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 Tunk, 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.
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 ranger, 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 \( \mathbf{V} = (v_1, v_2, ..., v_n) \) of all neurons, (2) there are a specific set of stable states \( \mathbf{V}_s = (v_1, v_2, ..., v_n) \) 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 orientation. The Hopfield neural network is to find the corresponding faces between the images. Such correspondences are represented in the state array of \( \mathbf{V} = (v_{00}, v_{01}, ..., v_{33}) \) where a high correspondence between face \( i \) in image I and face \( k \) in Image II is expressed by a high state value \( 0 \leq v_{i,k} \leq 1 \) of neuron \( (i, k) \). The final state \( \mathbf{V} \) is obtained by a minimization of the following energy function:

\[
E = -\sum_{i} \sum_{j} \sum_{k} C_{ijk} v_{i,k} v_{j,k} + \sum_{i} (1 - \sum_{k} v_{i,k})^2 + \sum_{k} (1 - \sum_{i} v_{i,k})^2 \tag{1}
\]

where the first term is a compatibility constraint; the second and third terms are uniqueness constraints. \( v_{i,k} \) 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), \( C_{ijk} \) is the interconnection strength between neuron \( (i, k) \) (row \( i \) and column \( k \) ) and neuron \( (j, l) \) (row \( j \) and column \( l \) ); i.e.,

\[
C_{ijk} = \sum_{n} W_n \ast F(x_n, y_n) \tag{2}
\]

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 \( \sum_{n} W_n = 1 \); and the transfer function \( F \) is

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

where \( \theta \) 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 \( p_2 \) (\( p_1 \geq 0, p_2 \geq 0 \), and \( p_1 + p_2 = 1 \)) are assigned to the groups, respectively: i.e.,
The relational projective pole edges and the extracted image edge features parameters. The light pole model is then represented as two 2 back-projected onto images using the camera orientation because the images are georeferenced, given its approximate location in the scene domain, the 3D light pole model can be recognized and the images. Note that object location requires the determination of corresponding stereo features among 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 (l): length of an edge, azimuth (a): azimuth of an edge (measured clockwise from the x-axis), distance (d): distance between two edges, and local gradient (+1 or -1), east-west gradient (+1 for the case from background to the interior of the pair, -1 otherwise).

The relational measuring features are ratio (d/a): width-height ratio, and relative gradient (+1 or -1), relative gradient defined on edges i-j and k-l similar to the local gradient.

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)

\[ V_k^0 = \mathbb{g}(u_k^0) = \left(1 + \exp \left(-2u_k^0 \right) \right)^{-1} \text{ and } u_k^0 = u^0_a + d \]

where \( u^0_a \) is called input voltage, \( u^0_a = 0.0002 \) and \( d \) is a random number uniformly distributed between 0 and 1.

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

\[
G_{a0} = W_1 * F(\alpha_{a0}, \delta_{a0}, \alpha_{a0}) + W_2 * F(\alpha_{a0}, \alpha_{a0}) + W_3 * F(\lambda_{a0}, \alpha_{a0}) + W_4 * F(\gamma_{a0}, \alpha_{a0}) + W_5 * F(\alpha_{a0}, \gamma_{a0}) + W_6 * F(\lambda_{a0}, \gamma_{a0}) + W_7 * F(\gamma_{a0}, \gamma_{a0}) + W_8 * F(\gamma_{a0}, \gamma_{a0})
\]

The measuring features of \( \delta, \alpha, \lambda \), 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 \( W_{10} = W_{11} = W_{12} = W_{13} = W_{14} = W_{15} / 2 \). 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 \( u^i_k \) and \( V_k^i \) (Lin et al., 1991) for \( (i = 0; i < M; i++) \) and for \( (k = 0; k < I, k++) \): i.e.,

\[
\hat{u}_{ik} = u_{ik} + \frac{1}{6}(K_1 + 2 \cdot K_2 + 2 \cdot K_3 + K_4) \tag{6}
\]

where

\[
K_1 = h \times f(u_{ik}^i) = h \times \left( \sum C_{a0} V_j - \sum V_j - \sum V_j^k - u_{ik}^k + 2 \right),
K_2 = h \times f(u_{ik}^i + \frac{1}{2} K_1),
K_3 = h \times f(u_{ik}^i + \frac{1}{2} K_2),
K_4 = h \times f(u_{ik}^i + \frac{1}{2} K_3),
\]

\( h \) is a constant (0.0001).

