Detection and Location of Objects from Mobile Mapping Image Sequences by Hopfield Neural Networks

Rongxing Li, Weian Wang, and Hong-Zeng Tseng

Abstract

Detection and location of objects is a challenging issue in mobile mapping data processing for reducing human operations and enhancing efficiency, considering the vast amount of data and information acquired by mobile mapping systems. This paper describes research results of algorithms based on Hopfield neural networks for utility object detection and location. Specifically, street light poles are modeled in the three-dimensional (3D) scene domain and detected by the network with neurons formed by vector edge features from the model and the mobile mapping images. The established Hopfield neural network is able to detect light poles at specific locations. It can also be used to detect and locate all light poles from a mobile mapping sequence, regardless of their positions. Such automation is particularly important for automatic generation of special layers in a utility GIS, for example, traffic signs, fire hydrants, road centerlines, and others. The developed algorithms and implementation results are described.

Introduction

Mobile mapping technology has demonstrated an innovative way for large-scale spatial data acquisition (Bossler *et al.*, 1991; Li *et al.*, 1994; Novak, 1995; Schwarz and El-Sheimy, 1996; He, 1996; Li, 1997). The rich information contained in the mobile mapping image sequences can be extracted and used for generating spatial databases in a wide range of applications. Despite many advantages, current mobile mapping data processing techniques are limited by the following factors: (1) most data processing procedures are manual, (2) the great scale variation in images causes difficulties in accurate photogrammetric measurement and object recognition, and (3) geometric constraints provided by navigation sensors are not fully utilized.

Automation of mobile mapping image processing has been prioritized as an important research issue ever since the early development stage of the technology. He and Novak (1992) initiated research on automatic analysis of highway features from images. Tao *et al.* (1998) presented an improved approach by using a "snake" based physical model. Enhancement of the point accuracy by selecting an optimal set of images along a mobile mapping sequence was reported by Li *et al.* (1996). Multiple image matching techniques were applied in processing the image sequences by Xin (1995) and Braess (1997). Such research efforts improved the accuracy and efficiency of mobile mapping image processing. However, the automation is supported by primary features such as pixels and extracted lines. Thus, the level of automation and reliability of results may be limited. In contrast, approaches using neural networks recognize objects from images considering characteristics of objects at the object level (Lin *et al.* 1991; Bishop 1995).

Object detection and location from stereo images have been researched based on principles of pattern recognition and photogrammetry. Human operators perform object recognition most efficiently, although in some cases object location can be automated by digital image matching and triangulation techniques. It is so far not possible to fully simulate the biological process of object recognition by a computer system. Computational approaches have been used in artificial neural networks. In fact, Hopfield neural networks demonstrated promising potential in finding corresponding primitives and objects from stereo images (Hopfield and Tank, 1985; Li and Nasrabadi, 1989; Lin *et al.*, 1991; Zhang, 1996; Tseng *et al.*, 1997). In the present research, it was expected that a vector-based Hopfield neural network would solve some of the problems faced by traditional raster-based pattern recognition methods when applied in mobile mapping data processing. The objectives were

- To develop algorithms for automatic detection of utility objects from mobile mapping images,
- To study neural networks and apply them to recognition at object level, and
- To integrate geometric constraints existing in mobile mapping image sequences.

This paper presents the algorithms based on Hopfield neural networks considering geometric constraints that are derived from GPS/INS data and mobile mapping image sequences. Specifically, street light poles are modeled in the three-dimensional (3D) scene domain and detected by the network with neurons formed by vector edge features from the model and the mobile mapping images. The Hopfield neural network is able to detect light poles at specific locations. It can also be used to detect and locate all light poles from a mobile mapping sequence, regardless of their positions. Such automation is particularly important for automatic generation of special layers in utility GIS, for example, traffic signs, fire hydrants, road centerlines, and others. The developed algorithms and implementation results are described.

Department of Civil and Environmental Engineering and Geodetic Science, The Ohio State University, Columbus, OH 43210-1275 (li.282@osu.edu).

Weian Wang is presently with Department of Surveying, Tongji University, Shanghai, China.

Photogrammetric Engineering & Remote Sensing, Vol. 65, No. 10, October 1999, pp. 1199–1205.

0099-1112/99/6510–1199$3.00/0

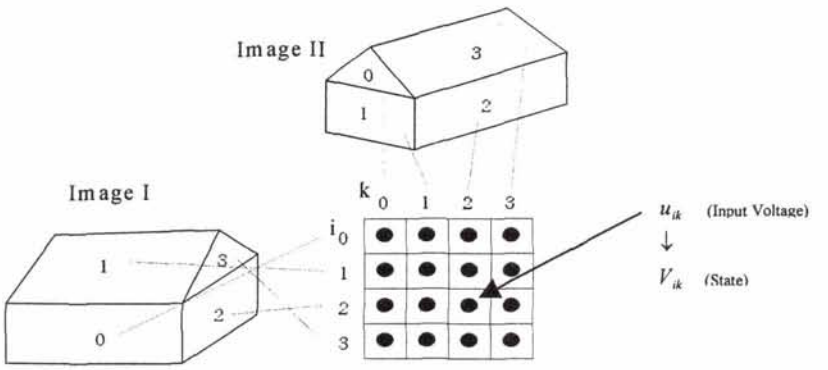


Figure 1. A Hopfield neural network for recognition of corresponding face features.

