control over the quantization level, and estimating the SNR of an arbitrary image is difficult, SITECITY only considers the effect of displayed resolution on the accuracy of image measurements. Every measurement is classified into one of four categories: User Input, Automated Process, Initial Position and Invisible [15]. These classifications are used to derived initial measurement uncertainty.

However, despite careful image measurements, the three dimensional object calculated by triangulating multiple image measurements often does not conform to our precise expectations for the real 3D object. In addition, direct image measurements are sometimes impossible due to occlusions or shadows. Therefore, geometric constraints [27; 15] that incorporate knowledge about the object shape are utilized to produce a more accurate 3D model. Another application of geometric constraints is the construction of complex objects by composition of a few primitive object types; this alleviates the need to define a model for every type of object one can encounter [15].

## 4 Analysis of the Automated Processes

Three scenes are used for this analysis: radt9-A(Figure 4), radt5(Figure 4) and radt10(Figure 4). These scenes are part of the Fort Hood aerial images distributed under the RADIUS program. There are four images in each scene, consists of two near-vertical and two oblique shots

### 4.1 Establishing the Ground Truth

In order to evaluate the accuracy and precision of the delineation of an automated building detection system, a *ground truth* is needed to compare results. In previous studies [9; 28], the ground truth used in the automated building detection system was typically generated by one user. However, the *ground truth* generated by one user will vary from that generated by another user.

In our experiment, a building in an image is measured by twelve users. A *ground truth* in image space can be established by determining an average position of a point using multiple measurements. An advantage of generating a *ground truth* using multiple measurements is the ability to generate an error measure for each ground truth position. This error measure, expressed as a covariance matrix, allows us to quantify the accuracy and precision of any individual measurement compared to the mean. Figure 7 shows the covariance ellipse (drawn as a diamond shape) for measured building points on one of the four images for the radt9 scene and suggests that the measurement uncertainty of a point in an image is related to the *quality* of a corner

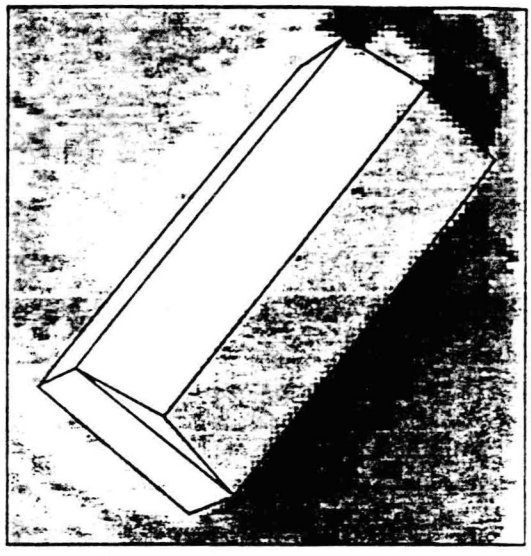


Figure 4: **Scene 1: Radt9-A**

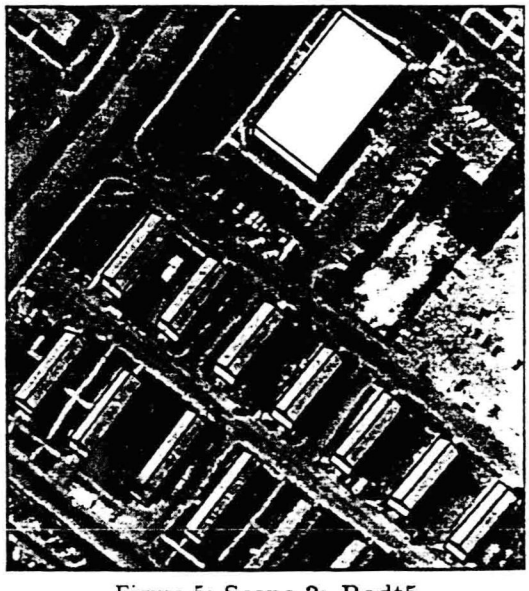


Figure 5: **Scene 2: Radt5**

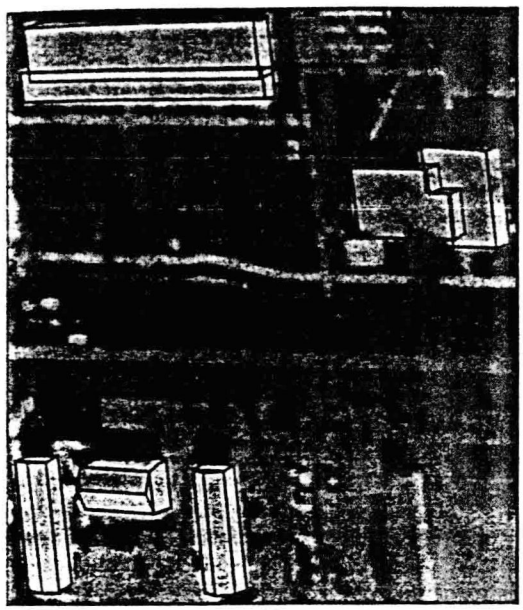


Figure 6: Scene 3: Radt10

in an image. The average position of these measurements is the center of the ellipse (diamond) and the wireframe building shown in Figure 7 is drawn using the average positions. The size of the ellipse denotes the uncertainty of an image measurement and is drawn so that the boundary of the ellipse represents a distance of one standard deviation away from the mean.

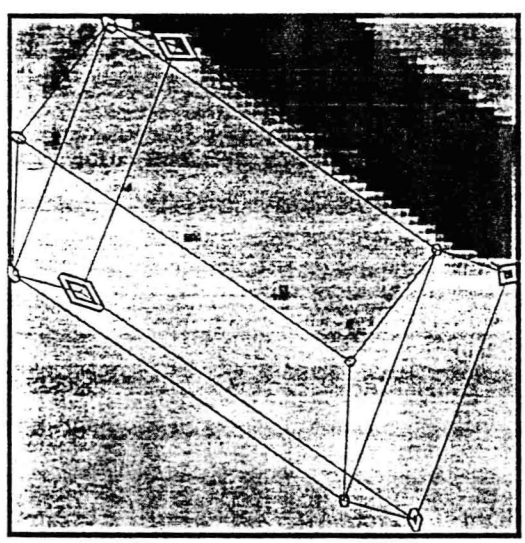


Figure 7: Covariance ellipse for radt9wob-A images

## 4.2 Evaluation of the Automated Processes

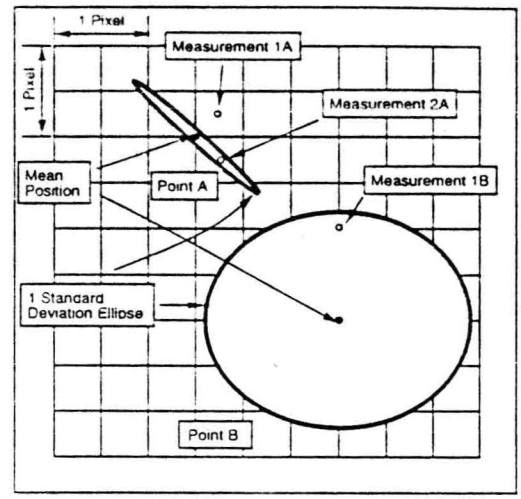



Figure 8: Example contrasting Euclidean distance and number of standard deviation for assessing accuracy of measurements

For the analysis of the floor/peak detection process, the manually measured roofs are extracted from the manually measured buildings in all images. For the evaluation of the automated peak/floor detection process, a total of 1920 roofs delineations in 12 images (4 images per scene) were used. A detail description of the collection of these roof delineation can be found in [15]. Since there are only 20 buildings in all three scene, out of the 1920 roofs, there are only 80 unique image buildings that were estimated; this redundancy allows us to account for the sensitivity of the automated processes due to different user inputs.

