Recognizing Objects in Cluttered Images using Subgraph Isomorphism *

Tushar Saxena, Peter Tu and Richard Hartley
CMA Consulting, Schenectady, NY *and*
GE - Corporate Research and Development,
P.O. Box 8, Schenectady, NY, 12301.

Abstract

This paper reports on a new approach to object recognition based on exploiting both geometric and spectral information in an image. An algorithm is described for finding objects in an image based on inexact graph matching against a template that incorporates information about the geometry and color of segmented regions of the image. The image is encoded as an attributed graph in which vertices represent regions, and are annotated with the position, shape and color of the image. Finding the template in a new image takes place in three steps: local, neighborhood and global matching. In the last step a maximal set of mutually compatible template / candidate image region assignments is sought. Currently the system has been evaluated using 3-band color images, though experiments carried out with 4-band images suggest improved performance with such images.

1 Outline

The present paper describes a method for finding objects in images. The typical situation is that one has an image of the object sought. The task is to find the object in a new image, taken from a somewhat different viewpoint, possibly under different lighting. The method used is based on approximate attributed graph matching. As a first step, the image is segmented into regions of approximately constant color. The geometrical relationship of the segmented colored regions is represented by an attributed graph, in which each segment corresponds to a vertex in the graph, and proximate regions are joined by an edge. Vertices are annotated with the size, shape and color of the corresponding segment. Finding an object in a new image then comes down to an approximate graph-matching problem in which a match is sought in the new image for a subgraph approximating the one corresponding to the template. The graph matching can only be approximate, because of the inexactness of the segmentation process, and the changed aspect of the object, due to change of lighting, viewpoint, and possible partial occlusion.

Figure 1: *The result of edge detection in a template image containing a cup.*

2 Extracting Object Faces from Images

The first step in the algorithm is the division of the image into faces (or regions) of approximately constant color. The face extraction process proceeds in the three basic steps, which will be outlined in the following subsections.

2.1 Detecting Approximate Region Boundaries

First the boundaries (that is edges) hypothesized to enclose the region are detected. As a first step in this process, the edges in the image are detected using a Canny-style edge detector, and line segments (larger than a threshold) are fitted to the resulting edgels. It is reasonable to assume that the region boundaries pass through the resulting line segments, since under

*This work was supported by DARPA contract F33615-94-C-1021, monitored by Wright Patterson Airforce base, Dayton OH. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied of the Defense Advanced Research Projects Agency, the United States Government, or General Electric

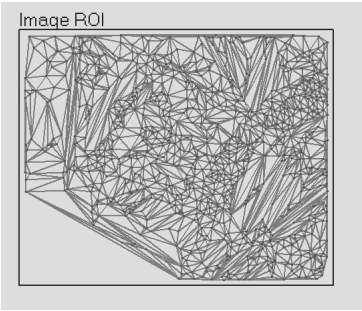

Figure 2: *Constrained triangulation on the adjusted cup lines.*

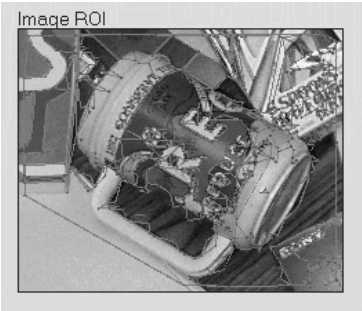

Figure 3: *Faces of the cup extracted by our algorithm.*

our assumptions, a face boundary will produce a discontinuity in the intensity and color variation of the enclosed region, and thus show up during edge detection. However, typically, these boundaries are detected in the form of numerous small broken line segments (see figure 1). It is difficult to identify the exact geometry of the enclosed faces directly from these line segments. To improve the boundary geometries, we use some heuristics to further process the line segments. Some of these heuristics are: merge adjacent near-collinear lines; complete T-junctions and intersections by filling small gaps when appropriate. In our experience, these heuristics aid significantly in correcting most of the degenerate boundary segments.