Mobile Mapping Image Sequences

A mobile mapping system is usually equipped with a mobile platform, navigation sensors, and mapping sensors. The mobile platform may be a land vehicle, a vessel, or an aircraft. Generally, navigation sensors, such as Global Positioning System (GPS) receivers, vehicle wheel sensors, and inertial navigation systems (INS), provide both the track of the vehicle and position and orientation information of the mapping sensors. Objects to be mapped are sensed directly by mapping sensors, for example, charge-coupled-device (CCD) cameras, laser rangers, and radar sensors. Because the orientation parameters of the mapping sensors are directly supplied by the navigation sensors, complicated computations such as photogrammetric triangulation are reduced or avoided. Spatial information regarding the objects is extracted directly from the georeferenced mapping sensor data by integrating navigation sensor data. In this paper, our focus is on a typical land-based mobile mapping system with a land vehicle, GPS receivers, an INS unit, and a pair of synchronized CCD cameras. Stereo image pairs taken by such a system form a georeferenced image sequence with known camera orientation parameters, namely, the exposure center positions and camera attitudes in the scene domain. Interactive measurements in the image domain lead to object positions in the scene domain (Li *et al.*, 1994). The algorithms introduced in this paper will automate the measuring procedure by object detection and subsequent object location.

Hopfield Neural Networks and Object Recognition

So far it has not been possible to fully simulate in a computer system the human biological process of object recognition. Instead, computational approaches have been used in artificial neural networks. The vector-based Hopfield neural network applied in this paper is based on the network models published by Hopfield and Tank (1985) and Lin *et al.* (1991). A Hopfield neural network has the following key behavior characteristics: (1) it is completely described by a state vector $\mathbf{V} = (v_1, v_2, \ldots, v_n)$ of all neurons, (2) there are a specific set of stable states $\mathbf{V}_s = (v_1, v_2, \ldots, v_n)$ that correspond to the stored patterns, and (3) the system evolves in time from any arbitrary starting state $\mathbf{V}_0$ to the stable state $\mathbf{V}_s$ by decreasing its energy E. The Hopfield neural network is built from a single layer of neurons (units), with feedback connections from each unit to every other one (except itself). The change of the unit states is associated with an energy function. It uses a two-dimensional array for storing the neuron states $\mathbf{V}$ (Figure 1). The rows represent features of Image I, for instance, faces numbered $i = 0, 1, 2$, and 3. The columns represent face features of Image II numbered $k = 0, 1, 2$, and 3. Faces 2 and 3 in Image I, for example, may not be the same as faces 1 and 0 in Image II because of camera orientations. The Hopfield neural network is to find the corresponding faces between the images. Such correspondences are represented in the state array of $\mathbf{V} = (v_{00}, v_{01}, \ldots, v_{33})$ where a high correspondence between face i in image I and face k in Image II is expressed by a high state value ($0 \le v_{ik} \le 1$) of neuron (i, k). The final state $\mathbf{V}_s$ is obtained by a minimization of the following energy function:

$$E = -\sum_i\sum_k\sum_j\sum_l C_{ikjl}V_{ik}V_{jl} + \sum_i (1 - \sum_k v_{ik})^2 + \sum_k (1 - \sum_i v_{ik})^2 \quad (1)$$

where the first term is a compatibility constraint; the second and third terms are uniqueness constraints. v_{ik} converges to 1.0 if face i in Image I matches face k in Image II perfectly. Otherwise, it is greater than or equal to 0. According to Lin *et al.* (1991), C_{ikjl} is the interconnection strength between neuron (i, k) (row i and column k) and neuron (j, l) (row j and column l); i.e.,

$$C_{ikjl} = \sum_n W_n * F(x_n, y_n) \quad (2)$$

where x_n is the n-th measuring feature of neuron (i, k) and y_n is the n-th measuring feature of neuron (j, l); w_n is a weight with $\sum W_n = 1$; and the transfer function F is

$$F(x, y) = \begin{cases} 1 & \text{if } |x - y| < \theta \\ -1 & \text{otherwise} \end{cases}, \quad (3)$$

where θ is a threshold. The measuring features x_n and y_n evaluate the differences between characteristics of objects to be recognized in the two images. Examples of the measuring features may be shape similarity, orientation consistency, and conformance to constraints. If the measuring features from the two images do not match through the thresholds, the transfer function, and the weights, the energy function in Equation 1 will be penalized by a large value. Otherwise, the measuring features will have a smaller contribution to the energy function. In applications, it is very important to design the detailed measuring features, weights, and thresholds. We separate the energy terms into two groups and give them different weights. Group 1 is the first term and Group 2 contains the second and third terms. Two weight coefficients p_1 and p_2 ($p_1 \ge 0$, $p_2 \ge 0$, and $p_1 + p_2 = 1$) are assigned to the groups, respectively: i.e.,

