Learning and Feature Selection in Stereo Matching

Michael S. Lew, Thomas S. Huang, *Fellow, IEEE*, and Kam Wong

***Abstract*—We present a novel stereo matching algorithm which integrates learning, feature selection, and surface reconstruction. First, a new instance based learning (IBL) algorithm is used to generate an approximation to the optimal feature set for matching. In addition, the importance of two separate kinds of knowledge, image dependent knowledge and image independent knowledge, is discussed. Second, we develop an adaptive method for refining the feature set. This adaptive method analyzes the feature error to locate areas of the image that would lead to false matches. Then these areas are used to guide the search through feature space towards maximizing the class separation distance between the correct match and the false matches. Third, we introduce a self-diagnostic method for determining when apriori knowledge is necessary for finding the correct match. If the *a priori* knowledge is necessary then we use a surface reconstruction model to discriminate between match possibilities. Our algorithm is comprehensively tested against fixed feature set algorithms and against a traditional pyramid algorithm. Finally, we present and discuss extensive empirical results of our algorithm based on a large set of real images.**

***Index Terms*— Learning, optimal feature selection, matching, stereo, range finding, automated terrain modeling, error analysis, self-diagnosis and image dependent knowledge.**

I. Introduction

STEREO vision has been an important problem in computer vision for decades, but it is still an unsolved problem due to its complexity. It is a process that transforms the information of two photographic images to a three dimensional description of the world. With this three dimensional world description, we can create models of terrain and other natural environments for use in flight simulation, virtual reality, and human-computer interactions. Stereo vision is essentially a range finding process which currently has two major advantages over other methods such as laser range finding. First, it is a passive method, which means that it does not alter the environment. Second, it is potentially better for high resolution three dimensional descriptions of moving objects because a camera can take a high resolution image in less than a thousandth of a second, whereas the current laser range finding technology usually requires objects to be relatively motionless.

Manuscript received August 1, 1993; revised January 18, 1994. This work was conducted as part of University of Illinois Advanced Construction Technology Center research program sponsored by the U.S. Army Research Office under the DoD-University Research Initiative Program, and National Science Foundation Grant IRI-89-02728. Recommended for acceptance by Associate Editor B. Bhanu.

M. Lew is with the Beckman Institute, 405 N. Mathews, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA; e-mail: m-lew@uiuc.edu.

T. Huang is with the Deparment of Electrical and computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA.

K. Wong is with the Department of Civil Engineering, University of Illinois at Urbana-Champain, Urbana, IL 61801 USA.

IEEE Log Number 9403769.

Since this paper integrates work from multiple fields, we will be using additional terminology which is defined next. L and R are two intensity images of overlapping content. The axes of L and R are denoted as (x_L, y_L) and (x_R, y_R), respectively. Specific points in (x_L, y_L) are (x_p, y_p) and (x_l, y_l). Similarly, points in (x_R, y_R) are (x_c, y_c) and (x_r, y_r). The term *correspondence* denotes a list of two points (e.g., (x_l, y_l) and (x_r, y_r)). It is assumed that the number of feature classes (e.g., intensity, Laplacian, etc.) given is n. *Feature Error* refers to the absolute difference between the value of the feature class at (x_p, y_p) and either (x_l, y_l) or (x_r, y_r). *Feature set* refers to a set whose members are feature classes. *Feature vector* refers to a vector of size n composed of the values of every feature class at a specific point.

The matching problem can be defined as follows: Given a template point, (x_p, y_p), we attempt to find a correspondence which minimizes a measure of error between the template point and the matching point. In stereo matching, our goal is to find correspondences between two intensity images of roughly the same content. Given the knowledge of the camera calibration and the correspondence (x_l, y_l) to (x_r, y_r), we can then reconstruct the 3-D coordinates of the object in the world as described next.

A. Stereo Setup

We will be discussing image matching with respect to the *normal* stereo camera configuration as shown in Figure 1, which generates normal images.

In this diagram, we show how a point on one object is projected onto the image planes of the left and right imaging stations as (x_l, y_l) and (x_r, y_r). This pair of points is referred to as a *conjugate pair* or as a correspondence. If we are given a correspondence, (x_l, y_l) matches to (x_r, y_r), then the object space coordinates can be found from (1), (2), and (3):

$$X = b(x_l + x_r)/(2(x_l - x_r)), \tag{1}$$

$$Y = b(y_l + y_r)/(2(x_l - x_r)), \tag{2}$$

$$Z = bf/(x_l - x_r). \tag{3}$$

Note that the term $(x_l - x_r)$ is often referred to as disparity, or d. Furthermore, in this configuration, the match for (x_l, y_l) must be found on $y_r = y_l$. This is a special case of the epipolar line constraint, which is important in that it reduces the search area from the entire image to one horizontal line across the image.

Does the normal camera configuration incur a loss of generality with regard to arbitrary camera configurations? Mathematically, it does not because we can generate the normal images from arbitrarily posed images through the

TABLE I
THE FEATURE CLASSES USED IN THE LANDMARK STEREO MATCHING ALGORITHM

Feature Class	Description	Mathematical Form
Intensity	the gray shade of the pixel at (x,y)	$I = h(x,y)$ where h is the discrete image and I is the intensity
X Gradient (x derivative)	the first derivative of intensity with respect to x	$G_x = \frac{\partial I}{\partial x}$
Y Gradient (y derivative)	the first derivative of intensity with respect to y	$G_y = \frac{\partial I}{\partial y}$
Gradient Magnitude	the magnitude of the x and y gradients	$G_M = \left(G_x^2 + G_y^2\right)^{\frac{1}{2}}$
Gradient Orientation	the angle made by the x and y gradients	$Go = \tan^{-1}\left(\frac{G_y}{G_x}\right)$
Laplacian	the Laplacian of intensity	$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$
Curvature	the curvature at the pixel s is the path along G_o	$Co = \frac{\partial Go}{\partial s}$

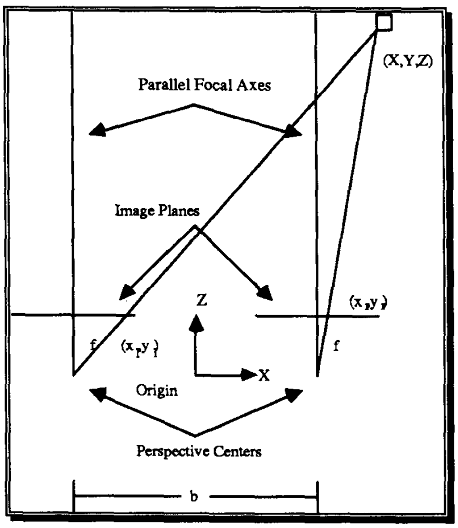


Fig. 1. The normal stereo camera configuration. The focal length is f. The baseline between the perspective centers is b. The origin of the object space coordinates is located midway between the perspective centers in a left-handed coordinate system.

process of rectification assuming that we are given the interior and exterior camera orientation [10]. Furthermore, we assume that the lens distortion model is sufficiently accurate to allow us to use the epipolar line constraint [10].

B. Algorithm Overview

This paper presents a new multifeature stereo matching algorithm which uses points and the attributes found from a window around the point for matching as shown in Table I. Our reasons for using points derives from three sources. First, there has been significant past research performed on using points as described below. Second, points do not require a special pre-processing step. Third, many structured primitives are not prevalent in all images.

For instance, if we chose line segments instead of points, we would need to choose one method of finding the line segments, and more importantly, we would be limiting our algorithm to classes of images where line segments are prevalent.

Although we treat the special case of points as matching primitives, we argue that the methods introduced in this paper can be easily generalized to most primitives (i.e., lines, curves, and segmented areas). Specifically, the chosen primitive will usually have quantifiable attributes, which can be used in place of the feature classes from Table I. Subsequently, the learning and feature selection modules of Section III can be applied directly to the new primitives and associated attributes.

