ROAD CENTERLINE VECTORIZATION BY SELF-ORGANIZED MAPPING

Peter DOUCETTE[*], Peggy AGOURIS[*], Mohamad MUSAVI[**], Anthony STEFANIDIS[*]
University of Maine, USA
[*]Department of Spatial Information Science & Engineering
(doucette, peggy, tony)@spatial.maine.edu
[**]Department of Electrical and Computer Engineering
musavi@eece.maine.edu

Working Group III/3

KEY WORDS: Semi-Automated Extraction, Self-Organization, Cluster Analysis, Minimum Spanning Tree.

ABSTRACT

A novel approach to semi-automated road centerline extraction from remotely sensed imagery is introduced. Providing inspiration is Kohonen's self-organizing map (SOM) algorithm. With IFOV < 2m, road features are open to region-based analysis. A variation of the basic SOM algorithm is implemented in a region-based approach to road vectorization from high spatial (1.0m) and spectral resolution imagery. Using spectrally classified road pixels as input, centerline nodes are located via cluster analysis of the local density fluctuations in the input space. Linking the self-organized locations of the nodes with a minimum spanning tree algorithm provides global topological structure, which is subsequently refined. The idea is use contextual analysis from which to derive optimum topology. The result is a vectorized road centerline network suitable for direct GIS database population. Preliminary results demonstrate the algorithm's potential for robust vectorization when presented with noisy input.

1 INTRODUCTION

Road network extraction ranks among the most fundamental of image analysis operations in support of many GIS applications. A complete road extraction task entails a raster-to-vector conversion, which is basically a lossy signal compression operation whose objective is to derive compact geometrical representations for raster features. A primary goal within the Image Understanding (IU) community is the minimization of human effort required to vectorize roads (among other features) from images when populating GIS databases. Despite substantial IU advancements with automated feature extraction techniques over the years, the population of geospatial feature databases remains in large part a costly manual process. With a growing variety of next-generation imaging platforms offering enhanced spatial and spectral resolutions, more robust extraction tools will play a larger role in improving future productivity. With IFOV < 2m, most road features are not characterized as single raster edges. Rather, they manifest as *elongated regions*, a.k.a. 'ribbon' features, and as such become subject to region-based analysis in addition to (dual) edge analysis. However, an inherent dilemma with high spatial resolution is the potential for a decrease in local signal-to-noise ratio, and therein lies the trade-off for higher geometric accuracy. The goal of this paper is to investigate the use of self-organization methods for robust raster-to-vector conversion of road centerlines from high spatial and spectral resolution imagery.

Road extraction algorithms are often categorized according to their degree of automation. Most prevalent are the *semi*-automated methods, which attempt to strike a synergistic compromise between human and machine. Semi-automated methods are further divided into two broad categories. The first includes *line-following* methods, in which local exploratory image filters sequentially trace a minimum cost path from one point to the next (Mckeown and Denlinger, 1988; Vosselman and Knecht, 1995). The second group includes active contour models, i.e., *snakes* (Kass et al., 1988), which are adapted for 2D and 3D road extraction (Trinder and Li, 1995; Neuenschwander et al., 1995; Gruen and Li 1997). Snakes are fundamentally edge-based curvilinear feature extractors that are distinct from line-following in that they exemplify 'simultaneous' curve-fitting as opposed to sequential. Notwithstanding, either method is considered semi-automatic owing to their requirement of user provided seed point inputs. On the other hand, *fully* automated methods attempt to completely circumvent user intervention during the extraction process. A considerably more rigorous approach, it usually requires a skillful integration of contextual information and *a priori* knowledge into the road extraction task (Mckeown et al., 1985; Baumgartner et al., 1999). Still, a basic concern of fully automated methods is limited generalization ability, particularly as *a priori* knowledge becomes more specialized with scene complexity.

This paper presents a semi-automated clustering-based approach to road centerline vectorization from high resolution imagery that represents a variation of Kohonen's *self-organizing map* (SOM) algorithm (Kohonen, 1997). The neurobiologically inspired SOM algorithm belongs to the unsupervised learning (clustering) variety of artificial neural networks (ANN). While cluster analysis is often used as a form of unsupervised spectral classification, the interest here is to use cluster analysis to find and map spatial structure within spectrally classified road pixels. The spectral classification step is what renders the method semi-automated. Beyond this, we assume that no ancillary cartographic information is available, e.g. DEM or coarse-resolution vectors. From a binary image of spectrally classified road pixels, localized road centerline nodes are mapped via an iterative clustering method. The nodes self-organize according to the local density fluctuations of the input, and global structure is provided by linking cluster centers with a minimum spanning tree (MST) algorithm. The MST cost function incorporates global cluster statistics to deal with image noise. The subsequent tree structure is densified and pruned locally as needed, and candidate loops are closed where appropriate. Finally, nodes are ordered for topological consistency, and a vectorized road network is available for direct GIS database entry. Since clustering exemplifies a local center-of-gravity approach, it is not sensitive to edge definition of road features, and allows for more meaningful exploitation of multispectral imagery versus edge-based methods.

