3.12

Local and Global Stereo Methods

Yang Liu and
J.K. Aggarwal
The University of Texas at Austin

1 Introduction

Computational stereo vision is one of the core topics in image analysis and computer vision. The purpose of every kind of vision system is to understand the outside world through the interpretation of the images formed, either in the eyes or in cameras, by photons received from many directions. Humans have two frontal-parallel eyes that see the world from two slightly different positions. These views are combined to recover the 3D information about the environment. It is intriguing that evolution has created this particular structure and the systems that support it. Can we do the same thing with computers? In the past 30 years, a lot of effort has been devoted to this area by various researchers, and it has become one of the most fundamental and fruitful areas of computer vision. Generally speaking, there are two directions of exploration for stereo vision. The first is that of neurophysiologic approach, which concentrates on investigating biologic vision systems by learning the functions of the neural pathway and visual cortical cells. Since Hubel and Wiesel's milestone paper on the receptive field of V1 cortical cells [12], various kinds of models have been presented trying to explain the function of stereoscopic vision. The other direction is of computer science, which focusses the implementation of stereo vision using computer hardware and software. Computer scientists aren't focussed on the biologic feasibility of a model, as the capabilities of the human vision system are too general for today's limited computing power. The problem is how to program a computer to estimate depth from digitized images of the world. While the solution does not have to be biologically motivated by the human vision system, the purpose of computational stereo algorithms is to infer depth from images taken from different known locations and to use geometry to recover the three dimensional scene. There is often overlap between these two directions of study. Sometimes a breakthrough in one area significantly affects research in the other area. For instance, Marr and Poggio accelarated the study of computational stereo in [16], based on physiologic studies of the human visual system. Also, Ohzawa [19], DeAngelis [8], and Qian [21] proposed neurophysic models of cortical cells that can recover depth from both random dot stereogram and photorealistic image pairs.

Earlier, Barnard et al. [2] and Dhond et al. [9] presented reviews covering major progresses from the mid 1970s to the late 1980s. Szeliski et al. [25] quantitatively compared stereo algorithms with different approaches. In [24], Scharstein and Szeliski provided more detailed classification of different kinds of stereo algorithms. The most recent review by Brown et al. [6] gave a very broad summary of the many advances in computational stereo algorithms in the last decade.

In this paper, we give a brief overview of the background knowledge, then focus on recent advances in global methods: dynamic programming and graph cuts.

2 Background of Computational Stereo Vision

The basic purpose of computational stereo vision is to recover the depth of the scene from two or more images taken at known positions. In this section we will summarize the process and geometry of stereopsis, or stereoscopic vision.

The mathematic foundation of stereo vision is quite simple. That is, given the projection of a 3D physical point on the two image planes and all the parameters of the stereo imaging system, we can find the exact location of the 3D physical point with some basic geometry skills.

The simplest geometry of a stereo imaging system is formed by two parallel cameras with a horizontal displacement which is called the stereo base line. The two cameras' optical axes are perpendicular to the stereo baseline, and their image scanlines are parallel to the baseline. Apart from their displacement, both cameras should have identical parameters. Figure 1 shows a simplified model of two parallel pin-hole cameras. The optical centers are O_l and O_r. The origin of the world coordinate system is the same as O_l, and the world coordinate axes X_w, Y_w and Z_w are coincident with the left camera's coordinate axes X_l, Y_l and Z_l. This overlap of coordinate systems is just for mathematic convenience. I_l and I_r represent the left and right image planes respectively. The point $P(x, y, z)$ is one point in the 3D scene that can be viewed by both cameras. The projections of $P(x, y, z)$ on the two image planes are called $P_l(x_l, y_l, z_l)$ and $P_r(x_r, y_r, z_r)$. In the stereo matching problem, the key issue is to relate $P_l(x_l, y_l, z_l)$ with $P_r(x_r, y_r, z_r)$ correctly and efficiently. As shown in Fig. 2, the **epipolar plane** is defined by the scene point $P(x, y, z)$ and the optical centers O_l and O_r. The epipolar plane PO_lO_r intersects the two image planes I_l and I_r at two lines called **epipolar lines.** Since the two epipolar lines are intersected by the same plane, they are also called **conjugate epipolar lines.** In this conventional parallel-axis geometry, all epipolar lines are horizontally parallel to the x axis. The epipolar constraint is one of the most significant conditions in stereo vision.

For example, in a standard stereo matching problem, given P_l, we can draw a back projection line from O_l through P_l to the world. It is obvious that the 3D scene point $P(x, y, z)$ which caused P_l, must lie on this back projection line, but where on the line is it? This cannot be deduced from only one image. Any point on the line P_lO_l is a possible solution. One projection point is not enough to accurately measure the distance of the objects from the image plane. With second eye or camera, we can ensure that the $P(x, y, z)$ is also projected on the image plane I_r. If we can find the correct corresponding point P_r in the right image I_r, we can pick the only correct $P(x, y, z)$ from the infinite possible candidates on the line O_lP_l. We can then find the intersection line of the plane $P_lO_lO_r$ and the right image plane I_r. All candidates of the correct corresponding point P_r lie on this epipolar line. Thus the stereo matching problem has been reduced from a 2D problem to a 1D problem, which can save a lot of computational effort. As Fig. 2 shows, if the stereo algorithm has found the correct match P_r, then we can determine the correct location of $P(x, y, z,)$; however, if another point P'_r has been incorrectly selected as the matching point, then we will choose the wrong 3D scene point $P'(x', y', z')$. Thus, the accuracy of the 1D point-matching algorithm is of essential importance in stereo vision.