Just as the word *landmark* defines a unique location, we call our algorithm the Landmark stereo matching algorithm because the central idea of the method is to find a feature set for (x_p, y_p) that will uniquely define the point. There are essentially three steps in our method which are shown in Fig. 2. The first step produces an approximation of the optimal feature set. The second step refines the feature set toward the optimal feature set in the sense of making the selected point (x_p, y_p) unique. The third step treats the case in which the feature set is ambiguous.

An important general question is whether an adaptive method for feature selection, which selects the best subset of

Landmark stereo matching algorithm

(1) **Find an initial feature set**
Method: apply a concept learning algorithm
Input: values of the feature classes at (x_p, y_p)
Output: a subset of the available feature classes (ie. {intensity, gradient magnitude})

(2) **Improve the feature set**
Method: find the feature set which maximizes matching accuracy
Input: a feature set (ie. {intensity, gradient magnitude})
Output: a feature set (ie. {gradient magnitude, Laplacian of intensity})

(3) **If the feature set is unambiguous then use the feature set for matching. Else apply apriori knowledge to resolve the matching ambiguity.**
Method: apriori knowledge is to fit to a thin metal surface model.
Input: a feature set
Output: (i) a correspondence, $(x_p, y_p) \to (x_c, y_c)$
(ii) incremental update of concept in learning algorithm

Fig. 2. The Landmark stereo matching algorithm.

the available feature classes, is warranted for a small number of feature classes. Two alternatives are 1) Use the single best feature class (Single Best algorithm), and 2) Use all the feature classes (Sum All—sum all the feature errors into one.). These alternatives will be addressed from both qualitative and quantitative perspectives in Section IV. Specifically, we will quantitatively show that the alternatives usually yield worse results, and we will qualitatively describe conditions when the alternatives are not preferable.

C. Background

The history of stereo research has provided a rich and extensive background for the ideas in our research. The progression of stereo research appears to be towards utilizing more feature classes of varying levels of abstraction. With these facts in perspective, we look at the previous work in stereo image matching. Moravec [18] found interesting points in the left image and used the binary-search method of an image pyramid to find the correspondences. Hannah [9] made improvements to his method but kept the unidirectional coarse-to-fine search method. Marr and Poggio [16] and Grimson [6] used zero-crossings of the Laplacian of the Gaussian at different spacings as matching primitives. They found matches at a particular initial level, enforced continuity of zero-crossings, and then approximated the match down the pyramid from coarse-to-fine. Hoff and Ahuja [11] used zero-crossings to integrate surface modeling and stereo matching at a particular initial level and strictly approximated toward the finest level. Surfaces were modeled as planar and quadratic patches. Lim and Binford [13] applied a hierarchical structure based on different scale feature classes, specifically, bodies, surfaces, junctions, curves, and edges. Barnard [2] chose an annealing approach for finding global optima from matching all points simultaneously. The match error was an energy function combining intensity difference and local changes in disparity. Cohen, Vinet, and Sander [3] used an edge hierarchy to integrate segmentation with stereo matching.

The most recent work includes utilizing a variety of wave forms as primitives [17]. Marapane and Trivedi [14] applied multiple primitives in a hierarchy. In addition, Weng, Ahuja, and Huang [23] used edgeness, positive, and negative cornerness in a hierarchical based matcher.

The organization of this paper is as follows. Section II briefly reviews classical optimal feature selection. This motivates us toward adaptive feature selection, which is the core of the Landmark Stereo Matching algorithm. Section III describes the learning step, the adaptive feature selection step, and the matching step of our algorithm. Section IV discusses the results of comparing the adaptive method to the alternate methods (Best Single and Sum All), and of comparing the Landmark algorithm to a conventional pyramidal algorithm. Section V summarizes the conclusions and contributions.

II. Feature Selection in Stereo Matching

A. Classical Feature Selection

This section gives a brief review of classical feature selection as described by Devijver and Kittler [5]. The problem of feature selection lies in selecting the best subset U of d feature classes:
Select $d \leq D$ from

$$V = \{v_j \mid j = 1, 2, \cdots, D\} \tag{4}$$

arriving at

$$U = \{u_i \mid i = 1, 2, \cdots, d\} \tag{5}$$

where each u_i is an element of V, and U optimizes a criterion function $J(U)$.
The number of combinations of d out of D feature classes is given by

$$q = \binom{D}{d} = \frac{D!}{(D-d)!d!}. \tag{6}$$

The significance of q is to show the computational power required to search through all of the feature sets. For instance, selecting ten feature classes out of a hundred would necessitate evaluation of more than 10^{13} feature sets. Thus, in practical situations computationally feasible methods must be employed. Typical search methods are shown in Table II.

Note that all of the search methods except for Branch and Bound are suboptimal. Only Branch and Bound implicitly searches all of the combinations and guarantees a globally optimum feature set.

Intuitively, classifying patterns requires an assumption that classes occupy distinct regions in the pattern space. When the classes are more distant, the probability of successful recognition of class membership increases. Thus, the general approach is to select the d-dimensional feature subspace which maximally separates the classes.

Define the distance between d-dimensional feature vectors f_{ik}, f_{jl}, from classes w_i and w_j respectively by $\delta(f_{ik}, f_{jl})$, the class probabilities as P_i, P_j, the number of training patterns from class w_i in set S_n as n_i, then the optimality criterion as the average distance between elements of c classes,

$$J(f) = \frac{1}{2}\sum_{i=1}^{c} P_i \sum_{j=1}^{c} P_j \frac{1}{n_i n_j} \sum_{k=1}^{n_i}\sum_{l=1}^{n_j} \delta(f_{ik}, f_{jl}) \tag{7}$$

where

$$\delta_M(\mathbf{f}_k, \mathbf{f}_l) = \left[\sum_{j=1}^{d} \mid f_{kj} - f_{lj} \mid^s\right]^{\frac{1}{s}} \tag{8}$$

TABLE II
SEARCH METHODS FOR FINDING OPTIMAL AND SUBOPTIMAL FEATURE CLASS SETS

Branch and Bound -Narendra and Fukunaga [19]	A top-down search procedure with backtracking , which allows all of the combinations to be implicitly search without an exhaustive search.
sequential Forward Selection (SFS) -Whitney [24]	A bottom up search procedure where the feature which will yield a maximum of the criterion function is added one a a time to the NULL set.
Generalized Sequential Forward Selection (GSFS) Kittler [12]	Similar to SFS, but instead of adding one feature at a time, r features are added.
Sequential Backward Selection (SBS) -Marill and Green [15]	Starting from the complete set of features, we discard one feature until 2-D features have been deleted.
Generalized Sequential Background Selection (GSBS) -Kittler [12]	Similar to SBS, but instead of discarding one feature, multiple features are discarded.

is the Minkowski Metric of Order s. For an in depth discussion on the class separation distance and different distance metrics, consult [5]. The distance metric used for this paper is the absolute value of the difference, or the Minkowski Metric of Order 1.

Thus, selecting the optimal feature class subset is a computationally intensive problem. The following section describes the Landmark stereo matching algorithm, which integrates learning and adaptive feature selection toward minimizing computational expense.

III. LANDMARK STEREO MATCHING ALGORITHM

There are four important properties of the Landmark algorithm. First, it uses an instance based learning algorithm to find the initial feature class as opposed to evaluating every available feature error. Second, it uses left image to left image matching to perform the adaptive searching used to increase the class separation distance. Third, it identifies when the discriminatory power of the available feature classes is insufficient to determine the correct correspondence. Fourth, it integrates learning, feature selection, and surface reconstruction.

