SRI International

# IMAGE-TO-IMAGE CORRESPONDENCE: LINEAR-STRUCTURE MATCHING

Technical Note No. 331

July 13, 1984

By: Grahame B. Smith, Senior Computer Scientist
Helen C. Wolf, Computer Scientist

Artificial Intelligence Center
Computer Science and Technology Division

SRI Projects 4748 and 5355



SRI International®

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

## Abstract

We examine the task of matching images of a scene when they are taken from very different vantage points, when there is considerable scale change, and when the image orientations are unknown. We use the linear structures in the scene as the basis of our correspondence procedure. This paper considers the problem of describing the linear structures in a manner that is invariant relative to the variations that can occur among images, and discusses a method of finding the best description of the linear structures.

# 1. Introduction

When the human visual system is presented with two views of a single scene, it determines the relative viewing positions of the two images and brings the latter into correspondence. That is, the relationship of each image to the scene is understood so that both images can be used as information sources for further processing. This human ability functions well over a wide range of viewing positions and conditions. It is this ability to place two very different views of a single scene into correspondence that we address in this paper.

We should draw a distinction between two forms of the image correspondence task. Traditionally, image registration has been a task undertaken by photogrammetrists. One application involves registering an image to a map so that new information, present in the image, may be transferred to the map. Another is the registration of the two images of a stereo pair so that disparity information can be extracted. In each of these tasks the two images, (or, in the first instance, the image and the map), are similar in terms of both their viewing position and their scale. The techniques used for registering the two images are point-based. A feature point in one image is matched to the same feature point in the other image. In automated systems this is achieved by selecting a small window about the feature in one image and then correlating this window with one in the second image. If there is little distortion or occlusion, this technique performs well; it has become the basis of current automated image-registration systems.

---

The research reported herein was supported by the Defense Advanced Research Projects Agency under Contract MDA903-83-C-0027 and by the National Aeronautics and Space Administration under Contract NASA 9-16664. These contracts are monitored by the U.S. Army Engineer Topographic Laboratory and by the Texas A&M Research Foundation for the Lyndon B. Johnson Space Center.

The other form of the image correspondence task seeks to find the relationship among views that differ widely in vantage point, scale, etc. We will refer to this as the correspondence task, and use registration as the name for the form of the task outlined above. In correspondence tasks there is significant distortion between the images, the scale may differ and may not even be constant across a single image, as is the case in oblique aerial imagery, occlusion is common, and the response of the various sensors to a single feature differs greatly. Feature point matching, as used in image registration, is prone to error. However, feature point matching is not the only means of placing images into correspondence. It appears that the human visual system makes use of nonpoint features, such as linear structures and extended landmarks. The aspects of our investigation reported here utilize the linear structures of the images as the prime elements for achieving correspondence.

In classifying the methods that could be employed to find linear structures in images, we draw a distinction between techniques that use semantic information and those that do not. If, for example, we apply a road operator to locate some of the linear structures in an image, that operator has had built into it semantic knowledge about the appearance of roads. We could proceed in this manner and build comparable operators for all the scene objects that manifest themselves as linear structures in images. Alternatively, we could seek to find the linear structures in an image without "identifying" their nature. In this case, we identify the image behaviour interpreted by us as a linear structure without knowledge of the world objects that gave rise to that structure. We choose this latter course because we wish to establish the correspondence among images without first having to identify the scene objects.

The correspondence task is carried out in three stages: we must find the linear structures, we must build their descriptions and, finally, we must match these descriptions. The details of the first stage is reported in Fischler and Wolf [1]. In this paper we explain how those procedures are employed in the correspondence task. We present a detailed account of our implementation of the second stage – namely, building structure descriptions – along with an outline showing how these descriptions are to be used in the final matching stage.

## 2. Finding the Linear Structures

Descriptions of the semantically free procedures we use to find linear structures in images can be found in Fischler and Wolf[1]. In essence, these procedures first find those pixels whose intensity levels are local maximums and minimums, then cluster such pixels and identify the minimal spanning tree for each cluster. The long paths in each of the spanning trees are found, whereupon these form the basis for the linear structure reported by the procedures. The results of applying these procedures are shown in Figures 1-4. Figure 1 is a natural-color oblique view of the Eel river in northern California; Figure 2 is a vertical infrared view of the same scene. Each was scanned through red, green, and blue filters; the results of the procedures for finding linear structures in each of these separation images are shown in Figures 3(a),3(c),3(e) and 4(a),4(c),4(e). In addition, the red, green, and blue separation images were combined into images of hue, saturation, and intensity; these were also processed to find the linear structures contained in them. The results are shown in Figures 3(b),3(d),3(f) and 4(b),4(d),4(f).