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

Step 7. Calculate the final states.
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 projection of the light pole described by the utility database remain in the same place. Among the edges in Figure 5a, 4 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), width-length ratio (photo-scale invariant), and relative gradient (compatibility of a line 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 ($\theta$) and weights ($W_v$) in Equations 2 and 3. The maximum number of iterations is set at 1,000. The unit of the thresholds of $\alpha$ and $\delta$ are in pixels. That of the threshold of $\alpha$ 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 $\delta = 4.0$ Weight of $\delta = 0.25$
Threshold of $\delta/\lambda = 0.05$ Weight of $\delta/\lambda = 0.00$
Threshold of $\alpha = 3.0$ Weight of $\alpha = 0.25$
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 Non-specific 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 $\lambda = 0.00$ Weight of $\lambda = 0.00$
Threshold of $\alpha = 3.00$ Weight of $\alpha = 0.25$
Threshold of $\delta = 0.00$ Weight of $\delta = 0.00$
Threshold of $\delta/\lambda = 0.05$ Weight of $\delta/\lambda = 0.50$
Threshold of gradient $= 0$ Weight of gradient $= 0.25$
$P_1 = 0.5$ $P_2 = 0.5$

Table 1. Final States ($V_v$) of the Neurons for Detecting the Light Pole Represented by Edges 4 and 5 in Figure 4a.

<table>
<thead>
<tr>
<th>Threshold of $\lambda$</th>
<th>Weight of $\lambda$</th>
<th>Threshold of $\alpha$</th>
<th>Weight of $\alpha$</th>
<th>Threshold of $\delta$</th>
<th>Weight of $\delta$</th>
<th>Threshold of $\delta/\lambda$</th>
<th>Weight of $\delta/\lambda$</th>
<th>Threshold of gradient</th>
<th>Weight of gradient</th>
<th>$P_1$</th>
<th>$P_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.46</td>
<td>0.46</td>
<td>0.22</td>
<td>0.22</td>
<td>0.49</td>
<td>0.49</td>
<td>0.25</td>
<td>0.25</td>
<td>0.46</td>
<td>0.46</td>
<td>0.50</td>
<td>0.50</td>
</tr>
</tbody>
</table>

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

Figure 5. Extracted edges (a) and the back projection of the model (b).
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_n) 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 and 13, 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_b, Y_b, Z_b) 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_t, Y_t, Z_t) to the pole top in the scene domain (X_t, Y_t, Z_t + l): i.e.,

\[
\begin{align*}
X_b &= a_{11} (X_b - X_d) + a_{12} (Y_b - Y_d) + a_{13} (Z_b - Z_d) \\
Y_b &= a_{21} (X_b - X_d) + a_{22} (Y_b - Y_d) + a_{23} (Z_b - Z_d) \\
X_t &= a_{31} (X_b - X_d) + a_{32} (Y_b - Y_d) + a_{33} (Z_b + l - Z_d) \\
Y_t &= a_{41} (X_b - X_d) + a_{42} (Y_b - Y_d) + a_{43} (Z_b + l - Z_d)
\end{align*}
\]

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
mobile mapping image sequences have been researched and developed. A software system N^M^2 (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 N^M^2 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


Novak, K., 1995. Mobile Mapping Technology for GIS Data Collect-
In October 2000, the American Society for Photogrammetry and Remote Sensing will devote its issue of *Photogrammetric Engineering and Remote Sensing* (PE&RS) to Remote Sensing and Decision Support Systems (DSS). DSS would include the science-based predictive models, remote sensing information, verification and validation, and the communities that conduct the decision support. Authors are encouraged to submit manuscripts addressing remote sensing and GIS contributions to operational Decision Support Systems.

Possible categories of manuscripts include Decision Support Systems used for:

- Precision Agriculture (Food and Fiber)
- Coastal Ecosystem Management (Natural Resources)
- Flood Plain Risk Assessment (Disaster Management)
- Water Quality (Environmental Quality)
- Urban Planning (Urban and Infrastructure)
- Public Health (Human Health and Safety)

We also encourage the submission of short manuscripts that present the experience of remote sensing/GIS by Federal, State, or Local agencies using decision support systems operationally. Private sector companies under contract to these agencies or otherwise involved in some aspect of developing or operating decision support systems using remote sensing/GIS technology are also invited to submit a short manuscript.

**Guest Editors**

Ronald J. Birk, Intermap Technologies Inc.
Dr. Timothy W. Foresman, University of Maryland-Baltimore

All manuscripts must be prepared according to the “Instructions to Authors” published in each issue of *PE&RS*. Papers will be peer-reviewed in accordance with established ASPRS policy. Please send manuscripts to:

Ronald J. Birk
Intermap Technologies Inc.
Bldg. 1110
Stennis Space Center, MS 36529
228.688.1673 (fax)
rbirk@intermaptechnologies.com

**DEADLINE**

**MAY 15, 2000**