ADVANCES IN COMPUTATIONAL STEREO

M. Z. Brown, D. Burschka and G. D. Hager

Department of Computer Science
The Johns Hopkins University

ABSTRACT

Extraction of three-dimensional structure of a scene from stereo images is a problem that has been studied by the computer vision community for decades. Early work focused on the fundamentals of image correspondence and stereo geometry. Stereo research has matured significantly throughout the years, and many advances in computational stereo continue to be made, allowing stereo to be applied to new and more demanding problems. In this paper, we review recent advances in computational stereo, focusing primarily on three important topics: correspondence methods, methods for occlusion and real-time implementations. Throughout, we present tables that summarize and draw distinctions among key ideas and approaches. Where available, we provide comparative analyses, and we make suggestions for analyses yet to be done.

INDEX TERMS

Computational stereo, stereo correspondence, occlusion, real-time stereo, review

1. INTRODUCTION

Computational stereo for extraction of three-dimensional scene structure has been an intense area of research for decades. Early work, conducted in the seventies and early eighties, was primarily done by the Image Understanding (IU) community and funded by the Advanced

Research Projects Agency (ARPA). Barnard and Fischler [4] reviewed stereo research through 1981, focusing on the fundamentals of stereo reconstruction, criteria for evaluating performance and a survey of well-known approaches at that time. Stereo continued to be a significant focus of research in the computer vision community through the eighties. Dhond and Aggarwal [23] reviewed many stereo advances in that decade, including a wealth of new matching methods, the introduction of hierarchical processing, and the use of trinocular constraints to reduce ambiguity in stereo. By the early 1990s, stereo research had, in many ways, matured. Although some general stereo matching research continued, much of the community's focus turned to more specific problems. In an unpublished report, Koschan [47] surveyed stereo techniques developed between 1989 and 1993, including early research on occlusion and transparency, active and dynamic stereo and real-time stereo implementations. Substantial progress in each of these lines of research has been made in the last decade, and new trends have emerged.

In this paper, we review advances in computational stereo over roughly the last decade. Overall, significant progress has been made in several areas, including new techniques for area- and feature-based matching, methods for dealing with occlusion, multi-camera stereo, stereo and motion and real-time implementations. Due to space limitations, we have decided to focus this review on three topics: correspondence methods, methods for dealing with occlusion and real-time implementations. Recent books by Hartley and Zisserman [33] and Faugeras and Luong [28] provide a wealth of information on the geometric aspects of multiple view stereo. We have also elected not discuss recent developments in performance analysis but instead refer to a very complete and recent discussion by Scharstein and Szeliski [69] (See also http://www.middlebury.edu/stereo for implementations, test data and results).

No review of this nature can cite every paper that has been published. We have included

what we believe to be a representative sampling of important work and broad trends from the previous decade. In many cases, we provided additional tables of references in order to better summarize and draw distinctions among key ideas and approaches. We have also provided quantitative comparisons of algorithm complexity and performance wherever possible.

The remainder of this article is structured as follows. Section 2 briefly reviews the fundamentals of computational stereo and establishes basic terminology. Section 3 discusses local and global correspondence methods. Section 4 discusses the problem of occlusion and presents a taxonomy of methods for handling it. Section 5 surveys the state of the art in real-time stereo implementations and discusses the progression of stereo systems from special-purpose hardware to general-purpose computers. We conclude in section 6 and offer our impressions of current and future trends in computational stereo.

2. COMPUTATIONAL STEREO

Computational stereo refers to the problem of determining 3-dimensional structure of a scene from two or more images taken from distinct viewpoints. The fundamental basis for stereo is the fact that a single three-dimensional physical location projects to a unique pair of image locations in two observing cameras (Figure 2-1). As a result, given two camera images, if it is possible to locate the image locations that correspond to the same physical point in space, then it is possible to determine its three-dimensional location.

The primary problems to be solved in computational stereo are *calibration*, *correspondence* and *reconstruction*. Calibration is the process of determining camera system external geometry (the relative positions and orientations of each camera) and internal geometry (focal lengths, optical centers and lens distortions). Accurate estimates of this geometry are necessary in order

to relate image information (expressed in pixels) to an external world coordinate system. The problem of estimating calibration is at this point well understood, and high-quality toolkits are available (e.g., http://www.vision.caltech.edu/bouguetj/calib_doc/ and links therein). For good discussions of recent work on calibration, see [28] and [33]. In our discussions below, we assume the camera calibration to be static and known.

Consider now the camera configuration shown in Figure 2-1. We define the *baseline* of the stereo pair to be the line segment joining the optical centers O_L and O_R. In the *non-verged* geometry depicted in the figure, both camera coordinates axes are aligned, and the baseline is parallel to the camera x coordinate axis. It follows that, for the special case of non-verged geometry, a point in space projects to two locations on the same scan line in the left and right camera images. The resulting displacement of a projected point in one image with respect to the other is termed *disparity*. The set of all disparities between two images is called a *disparity map*. Clearly, disparities can only be computed for features visible in both images; features visible in one image but not the other are said to be *occluded*. How to handle occluded features is one of the key problems in computational stereo and is further discussed in Section 4.

In practice, we are given two images, and from the information contained in this image, we must compute disparities. The *correspondence problem* consists of determining the locations in each camera image that are the projection of the same physical point in space. No general solution to the correspondence problem exists, due to ambiguous matches (e.g., due to occlusion, specularities, or lack of texture). Thus, a variety of constraints (e.g., epipolar geometry) and assumptions (e.g., image brightness constancy and surface smoothness) are commonly exploited to make the problem tractable. For a complete discussion, see chapter 4 of [43].

The *reconstruction problem* consists of determining 3-dimensional structure from a

disparity map, based on known camera geometry. The depth of a point in space P imaged by two cameras with optical centers O_L and O_R is defined by intersecting the rays from the optical centers through their respective images of P, p and p' (see figure 2-1). Given the distance between O_L and O_R, called the *baseline* T, and the focal length f of the cameras, depth at a given point may be computed by similar triangles as

$$Z = f\frac{T}{d}, \tag{2-1}$$

where d is the disparity of that point, $d=x-x'$ (from figure 2-1), after being converted to metric units. This process is called *triangulation*.

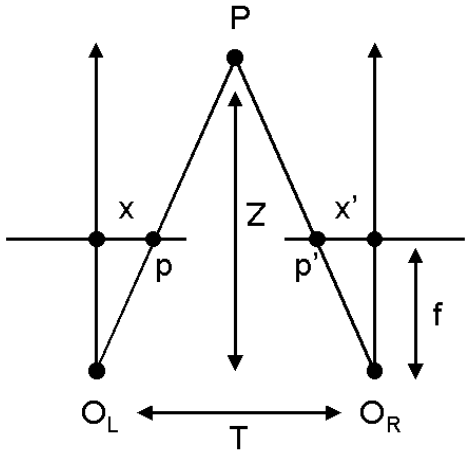

Figure 2-1 The geometry of non-verged stereo

In practice, it is difficult to build stereo systems with non-verged geometry. However, it is well-known that arbitrary stereo image pairs (i.e., with verged geometry) may also be rectified (resampled) to non-verged geometry by exploiting a binocular geometric constraint, commonly referred to as the *epipolar constraint*. Figure 2-2 shows the imaging geometry for two cameras with optical centers O_L and O_R. A point P in the scene is imaged by the left and right cameras respectively as points p and p'. The baseline T and optical rays O_L to P and O_R to P define the

plane of projection for the point *P*, called the *epipolar plane*. This epipolar plane intersects the image planes in lines called *epipolar lines*. The epipolar line through a point p' is the image of the opposite ray, O_L to *P* through point *p*. The point at which an image's epipolar lines intersect the baseline is called the *epipole* (*e* and e' for *p* and p' respectively), and this point corresponds to the image of the opposite camera's optical center as imaged by the corresponding camera. Given this unique geometry, the corresponding point p' of any point *p* may be found along its respective epipolar line. By rectifying the images such that corresponding epipolar lines lie along horizontal scan-lines, the two-dimensional correspondence search problem is again reduced to a scan-line search, greatly reducing both computational complexity and the likelihood of false matches. See [82] and [91] for details on algorithms to compute this rectification.

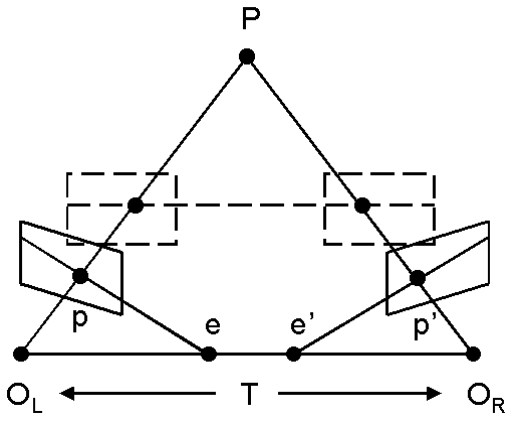

Figure 2-2 Two arbitrary images of the same scene may be rectified along epipolar lines (solid) to produce collinear scan lines (dashed).

3. CORRESPONDENCE

As noted previously, stereo disparities may be determined in a number of ways and by exploiting a number of constraints. All of these methods attempt to match pixels in one image with their corresponding pixels in the other image. For simplicity, we refer to constraints on a

small number of pixels surrounding a pixel of interest as *local* constraints. Similarly, we loosely refer to constraints on scan-lines or on the entire image as *global* constraints. Table 3-1 outlines the principal methods for exploiting both local and global constraints, excluding methods that rely explicitly on more than two views.

Table 3-1 Stereo matching approaches

APPROACH	REFERENCES	BRIEF DESCRIPTION
LOCAL METHODS		
Block Matching	[1], [7], [26], [89]	Search for maximum match score or minimum error over small region, typically using variants of cross-correlation or robust rank metrics.
Gradient-Based Optimization	[51], [44]	Minimize a functional, typically the sum of squared differences, over a small region.
Feature Matching	[8], [11], [23], [62], [72], [84]	Match dependable features rather than intensities themselves.
GLOBAL METHODS		
Dynamic Programming	[5], [9], [10], [19], [36], [60]	Determine the disparity surface for a scanline as the best path between two sequences of ordered primitives. Typically, order is defined by the epipolar ordering constraint.
Intrinsic Curves	[80], [81]	Map epipolar scanlines to intrinsic curve space to convert the search problem to a nearest-neighbors lookup problem. Ambiguities are resolved using dynamic programming.
Graph Cuts	[13], [14], [45], [65], [79], [92]	Determine the disparity surface as the minimum cut of the maximum flow in a graph.
Nonlinear Diffusion	[52], [68], [71]	Aggregate support by applying a local diffusion process.
Belief Propagation	[77]	Solve for disparities via message passing in a belief network.
Correspondenceless Methods	[27], [30], [48]	Deform a model of the scene based on an objective function.

Local methods can be very efficient, but they are sensitive to locally ambiguous regions in images (e.g., occlusion regions or regions with uniform texture). Global methods can be less sensitive to these problems, since global constraints provide additional support for regions difficult to match locally. However, these methods are more computationally expensive. Comparative results of three of the most commonly used algorithms are shown in Figure 3-1. The following subsections discuss the principal local and global methods for stereo correspondence, their efficiencies and their limitations. Where available, complexity and

performance comparisons are provided.