**Figure 1.** Oblique Natural-Color Image of the Eel River

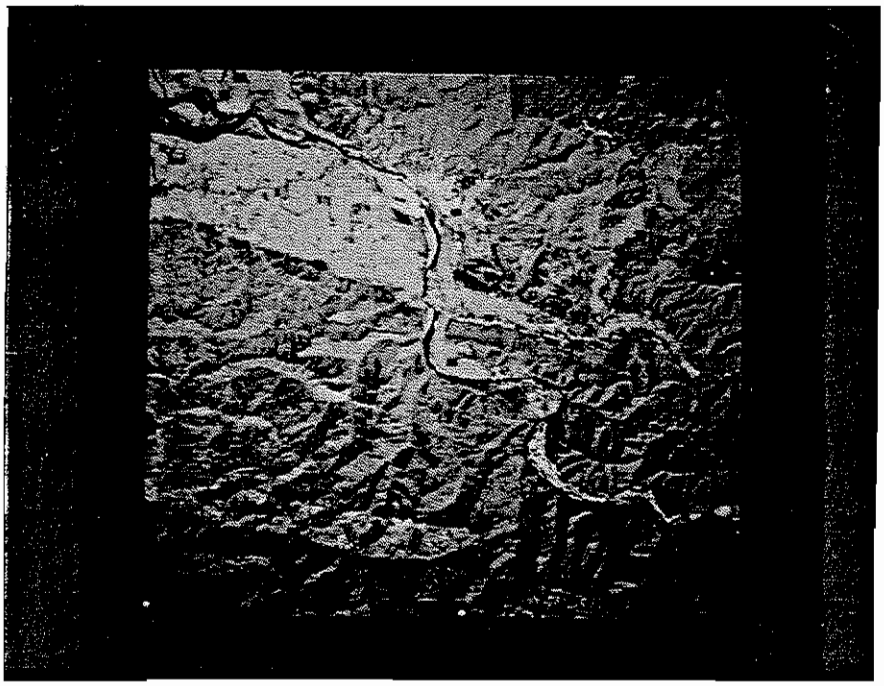

**Figure 2.** Vertical Infrared Image of the Eel River

These separation images differ appreciably in their linear structure. Certainly no one separation image can be selected as providing a complete delineation of the river. The philosophy we adopt is to view the original image from as many