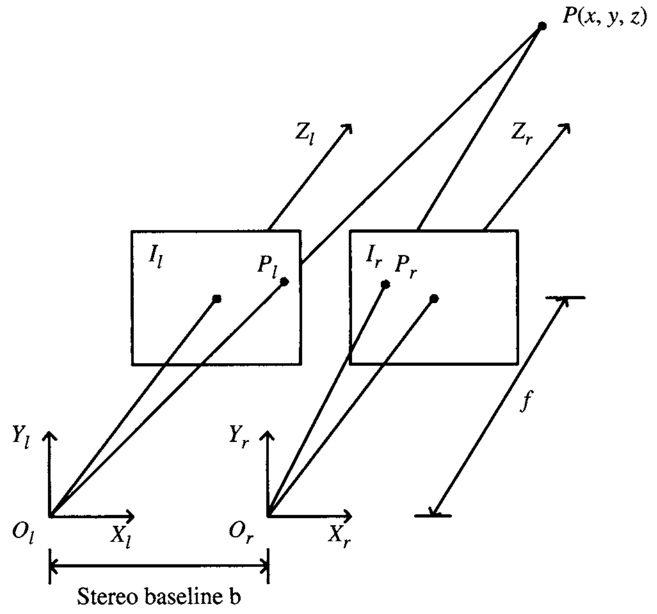


FIGURE 1 Parallel stereo geometry.

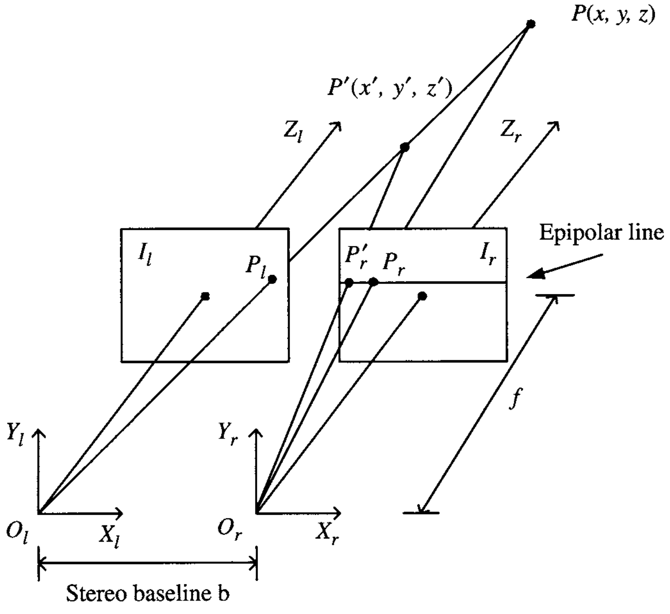


FIGURE 2 The epipolar lines of parallel stereo geometry.

The conventional parallel stereo geometry provides a disparity value d for each correctly matched pair $P_l(x_l, y_l, z_l)$ and $P_r(x_r, y_r, z_l)$, where $d = x_l - x_r$. Using the properties of similar triangles, it is easy to infer the exact location of $P(x, y, z)$:

$$x = \frac{bx_l}{d}, y = \frac{by_l}{d}, z = \frac{bf}{d} \qquad (1)$$

where b is the distance between the two optical centers (baseline) and f is the focal length of the camera's lens.

However, the parallel stereo geometry is the simplest geometry. Real situations can be much more complicated. For example, in biologic vision systems, two eyes are rarely looking in parallel directions. Most of the time when a subject fixates on an object, both eyes have a convergent angle toward the object. In such circumstances, the parallel stereo geometry is an oversimplified model. One advantage of convergent eye movements is that there is more overlap between the two visual fields. Hence the disparities of the interesting points, features, objects, and areas are smaller, which makes matching easier for the brain or computer.

However, if the optical axis of any one of the cameras is not parallel to the world z-direction, we lose the nice property of horizontally parallel epipolar lines. Figure 3 shows a general case of non-parallel stereo geometry. Once again we want to know how to locate the correct $P(x, y, z)$, given O_l, O_r, P_l, and the orientation of each of the two cameras or eyes.

Since we know the exact locations and orientations of the cameras, we can get the epipolar plane PO_lO_r (which is the same as $P_lO_lO_r$). Hence, the corresponding epipolar line on the right image plane I_r can be computed by solving the intersection of $P_lO_lO_r$ and I_r. Suppose we can correctly figure out which one is the correct matching P_r. Then by back-projecting a line from O_r through P_r, we can find the intersection point of O_rP_r and O_lP_l, which is $P(x, y, z)$. In non-parallel situations, P_r is not on a horizontal epipolar line. The search for the correct P_r is not limited to a horizontal scanline on the right image, which introduces more computational complexity.

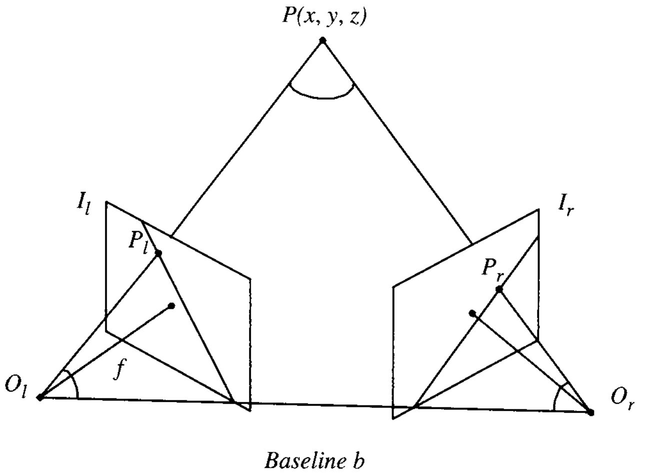


FIGURE 3 Non-parallel stereo geometry.