2 METHODS FOR SELF-ORGANIZATION

Cluster analysis is a basic method of self-organization. It describes a regression solution that optimally partitions an m-dimensional input space containing a set of N vectors, $\mathbf{P} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$, where $\mathbf{X}_i = [x_{(i,1)}, x_{(i,2)}, \ldots, x_{(i,m)}]^T \in \Re^m$, into a representative set of K cluster centers, or "codebook" vectors, $\mathbf{C} = \{\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_K\}$, where $\mathbf{W}_i \in \Re^m$. Each $\mathbf{W}_i$ represents a single "codeword" in the codebook, and in general $K << N$, where K must be prespecified. Cluster analysis is sometimes referred to as *vector quantization* (VQ). Usage of the latter term is generally associated with signal compression, and the former with pattern recognition. In the present context, we are interested in aspects of both. In (Linde et al., 1980) an important distinction is drawn between aspects of cluster analysis and VQ concerning the intended application. For example, one application may require the design of a vector codebook from a set of training vectors, which are subsequently fixed for use in quantizing (classifying) new vectors outside of the training set, e.g., test vectors. In contrast, another application may use the codebook only for purposes of pattern detection within the training set; i.e., the training set *is* the entire set. Our road extraction strategy is an example of the latter, where the dimensionality of the input vectors is $m = 2$ for (x, y) pixel coordinates.

2.1 Iterative Optimization

Iterative optimization is the most widely used approach to cluster analysis. The basic idea is to iteratively adjust a set of initialized K codebook vectors in such a way as to globally improve a criterion function, e.g., sum-squared-error (SSE). The limitations associated with this "hill-climbing" technique include a guarantee of only local optimization (not global), and sensitivity to the initial quantity and location of the codebook vectors. Despite these shortcomings, the simplicity of implementation and its computational tractability makes iterative optimization a viable approach for many applications, including the current one. We consider two familiar techniques: K-means and Kohonen learning.

2.1.1 Batch Codebook Updating. The most recognizable clustering method by name is perhaps the K-means algorithm (MacQueen, 1967). It is also identified as the generalized Lloyd algorithm (GLA) based on an earlier work by Lloyd, which was formally published in (Lloyd, 1982). For completeness, we list the steps of the K-means algorithm:

1. Initialize K codebook vectors in the input space.
2. Partition the samples according which codeword each sample is closest to; ties resolved arbitrarily.
3. For each partition, compute the mean of its associated samples from the previous step.
4. Iterate from step 2 until a stopping criterion is satisfied.

Though many different measures are cited in the literature, the L_2 (Euclidean) norm is a standard choice for the similarity measure of step 2. The L_1 norm is sometimes used to improve computational performance. K-means is considered a batch method because the entire codebook is updated at each iteration, which differs from the sequential method of Kohonen discussed next.

2.1.2 The Self-Organizing Map (SOM) is the artificial neural network based on competitive learning developed by (Kohonen, 1982). It is distinct from K-means in three respects: 1) codewords are updated sequentially (as opposed to batch), 2) codewords are spatially ordered in the input space according to a predefined topology in a independent network space, and 3) use of a smoothing kernel and learning rate schedule during codeword updating. A primary goal of the SOM is to define a *mapping* from an input space $\Re_I^m$ onto a network space $\Re_N^d$, where $m \geq d$, and typically $d \leq 2$. Fig. 1 illustrates the topological ordering of a simple one-dimensional SOM network with four *neurons*, and a two-

dimensional input. Each connection or *synapse* between a component of a sample vector **X** and any single neuron has an associated 'weight'. Collectively, the weights represent the codebook. "Winning" codeword(s) are determined for each input sample based on similarity metric, and subsequently updated according to Kohonen's learning rule,

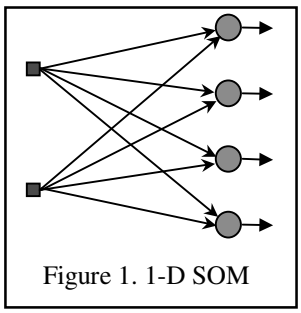


Figure 1. 1-D SOM

$$\mathbf{W}_i(n+1) = \mathbf{W}_i(n) \; + \; \eta(t) \cdot h_q(t) \cdot \big(\mathbf{X}(n) - \mathbf{W}_i(n)\big). \tag{1}$$