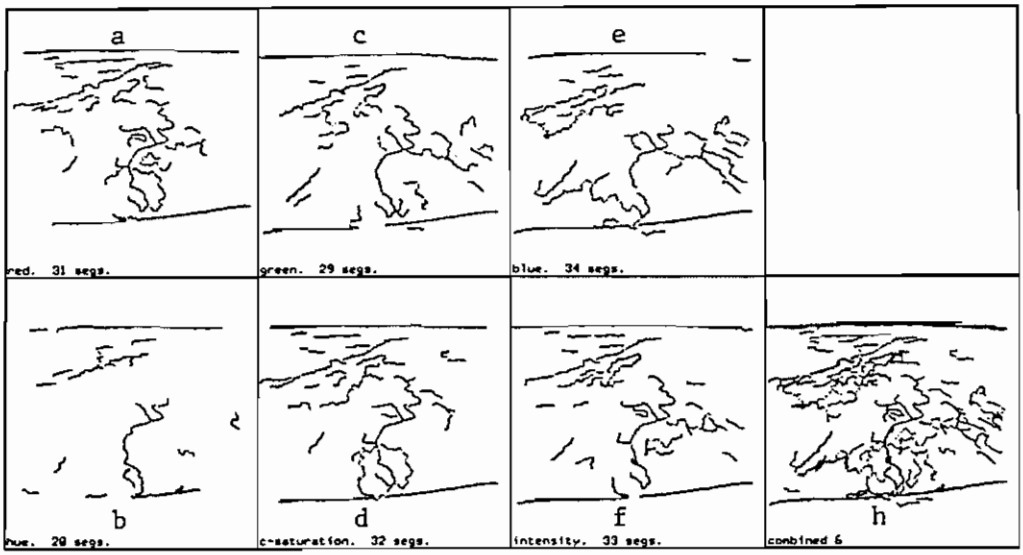


**Figure 3.** Linear Structure in the Oblique Image

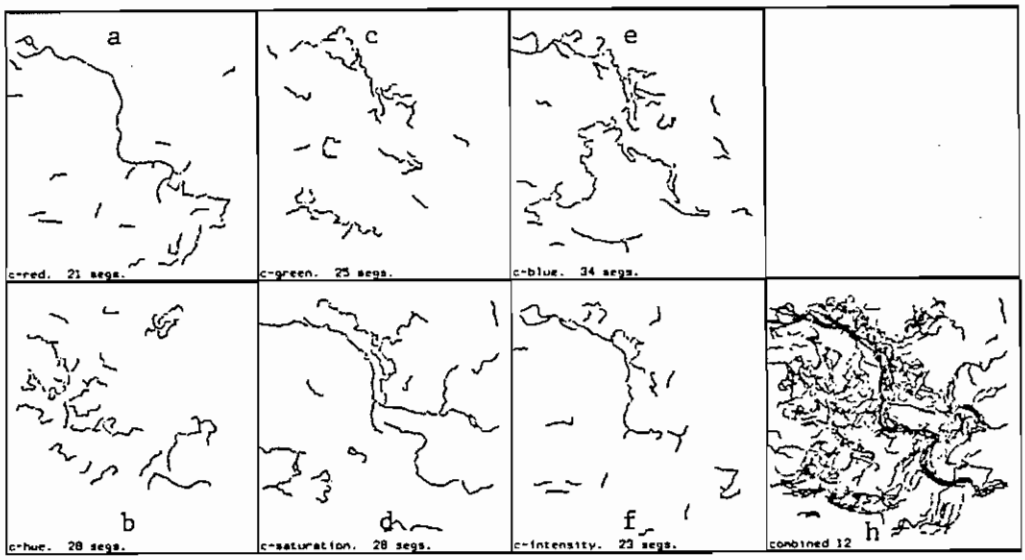


**Figure 4.** Linear Structure in the Vertical Image

perspectives as possible, obtaining the linear structures as seen from each of these. That is, we look for structures in hue, in the green spectral band, and so on. Of course, the hue image is derived from the red, green, and blue images, and contains only redundant information, but this presentation of the information may show structure that was masked in other presentations. In this sense, the additional