Non-parallel epipolar geometry can be simplified to parallel by stereo rectification. Given a pair of stereo images, rectification determines a transformation of each image plane such that pairs of conjugate epipolar lines become collinear and parallel to one of the image axes. The rectified images can be thought of as obtained by rotating the original cameras. The most important advantage of rectification is that computing stereo correspondences is reduced to a 1D search problem along the horizontal raster lines of the rectified images.

Correspondence, or matching is the key to stereo vision. It is the process that determines the locations, in both images, of the projections of the same physical point in space. Unfortunately, the stereo correspondence problem has also been a very difficult task since the beginning of stereo vision research. One reason for this is that the complicated 3D geometry can result in partial occlusion, where some areas in one image may not be visible in the other image. Even without occlusions, there are still many other problems posed by homogeneous regions and textured regions. Homogeneous regions are problematic because insufficient information is available to establish unique correspondences. Especially when dense pixel-by-pixel correspondence is needed, the matching of each pixel is very uncertain. Textured regions cause similar problems. In this case, there is an excess of available choices making it impossible to distinguish between matches, since multiple pixels or features have the same local characteristics. A periodic surface textures will cause many matches with the same confidence. As a result, unique correspondence is once again evasive.

In order to formalize stereo correspondence as a computational task, Marr and Poggio [16] proposed two main constraints to make it tractable:

1. Uniqueness: a pixel in one image can only be matched to one pixel in the other image.
2. Continuity: the world is composed by piecewise continuous surfaces, so the disparity varies smoothly almost everywhere.

In addition, there are some other constraints that can be utilized, such as the ordering constraints and the epipolar constraints.

In the following section, we will review some important stereo algorithms and classify them into different categories.

3 A Taxonomy of Stereo Correspondence Algorithms

To classify stereo correspondence algorithms is a formidable task, Since the numerous techniques have been introduced over the course of more than 30 years. However, we can still conveniently divide these algorithms into two groups: local methods and global methods.

3.1 Local Methods

In local methods, the correspondence of a pixel is decided by the relationships between the pixels in its neighborhood (usually defined as a rectangular window centered on the pixel) and those in the neighborhood of the corresponding pixel in the other image. The matching criterion includes such measures as normalized cross-correlation, sum of squared difference (SSD), normalized SSD, or sum of absolute differences (SAD). Local methods can be fast since the disparity values are decided by a relatively small number of computations in the local window. Their disadvantage is that they are easily affected by locally ambiguous regions, such as occlusions, textures, and homogeneous regions. In the following paragraphs, we shall review some influential local methods proposed in the 1990s.

Kanade et al., Adaptive Window. One critical problem in window-based matching algorithms is the selection of the window size. The dilemma is that while the window size should be large enough to include enough image details for a reliable matching, it also should be small enough to avoid projective distortions. Most of the correlation-based or SSD-based algorithms before [14] used a window with a fixed size chosen empirically for each application. Based on the fact that the intensity and disparity variance of the local window affect the performance of a window-based algorithm, Kanade et al. proposed an adaptive scheme to vary the window size by evaluating the local variations of the intensity and the disparity.

It is a chicken-and-egg problem, since it is necessary to know the variations of disparities prior to computing the disparities. To solve this dilemma, a statistical model of the local disparity distribution was proposed. Let's assume the two intensity images are $f_1(x,y)$ and $f_2(x,y)$, which have the following relation:

$$f_1(x,y) = f_2(x + d_r(x,y), y) + n(x,y) \tag{2}$$

where $n(x,y)$ represents the Gaussian white noise, and $d_r(x,y)$ is the disparity of the pixel at (x, y). Suppose the origin or point of interest within a window is (0, 0), then the statistical model for the disparity $d_r(\xi,\eta)$ within the window is:

$$d_r(\xi,\eta) - d_r(0,0) \sim N(0, \alpha_d\sqrt{\xi^2+\eta^2}) \tag{3}$$

Here α_d is a constant that represents the amount of variation of the disparity. The farther the pixel (ξ,η) is from the window origin $(0,0)$, the greater its disparity variation will be. This is consistent with properties of the natural world. With an intensity model and a disparity model, the variances of intensity and disparity can be computed between the same window applied to potential matches in both images. Kanade et al. [14] named these variances uncertainties. At the same time, the increment of disparity Δd at the window center $(0,0)$ can also be derived from the models. To summarize: for two images f_1, f_2, a local window W, and the current estimation of disparities $d_i(\xi,\eta)$ within W, a better estimate of disparity $d_{i+1}(0,0) + \Delta d$ may be available if a change in the size of W can reduce the uncertainties. The point is to find the disparity estimation and window size that can minimize the uncertainties.

An iterative algorithm such the following is performed:

1. Start with an initial disparity estimate $d_0(x,y)$.
2. At each pixel (x,y), choose a window that provides the estimate of disparity with lowest uncertainty, and update the disparity by $d_{i+1}(x,y) = d_i(x,y) + \Delta d(x,y)$.
   (a) Put a 3 × 3 window over the current pixel (x,y) and compute the uncertainties.
   (b) In turn, expand the window by one pixel in each direction (Fig. 4), and compute the uncertainties for the expanded window. Prohibit any direction that increases the uncertainties.
   (c) Compare the uncertainties for all the non-prohibited directions and choose the direction with the minimum uncertainties.
   (d) Expand the window by one pixel in the chosen direction.
   (e) Repeat (b) to (d) until all directions are prohibited.
3. Iterate steps 1 and 2 until the disparity $d_{i+1}(x,y)$ converges or a maximum (set) number of iterations have been conducted.

The experimental results showed the advantage of the adaptive window method over those algorithms with fixed-size windows, on both synthetic and real images.