2.2 Estimating Initial Uniform Regions: Constrained Triangulation

Using the boundary line segments from the previous step, we now generate an initial partition of the image into triangles of uniform intensity and color. This is accomplished by a *constrained triangulation* of the boundary lines. A constrained triangulation produces a set of triangles which join nearest points (end-points of the lines), but respect the constraining boundary lines. That is, each boundary line segment will be an edge of some triangle. Since all triangles are formed from the end-points and lines on the boundaries of the faces, each triangle lies completely inside a face. Moreover, since these triangles cover the whole image, each face can be represented by a union of a finite number of these triangles. As an example, see the result of constrained triangulation on an image segmentation in figure 2.

In the next step, we describe the region merging process, using which we derive the object faces from this initial triangulation.

2.3 Region Merging

In this step, a region-merging procedure is used to incrementally generate the visible object faces in the image. Starting with the triangular regions from the constrained triangulation, neighboring regions are successively merged if they have at least one of the following properties:

1. **Similar color intensities:** Two adjacent regions are merged if the difference between their average color intensity vectors is less than a threshold. This is a reasonable merging property since neighboring faces in objects are at angles to each other, and are likely to cast images of different intensities. As a refinement of this method, one could merge two regions based on a decision of which of two hypotheses (the two regions are separate; the two regions should form a single region) is preferable based on the color statistics of the regions. In addition, a linear or more complex color gradient over a face could be modelled. These methods have been suggested in [7] but we have not tried them yet.

2. **Unsupported bridge:** Two adjacent regions are merged if the percentage of edges common between them, which are *unsupported*, is larger than a threshold. An edge is said to be supported if a specified percentage of its pixels belong to an edgel detected by the edge detector. Merging based on this property will ensure the inclusion of those boundary segments which were missing from the set of line segments derived from edgels.

After each merge, the properties (size and color) of the new, larger region are recomputed from the properties of the two regions being merged.

The merging iterations continue until the color intensities of each pair of neighboring regions are suffi-

ciently different, and most of the edges common between them have support from the segmentation. As an illustration, see figure 3. There, our complete merging process extracts faces corresponding to the cup. The merging process is followed by a cleaning-up phase to remove residual small and narrow regions that slight variations of color have prevented from being merged with adjacent larger regions. The result of the segmentation and merging algorithm is a set of regions with associated color (RGB) values.

3 Deriving Graph Representations of Objects

Once all the object faces in the image have been generated, they are represented as a graph that captures the relative placements of the faces in the image.

Each vertex in the graph represents a region, and is annotated with its shape, position and color attributes. Shape is represented by the moment matrix of the region, from which one may derive the area, along with the orientation and ratio of the principal axes of the region. In effect, the region is being represented as an ellipse. This shape representation is of course an extremely rough representation of the shape of the region. However, it is also quite forgiving of variations of shape along the boundaries, or even a certain degree of fragmentation of the region. Since matching will not be done simply on the basis of a region-to-region match, but rather on matching of region clusters, this level of shape representation has proven to be adequate. More precise shape estimates have been considered, however. Their use must be dictated by the degree of accuracy and repeatability of the segmentation process. The color of the region is represented by a spectral (in 3 band images, RGB) vector. Other color representations are of course possible, and have been tried by other authors ([7, 8]).

Because of the possibility of regions being fragmented or regions being improperly merged, it turns out to be inappropriate to use edges in the graph to represent physically adjacent regions. The adjacency graph generated by such a rule is too sensitive to minor variations in the image segmentation. Instead the choice was made of joining each vertex to the vertices representing the N closest regions in the segmented image. A value of $N = 8$ was chosen. Thus, each vertex in the graph has 8 neighbors.

4 The three-tier matching method.

The reduction of the image to an attributed graph represents a significant simplification. The graph corresponding to a typical complicated image (the search image) may contain up to 500 or so vertices, whereas the graph corresponding to an object to be found (the template) may contain 50 vertices or so. Thus a complete one-on-one comparison may be carried out in quite a short time.

The search is carried out in three phases, as follows:

1. **Local comparison.** A one-to-one comparison of each pair of vertices is carried out. Each pair of vertices, one from the *template graph* and one from the *search graph* is assigned a score based on similarity of shape, size and color, within rather liberal bounds.

