A System for Automatic Pose-Estimation from a Single Image in a City Scene

Björn Johansson
Centre for Mathematical Sciences
Lund University
Box 118, S-221 00 Lund, Sweden

Roberto Cipolla
Department of Engineering
Cambridge University
Cambridge CB2 1PZ, England

Abstract

We describe an automatic system for pose-estimation from a single image in a city scene. Each building has a model consisting of a number of parallel planes associated with it. The homographies for the best match of the planes to the image is estimated automatically for each of the possible buildings. We show how the estimation of homographies can be done effectively by reducing the search space and using fast convolution. The model having the best match is then used to determine the position and orientation of the camera.

The results of a number of experiments of the system in realistic circumstances is also presented.

1 Introduction

In this paper the problem of automatically determining the position and orientation of a camera using a priori information about the surroundings is considered. In particular we study the application of estimating pose in a city scene when models of the surrounding buildings are available.

The objective was to develop a system that could handle different lighting as well as shadows, specularities and occlusions. To be able to handle different lighting conditions either you have to compensate for the lighting or you have to use features which are indifferent of the lighting (or possibly do both). The work so-far has been focused on features which are not so sensitive of the lighting (edges).

The first part of the problem is to recognize the building from a set of models. Object recognition is a hard problem and has been the subject of intense research for many years. However, a few assumptions about buildings will simplify the task for this application. We assume that the buildings are mainly planar and that these planes have both horizontal and vertical edges. This is often the case for buildings as they have windows and doors. To each building that we want to recognize in the images we assume to have been given a model. The model consists of a number of planes in 3D, indicating where are dominant edges. For each of the building candidates, the corresponding model is matched to the image and a measure of fit is estimated. The building having the best fit is chosen, providing it exceeds a threshold.

The second part of the problem consists of estimating the position and orientation of the camera. This is possible having metric information about the model and the match in the image.

Recognizing planar objects has been studied e.g. in [6, 7]. The approach taken in this paper has similarities with the approach in [2]. However, searching for the best match in [2], a local search in six parameters is performed whereas we reduce the search to two searches in two parameters. These are effectively implemented by fast convolution. In addition to this is matching with 3D models examined and it it shown that also in this case the search space is reduced to two dimensions. We further deal with uncalibrated as well as calibrated cameras. This automated system approach for pose estimation is validated by experiments on images from real city scenes.

2 Preliminaries

Camera Model. We model the camera as a pinhole camera. A point in 3D space with homogeneous coordinates $\mathbf{X}$, is projected onto an image point with homogeneous coordinates $\mathbf{x}$, according to:

$$\lambda \mathbf{x} = P\mathbf{X}. \tag{1}$$

Here P is a 3×4 matrix and λ is a scale. The camera matrix P can be decomposed as,

$$P = KR(I_3 | -\mathbf{t}), \tag{2}$$

where I_3 is the identity matrix, R is a 3×3 orthogonal matrix representing the orientation of the camera, $\mathbf{t}$ is a 3 vector representing the position of the camera, and K is the calibration matrix:

$$K = \begin{bmatrix} \tau f & s & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}.$$

K contains the intrinsic camera parameters: the focal length f, the aspect ratio τ the principal point (x_0, y_0) and the skew s. If these parameters are known the camera is said to be calibrated.

Homographies A plane π in the scene is mapped to an image by a homography. The homography can be represented by a 3×3 matrix H_π determined up to scale, and for corresponding points, in the plane $\mathbf{x}$ and in the image $\mathbf{x}'$, the mapping can be written $\lambda\mathbf{x}' = H_\pi\mathbf{x}$, for some scale λ. Without loss of generality a set of parallel planes may be assumed to have equations, $z = k_i$. Then the homography from this plane to an image of the plane may by letting in $X = (x, y, k_i, 1)^T$ in (1) be expressed as,

$$H_i = KR \begin{bmatrix} 1 & 0 & -t_1 \\ 0 & 1 & -t_2 \\ 0 & 0 & k_i - t_3 \end{bmatrix} . \qquad (3)$$

Here the camera used for the projection is $P = KR(I_3 | -\mathbf{t})$ with $\mathbf{t} = (t_1, t_2, t_3)^T$.