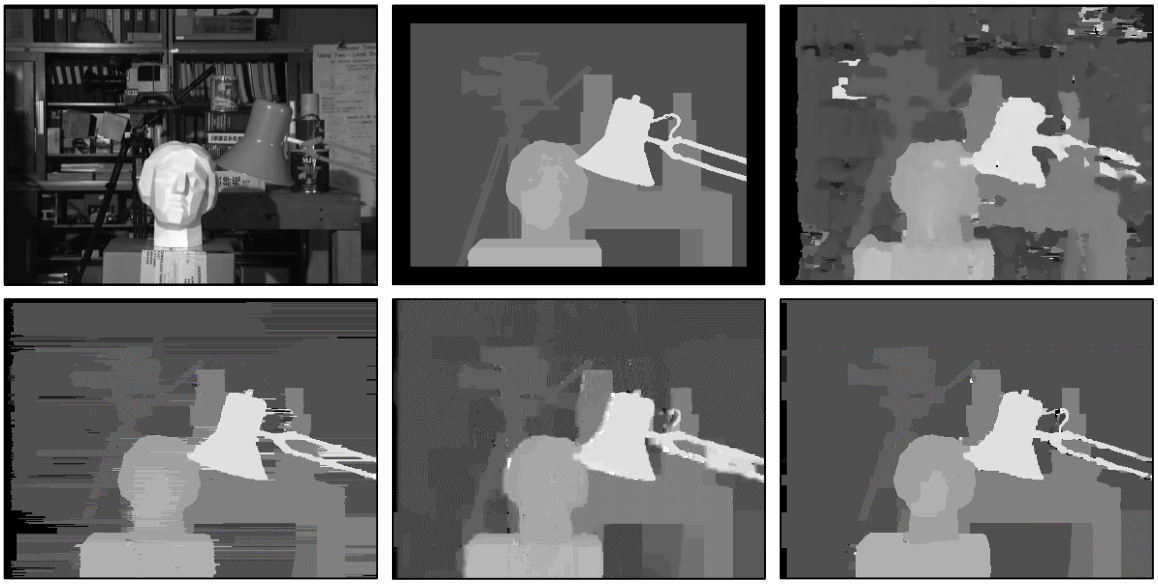

Figure 3-1 Comparative results on images from the University of Tsukuba, provided by Scharstein and Szeliski [69]. Left to right: left stereo image, ground truth, Muhlmann et al.'s area correlation algorithm [57], dynamic programming (similar to Intille and Bobick [36]), Roy and Cox's maximum flow [65] and Komolgorov and Zabih's graph cuts [45].

3.1 Local Correspondence Methods

In this section, we compare and contrast several local correspondence algorithms in terms of both performance and efficiency. These methods fall into three broad categories: block matching, gradient methods and feature matching.

3.1.1 Block Matching

Block matching methods seek to estimate disparity at a point in one image by comparing a small region about that point (the template) with a series of small regions extracted from the other image (the search region). As stated before, the epipolar constraint reduces the search to

one dimension. Three classes of metrics are commonly used for block matching: correlation, intensity differences, and rank metrics (see Table 3-2).

Table 3-2 Common block-matching methods (see Figure 3-2 for visual description of terms)

MATCH METRIC	DEFINITION
Normalized Cross-Correlation (NCC)	$\frac{\sum_{u,v}\left(I_1(u,v)-\bar{I}_1\right)\cdot\left(I_2(u+d,v)-\bar{I}_2\right)}{\sqrt{\sum_{u,v}\left(I_1(u,v)-\bar{I}_1\right)^2\cdot\left(I_2(u+d,v)-\bar{I}_2\right)^2}}$
Sum of Squared Differences (SSD)	$\sum_{u,v}\left(I_1(u,v)-I_2(u+d,v)\right)^2$
Normalized SSD	$\sum_{u,v}\left(\frac{\left(I_1(u,v)-\bar{I}_1\right)}{\sqrt{\sum_{u,v}\left(I_1(u,v)-\bar{I}_1\right)^2}}-\frac{\left(I_2(u+d,v)-\bar{I}_2\right)}{\sqrt{\sum_{u,v}\left(I_2(u+d,v)-\bar{I}_2\right)^2}}\right)^2$
Sum of Absolute Differences (SAD)	$\sum_{u,v}\left\lvert I_1(u,v)-I_2(u+d,v)\right\rvert$
Rank	$\sum_{u,v}\left(I_1'(u,v)-I_2'(u+d,v)\right)$ $I_k'(u,v)=\sum_{m,n}I_k(m,n)<I_k(u,v)$
Census	$\sum_{u,v}HAMMING\left(I_1'(u,v),I_2'(u+d,v)\right)$ $I_k'(u,v)=BITSTRING_{m,n}\left(I_k(m,n)<I_k(u,v)\right)$

Normalized cross-correlation (NCC) is the standard statistical method for determining similarity. Its normalization, both in the mean and the variance, makes it relatively insensitive to radiometric gain and bias. The sum of squared differences (SSD) metric is computationally simpler than cross-correlation, and it can be normalized as well. In addition to NCC and SSD, many variations of each with different normalization schemes have been used. One popular example is the sum of absolute differences (SAD), which is often used for computational efficiency. See Aschwanden and Guggenbuhl [1] for an extensive comparison of these metrics.

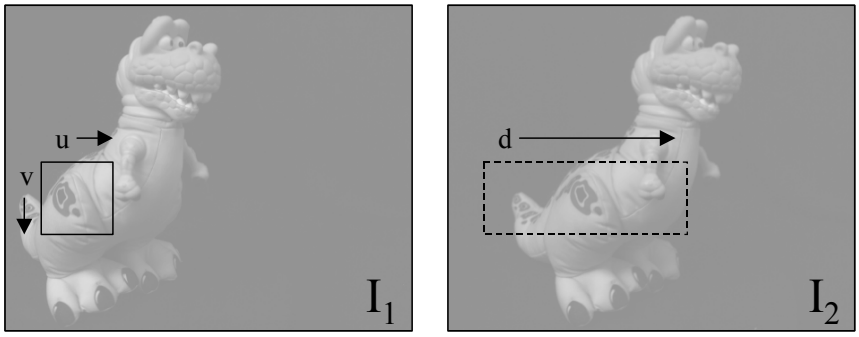

Figure 3-2 Block matching searches one image for the best corresponding region for a template region in the other image. Correspondence metrics are outlined in Table 3-2.

Zabih and Woodfill [89] propose an alternative method for computing correspondence by applying local non-parametric transforms to the images before matching. In order to eliminate sensitivity to radiometric gain and bias, a rank transform is applied locally to regions in both images. The rank transform for a local region about a pixel is defined as the number of pixels in that region for which the intensity is less than that of the center pixel. The resulting values are based on the relative ordering of pixel intensities rather than the intensities themselves (see Figure 3-3). Since the magnitudes of these values are much compressed, sensitivity to outliers (e.g., due to occlusion) is also reduced. After the rank transform is applied, block matching is performed using the L_1 norm (i.e., sum of absolute differences).

While the rank transform method reduces sensitivity to radiometric gain and bias, it also reduces the discriminatory power of the matching procedure, since information is lost. The relative ordering of all of the pixels surrounding a given pixel to be transformed is encoded in a single value. Zabih and Woodfill [89] propose a variation of the rank transform, called the census transform, that preserves the spatial distribution of ranks by encoding them in a bit string (see Figure 3-3). Matching is then performed using the Hamming distance (i.e., the number of bits that differ) between bit strings. This transform increases the dimensionality of the image

data by a factor of the local region size, making it computationally expensive. This algorithm requires a massively parallel machine for real-time implementation. Two such implementations exploiting field programmable gate arrays (FPGAs) are discussed in section 5. Banks and Corke [3] compare the performance of rank and census matching with those of correlation and difference metrics. Their results indicate that rank and census methods perform comparably to standard metrics and are more robust to radiometric distortion and occlusion. For many of the test scenes, the difference between normalized cross-correlation and census matching was between 5 and 9 percent of the total number of pixels.

89	63	72	89	63	72
67	55	$64 \Rightarrow 2$	67	55	$64 \Rightarrow 00000011$
58	51	49	58	51	49

Figure 3-3 Example rank (left) and census (right) transforms.

The naïve implementation of any block matching method is very inefficient due to redundant computations. For an image with N pixels, a template size of n pixels and a disparity search range of D pixels, the complexity of naïve block matching is $O(NDn)$ operations. By keeping running block sums, redundant computations may be avoided and block matching complexity may be reduced to $O(ND)$ operations, making it independent of the template size. Variations include [26], [56], [57] and [76]; [57] provides a thorough discussion.

3.1.2 Gradient Methods

Gradient-based methods, or optical flow [35], seek to determine small local disparities between two images by formulating a differential equation relating motion and image brightness.

In order to do this, the assumption is made that the image brightness of a point in the scene is constant between the two views. Then, the horizontal translation of a point from one image to the other is computed by a simple differential equation,

$$(\nabla_x E)v + E_t = 0 \tag{3-1}$$

where $\nabla_x E$ denotes the horizontal component of the image gradient, E_t denotes the temporal (here referring to the intensity differences between left and right stereo images) derivative, and v denotes the translation between the two images. The complexity of matching along epipolar lines using optical flow is simply *O(N)*.

Note that only translation in the direction of the gradient at a given point may be estimated accurately. Therefore, an additional constraint is necessary to achieve reliable results. If it may be assumed that disparity varies smoothly over a small window of pixels, then the disparity at a point may be estimated using least squares on the system of linear differential equations at each pixel in an n pixel window of points $p_1, p_2, \ldots, p_n$ about that point [51],

$$v = \left(A^T A\right)^{-1} A^T b \tag{3-2}$$

where

$$A = \begin{bmatrix} \nabla_x E(p_1) \\ \nabla_x E(p_2) \\ \vdots \\ \nabla_x E(p_{nxn}) \end{bmatrix} \text{ and } b = -\begin{bmatrix} E_t(p_1) \\ E_t(p_2) \\ \vdots \\ E_t(p_{nxn}) \end{bmatrix}. \tag{3-3}$$

The metric used here is the same sum of squared differences (SSD) used in the block matching technique. The asymptotic complexity is again *O(Nn)*, which is comparable to exhaustive search. In addition to estimating translation (i.e., disparity) for each pixel, this framework may be extended to also estimate more general transformations. For instance, Gruen [32] estimates local affine transformations between images in order to compensate for perspective effects. It is

worth noting that computation required to perform the optimization grows linearly with the number of parameters estimated.

Kluth et al. [44] have proposed an efficient implementation of least squares matching that imposes global constraints using array algebra [63]. The cost function,

$$(\nabla_x E)v + b + E_t = 0\,, \tag{3-4}$$

includes a term b for radiometric bias. Their approach solves for N disparities in $O(N \log N)$ operations [44]. While this is a factor of $\log N$ less efficient than block matching with running sums for a single pixel of disparity (i.e., D=1), it is about as efficient as the gradient-based method and has the advantage of incorporating global constraints.