A. Learning

This section explores the problem of finding an initial feature set from step 1 of Fig. 2. Given that the combinatorial explosion from searching for an optimal feature set may be prohibitive, we explore a method of finding an initial point from which to begin the search through feature space. Computational expense can be saved by generating a first approximation of the optimal feature set using a concept learning algorithm such as a neural net or an instance based learning algorithm [1], [4]. With respect to instance based methods, we need to review some terminology. *Exemplar* refers to a list of two elements, where the first element is a feature vector, and the second element is a feature set. The *classification* of the feature vector is assumed to be the associated feature set. *Exemplar list* refers to a list of exemplars.

There are three fundamental questions with respect to the learning algorithm which will be addressed in this section. First, how does the representation affect the learning? Second, what assumptions must be made to ensure that the learning algorithm will be effective? Third, how is the effectiveness of the learning algorithm evaluated?

There are essentially two distinct ways in which the learning algorithm finds the initial feature class:

- Feature class interactions (image independent knowledge);
- feature classes are locally constant with respect to image x and y axes (image dependent knowledge).

Feature class interactions refers to relationships between the feature classes. This means that the values of the feature classes can be used to determine whether to use a particular feature class. For instance, there is a strong relationship between gradient magnitude and gradient orientation. Gradient magnitude measures the degree of change in the intensity along the image x and y axes. Gradient orientation measures the direction of the greatest change along the image x and y axes. If the gradient magnitude is zero, as it would be on a white wall, then the gradient orientation is unreliable since there is no direction of greatest change. As the gradient magnitude increases, the reliability of the gradient orientation increases. We call this *image independent knowledge* because the knowledge depends upon the interactions between feature classes and not upon the image coordinates. These exemplars will have the form

$$([f_1, f_2, f_3, f_4, f_5, f_6, f_7], \ \{\text{intensity}\})$$

where f_i refers to the value of the ith feature class at (x, y).

Local constancy of the feature classes assumes that if a point (x_1, y_1) with feature vector, V_1, can be distinguished by feature class, C_1, then a point (x_2, y_2), which is sufficiently close to (x_1, y_1), and which has a sufficiently similar feature vector, V_2, with regard to V_1, should also be distinguished by C_1. Nearby points with similar feature vectors should be able to be distinguished with the same feature class. We call this image dependent knowledge because the knowledge depends upon the image coordinates. Only spatially (with respect to image coordinates) similar images will be able to be classified

TABLE III
AN EXAMPLE OF A CONCEPT CONTAINING 4 EXEMPLARS. THE FEATURE CLASSES OF THE FEATURE VECTOR ARE [INTENSITY, MAGNITUDE, ORIENTATION]

Line	Concept C
(1)	([100, 0, 20], {intensity})
(2)	([100, 10, 20], {intensity})
(3)	([100, 30, 20], {orientation})
(4)	([100, 40, 20], {orientation})

Intensity and orientation are the only possible classifications.

Instance Based Learning Algorithm	
(1)	Initialize C to the set of first exemplar in T.
(2)	For all subsequent training exemplars t in T: Repeat steps (3) and (4)
(3)	k = Gaussian Support Classification of t by C.
(4)	If (k equals the associated classification of t) THEN add t to the discard list ELSE add t to C and check the discard list for incorrectly classified instances.
(5)	Delete redundant and noisy exemplars from C.

Fig. 3. NT2, the instance based learning algorithm. This algorithm shows the first stage of instance based learning algorithms, where the concept is created.

with image dependent knowledge. These exemplars will have the form

$$([x, y, f_1, f_2, f_3, f_4, f_5, f_6, f_7],\ \{\text{intensity}\})$$

Thus, if the feature classes are closely interrelated (image independent knowledge), and if the optimal feature classes vary slowly over the image x and y axes (image dependent knowledge), then the learning algorithm should be effective.

There are two stages in the instance based learning paradigm. First, a training set which has the form of an exemplar list is used to build a concept, C. This is called the learning stage. Second, after the training set has been fully processed, new input feature vectors are classified using C. Specifically, we first find the Euclidean distance between the input feature vector and the feature vector of each exemplar in C. Second, we classify the input vector as the feature set of the vector in C which has the minimum distance.

Advantageous characteristics of instance based learning algorithms [1] are 1) simple representations for concept descriptions, 2) low incremental learning costs, 3) small storage requirements, 4) produce concept exemplars on demand, 5) ability to learn continuous functions, and 6) ability to learn non-linearly separable categories. Instance based learning algorithms were chosen over neural networks because of the low incremental learning costs.

Consider an example where the input feature vector for (x_p, y_p) is [100, 9, 20]. The feature vector of the exemplar on line (2) of Table III is closest to the input feature vector. Then, we would classify the input feature vector as $\{\text{intensity}\}$.

The first fundamental modification of the IBL algorithm is toward using the two kinds of information which will be passed to the learning algorithm concept. The modification is to have a conditional in the comparison module which compares only the feature class information if the knowledge of the exemplar is image independent, and compares both the image coordinates and the feature class information if the knowledge in the exemplar is image dependent.

Our approach toward noise tolerance is to classify the new element using the entire concept C instead of the nearest neighbor. Specifically, we accumulate the support from each exemplar in C toward a particular classification. The classification with the largest support is then chosen. The Gaussian weighted support function is chosen as

$$S(k) = \sum_{l \in C, \mathbf{k}} e^{-\frac{1}{2}\left(\frac{\delta(\mathbf{f}_1, \mathbf{f}_r)}{\sigma}\right)^2} \tag{9}$$

where $\delta(\mathbf{f}_1, \mathbf{f}_r)$ is the metric between the exemplars in C which are in class, k, and the new instance, $\mathbf{f}_T$. Then the classification for $\mathbf{f}_T$ would satisfy

$$\max(S(k)), k = 1 \cdots c \tag{10}$$

where c is the number of classes. This will result in a few incorrect points being suppressed by the vote of the many correct points as the Gaussian weighting will give greater support to nearby exemplars and less support from farther exemplars in feature space. Furthermore, in practice we can discard distant exemplars from the voting process since their weighted vote will be negligible.

Another issue in designing instance based learning algorithms is minimizing the size of the concept C. Note that in Table III, the classification for any new input feature vector will not change if we eliminate the exemplars on lines 1 and 4. In general, we can reduce the size of C by grouping exemplars, which are close in feature space, into a single exemplar. Furthermore, if the feature vector of an exemplar in C has a different Gaussian support classification than its associated classification, then we can delete the exemplar from C.

The instance based learning algorithm is trained using preclassified feature vectors. These preclassified feature vectors are found both manually and automatically. The automatic classification is performed by determining which feature errors for a prematched stereo pair will yield the correct correspondence and a sufficiently large class separation distance.

The instance based learning algorithm which is used for the Landmark algorithm is based upon Growth NT [1] with some modifications toward integrating two different kinds of knowledge in the concept, improving training order independence (creating the discard list), noise tolerance, and concept size minimization. Given that T denotes the training set, NT2 is shown in Fig. 3.

Evaluating the performance of the learning module is a difficult question. We have chosen to compare the Landmark algorithm with the learning module, and the Landmark algorithm without the learning module. When we do not utilize the learning module, we evaluate the class separation function for every feature error, and initialize the feature set with the feature class which has the largest class separation distance.

B. Adaptive Feature Selection

This section explores the problem of improving the current feature set in step 2 of Fig. 2. In order to optimize the feature set, we need a function to maximize. We shall use the concept of the class separation distance in formulating the optimality criterion. Consider that each feature set will result in a specific error function with respect to (x_p, y_p). The optimal feature set

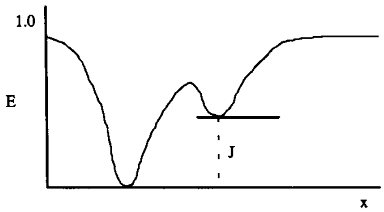


Fig. 4. A plot of feature error and the image x axis. One definition of the class separation distance, J, is the difference between the second smallest minimum and the smallest minimum of the feature error.

should have the attribute that its minimum is at the correct (x_c, y_c). The ideal error function has one minimum at the correct (x_r, y_r), whereas the worst case error function is a flat plane, which would be the most ambiguous situation.