Calibration. The purpose of the calibration is to determine the intrinsic parameters. To do this either you have to use metric or affine information about the scene, or you use prior information about the intrinsic parameters in a sequence of images, e.g that some parameters are constant during the sequence. The latter method is called autocalibration. In our application we work with a single plane in a single image. The camera has 11 parameters, 6 external and 5 internal. Since the metric of the plane is known, the homography (eight degrees of freedom) gives two constraints on the calibration cf. [9, 11]. When the equation of the plane has the form above these constraints can be expressed as:

$$\mathbf{h}_1^T \omega \mathbf{h}_1 - \mathbf{h}_2^T \omega \mathbf{h}_2 = 0, \; \mathbf{h}_1^T \omega \mathbf{h}_2 = 0 \qquad (4)$$

Here $\mathbf{h}_1$ is the first column of the homography, $\mathbf{h}_2$ is the second column and $\omega = K^{-T}K^{-1}$ is the image of the absolute conic.

3 System Approach - 2D model

In the first approach, to each building we associate a template corresponding to a dominant plane of the building. The template should indicate where there are dominant edges on this plane and should have known metric. The template should further consist of mainly horizontal and vertical edges. We assume to have been given such templates, e.g. as images, to each of the building we want to recognize in the image. Figure 1(left) shows an image of a building and Figure 1(right) the corresponding template.

The task is now to recognize one of the buildings in the image and then to determine where the camera is located and oriented.

Recognition

For each of the possible templates, the homography that best maps the template to the given image is estimated and a measure of how good the match is is calculated. The template having the best match is chosen, providing the measure exceeds a threshold.

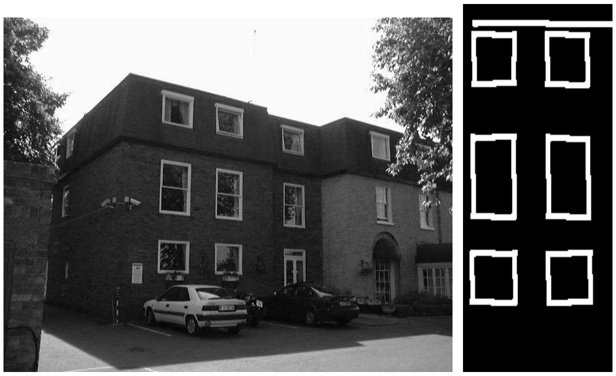

Figure 1. Original image (left) and the corresponding template (right).

It is impractical to search in eight parameters to find the best homography. We will now show that, with certain assumptions, the search can be reduced to two searches in two parameters. These can be performed effectively with fast convolution.

The assumption that there are dominant edges in both the vertical and horizontal directions makes it suitable to decompose the homography into two parts. The first part transforms the image so that lines in the image, corresponding to horizontal/vertical lines in the dominant plane, are horizontal/vertical in the transformed image (rectification). This transformation has four degrees of freedom. The second part determines the scale and translation parameters in the $x-$ and $y-$directions. Also this transformation has four degrees of freedom. But since the template mainly consists of horizontal and vertical edges the x parameters can be separated from the y parameters. Consequently may the homography be decomposed as,

$$H = S_x S_y T,$$

where

$$S_x = \begin{bmatrix} c_x & 0 & d_x \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \; S_y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_y & d_y \\ 0 & 0 & 1 \end{bmatrix}$$

and T is the rectifying homography. We determine H by estimating these components.

Rectification

The rectifying homography have four degrees of freedom, thus it is enough to know how two points should be mapped

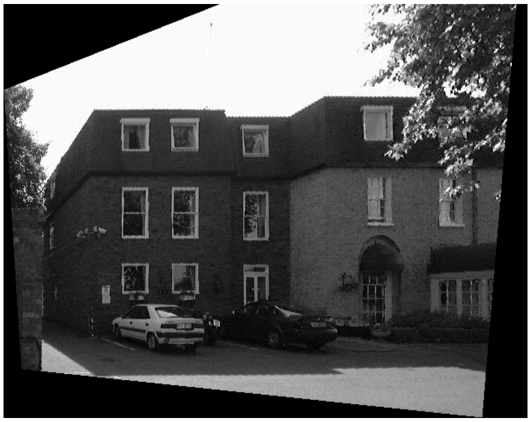

Figure 2. Rectification of the image in Figure 1(left).