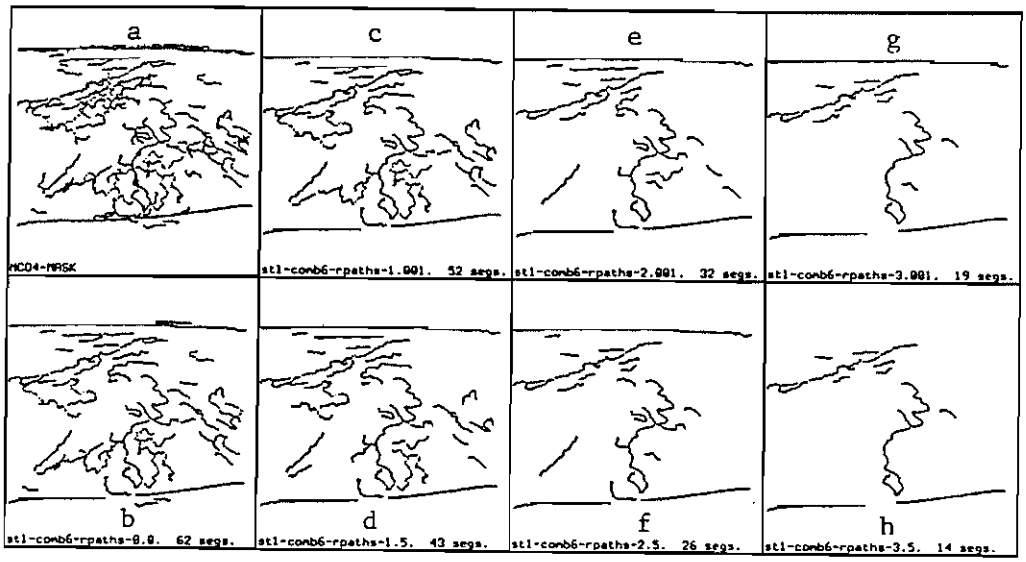


**Figure 5.** Linear Structure in the Composite Oblique Image

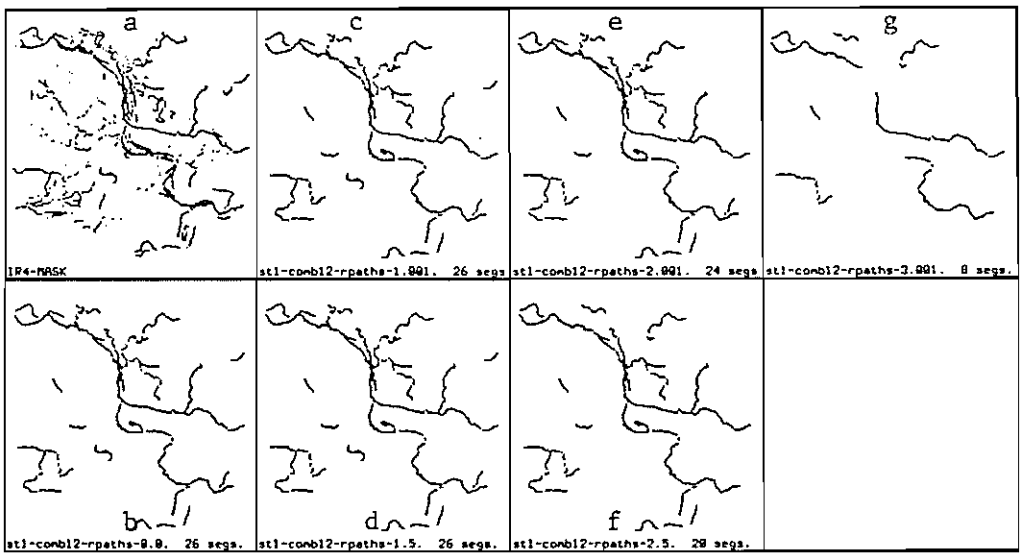


**Figure 6.** Linear Structure in the Composite Vertical Image

perspectives provide new information on which the linear-structure finders can act. The results of combining the linear structures extracted in all the various perspectives are shown in Figures 3(h) and 4(h). Clearly, some of this structure comes from shading effects rather than from physical structure in the scene. We need to separate the real physical structure from all else.