In order to compare different feature errors, we need to normalize each feature error to the range $[0 \cdots 1]$. The simplest definition of the class separation distance, J, is the difference between the second smallest minimum of the feature error and the smallest minimum as shown in Fig. 4, or if the second smallest minimum does not exist, then J is equal to 1.

By defining the class separation distance in terms of the error function, we only need to consider the minima instead of every point in the image. This allows us to significantly reduce computational expense. Let us consider the case of multiple minima in the error function. Each minimum is associated with a different stereo correspondence, where only one correspondence is correct. The other minima are called sources of mismatches since they lead to incorrect correspondences.

With respect to stereo matching, we want to maximize the distance between the value of the error function at the correct correspondence and the value of the error function at all of the sources of mismatches. Thus, we define a measure which assigns a larger class separation distance to greater differences between minima in the error function, that is, when the difference between the first and second minima is large, the class separation distance should be large. But we would also like the class separation distance to gracefully diminish with additional minima which are in the spatial neighborhood of the global minimum. The Gaussian function was chosen because of these properties.

Recalling that n is the total number of available feature classes, then the error function for a feature set is

$$\text{Error} = \mathbf{w} \cdot \mathbf{F} \tag{11}$$

where

$$\mathbf{w} = [w_1, w_2 \cdots w_n] \tag{12}$$

with

$$\sum_{i=1}^{n} w_i = 1 \text{ and } w_i \geq 0 \text{ for } i = 1, 2, \cdots, n \tag{13}$$

and

$$\mathbf{F} = [f_1, f_2 \cdots f_n] \tag{14}$$

where f_n is the nth feature error with respect to (x_p, y_p).

Then if we choose Gaussian weighting, the class separation distance becomes

$$J(\mathbf{w} \cdot \mathbf{F}) = 1 - \frac{1}{N_v} \sum_{\mathbf{x} \in M} e^{-\frac{1}{2}\left(\frac{\mathbf{w} \cdot \mathbf{F}(\mathbf{x}_G) - \mathbf{w} \cdot \mathbf{F}(\mathbf{x})}{\sigma}\right)^2} \tag{15}$$

where $\mathbf{x}_G$ is the global minimum of the error function, N_ν is the total number of minima, and M is the list of all x such that x is a local minimum but not $\mathbf{x}_G$. The distinction between N_ν and M is that M is a list of the minima excluding the global minimum. Thus the class separation, J varies between 0 and 1, the minimum and maximum class separation distances, respectively.

In image space, sufficient conditions for a local minimum are

$$(\mathbf{w} \cdot \mathbf{F}(\mathbf{x}))_x = 0 \text{ and } (\mathbf{w} \cdot \mathbf{F}(\mathbf{x}))_y = 0 \tag{16}$$

and

$$\begin{vmatrix} (\mathbf{w} \cdot \mathbf{F}(\mathbf{x}))_{xx} & (\mathbf{w} \cdot \mathbf{F}(\mathbf{x}))_{xy} \\ \mathbf{w} \cdot \mathbf{F}(\mathbf{x}))_{yx} & (\mathbf{w} \cdot \mathbf{F}(\mathbf{x}))_{yy} \end{vmatrix} > 0. \tag{17}$$

Now, the goal is to find $\mathbf{w}$ such that J is maximized, or

$$J(X) = \max_{\mathbf{w}} J(\mathbf{w} \cdot \mathbf{F}). \tag{18}$$

For suboptimal search we could stop searching at a sufficiently large class separation. Henceforth, this will be called a *distinct feature set* as opposed to an optimal feature set. Nevertheless, it is possible that there is no $\mathbf{w}$ which results in a single distinct minimum. This case is explored in Section III-C.

A straightforward method of feature selection is to maximize $J(\cdot)$ between (x_p, y_p) and R. This will result in obtaining the most distinct error function from the set of feature classes. If we were to use this approach, we would also use one of the traditional feature selection search methods: Branch and Bound [19] for the optimal feature set, or one of many feature selection algorithms [24], [12], [15] for a less computationally expensive but suboptimal set. We have chosen to pursue another interesting possibility.

Instead of maximizing $J(\cdot)$ between (x_p, y_p) and R, we maximize $J(\cdot)$ between (x_p, y_p) and L. This possibility has the following significant advantages: 1) we would know which minimum in the error function corresponds to (x_p, y_p) and which minima correspond to sources of mismatches; 2) if we compute the error only at the minima for the features classes not in the feature set, then we could guide the addition of feature classes to the feature set by adding the feature class which has the greatest total error at the minima.

Thus, we prune the search space by considering only those feature classes which will increase the sum of the secondary minima. This heuristic requires evaluating the feature class only at the points corresponding to the minima. When a likely candidate is found, we compute J of the expanded feature set. If the J of the expanded feature set is greater than the J of the old feature set, we permanently add the candidate to the feature set. If not, we discard the candidate feature class and try another.

The disadvantage is that once the feature set is selected, we will have to search through R for the global minimum,

Feature Set Improvement Algorithm

(1) $\mathbf{F_i}$ = Feature set approximation from the IBL algorithm with respect to (x_p, y_p)
(1.1) $J_{old} = J(\mathbf{F_i})$ = Initial class separation
(2) If the feature set is distinct ($J(\mathbf{F_i}) > J_t$) then go to **Stereo Matching Algorithm** (in Section III. C. or step 3 of Figure 4.)
(3) $\mathbf{M_t}$ = minima in $\mathbf{F_i}$ applied to L and (x_p, y_p)
(4) Apply $\mathbf{M_t}$ to $\mathbf{F_c}$.
(5) Let *f* be the element in $\mathbf{F_c}$ which has the maximum total error over $\mathbf{M_t}$,
(5.1) If there are no features errors left (*f*=NULL) then go to **Stereo Matching Algorithm**
(5.2) Add *f* to $\mathbf{F_i}$.
(5.3) Delete *f* from $\mathbf{F_c}$
(5.4) if the current feature set is distinct ($J(\mathbf{F_i}) > J_t$) then go to **Stereo Matching Algorithm**
(5.5) if $J(\mathbf{F_i}) > J_{old}$ then $J_{old} = J(\mathbf{F_i})$; go to 5
(5.6) Delete *f* from $\mathbf{F_i}$. Go to 5

Fig. 5. The feature set improvement algorithm.

which in the straightforward method would already have been performed.

Both methods share the important advantage of being able to determine when the feature set is insufficient for matching (x_p, y_p). This situation occurs when the class separation distance is lower than a threshold, J_t. Although many stereo matchers will reject a correspondence if the final feature error is too large, it is rare for a stereo matching algorithm to be able to determine if there are too many points with small feature errors (i.e., when $J < J_t$), which is considered by our algorithm in the next section. Let $\mathbf{F_c}$ = list of n feature errors. The algorithm is shown in Fig. 5.

In summary, the central idea of the guided or adaptive method of finding distinct feature sets is to record the points of minima of the current feature set with respect to (x_p, y_p). Thus, we perform an informed addition of feature classes to the feature set in the direction in feature space which maximally increases the class separation distance.

C. Stereo Matching Algorithm

After we have refined the feature set toward maximizing the class separation distance, we have two possibilities.

Case 1: The feature set $\mathbf{F_i}$ is distinct.

Case 2: The feature set $\mathbf{F_i}$ is not distinct.

If Case 1 is true, then the feature set is able to discriminate between the correct match and the sources of mismatches. Consequently, we apply the feature set to the corresponding epipolar line of R, and determine the correspondence as the point of minimum error.

In Case 2, the discriminatory ability of our feature set is insufficient to properly distinguish between the possible matches in L. This implies that the feature set will also be insufficient to discriminate between the possible matches in R. There are two options in this situation. We could reject the point, or apply a heuristic to select one of the minima. Thus, the solution will depend upon the particular application to which the feature selection is being applied. In the application of stereo matching, we chose to apply a heuristic in the form of the quadratic variation assumption.