The n-th sample drawn is represented by $\mathbf{X}(n)$; $\mathbf{W}_i(n)$ is a codeword at the n-th iteration, and $\mathbf{W}_i(n+1)$ is the updated codeword. A time variable t is measured in *epochs*, each of which represents a complete presentation of all input samples to the network. Therefore, each epoch is the SOM analog to a single iteration in the K-means. A learning rate function, $\eta(t)$, is defined as $0<\eta(t)<1$, and $h_q(t)$ is a neighborhood function centered on winning codeword q, and defined as $0 < h_q(t) \leq 1$. As $t \rightarrow \infty$, $h_q(t) \rightarrow 0$ (for all but the winning codeword) and $\eta(t) \rightarrow 0$. Depending upon $\eta(t)$, the sequential updating of codewords in the SOM lead to a generally smoother asymptotic convergence relative to the batch updates of K-means. A batch version of the SOM is described in Kohonen (1993).

2.2 Simulations

Let $p(\mathbf{X})$ describe the probability density function for a feature in an image space $\Re^2$ as shown by the gray point samples in fig. 2. Fig. 2a shows a randomly initialized codebook for K = 10, where the codewords are depicted as nodes linked according to their sequence. Although many initialization strategies exist in the literature (Duda and Hart, 1973; Gersho and Gray, 1992), the current one is used for demonstrational purposes which will become apparent. Fig. 2b shows the result of K-means clustering, where each node represents a local centroid. Notice that only half of the available nodes have been used in describing $p(\mathbf{X})$. This is not necessarily problematic since such "dead" nodes are relatively simple to identify. However, if the dead nodes {2, 4, 7, 8, 10} are eliminated, K-means still does not provide a way of meaningfully linking the active nodes {1, 3, 6, 9, 5} with respect to vectorizing (delineating) the feature's medial axis. Ordering may be derived from subsequent procedures, but it is built in to the SOM algorithm.

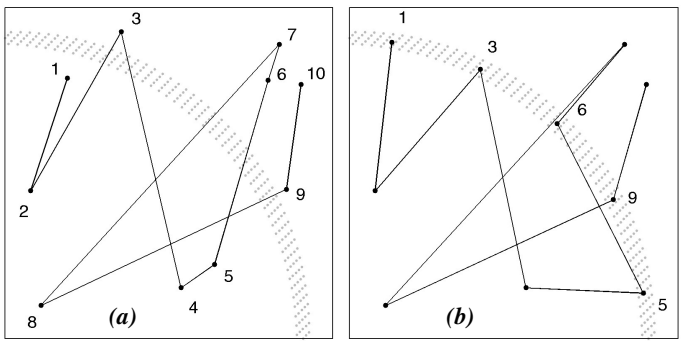


Figure 2. K-means (a) initialization; (b) convergence

Given the same initial configuration of fig. 2a, let the node sequence be represented by a one-dimensional SOM network. Fig. 2c-d demonstrate the ordering and refinement phases respectively of the SOM algorithm. During the process the neighborhood function progressively shrinks until only the winning node is updated. At this point the SOM becomes a simple winner-take-all (WTA) network, which is the sequential analog to the K-means. At convergence, the codewords are sequentially linked to delineate the feature's curvilinear axis as a kind of vectorized skeletonization of $p(\mathbf{X})$. However, as scene complexity increases, the SOM approach generally requires *a priori* information to preserve topological consistency in the input space (Doucette et al., 1999).

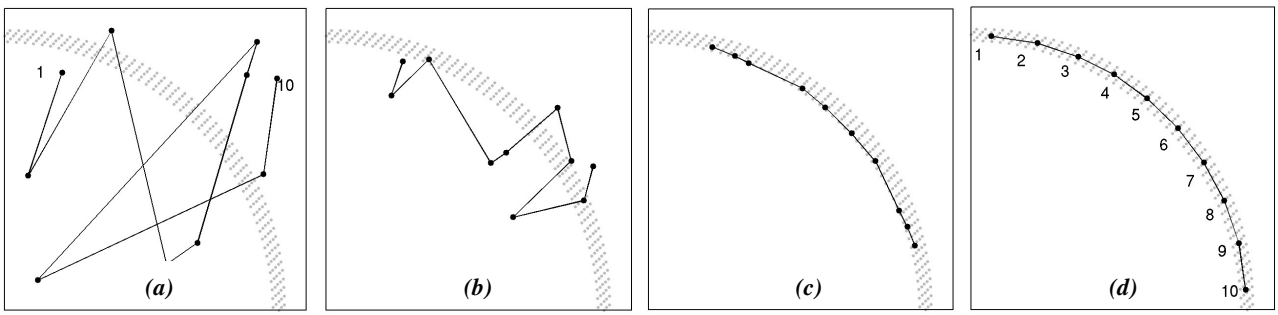


Figure 3. SOM learning (a) random initialization; (b) after a few iterations; (c) ordering complete; (d) refinement