For each manually measured roof in each image, the floor detection process is invoked using the extracted roof as the initial condition to estimate the floor position. If the building is a peak roof building, the position of the peak edge is also estimated. Once a building is estimated in an image, the location of every visible image point estimated by the automated processes is compared with the *ground truth* measurements, the mean position and the covariance matrix (Section 4.1). Only the measurable points, visible in the image, estimated by the automated processes are used. For each image point estimated by the automated processes, the distance to the expected *ground truth* position, in terms of both the number of standard deviations away from the mean, and the Euclidean distance in terms of pixels, is computed. The Euclidean distance measures the *absolute* accuracy of a measurement in relationship to the *ground truth* position. On the other hand, the number of standard deviations measures the *relative* accuracy o

a measurement that accounts for the expected human performance.

Figure 4.2 shows some examples contrasting the two distance measures. The mean positions and the ellipses representing 1 standard deviation for both points A and B are shown. Measurements 1A and 2A estimate the position of point A and have the same distance from the mean position of point A. The accuracy in terms of the Euclidean distance will be the same for both measurements. However, in terms of the number of standard deviations, measurement 1A is about $4\sigma$ away from the mean and measurement 1B is only about $0.5\sigma$ away from the mean. The Euclidean distance measure suggests that both measurements are equally good, but compared to the performance of user measurements, measurement 2A is significantly better than 1A.

For evaluation of a semi-automated system where users are involved, the measurements that agree with users perception should be considered better. For point A, users disagree on the location of the point along the major axis of the standard deviation ellipse, but agree about the location of the point along the minor axis. Since measurement 2A agreed with users on this respect, we expect the likelihood of a user to correct measurement 2A to be less than measurement 1A. For point B, the Euclidean distance of measurement 1B to the mean position of point B is twice the distance of measurements 1A and 2A to the mean position of point A. Yet, the standard deviation measure of measurement 1B would be better than measurement 1A. This is because the position of point B is ambiguous in the image.

Using the number of standard deviations to compare the accuracy treats the automated processes with the same standard as human subjects; the automated processes need to perform as well as humans. At one extreme, for a point where there is no disagreement on its position, then any deviation in measurements using the automated processes will result in an infinite number of standard deviations away from the expected position, and we consider that the automated processes failed to measure the position correctly, even if the Euclidean distance of the measurement is less than 1 pixel away from the expected position. On the other hand, if users cannot agree on the position of a point due to poor images, then measurements made by the automated processes that is within one standard deviation of the expected position should be considered good regardless of Euclidean distance because there is no consensus in which to compare the measurements.

Both Euclidean distance and standard deviation distance are used to evaluate the performance of the automated processes. Table 1 shows the result of the floor/peak detection process for each scene and image. The image column shows the image where the floor/peak detection was performed, while *npts* is the number of measurable points estimated by the automated processes. The remaining columns shows the cumulative percentage of points within N standard deviation and N pixels of the manual measurements.

The evaluation of the epipolar matching process uses the same metrics. To evaluate the epipolar matching process, manually measured buildings in one image are projected to other images and the epipolar matching process is performed to locate the object position in those images. The position estimated by the automated process is not manually corrected, to evaluate the performance of the automated process. For this experiment, 5760 epipolar matching processes are used for this experiment for the buildings in 12 images [15]. A building delineation in one image is projected and epipolar matched to 3 other images. Since there are 80 unique building delineations in all the images, 240 epipolar matching processes are performed for each set of delineations. A total of 24 sets of different user delineated buildings are used to account for variations of the initial condition. Once the building objects are matched, the analysis of the epipolar matching process is similar to the analysis for the peak/floor detection processes. The summary of the errors are shown in Table 2.

## 5 Conclusions

In this paper, we presented a semi-automated system for building delineation using multiple images. The semi-automated system uses image understanding algorithms to assist users. These automated algorithms are unobtrusively integrated and reduce the complexity of the measurement tasks. SITECITY allows the user to measure building points on multiple images using a simple graphical interface. These building points are triangulated to estimate the three-dimensional position and precision, and constrained according to the geometric model. Complex building objects can be constructed by applying geometric constraints on several primitive components. The current system allows us to measure a large variety of building types. However, more general building object models are needed to handle scenes where the roof structures are complex. We are currently studying methods to construct other difficult building objects within SITECITY, such as the use of a generic peak roof building type, and building models with overhanging roofs.

Currently, the automated processes are affected by a combination of scene complexity, image contrast, view angles, and user performance; the automated processes perform well when the scenes are simple with few occlusions and good contrast. We originally expected that an oblique image would have better performance than a vertical image, because there are more visible building facets in oblique images. However, our experiments do not show a clear advantage for oblique images. One possible explanation is that while oblique images provide more information about the object, the search areas in the image also increase for the oblique image and

| Image | npts | % ≤ 1σ | % ≤ 2σ | % ≤ 3σ | % ≤ 1 pixel | % ≤ 2 p. | % ≤ 3 p. |
|---|---|---|---|---|---|---|---|
| radt9 | 96 | 3.13 | 17.71 | 38.54 | 35.42 | 80.21 | 94.79 |
| radt9ob | 96 | 16.67 | 55.21 | 79.17 | 43.75 | 87.50 | 97.92 |
| radt9s | 96 | 29.17 | 64.58 | 83.33 | 62.50 | 93.75 | 98.96 |
| radt9wob | 96 | 17.71 | 31.25 | 41.67 | 30.21 | 70.83 | 88.54 |
| radt5 | 864 | 19.21 | 38.31 | 51.50 | 62.38 | 25.81 | 64.12 |
| radt5ob | 1008 | 14.19 | 35.32 | 55.16 | 20.93 | 56.35 | 87.00 |
| radt5s | 1128 | 24.11 | 47.52 | 59.66 | 28.10 | 59.75 | 70.92 |
| radt5wob | 1296 | 11.88 | 31.56 | 48.69 | 27.47 | 65.51 | 77.16 |
| radt10 | 576 | 14.24 | 38.37 | 56.94 | 17.36 | 47.22 | 65.63 |
| radt10ob | 504 | 17.46 | 38.49 | 55.36 | 5.75 | 24.80 | 43.25 |
| radt10s | 456 | 15.13 | 39.25 | 52.63 | 5.26 | 31.58 | 47.37 |
| radt10wob | 672 | 2.83 | 14.14 | 27.68 | 6.25 | 27.98 | 46.88 |
| Total | 6888 | 15.35 | 36.08 | 51.84 | 21.30 | 53.61 | 70.95 |