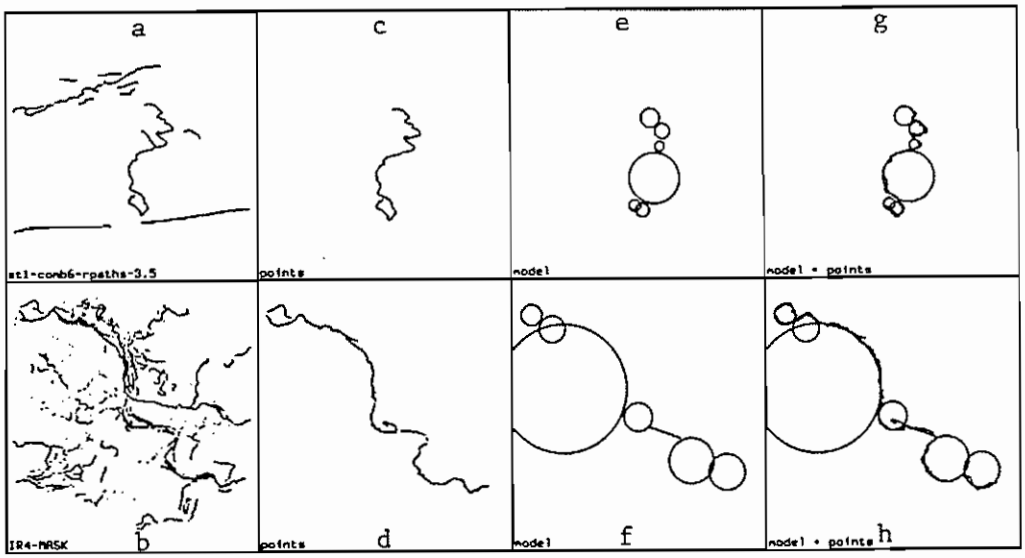


**Figure 7.** Structure Descriptions

Figures 3(h) and 4(h) were obtained by adding the binary images produced by the linear-structure finders. Consequently, in the combined image the values are greater than one at those pixel positions where linear structure was seen in more than one separation image. We treat this combined produce as a new "grey-level" image and, once again, apply the linear-structure finders. The results obtained from applying these procedures to Figures 3(h) and 4(h) are depicted in Figures 5(b) and 6(b). Figures 5(a) and 6(a) show an intermediate result before we cull short structures. For each of the structures in Figures 5(b) and 6(b), we calculate the average "intensity", that is the average number of original separation images exhibiting that linear structure. Figures 5(c),5(d),5(e),5(f),5(g),5(h) and 6(c),6(d),6(e),6(f),6(g),6(h) reveal which segments would remain if we thresholded the "intensity" values at 1, 1.5, 2, 2.5, 3, and 3.5, respectively.

We build a description of the linear structures from one of these images. The image we use will depend on the final matching procedure. If we wish to attempt

to first match the "strongest" structures we use the image resulting from a high threshold. On the other hand, if we wish to match the complete structure, the unthresholded image would appear to be more appropriate. In the next section, where we discuss the nature of the structure description, we use as examples the foregoing two extremes. In the case of the oblique image, we have used the "intensity" image at a threshold of 3.5 (Figure 7a), while for the other case, the vertical infrared image, we employ the unthresholded image (Figure 7b).

## 3. Describing the Linear Structures

The means used to describe a linear structure is not independent of the use to which this description will be put. A description that makes it possible to reproduce the structure is different from one that is sufficient to recognize it. As matching is our goal, we want a description that is general enough to be unaffected by noise in the data, but specific enough to distinguish among structures that the human visual system would classify as different. To the extent feasible, the description must be invariant with respect to the variations that can occur in the data. Specifically, we want the description to be independent of orientation, scale, and vantage point.

Our matching process will compare graphs of symbolic descriptions. We will use as little metric information as possible. Consequently, the descriptions we employ are symbolic ones, the primitive entities in each of which have qualities that are themselves symbolic. For example, a primitive may be a straight-line segment whose properties, such as an intersection angle (with some other primitive), have values *acute*, *near-colinear*, etc. rather than a value in degrees.

The primitives we have chosen to use are straight-line segments, arcs of circles, and model-less, that is, data we prefer to describe as indescribable, data for which the data set itself is the most apt description. The choice of these few primitives stems from the observation that human description of linear structures seems to be based on curves and straight lines – moreover on whether adjoining curves curve the same or opposite ways and whether adjoining pieces of the structure intersect in particular ways. It is also a fact that humans find certain parts of the structure too difficult to describe, and assign them some generic term like "wiggles".

Selection of the description primitives is only half the task of description building. We need to be able to divide the linear structure into parts and assign a primitive to each. Usually the task of dividing the linear structure into parts and describing each of these parts has been handled as two relatively independent processes in which partitioning has preceded parts description. The difficulty with this approach is that some characterization of the breakpoints between parts has to be found. Generally, this characterization is based only on local properties of the linear structure, even though neighborhood information or local inhibition may be employed so as to benefit from more broadly based information. In this respect, the task of describing a structure in terms of its primitive parts appears to have been replaced by the more difficult undertaking of describing breakpoints. Our concern is to find the "best" description without first having to find the "best" subdivision. Furthermore, we would like "best" to be defined in terms of a global criterion rather than local properties of the structure.

The advantage of defining best in terms of a local criterion is that many candidates for the definition of "best" spring to mind. The disadvantage of defining

"best" in a global sense is the lack not only of likely definitions, but also of computationally effective algorithms for finding this optimal solution. However, a description that views the data from a "gestalt" perspective seems more likely to be independent of image orientation, scale, and vantage point than one that applies local data measures to define the optimal description. We define best description, as the one that minimizes the number of symbols needed to encode the linear structure in terms of our description primitives.