In theory, gradient-based methods can only estimate disparities up to half a pixel, since the local derivatives are only valid over that range. Since adjacent image pixels are typically highly correlated, a pixel or more can often be estimated in practice. However, hierarchical processing is a necessity for applying these methods to stereo, where disparity ranges are typically much larger than one pixel. However, even with hierarchical processing, the amount of parallax that this method can capture is fundamentally limited by the size of the feature being measured.

3.1.3 Feature Matching

Block matching and gradient methods are well known to be sensitive to depth discontinuities, since the region of support near a discontinuity contains points from more than one depth. These methods are also sensitive to regions of uniform texture in images. Feature-based methods seek to overcome these problems by limiting the regions of support to specific reliable features in the images (e.g., edges [8, 84], curves [72], etc.). Of course, this also limits

the density of points for which depth may be estimated. Throughout the 1980s, feature-matching methods for stereo correspondence received significant attention, largely due to their efficiency. The review by Dhond and Aggarwal [23] provides an account of much of this work. Due to the need for dense depth maps for a variety of applications and also due to improvements in efficient and robust block matching methods, interest in feature-based methods has declined in the last decade. In the following paragraphs, we discuss two classes of feature-based approaches that have received recent attention: hierarchical feature matching and segmentation matching.

Venkateswar and Chellappa [84] have proposed a hierarchical feature-matching algorithm exploiting four types of features: lines, vertices, edges and edge-rings (i.e., surfaces). Matching begins at the highest level of the hierarchy (surfaces) and proceeds to the lowest (lines). The feature-based hierarchical framework serves much the same purpose as area-based hierarchical frameworks. It allows coarse, reliable features to provide support for matching finer, less reliable features, and it reduces the computational complexity of matching by reducing the search space for finer levels of features. First, edges are extracted and the feature hierarchy is built from the bottom up based on structural (i.e., connectivity) and perceptual (i.e., parallel, collinear and proximate) relationships. Incompatibility relations (e.g., intersects, overlaps and touches) are also used to enforce consistent feature groupings. All potential features in the hierarchy are stored as hypotheses in a relational graph (see Figure 3-4). Inconsistent groupings are pruned from the graph by a truth maintenance system (TMS). Feature matching is then performed between the relational graphs of the stereo images, beginning with surfaces and proceeding to lines. For two surfaces to match, their component edges must match, and so on. Once a higher-level feature match has been confirmed, the component features are no longer included in the search for other lower-level matches, since a feature cannot belong to more than

one group. This reduces the search space significantly at each level of the hierarchy.

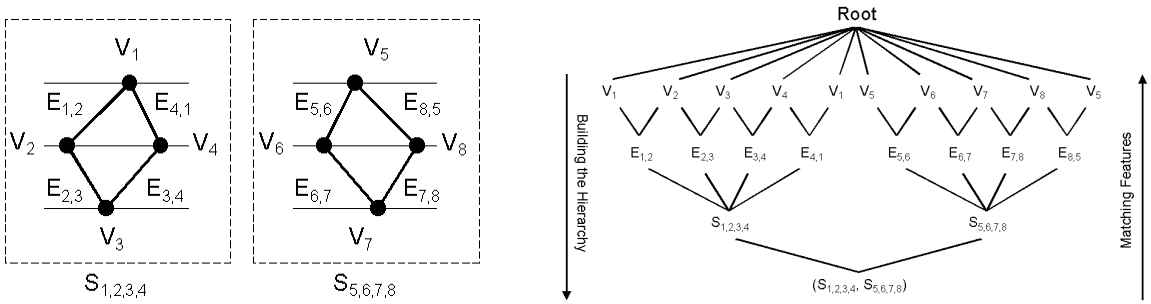

Figure 3-4 Two hypothesized surfaces (left) and their component features are represented in a relational graph (right). The hierarchy is built from the lowest level to the highest (vertices, edges and surfaces in this example), and matching is performed from the highest to the lowest.

Venkateswar and Chellappa [84] report the matching complexity of each level of the hierarchy to be $O(N^4)$, where N is the number of features examined at that level, and the typical improvement provided by hierarchical matching to be a factor of 100. Of course, the improvement will vary with the percentage of higher-level features successfully matched.

Another feature-based approach is to first segment the images and then match the segmented regions [11, 62]. Birchfield and Tomasi [11] segment stereo images into small planar patches for which correspondence is then determined. As with most feature-based methods, this reduces the match sensitivity to depth discontinuities. However, these planes are likely to be slanted rather than fronto-parallel (i.e., directly facing the cameras), so the relationships between segments in the two images are modeled by six parameter affine transformations, such that

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = A \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + d \qquad (3\text{-}5)$$

where

$$A = \begin{bmatrix} 1 + d_{xx} & d_{xy} \\ d_{yx} & 1 + d_{yy} \end{bmatrix} \text{ and } d = \begin{bmatrix} d_x \\ d_y \end{bmatrix} \tag{3-6}$$

and where (x_1,y_1) and (x_2,y_2) are the coordinates of corresponding points in the left and right images respectively. The vector d defines the translation of a segment between frames, and the matrix A defines the in-plane rotation, scale and shear transformations between frames. The parameters are computed directly from spatio-temporal intensity gradients, as in [73]. For epipolar-rectified imagery, only the horizontal parameters are computed. Segmentation and affine parameter estimation are computed iteratively, and patches with similar affine parameters are merged after each iteration. The segmentation algorithm used is based on the multiway cut algorithm of Boykov et al. [12]. A similar graph theoretic algorithm is discussed in section 3.2.3 for global stereo correspondence. Unlike most feature-based methods, dense disparities are explicitly defined for this segmentation-based method by planar transformations. However, this approach is also sensitive to the quality of the original segmentation.

3.2 Global Correspondence Methods

As stated above, global correspondence methods exploit non-local constraints in order to reduce sensitivity to local regions in the image that fail to match, due to occlusion, uniform texture, etc. The use of these constraints makes the computational complexity of global matching significantly greater than that of local matching. This section reviews three global optimization approaches. By far the most common approach to global matching is dynamic programming, which uses the ordering and smoothness constraints to optimize correspondences in each scan-line. Two variants of dynamic programming are presented here, the standard method and one based on Tomasi and Manduchi's intrinsic curves [81]. More recent graph cut, diffusion and belief propagation methods, all of which incorporate more general two-

dimensional local cohesion constraints, are also presented.

3.2.1 Dynamic Programming

Dynamic programming is a mathematical method that reduces the computational complexity of optimization problems by decomposing them into smaller and simpler sub-problems [20]. A global cost function is computed in stages, with the transition between stages defined by a set of constraints. For stereo matching, the epipolar monotonic ordering constraint allows the global cost function to be determined as the minimum cost path through a disparity-space image (DSI). The cost of the optimal path is the sum of the costs of the partial paths obtained recursively. The local cost functions for each point in the DSI may be defined using one of the area-based methods in section 3.1.1. There are two ways to construct a DSI, shown in Figure 3-5. First, the axes may be defined as the left and right scanlines, as is done by Ohta and Kanade [60] and Cox et al. [19]. For this case, dynamic programming is used to determine the minimum cost path from the lower left corner to the upper right corner of the DSI. With N pixels in a scanline, the computational complexity using dynamic programming for this type of DSI is $O(N^4)$, in addition to the time required to compute the local cost functions. The number of nodes in the search plane is $O(N^2)$, and the number of paths to evaluate per node is $O(N^2)$. By placing a constraint on the maximum disparity, the number of paths to evaluate and hence the computational complexity may be greatly reduced. Also, computational complexity may be reduced at the expense of optimality. For instance, Cox et al. [19] perform a greedy search in $O(ND)$. The second method for constructing a DSI is to define the axes as the left scanline and the disparity range, as is done by Intille and Bobick [36]. For this case, dynamic programming is used to determine the minimum cost path from the first column to the last column (see Figure 3-

5). With N pixels in a scanline and a disparity range of D pixels, full global optimization requires $O(D^N)$ operations per scanline, in addition to time required to compute local cost functions. Using dynamic programming, the complexity is only $O(ND^2)$, much smaller than that of the other representation. Birchfield and Tomasi [9] propose a greedy algorithm for reducing this complexity to $O(NDlogD)$ by pruning nodes when locally lower cost alternatives are available.

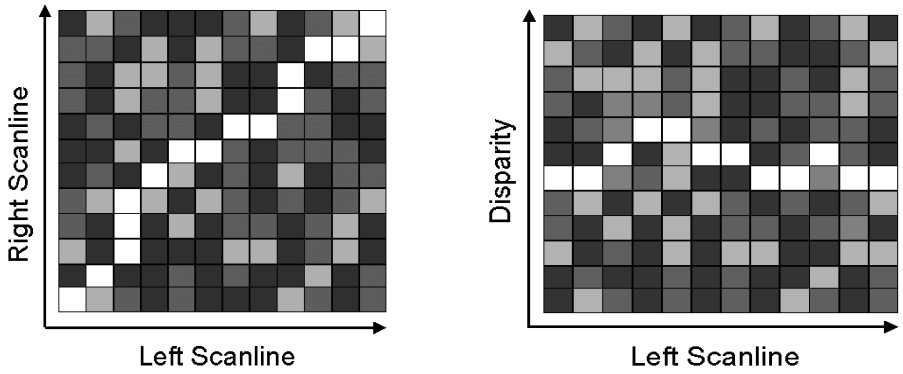

Figure 3-5 An example disparity-space image using left-right axes (left) and another using left-disparity axes (right). Intensities shown represent the respective costs of potential matches along the scan-lines, with lighter intensities having lower cost.

In addition to minimizing the global cost for independent scanlines (i.e., intra-scanline search), inter-scanline constraints may also be applied to reduce ambiguity. Baker [2] proposed a dynamic programming method that first computes disparities independently for each scanline and then detects and corrects estimates that violate inter-scanline consistency constraints. Ohta and Kanade [60] proposed to integrate the inter-scanline constraints into the match process by minimizing the sum of costs over two-dimensional regions defined as intervals between vertical edges. The local cost function for each interval is defined as the variance of the pixel intensities in that interval. Belhumeur [5] proposed a two-stage approach, first computing intra-scanline

solutions using dynamic programming and then smoothing disparities between scanlines. This smoothing is done by taking each adjacent three scanlines, fixing the disparities of the outer two and then recomputing the optimum solution for the middle scanline using dynamic programming. Cox et al. [19] propose enforcing two-dimensional cohesiveness constraints without smoothing by minimizing the number of horizontal and vertical discontinuities (i.e., by penalizing discontinuities). Vertical discontinuities are minimized between adjacent scanlines, either in a one-pass scheme where minimization for a scanline is constrained by the previous line, or in a two-pass scheme where scanlines above and below a given scanline are used to constrain the minimization. Birchfield and Tomasi [10] introduce a vertical constraint by propagating highly reliable disparity regions, defined by the amount of vertical support (i.e., the number of like disparities in a local column), into less reliable regions, bounded by intensity gradients which are assumed to be depth discontinuities.