Table 1: **Summary of floor/peak detection performances based on standard deviation and distance**

| Image | npts | % ≤ 1σ | % ≤ 2σ | % ≤ 3σ | % ≤ 1 pixel | % ≤ 2 p. | % ≤ 3 p. |
|---|---|---|---|---|---|---|---|
| radt9 | 576 | 7.47 | 22.74 | 37.85 | 30.56 | 65.10 | 82.29 |
| radt9ob | 576 | 14.93 | 41.15 | 62.15 | 34.38 | 72.05 | 88.02 |
| radt9s | 576 | 20.31 | 51.22 | 68.40 | 29.86 | 70.83 | 83.33 |
| radt9wob | 576 | 7.64 | 25.52 | 42.71 | 29.86 | 71.88 | 89.76 |
| radt10 | 4032 | 19.12 | 43.77 | 60.74 | 28.52 | 63.00 | 77.75 |
| radt10ob | 3812 | 33.13 | 57.58 | 70.91 | 13.03 | 37.09 | 54.20 |
| radt10s | 3672 | 29.90 | 55.64 | 69.36 | 23.77 | 55.07 | 72.60 |
| radt10wob | 4236 | 14.90 | 29.77 | 42.35 | 17.04 | 42.78 | 58.85 |
| radt5 | 5760 | 19.17 | 43.00 | 60.23 | 29.72 | 65.89 | 81.94 |
| radt5ob | 6479 | 21.86 | 48.76 | 66.06 | 23.74 | 60.07 | 79.92 |
| radt5s | 6768 | 24.10 | 53.87 | 71.35 | 26.71 | 63.84 | 81.44 |
| radt5wob | 7245 | 12.39 | 31.79 | 48.57 | 27.83 | 64.83 | 81.01 |
| Total | 44308 | 20.54 | 44.37 | 60.50 | 24.98 | 58.99 | 75.87 |

Table 2: **Summary of epipolar matching performances based on standard deviation and distance**

increase the chance for mismatches. Another explanation for the performance on the near vertical images is the use of the shadow verification component. Building shadows can be a useful clue in vertical images [8], where walls and floors are not visible.

The evaluation of point estimation errors in terms of both the standard deviation and the pixel distance shows the need for automated processes that can align edges and corner points with subpixel accuracy. Overall, approximately 50% of the positions estimated by the automated processes are within 2 pixels of the manual measurements. Users can often perform building delineation within subpixel accuracies by adjusting the display resolution. If the automated processes cannot produce edges and points at the expected location, even when the error is less than one pixel, most users will be compelled to fine-tune the automated solutions, reducing the usefulness of the automated processes. In future work, we hope to combine the use of fully automated systems with SITECITY to create a comprehensive environment for building delineation.

At present, SITECITY is actively being used to generate accurate three dimensional ground truth for our automated building extraction research, and to populate our spatial databases with buildings for research in simulation of virtual worlds. The utility of SITECITY for the task of building extraction illustrates the effectiveness of integrating computer vision algorithms in an interactive setting. The semi-automated approach combines the advantages of fully manual and fully automated systems to improve the overall productivity of site modeling, essential for the rapid construction of cartographic databases.

## 6 Acknowledgements

This work would not have been possible without the support of the members of the Digital Mapping Laboratory. I would like to thank Chris McGlone, Steven Cochran, Jeff Shufelt and Dave McKeown for their useful comments and insights. I would also like to acknowledge Stephen Gifford and Chris Olson who provided the the foundation of the user interface in SITECITY.

## References

[1] C. Heipke, "State-of-the-art of digital photogrammetric workstations for topographic applications," *Photogrammetric Engineering and Remote Sensing*, vol. 61, pp. 49–56, Jan. 1995.

[2] E. Gülch, "Fundamentals of softcopy photogrammetric workstations," in *Mapping and Remote Sensing Tools for the 21st Century*, (Washington, D. C.), pp. 193–204, ASPRS, Aug. 1994.

[3] P. R. Boniface, "State-of-the-art in softcopy photogrammetry," in *Mapping and Remote Sensing Tools for the 21st Century*, (Washington, D. C.), pp. 205–210, ASPRS, Aug. 1994.

[4] F. W. Leberl, "Practical issues in softcopy photogrammetric systems," in *Mapping and Remote Sensing Tools for the 21st Century*, (Washington, D. C.), pp. 223–230, ASPRS, Aug. 1994.

[5] Y.-T. Liow and T. Pavlidis, "Use of shadows for extracting buildings in aerial images," *Computer Vision, Graphics, and Image Processing*, vol. 49, pp. 242–277, Feb. 1990.

[6] Y. Cheng, R. Collins, A. Hanson, and E. Riseman, "Model matching and extension for automated 3D site modeling," in *Proceedings of the DARPA Image Understanding Workshop*, (Washington, D. C.), pp. 197–204, Morgan Kaufmann Publishers, Inc., Apr. 19–21 1993.

[7] C. Lin, A. Huertas, and R. Nevatia, "Detection of buildings using perceptual grouping and shadows," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, (Seattle, Washington), pp. 62–69, June 19–23 1994.

[8] R. B. Irvin and D. M. McKeown, "Methods for exploiting the relationship between buildings and their shadows in aerial imagery," *IEEE Transactions on Systems, Man & Cybernetics*, vol. 19, no. 6, pp. 1564–1575, 1989. Also available as Technical Report CMU–CS–88–200, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213.

[9] J. C. McGlone and J. A. Shufelt, "Projective and object space geometry for monocular building extraction," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, (Seattle, Washington), pp. 54–61, June 19–23 1994.

[10] M. Roux and D. M. McKeown, Jr., "Feature matching for building extraction from multiple views," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, (Seattle, Washington), pp. 46–53, June 19–23 1994.

[11] P. V. Fua and A. J. Hanson, "Objective functions for feature discrimination: Applications to semiautomated and automated feature extraction," in *Proceedings of the DARPA Image Understanding Workshop*, (Palo Alto, California), Defense Advanced Research Projects Agency, May 1989.

[12] W. J. Mueller and J. A. Olson, "Model-based feature extraction," in *Integrating Photogrammetric Techniques with Scene Analysus and Machine Vision* (E. B. Barrett and D. M. McKeown, eds.), vol. 1944, (Orlando, Florida), pp. 263–272, SPIE, Apr. 1993.