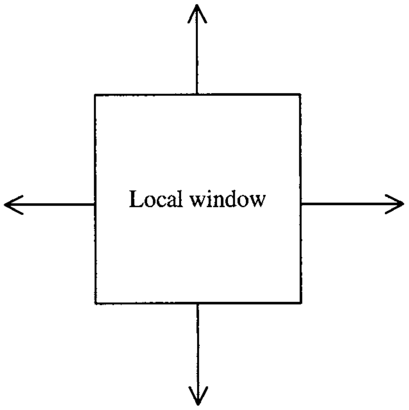


FIGURE 4 The expansion of the local window.

Okutomi et al., Multiple Baseline. For a given point observed using a parallel geometry of stereo images, the longer the baseline, the larger the disparity. Shorter baselines and longer baselines each have their own advantages and disadvantages. For example, it is easier to find a match between images with shorter baselines, because the disparity and intensity changes are smaller. Unfortunately, the errors in estimated distances will be amplified because of the small disparities (see Equation (1)), also the possible estimation errors. With a long baseline, a larger disparity range must be searched, introducing a higher probability of false matches. In [20], Okutumi et al. presented a method that used multiple stereo pairs with various baselines to obtain stereo correspondence.

SSD (sum of squared difference) is a widely used criterion for finding correspondences. The idea of [20] is that global mismatches can be reduced by adding the SSD values from different baselined images. This SSSD (sum of SSD) exhibits a unique and clear minimum at the correct matching position. Suppose the baseline distances are $B_1, B_2, \ldots B_n$, with $B_1 < B_2 < \ldots < B_n$, and the candidate disparity with baseline B_n is $d_{(n)}$. The SSSD value at pixel x is:

$$\mathrm{SSSD}(x, d_{(n)}) = \sum_{i=1}^{n} \sum_{j \in W} (f_0(x + j) - f_i(x + \frac{B_i}{B_n} d_{(n)} + j))^2 \quad (4)$$

where $f_0(x)$ is the reference image, $f_i(x)$ is the correspondent image with baseline B_i, and W is the local window. The correct $d_{(n)}$ will minimize the SSSD value uniquely, and give a better estimation than the single baseline method. The success of this method has been supported by theoretical analysis and the experimental results in [20].

3.2 Global Methods

Global methods are based on optimization. Most global algorithms try to define a well-behaved energy function associated with possible disparity maps. Then the correct disparity map is selected by minimizing this energy function. Since the energy function is defined on all the pixels of the image, global methods are less sensitive to local ambiguities (occlusions, textures, etc.) than are local methods. For the last 20 years, dynamic programming has been the technique used most often. That is because the nature of stereo correspondence coincides with dynamic programming quite well. Recently, a new technique called graph cut has produced impressive results. We will review and compare several influential dynamic programming algorithms and graph cut algorithms below.

3.2.1 Global Methods: Dynamic Programming

Dynamic programming is an algorithm design method that can be used when the solution to a problem can be viewed as the result of a sequence of decisions [11]. Baker and Binford [1] used the Viterbi algorithm, a dynamic programming technique, to partition the stereo correspondence problem recursively based on a left-to-right ordering of edges preserved along a scanline in a stereo image pair. From the 1980s, Ohta and Kanade [18], Belhumeur and Mumford [3], Cox et al. [7], Birchfield and Tomasi [4], and Van Meerbergen et al. [17] all applied dynamic programming techniques in the stereo correspondence problem. In this section, we review three important dynamic programming algorithms.

Ohta et al. [18]. As presented in [18], dynamic programming solves an N-stage decision process as N single-stage processes. In order to apply dynamic programming, two requirements must be met: (1) The decision stage must be ordered, which means that all stages whose results are needed at the current stage have been processed prior to the current stage. (2) The decision process must be Markovian: at any stage, the behavior of the process itself depends only on the current state and not on the previous history. In [18], Ohta and Kanade clarified that the problem of finding stereo correspondence between connected edges in stereo images is a combination of intra-scanline and inter-scanline matching. With the ordering constraint on the features at each scanline, the stereo correspondence can be solved by dynamic programming.

After the stereo pairs have been rectified, the epipolar lines are horizontal scanlines. First, an edge detector extracts all possible edges in the left and right scanlines, then left and right groups of edges are matched with each other. This is called intra-scanline matching. In Fig. 5, the horizontal and vertical axes are image intensities on the left and right scanlines respectively. The edges in the left scanline are indexed in left-to-right order as [0 : N], and the edges of the

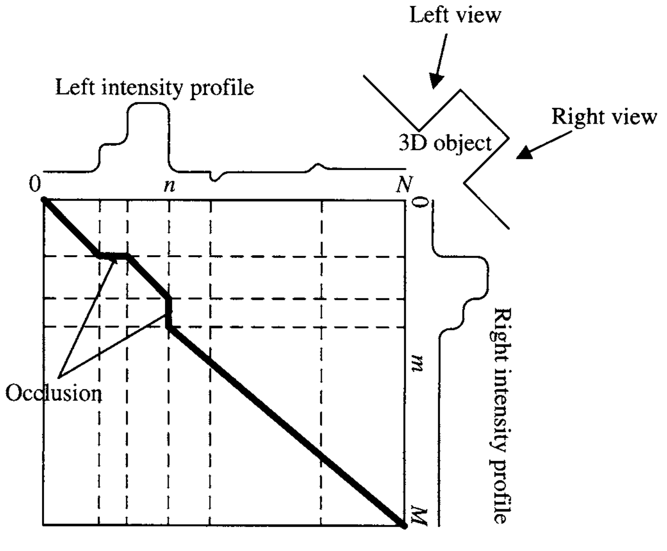


FIGURE 5 The intra-scanline search.

right scanline are indexed as $[0 : M]$. Intra-scanline matching is accomplished by finding the optimal legal path on this 2D search plane. Here, a **node** (i,j) is a possible pair of matched edges, where i, j are the edge indices of the left and right scanline respectively. A **primitive path** is a partial path between two nodes that contains no intervening nodes, and is represented by a straight line segment as shown in Fig. 5. A **legal path** is a sequence of connected primitive paths from node $(0,0)$ at the upper-left corner to the node (N,M) on the lower-right corner with the ordering constraint:

if (p,q) *and* (r,s) *are two matched pairs, where* $p< r$, *then* q *must be less than* s.