## 4. Minimal Encoding

The need to match data to description primitives is a central aspect of decision theory and pervades artificial intelligence research. It is a human's ability to abstract data in terms of descriptive models that distinguishes human information processing from its electronic namesake. Effective data abstraction is a balance between two competing requirements. On the one hand a descriptive model must fit the data adequately, while, on the other, the descriptive model must not be needlessly complex. The criterion we use to select among competing descriptions is based on the work of Georgeff and Wallace [2], in which the description considered "best" is the one that can be encoded in the fewest symbols.

Suppose we wish to send data to some receiver so that he can recreate the data to some preselected level of resolution. The sender and receiver have agreed on a language for this communication that consists of a set of primitive elements. What is the most efficient encoding of the data; which message has the minimal encoding length? Consider the example of sending a message that describes a linear

structure. The latter can be thought of as a list of $x$ and $y$ coordinates. Let us further suppose that the language of communication contains three primitives: straight-line-segments, arcs-of-circles, and model-less-segments. Is it more efficient to send the data as a single model-less-segment primitive, that is, as a list of ($x$,$y$) coordinates, or might it be more efficient to describe the data by one or more of the other primitives, specifying sufficient information to describe how the actual data differ from the primitives?

The message can be viewed as a list

$$((M_1, D_1), (M_2, D_2), ...) \quad ,$$

where $M$ is the specification of the primitive, $D$ the specification of the data in terms of the selected primitive $M$. Let us consider an example. Suppose we have a data set that approximates a straight-line segment. We could communicate this by specifying a straight-line-segment primitive $M$, where $M$ consists of a code for the straight-line-segment primitive and parameters that specify the actual straight line segment. These parameters might be the endpoints of the line. We also need to specify the actual data in terms of this primitive $M$. The data specification $D$ might, for each data point, specify its coordinates as a distance along the line (from its centre) and the perpendicular distance from the point to the line.

As the expected distances from the points to the line are small, we shall choose an encoding of these distances so that the more probable of these, the smaller distances, are encoded in fewer symbols (or bits) than those that are less likely. In the actual examples we shall describe later, we assumed a Gaussian distribution for these perpendicular distances and we encoded optimally for that distribution. The

optimal encoding length is just the negative logarithm of the probability, i.e., the function denoted as "information" in information theory.

If we have a small number of data points fewer symbols may be needed to communicate the data as a list of points; if, however, there is a large number of data points that exhibit behaviour consistent with a primitive, it will probably be cheaper to encode this data set as the primitive and then specify the data in terms of that primitive. Of course we are not just comparing the encoding of all the data with either one primitive or another. It might be more efficient to encode the data as a few primitives, with each primitive "explaining" a different part of the data. The encoding we select is the one that is globally best in explaining all the data.

A way of viewing the message form outlined above,

$$((M_1, D_1), (M_2, D_2), ...) \quad ,$$

is to look upon $M$ as the overhead of introducing another primitive while $D$ represents the quality of the fit between the data and the primitive. Of course, since different primitives have different $M$'s, $M$ also weights each primitive's efficiency for encoding data. In comparing message length we are balancing the complexity introduced by adding an extra primitive to the description of the data against the quality of fit between the assembled primitives and the data values.

Although the above discussion focused on encoding messages for communication, we use minimal encoding length as the criterion for finding the best description of a linear structure – without any interest on our part in actually transmitting the data. This of course means that we only have to decide how many symbols would be used if we were to encode the linear structure in a particular manner rather

than actually doing the encoding. We can use the results of information theory to determine the optimal encoding length without even having to understand what the optimal encoding scheme is. That is, information theory gives us an operator, or a measure, that we can apply to a description to determine how many symbols we would need if we were to encode it optimally, without any consideration of the actual encoding scheme and without the need to do the encoding.

Let us consider our application, encoding linear structures in terms of three primitives: straight-line-segments, arcs-of-circles, amd model-less-segments. We will assume that the data are specified on a $N$x$M$ grid, and that the noise in the data will induce a Gaussian distribution of the data points around the generating primitive. Given that all grid points are equally likely, the cost in bits of encoding a grid point is $logN + logM$, ($log$ is to the base 2). Now consider the three alternative ways of encoding $r$ data points (using one primitive only).