[13] L. H. Quam and T. M. Strat, "SRI image understanding research in cartographic feature extraction," in *Digital Photogrammetric Systems* (H. Ebner, D. Fritsch, and C. Heipke, eds.), (Karlsruhe, Germany), pp. 111–122, Wichmann, 1991.

[14] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1987.

[15] Y. Hsieh, "Design and evaluation of a semi-automated site modeling system," Tech. Rep. CMU–CS–95–195, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, Nov. 1995.

[16] D. McKeown and J. C. McGlone, "Integration of photogrammetric cues into cartographic feature extraction," in *Proceedings of the SPIE: Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision*, vol. 1944, pp. 2–15, Sept. 1993.

[17] Y. Hsieh, "Design and evaluation of a semi-automated site modeling system," in *Proceedings of the ARPA Image Understanding Workshop*, (Palm Springs, California), Advanced Research Projects Agency, Morgan Kaufmann Publishers, Inc., Feb. 1996.

[18] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 850–863, Sept. 1993.

[19] G. Borgefors, "Hierarchical chamfer matching: A parametric edge matching algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 849–865, Nov. 1988.

[20] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, New Jersey: Prentice-Hall, 1982.

[21] Z. Aviad, "Locating corners in noisy curves by delineating imperfect sequences," Tech. Rep. CMU–CS–88–199, Carnegie Mellon University, Dec. 1988.

[22] E. M. Mikhail, *Observations and Least Squares*. New York: Harper and Row, 1980.

[23] W. Förstner, "On the geometric precision of digital correlation," *Proceedings of ISPRS Commission III Symposium*, vol. XXV, pp. 176–189, 1982.

[24] D. I. Havelock, "Geometric precision in noise-free digital images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 1065–1075, Oct. 1989.

[25] J. E. Unruh and E. M. Mikhail, "Mensuration tests using digital images," *Photogrammetric Engineering and Remote Sensing*, vol. 48, pp. 1343–1349, Aug. 1982.

[26] J. C. Trinder, "Precision of digital target location," *Photogrammetric Engineering and Remote Sensing*, vol. 55, pp. 883–886, June 1989.

[27] C. McGlone, "Bundle adjustment with object space constraints for site modeling," in *Proceedings of the SPIE: Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision*, vol. 2486, pp. 25–36, Apr. 1995.

[28] M. Roux, Y. C. Hsieh, and D. M. McKeown, Jr., "Performance analysis of object space matching for building extraction using several images," in *Proceedings of the SPIE: Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision II*, vol. 2486, pp. 277–297, 1995.

# A Shock Grammar For Recognition

Kaleem Siddiqi

Department of Electrical Engineering
McGill University
Montreal, Canada H3A 1Y2

Benjamin B. Kimia

Division of Engineering
Brown University
Providence, RI 02912

## Abstract

*We confront the theoretical and practical difficulties of computing a representation for two-dimensional shape, based on* **shocks** *or singularities that arise as the shape's boundary is deformed. First, we develop subpixel* **local** *detectors for finding and classifying shocks. Second, we show that shock patterns are not arbitrary but obey the rules of a grammar, and in addition satisfy specific topological and geometric constraints. Shock hypotheses that violate the grammar or are topologically or geometrically invalid are pruned to enforce* **global** *consistency. Survivors are organized into a hierarchical graph of shock groups computed in the reaction-diffusion space, where diffusion plays a role of regularization to determine the significance of each shock group. The shock groups can be functionally related to the object's* **parts**, **protrusions** *and* **bends**, *and the representation is suited to recognition: several examples illustrate its stability with rotations, scale changes, occlusion and movement of parts, even at very low resolutions.*

## 1 Introduction

What does it mean to *recognize* an object from its shape? Informally, this implies an identification of the shape with a familiar category or class of objects, Figure 1. This notion of categorization is crucial to many vision tasks, such as searching a database of shapes rapidly, reasoning about the attributes of new or unfamiliar shapes, *etc.* Curiously, whereas this ability to categorize appears to come naturally and effortlessly to humans, it has been extremely difficult to formalize for computers. In this paper, we address the computational aspects of this problem; specifically, we investigate the description of generic shape classes from the mathematical perspective of curve evolution.

Figure 1: These birds are effortlessly grouped into two categories, based on similarity in "form".

Existing proposals for shape representation emphasize properties of its region, *e.g.*, symmetry and thickness [1], or of its boundary, *e.g.*, curvature extrema [20] and inflection points, or of both [2]. An alternate classification is according to those where shape is viewed statically as a combination of *primitives*, *e.g.*, generalized cylinders, versus those where shape is explained developmentally via a set of processes acting on a simpler shape [14]. Returning to the region-based *symmetric axis transform* (SAT) [1], this view has spawned a vast literature on the theoretical and computational aspects of skeletons. However, it is unfortunate that Blum's key insight that the SAT provides for qualitative shape descriptions in terms of "shape morphemes", *e.g.*, disc, worm, wedge, flare, *etc.*, is usually forgotten. Curiously, an evolutionary approach to shape description supports and complements this view, and gives it a sound mathematical foundation [8, 10]. To elaborate, Kimia *et al.* explore deformations of the shape's boundary, a special case of which is deformation by a linear function of curvature $\kappa$:

$$\boxed{\begin{cases} \frac{\partial C}{\partial t} & = (\beta_0 - \beta_1 \kappa)\vec{N} \\ C(s,0) & = C_0(s). \end{cases}} \quad (1)$$

Here $C$ is the boundary vector of coordinates, $\vec{N}$ is the outward normal, $s$ is the path parameter, $t$ is the time duration (magnitude) of the deformation, and $\beta_0, \beta_1$ are constants. The space of all such deformations is spanned by the ratio $\beta_0/\beta_1$ and time $t$, constituting the two axes of the *reaction-diffusion space*. Underlying the representation of shape in this space are a set of *shocks* [11], or entropy-satisfying singularities, which develop during the evolution and are classified into four types, Figure 2 (left): 1) A FIRST-ORDER SHOCK is a discontinuity in orientation of the shape's boundary; 2) A SECOND-ORDER SHOCK is formed when two distinct non-neighboring boundary points collide, but none of their immediate neighbors collapse together; 3) A THIRD-ORDER SHOCK is formed when two distinct non-neighboring boundary points collide, such that the neighboring boundary points also collapse together[1]; and 4) A FOURTH-ORDER SHOCK is formed when a closed boundary collapses onto a single

[1] Whereas third-order shocks are not generic they merit a distinct classification because of their psychophysical relevance [9] and the abundance of biological and man-made objects with "bend-