2. **Neighborhood comparison.** The local neighborhood consisting of a vertex and its neighbors in the template graph is compared with a local neighborhood in the search graph. A score is assigned to each such neighborhood pairing based on compatibility, and the individual vertex-pair scores.

3. **Global matching.** A complete graph-matching algorithm is carried out, in which promising matches identified in the stage-2 matching are pieced together to identify a partial (or optimally a complete) graph match.

Each of these steps will be described in more detail in later sections. The idea behind this multistage matching approach is to avoid ruling out possible matches at an early stage, making the matching process robust to differences in the segmentation and viewpoint.

4.1 Local matching.

In local matching, individual vertex pairs are evaluated. Each pair is assigned a score based on shape and color. Recall, that each region is idealized as an ellipse. Shapes are compared on the basis of their size and eccentricity. Up to a factor of 2 difference in size is allowed without significant penalty. This allows for different scales in the two images, within reasonable bounds.

Because of different lighting conditions, colors may differ between two images. The most significant change in color, however is due to a brightness difference. To allow for this, colors are normalized before being compared. The color of a region is represented by a vector, and vectors that differ by a constant multiple are held to represent the same color. The cost of a local match between two vertices is denoted by C_{local}.

4.2 Neighborhood matching.

Each vertex (here called core) in the graph has eight neighbors representing the eight closest regions. In comparing the local neighborhood of one core vertex

v_0 with the local neighborhood of a potential match v'_0, an attempt is made to pair the neighbor nodes of v_0 with those of v'_0. In this matching the order of the neighbor vertices must be preserved. Thus, let $v_1, v_2, \ldots, v_n$ be the neighbors of one core vertex, given in cyclic angular order around the core, and let $v'_1, \ldots, v'_m$ be the neighbors of a potential match core, similarly ordered. One seeks subsets S of the indices $\{1, \ldots, n\}$ and S' of the indices $\{1, \ldots, m\}$ and a one-to-one mapping $\sigma : S \to S'$ so that the matching $v_i \leftrightarrow v'_{\sigma(i)}$ preserves cyclic order. The total cost of a neighborhood match is equal to

$$C_{\text{nbhd}} = w_0 * C_{\text{local}}(v_0, v'_0) + \sum_{i \in S} C_{\text{local}}(v_i, v'_{\sigma(i)}) w_i$$

where w_i is a weight between 0 and 1 that depends on the ratio of distances between the core vertices and the neighbors v_i and $v'_{\sigma(i)}$. For each pair of core vertices v_0, v'_0, the neighborhood matching that maximizes this cost function is speedily and efficiently found by dynamic programming.

4.3 Graph matching

In previous sections, the template image and the search image were reduced to a graph, and candidate matches between vertices in the two images were found. The goal of this section is to generate a mutually consistent set of vertex matches between the template and the search image. An association graph $\mathbf{G}$ [2, 4] provides a convenient framework for this process. In considering the association graph, it is important not to confuse it with the region adjacency graph that has been considered so far. In the association graph, vertices represent pairs of regions, one from each image. Such a vertex represents a hypothesized matching of a region from the template image with a region from the search image. Weighted edges in the association graph represent compatibilities between the region matchings denoted by the two vertices connected by the edge.

Thus, a vertex in the association graph is given a double index, and denoted v_{ij}, meaning that it represents a match between region R_i in the template image and region R'_j in the search image. This match may be denoted by $R_i \leftrightarrow R'_j$. As an example, if $j_1 \neq j_2$ then v_{ij_1} is not compatible with v_{ij_2}. This is because vertex v_{ij} represents a match $R_i \leftrightarrow R_{j_1}$ and v_{ij_2} represents the match $R_i \leftrightarrow R_{j_2}$, and it is impossible that region R_i should match both R'_{j_1} and R'_{j_2}. Thus, vertices v_{ij_1} and v_{ij_2} are incompatible and there is no edge joining these two vertices in the association graph. There are other cases in which matches are incompatible. For instance, consider a vertex v_{ij} representing a match $R_i \leftrightarrow R'_j$ and a vertex v_{kl} representing a match $R_k \leftrightarrow R'_l$. If regions R_i and R_k are close together in the template image, whereas R'_j and R'_l are far apart in the search image, then the matches $R_i \leftrightarrow R'_j$ and $R_k \leftrightarrow R'_l$ are incompatible, and so there is no edge joining the vertices v_{kl} and v_{ij}. Matches may also be incompatible on the grounds of orientation or color.