**Model-less-segment:**

We need a code to specify that the primitive being used is the model-less-segment. As there are only three primitives, and we assume that they are all equally likely, it costs $log3$ bits to specify the code. Specification of the data in terms of this primitive will require in turn that we specify $r$ grid coordinates, that is, a cost of $r(logN + logM)$ bits.

**Straight-line-segment:**

We can specify the straight-line-segment primitive by specifying the endpoints of the line segment. This costs $2(logN + logM)$ bits. In addition, the cost of specifying the code for this primitive is $log3$. To specify the data in terms of this primitive we will, for each data point, specify a distance along the line and

the perpendicular distance from the point to the line. If the line segment is of length $l$ (in grid units) then, to specify $r$ distances, if we assume that all distances are equally likely, will cost $rlogl$ bits. If it is also assumed that the data points have a Gaussian distribution about the primitive model, the cost of specifying $r$ perpendicular distances is

$$\sum_{r\,pts} -log\left(\frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-d^2}{2\sigma^2}}\right) \quad ,$$

where $d$ is the perpendicular distance from the point to the line, and $\sigma$ the standard deviation associated with the distribution. When the above expression is expanded, the sum over the $d$'s is just the sum of the residuals squared that is calculated when the line is fitted to the data by least-squares methods.

**Arcs-of-circles:**

We specify the arcs-of-circles primitive by specifying the endpoints of the arc and one other point on the arc. This costs $3(logN + logM)$ bits, while the cost of specifying the code for this primitive is $log3$ bits. To specify the data we use the same scheme as we did for the straight-line-segment primitive.

Using these costing functions and a search algorithm that examines the various ways for partitioning a linear structure into primitives, we find the best description of that structure.

## 5. Results

The results of using the foregoing procedure on some of the linear segments found in Figures 1 and 2, (and shown in Figures 7(a) and 7(b)), are depicted in the remaining panes of Figure 7. From Figures 7(a) and 7(b) we have selected some

linear structures. The selected structures, which form the main course of the Eel river, are shown in Figures 7(c) and 7(d). Our interest is in determining whether the description built from one image is the same as that from the other. Of course, in the final version of the structure builder we would need to handle all the segments simultaneously, but this will necessitate considerable improvement in the search algorithm to keep computational costs down to a reasonable level.

Figures 7(e) and 7(f) show the primitives returned. The arc of circles are shown as full circles to improve readability. In Figures 7(g) and 7(h) the primitives have been overlaid on the data to show the quality of fit. In assessing these results, one should keep the purpose of this description in mind. We want to extract a description of the linear structure in terms of lines and curves, in terms of the manner in which parts intersect (acute angles, near-colinearity, etc.), in terms of relative curvature (tight curves, gentle curves, and the like), and in terms of the sequencing of parts in the structure. Given that the two images are viewed from very different vantage points, that the scale is quite different (not even constant in one image), that one image was taken in the infrared band and one in the visible band, that the images were taken one-and-a-half years apart during different seasons, and that no semantic information was used in the processing, the closeness of the resulting descriptions is noteworthy. This points to the usefulness of processing the data in the above manner; namely, the method of finding the linear structures; the primitives used to encode the structure; and the encoding-length measure as a criterion for best description.

Figure 7 shows the results obtained with real data. Similar results have been obtained in experiments that employ other real data sets. Justification of the