We chose to decide between match possibilities by fitting the previous matches and the current match to the quadratic variation, E [6], [21], [22]:

$$E = \int\int \left(\frac{\partial^2 u}{\partial x^2}\right)^2 + \left(\frac{\partial^2 u}{\partial y^2}\right)^2 dx dy. \tag{19}$$

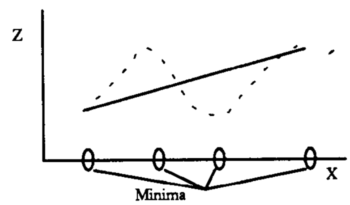


Fig. 6. Given the possibilities of a flat surface or a jagged surface, the quadratic variation will bias the surface toward the flat surface.

The surface reconstruction method of Harris [8] was chosen because it could potentially be implemented in hardware, and it can incorporate slope information. The quadratic variation including slope information is

$$E = \int\int [(u_x - p)^2 + (u_y - q)^2 + p_x^2 + p_y^2 + q_x^2 + q_y^2]\, dx dy. \tag{20}$$

If we consider (x_p, y_p) and the sources of mismatches as a set of points in L, and then map them to the minima in R, the interpolated depth can be used to compute the quadratic variation. Thus, we assign the correspondence of (x_p, y_p) as the correspondence that satisfies

$$\min\{\mathbf{w} \cdot \mathbf{F} + E\} \tag{21}$$

over (x_p, y_p), the sources of mismatches in L, and the minima in R.

This biases the possible surfaces towards flat, planar surfaces as shown in Fig. 6.

Furthermore, if we have access to a dense surface map generated from previous correspondences between L and R, then we can also use the surface map to fit (x_p, y_p) directly.

One disadvantage of methods which incorporate surface model assumptions throughout the matching algorithm is that the matching algorithm will fail on surfaces which violate the

Stereo Matching Algorithm

(1)	If the feature set, $\mathbf{F_i}$ is distinct, then go to 2 Else go to 3
(2)	Apply $\mathbf{F_i}$ to R and (x_p, y_p) to find the correspondence, (x_c, y_c)
(2.1)	Update surface reconstruction map z=s(x,y)
(2.2)	If the new value for z agrees with the interpolated value, then update the learning algorithm concept, *C*.
(2.3)	STOP
(3)	If the reconstructed surface map is dense then set (x_c, y_c) to the minimum which agrees closest with the surface reconstruction map. If the map is not dense then select the correspondence which minimizes the normalized sum of the feature error and the quadratic variation, *E* over (x_p, y_p) and the sources of mismatches..
(3.1)	Update surface reconstruction map.
(3.2)	STOP

Fig. 7. The stereo matching algorithm.

surface model assumptions. For instance, a typical problem of methods which use the quadratic variation assumption is that they cannot reconstruct jagged surfaces. Our method only uses the quadratic variation assumption when the discriminatory power of the feature set is insufficient. If the feature set can discriminate between the match possibilities then the quadratic variation assumption is never invoked. Thus, our algorithm only makes apriori assumptions when there is no other alternative. The stereo matching algorithm which incorporates the two cases is shown in Fig. 7.

After we have updated the surface map, we can consider the conditions when we wish to pass a new exemplar to the learning algorithm concept in the form of

$$([x_p, y_p, f_1, f_2, f_3, f_4, f_5, f_6, f_7], \ \{\text{feature class}\})$$

where f_i refers to the value of the i^{th} feature class at (x_p, y_p), and "{feature class}" refers to the feature class which has a maximum class separation distance from the feature set to the learning algorithm concept. Since there will be incorrect matches, we would prefer only to pass exemplars which give additional confidence of the correctness of using the feature set. In the surface reconstruction module, a correspondence becomes a point in 3-D coordinates, $(X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}})$. From the surface map and from X_{new} and Y_{new}, we can generate an interpolated value for Z called Z_{map}. If Z_{new} is sufficiently close to Z_{map}, then we pass a new exemplar to the learning algorithm. Note that if the feature set was insufficient to distinguish (x_p, y_p), then we do not pass a new exemplar to the learning algorithm concept.

D. Integration

This section addresses the integration of the learning module, the feature selection module, and the surface reconstruction module. The learning module uses feature class interrelationships to find the initial feature class for the feature selection module. The feature selection module refines the feature set toward the largest class separation distance. The surface reconstruction module uses the feature set to find the match, and refine its surface map. The surface map is used to incrementally update the concept in the learning algorithm by supplying new instances with spatial locations and their

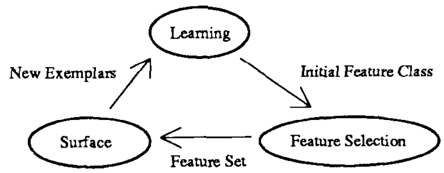


Fig. 8. The integration of learning, feature selection, and surface reconstruction.

corresponding feature vectors. These relationships and the integrated feedback structure are shown in Fig. 8.

By integrating these modules, the concept of the learning algorithm is enhanced so that a distinct feature set can be achieved with minimal searching. The learning algorithm uses the surface map to determine whether to add the new exemplar to the concept. Without the surface map information we would not be able to verify the correctness of the exemplar. Furthermore, when the feature set is not distinct, the match is determined by using both the error from the feature set and the quadratic variation. This integration of the feature selection and surface reconstruction modules results in a more accurate matching than using either separately.

IV. Results

In this section we present two important comparisons. First, we quantitatively show that the adaptive learning method is more accurate than the alternate methods. Second, we compare the Landmark algorithm with a conventional pyramidal stereo matching algorithm. We present the comparative matching accuracies, and then we show the test images along with other visual representations of the matches found from the Landmark algorithm. Note that all of the algorithms were run on an IBM-AT/486 at 50 Mhz.

A. Comparison of Alternate Methods with Adaptive Method

There are two alternate methods which we will consider. These alternatives are 1) Sum All which means summing all the feature errors, and 2) Best Single which means using the single feature error which has the maximum class separation distance. Note that since the alternate methods do not use

TABLE IV
THE MATCHING ACCURACIES BETWEEN THE ALTERNATE METHODS AND THE ADAPTIVE METHOD

	Poster	Person	Street	Wall	Robots	Face
Sum All	71	68	35	61	69	81
Best Single	81	71	62	69	80	81
Adaptive	92	85	60	74	90	92

TABLE V
THE AVERAGE TIME REQUIRED TO MATCH ONE POINT

	Time/point
Sum All	0.998
Best Single	1.029
Adaptive	0.366

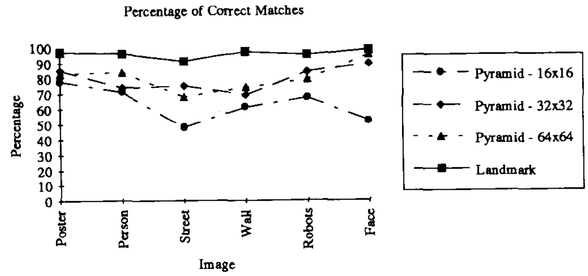


Fig. 10. The percentage of correctly matched points over a variety of images between a pyramidal matcher with starting resolutions at 16 × 16, 32 × 32, and 64 × 64, and the landmark algorithm.

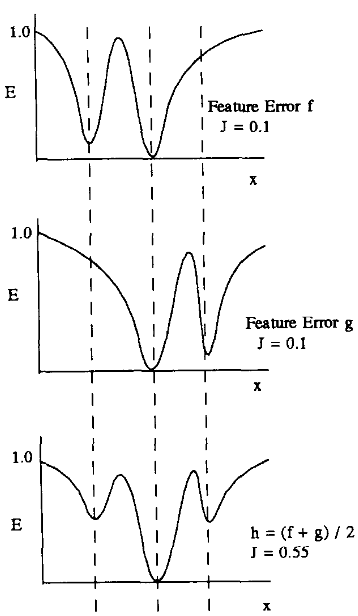


Fig. 9. This diagram depicts how the class separation function, J, increases due to averaging two feature errors which have secondary minima at different locations.

a surface model, we did not use the surface reconstruction module in the matching process for the results of the adaptive method in this section.