1063-6919/96 $5.00 © 1996 IEEE

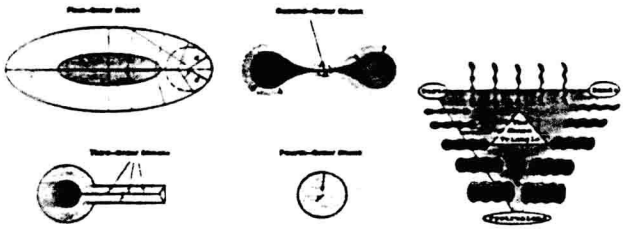


Figure 2: LEFT: The four shock types. RIGHT: The sides of the shape triangle represent continua of shapes; the extremes correspond to the "parts", "bends" and "protrusions" nodes [9].

point. While these definitions are intuitive, they do not easily lend themselves to algorithms for shock detection. A key idea of this paper is that shock computations can be made robust by relying not only on better (subpixel) *local* detectors and classifiers, but also on *global* interactions between shocks, through a shock grammar. In related work, Leymarie and Levine have simulated the grassfire transform using active contours [13]; Scott *et al.* have suggested the use of wave propagation to obtain the full symmetry set [21]; Kelly and Levine have demonstrated the use of annular operators in obtaining coarse object descriptions from real imagery [7]; and Pizer *et al.* have proposed a computational model for object representation via "cores", or regions of high medialness in intensity images [2]. Our work extends the above approaches in a number of ways, which are perhaps best understood in the context of the distinction between shocks and skeletons.

The set of shocks which form along the reaction axis reduces to the traditional skeleton when information regarding *type*, *group*, and *salience* is discarded [23]. However, first, the notion of *type* is essential to capture *qualitative* aspects of shape, leading to generic perceptual shape classes[2] and algorithms for obtaining them, Section 2. Second, the *grouping* of shocks depends not only on their type but also on *sequential*, *geometric* and *topological* constraints obtained from a history of shocks. Section 3. This results in a *hierarchical* representation of shape by shock groups, as illustrated by numerous examples. Section 4. Third, the notion of *salience* connects "nearby" shapes, *e.g.*, Figure 19, providing a foundation for a topology over shape for recognition. In conclusion, we suggest how the shock-based framework might be extended to apply directly to images, Section 6.

like" components, *e.g.*, fingers, limbs, legs of a table, *etc.* Also, they are simultaneously the limit of first-order shocks travelling with infinite speed, but in opposite directions.

[2] First-order shock groups describe "protrusions", second-order shocks occur at "necks", third-order shock groups describe "bends", and viewing the evolution in reverse, fourth-order shocks are *seeds* from which the shape is grown [9].

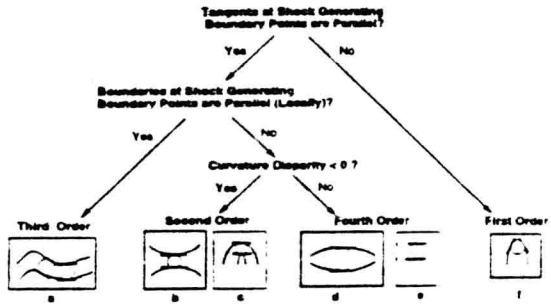



Figure 3: A classification of shock types based on the tangents and the local neighborhood of the two shock generating boundary points. The curvature disparity is the sum of the two (signed) curvatures.

## 2 Shock Classification and Detection: Local Operators

In the design of shock detection operators we face two primary challenges: that of arriving at a complete shock classification scheme which leads to a computational algorithm for detection, and that of obtaining accurate geometric estimates without blurring across singularities. We discuss shock classification and detection in turn.

### 2.1 Classification of Shocks

An intuitive approach is to classify a shock based on properties of the boundary points which collide at it, Figure 3. Whereas this classification provides insight it is difficult to implement directly, *e.g.*, the mapping of a shock to its associated bi-tangent points can become intractable in the presence of multiple nearby topological splits. Alternatively, one may rely on the differential properties of an *embedding surface*, an approach which proves to be computationally efficient and robust. For theoretical as well as numerical reasons, the original curve flow is embedded in the level set evolution of an evolving surface [3, 17], $z = \phi(x, y, t)$:

$$\phi_t + \beta(\kappa)|\nabla\phi| = 0, \tag{2}$$

with the correspondence that the evolving shape is represented at all times by its zero level set $\phi(x, y, t) = 0$. For convenience we take the initial surface $\phi_0$ to be the signed distance function to the shape's boundary (although any Lipshitz continuous function will suffice [3]). The classification of shocks based on differential properties of $\phi$ is summarized in Figure 4 and Table 1. A **first-order shock** corresponds to a discontinuity in the orientation of the tangent $\vec{T}$ to the level curve, computed from $\phi$ as $\arctan(\frac{-\partial\phi/\partial x}{\partial\phi/\partial y})$. Since the colliding boundary points have normals pointing in opposite directions, $|\nabla\phi| = 0$ at **second-, third- and fourth-order shocks.** These shocks can be distinguished from one another by the Gaussian curvature, Table 1. Note that this classification is invariant to the choice of the embedding surface and that all

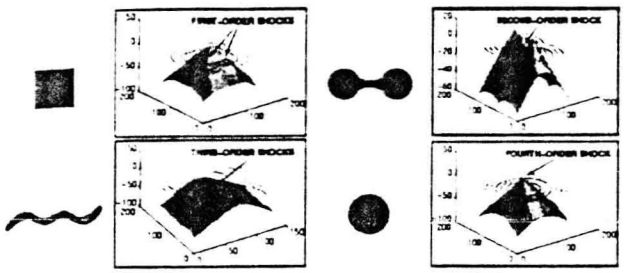


Figure 4: Shock classification based on properties of an embedding surface. TOP LEFT: First-order shocks occur at corners, corresponding to creases on the surface with $|\nabla\phi| > 0$. TOP RIGHT: A second-order shock corresponds to a hyperbolic point with $|\nabla\phi| = 0$. BOTTOM LEFT: Third-order shocks correspond to parabolic points with $|\nabla\phi| = 0$. BOTTOM RIGHT: A fourth-order shock corresponds to an elliptic point with $|\nabla\phi| = 0$.

| Shock Type | Orientation | Curvature |
|---|---|---|
| First | non-vanishing $\nabla\phi$ | high $\kappa$ |
| Second | isolated vanishing $\nabla\phi$ | $\kappa_1\kappa_2 < 0$ |
| Third | non-isolated vanishing $\nabla\phi$ | $\kappa_1\kappa_2 = 0$ |
| Fourth | isolated vanishing $\nabla\phi$ | $\kappa_1\kappa_2 > 0$ |

Table 1: Shock classification based on the gradient $|\nabla\phi|$, the level set curvature $\kappa$, and the principal curvatures $\kappa_1, \kappa_2$ of the surface.

the necessary quantities can be computed locally[3].

## 2.2 Subpixel Shock Detection

We develop a subpixel implementation of the above ideas in order to obtain accurate geometric estimates in the vicinity of discontinuities and to localize shocks. Note that whereas the level set formulation supports subpixel curve evolution an algorithm that only attempts to locate shocks at grid points will suffer from discretization artifacts.