3 EXPERIMENTS WITH IMAGERY

The test image shown in fig. 4 is a 1.0m HYDICE strip flown over the U.S. Army's Yakima Training Center (YTC), located near Yakima, Washington. The HYDICE aerial sensor is a push broom system that generates images with a spectral resolution of 210 bands within the range of 0.4-2.5µm (10nm nominal bandwidth). An optimized band reduction technique described in Lundeen et al. (1996) was applied by the provider to condense the original image to 44 bands. A relatively rural 300x300 pixel test area is used in this study are so indicated in fig. 4. An appropriate band is selected from each of the RGB regions to simulate a true color image.

Figure 4. Selected bands from a 1.0m HYDICE strip with 300x300 pixel test area outlined.

3.1 Spectral Classification

Supervised spectral classification is a relatively time consuming image processing step. The trade-off for a time-economical approach is generally somewhat higher noise levels in the classification results. The idea was to input the results from a relatively 'low-cost' spectral classification of the image into our road vectorization algorithm to determine how robustly it could perform. A maximum likelihood classification (MLC) was performed on the first three layers from a PCA transformation of the 44-layer image. The eigenvalues indicated an accounting of 99% of the image variance in the first three PCs. Fig. 5 shows the spectral classification results for the paved road class that is used as input for our vectorization algorithm. Presence of both clutter and omission noise is apparent. The former is due to spectrally inseparable samples of rooftops and driveways, and the latter is largely attributable to a decrease in feature homogeneity with higher spatial and spectral resolution.

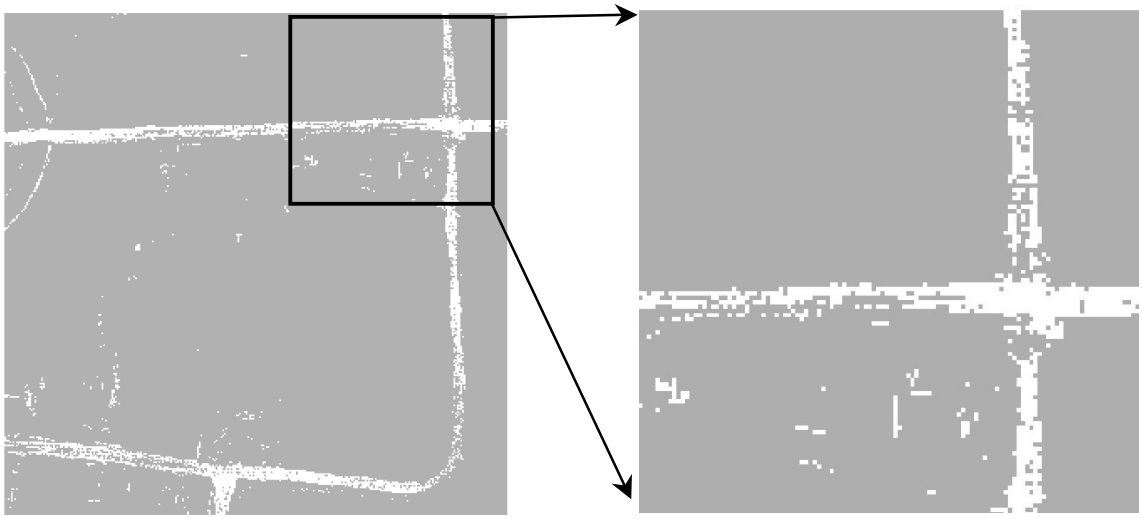
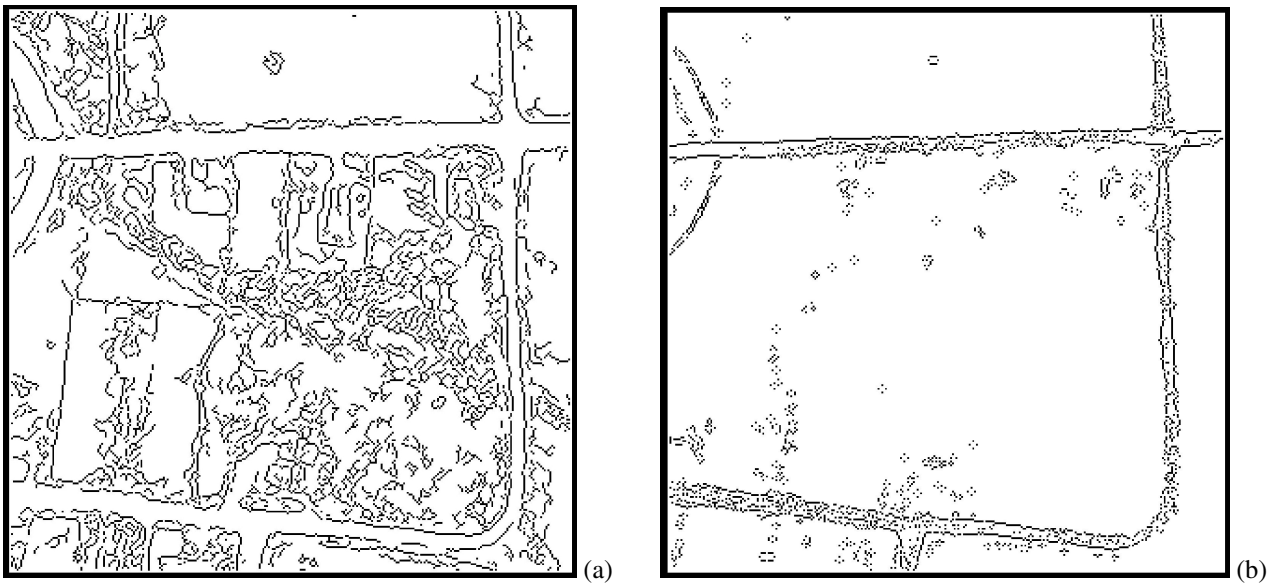