Formally, the association graph $\mathbf{G} = \{\mathbf{V}, \mathbf{E}\}$ is composed of a set of vertices $\mathbf{V}$ and a set of weighted edges $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$. Each vertex v represents a possible match between a template region and a search region. If there are N template regions and M search regions then $\mathbf{V}$ would have NM vertices (see figure 4). In order to reduce the complexity of the problem, the graph $\mathbf{G}$ is pruned so that only the top 5 assignments for each template region are included in $\mathbf{V}$. These nodes are labeled v_{ij} which is interpreted at the jth possible assignment for the ith template region. A slack node for each template region is inserted into the graph. The slack node v_{i0} represents the possibility of the NULL assignment for the ith template region. If an edge $e = (v_{ij}, v_{kl})$ exists then the assignments between nodes v_{ij} and v_{kl} are considered compatible. The weights for the edges are derived from the compatibility matrix $\mathbf{C}$ which is defined as:

$$C_{(ij)(kl)} = \begin{cases} 0 & \text{if } j = 0 \text{ or } k = 0 \\ 0 & \text{if } i = k \text{ and } j \neq l \\ 0 \text{ to } 1 & \text{if } (i, j) = (k, l) \\ 0 \text{ to } 1 & \text{if } v_{ij} \text{ and } v_{kl} \text{ are compatible} \\ -N & \text{if } v_{ij} \text{ and } v_{kl} \text{ are not compatible} \end{cases}$$

Where N is the number of template regions. The value of $C_{(ij)(ij)}$ represents the score given to the individual assignment defined by node v_{ij}. A subgraph of $\mathbf{G}$ represents a solution to the matching problem. In section 4.5 criteria for selecting the solution set will be described.

4.4 Compatibility scores

The method of assigning compatibility scores is as follows. Consider a candidate region pair $R_i \leftrightarrow R'_j$. The local neighborhood of region R_i has been matched with neighborhood R'_j during the neighborhood matching stage. In doing this, a set of neighbors of the region R_i have been matched with the neighbors of the region R'_j. This matching may be considered as a correspondence of several regions (a subset of the neighbors of R_i) with an equal number of regions in the other graph. From these correspondences a projective transformation is computed that maps the centroid of R_i to the centroid of R'_j while at the same time as nearly as possible mapping the neighboring regions of R_i to their paired neighbors of R'_j. Thus, the neighborhood correspondence is modelled as closely as

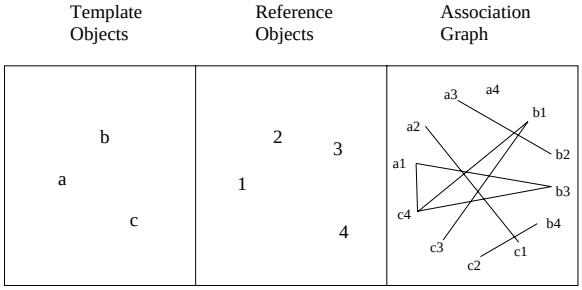

Figure 4: *The template and search images are reduced to a set of regions. Each possible pair of assignments are assigned to a node in the association graph. Edges in the graph connect compatible assignments. Note that although match b1 is compatible with c4 and c3, c3 and c4 are not compatible. The best solution is the clique* $\{a1, b3, c4\}$ *representing a match of regions a, b and c from the first image with regions* $1, 2, 3$ *from the second.*

possible by a projective transformation of the image. Let H be the projective transformation so computed.

Now let $R_k \leftrightarrow R'_l$ be another candidate region match. To see how well this is compatible with the match $R_i \leftrightarrow R'_i$, the projective transformation H is applied to the region R_k to see how well $H(R_k)$ corresponds with R'_l. As a measure of this correspondence, the vector from $R'_j R'_l$ is compared with the vector $R'_j H(R_k)$. A compatibility score is assigned based on the angle and length difference between these two vectors. The two assignments are deemed incompatible if the angle between the two vectors exceeds 45 degrees, or their length ratio exceeds 2.