A class of techniques called *essentially non-oscillatory* (ENO) schemes have recently been introduced in the numerical analysis literature to address the problem of inaccurate differential estimates in the vicinity of discontinuities [6]. The basic idea is to select between two contiguous sets of data points for interpolation the one which gives the lower variation, such that at regions neighboring a discontinuity the smoothing is always from the side *not* containing it. By replacing polynomials with *geometric* interpolants: lines, circular arcs, *etc.*, these ideas have been adapted to the 2D problem of locating level curves of an embedding surface while preserving and explicitly placing orientation discontinuities (first-order shocks) [24]. The method provides a subpixel contour tracer (for open and closed curves) which can be used to recover the shape's contour from the evolving embedding surface, Figure 5, and can be extended to higher order shock detection as follows. Recall that $|\nabla\phi| = 0$ at higher order shocks. Therefore, the geometric interpolation method may be used to find $\epsilon$ crossings of $|\nabla\phi|$, Figure 6 (left). However, this approximation always yields 2D *regions* surrounding the putative shock points. As a solution, since $\phi_x$ *and* $\phi_y$ *must each go to zero independently* for $|\nabla\phi|$ to go to zero, the problem can be reduced to two 1D problems by considering zero-crossing curves of $\phi_x$ and $\phi_y$[4], and finding overlaps, Figure 6 (right). This suggests the algorithm for higher order shock detection outlined in Figure 7; further details appear in [23].

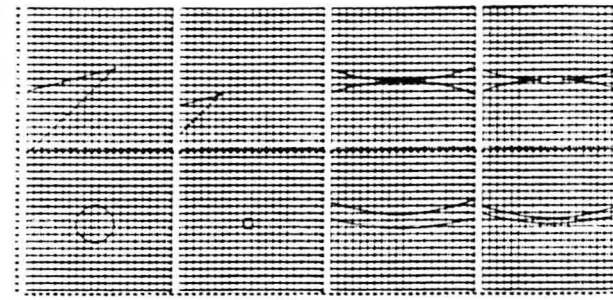


Figure 5: CLOCKWISE FROM TOP LEFT: The *geometric* ENO interpolation technique [24] preserves discontinuities in the vicinity of first- , second-, third-, and fourth-order shocks; gridlines are overlayed and detected corners are marked.

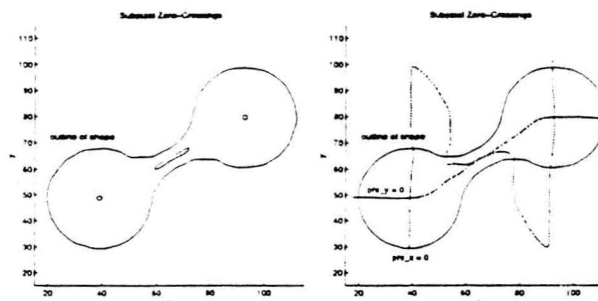


Figure 6: LEFT: The zero crossing contours of $(|\nabla\phi| - \epsilon)$ demarcate regions around the putative shock points. RIGHT: Zero-crossing curves of $\phi_x$ and $\phi_y$ intersect at *exactly* three points, two of which are fourth-order shocks, and one of which is a second-order shock, as determined from the sign of $\kappa_1\kappa_2$.

# 3 Shock Grouping: Global Interactions

The fact that the set of shocks formed under pure reaction ($\beta_1 = 0$) provides the SAT [23] implies that geometric and topological properties that hold for skeletons, *e.g.*, those studied in [4, 22], must hold for shocks as well. We examine three types of constraints on shock formation in Figure 8: *sequential*, *geometric* and *topolog-*

[3] $|\nabla\phi| = (\phi_x^2 + \phi_y^2)^{1/2}$; $\kappa_1\kappa_2 = \frac{\phi_{xx}\phi_{yy} - \phi_{xy}^2}{(1+\phi_x^2+\phi_y^2)^2}$;
$\kappa_1 + \kappa_2 = \frac{(1+\phi_x^2)\phi_{yy} - 2\phi_x\phi_y\phi_{xy} + (1+\phi_y^2)\phi_{xx}}{(1+\phi_x^2+\phi_y^2)^{3/2}}$.

[4] Care must be taken to avoid regions where either $\phi_x$ or $\phi_y$ is identically zero over a neighborhood of grid points. Fortunately, $\phi_x$ and $\phi_y$ cannot both be identically zero over the same regions, since that would imply a 2D region of third-order shocks, which is an impossibility.

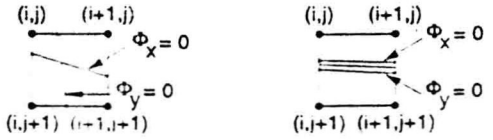



Figure 7: Higher order shock detection based on overlaps of $\phi_x = 0$ and $\phi_y = 0$ within a cell. The four neighboring grid points are marked with filled circles. LEFT: When the two curves pass through the same cell and are not parallel, a second-order or fourth-order shock is placed at the point of intersection (provided that it lies within the cell), based on the sign of the Gaussian curvature. RIGHT: When the two curves pass through the same cell and are close to parallel (their slopes are within 10 degrees), a set of third-order shocks is interpolated as a line drawn through the averaged endpoints.

*real*: the first type pertains to allowed sequences of shocks in a group; the latter two relate to properties of one or more shock groups (smoothness, connectivity, *etc.*). The constraints suggest a course of actions to be taken in order to prune impossible shock configurations and organize survivors into more global structures. Figure 10. Further, the sequential constraints can be concisely described via a shock grammar. Formally, a *grammar* $G$ is a language generating device, which is defined by a quadruple $(V, \Sigma, R, S)$ [12]. Here $V$ is an alphabet divided into two parts, the set of *terminal* symbols $\Sigma \subseteq V$ and the set of *non-terminal* symbols $V - \Sigma$. $S$, the *start* symbol, is an element of $V - \Sigma$ and $R$, the set of *rules*, is a finite subset of $V^*(V - \Sigma)V^* \times V^*$ The grammar operates by beginning with a start symbol and then constructing a string via repeated applications of the rules, i.e., by identifying a substring in the current string which appears on the left hand side of one of the rules, and replacing it with the string that appears on the right hand side of that rule. We introduce a **shock grammar**, $SG$, as follows:

$V = \{S_1, S_2, S_{\bar{3}}, S_4, S_I, S_T, E\}$
$\Sigma = \{S_T\}$
$R = \{S_I \rightarrow S_1 E, S_I \rightarrow S_2 E, S_I \rightarrow S_{\bar{3}} E, S_I \rightarrow S_4, S_1 E \rightarrow S_1 S_1 E, S_1 E \rightarrow S_1 S_{\bar{3}} E, S_1 E \rightarrow S_4, S_2 E \rightarrow S_2 S_1 E, S_{\bar{3}} E \rightarrow S_{\bar{3}} S_1 E, S_{\bar{3}} E \rightarrow S_{\bar{3}} S_T, S_4 \rightarrow S_4 S_T\}$.