Figure 5. Paved road class pixels from unedited maximum likelihood classification

3.2 Edge Detection

When the spatial resolution of an image is such that road features manifest as elongated regions instead of single edges, edge-based extraction methods often impose the additional constraint of anti-parallelism (Neuenschwander et al., 1995). Another approach is to extract single edges from reduced resolution layers, and fuse the multi-scale space results (Baumgartner et al., 1999). These methods can perform quite well, yet may carry some extra algorithmic overhead. In general, edge-based road extraction requires reasonably well-defined edges. The Canny operator edge detection results in fig. 6 demonstrate that edge definition can be locally problematic in high spatial and spectral resolution images. Moreover, the stream feature in the upper left can be easily misconstrued as a road in this case. We conjecture that edge detection methods in general benefit relatively little from the additional spectral information contained in multispectral versus single layer images. On the other hand, region-based methods may have potentially more to gain from additional spectral information, which is a premise of our approach.



(a) (b)

Figure 6. Canny edge detection from (a) PC1 of original image; (b) MLC road class binary image

3.3 Deriving Topology from Spatial Clusters

3.3.1 MST and *K*-means. Graph-theoretic methods have been suggested as an effective means of deriving structural information for curvilinear features (Zahn, 1971; Suk and Song, 1984; Dobie and Lewis, 1993) or elongated regions (Lipari et al., 1989) via selective linkaging of samples. Similarly, Kangas et al. (1995) describes a variation of the SOM in which the mimimum spanning tree (MST) algorithm is used as the neighborhood criterion between codewords in sample space to enhance the mapping of certain feature types. Inspired by this variation, we employ MST to link codewords (post-convergence) as opposed to individual samples to derive road topology. But unlike Kangas, *K*-means clustering is used instead of Kohonen learning. The incentives for doing so include a generally faster convergence, no learning rate or neighborhood parameter requirements, a lighter computational load, less susceptibility to local minima traps, and less dependence on the order of sample presentation. (The 'arbitrary' tie rule in step 2 makes *K*-means sensitive to sample presentation order.)

Fig. 7 demonstrates the vectorization process with the road class binary image of test area 1 as input. In fig.7a the codebook is initialized as a grid in the input space. Such initialization achieves an efficient balance between final cluster size uniformity and algorithmic convergence time. An advantage with batch codeword updating is that the median or trimmed mean can be easily substituted for the mean computation in step 3 of *K*-means to enhance noise tolerance. Although this adds slightly to computational requirements, it is particularly effective for samples exhibiting strong central tendencies, as is the case here. This reveals yet another limitation of sequential updating, which intrinsically finds the local simple mean. Fig. 7b shows the codebook convergence using '*K*-trimmed means' clustering. Convergence is fast, requiring $\approx$ 20 iterations within 10 seconds from an uncompiled script routine on a 300Mhz Pentium. Thin dotted lines indicate the Voronoi tessellation polygons (the dual of Delaunay triangulation) for the nodes, which provide a helpful visualization of each codeword's "influence region" in the input space. At this stage, dead and weak codewords are determined for removal from further consideration. To assess relative cluster strength, eq. (2) shows the normalized cluster scatter measure used, where $\mathbf{S}_i$ is the scatter matrix for the ith codeword, $\mathbf{C}_i$.