Next, Table V displays the average time over all the images required to match a single point. The adaptive method is almost three times faster than the Sum All and the Best Single methods.

Sum All is the most common way of using multiple feature classes. In this method, the match that corresponds to the global minimum of the error function defined by averaging all the feature errors together is taken as the correct match. In the Best Single method, the class separation distance is calculated for each feature error. The feature error which has the largest class separation distance is used for the matching. Specifically, the global minimum of the selected feature error is used as the correspondence.

Can we justify using an adaptive method for the matching? With only seven feature classes in our implementation one might argue that averaging all the feature classes would be simpler and equally accurate. In general, the answer depends upon the exact choices of feature classes. We argue that the adaptive method is a general method designed for a large number of feature classes, which could be generated by simply taking the seven feature classes in our implementation and applying each to a different scale image. Then, for four scaled images in an image pyramid, we could have 28 relevant feature classes.

Table IV displays the matching accuracies of the alternate methods and the adaptive method. Note that the adaptive method is on average 18 percent more accurate than the Sum All method. Also, the adaptive method is on average 8 percent more accurate than the Best Single method.

From this data we can assert that in at least some images, the use of the adaptive method will result in greater matching accuracy. Furthermore, the adaptive method is significantly faster than either the Sum All or the Best Single methods.

Why is the adaptive method more accurate than the alternate methods? The reason why it is usually more accurate than the Best Single method is that the class separation function can often be increased by adding other feature classes to the set. Fig. 9 shows that if the secondary minimums occur at different locations, then the class separation function can increase.

One reason why the adaptive method is more accurate than the Sum All method is that the class separation distance can be decreased by adding the wrong feature classes to the set. Suppose we have one feature error with J equal to 0.5, and three feature errors with J equal to 0.1. If we average the four feature errors, we will find that J is 0.2, which is a significantly worse class separation distance. In the extreme case of many feature errors, suppose we have one ideal feature error, and n worst case feature errors, then

$$E = \frac{1}{n+1} f_{\text{ideal}} + \frac{n}{n+1} f_{\text{worst}} \tag{22}$$

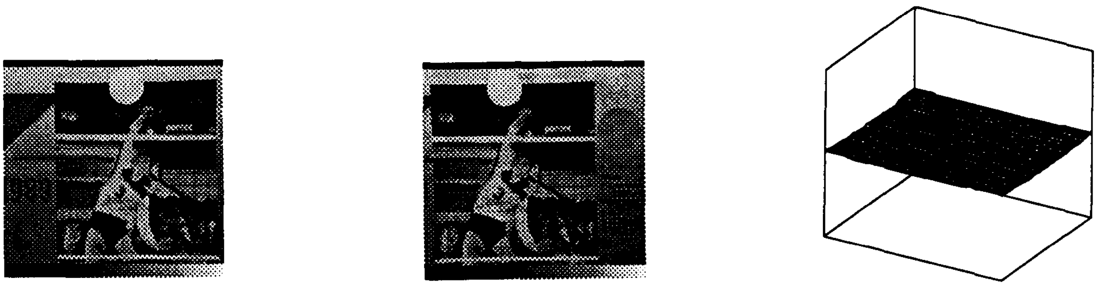
and as n approaches infinity,

$$E = f_{\text{worst}}. \tag{23}$$

Furthermore, some feature classes will give unreliable responses depending upon the image content. One example of this is gradient orientation. If the gradient magnitude is very small, then the gradient orientation is unreliable. Specifically, the orientation will be random depending upon the noise in the imaging process. This means that if we always use gradient

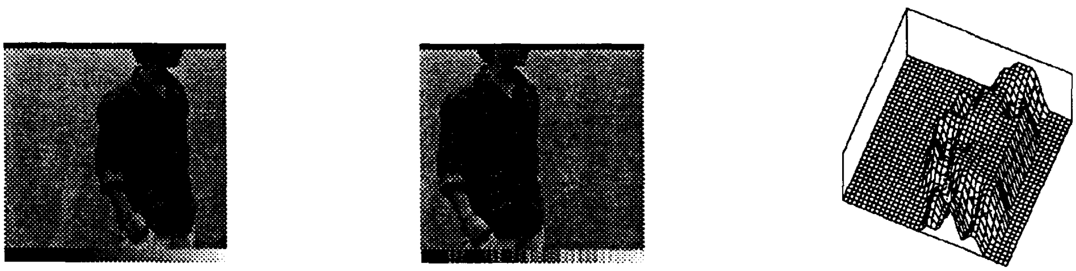
TABLE VI
THE AVERAGE PERCENTAGE OF FEATURE CLASS USAGE

	Intensity	x Deriv	y Deriv	Magnitude	Orientation	Laplacian	Curvature
% Used	62.6	23.8	3.3	0.2	8.6	1.4	0.0



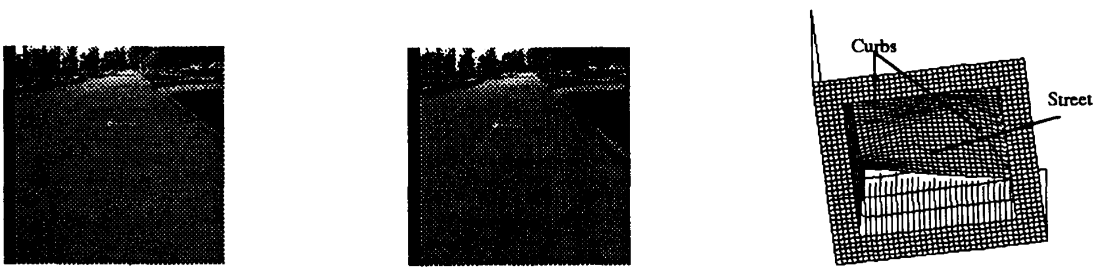
(a) (b) (c)

Fig. 11. Volleyball poster stereo pair and reconstructed susrface.



(a) (b) (c)

Fig. 12. Person stereo pair and reconstructed surface.



(a) (b) (c)

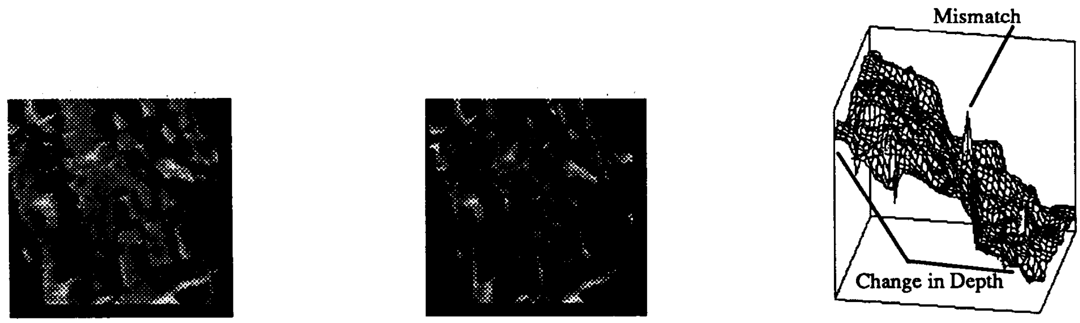
Fig. 13. Street stereo pair and reconstructed surface.

orientation regardless of the image content, the orientation at (x_p, y_p) will be wrong, and it will generate false minima at locations of low gradient magnitude. Thus, the Sum All method has the additional problem of always using feature classes which should only be used in certain situations.