One of the principal advantages of dynamic programming, or any global search method, is that it provides global support for local regions that lack texture and would otherwise be matched incorrectly. These local regions present little difficulty for a global search, since any cost function (e.g., intensity difference or variance) in these regions is low. Another problem that global search seeks to resolve is occlusion. This is more difficult, since a cost function applied near an occlusion boundary is typically high. Methods for dealing with this difficulty have been proposed in [5] and [10]. These methods replace matching costs at occlusion boundaries with a small fixed occlusion cost. In section 4.4, we examine these approaches as part of a broader discussion of the occlusion problem. The principal disadvantage of dynamic programming is the possibility that local errors may be propagated along a scan-line, corrupting other potentially good matches. Horizontal streaks caused by this problem may be observed in many of the

disparity map results reported in the literature (see Figure 3-1).

3.2.2 Intrinsic Curves

Tomasi and Manduchi [81] propose an alternative to conventional search for global matching, using a different representation of image scanlines, called intrinsic curves. An intrinsic curve is a vector representation of image descriptors defined by applying operators (e.g., edge and/or corner operators) to the pixels in a scanline. For N operators $p_1, p_2, \ldots, p_N$, the intrinsic curve C is defined in R^N as

$$C = \{p(x), x \in R\}, \text{ where } p(x) = (p_1(x), p_2(x), \ldots, p_N(x)). \quad (3\text{-}7)$$

A simple example is shown in Figure 3-6. The intrinsic curves here are defined by plotting the intensities of scanline pixels against their respective derivatives. This mapping is invariant to translation (i.e., disparity), so in the ideal case, matching pixels map to the same points along a common curve. In the general case, however, due to noise and perspective differences, matching pixels do not always map to exactly the same points, as can be seen in Figure 3-6. Thus, the disparity search problem is cast in intrinsic curve space as a nearest neighbor problem. See [80] and [81] for a discussion of efficient nearest-neighbors solutions. Ambiguities are resolved by maximizing a global metric using dynamic programming. Once matching has been achieved in intrinsic curve space, an inverse mapping is applied to determine disparities. Samples of uniform arc length on an intrinsic curve result in a non-uniform grid in image space, more densely sampled where the image content is busy and less densely sampled where the image lacks texture (i.e., many pixels map to the same point in intrinsic curve space). Thus, intrinsic curve matching is, in a sense, a feature-based algorithm and produces a sparse depth map rather than a dense one.

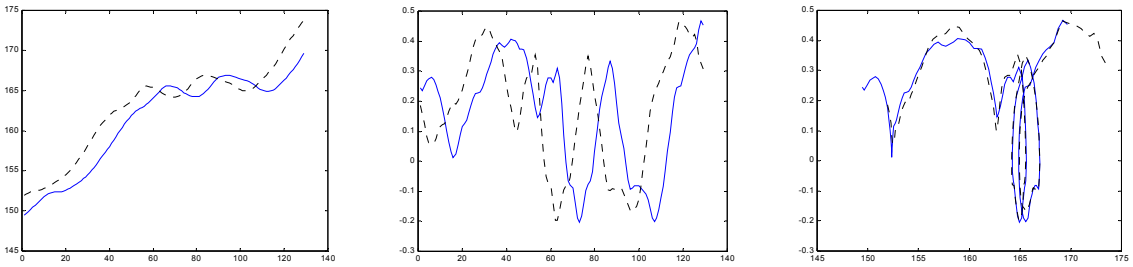

Figure 3-6 Left and right image scanline intensities (left), their derivatives (center) and the intrinsic curves formed by plotting one against the other (right).

The principal benefit of the intrinsic curve representation is its invariance to disparity. The nearest-neighbors distances between points on two curves representing left and right scanlines are not directly affected by the amount of disparity between them in image space. Therefore, multi-resolution methods commonly used to reduce computational complexity are unnecessary with this approach. Of course, this representation is still affected by occlusion and uniform or repetitive texture in the imaged scene. Global optimization via dynamic programming is used to compensate for such errors in much the same way it is used for conventional search. Occlusions in intrinsic curve space appear as unmatched arcs in a curve. Although this is a visibly noticeable indicator, no computational algorithm has yet been proposed to measure it. No comparative analysis of intrinsic curves with other correspondence methods has been published to date. Qualitative results (i.e., disparity maps) are reported in [81].

3.2.3 Graph Cuts

The most significant limitation of dynamic programming for stereo matching is its inability to strongly incorporate both horizontal and vertical continuity constraints. As already discussed, many approaches have been proposed to improve this situation while maintaining the dynamic

programming framework. However, these do not fully exploit the two-dimensional coherence constraints available. An alternative approach that exploits these constraints is to cast the stereo matching problem as that of finding the maximum flow in a graph [20].

Let us define a directed graph G=(V, E), where V is the vertex set and E is the edge set. The vertex set is defined based on the selected matching representation. Suggested representations for maximum flow graph construction parallel those proposed for DSI construction, left-disparity and left-right. Roy and Cox [65] propose a left-disparity representation. The vertex set is thus defined to be

$$V = V^* \cup \{s, t\} \tag{3-8}$$

where s is the source, t is the sink and

$$V^* = \{(x, y, d), x \in [0, x_{\max}], y \in [0, y_{\max}], d \in [0, d_{\max}]\}. \tag{3-9}$$

The graph axes correspond to the image horizontal and vertical axes and the disparity range. The edges are defined to be

$$E = \begin{Bmatrix} (u, v) \in V^* x V^* & : \|u - v\| = 1 \\ (s, (x, y, 0)) & : x \in [0, x_{\max}] \\ ((x, y, d_{\max}), t) & y \in [0, y_{\max}] \end{Bmatrix}. \tag{3-10}$$

Internally, the mesh is six-connected, and the vector norm in (3-10) constrains node pairs to be connected. Each node has an associated cost that is defined in the same way that the local costs are defined for dynamic programming. Each edge has an associated flow *capacity* that is defined as a function of the costs of the adjacent nodes it connects. This non-negative capacity limits the amount of flow that can be sent from the source to the sink. The capacity is defined to be infinity for both the source and the sink. A *cut* is a partition of the vertex set V into two subsets separating the source from the sink. The capacity of a cut is simply the sum of the edge capacities making up that cut. The cut with minimum capacity, the *minimum cut*, maximizes

flow through the graph (see Figure 3-7). It is more intuitive with respect to the stereo problem to consider the minimum cut rather than its associated maximum flow. This minimum cut is analogous to the best path along a pair of scan-lines determined by dynamic programming extended to three dimensions. Thus, the disparity estimates associated with the minimum cut are not only consistent across one scan-line but are also consistent globally throughout the image.

Naturally, graph cut methods generally require more computations than dynamic programming. Fortunately, many approaches have been developed for its efficient solution. Roy and Cox [65], Zhao [92] and Thomos et al. [79] use the well-known preflow-push lift-to-front algorithm [20]. The worst-case complexity of this algorithm is $O(N^2D^2\log(ND))$, significantly greater than that of dynamic programming algorithms. However, the average observed time reported by Roy and Cox [65] is $O(N^{1.2}D^{1.3})$, much closer to that of dynamic programming. One limitation of the lift-to-front algorithm is that classical implementations require significant memory resources, making this approach cumbersome for use with large images. Thomos et al. [79] have developed an efficient data structure that reduces the memory requirements by a factor of approximately four, making this algorithm more manageable for large data sets.

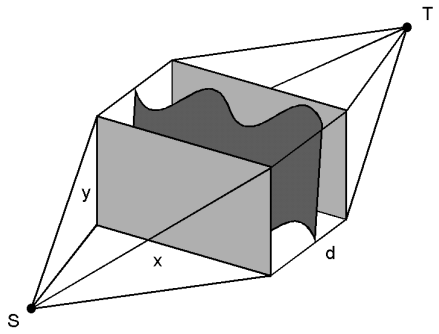

Figure 3-7 Maximum flow representation of disparity estimation as posed by Roy and Cox [65].

Roy and Cox [65] provide several qualitative comparisons of their method versus the dynamic programming method of Cox et al. [19]. These indicate that their maximum flow

method produces disparity maps with fewer horizontal streaks than dynamic programming. Thomos et al. [79] compared their implementation of maximum flow to a variety of dynamic programming methods using a synthetic stereo pair with ground truth. Their method correctly matched (up to an integer disparity) only 1.62% more pixels than the best performing dynamic programming method, that of Intille and Bobick [36]. Results for Thomos et al.'s maximum flow and Intille and Bobick's dynamic programming were 46.74% and 45.12% correct respectively.

Recent work on graph cuts has produced both new graph architectures and energy minimization algorithms. Boykov and Komolgorov [14] have developed an approximate Ford-Fulkerson style augmenting paths algorithm, which they show to be much faster in practice than standard push-relabel approaches (by factors of 2 to 5 for the examples provided). Boykov et al. [13] propose *expansion move* and *swap move* algorithms that can simultaneously modify labels of arbitrarily large pixel sets. Komolgorov and Zabih [45] propose a graph architecture in which the vertices represent pixel correspondences (rather than pixels themselves) and impose uniqueness constraints to handle occlusions. These recent graph cut methods have been shown to be among the best performers in [69], and examples are shown in Figure 3-1.

3.2.4 Other Global Methods

While dynamic programming and more recently graph cuts have been the most often exploited energy minimization methods for global stereo matching, a number of other approaches have been used as well. Two of the most notable are nonlinear diffusion and belief propagation. Shah [71], Scharstein and Szeliski [68] and Mansouri [52] aggregate support using various models for non-uniform diffusion, rather than using fixed-size, rectangular windows. We discuss these methods more in section 4, focusing on their abilities to handle occlusion. Sun

et al. [77] cast the global matching problem in a Markov network framework and solve using belief propagation. This approach has been shown to yield results comparable to the best graph cut methods in [69].

Another class of global methods seeks to reconstruct a scene without explicitly establishing correspondences. Fua and Leclerc [30] model the scene as a mesh that is iteratively updated to minimize an objective function. Faugeras and Keriven [27] propose a similar method that models the scene using level sets. Kutulakos and Seitz [48] represent the scene as a volume and propose a space carving method to refine the surface. While these methods may be applied to binocular stereo, the object-centered representations are most powerful when exploiting constraints from multiple views (greater than two) of the scene to reduce sensitivity to view-dependent effects (e.g., occlusion and shading).

4. OCCLUSION

Much of the stereo research in the last decade has focused on detecting and measuring occlusion regions in stereo imagery and recovering accurate depth estimates for these regions. This section defines the occlusion problem in stereo vision and reviews three classes of algorithms for handling occlusion: methods that detect occlusion, methods that reduce sensitivity to occlusion and methods that model the occlusion geometry. Table 4-1 provides a summary of these approaches. A comparative analysis of some of these methods is also discussed.