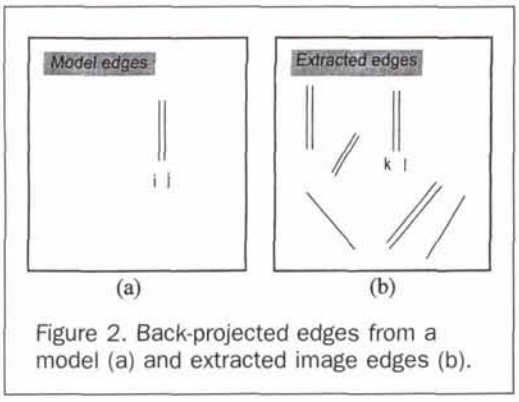


Figure 2. Back-projected edges from a model (a) and extracted image edges (b).

$$E = p_1 \left[- \sum_i \sum_k \sum_j \sum_l C_{ikjl} v_{ik} v_{jl} \right] + p_2 \left[\sum_i (1 - \sum_k v_{ik})^2 + \sum_k (1 - \sum_i v_{ik})^2 \right]. \quad (4)$$

If we increase p_2, unique matches between features in rows and columns will be favored. A high value of p_1 will allow a feature in a row/column to match multiple features in columns/rows. The minimization of Equation 4 is performed by updating the state array **V** iteratively. Note that the image features in Figure 1 may also be edge or point features.

Automatic Object Detection and Location

In the context of this paper, object detection involves finding all image features of a certain object to be detected in the image domain. Subsequently, object location requires the determination of corresponding stereo features among the detected image features and the photogrammetric triangulation of object positions in the scene domain using the corresponding stereo features. We use street light poles as an example of objects to explain the principles. To detect and locate other objects, only those parts that are object specific, for example, measuring features in Equation 2, need to be modified. In the scene domain, a light pole can be modeled as a cylinder with a certain diameter and a certain length. It should stand vertically not far from the mobile mapping van. Because the images are georeferenced, given its approximate location in the scene domain, the 3D light pole model can be back-projected onto images using the camera orientation parameters. The light pole model is then represented as two parallel edges in the images (Figure 2a). The measuring features in Equation 2 evaluate differences between the back-projected pole edges and the extracted image edge features (Figure 2b). Two groups of measuring features are defined in the image domain, namely, local measuring features and relational measuring features. The local measuring features include (Figure 3)

- Length (λ): length of an edge,
- Azimuth (α): azimuth of an edge (measured clockwise from the x-axis),
- Distance (δ): distance between two edges, and
- Local gradient $(+1 \text{ or } -1)_{ik}$: east-west gradient (+1 for the case from background to the interior of the pair, −1 otherwise).

The relational measuring features are

- Ratio (δ/λ): width-height ratio, and
- Relative gradient $(+1 \text{ or } -1)_{ikjl}$: relative gradient defined on edges *i-j* and *k-l* similar to the local gradient.

The Hopfield neural network uses various combinations of the above measuring features to recognize poles in different situations.

The following steps depict the algorithm developed for recognizing light poles from the mobile mapping data:

Step 1. Create a matrix with a dimension of M by I (M is the number of back-projected model edges and I is the number of edge features in the image) to store neuron states **V**. Figure 2 shows an example of a pair of back-projected edges from a light pole model and the extracted image edges. $M = 2$ and $I = 10$.

Step 2. Set the initial states of **V** as (Lin *et al.*, 1991)

$$V_{ik}^0 = g(u_{ik}^0) = [1 + \exp(-2u_{ik}^0/u_0)]^{-1} \text{ and}$$
$$u_{ik}^0 = u_{\text{init}} + d = u_0 + d$$

where u_{ik} is called input voltage. $u_0 = 0.0002$ and d is a random number uniformly distributed between $-0.1u_0$ and $0.1.u_0$

Step 3. Build the C coefficients in Equation 2: i.e.,

$$\begin{aligned} G_{ikjl} = {} & W_1 * F(\delta_{I_iI_j}, \delta_{M_kM_l}) + W_2 * F(\alpha_{I_i}, \alpha_{M_k}) \\ & + W_3 * F(\alpha_{I_j}, \alpha_{M_l}) + W_4 * F(\lambda_{I_i}, \lambda_{M_k}) \\ & + W_5 * F(\lambda_{I_j}, \lambda_{M_l}) + W_6 * F(G_{I_i}, G_{M_k}) \\ & + W_7 * F(G_{I_j}, G_{M_l}) + W_8 * F(G_{I_iM_k}, G_{I_jM_l}) \end{aligned} \quad (5)$$

The measuring features of δ, α, λ, and G are defined. Their subscripts of M and I are those of the model and image, respectively. W_i ($i = 1, 2, \ldots, 8$) are weight factors of the measuring features ($\sum W_i = 1$). They are set as $W_2, = W_3$, $W_4 = W_5$, $W_6 = W_7 = W_8/2$. The weights are adjusted according to the object to be recognized and the images. Note that $F(x,y) = 0$ if $i = j$ or $k = l$.