Overall, if the number of feature classes is small, it is still preferable to maximize the class separation distance in order to find the best subset of feature classes. Furthermore, maximizing the class separation distance is a generalization of Sum All and Best Single in the sense that it will use all, some, or only one of the feature classes as needed.

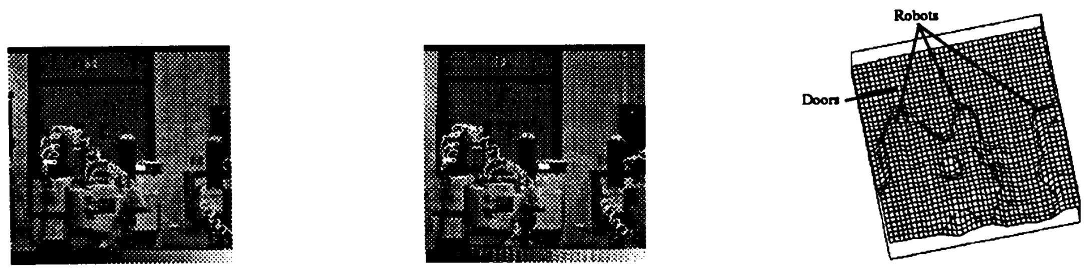
B. Comparison of Landmark Algorithm and Pyramidal Algorithm

The Landmark Algorithm and a pyramidal algorithm were tested on stereo pairs of real images. The left to left image matching was chosen in order to implement the adaptive class separation distance maximization. The feature errors were normalized between $[0 \cdots 1]$. The patch sizes for the feature errors were 5×5 windows. Backmatching is used for occlusion. The pyramidal algorithm is presented as a benchmark. The data structure for the pyramidal algorithm is



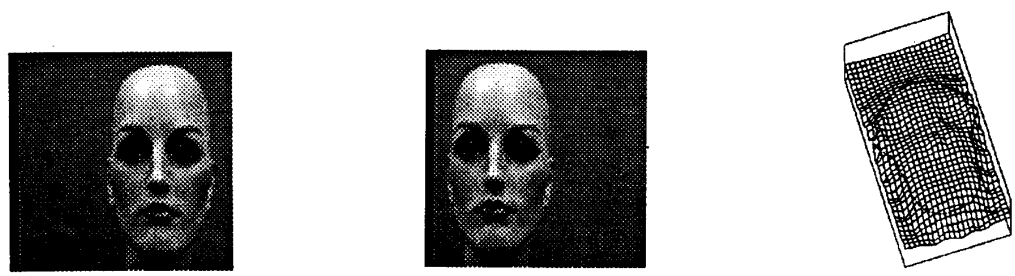
(a) (b) (c)

Fig. 14. Rock wall stereo pair and reconstructed surface.



(a) (b) (c)

Fig. 15. Robots stereo pair and reconstructed surface.



(a) (b) (c)

Fig. 16. The face stereo pair and reconstructed surface.

a Gaussian intensity pyramid. Linear search was performed at the starting resolution level, and then hill climbing through the pyramid image structure was used to implement the refinement to the finest resolution. The matching feature class was intensity and the metric was the normalized correlation coefficient.

The 256 $\times$ 256 images include a poster of a volleyball player, a person standing in front of a plain background, an outdoor scene of the street outside the Beckman Institute, a rock wall image from the Stuttgart standardized image set [7], robot arms against a complex background, and a face. The size of the ambiguous sets, which are a list of the template point and the sources of mismatches, varied from 2 for the face and person images to 7 for the street image. Table VI displays the percentage of the feature class usage averaged over all the images. The matching accuracy results are shown in Fig. 10. Over all the images, the most dominant feature classes in order of importance are intensity, x derivative of intensity, and gradient orientation.

Gradient magnitude, Laplacian, and Curvature were rarely used. This probably occurred because gradient magnitude and Laplacian are directionally independent, which means that more points would appear to be similar. For instance, if we consider an image of a white poster on a black wall, the gradient magnitude and Laplacian will essentially be the same between the left, right, top, and bottom edges of the white poster. This will cause the class separation distance to be near zero. Curvature was rarely used probably because a 5 $\times$ 5 patch is too small to clearly distinguish between different curvatures.

In the specific image results, we evaluated the computational efficiency of the algorithm with and without the learning module. On average, the learning module resulted in a 51.7% decrease in matching time.

Figs. 11–16 display the left image in (a), the right image in (b), and the reconstructed surface in (c). For the images in Figs. 12–16, the matches were checked visually. Thus, the expected accuracy is approximately one pixel. Also, the template points (the points to be matched) were chosen as those points which had a large absolute value of the x derivative except where stated explicitly.

The volleyball poster in Fig. 11 was chosen because it demonstrates basic matching accuracy in object space. Since all of the points lie on a plane, it is trivial to check whether a match is correct. The greatest deviation in the Z axis for the reconstructed surface from the average was 0.8 cm. For this image the dominant feature classes are intensity and x derivative of intensity.

The person stereo pair in Fig. 12 was selected to demonstrate the potential of using stereo to perform human body measurement. Note that for this image and for the street image, points with $Z = 0$ were added to aid the visual interpretation of the surface plot. The street scene in Fig. 13 demonstrates the possibility of automated terrain mapping. The content includes a street, trees, and telephone poles. This image is particularly difficult to interpret from the surface plot. An ideal surface plot would show a plane extending away from the user. The trees in the background are effectively at infinity due to resolution limitations. One major difficulty in matching images of a street is that the pavement has minimal texture. Another difficulty is the extreme change in depth which results in perspective distortions. The rock wall stereo pair in Fig. 14 depicts a rock wall which changes rapidly in depth. This image was selected because it shows the potential for a stereo matcher to perform automated terrain mapping and because it is a benchmark image from the difficult category of the Stuttgart standardized image set [7]. The reconstructed surface shows the mismatched points as sharp jagged peaks. The robots stereo pair in Fig. 15 shows three industrial robots. This stereo pair was chosen for the similarity to industrial manufacturing environments. The face stereo pair shown in Fig. 16 was chosen to show the potential for human body measurement using stereo matching. Note that the resolution in the surface reconstruction was sufficient to show the eyes and nose, but not the lips. The reconstructed surface shows that the eyes are slightly too sunken. This is due to the limitations of the resolution of the image to resolve depth sufficiently accurately.

Overall, the percentage of correct matches from the Landmark algorithm was consistently higher than that of the pyramidal method, although it should be noted that the pyramidal method required less processing time, roughly 0.06 sec. versus 0.4 sec.

V. Conclusion

From the final report of the NSF Workshop on "Challenges in Computer Vision Research; Future Research Directions," two of the major recommendations on research topics and issues included: 1) more experimental rigor in vision research and 2) researchers should address the integration of isolated modules at each visual processing level [20]. The Landmark algorithm addresses both of these recommendations.

The set of test images were real images of complex scenes which would be found in practical applications such as terrain mapping, human body measurement, and industrial manufacturing. In order to show the efficacy of the adaptive feature selection, we compared it to two alternate methods. These alternate methods are 1) Sum All which finds a correspondence by finding the minimum of a function defined as the sum of all of the feature errors, and 2) Best Single which finds a correspondence by finding the minimum of the feature error which has the largest class separation distance. The proposed adaptive method was on average 18 and 8 percent more accurate than the Sum All and Best Single methods, respectively.

For the purposes of bench marking, the Landmark algorithm was compared to a single feature pyramid matching algorithm. The matching accuracy of the Landmark algorithm ranged from 91% to 99% while the matching accuracy for the pyramid algorithm ranged from only 52% to 95%.

The learning module was evaluated upon the merit of increasing computation efficiency since that was the motivating reason for including it. On average, the learning module more than doubled the computational efficiency of the Landmark algorithm.