The symbols $S_1, S_2, S_4$ represent first-, second-, and fourth-order shocks. $S_I$ is a start symbol, $S_T$ is a terminal, and since third-order shocks never appear in isolation, a group of third-order shocks is an element of the alphabet, denoted by $S_{\bar{3}}$. $E$ represents the end of a growing shock sequence and is used to enforce the requirement that shocks be added only to that end, making the grammar context dependent. Figure 9 illustrates the application of the grammar. Note that whereas the grammar suffices to describe the composition of a shock group, it does not reflect the geometric and topological constraints; this may be possible by embedding the grammar in a graph.

**P1.** First order shocks flow with finite speed, except for a set of isolated points (*e.g.*, initial first-order shocks flowing outwards from a second-order shock).
**P2.** First and third-order shock directions change continuously, i.e., these shock branches cannot have any corners.
**P3.** Second-order shocks are initial and are isolated from other second-order, third-order, and fourth-order shocks.
**R1.** Once formed, a second-order shock must give rise to two first-order shocks that flow out of it. The speed of each first-order shock is infinite.
**P4.** A first-order shock branch can either merge with another first-order shock branch, terminate in a third-order shock branch, or terminate in a fourth-order shock.
**P5.** Two third-order shock branches cannot intersect.
**P6.** A first-order branch can flow into or out of a third-order branch's endpoints, but never into or out of a point that lies in the interior of a third-order branch.
**R2.** A single first-order branch that flows into or out of a third-order branch's endpoints, should maintain continuity of orientation.
**P7.** Fourth-order shocks are terminal.
**P8.** A circle is the only shape described by an isolated (fourth-order) shock. Non-circular shapes cannot have any isolated shocks.
**R3.** For non-circular shapes, each fourth-order shock must have at least one first-order shock branch flowing into it.

Figure 8: Proofs and explanations of propositions P1-P8 and remarks R1-R3 appear in [23]. An *initial* shock is one which may give rise to other shocks, but can have no shocks flowing into it; a *terminal* shock has no shocks flowing out of it, but may have shocks flowing into it.

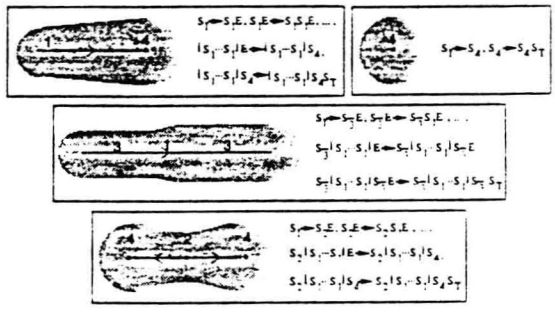


Figure 9: The examples illustrate the construction of different shock groups by repeated application of the rules of the shock grammar.

## 4 Examples

We illustrate the robustness of our two-stage numerical algorithm for shock detection and classification with several examples. The reconstructions are simulations of the "growth" of each shape from its shock-based representation, with linear interpolation of the radius function between successive shocks on the same branch[5]. Figure 11 depicts the evolution of shocks for a

[5]The initial distance transform is blurred very slightly to combat discretization effects, hence the reconstructions have slightly rounded corners.

**A1.** A first-order shock should be appended to the end of an existing first-order shock group so long as it: 1) maintains continuity in position as well as direction of flow with the last shock added to the group, and 2) has finite speed. Otherwise, a new first-order shock branch should be initiated.
**A2.** A second-order shock hypothesis should be discarded if it is not initial, or if it does not subsequently give rise to two outward flowing first-order shock branches. Otherwise it should be kept and identified as the parent of the two first-order shock branches.
**A3.** A single first-order shock branch that intersects a third-order branch, or that terminates or emanates from a third-order shock branch's endpoints without maintaining continuity in orientation, should be discarded.
**A4.** Two third-order shock hypotheses should be grouped together if they are neighbors, and if their orientations are consistent (the shock group has to be smooth). Distinct groups of third-order shocks should not intersect, and any third-order shock that remains isolated should be interpreted as a fourth-order shock.
**A5.** A fourth-order shock hypothesis that is not isolated from other second-, third-, or fourth-order hypotheses should be discarded. A fourth-order shock that is isolated should be interpreted as a circle, otherwise it should be identified as the point of annihilation of the merging first-order branches.

Figure 10: Actions A1-A5 are used to prune impossible shock configurations and organize surviving shocks.

dumbbell shape, leading to its description as two "seed-based" parts (fourth-order shocks) connected at a "neck" (second-order shock), with each part having three protrusions (first-order shock branches). Figure 12 illustrates the robustness of shock detection under rotation and stretching: the structural description of each triangle as a "seed with three protrusions merging onto it" and of each rectangle as a "bend with two protrusions at each end", is preserved. Next, the description of the shape in Figure 13 (top) as a hierarchical collection of protrusions converging onto a single seed is intuitive and can be used for recognition. The representation of the tool in Figure 13 (bottom) is suited to recognition: a different pair of pliers would match the structural description of "two large bends, attached at one end" (the handles) connected to "two smaller protrusions, attached at the other end" (the head); the same pair of pliers would have to match relative shock locations, formation times and velocities as well. Figure 14 illustrates the robustness of the representation in the face of occlusion, movement, and bending of parts: regions remote to the deformations are not affected and a qualitative description as a collection of bends attached to a hierarchy of protrusions emerges throughout. Finally, Figure 15 depicts the shock-based description of two handwritten letters (left) and the computation of shock speed and acceleration (right). In all the examples the shock branches are smooth and the representation allows for precise reconstruction and accurate metric measurements, as well as for qualitative perceptual shape classes. The latter are crucial for the identification of two different shapes as instances of the same category.

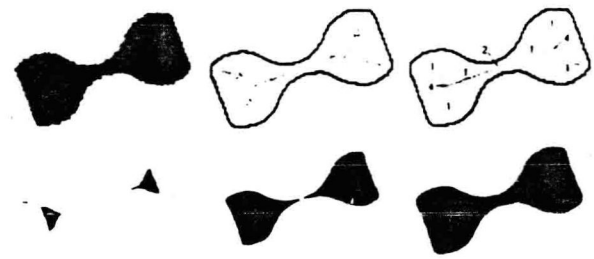


Figure 11: TOP: The evolution of shocks under inward reaction for a rotated dumbbell shape; the arrows depict the velocity of the last shock added to each branch. BOTTOM: The growth of the dumbbell from its shock-based description.