Step 4. Set the current number of iteration as $t = 1$ and the limit of iterations as n.

Step 5. Update the values of u_{ik} and V_{ik} (Lin *et al.*, 1991) for ($i = 0$; $i < M$; i++) and for ($k = 0$; $k < I$;k++): i.e.,

$$u_{ik}^{t+1} = u_{ik}^t + \frac{1}{6}(K_1 + 2 * K_2 + 2 * K_3 + K_4) \quad (6)$$

where

$$K_1 = h \times f(u_{ik}^t = h \times (\sum_j \sum_l C_{ikjl} V_{jl}^t - \sum_j V_{il}^t - \sum_j V_{jk}^t - u_{ik}^t + 2),$$
$$K_2 = h \times f(u_{ik}^t + \frac{1}{2} K_1),$$
$$K_3 = h \times f(u_{ik}^t + \frac{1}{2} K_2),$$
$$K_4 = h \times f(u_{ik}^t + K_3), \text{ and}$$

h is a constant (0.0001).

Step 6. $V_{ik}^{t+1} = g(u_{ik}^{t+1})$ and repeat Steps 5 and 6 until $t \geq n$.

Step 7. Calculate the final states

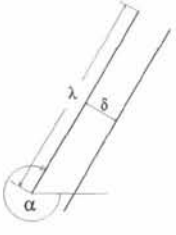
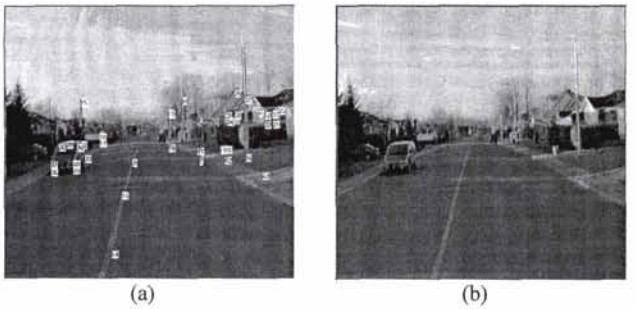
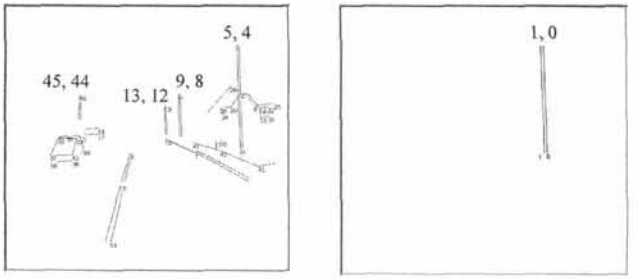


Figure 3. Definition of measuring features of a light pole.



(a) (b)

Figure 4. Extracted and labeled edges (a) and the original image (b).



(a) (b)

Figure 5. Extracted edges (a) and the back projection of the model (b).

$$v_{ik} = \begin{cases} 1 & \text{if } v_{ik} > \theta_1 = 0.8 \\ 0 & \text{otherwise} \end{cases}.$$

Detection and Location of a Specific Object

In Figure 4a, the extracted edges are labeled and overlaid on an original image. The same extracted edges and the back projection of the 3D pole model are shown in Figures 5a and 5b, respectively. Data in the scene domain include a light pole model defined as a cylinder with a given radius and length of 0.212 m and 6.795 m, respectively. The values were obtained by manual photogrammetric measurements of light poles from the images in the project area. Data in the image domain are the edge features in images that are supposed to be extracted by digital image processing. At this time, they are manually extracted by an operator. Each edge has a label ID, a starting node and end node with coordinates, and a gradient. The image format is 512 pixels by 480 pixels.