A color compatibility score is also defined. The correspondence of a core vertex and its neighors with the matched configuration in the other image can be used to define an affine transformation of color space from the one image to the other. An affine color transformation is a suitable model for color variability under different lighting conditions ([9]). The affine transformation defined for one matched node pair is used to determine whether another matched node pair is compatible.

4.5 Clique finding

A popular graphical approach which can take advantage of some of the information contained in the edge structure of the association graph is a node clustering technique where a simple depth first search is used to determine the largest connected subgraph of $\mathbf{G}$, that is, the largest set of compatible region matches. A connected graph is defined such that a path of edges exists between every pair of nodes in the graph. This solution represents a certain amount of consistency. However, The statement that node a is consistent with node b and node b is consistent with node c does not necessarily imply that node a is consistent with node c. This leads to the conclusion that in order to take full advantage of the mutual constraints embedded in the association graph, the final solution should represent a clique on $\mathbf{G}$.

A subset $\mathbf{R} \subseteq \mathbf{V}$ is a clique on $\mathbf{G}$ if $v_{ij}, v_{kl} \in \mathbf{R}$ implies that $(v_{ij}, v_{kl}) \in \mathbf{E}$. The search for a maximum clique is known to be an NP complete problem [6]. Even after pruning, the computational costs associated with exhaustive techniques such as [1] would be prohibitive. It has been reported [3] that determining a maximum clique is analogous to finding the global maximum of a binary quadratic function. Authors such as [10, 11] have taken advantage of this idea by using relaxation and neural network methods to approximate the global maximum of a quadratic function, where this maximum corresponds to the largest clique in the association graph. Although the largest clique, which is based on the information contained in $\mathbf{E}$, ensures a high level of mutual consistency, the nuances of the compatibility measures in $\mathbf{C}$ are lost. In order to take advantage of the continuous nature of these edge strengths, a quadratic formula is specified where the global maximum corresponds to the clique that has the maximum sum of internal edge strengths. An approach based on Gold and Rangarjans's gradual assignment algorithm (GAA, [5]) is used to estimate the optimal solution. The GAA is an iterative optimization algorithm which treats the problem as a continuous process but converges to a discrete solution. Even though the solution might be generated based on a local maximum this solution will be guaranteed to be a maximal clique, because of the choice of the compatibility measure which strongly discourages incompatible matches. Although generation of a maximal clique is an NP complete problem, nevertheless the quadratic function approach provides a way of rapidly finding an approximate solution. The problem is formulated in terms of minimizing a continuous quadratic form via a descent algorithm, and then gradually imposing constraints to converge towards a binary solution. Details of the clique-finding algorithm are given in [13].

5 Results

The system was evaluated on several sets of images. Figures 5 and 6 show the results of searching in different types of 3-band images.

6 More advanced color models.

The color constancy model used in the existing algorithm is based on the observation ([9]) that colors