4.1 The Occlusion Problem Defined

The occlusion problem in stereo vision refers to the fact that some points in the scene are visible to one camera but not the other, due to the scene and camera geometries. Figure 4-1

depicts two scenes, each with two points, one, P_V, being visible to both cameras and the other, P_O, visible to only one camera. We call the point P_O *half-occluded* because it is occluded in one of the views and not the other. While the depth at point P_V may be computed by stereopsis, the depth at P_O is inestimable, unless additional views are added on which the point is not occluded or assumptions are made about the scene geometry. The half-occlusion in the left example of Figure 4-1 is commonly observed in most scenes. The half-occlusion observed in the right example is less common, since narrow structures (e.g., fences) that might obstruct one's view are simply not observed in most scenes. We call these structures *narrow occluding objects*. Dhond and Aggarwal [24] define a narrow occluding object as an occluding object the width of which is narrower than (a) the region of support for matching or (b) the largest disparity difference between the foreground and the background objects. In the first case, matching is biased, since two distinct depths are competing within the same window. In the second case, the stereo ordering constraint fails. While the problem of narrow occlusion objects is more difficult than that of general occlusion, both are generally addressed similarly.

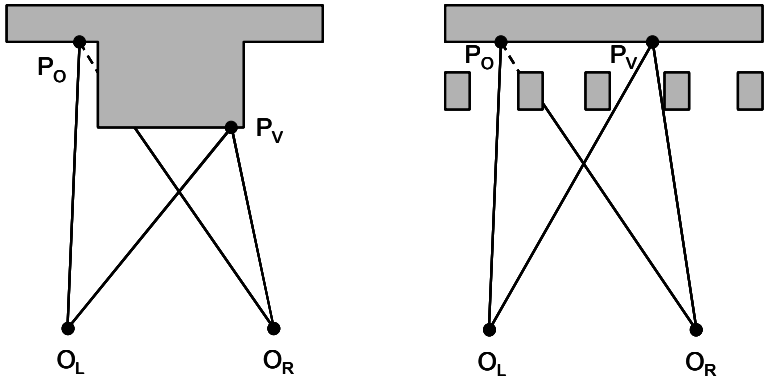

Figure 4-1 Examples of typical occlusion (left) and less common narrow occlusion (right) that may be observed when viewing a fence, for example. The points P_V in the examples are visible to both cameras, so their depths may be estimated by stereopsis. The half-occlusion points P_O in the examples are only visible in one image, so their depths may not be estimated.

Table 4-1 Occlusion methods

APPROACH	REFERENCES	BRIEF DESCRIPTION
METHODS THAT DETECT OCCLUSION		
Depth Map Discontinuities	[34], [86]	Discontinuities in the depth map are assumed to be occlusion regions.
Left-Right Matching	[15], [29]	Matches that are not unique when estimated from left-to-right and right-to-left are assumed to be in occlusion regions.
Ordering Constraint	[50], [74], [88]	Oppositely ordered adjacent matches indicate occlusion.
Intensity Edges	[11], [17], [60]	Intensity edges are assumed to correspond to occlusion boundaries.
METHODS THAT REDUCE SENSITIVITY TO OCCLUSION		
Robust Similarity Criterion	[7], [66], [70], [74], [89]	Robust methods are employed in the match metric to reduce sensitivity to occlusion.
Adaptive Regions of Support	[31], [39], [52], [68], [71], [90]	Regions of support are adaptively resized, reshaped or diffused to obtain the best match and minimize the effects of occlusion.
METHODS THAT MODEL OCCLUSION GEOMETRY		
Global Occlusion Modeling	[5], [10], [36], [65]	Occlusion is modeled and included in the match procedure, usually using dynamic programming.
Multiple Cameras	[58], [67]	Multiple cameras ensure that every point in the scene is visible by at least two cameras.
Active Vision	[16], [49], [61], [64]	The camera or stereo rig is moved in order to detect occlusion and to determine occlusion width.

4.2 Methods that Detect Occlusion

The simplest approaches to handling occlusion regions merely attempt to detect them either before or after matching. These regions are then either interpolated based on neighboring disparities to produce a dense depth map or simply not used for applications requiring only a sparse depth map. The most common approach of this type is to detect discontinuities in the depth map itself after matching. Median filters are commonly used to eliminate outliers in depth maps, which are often caused by occlusion regions. Wildes [86] detects discontinuities in both surface orientation and depth by comparing local histograms of these values with a Kolmogorov-Smirnov test. Hoff and Ahuja [34] detect depth and orientation discontinuities by fitting local planar patches to semicircular regions of the data. If two of these planes making up a circular region differ in depth or orientation by more than a threshold, there is evidence of occlusion.

Stereo match consistency may also be used to detect occlusion boundaries. Chang et al. [15] and Fua [29] compute two disparity maps, one based on the correspondence from the left image to the right and the other based on the correspondence from the right image to the left (see Figure 4-2). Inconsistent disparities are assumed to represent occlusion regions in the scene. There are, of course, other possible explanations for inconsistent matches, including perspective differences, non-uniform lighting and sensor noise. The left-right consistency check treats all of these phenomena the same, as sources of blunders to be replaced, typically by interpolation. The naïve implementation of left-right consistency checking doubles the number of computations for correlation. However, when correlation is implemented using running sums as in section 3.1.1, the additional consistency checks may be implemented with only additional bookkeeping and memory, not additional correlation computations [3]. Due to its simplicity and overall good performance at eliminating bad depth estimates, left-right consistency has been implemented in many real-time stereo systems [26, 46, 55].

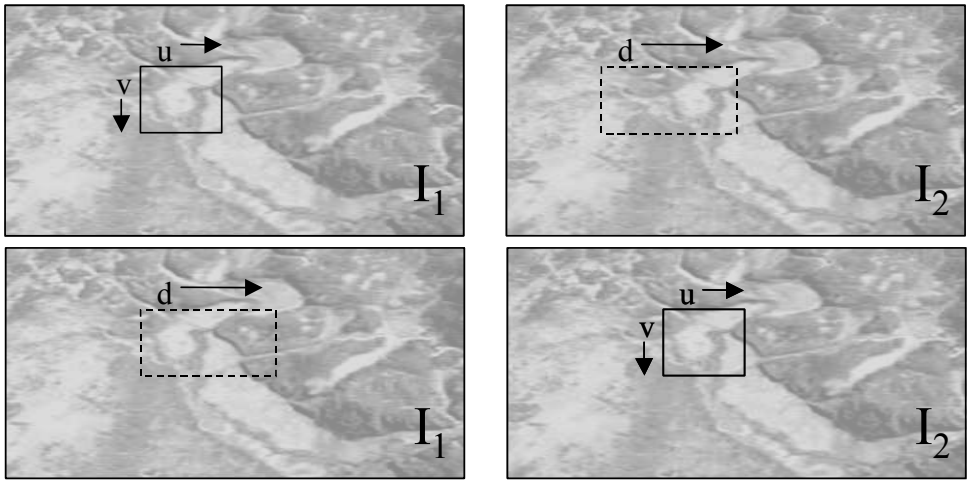

Figure 4-2 Left-right matching reverses the roles of the left (I_1) and right (I_2) images to obtain a consistency constraint that may be used to detect occlusion or other sources of mismatch.

The ordering constraint in stereo may also be used to detect occlusion. The relative ordering of points along a pair of stereo scan-lines is generally monotonic, assuming that there are no narrow occluding objects in the scene. Yuille and Poggio [88], Little and Gillett [50] and Silva and Santos-Victor [74] have proposed methods that check for out of order matches, which may indicate the presence of occlusion.

Another approach to detecting occlusion boundaries is based on the observation that depth and orientation discontinuities typically give rise to image intensity edges. Cochran and Medioni [17] incorporate an edge map into their post-filtering scheme. The disparity map is smoothed, keeping only the disparities associated with edges unaltered. Then, those points with large disparity differences between the original and smoothed versions are rejected as belonging to occlusion regions. Ohta and Kanade [60] match regions between edge segments using dynamic programming, thus avoiding the occlusion problem altogether. The method of Birchfield and Tomasi [11], discussed in section 3.1.3, also avoids the occlusion problem by matching segments delineated by intensity edges. The assumption that occlusion boundaries and intensity edges are coincident may also be used to model occlusion directly rather than avoid it. Methods that take this route are discussed in section 4.4.

4.3 Methods that Reduce Sensitivity to Occlusion

The use of robust methods is one way to reduce the sensitivity of matching to occlusion and other image differences (e.g., perspective differences and sensor noise). Sara and Bajcsy [66] propose a robust normalized cross-correlation algorithm based on robust covariance matrix estimation. Stewart [75] characterizes the behavior of a number of robust correlation measures near discontinuities (i.e., occlusion regions). These include least median of squares (LMS), least

trimmed squares (LTS), M-estimators, Hough transforms, RANSAC and MINPRAN. All of these methods provide some robustness to gross outliers. However, the presence of occlusion in a stereo image pair produces disparity discontinuities that are coherent. That is, while they are outliers to the structure of interest, they are inliers to a different structure. This coherence introduces a bias in robust estimates. Stewart calls these coherent outliers pseudo-outliers and provides suggestions for the careful selection of robust estimators to handle them.

Zabih and Woodfill [89] propose non-parametric transforms, rank and census, that are applied to image intensities before correlation (see section 3.1.1). Since these methods rely on relative ordering of intensities rather than the intensities themselves, they are somewhat robust to outliers. These rank measures, however, are sensitive to distortion due to perspective projection. To remedy this problem, Bhat and Nayar [7] propose an alternative ordinal measure based on the distance between rank value positions rather than the rank values themselves. This provides some robustness to perspective distortion. The fundamental limitation of all of these approaches is that information is lost at each step, making the discriminatory power of these methods low. Scherer et al. [70] propose a minor modification to Bhat and Nayar's ordinal coefficient, incorporating some of the information lost in the original method. They report significantly improved discriminatory power using this new coefficient. A comparison of rank and robust normalized cross-correlation for dealing with occlusion may be found in [66]. Their results indicate that rank correlation and robust normalized cross-correlation perform comparably and that both reduce the number of erroneous disparity estimates due to occlusion by about a factor of 2 for small window sizes (7x7 pixels) and significantly more for larger window sizes.

Another approach to reduce sensitivity to occlusion is to adaptively resize the window size and shape in order to optimize the match similarity near occlusion boundaries. Kanade and

Okutomi [39] propose an iterative method for determining the window size. The window size is initialized to be very small, and the match uncertainty is computed. The window size is then expanded by one pixel in each direction independently, and the uncertainty is computed. If the uncertainty increases when the window is expanded in a given direction, then that direction is prohibited from expanding. This procedure is applied iteratively until all directions become prohibited. As occluding boundaries need not appear as vertical edges in an image, rectangular windows cannot provide maximum local support. To handle the more general case, Shah [71] and Scharstein and Szeliski [68] propose using non-uniform diffusion (i.e., weighted local aggregation of support rather than a fixed rectangular window) to expand the support region. Mansouri et al. [52] propose a similar approach, but allow the diffusion equations to be anisotropic near intensity edges to better localize depth discontinuities. An interesting approach by Zitnick and Kanade [90] combines diffusion of the support region with inhibition of support for pixels along similar lines of sight. This ensures that the diffusion process does not violate the uniqueness constraint. Fusiello et al. [31] propose a method based on multiple windows. This method considers nine correlation windows centered about the desired point and takes the estimated disparity with the minimum error. The window associated with this disparity is likely to cover a relatively constant depth region. Note that this does not require any additional correspondence computations and that it effectively blurs less certain regions of the depth map, limiting the achievable localization. Of course, in addition to improving stereo matching performance near occlusion boundaries, all of these methods help reduce erroneous estimates in small image regions with little texture by providing a more sufficient region of support.