The initial position of the pole in the scene can be given by an approximation from the vehicle navigation data and common sense. It can also be from a spatial utility database. The neural network is supposed to answer questions such as "is there a light pole in the immediate neighborhood of the approximate position in the scene domain?" or "does the light pole described by the utility database remain in the same place?" Among the edges in Figure 5a, 4 and 5, 8 and 9, and 12 and 13 are pairs of edges of light poles. Edges 44 and 45 represent a partial light pole. The parallel edges 10 and 11 represent a trashcan. To detect the specific light pole represented by edges 4 and 5 with a known approximate position in the scene domain, we use the measuring features of edge length (scale variant), edge azimuth (close to vertical), and local gradient (compatibility of an edge pair to form a pole). Edges 0 and 1 in Figure 5b are artificial lines back projected from the 3D light pole model. The weights p_1 and p_2 of the two energy groups are set equal in the energy function. Numerical values of the measuring features of some typical pole(s) in the images can be calculated and analyzed to determine thresholds (θ) and weights (W_n) in Equations 2 and 3. The maximum number of iterations is set at 1,000. The unit of the thresholds of λ and δ are in pixels. That of the threshold of α is degrees. A partial differential equation is used to link the updated neuron states and the energy decrease/increase in Equation 4. This differential equation is solved iteratively by the aforementioned algorithm. Table 1 lists the final states of the neurons, indicating the recognition result with the following parameters:

Threshold of λ = 4.0	Weight of λ = 0.25
Threshold of α = 3.0	Weight of α = 0.25
Threshold of δ = 4.0	Weight of δ = 0.25
Threshold of δ/λ = 0.05	Weight of δ/λ = 0.00
Threshold of gradient = 0	Weight of gradient = 0.25
p_1 = 0.5	p_2 = 0.5.

According to Table 1, the system found the specific pole with edges 4 and 5 successfully by giving high state values of 1 for neurons (model line 0, extracted line 4) and (model line 1, extracted line 5). Other light pole edges gained relatively high state values, while nonpole edges have low values. In the same way, the 3D pole model is projected to the right image and compared with the extracted edges in the right image by the neural network. The final coordinates of the pole in the scene domain are determined by a photogrammetric intersection of the corresponding image poles in the left and right images detected by the neural network.

Detection of Nonspecific Objects in the Image Sequence

To detect all light poles in the image (Figure 4a), regardless of their positions, measuring features that are not significantly affected by object positions in the scene domain should be applied. It is clear that lengths of pole edges in the image domain vary because of pole positions and the perspective projection. The selected measuring features include azimuth of edges (close to vertical), width-length ratio (photo-scale invariant), and relative gradient (compatibility of a line pair to form a pole, photo-scale invariant). Other measuring features are excluded by setting the corresponding weights to 0. For the image data, the applied parameters are

Threshold of λ = 0.00	Weight of λ = 0.00
Threshold of α = 3.00	Weight of α = 0.25
Threshold of δ = 0.00	Weight of δ = 0.00
Threshold of δ/λ = 0.05	Weight of δ/λ = 0.50
Threshold of gradient = 0	Weight of gradient = 0.25
p_1 = 0.5	P_2 = 0.5

TABLE 1. FINAL STATES (V_{ik}) OF THE NEURONS FOR DETECTING THE LIGHT POLE REPRESENTED BY EDGES 4 AND 5 IN FIGURE 4A.

ID of extracted edge	0	1	2	3	4	5	6	7	8	9	10	11
V_{0*} (model edge 0)	0	0	0	0	1	0	0	0	0.46	0	0.22	0
V_{1*} (model edge 1)	0	0	0	0	0	1	0	0	0	0.49	0	0.26
	12	13	14	15	16	17	18	19	20	21	22	23
	0.46	0	0	0	0	0	0	0.22	0	0	0	0
	0	0.49	0	0	0	0	0	0	0	0.26	0	0
	24	25	26	27	28	29	30	31	32	33	34	35
	0	0	0	0	0	0	0	0	0	0	0.22	0
	0	0	0	0	0	0	0	0	0	0	0	0
	36	37	38	39	40	41	42	43	44	45		
	0	0	0	0	0	0	0	0	0.46	0		
	0	0	0	0	0	0	0	0	0	0.49		

The pole model can be placed in an arbitrary position in the scene domain covered by the image. It is then back projected onto the image. It should be expected that the back projected model edges would match all image pole edges. Table 2 gives the final states (v_{ik}) of the neurons for detecting all light poles in the image. The neural network recognized all light poles correctly by indicating matches of model edges 0 and 1 with image pole edges of 4 and 5, 8 and 9, 12 and13, and 44 and 45 with large state values of 0.98, respectively.

This detection process can proceed to detect all light poles in the image sequence.

Location of Detected Objects in the Scene Domain

Figure 6 depicts a sequence of 25 stereo image pairs from a mobile mapping survey line. Exposure stations of stereo images taken simultaneously by the mobile mapping system are indicated by + + symbols. The exposure stations are numbered from 102 to 127. The two stereo images at each station are distinguished by their station numbers with an L (left) or R (right) extension, for example, 112L and 112R. Six light poles, from LP1 to LP6, are marked by symbols. The exposure stations and/or images are listed for each light pole that appears in the images. The algorithms discussed above are able to detect and locate a position specific light pole or to detect all light poles in the image sequence in the image domain.