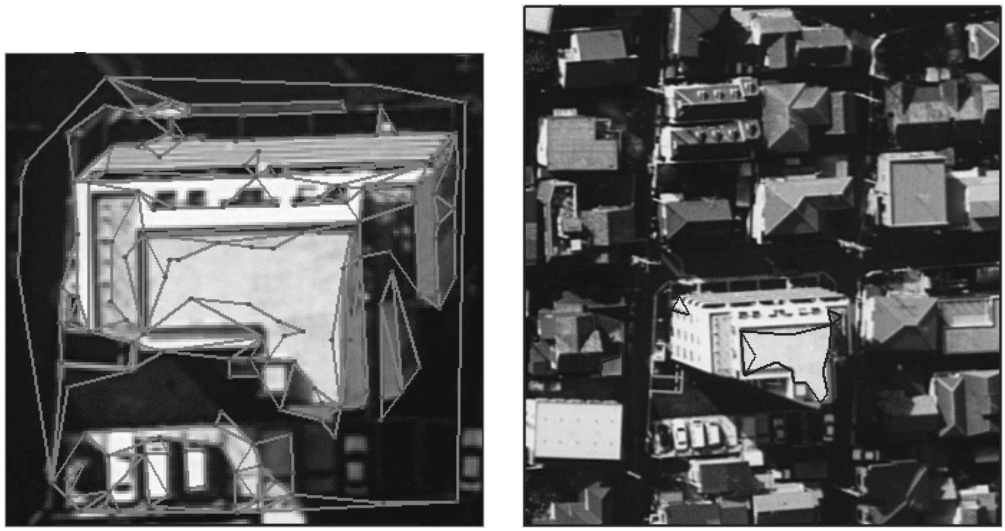

Figure 5: *Recognizing a building. On the left the template, and on the right the search image showing the recognized building.*

Figure 6: *Recognition of cup image. On the left is the template, in the center the search image and on the right the identified regions of the located cup. Note that the cup in the search image is seen from a different angle from the template image.*

are modified by an affine transformation under different lighting conditions. Thus, the color of a core vertex and its neighbors may be modified by an unknown (but constant over the image) affine transformation. This observation is used in defining the assignment compatibility function. At the local matching level, however, single region pairs are compared, using simple color scaling in color comparison. Our intention is to use a more sophisticated color comparison method based on the methods of Healey and Slater ([12]). In this paper, multi-band spectral variation ise modelled as a linear subspace of the high-dimensional color space. We intend to apply this method to 4-band imagery. Use of this technique should give better local region matching. In addition, exploiting the extra band should give better results that we have been able to achieve with three-band color imagery.

To support this expectation, we have implemented a color-segmentation algorithm based on modelling colors as one and two-dimensional subspaces in the 4-dimensional color space. Mahalanobis distance is used to determine whether a pixel belongs to a given color cluster or not. The results of these experiments are shown in figure 7 and 8.

7 Conclusion.

The amalgamation of region segmentation algorithms with modern color constancy methods gives the possibility of improved object recognition in color and multi-spectral imagery. The adoption of an inexact graph-matching approach makes recognition independent of moderate lighting and view-point changes.

References

[1] Ambler A.P., Barrow H.G., Brown C.M., Burstall R.M., Popplestone R.J., 'A versatile computer-controlled assembly system', IJCAI, pages 298-307, 1973.

[2] Ballard D.H., Brown C.M., 'Computer Vision', Prentice-Hall, Englewood Cliffs, NJ, 1982.

[3] Batahan F., Junger M., Reinelt G., 'Experiments in Quadratic 0-1 programming' Mathematical Programming, vol. 44, pp. 127-137, 1989.

[4] Faugeras O.,'Three-dimensional computer vision', MIT Press, 1993.

[5] Gold S. and Rangarjan A., 'A gradual assignment algorithm for graph matching', IEEE Transactions on PAMI, Vol. 18 No 4, April 1996, pages 377 – 387.

[6] Gibson A.,'Algorithmic graph theory', Cambridge University Press, Cambridge (MA), USA, 1985

[7] Allen R. Hanson and Edward M. Riseman, 'Segmentation of Natural Scenes', in Computer Vision Systems, (edited A. Hanson and E. Riseman), Academic Press, 1978, pages 129 – 164.

[8] Allen R. Hanson and Edward M. Riseman, 'VISIONS : A computer system for interpreting scenes', in Computer Vision Systems, (edited A. Hanson and E. Riseman), Academic Press, 1978, pages 303 – 334.

[9] Glenn Healey and David Slater, 'Computing Illumination-Invariant Descriptors of Spatially Filtered Color Image Regions', IEEE Transactions on Image Processing, Vol. 6 No 7, July 1997, pages 1002 – 1013.

[10] Lin, F.'A parallel computation network for the maximum clique problem', Proceeding 1993 international symposium on circuits and systems, pp. 2549-52, vol. 4, IEEE, May 1993.

[11] Pelillo M., 'Relaxation labeling Networks that solve the maximum clique problem', Fourth international conference on artificial neural networks, pp. 166-70, published by IEE June 1995.

[12] David Slater and Glenn Healey, Exploiting an Atmospheric Model for Automated Invariant Material Identification in Hyperspectral Imagery, preprint.