4.4 Methods that Model Occlusion Geometry

While the above methods for detecting and reducing sensitivity to occlusion each offer some benefit and most are computationally affordable, they do not take advantage of all available search constraints. It is desirable to integrate knowledge of the occlusion geometry itself into the search process. One framework within which this may be done simply is global search. Belhumeur [5] defines priors for a series of Bayesian estimators, each handling a more complicated model of the world. These are used to define cost functions for dynamic programming. The first and simplest model assumes surface smoothness, and the dynamic programming complexity is simply $O(ND^2)$, where N is the number of pixels in the scan-line and D is the disparity range. The second model assumes object boundaries in addition to surface smoothness and its implementation also has complexity $O(ND^2)$. Belhumeur's third and most realistic model of the world includes terms for surface slope and creases in addition to object boundaries and surface smoothness and has complexity $O(ND^2Q^2)$, where D is the disparity range and Q is the range of slope values modeled.

Variations of Belhumeur's models, particularly the second model, have been used within both dynamic programming (see 3.2.1) and graph cut (see 3.2.3) frameworks for determining optimal disparity maps. Birchfield and Tomasi [10] have implemented a dynamic programming algorithm with an occlusion boundary cost and pruning to reduce the computational complexity to $O(NDlogD)$. Occlusion boundaries are assumed to lie on intensity edges. The lower the cost associated with an occlusion boundary, the more depth discontinuities are facilitated. The higher the cost, the smoother the disparity maps. Intille and Bobick [36] reduced the cost of diagonal edge segments associated with orientation occlusions to zero, while enforcing the usual occlusion cost for vertical segments. The graph cut methods discussed in section 3.2.3 also make

similar assumptions and include similar occlusion costs.

Another method of detecting occlusion regions and recovering depth in those regions is to exploit multiple cameras. Kanade et al. [38] propose a multiple baseline algorithm exploiting a translating camera. By measuring disparity in terms of inverse distance, SSD metrics for multiple stereo pairs with variable but known baselines may be integrated easily. Ambiguity in one stereo pair due to occlusion is not present in another stereo pair in which the given point is jointly visible. Thus, by summing the SSD metrics over all stereo pairs, the effects of these ambiguities are significantly reduced. Nakamura et al. [58] and Satoh and Ohta [67] propose a similar approach in what they call stereo by eye array (SEA), but they directly model occlusion in an attempt to remove its effect altogether. They define occlusion masks based on estimated probabilities of occlusion patterns observable by a square nine-camera configuration. These masks are hypothesized as occlusion configurations for each point to be matched in all stereo camera images, and the error is minimized over the masks to determine the correct configuration. Thus, only those images in which a point is visible are used to determine that point's disparity.

Active vision may also be exploited to detect occlusion regions and to recover depth in those regions by moving the camera (for motion stereo) or stereo rig so as to bring the occluded point into joint view. Ching [16] exploits active vision to discriminate between occlusion and specular highlights. One of the two cameras for stereo is rotated, and the change in width of the suspected occlusion region is examined. Those regions that do not grow or shrink predictably are assumed to be due to specular highlights. Li and Chin [49] propose a method for detecting occlusion in a sequence of images captured by a translating camera. Occlusion is predicted for a frame or set of frames based on the relative rates of position change between edges in stereo image pairs as the camera translates. The cases of occlusion at the beginning of the sequence, in

the middle of the sequence and at the end of the sequence are considered. Pilon and Cohen [61] propose an active stereo method for a translating stereo rig. At a given time instance, the stereo rig is translated such that one of the cameras from the previous time instance is roughly centered between the current camera positions. The image from the roughly centered camera is taken to be a reference, and it is assumed that occlusions between the two cameras at the new time instance and the reference image are small. Note that this is equivalent to mounting a third camera to the rig. Each current image is mapped to the reference using optical flow and then projected into the other camera's coordinates. Occlusion regions are detected as position jumps between views. The depths for these occlusion regions are then recovered based on the jump lengths as measured from two independent positions, again requiring active camera motion. A more general approach to handling occlusion via active vision is proposed by Reed and Allen [64], who describe a sensor planning system that minimizes the number of views required to capture all points in the scene. A detailed discussion of sensor planning is beyond the scope of this paper. A review of this topic is provided in Tarabanis et al. [78].

4.5 Comparisons

Relatively little has been done to compare the variety of algorithms proposed for detecting, reducing sensitivity to or measuring occlusions. Egnal and Wildes [25] recently conducted an analysis of four of the simpler approaches. All of these fall into the *detecting occlusion* class of algorithms discussed above. The four methods compared are *bimodality*, *match goodness jumps*, *left-right checking* and *order checking*. Bimodality measures a ratio of disparity peaks in a small window. Occlusions are hypothesized for points for which the ratio is large. This is similar to the disparity discontinuity method described above. Match goodness jumps measure sharp drops

in the match metric (e.g., correlation score). Points with low scores that are surrounded by points with high scores are assumed to be occlusions. Left-right checking measures the consistency of matching in one direction versus the other and is discussed above. Inconsistency is taken to imply occlusion. Order checking determines whether or not neighboring points violate the epipolar ordering constraint described above. If so, an occlusion is assumed to be present. The first two methods were found to predict occlusion points well, but they also falsely detect a large number of matchable points. The second two approaches, particularly left-right checking, reliably predicted occlusion points without a large number of false detects. None of these methods is capable of determining whether occlusion is the source of the error or if it is something else (e.g., image noise, lack of texture, etc.). For many applications, this distinction is irrelevant. However, when precise knowledge of occlusion is desired, model-based methods like those described above may be more suitable. The comparative study of Egnal and Wildes [25] is a good first step toward evaluating occlusion methods. A more comprehensive study is needed that includes many of the methods from all three classes (detecting occlusion, reducing sensitivity to occlusion and modeling occlusion directly) discussed above.

5. REAL-TIME STEREO IMPLEMENTATIONS

In the past decade, real-time dense disparity map stereo (30 frames per second or faster) has become a reality, making the use of stereo processing feasible for a variety of applications, some of which are discussed in the next section. Until very recently, all truly real-time implementations made use of special purpose hardware, like digital signal processors (DSP) or field programmable gate arrays (FPGA). However, with ever increasing clock speeds and the integration of single instruction multiple data (SIMD) co-processors (e.g., Intel MMX) into

general-purpose computers, real-time stereo processing is finally a reality for common desktop computers. This section reviews the progression of real-time stereo implementations over the past decade. A summary of real-time stereo systems and their comparative performances is provided in Table 5-1. Timing estimates have been extracted from previous reports [18, 41, 46].

In 1993, Faugeras et al. reported on a stereo system developed at INRIA and implemented for both DSP and FPGA hardware [26]. They implemented normalized correlation efficiently using the method discussed briefly in section 3.1.1 and included left/right matching for consistency checking. They used a right-angle trinocular stereo configuration, computing two depth maps and then merging them to enforce joint epipolar constraints. The DSP implementation exploited the MD96 board [53], which was made up of 4 Motorola 96002 DSPs. The FPGA implementation was designed for the PeRLe-1 board, which was developed at DEC-PRL and was composed of 23 Xilinx logic cell arrays (LCA). The algorithms were also implemented in C for a Sparc 2 workstation. While much more difficult to program, the FPGA implementation outperformed the DSP implementation by a factor of 34 and the Sparc 2 implementation by a factor of 210, processing 256x256 pixel images at approximately 3.6 fps.

Also in 1993, Nishihara reported on a stereo system based on the PRISM-3 board developed by Teleos Research [59]. This system used Datacube digitizer hardware, custom convolver hardware and the PRISM-3 correlator board, which makes extensive use of FPGAs. For robustness and efficiency, this system used area correlation of the sign bits after applying a Laplacian of Gaussian filter to the images. Konolige also reports on the performance of a PC implementation of these algorithms by Nishihara in 1995 [46]. This system was capable of 0.5 fps with 320x240 pixel images.

Table 5-1 Real-time stereo implementations

REAL-TIME SYSTEM	IMAGE SIZE	FRAME RATE	RANGE BINS	METHOD	PROCESSOR	CAMERAS
INRIA 1993	256x256	3.6 fps	32	Normalized Correlation	PeRLe-1	3
CMU iWarp 1993	256x240	15 fps	16	SSAD	64 Processor iWarp Computer	3
Teleos 1995	320x240	0.5 fps	32	Sign Correlation	Pentium 166 MHz	2
JPL 1995	256x240	1.7 fps	32	SSD	Datacube & 68040	2
CMU Stereo Machine 1995	256x240	30 fps	30	SSAD	Custom HW & C40 DSP Array	6
Point Grey Triclops 1997	320x240	6 fps	32	SAD	Pentium II 450 MHz	3
SRI SVS 1997	320x240	12 fps	32	SAD	Pentium II 233 MHz	2
SRI SVM II 1997	320x240	30+ fps	32	SAD	TMS320C60x 200MHz DSP	2
Interval PARTS Engine 1997	320x240	42 fps	24	Census Matching	Custom FPGA	2
CSIRO 1997	256x256	30 fps	32	Census Matching	Custom FPGA	2
SAZAN 1999	320x240	20 fps	25	SSAD	FPGA & Convolvers	9
Point Grey Triclops 2001	320x240	20 fps 13 fps	32	SAD	Pentium IV 1.4 GHz	2 3
SRI SVS 2001	320x240	30 fps	32	SAD	Pentium III 700 MHZ	2

One system came close to achieving frame-rate in 1993. Webb implemented the multi-baseline stereo algorithms of Kanade et al. on the CMU Warp machine [21, 38, 85]. Three images were used for this system. For efficiency, the sum of SAD (SSAD) was implemented. 64 iWarp processors were used to achieve 15 fps with 256x240 pixel images.

In 1995, Matthies et al. reported on a real-time stereo system developed at the Jet Propulsion Laboratory (JPL) using a Datacube MV-200 image processing board and a 68040 CPU board [55]. This performed SSD matching on a Laplacian (equivalently, difference of Gaussians) image pyramid to determine disparities. Left/right matching was used for consistency checking. This system was capable of processing approximately 1.7 fps with

256x240 pixel images. The application of this system to obstacle detection for unmanned ground vehicles may be found in [55]. A complete discussion of their algorithm development and details on an earlier version of the system is provided in [54].

Also in 1995, Kimura et al. reported on a video-rate stereo machine developed at CMU [41]. This was the first published stereo system capable of 30 fps (with 256x240 pixel images). Like the iWarp implementation at CMU, this system also exploited multi-baseline stereo to improve depth estimates. The prototype system was equipped with six cameras. Also like the iWarp implementation, the SSAD was implemented for efficiency. These algorithms were implemented on custom hardware and an array of 8 C40 DSPs. The CMU video-rate stereo machine has been used for a variety of applications, including virtual reality and z keying [40].