Suppose $a_1 \ldots a_k$ and $b_1 \ldots b_l$ are the intensity values between two neighboring nodes (edges) on the left and right scanlines respectively. Then the mean and variance of all pixels in the two intervals are defined as:

$$m = \frac{1}{2}\left(\frac{1}{k}\sum_{i=1}^{k} a_i + \frac{1}{l}\sum_{j=1}^{l} b_j\right) \tag{5}$$

$$\sigma^2 = \frac{1}{2}\left(\frac{1}{k}\sum_{i=1}^{k}(a_i - m)^2 + \frac{1}{l}\sum_{j=1}^{l}(b_j - m)^2\right). \tag{6}$$

The cost of the primitive path that matches the two edges is defined as:

$$C_p = \sigma^2\sqrt{k^2 + l^2}. \tag{7}$$

If the pixels in the two intervals are correctly matched to each other, then usually they are from the same homogeneous surface in the scene and therefore have similar intensities. In other words, their variances C_p should be small. The intra-scanline search tries to find the optimal division of each scanline into line segments, with an eye toward matching the segments from one image to the other to minimize the cost function C_p.

When there is occlusion, as in Fig. 5, a vertical or horizontal primitive path appears. In the definition of (7), occlusion has not been considered. Ohta and Kanade suggested that a mismatch caused by an occlusion can be treated as the consequence of avoiding two paths drawn with dotted lines, as Fig. 6 shows.

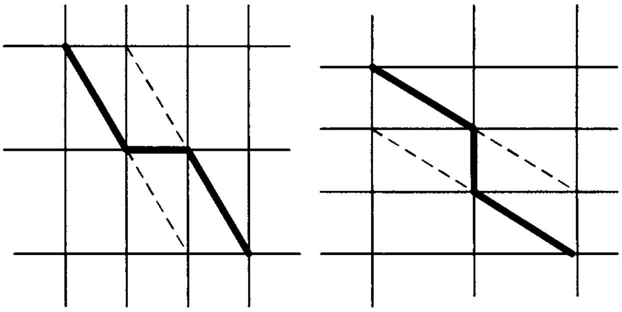


FIGURE 6 Primitive paths for occlusion, the cost of the horizontal/vertical path is defined based on the dotted paths, from [18]. (© 2004 IEEE.)

The cost of an occlusion primitive path is defined as a function of the costs of those two imaginary dotted paths. The variances σ_1 and σ_2 are computed the same way as (7). The cost of an occlusion primitive path is defined as:

$$C_{\text{Occ}} = k \times f((\sigma_1^2 + \sigma_2^2)/2;\ Th) \tag{8}$$

where k is the length of the occlusion primitive path, Th is a threshold, and f is a function that has the shape of Fig. 7. If the occlusion occurs, then the two alternative dotted paths are not likely to be correct and the $\sigma_1^2 + \sigma_2^2$ tends to be larger. As this value increases beyond the threshold, the cost of the occlusion becomes a constant and is overtaken by the costs of the individual paths. Hence when a correct occlusion is detected, its cost is smaller. When there is no occlusion, one of the two dotted path is likely to be correct, and the cost-of-occlusion term C_{Occ} will contain a smaller value of $\sigma_1^2 + \sigma_2^2$, resulting in a larger cost for the nonexistent occlusion.

After assigning a cost to each possible primitive path, the intra-scanline matching process is a typical dynamic programming problem of finding the optimal path with the smallest overall cost.

Since the neighboring scanlines are highly correlated, it is very likely that the edges run across the neighboring scanlines. To utilize this correlation, an inter-scanline search is performed after the intra-scanline search. The problem is posed as that of finding the minimum-cost path between 3D nodes in a stack of the neighboring 2D search planes. A 3D node is a collection of 2D nodes that is on the same edge crossing the neighboring scanlines. This 3D optimal-path-finding method is formalized to a dynamic programming problem in the same way as the intra-scanline search.

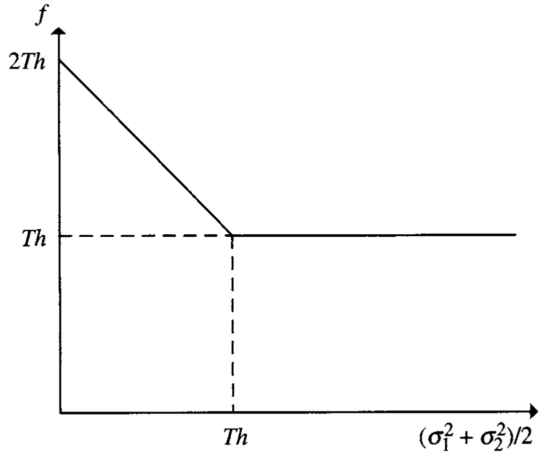


FIGURE 7 Mapping functions for the cost of a occlusion primitive path, from [18]. (© 2004 IEEE.)

Cox et al. [7]. In [18], the edges on a scanline are first extracted as features for stereo matching. Cox et al. [7] argued that a stereo matching based on features is not as reliable as pixel-based matching for three reasons: (1) Feature extraction is not 100% reliable. (2) Features in one image might be occluded in the other image. (3) The similarity between two features maybe perturbed by noise. They proposed a maximum-likelihood, pixel-based stereo algorithm using dynamic programming. They also pointed out that the cost function used in Ohta and Kanade [18] (Equation 7) is based on the image intensity variance of each primitive path, which is not appropriate. One reason is that the textured regions usually have a large variance, which generates a biased cost.