$$\left[\frac{1}{N_i^2}\right] \mathrm{tr}\,\mathbf{S}_i, \quad \text{where}\,\mathbf{S}_i = \sum_{\mathrm{X}\in\mathrm{C}_i} (\mathrm{X} - \mathrm{W}_i)(\mathrm{X} - \mathrm{W}_i)^t \ , \tag{2}$$

The MST linked codebook is displayed in fig. 7c. A link strength assessment between codewords is incorporated into the MST cost function as a cross sectional scatter measure similar to eq. (2). In this case, $\mathbf{W}_i$ represents a midpoint vector between two codewords. The idea is to use global cluster statistics to help determine codeword usage and link thresholds dynamically versus *a priori* hard setting. Dashed connections reveal weak links found during the first MST pass. Since a MST by definition cannot form closed loops, loop closing must occur as a subsequent step that is based on the cross sectional measure between identified end nodes. The overall objective is to isolate a 'trunk' of the MST that corresponds to the road network via pruning the 'noise' branches. The use of the trimmed mean in *K*-means allows for fairly robust road centerline estimates without the support from anti-parallel edges. The extraction of the 3-way intersection reveals a shortcoming with local optimization in clustering, which must be corrected in a subsequent procedure. Finally, fig. 7d shows the results from a codeword resolution adjustment and topology-checking procedure. Similar results have been achieved with comparable test areas from the HYDICE strip.