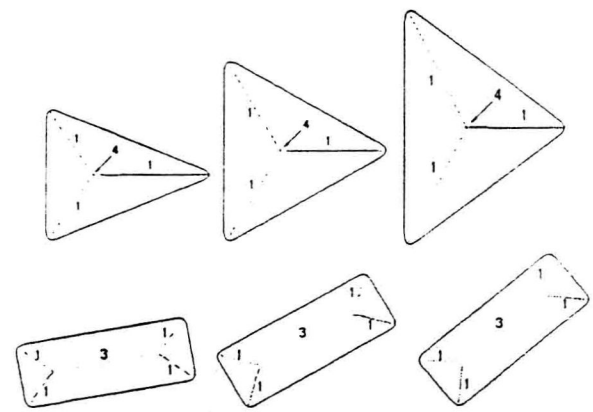


Figure 12: The shock branches remain smooth and no spurious branches are added under rotation or stretching. Further, the structural description of each triangle as "three protrusions converging onto a single seed" and of each rectangle as a "bend with two protrusions at each end" is preserved.

## 5 Structural Diffusion

A variety of approaches have been proposed to deal with the sensitivity of the SAT to boundary details, *e.g.*, blurring to create a multiresolution SAT [19], the use of residual functions [16], and non-linear diffusion of the shape's angle function [18]. Following the theoretical development of [8], the approach we suggest is to use curvature deformation ($\beta_1$) as a smoothing process to assign a significance to each shock group[6]:

**Remark 1 (Significance)** *The significance of a shock group is proportional to its survival with increasing amounts of curvature deformation.*

[6] This choice enforces a number of desirable properties, *e.g.*, in the case of $\beta_1/\beta_0 \to \infty$, any embedded curve will evolve to a round point without developing self-intersections or singularities [5], and the number of extrema and inflection points is non-increasing, implying that no new shock branches can form.

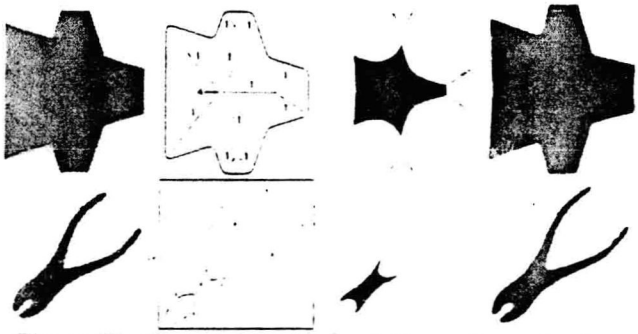


Figure 13: The shock-based description and growth of a shape composed of trapezoids (TOP), and of an industrial shape (BOTTOM). The originals shapes are on the left, and the reconstructions on the right.

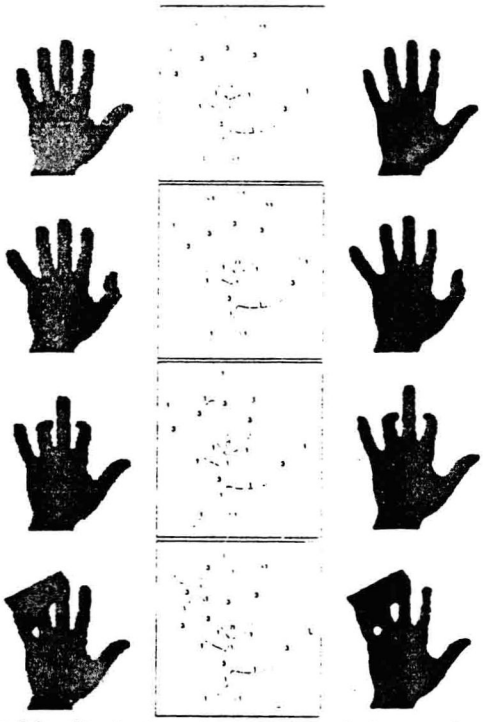


Figure 14: Shock detection under occlusion, and movement/bending of parts. LEFT: The original shapes. MIDDLE: The shock-based description. RIGHT: The reconstruction from shocks.

We consider the effect of diffusion on each shock type; the detection of shocks with diffusion is coarse (not subpixel), and *is only intended to provide a measure of significance for shocks obtained under pure reaction.* When $\beta_i \neq 0$ we interpret a first-order shock as a maxima of (sufficiently high) positive curvature. The survival of a first-order shock group with increasing diffusion reflects the "scale" of the corresponding protrusion, Figure 16

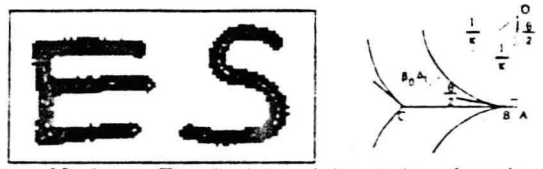


Figure 15: LEFT: The shock-based description of two hand-written letters. RIGHT: First-order shock speed and acceleration. The shock occurs at point $B$, and after one time step has moved to point $C$. With $AB = \kappa^{-1} sin(\theta/2)$, the speed of the shock is obtained as: $s = AB' = \beta_0 / sin(\theta/2)$. The acceleration is obtained by differentiating the speed as: $a = s(\beta_0^2 - s^2)\kappa/\beta_0$.

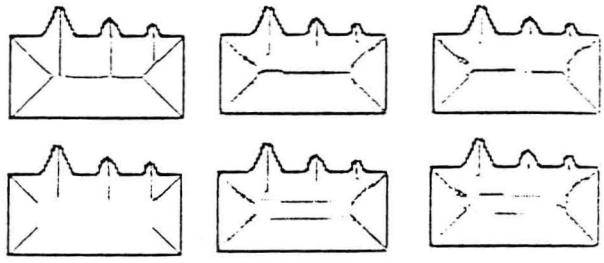


Figure 16: LEFT TO RIGHT: $\beta_0 = -0.2$, $\beta_1 = 0.0, 0.25, 0.5$. Each column depicts the shock groups that have been detected up until the present time, with the evolved shape overlaid. Observe that branches are annihilated in order of the scale of the protrusion they represent.

[7]: the survival of a second-order shock with diffusion reflects how narrow the corresponding neck is, Figure 17; diffusion regularizes bend-like shapes with boundary perturbations [23]; and the survival of a fourth-order shock with diffusion reflects the degree to which it represents a local center of mass for a shape, *e.g.*, compare the rightmost and leftmost fourth-order shocks in Figure 17.

The above notion of significance induces a hierarchical ordering of shock branches from fine to coarse, *i.e.*, branches obtained under pure reaction are removed in the order that they annihilate under diffusion, and the structures that they represent are literally broken off, Figure 16. This brings out the coarse level similarity between shapes belonging to the same category, Figure 19, an essential requirement for recognition.

## 6 Shocks from Images

In conclusion, we suggest that the shock-based representation can be extended to apply to fragmented shapes as they typically arise in real imagery by allowing local edge hypotheses to interact via the evolution of a local embedding surface; recall that any Lipshitz continuous surface can be used. Such a surface can be constructed using the output of an edge operator, *i.e.*, by first placing oriented receptive fields at each edge, Figure 20 (top),

[7] In analogy to the lifetime of a grey-level blob in scale space [15], when two protrusions are nearby the shock branches may merge.