In order to locate the detected poles in the scene domain, corresponding image pole features in stereo images have to be identified. Subsequently, their locations in the scene domain can be triangulated from the detected corresponding image features. Thus, the major task at this stage is to find the corresponding image pole features in the image domain. Such corresponding features are mostly found in stereo image pairs with "hard" baselines at the same exposure stations, but they may also be found in stereo image pairs with "soft" baselines formed by images of preceding and/or following exposure stations. Considering the effective baseline (baseline component vertical to the line that links the pole and the middle point of the baseline) of possible combinations of stereo pairs with hard/soft baselines, for each pole, only three stations that are close to the pole are used for location of the pole. For example, light pole LP2 in Figure 6 is covered by images of exposure stations 102, 103, 104, 105, 106, 107, and 108. However, only three stations (106, 107, and 108) are used.

To start the location process of the sequence, a detected image pole feature (a pair of edges) with the greatest length (closest to the cameras) in either image at the first station (102), for instance 102L, is automatically selected. The image coordinates of the pole bottom and pole top are known as (x_b, y_b) and (x_t, y_t). Their corresponding coordinates in the scene domain are (X_B, Y_B, Z_B) and (X_T, Y_T, Z_T). Assume that the pole is vertical and the pole length is l; we then have $X_T = X_B$, $Y_T = Y_B$, and $Z_T = Z_B + l$. Because the camera orientation parameters are known, there are four collinearity equations from two spatial lines in the scene domain, namely, the line from the pole bottom in the image $(x_b, y_b, -f)$ through the exposure center (X_o, Y_o, Z_o) to the pole bottom in the scene domain (X_B, Y_B, Z_B), and another line from the pole top in the image $(x_t, y_t, -f)$ through the exposure center (X_o, Y_o, Z_o) to the pole top in the scene domain $(X_B, Y_B, Z_B + l)$: i.e.,

$$x_b = -f\frac{a_{11}(X_B - X_O) + a_{12}(Y_B - Y_O) + a_{13}(Z_B - Z_O)}{a_{31}(X_B - X_O) + a_{32}(Y_B - Y_O) + a_{33}(Z_B - Z_O)},$$

$$y_b = -f\frac{a_{21}(X_B - X_O) + a_{22}(Y_B - Y_O) + a_{23}(Z_B - Z_O)}{a_{31}(X_B - X_O) + a_{32}(Y_B - Y_O) + a_{33}(Z_B - Z_O)},$$

$$x_t = -f\frac{a_{11}(X_B - X_O) + a_{12}(Y_B - Y_O) + a_{13}(Z_B + l - Z_O)}{a_{31}(X_B - X_O) + a_{32}(Y_B - Y_O) + a_{33}(Z_B + l - Z_O)}, \text{ and}$$

$$y_t = -f\frac{a_{21}(X_B - X_O) + a_{22}(Y_B - Y_O) + a_{23}(Z_B + l - Z_O)}{a_{31}(X_B - X_O) + a_{32}(Y_B - Y_O) + a_{33}(Z_B + l - Z_O)}. \quad (7)$$

In the above four equations, there are three unknowns (X_B, Y_B, Z_B) which can be solved for by a least-squares adjustment. The coordinates are then used to back project a 3D pole model to the right image 102R. The two back-projected model edges are then used to match the image pole edges in the right image by a neural network (position specific). Once the corresponding image pole feature in the right image is found, the location of the pole in the scene domain can be calculated by a photogrammetric triangulation. To obtain a more precise location of a pole, for example LP2, the pole whose location is first calculated from Equation 7 is back

TABLE 2. FINAL STATES (V_{ik}) OF THE NEURONS FOR DETECTING ALL LIGHT POLES IN FIGURE 4A.

ID of extracted line	0	1	2	3	4	5	6	7	8	9	10	11
V_{0*} (model edge 0)	0	0	0	0	0.98	0	0	0	0.98	0	0.02	0
V_{1*} (model edge 1)	0	0	0	0	0	0.98	0	0	0	0.98	0	0.02
12	13	14	15	16	17	18	19	20	21	22	23	
0.98	0	0	0	0	0	0	0.02	0	0	0	0	
0	0.98	0	0	0	0	0	0	0	0.02	0	0	
24	25	26	27	28	29	30	31	32	33	34	35	
0	0	0	0	0	0	0	0	0	0	0.02	0	
0	0	0	0	0	0	0	0	0	0	0	0	
36	37	38	39	40	41	42	43	44	45			
0	0	0	0	0	0	0	0	0.98	0.98			