Let the two cameras be denoted as $s = 1, 2$. $\mathbf{Z}_s = \{\mathbf{z}_{i_s}, i_s = 0, 1 \ldots m_s\}$ is the set of m measurements obtained from camera s on a scanline, where m_s is the number of measurements taken on one scanline. Each $\mathbf{z}_{i_s}$ is the value of either a pixel's intensity or a higher level feature. It is assumed that $\mathbf{z}_{i_s}$ has an additive white noise. Let $\mathbf{X}$ be a location in the 3D space. Z_{i_1, i_2} represents the relationship that $\mathbf{z}_{i_1}$ and $\mathbf{z}_{i_2}$ are originated from the same 3D point $\mathbf{X}$. The occlusion of $\mathbf{z}_{i_1}$ in camera 2 is denoted as $Z_{i_1,0}$. Then, Cox et al. [7] defined a likelihood function for when $\mathbf{z}_{i_1}$ and $\mathbf{z}_{i_2}$ are correspondent:

$$\Lambda(Z_{i_1,i_2}|\mathbf{X}) = \left(\frac{1 - P_D}{\phi}\right)^{\delta_{i_1,i_2}} [P_D p(\mathbf{z}_{i_1}|\mathbf{X}_k) \times P_D p(\mathbf{z}_{i_2}|\mathbf{X}_k)]^{1-\delta_{i_1,i_2}}. \tag{9}$$

Here δ_{i_1,i_2} is an indicator function that takes the value 1 when i_1 and i_2 are **not** correspondent, 0 otherwise. ϕ is the field of view of the camera. $p(z_{i_s}|X)$ is the likelihood function of z_{i_s} from the 3D point X. P_D is the probability of detecting a measurement from X at camera $\boldsymbol{s}$. Hence, $(1 - P_D)$ is the probability of occlusion. Since the measurement is assumed to have additive white gaussian noise:

$$p(\mathbf{z}_{i_s}|X) = |(2\pi)^d S_s|^{-1/2} \exp\left(-\frac{1}{2}(z_{i_s} - z)' S_s^{-1}(z_{i_s} - z)\right). \tag{10}$$

Here z is the true value of the measurement, which is approximated by its maximum likelihood estimation in [7]. After the likelihood of each pair z_{i_1} and z_{i_2} is defined, the overall likelihood along a scanline can be defined as:

$$L(\gamma) = \prod_{Z_{i_1,i_2} \in \gamma} \Lambda(Z_{i_1,i_2}|X) \tag{11}$$

where γ is a feasible pairing of all measurements. Let Γ be all possible pairings, then the stereo matching problem is transformed to finding the pairing $\gamma \in \Gamma$ that maximizes the likelihood $L(\gamma)$, or minimizes $1/L(\gamma)$.

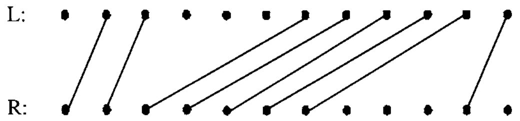


FIGURE 8 The matched sequence along a scanline, from [4]. (With kind permission from Springer and Business Media.)

This maximization (minimization) can be solved by dynamic programming. Cox et al. [7] used two constraints here, the uniqueness constraint and ordering constraint, omitting the smoothness constraint, which Ohta and Kanade [18] implied when they used variance as the cost. Also, for multiple global minima, the authors enforced 2D cohesiveness constraints by minimizing the number of horizontal and vertical discontinuities.

Birchfield et al. [4]. Birchfield and Tomasi [4] posed the stereo matching as a problem of corresponding pixels in the left scanline to those in the right scanline, as in most of previous algorithms.

Figure 8 shows an example of matching between two scanlines. Suppose there are n pixels on each scanline. Among the n pixels, there are N_m matched pairs. $\{x_1, x_2, \ldots. x_{N_m}\}$, $\{y_1, y_2, \ldots. y_{N_m}\}$ are the horizontal indices of the left and right matched pixels respectively. Some important constraints are imposed on the match sequence:

1. $0 \le x_i - y_i \le \Delta$. This means there is a maximum disparity Δ, which limits the search region, and the disparity is always non-negative because the images has been rectified.
2. $y_1 = 0$ and $x_{N_m} = n - 1$. The matching is forced to start from the first pixel of the right scanline, and to end at the last pixel of the left scanline.
3. $x_i < x_j$ and $y_i < y_j$, if $1 \le i < j \le N_m$. These are the ordering constraint and the unique constraint.
4. $x_{i+1} = x_i + 1$, or $y_{i+1} = y_i + 1$, $i = 1, 2, \ldots, N_m - 1$, which implies that the pixel is visible to either the left camera or the right camera.

So, in Fig. 8, the match sequence is: $M = \{(1,0), (2,1), (6,2), (7,3), (8,4), (9,5), (10,6), (11,10)\}$, the disparity sequence here is $\{1, 1, 4, 4, 4, 4, 4, 1\}$. The unmatched pixels in the left (right) image are occluded in the right(left) scanline.