[13] Peter Tu, Tushar Saxena and Richard Hartley, 'Recognizing objects in cluttered images using subgraph isomorphism', to appear in Springer LNCS series, conference Palermo, Sicily, 1998.

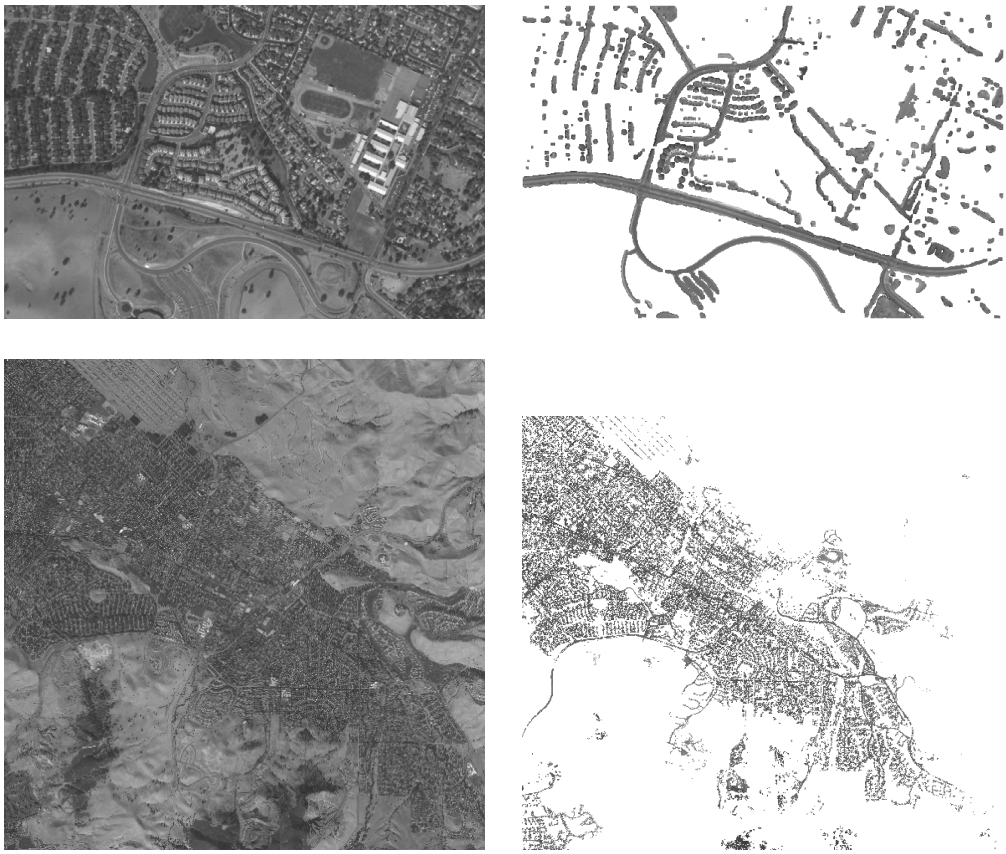

Figure 7: **Top.** *Extraction of roads from four-band imagery. In the four band image (left) colours are quite muted (though this is not evident from this grey-scale image), and different surface types are hard to distinguish on the basis of 3-band color. Roads may be extracted from the four-band imagery (RGB + near infrared) based on pixels' spectral content. A sampled region of road is analyzed and modelled as a 2-dimensional or (as here) 1-dimensional space passing through the origin. Pixels are then classified as being road or non-road based on Mahalanobis distance from the classifying hyperplane. The image is then filtered to remove scattered wrongly-classified points. The roads stand out clearly in the processed image (right).* **Bottom.** *The same experiment on a larger scale image. Pixels are classified as road/non-road. The agglomeration of roads in built-up areas is evident. Zooming in reveals details of the road structure.*

Figure 8: *This figure shows the trade-off between using a 1-dimensional (left) or a 2-dimensional (right) linear subspace to identify road pixels. The 2-dimensional classifier allows more natural and lighting variability of the pixel color. This has the effect of picking up more slightly differently colored road regions, while at the same allowing other non-road pixels to be wrongly classified.*