in order to determine it. We know that the vanishing points for horizontal and vertical edges should be mapped to $(1,0,0)^T$ and $(0,1,0)^T$ respectively. If these can be determined the rectifying homography may be computed. A number of approaches for determining vanishing points have been proposed, cf. [5, 8, 12] where earlier references also are given. In [12] it is suggested to use a cascaded Hough transformation on the derivative of the image. Each point in the derivative of the image votes on a number of lines. After thresholding the lines, each line votes on a number of points, corresponding to possible intersection points. In [5] the vanishing points are estimated using a Maximum Likelihood estimator.

When the two vanishing points are determined the image can be rectified by applying a homography, T, that maps the vanishing points to infinity in the horizontal and vertical directions. If the camera calibration is known other constraints must also be fulfilled as well. First if v_1 and v_2 are projections in the image of orthogonal directions in the scene, then $v_1^T K^{-T} K^{-1} v_2 = 0$. In this case there are consequently only three degrees of freedom for the two vanishing points. Secondly, since the rectified image has camera matrix $P_r = [I - \mathbf{t}]$, for the homography T it holds that $TR[I - \mathbf{t}] = [I - \mathbf{t}]$. Then T must be a orthogonal matrix. To determine T when the vanishing points are determined the concept of quaternions may be used, cf [3].

Figure 2 shows the rectification of the uncalibrated image in Figure 1.

Translation and scale

There are in the uncalibrated case four parameters left to be determined, scale and translation in x and y. Using the fact that there are mainly vertical and horizontal edges in the template we determine the scale and translation in x independently of the scale and translation in y.

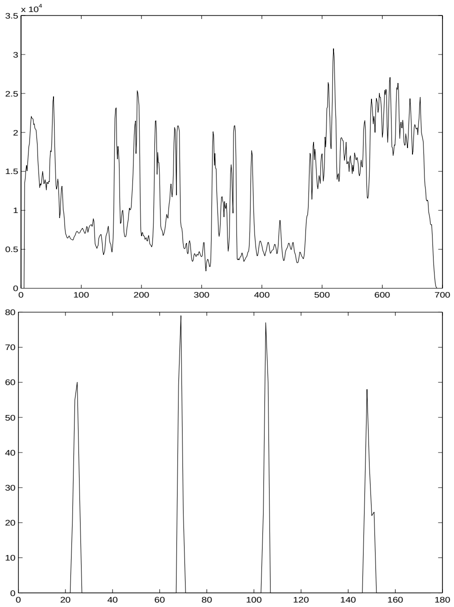

Figure 3. Plot of the column sum of the derivative (left) and the template (right).

By a summation the columns of a thresholded derivative image, we get a plot as shown in Figure 3. If the low frequency component is subtracted, the algorithm becomes more robust to unwanted derivatives, e.g. the trees in Figure 1. We want to match this to a similar plot derived from the template by finding the correct scale and translation. The corresponding template plot is shown in Figure 3.

One way to do this is to create a matrix where each row is a different scaling of the plot in Figure 3. The template plot is then correlated with the matrix to find the scale and translation in x. The correlation can be done effectively using FFT. The same procedure is carried out on the rows.

When a number of local maxima for the x and y parameters has been determined, the next step is to find the best pair of x and y parameters. This is done by, for all possible parameter-combinations, estimating the number of pixels which is decided to be edge-pixels in the template as well as in the derivative image. The parameter-combination that has the greatest numbers of such pixels is chosen. The quotient between these two numbers of edge pixels can serve as a measure of quality for the match and may be used to decide if the current building is present in the image. The template could also have a number of pixels where there should be no edges in the images, corresponding to smooth areas on the building. The number of such pixels mapped to edge pixels in the image may also be used in the search