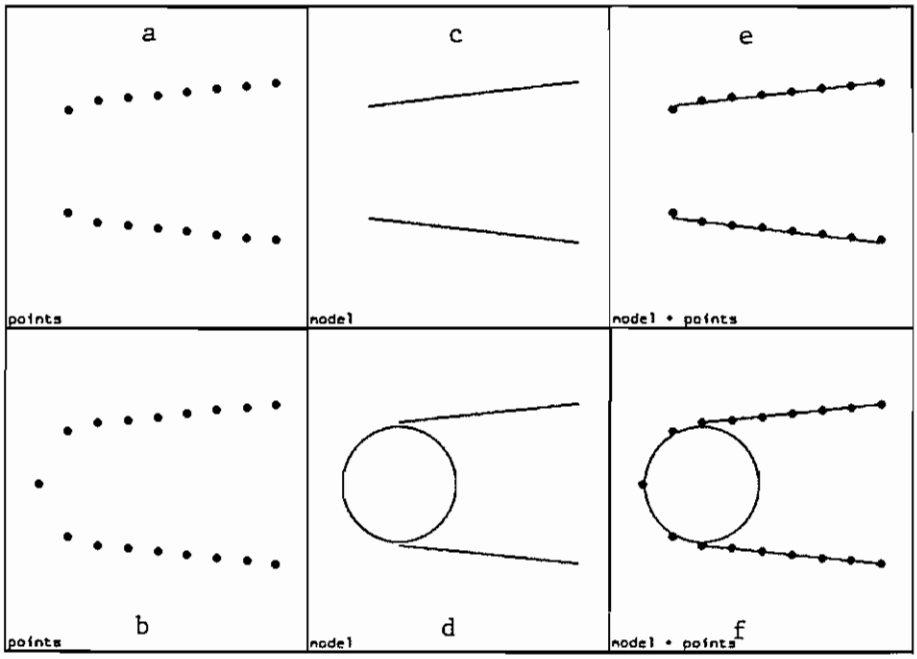


**Figure 8.** Encoding of Synthetic Data

method, however, requires further extensive experimentation. To better understand the behavior of the description builder we include an example using synthetic data. The data points are shown in Figures 8(a) and 8(b). In Figure 8(b) one extra data point has been added to those shown in Figure 8(a). The resulting descriptions are shown in Figures 8(c) and 8(d) and overlaid on the data in Figures 8(e) and 8(f). The addition of one critical point alters the description, an effect not unknown in the human visual system. The resulting descriptions seem to match those perceived by humans when they are presented with Figures 8(a) and 8(b). While we could not claim that minimal encoding is the criterion used by the human visual system for description building, we note that this criterion conforms to the type of behavior we would want to achieve if we were modeling the visual system. Of course, if the resultant description is sensitive to every addition or deletion of a data point it is of

little use. In general, the minimal-encoding-length description appears to be stable with respect to data changes, except when "critical" points are added or deleted.

## 6. Matching the Descriptions

If the description we obtain from the description builder characterizes the data and is invariant with respect to orientation, scale, and vantage point, the burden of matching descriptions is lightened considerably. It is our intent to match descriptions at the symbolic level, to represent the descriptions found by minimal encoding as graphs of symbolic entities, and to match those graphs on the basis of their structure. Of course, it is unlikely that the graphs derived from different images will match perfectly. Nevertheless, from a prospective match we can find correspondences in the original images, and calculate the camera transformation between the images.

This procedure allows data in one image to be transformed into the other. It means that we can transform a linear structure found in one image into the other image. For those parts of the graph where there is a mismatch we can ask the question: how would the linear structure that is associated with the mismatch be encoded if it were first transformed into the other image and then encoded? In this manner we can attempt to explain the graph mismatches. If we cannot explain the mismatches we should consider another match of the graphs. Through this process of hypothesis and verification, we seek to avoid acceptance of a transformation that does not explain "all" the data.

## 7. Conclusion

Having found the linear structures in an image, we are faced with two major tasks before we can use these structures to find the correspondence between different images of a scene. We need to be able to describe these structures in a way that is independent of the variations that can occur between the images, and we need to be able to match these descriptions to find the relationship between the images.

In considering structure description we show that the usual technique of dividing the structure into parts and then describing the latter can be replaced by a procedure that finds the "best" description of the data on the basis of a global view of that data. This technique simultaneously divides the structure into parts and describes them. "Best" is defined as the cheapest encoding of the data when we consider the trade-off between the quality of explanation of the data and the complexity of that explanation.

This approach produces a description of linear structures that appears relatively insensitive to the vantage point, scale, and orientation of the original images. It may prove to be a description that enables easy matching, and hence an effective approach to solving the problem of image-to-image correspondence.

## References

1. Fischler, M.A. and Wolf, H.C., Linear Delineation, *Proceedings of Computer Vision and Pattern Recognition Conference*, Washington, D.C. 1983, pp 351-356.

2. Georgeff, M.P. and Wallace, C.S., A General Selection Criterion for Inductive

Inference, in: O'Shea, T. (Ed.), *Proceedings of Advances in Artificial Intelligence, Pisa, Italy, September* 1984, North-Holland, Amsterdam, 1984.