In 1997, Konolige reported on a real-time stereo system developed at SRI International [46]. The SRI Small Vision Module (SVM) was designed to operate at low power and in a small package, as opposed to the large custom hardware arrays previously developed for stereo processing. The original SVM consisted of two CMOS 320x240 grayscale imagers and lenses, low power A/D converters, a DSP (ADSP 2181, running at 33 MHz) and a small flash memory, all on a 2" by 3" circuit board. The second generation SVM, or SVM II, uses a newer Texas Instruments DSP (TMS320C60x) than runs at 200 MHz and outperforms the ADSP 2181 by a factor of 30. SVM II is capable of processing 320x240 pixel images at greater than 30 fps. In addition to the SVM, SRI has implemented a version of their algorithms in C for Pentium microprocessors with MMX, called the Small Vision System (SVS). The SVS is capable of processing 320x240 pixel images at 12 fps on a 233 MHz Pentium II. Both the SVM II and the SVS use SAD on LoG transformed image pixels to determine the disparities and left/right checking for post-filtering. The SVS has been used for real-time tracking [6] and is distributed

commercially by Videre Design. Another commercial product for real-time stereo is the Point Grey Digiclops trinocular stereo vision system. The Triclops software development kit (SDK) bundled with it as of 1997 performed stereo at 6 fps for 320x240 pixel images on a 450 MHz Pentium II, as reported in [42]. The algorithms are similar to those used for SRI's SVS. Both of these commercial systems provide sub-pixel interpolation. Videre Design claims that SVS provides 1/16 pixel interpolation, while Point Grey claims that their Digiclops provides 1/256 pixel interpolation. Note that the precision of any sub-pixel estimate depends greatly on image quality and content and that the amount of interpolation required varies by application and by camera selection. The required interpolation s for a given range resolution is defined by

$$s = \frac{Tf\Delta Z}{pZ^2 - pZ\Delta Z}, \tag{5-1}$$

where p is the pixel size, Z is the range to the object imaged, T is the baseline, f is the focal length, ΔZ is the desired range resolution, and all measurements are in millimeters. For example, with a 7.5 μm pixel size, a 5 m range, a 90 mm baseline, 4.8 mm focal length, and a desired range resolution of 20 mm, the required sub-pixel interpolation would be 1/22. In general, it is difficult to achieve sub-pixel precision better than 1/10 pixel for arbitrary scenes.

Two new real-time stereo systems exploiting FPGAs were also developed in 1997. Woodfill and Von Herzen [87] implemented census matching [89] for stereo on the custom PARTS engine developed at Interval Research Corporation. The PARTS engine is made up of 16 Xilinx 4025 FPGAs and fits on a standard PCI card. It is capable of processing 320x240 pixel images at 42 fps. Corke and Dunn [18] also implemented census matching on their Configurable Logic Processors (CLP), VME bus circuit boards made up of several FPGAs each. Their system is capable of processing 256x256 pixel images at 30 fps.

In 1999, Kimura et al. reported on a nine camera stereo machine called SAZAN [42]. Like

the CMU stereo machine and iWarp implementations, this system is also based on the multi-baseline stereo algorithm. In addition to the added robustness that multiple cameras provide, low-pass filtering local regions of the individual SAD surfaces further resolve matching ambiguities. The name SAZAN, a Japanese term for 3x3 multiplication, is a reference to the significant amount of convolution used in this system. SAZAN is implemented with linear shift invariant (LSI) digital filters for LoG filtering and Gaussian smoothing and FPGAs for similarity comparisons. It is capable of processing 320x240 pixel images at 20 fps.

With ever-increasing clock speeds and the integration of SIMD co-processors into general-purpose computers, real-time stereo processing is finally a reality for common desktop computers. Point Grey's Triclops now runs at 20 fps for 320x240 pixel images on a 1.4 GHz Pentium IV machine. Likewise, SRI's SVS now runs at 30 fps for 320x240 pixel images on a 700 MHz Pentium III. With inexpensive and compact real-time systems like these now commercially available, many applications that previously were impractical (e.g., tracking [6, 22] and virtual reality [40]) are now being intensely explored.

6. CONCLUSIONS

After roughly thirty years of research on computational stereo (in the computer vision community), many elements of stereo algorithms are well understood. In particular, accurate stereo calibration and efficient algorithms for local correspondence are now well understood. As a result, during the past decade, we have seen the focus turn from the fundamentals of stereopsis to more difficult problems, such as global correspondence and methods for handling occlusion. However, a comprehensive evaluation and comparison of these more advanced algorithms (and even some standard correspondence methods) has yet to be done. One of our goals in this review

has been to consolidate existing quantitative results and comparative analyses and to suggest analyses remaining to be done. We believe that much of the stereo work in the coming decade should and will be bolstered by more complete quantitative performance evaluations. The recent article by Scharstein and Szeliski [69] is a promising first step.

Perhaps the most practically significant advance in the last decade has been the appearance of real-time stereo systems, first on special-purpose hardware and more recently on general-purpose computers. Due in large part to these real-time implementations, research on real-time stereo applications has blossomed in the latter part of this decade. Real-time algorithms, however, are still relatively simplistic, and most of the global matching and occlusion handling methods discussed do not currently run in real-time. More demanding potential applications (e.g., virtual reality) require both real-time algorithms and very precise, reliable and dense depth estimates. Solving the problems of stereo timing, precision and reliability jointly remains a challenging research topic that we predict will see progress in the coming decade.

REFERENCES

[1] P. Aschwanden and W. Guggenbuhl, "Experimental Results from a Comparative Study on Correlation-Type Registration Algorithms," In *Robust Computer Vision* (Forstner and Ruwiedel, Eds.), Wickmann, pp. 268-289, 1993.

[2] H. H. Baker, "Depth from Edge and Intensity Based Stereo," *Technical Report AIM-347*, Stanford University Artificial Intelligence Laboratory, 1982.

[3] J. Banks and P. Corke, “Quantitative Evaluation of Matching Methods and Validity Measures for Stereo Vision,” *Int’l J. Robotics Research*, vol. 20, no. 7, 2001.

[4] S. T. Barnard and M. A. Fischler, "Computational Stereo," *ACM Computing Surveys*, vol.

14, pp. 553-572, 1982.

[5] P. N. Belhumeur, "A Bayesian Approach to Binocular Stereopsis," *Int'l J. Computer Vision*, vol. 19, no. 3, pp. 237-260, 1996.

[6] D. Beymer and K. Konolige, "Real-Time Tracking of Multiple People Using Continuous Detection," *Proc. IEEE Frame Rate Workshop*, 1999.

[7] D. N. Bhat and S. K. Nayar, "Ordinal Measures for Image Correspondence," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, pp. 415-423, 1998.

[8] F. Bigone, O. Henricsson, P. Fua and M. Stricker, "Automatic Extraction of Generic House Roofs from High Resolution Aerial Imagery," *Proc. European Conf. Computer Vision*, pp. 85-96, 1996.

[9] S. Birchfield and C. Tomasi, "Depth Discontinuities by Pixel-to-Pixel Stereo," *Technical Report STAN-CS-TR-96-1573*, Stanford University, 1996.

[10] S. Birchfield and C. Tomasi, "Depth Discontinuities by Pixel-to-Pixel Stereo," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 1073-1080, 1998.

[11] S. Birchfield and C. Tomasi, "Multiway Cut for Stereo and Motion with Slanted Surfaces," *Proc. Int'l Conf. Computer Vision*, vol. 1, pp. 489-495, 1999.

[12] Y. Boykov, O. Veksler and R. Zabih, "Markov Random Fields with Efficient Approximations," *Proc. Computer Vision and Pattern Recognition*, pp. 648-655, 1998.

[13] Y. Boykov, O. Veksler and R. Zabih, "Fast Approximate Energy Minimization via Graph Cuts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222-1239, 2001.

[14] Y. Boykov and V. Komolgorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," *Proc. Third Int'l Workshop on Energy*

Minimization Methods in Computer Vision and Pattern Recognition, 2001.

[15] C. Chang, S. Chatterjee, and P. R. Kube, "On an Analysis of Static Occlusion in Stereo Vision," *Proc. Computer Vision and Pattern Recognition*, pp. 722-723, 1991.

[16] W.-S. Ching, "A New Method of Identifying Occlusion and Specular Highlights Using Active Vision," *Proc. Int'l Symp. Speech, Image Processing and Neural Networks*, pp. 437-440, 1994.

[17] S. D. Cochran and G. Medioni, "3-D Surface Description from Binocular Stereo," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 981-994, 1992.

[18] P. Corke and P. Dunn, "Real-Time Stereopsis Using FPGAs," *Proc. IEEE TENCON - Speech and Image Technologies for Computing and Telecom.*, pp. 235-238, 1997.

[19] I. J. Cox, S. L. Hingorani, S. B. Rao, and B. M. Maggs, "A Maximum Likelihood Stereo Algorithm," *Computer Vision and Image Understanding*, vol. 63, pp. 542-567, 1996.

[20] T. H. Cormen, C. E. Leiserson and R. L. Rivest, *Introduction to Algorithms*, McGraw-Hill, New York, 1990.

[21] J. D. Crisman and J. A. Webb, "The Warp Machine on Navlab," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 5, pp. 451-465, 1991.

[22] T. Darrell, G. Gordon, M. Harville, and J. Woodfill, "Integrated Person Tracking using Stereo, Color, and Pattern Detection," *Proc. Computer Vision and Pattern Recognition*, pp. 601-608, 1998.

[23] U. R. Dhond and J. K. Aggarwal, "Structure from Stereo - A Review," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 19, pp. 1489-1510, 1989.

[24] U. R. Dhond and J. K. Aggarwal, "Analysis of the Stereo Correspondence Process in Scenes with Narrow Occluding Objects," *Proc. Int'l Conf. Pattern Recognition*, vol. 1, pp.

470-473, 1992.

[25] G. Egnal and R. P. Wildes, "Detecting Binocular Half-Occlusions: Empirical Comparisons of Four Approaches," *Proc. Computer Vision and Pattern Recognition*, vol. 2, pp. 466-473, 2000.

[26] O. Faugeras, B. Hotz, H. Matthieu, T. Vieville, Z. Zhang, P. Fua, E. Theron, L. Moll, G. Berry, J. Vuillemin, P. Bertin, and C. Proy, "Real Time Correlation-Based Stereo: Algorithm, Implementations and Applications," *INRIA Technical Report 2013*, 1993.

[27] O. Faugeras and R. Keriven, "Variational Principles, Surface Evolution, PDE's, Level Set Methods, and the Stereo Problem," *IEEE Trans. Image Processing*, vol. 7, pp. 336-344, 1998.

[28] O. Faugeras and Q.-T. Luong, *The Geometry of Multiple Images*, The MIT Press, Cambridge, Massachusetts, 2001.

[29] P. Fua, "A Parallel Stereo Algorithm that Produces Dense Depth Maps and Preserves Image Features," *Machine Vision and Applications*, vol. 6, pp. 35-49, 1993.

[30] P. Fua and Y. G. Leclerc, "Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading," *Int'l J. Computer Vision*, vol. 16, pp. 35-56, 1995.

[31] A. Fusiello, V. Roberto, and E. Trucco, "Efficient Stereo with Multiple Windowing," *Proc. Computer Vision and Pattern Recognition*, pp. 858-863, 1997.