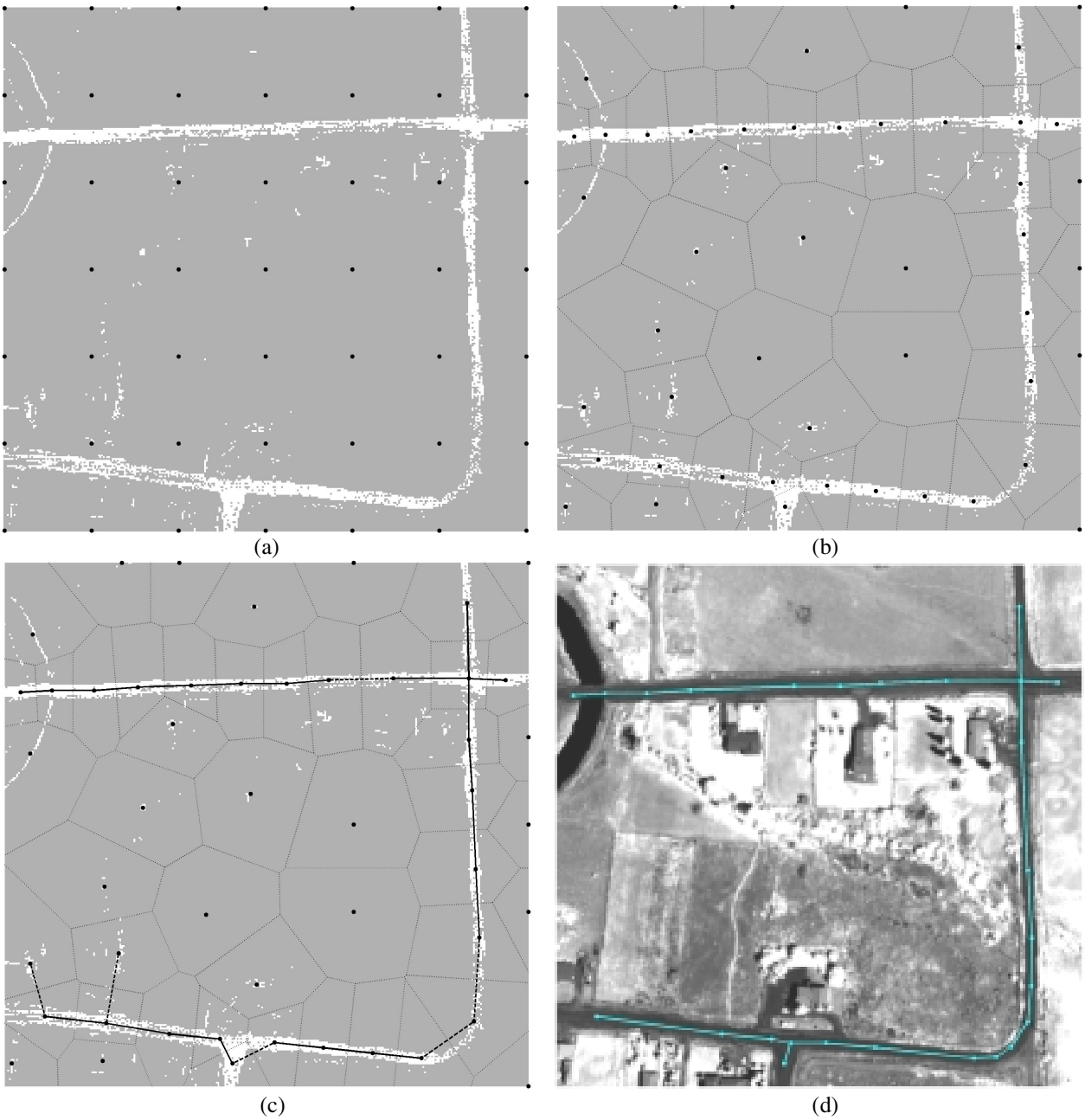


Figure 7. (a) 7x7 grid initialization; (b) clustering convergence; (c) first MST pass; (d) node resolution adjustment and topology checking

3.3.2 Dynamic Adjustment of *K*. The significance of K is that it represents the resolution of centerline vectorization, i.e., the local "shape-point" density. A well-known limitation of all clustering algorithms is the need to prespecify K, whose optimum value is unknown at the outset. The goal is to arrive at an optimal value, K_{opt}, dynamically. One option is to start with an initial K_i such that $K_i < K_{opt}$, and allow it to grow by selectively inserting new codewords locally. Since algorithmic execution time is appreciably sensitive to K, a potential advantage of this approach is improved computational performance (Fritzke, 1993; Rodrigues and Almeida, 1991). The alternative is to overestimate as $K_i > K_{opt}$, and merge codewords accordingly. Our strategy includes aspects of both since K_i varies locally with respect to K_{opt}. Since K-means uses only the closest local codewords, many dead nodes will result from an initial codeword grid configuration for $K_i > K_{opt}$. This mitigates the computational efficiency issue somewhat as uniform noise levels decrease. The objective is to select K_i more or less arbitrarily within an appropriate range, and densify codewords in regions of high road curvature as needed. Sections of low curvature are accordingly pruned. An angle threshold is applied to node-linked triplets post-MST to identify candidate insertion and deletion locations. Insertion codewords are initialized at the midpoints between existing codewords. The new triplet is allowed to readjust via local K-means while the remaining codebook is locked into position. This method represents a 'true' curve fitting as opposed to an interpolative estimation. Fig. 7d illustrates the final codebook adjustment. Following this, the codewords are sequentially ordered at the component level. Component end points are identified by codewords that possess more or less than two links, and direct GIS database entry of road centerline topology is possible.

4 CONCLUSIONS

A clustering-based approach to semi-automated road centerline vectorization from high spatial and spectral resolution imagery is investigated. Inspired by a variation of Kohonen's self-organizing map algorithm, it substitutes K-means clustering for Kohonen learning. K-means describes more feature detail with less computational load, while the basic SOM is intended for a comparatively generalized resolution of abstraction that is less sensitive to noise. Requirements of high positional accuracy for extracted centerline information ultimately dictated the approach taken. With IFOV < 2m, most road features are open to region-based analysis. Since clustering analysis exemplifies a center-of-gravity approach, it is not sensitive to edge definition. This allows for more meaningful exploitation of multispectral imagery versus edge-based methods.

The MST algorithm is used as the primary mechanism for linking convergent cluster centers (codewords). The MST cost function incorporates global cluster statistics to contextually differentiate image noise from the signal. The idea is to exploit global information to render decision-making more dynamic, which local road-following methods may lack. Preliminary experiments with imagery demonstrate a potential for robust road vectorization results suitable for direct GIS database population.

The goal of future research is to develop a fuzzy system that integrates more rigorous topology-checking methods from contextual road models in the literature. A further goal is to investigate this approach as a method for infusing topological information into the spectral classification step via a feedback loop. This concept holds the potential for fully automated processing that integrates curvilinear vectorization with feature recognition.

ACKNOWLEDGEMENTS

This work was partially supported by the National Science Foundation through CAREER grant number IIS-9702233 and through grant number SBR-8810917. Also, the National Aeronautics and Space Administration through NASA grant fellowship number MSTF 99-59.

REFERENCES

Baumgartner, A., Steger, C., Mayer, H., Eckstein, W., Ebner, H., 1999. Automatic Road Extraction Based on Multi-Scale, Grouping, and Context. Photogrammetric Engineering and Remote Sensing, 65(7), pp. 777-785.