Similarly, there is a cost function $\gamma(M)$ associated with each possible match sequence:

$$\gamma(M) = N_{\text{occ}}\kappa_{\text{occ}} - N_m\kappa_r + \sum_{i=1}^{N_m} d(x_i, y_i) \tag{12}$$

N_{occ} and N_m are the numbers of occlusions and matches. In Fig. 8, $N_{\text{occ}} = 7$ and $N_m = 8$. κ_r is the reward for correct matches. κ_{occ} is the penalty for each occlusion (the introduction of κ_{occ} in the cost function implies the smoothness

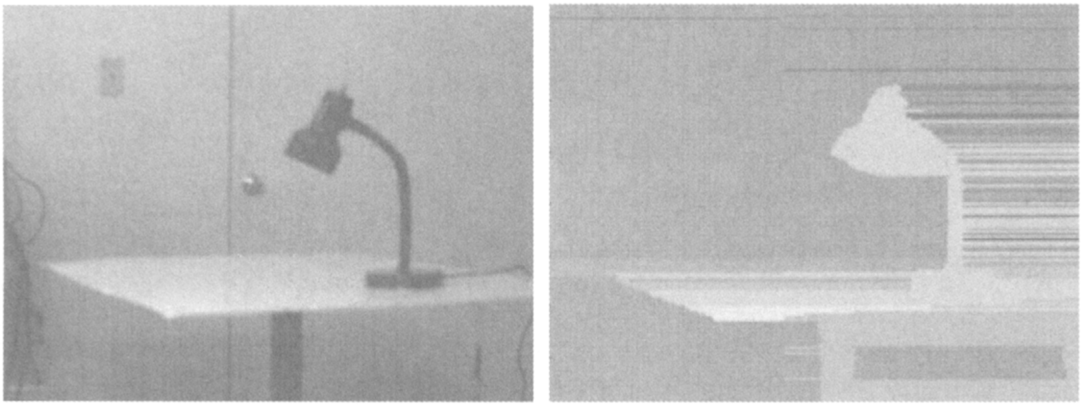


FIGURE 9 The horizontal streak along a scanline, from [4]. (With kind permission of Springer and Business Media.)

constraint). $d(x_i, y_i)$ is the dissimilarity between two matched pixels, which, for computational efficiency is usually the absolute difference between the intensity values of two pixels.

3.2.2 Global Methods: Graph Cuts

Because of the intrinsic properties of the pixels on the left and right scanline and the constraints given by researchers, dynamic programming is quite successful in solving the correspondence along a scanline. However, there are some disadvantages of dynamic programming. First, because of the sequential order of decision making, local errors can be propagated along a scanline, which typically results in a horizontal streak after the error pixel. Figure 9 shows one image and its depth map obtained from a dynamic programming algorithm. Heavy horizontal streaks are obvious to the right of the lamp. Another significant limitation of dynamic programming is the difficulty to combine the horizontal and vertical continuity constraints of the real scene together. Dynamic programming is basically a 1D optimization algorithm. It lacks the ability to exploit 2D coherence properties of the real scene. Roy et al. [22, 23] pointed out that the solutions at consecutive epipolar lines can vary significantly, causing serious artifacts across epipolar lines. These artifacts usually need some postprocessing to be removed. Instead, Roy et al. [23] formalized a multi-camera stereo problem as a maximum-flow (minimum-cut) problem, which does not rely on the epipolar geometry and the traditional ordering constraint. Ishikawa and Geiger [13] also presented a similar method in [23]. Recently, Boykov et al. proposed a 2D global algorithm, using graph cuts, which exhibits excellent performance [25]. Generally, graph cuts algorithms require more compuational time than dynamic programming methods. The computational complexity of the worst case is $O(n^{1.5}d^{1.5}\log(nd))$ for an image size of n pixels and disparity range d. In [23], the average running time observed by Roy et al. was $O(n^{1.2}d^{1.3})$. This is already close to the typical dynamic programming complexity $O(nd)$ or $O(n^2)$.

Let us review graph cuts. Define $G = \langle V, E\rangle$ to be a weighted graph with two distinguished vertices, called terminals, where V is the set of vertices and E is the edge set. A cut C is a set of edges that separate the terminals in the induced graph $G(C) = \langle V, E - C\rangle$. Also, no subset of C can separate the terminals in $G(C)$. The cost of a cut C is the sum of the edge weights in C, denoted by $|C|$.

For example, in Fig. 10, the two terminals are α and β. The edges connected to the two terminals are called t-links (terminal links), which are t_p^α, t_p^β, t_q^α and t_q^β. The edges between two non-terminal nodes, $e_{[p,q]}$, are called n-links (neighbor links). There are four possible cuts: $\{t_p^\alpha, t_q^\alpha\}$, $\{t_p^\beta, t_q^\beta\}$, $\{t_p^\alpha, t_q^\beta, e_{[p,q]}\}$, and $\{t_q^\alpha, t_p^\beta, e_{[p,q]}\}$. For the purpose of illustration, Fig. 10 only displays three of them. Since the edges of the graph are weighted, there exists a cut with the minimum cost among all possible cuts separating the two terminals. By carefully designing the graph, Boykov et al. [5] associated the stereo matching problem with the minimum cut problem.

Stereo matching can be viewed as labelling every pixel with a disparity value. Suppose P is the set of all pixels and L is the set of all possible disparity values. The goal of stereo matching is to find a labelling f that assigns each pixel $p \in P$ with a label $f_p \in L$ [5], where f should be as close as possible to the ground truth disparities, if applicable. Most of the time, researchers will assign some cost or energy functions to a labelling

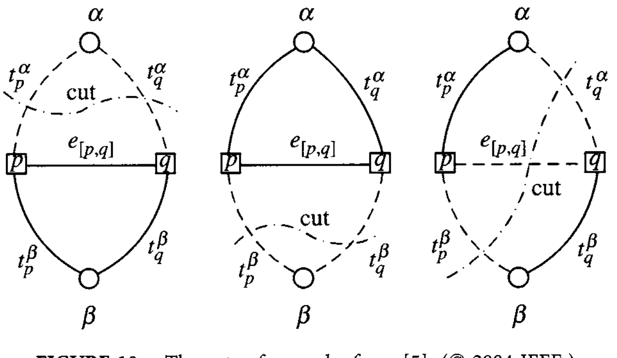


FIGURE 10 The cuts of a graph, from [5]. (© 2004 IEEE.)