The main contributions of this paper are

1) integrating the modules of learning, feature selection, and surface reconstruction;
2) extensive empirical testing on real images;
3) self-diagnosis to determine when the feature set is insufficient to discriminate between match possibilities;
4) a method of guided maximization of the class separation criterion;
5) an instance based learning algorithm designed for feature selection in stereo matching that uses image independent and image dependent knowledge.

Future research will focus on recognition applications.

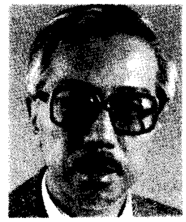
References

[1] D. W. Aha and D. Kibler, "Noise-tolerant instance-based learning algorithms," in *Proc. Int. Joint Conf. Artificial Intell.*, 1989, pp. 794–799.
[2] S. T. Barnard, "Stereo matching by hierarchical, microcanonical annealing," in *IEEE Proc. Image Understanding Workshop*, Feb. 1987.
[3] L. Cohen, L. Vinet, and P. Sander, "Hierarchical region based stereo matching," in *IEEE Proc. Comput. Vision and Pattern Recognit.*, 1989, pp. 416–421.
[4] J. D. Cowan and D. H. Sharp, "Neural nets and artificial intelligence," *Daedalus*, vol. 117, no. 1, pp. 85–121, 1988.
[5] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*. Englewood Cliffs, NJ: Prentice Hall, 1982.
[6] W. E. L. Grimson, *From Images to Surfaces: A Computational Study of the Human Early Visual System*. Cambridge, MA: MIT Press, 1981.
[7] E. Gulch, "Results of test on image matching of ISPRS WG III/4," *Int. Archives of Photogrammetry and Remote Sensing*, vol. 27-III, pp. 254–271, 1988.
[8] J. G. Harris, "A new approach to surface reconstruction: The coupled depth/slope model," in *First Int. Conf. Comput. Vision*, 1987, pp. 277–283.
[9] M. J. Hannah, "A system for digital stereo image matching," *Photogrammetric Eng. Remote Sensing*, vol. 55, no. 12, pp. 1765–1770, Dec. 1989.
[10] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision* Reading, MA: Addison-Wesley, vol. II, 1993, pp. 357–361.
[11] W. A. Hoff and N. Ahuja, "Depth from stereo," in *Proc. Fourth Scandinavian Conf. Image Anal.*, Trondheim, Norway, June 18–20, 1985, pp. 761–768.

[12] J. Kittler, "Une generalisation de quelques algorithmes sous-optimaux de recherche d'ensembles d'attributs," in *Proc. Reconn. des Formes et Trait. des Images*, Paris, 1978, pp. 678–686.
[13] H. S. Lim and T. O. Binford, "Stereo correspondence: A hierarchical approach," in *IEEE Proc. Image Understanding Workshop*, Feb. 1987, pp. 234–241.
[14] S. B. Marapane and M. M. Trivedi, "Multi-primitive hierarchical (MPH) stereo system," *IEEE Proc. Comput. Vision Pattern Recognit.*, 1992, pp. 499–505.
[15] T. Marill and D. M. Green, "On the effectiveness of receptors in recognition systems," *IEEE Trans. Inform. Theory*, vol. IT-9, pp. 11–17, Jan. 1963.
[16] D. Marr and T. Poggio, "A theory of edge detection," in *J. Roy. Soc. Lon. B*, vol. 204, pp. 301–328, 1980.
[17] D. M. McKeown and Y. C. Hsieh, "Hierarchical waveform matching: A new feature-based stereo technique," in *IEEE Proc. Comput. Vision and Pattern Recognit.*, 1992, pp. 513–519.
[18] H. P. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover," Ph.D. thesis, Stanford Univ., Rep. STAN-CS-80-813, 1980.
[19] P. M. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Trans. Comput.*, vol. C-26, pp. 917–922, Sept. 1977.
[20] S. Negahdaripour and A. K. Jain, "Challenges in computer vision: future research directions," *IEEE Proc. Comput. Vision Pattern Recognition*, pp. 189–199, 1992.
[21] D. Terzopoulos, "Multi-level computational processes for visual surface reconstruction," *Comput. Vision, Grahics, Image Processing* vol. 24, pp. 52–96, 1983.
[22] D. Terzopoulos, "The computation of visible-surface reconstruction," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 4, pp. 417–437, 1988.
[23] J. N. Weng, N. Ahuja, and T. S. Huang, "Matching two perspective views," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, no. 8. pp. 806–825, 1992.
[24] A. Whitney, "A direct method of nonparametric measurement selection," *IEEE Trans. Comput.*, vol. C-20, pp. 1100–1103, 1971.

Michael S. Lew received the B.S. degree from Rensselaer Polytechnic Institute, Troy, in 1987 and the M.S. degree from the University of Illinois at Urbana-Champaign, Urbana, in 1990. He is currently writing his doctoral dissertation.

He has been with the Center for Artificial Intelligence, Fort Belvoir, and then the Advanced Construction and Technology Center, Urbana, which granted him a fellowship toward his Ph.D. studies. He is currently working at the Beckman Institute for Advanced Studies and Technology. He has published papers involving stereo matching, image pyramids, surface interpolation and reconstruction, feature selection, camera calibration, and image compression. His current research interests are in Human Computer Interaction (HCI), video compression, and image processing. He is particularly interested in model based coding, and using optimal (Kalman type) filtering in image understanding applications.



Thomas S. Huang (S'61–M'63–SM'76–F'79) received the B.S. Degree in electrical engineering from National Taiwan University, Taipei, Taiwan, China; and his M.S. and Sc. D. Degrees in Electrical Engineering from the Massachusetts Institute of Technology, Cambridge, MA.

He was on the Faculty of the Department of Electrical Engineering at MIT from 1963 to 1973; and on the Faculty of the School of Electrical Engineering and Director of its Laboratory for Information and Signal Processing at Purdue University from 1973 to 1980. In 1980, he joined the University of Illinois at Urbana-Champaign, where he is now Professor of Electrical and Computer Engineering and Research Professor at the Coordinated Science Laboratory, and Head of the Image Formation and Processing Group at the Beckman Institute for Advanced Science and Technology. During his sabbatical leaves: Dr. Huang has worked at the MIT Lincoln Laboratory, the IBM Thomas J. Watson Research Center, and the Rheinishes Landes Museum in Bonn, West Germany, and held visiting Professor Positions at the Swiss Institutes of Technology in Zurich and Lausanne, University of Hannover in West Germany, INRS-Telecommunications of the University of Quebec in Montreal, Canada and University of Tokyo, Japan. He has served as a consultant to numerous industrial firms and government agencies both in the U.S. and abroad.

Dr. Huang's professional interests lie in the broad area of information technology, especially the transmission and processing of multidimensional signals. He has published 11 books, and over 250 papers in Network Theory, Digital Filtering, Image Processing, and Computer Vision. He is a Fellow of the Optical Society of America; and has received a Guggenheim Fellowship (1971–72), an A.V. Humboldt Foundation Senior U.S. Scientist Award (1976–77), and a Fellowship from the Japan Association for the Promotion of Science (1986). He received the IEEE Acoustics, Speech, and Signal Processing Society's Technical Achievement Award in 1987, and the Society Award in 1991. He is an Editor of the *International Journal Computer Vision, Graphics, and Image Processing*; and Editor of the Springer Series in *Information Sciences*, published by Springer Verlag; and Editor of the *Research Annual Series on Advances in Computer Vision and Image Processing* published by the JAI Press.

Kam W. Wong received his B.S. degree from the University of New Brunswick, and the M.S. and Ph. D. degrees from Cornell University.

He is a Professor of Civil Engineering at the University of Illinois at Urbana- Champaign, where he has taught photogrammetry and surveying since 1967. His current research interests include computer vision metrology, digital photogrammetry, and engineering surveying. He is the ASPRS Correspondent to Commission V of ISPRS, and is Co-Chair of ISPRS Working Group V/1: Knowledge Based Vision Metrology. He has coauthored two books and published more than 30 technical papers.