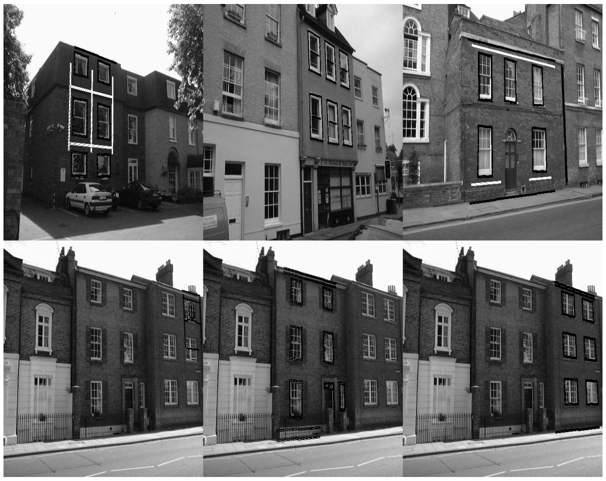

Figure 4. Results of the matching algorithm for a few images. Black pixels correspond to edges and white pixels to smooth areas.

of the best match.

In the calibrated case there are only three degrees of freedom left when the image has been rectified, one scale and translation in $x-$ and $y-$direction. The same procedure as above may be applied, however when choosing parameter combinations the scale in x and y must be equal.

3.1 Experiment 1

The matching algorithm has been tested on a number of different images with pleasing results. Figure 4 shows the results of experiments on uncalibrated images. In some cases templates indicating both edges and smooth areas have been used (white pixles in the template). For the case in the lower left of Figure 4, the algorithm failed to find the left most part of the building. The template used here was probably too simple to give a clear maximum.

3.2 Pose estimation

The estimated homography is now used to determine the position and orientation of the camera.

A homography, H, from a plane with known metric to an image give rise to the two constraints in (4) on the calibration matrix. From a calibrated camera it is well known that pose estimation is possible from a single homography. Hence, having a camera-model with two unkown internal parameters (e.g focal-length and aspect-ratio) we can determine the position of the camera from a single homography. First the two intrinsic camera parameters are estimated using (4). If the plane is assumed to have equation $z = 0$, then the position, $\mathbf{t}$, and orientation, R, of the camera may be recovered by a QR decomposition of

$$K^{-1}H = R\begin{bmatrix}1 & 0 & -t_1\\ 0 & 1 & -t_2\\ 0 & 0 & -t_3\end{bmatrix}.$$

This decomposition is not unique. There are in general two solutions corresponding to optical centers on two sides on the plane. This is not a problem in practice since it is usually known which side of the plane is visible.

In the presence of noise the QR-decomposition will not have the form above. In e.g. [10] techniques to find the best R and $\mathbf{t}$ in some sense are discussed.

3.3 Experiment 2

In a second experiment, 11 images of a scene from different views and at different times of the day were used. A template of a building was automatically matched to the images without any information about the calibration. The position and orientation of the camera was then calculated according to the above, for this part the calibration of the camera was used. The images are shown in Figure 5 with the best template match. The resulting reconstruction is shown from above in Figure 6. The asterisks are the estimated camera positions and the arrow shows the orientation. The line at the bottom is the building. In order to validate the results, a reconstruction based on manually determined point correspondences (not only in one plane) and a standard structure and motion algorithm, is shown as circles. As the two reconstructions are very similar, this indicates that the pose recovery is accurate.

4 System Approach - 3D Model

In this section we present some results concerning the use of a 3D model. When a 3D model is used instead of a single plane the discrimination of buildings will improve and the accuracy in the pose estimation will increase.

We assume that the model is given as a number of templates as described above for a number of parallel planes. The relative translation between the planes are further assumed to be known.

We follow the approach for the case of a single planar template. First the image is rectified. Since vertical/horizontal edges in parallel planes have the same vanishing point the rectification procedure is identical to the single plane case.

The next step is to find the best correlation. However, when the scale and translation in e.g. horizontal direction is chosen for one plane, we must find out how the templates in other planes should be scaled and translated before the correlation is done. It is clear that when the scale and translation in x and y have been chosen, then the pose can be estimated and the homography for every plane can be computed. We will now show that knowing the scale