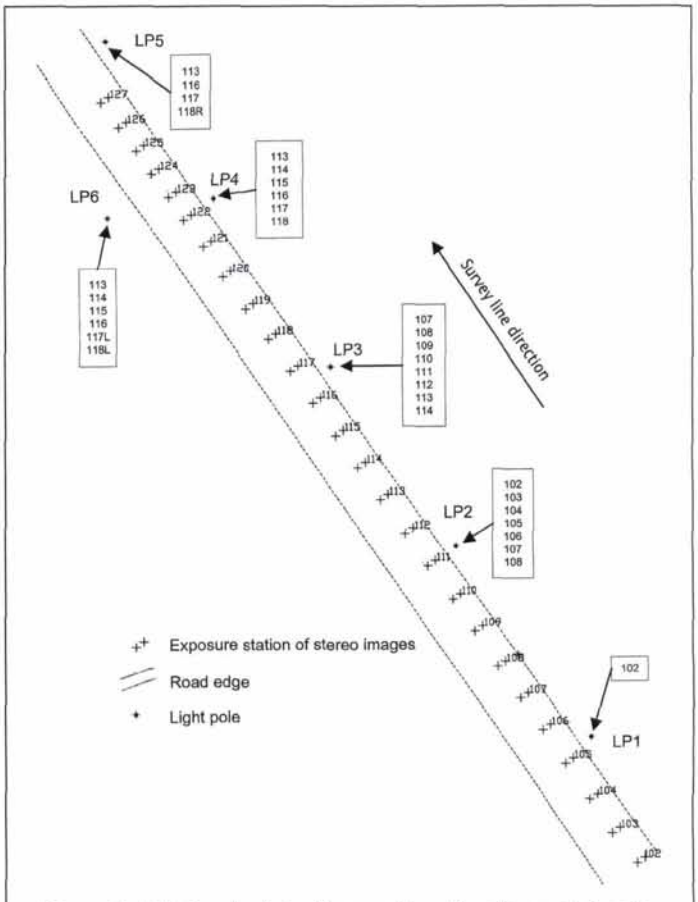


Figure 6. Light pole detection and location from an image sequence.

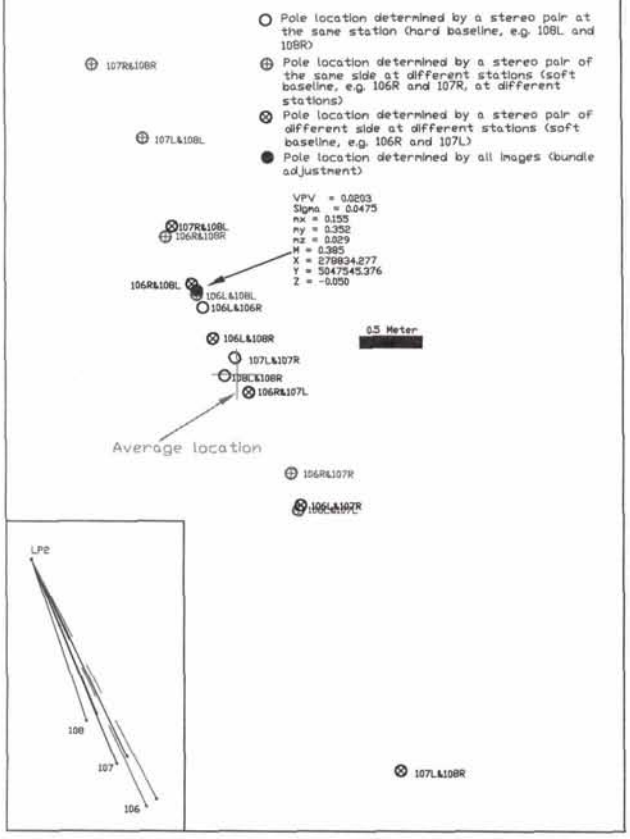


Figure 7. Distribution of locations of light pole LP2 estimated by various combinations of stereo pairs and a bundle adjustment.

projected onto multiple images (106L, 106R, 107L, 107R, 108L, and 108R). The neural network finds all corresponding image pole features in the selected images. A bundle adjustment is applied to estimate the optimal location of the pole in the scene domain using all corresponding image pole features. This process is performed on the entire image sequence, so that all light poles in the sequence are located.

Figure 7 presents locations of light pole LP2 estimated by 11 combinations of stereo image pairs and a bundle adjustment. Ideally, the locations should be at the same point or within a very small area. However, because of the relatively short effective baselines (small intersection angles) the locations are spread along the track direction. In the middle of the distribution are points calculated from the stereo pairs with large effective baselines (e.g., 106R & 108L and 106L & 108L) and those close to the pole. Two image pairs, namely 107L & 108R and 107R & 108R, have very small effective baselines and the locations calculated thereby are far away from the average location. A bundle adjustment using all detected corresponding image features of LP2 from three close exposure stations is performed and leads to the location that matches the results of triangulations with large effective baselines, instead of the average location from all individual locations. This confirms the conclusion from a previous study on optimization of photogrammetric triangulation using mobile mapping data (Li *et al.*, 1996).