[32] A. Gruen, "Adaptive Least Squares Correlation: A Powerful Image Matching Technique," *South African Journal of Photogrammetry, Remote Sensing and Cartography*, vol. 3, no. 14, pp. 175-187, 1985.

[33] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, United Kingdom, 2000.

[34] W. Hoff and N. Ahuja, "Surfaces from Stereo: Integrating Feature Matching, Disparity Estimation, and Contour Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 2, pp. 121-136, 1989.

[35] B. K. P. Horn and B. G. Schunk, "Determining Optical Flow," *Artificial Intelligence*, vol. 17, pp. 185-204, 1981.

[36] S. S. Intille and A. F. Bobick, "Incorporating Intensity Edges in the Recovery of Occlusion Regions," *Proc. Int'l Conf. Pattern Recognition*, vol. 1, pp. 674-677, 1994.

[37] M. Irani, B. Rousso, and S. Peleg, "Computing Occluding and Transparent Motions," *Int'l J. Computer Vision*, vol. 12, pp. 5-16, 1994.

[38] T. Kanade, M. Okutomi, and T. Nakahara, "A Multiple-baseline Stereo Method," *Proc. ARPA Image Understanding Workshop*, pp. 409-426, 1992.

[39] T. Kanade and M. Okutomi, "A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, September 1994.

[40] T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka, "A Stereo Method for Video-Rate Dense Depth Mapping and Its New Applications," *Proc. Computer Vision and Pattern Recognition*, 1996.

[41] S. Kimura, T. Kanade, H. Kano, A. Yoshida, E. Kawamura, and K. Oda, "CMU Video-Rate Stereo Machine," *Proc. Mobile Mapping Symp.*, 1995.

[42] S. Kimura, T. Shinbo, H. Yamaguchi, E. Kawamura, and K. Naka, "A Convolver-Based Real-Time Stereo Machine (SAZAN)," *Proc. Computer Vision and Pattern Recognition*, vol. 1, pp. 457-463, 1999.

[43] R. Klette, K. Schluns and A. Koschan, *Computer Vision – Three Dimensional Data from*

Images, Springer, Singapore, 1998.

[44] V. S. Kluth, G. W. Kunkel, and U. A. Rauhala, "Global Least Squares Matching," *Proc. Int'l Geoscience and Remote Sensing Symp.*, vol. 2, pp. 1615-1618, 1992.

[45] V. Komolgorov and R. Zabih, "Computing Visual Correspondence with Occlusions using Graph Cuts," *Proc. Int'l Conf. Computer Vision*, 2001.

[46] K. Konolige, "Small Vision Systems: Hardware and Implementation," *Proc. Eighth Int'l Symp. Robotics Research*, 1997.

[47] A. Koschan, "What is New in Computational Stereo Since 1989: A Survey of Current Stereo Papers," *Technical Report 93-22, Technical University of Berlin*, 1993.

[48] K. N. Kutulakos and S. M. Seitz, "A Theory of Shape by Space Carving," *Int'l J. Computer Vision*, vol. 38, no. 3, pp. 199-218, 2000.

[49] Z.-N. Li and H. W. Chin, "Depth and Occlusion Recovery in Motion Stereo," *Proc. IEEE Int'l Conf. Systems, Man and Cybernetics*, vol. 5, pp. 3890-3895, 1995.

[50] J. J. Little and W. E. Gillett, "Direct Evidence for Occlusion in Stereo and Motion," *Image and Vision Computing*, vol. 8, no. 4, pp. 328-340, 1990.

[51] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Proc. Int'l Joint Conf. A. I.*, pp. 674-679, 1981.

[52] A.-R. Mansouri, A. Mitiche and J. Konrad, "Selective Image Diffusion: Application to Disparity Estimation," *Proc. Int'l Conf. Image Processing*, vol. 3, pp. 284-288, 1998.

[53] H. Mathieu, "A Multi-DSP 96002 Board," *INRIA Technical Report 153*, 1993.

[54] L. Matthies, "Stereo Vision for Planetary Rovers: Stochastic Modeling to Near Real-Time Implementation," *Int'l J. Computer Vision*, vol. 8, no. 1, pp. 71-91, 1992.

[55] L. Matthies, A. Kelly, T. Litwin and G. Tharp, "Obstacle Detection for Unmanned Ground

Vehicles: A Progress Report," *Proc. Intelligent Vehicles '95 Symp.*, pp. 66-71, 1995.

[56] M. J. McDonnell, "Box-Filtering Techniques," *Computer Graphics and Image Processing*, vol. 17, pp. 65-70, 1981.

[57] K. Muhlmann, D. Maier, J. Hesser, and R. Manner, "Calculating Dense Disparity Maps from Color Stereo Images, an Efficient Implementation," *Proc. IEEE Workshop on Stereo and Multi-Baseline Vision*, pp. 30-36, 2001.

[58] Y. Nakamura, T. Matsuura, K. Satoh, and Y. Ohta, "Occlusion Detectable Stereo - Occlusion Patterns in Camera Matrix," *Proc. Computer Vision and Pattern Recognition*, pp. 371-378, 1996.

[59] H. K. Nishihara, "Real-Time Stereo- and Motion-Based Figure-Ground Discrimination and Tracking using LOG Sign-Correlation," *Proc. Twenty-Seventh Asilomar Conf. Signals, Systems and Computers*, pp. 95-100, 1993.

[60] Y. Ohta and T. Kanade, "Stereo by Intra- and Intra-Scanline Search Using Dynamic Programming," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 7, pp. 139-154, 1985.

[61] M. Pilon and P. Cohen, "Occlusion Detection and Interpretation Based on Image Reprojection," *Proc. S. W. Symp. Image Analysis and Interpretation*, pp. 166-171, 1996.

[62] S. Randriamasy and A. Gagalowicz, "Region Based Stereo Matching Oriented Image Processing," *Proc. Computer Vision and Pattern Recognition*, pp. 736-737, 1991.

[63] U. A. Rauhala, "Introduction to Array Algebra," *Photogrammetric Engineering and Remote Sensing*, vol. 46, no. 2, pp. 177-192, 1980.

[64] M. K. Reed and P. K. Allen, "Constraint-Based Sensor Planning for Scene Modeling," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1460-1467, 2000.

[65] S. Roy and I. J. Cox, "A Maximum-Flow Formulation of the N-camera Stereo Correspondence Problem," *Proc. Int'l Conf. Computer Vision*, pp. 492-499, 1998.

[66] R. Sara and R. Bajcsy, "On Occluding Contour Artifacts in Stereo Vision," *Proc. Computer Vision and Pattern Recognition*, pp. 852-857, 1997.

[67] K. Satoh and Y. Ohta, "Occlusion Detectable Stereo - Systematic Comparison of Detection Algorithms," *Proc. Int'l Conf. Pattern Recognition*, pp. 280-286, 1996.

[68] D. Scharstein and R. Szeliski, "Stereo Matching with Non-Linear Diffusion," *Int'l J. Computer Vision*, vol. 28, no. 2, pp. 155-174, 1998.

[69] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *Int'l J. Computer Vision*, vol. 47, no. 1, pp. 7-42, 2002.

[70] S. Scherer, P. Werth, and A. Pinz, "The Discriminatory Power of Ordinal Measures - Towards a New Coefficient," *Proc. Computer Vision and Pattern Recognition*, vol. 1, pp. 76-81, 1999.

[71] J. Shah, "A Nonlinear Diffusion Model for Discontinuous Disparity and Half-Occlusions in Stereo," *Proc. Computer Vision and Pattern Recognition*, pp. 34-40, 1993.

[72] C. Schmid and A. Zisserman, "The Geometry and Matching of Curves in Multiple Views," *Proc. European Conf. Computer Vision*, pp. 104-118, 1998.

[73] J. Shi and C. Tomasi, "Good Features to Track," *Proc. Computer Vision and Pattern Recognition*, pp. 593-600, 1994.

[74] C. Silva and J. Santos-Victor, "Intrinsic Images for Dense Stereo Matching with Occlusions," *Proc. European Conf. Computer Vision*, pp. 100-114, 2000.

[75] C. V. Stewart, "Bias in Robust Estimation Caused by Discontinuous and Multiple Structures," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 8, pp.

818-833, 1997.

[76] C. Sun, "A Fast Stereo Matching Method," *Proc. Digital Image Computing: Techniques and Applications (Auckland, New Zealand)*, pp. 95-100, 1997.

[77] J. Sun, H. –Y. Shum and N. –N. Zheng, "Stereo Matching Using Belief Propagation," *Proc. European Conf. Computer Vision*, pp. 510-524, 2002.

[78] K. A. Tarabanis, P. K. Allen and R. Y. Tsai, "A Survey of Sensor Planning in Computer Vision," *IEEE Trans. Robotics and Automation*, vol. 11, no. 1, pp. 86-104, 1995.

[79] I. Thomos, S. Malasiotis, and M. G. Strintzis, "Optimized Block Based Disparity Estimation in Stereo Systems Using a Maximum-Flow Approach," *Proc. SIBGRAPI '98 Conf.*, 1998.

[80] C. Tomasi and R. Manduchi, "Stereo Without Search," *Technical Report STAN-CS-TR-95-1543*, Stanford University, 1995.

[81] C. Tomasi and R. Manduchi, "Stereo Matching as a Nearest-Neighbor Problem," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, pp. 333-340, 1998.

[82] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*, Prentice Hall, New Jersey, 1998.

[83] C.-J. Tsai and A. K. Katsaggelos, "Dense Disparity Estimation with a Divide-and-Conquer Disparity Space Image Technique," *IEEE Trans. Multimedia*, vol. 1, pp. 18-28, 1999.

[84] V. Venkateswar and R. Chellappa, "Hierarchical Stereo and Motion Correspondence Using Feature Groupings," *Int'l J. Computer Vision*, vol. 15, pp. 245-269, 1995.

[85] J. A. Webb, "Implementation and Performance of Fast Parallel Multi-Baseline Stereo Vision," *Proc. DARPA Image Understanding Workshop*, pp. 1005-1012, 1993.

[86] R. Wildes, "Direct Recovery of Three-Dimensional Scene Geometry from Binocular Stereo

Disparity," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 761-774, 1991.

[87] J. Woodfill and B. Von Herzen, "Real-Time Stereo Vision on the PARTS Reconfigurable Computer," *Proc. IEEE Workshop on FPGAs for Custom Computing Machines*, pp. 242-250, 1997.

[88] A. L. Yuille and T. Poggio, "A Generalized Ordering Constraint for Stereo Correspondence," *A. I. Laboratory Memo 777*, MIT, Cambridge, MA, 1984.

[89] R. Zabih and J. Woodfill, "Non-Parametric Local Transforms for Computing Visual Correspondence," *Proc. 3rd European Conf. Computer Vision*, pp. 150-158, 1994.

[90] C. L. Zitnick and T. Kanade, "A Cooperative Algorithm for Stereo Matching and Occlusion Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 7, 2000.

[91] Z. Zhang, "Determining the Epipolar Geometry and its Uncertainty: A Review," *INRIA Technical Report 2927*, 1996.

[92] H. Zhao, "Global Optimal Surface from Stereo," *Proc. Int'l Conf. Pattern Recognition*, vol. 1, pp. 101-104, 2000.