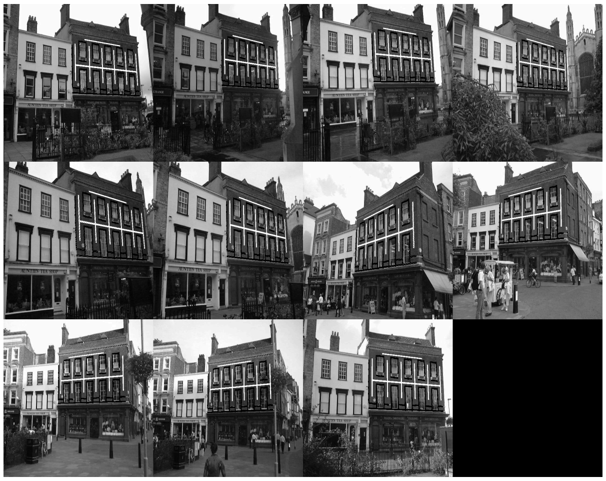

Figure 5. Images from different viewpoints taken at different times of the day with the template matched to a building

and translation in x but not in y makes it possible to compute the scale and translation in x for every other parallel plane. In order for this to work there can only be one degree of freedom in the calibration matrix.

We assume that the image has been rectified and the unknown parameter in the internal calibration has been obtained from $v_1^T K^{-T} K^{-1} v_2 = 0$ using the vanishing points. We further assume that the given image is taken with an perspective camera modeled by $P = KR[I\mathbf{t}]$. Using the rectifying homography, T, and the calibration matrix for the given image, K, the camera matrix for the rectified image P_r may be written as

$$P_r = TKR[I\ \mathbf{t}] = K_r R^T R[I\ \mathbf{t}] = K_r[I\ \mathbf{t}].$$

Knowing K and T the calibration matrix for the rectified image,K_r may be calculated by a RQ decomposition,(modified QR decomposition).

Now suppose a scale and translation in x is computed for one plane of the 3D template. This corresponds to a transformation of the plane by a homography on the form

$$H_1 = \begin{bmatrix} c_x & 0 & d_x \\ 0 & * & * \\ 0 & 0 & 1 \end{bmatrix}$$

where $*$ indicates an unknown y parameter. We want to find the scale and translation for another plane in the model.

$$H_2 = \begin{bmatrix} c_x' & 0 & d_x' \\ 0 & * & * \\ 0 & 0 & 1 \end{bmatrix}$$

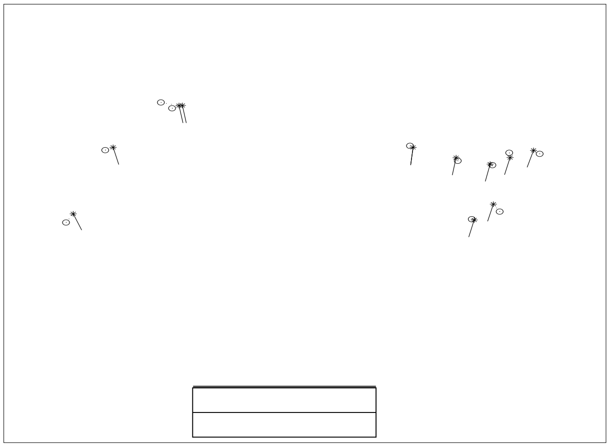

Figure 6. The pose estimation from the matches in Figure 5 (asterisk) compared to a reconstruction based on manually estimated point correspondences (circles)

From (3) we know that the homography for plane $z = k_i$ to the rectified image is given by

$$H_i = K_r \begin{bmatrix} 1 & 0 & -t_1 \\ 0 & 1 & -t_2 \\ 0 & 0 & k_i - t_3 \end{bmatrix}. \qquad (5)$$

where $\mathbf{t} = (t_1, t_2, t_3)^T$ is the focal point for the camera.

Since c_x, d_x, k_i and K_r are known it is possible to calculate t_1 and t_3. Knowing these, the scale and translations in x for a different plane can be calculated from (3).

In this case the search for the best parameter in x can not be done with a single fast convolution as for the single plane case. Instead a number of parameter candidates are determined from one of the planes. We then determine how well a second plane is matched to the image using these parameters. The parameters that give the best total match is chosen.

When a number of x- and y-parameter candidates are determined the same procedure as for the single plane case is performed to find the best combination of x- and y-parameters. Finally the pose of the camera may be computed.

4.1 Experiment 3

A model consisting of two planes were matched to three images according to the above. The result is shown in Figure 7. Here there are great differences in lighting and minor occlusions.