Dobie, M., Lewis, P., 1994. Extracting Curvilinear Features from Remotely Sensed Images Using Minimum Cost Path Techniques. Proceedings of the First IEEE International Conference on Image Processing, v 3, pp. 231-235.

Doucette, P., Agouris, P., Musavi, M., Stefanidis, A., 1999. Automated Extraction of Linear Features from Aerial Imagery using Kohonen Learning and GIS Data. Lecture Notes in Comp. Sci., v. 1737, Springer-Verlag, pp. 20-33.

Duda, R., Hart, P., 1973. Pattern Classification and Scene Analysis. John Wiley and Sons, New York.

Fritzke, B., 1993. Kohonen Feature Maps and Growing Cell Structures- a Performance Comparison, Advances in Neural Information Processing Systems 5, Hanson, S., Cowan, J., Giles, C. (eds.), Morgan Kaufmann, pp. 123-130.

Gersho, A., Gray, R., 1992. Vector Quantization and Signal Compression. Kluwer Academic Publishers.

Gruen A., Li H., 1997. Semi-Automatic Linear Feature Extraction by Dynamic Programming and LSB-snakes. Photogrammetric Engineering & Remote Sensing, 63(8), pp. 985-995.

Kangas, J., Kohonen, T., Laaksonen, J., 1990. Variants of the Self-Organizing Map. IEEE Trans on Neural Networks, 1, pp. 93-99.

Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes: Active Contour Models, Int J. of Comp Vision, 1(4), pp. 321-331.

Kohonen, T., 1997. Self-Organizing Maps (2nd Ed.). Springer-Verlag.

Kohonen, T., 1993. Things You Haven't Heard about the Self-Organizing Map. Proceedings of the IEEE International Conference on Neural Networks (San Francisco), pp. 1464-1480.

Kohonen, T., 1982. Self-Organized Formation of Topologically Correct Feature Maps. Biological Cybernetics, 43, pp. 59-69.

Lloyd, S., 1982. Least Squares Quantization in PCM. IEEE Transactions on Information Theory, IT-28, pp. 127-135.

Linde, Y., Buzo, A., Gray, R., 1980. An Algorithm for Vector Quantizer Design. IEEE Trans on Comm, COM-28, pp. 84-95.

Lipari, C., Mohan, T., Harlow, C., 1989. Geometric Modeling and Recognition of Elongated Regions in Aerial Images. IEEE Transactions on Systems, Man, and Cybernetics, 19(6), pp. 1600-1612.

Lundeen, T., Heasler, P., Petrie, G., 1996. Automatic Band Selection for Sensor Optimization. PNNL-11360, Pacific Northwest National Laboratory, Richland, Washington.

MacQueen, J., 1967. Some Methods for Classification and Analysis of Multivariate Observations. Proceedings of the Fifth Berkeley Symposium on Math. Stat. and Prob., v 1, pp. 281-296.

Mckeown, D., Harvey, W., McDermott, J., 1985. Rule-based interpretation of aerial images. IEEE Transactions on Pattern and Machine Intelligence, PAMI-7(5), pp. 570-585.

Mckeown, D., Denlinger, J., 1988. Cooperative Methods for Road Tracking in Aerial Imagery. IEEE Proceedings on Computer Vision and Pattern Recognition, Ann Arbor, MI, pp. 662-672.

Neuenschwander, W., Fau, P., Szekely, G., Kubler, O., 1995. From Ziplock Snakes to Velcro Surfaces. Automated Extraction of Man-Made Objects from Aerial and Space Images, Birkhauser Verlag, pp. 105-114.

Rodrigues, J. Almeida, L., 1991. Improving the Learning Speed in Topological Maps of Patterns. Neural Networks: Advances and Applications, Gelenbe, E. (ed.), North-Holland, pp. 63-78.

Suk, M., Song, O., 1984. Curvilinear Feature Extraction Using Minimum Spanning Trees. Computer Vision, Graphics, and Image Processing, 26, pp. 400-411.

Trinder, J., Li, H. 1995. Semi-automatic Feature Extraction by Snakes. Automatic Extraction of Man-Made Objects from Aerial and Space images, Birkhauser Verlag, pp. 95-104.

Vosselman, G., Knecht, J., 1995. Road Tracing by Profile Matching and Kalman Filtering. Automatic Extraction of Man-Made Objects from Aerial and Space images, Birkhauser Verlag, pp. 265-274.

Zahn, C., 1971. Graph-Theoretic Methods for Detecting and Describing Gestalt Clusters. IEEE Transactions on Computers, C-20(1), pp. 68-86.