Conclusions

Algorithms based on Hopfield neural networks for object detection and location, especially for street light poles, from mobile mapping image sequences have been researched and developed. A software system N²M² (Neural Networks for Mobile Mapping) is developed based on the C++ programming language in the Microsoft Windows 32-bit environment. Major contributions of this research are

- Establishment of a Hopfield neural network for object recognition from mobile mapping image sequences using 3D object models,
- Application of the developed algorithms for detection and location of light poles from a single image pair and/or from an image sequence,
- Understanding of the behavior of the neural network when applied in various mobile mapping situations, and
- Development of the N²M² system.

The measuring features of the network are crucial for characterizing objects and thus are object dependent. If different objects are to be recognized, new measuring features should be defined and integrated in the network. A further challenge is to develop a systematic learning process of the neural network for handling mobile mapping data. We believe that such research will result in a generic method for the optimal determination of thresholds and weight values in the network for different objects.

Acknowledgment

This research was supported by the OSU Center for Mapping (CFM), NASA Center for the Commercial Development of Space, and NSF (CMS-9812783). Mobile mapping data are provided by TRANSMAP Corp. and VISAT Technologies Inc. Discussions and collaboration with Dr. J. Bossler and colleagues at CFM are appreciated.

References

Bishop, C.M., 1995. *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford.

Bossler, J.D., C.C. Goad, P.C. Johnson, and K. Novak, 1991. GPS and GIS Map the Nation's Highways, *GeoInfo Systems*, March, pp. 27–37.

Braess, M., 1997. *Strukturbasierte Merkmalszuordnung in kurzen stereoskopischen Videosequenzen*, Fakultät für Bauingenieur- und Vermessungswesen der Rheinisch-Westfälischen Technischen Hochschule Aachen, 117 p.

He, G., 1996. Design of a Mobile Mapping System for GIS Data Collection, *International Archives of Photogrammetry and Remote Sensing*, 31(Part B2):154–159.

He, G., and K. Novak, 1992. Automatic Analysis of Highway Features from Digital Stereo Images, *International Archives of Photogrammetery and Remote Sensing*, 29(Part B3):119–124.

Hopfield, J.J., and D.W. Tank, 1985. "Neural" Computation of Decisions in Optimization Problems, *Biol. Cybernet.*, 52:141–152.

Li, R., 1997. Mobile Mapping — An Emerging Technology for Spatial Data Acquisition, *Photogrammetric Engineering & Remote Sensing*, 63(9):1085–1092.

Li, R., K.P. Schwarz, M.A. Chapman, and M. Gravel, 1994. Integrate GPS, INS, and CCD Cameras for Rapid GIS Data Acquisition, *GIS World*, 7(4):41–43.

Li, R., M.A. Chapman, and W. Zou, 1996. Optimal Acquisition of 3D Object Coordinates from Stereoscopic Image Sequences, *International Archives of Photogrammetry and Remote Sensing*, 31(Part B3):449–452.

Li, W., and N.M. Nasrabadi, 1989. Object Recognition Based on Graph Matching Implemented by a Hopfield-Style Neural Network, *Int. J. Conf. Neural Networks II*, Washington, D.C., 18–22 June, pp. 287–290.

Lin, W.-C., F.-Y. Liao, C.-K. Tsao, and T. Lingutla, 1991. A Hierarchical Multiple-View Approach to Three-Dimensional Object Recognition, *IEEE Transactions on Neural Networks*, 2(1):84–92.

Novak, K., 1995. Mobile Mapping Technology for GIS Data Collec-

tion, *Photogrammetric Engineering & Remote Sensing*, 61(5): 493–501.

Schwarz, K.-P., and N. El-Sheimy, 1996. Kinematic Multi-Sensor Systems for Close Range Digital Imaging, *International Archives of Photogrammetry and Remote Sensing*, 31(Part B3):774–785.

Tao, C., R. Li, and M.A. Chapman, 1998. Automatic Reconstruction of Road Centerlines from Mobile Mapping Image Sequences, *Photogrammetric Engineering & Remote Sensing*, 64(7):709–716.

Tseng, Y.-H., J.-J. Tzeng, K.-P. Tang, and S.-H. Lin, 1997. Image-to-image Registration by Matching Area Features Using Fourier Descriptors and Neural Networks, *Photogrammetric Engineering & Remote Sensing*, 63(8):975–983.

Xin, Y., 1995. *Automating Geostation: A Softcopy System*, Master's Thesis, Department of Geomatics Engineering, The University of Calgary, Canada.

Zhang, Y.-S., 1996. A Hierarchical Neural Network Approach to Three-Dimensional Object Recognition, *International Archives of Photogrammetry and Remote Sensing*, 31(Part B3):1010–1017.

(Received 20 February 1998; accepted 29 October 1998; revised 24 November 1998)