5 Conclusions

We have presented an automatic system for pose-estimation from a single image in a city scene using a 2D or a 3D model. It is difficult to get an objective validation

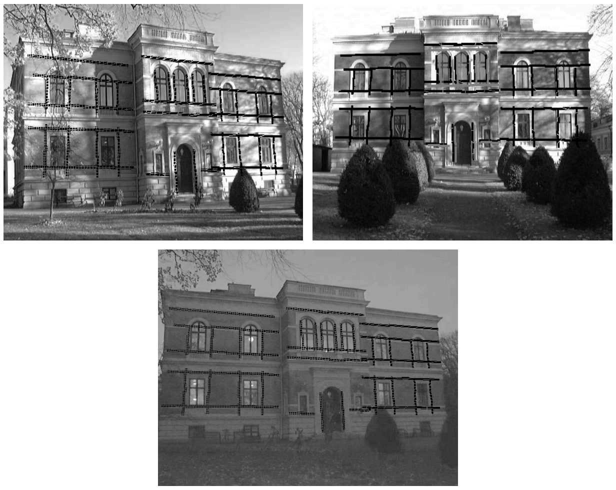

Figure 7. A two-plane model matched to three images.

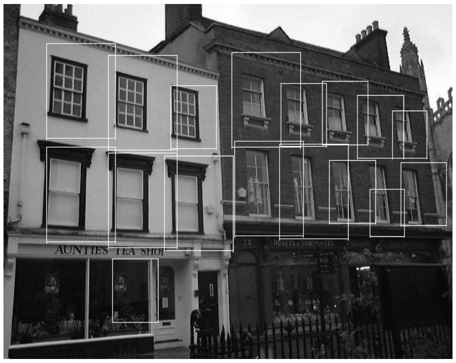

Figure 8. An image processed by a window detection system

of the system, but in a number of experiments the system performs well. It is able to handle different lighting as well as minor occlusions reasonable. For future work, it would be interesting to incoporate algorithms for detecting windows and doors in images. This would surely make the system more robust. Algorithms for detecting windows based on support vector machines has been developed in [4]. Figure 8 shows the detected windows marked with white squares in an image. It would also be interesting to compensate the image for the lighting and to use color information. In a urban scene you have usually access to a map, cf [1]. To incorporate such information in a system would be very challenging.

References

[1] B. Mourrain Bondyfalat and T. Papadopoulo. Using scene constraints during the calibration procedure. In *ICCV'2001*, 2001.

[2] Robert T. Collins and J. Ross Beveridge. Matching perspective views of coplanar structures using projective unwarping and similarity matching. Technical Report UM-CS-1994-006, , 1994.

[3] O. Faugeras. *Three-Dimensional Computer Vision*. MIT Press, Cambridge, Mass, 1993.

[4] B. Johansson and F Kahl. Detecting windows in city scenes. In *submitted to ICPR'02*.

[5] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *In Proceedings of the Conference on Computer Vision and Pattern Recognition*, 1998.

[6] C. Rothwell, A. Zisserman, D. Forsyth, and J. Mundy. Planar object recognition using projective shape representation. In *Int'l, J. Comput. Vision, 16,*, pages 57–99, 1995.

[7] C. A. Rothwell, A. Zisserman, J. L. Mundy, and D. A. Forsyth. Efficient model library access by projectively invariant indexing functions. In *Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 109–114, Champaign, Illinois, 15–18 1993. IEEE Computer Society Press.

[8] F. Schaffalitzky and A. Zisserman. Planar grouping for automatic detection of vanishing lines and points. In *IVC, 18(9)*, pages 647–658, 2000.

[9] P. Sturm and S. Maybank. On plane-based camera calibration: A general algorithm. In *IEEE Conf. Computer Vision and Pattern Recognition*, 1999.

[10] Peter Sturm. Algorithms for plane-based pose estimation. In *CVPR - IEEE International Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina*, pages 706–711, June 2000.

[11] Bill Triggs. Autocalibration from planar scenes. In *EVVC (1)*, pages 89–105, 1998.

[12] T. Tuytelaars, L. Van Gool, M. Proesmans, and T. Moons. The cascaded hough transform as an aid in aerial image interpretation. In *In Proc. ICCV*, pages 67–72, 1998.