scheme, and attempt to find the particular labelling that minimizes the energy. In [5], the total energy function is defined as:

$$E(f) = E_{\text{smooth}}(f) + E_{\text{data}}(f). \tag{13}$$

E_{smooth} represents the energy that f is not piecewise smooth, and E_{data} measures the disagreement between f and the observed data.

$$E_{\text{smooth}}(f) = \sum_{\{p,q\}\in N} V_{p,q}(f_p, f_q) \tag{14}$$

$$E_{\text{data}}(f) = \sum_{p\in P} D(I_p - I'_{p+f_p}). \tag{15}$$

In (14), N is a neighborhood system of pixel p. The choice of E_{smooth} is a critical issue. Various functions have been proposed. Some of them encourage smoothness everywhere, which will smear the object boundaries. An ideal energy function should encourage piecewise smoothness, but preserve the correct disparity discontinuities at the object boundaries. Such functions are also called discontinuity preserving. Informally, a function $V(x, y)$ is discontinuity preserving if $\sup_{x,y\in R}(V(x, y)) < K$ for some constant K, which means the maximum penalty for assigning different labels to neighboring pixels can be bounded. One example in [5] is $V(\alpha, \beta) = \min(K, (\alpha - \beta)^2)$, where K is some constant. Here I_p is the intensity value of the pixel p in the left image, and I'_{p+f_p} is the intensity of the matching pixel in the right image since f_p is the disparity label of p. By assuming similar intensities between two corresponding pixels, the function D should behave proportional to the intensity differences. For example, one possible form of D could be $|I_p - I'_{p+f_p}|$.

The Outline of the Algorithm. A labelling f can separate all the image pixels P into different partitions $P_l = \{p \in P \mid f_p = l\}$. P_l is the set of pixels with the same disparity label l. Thus, a labelling function f will uniquely define the partition of P.

Given two disparities α and β, a move from a partition $P(f)$ to a new partition $P'(f')$ is called *α-beta* swap if $P_l = P'_l$ for any label $l \neq \alpha, \beta$. An *α-β swap* indicates that some pixels with disparity α in f have been labelled as β in f', and some β pixels have become α pixels, while pixels labelled with other disparities have kept the same label as before.

The other move is called α expansion. In this move, $P_\alpha \subset P'_\alpha$ and $P_l \subset P'_l$ for $l \neq \alpha$. This move means some pixels labelled with other disparities have been labelled with disparity α.

Defining these two moves are preparatory steps in the simplification of the optimization of the energy function (13). Because of the *NP*-completeness of the optimization task proved by [15], Boykov et al. proposed the following two algorithms, which find an approximate solution:

Swap Algorithm

```
1. Start with an arbitrary disparity labelling $f$
2. Set success := 0
3. For each pair of disparities $\{\alpha, \beta\} \subset L$
   (a) Find $\hat{f} = \text{argmin}(E(f'))$ among $f'$ within one $\alpha$-$\beta$ swap of $f$
   (b) If $E(\hat{f}) < E(f)$, set $f := \hat{f}$ and success := 1
4. If success = 1 go to 2
5. Return $f$
```

Expansion Algorithm

```
1. Start with an arbitrary disparity labelling $f$
2. Set success := 0
3. For each disparity $\alpha \in L$
   (a) Find $\hat{f} = \text{argmin}(E(f'))$ among $f'$ within one $\alpha$ expansion of $f$
   (b) If $E(\hat{f}) < E(f)$, set $f := \hat{f}$ and success:= 1
4. If success = 1 go to 2
5. Return $f$
```

Given a labelling f, it is very expensive to find the optimal α-β swap or α expansion over all pairs of label $\{\alpha, \beta\}$. Instead of a global optimization, step 3 of the *Swap Algorithm* performs a search on every pair of disparities to find the labelling that minimizes the energy within one swap (expansion) move. Thus, the algorithm finds a sequence of local minima until the minima cannot reduce anymore. It cannot guarantee that this sequence of local minima will finally converge to the correct global minimum, but a careful proof [5] shows that a local minimum, when an expansion move is allowed, is within a known factor of the global minimum.

In the algorithms, the key step is 3(a). That is how the graph cut is utilized here. By carefully designing the graph and assigning edge weights, step 3(a) is proved to be a minimum cut problem.

The structure of the graph is illustrated in Fig. 11. The two terminal vertices represent a pair of labelling $\{\alpha, \beta\}$. All other non-terminal vertices are the pixels $p \in P_{\alpha\beta}$ (that is, the set of pixels whose disparities are either α or β). Each pixel is connected to the terminal α and β by the t-links t_p^α and t_p^β. Each pair of pixels p, q that belongs to the same neighborhood N is connected by the n-link $e_{\{p,q\}}$.

For each edge, the weight is assigned as shown in Table 1.

According to the property of the cut, any cut C will include **exactly** one t-link for any pixel $p \subset P_{\alpha\beta}$. The reason is that if the cut severs none of the two t-links of the same pixel, there would be a path between two terminals; if the cut severs both t_p^α and t_p^β, then both graphs $\langle V, E - C + t_p^\alpha\rangle$ and

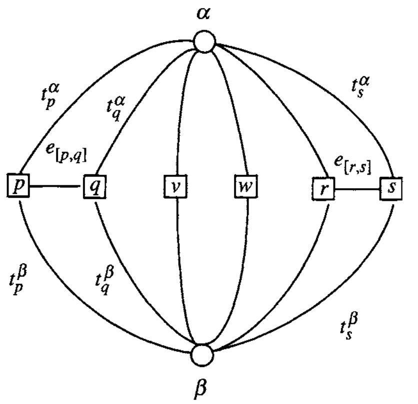


FIGURE 11 An example of a graph $G_{\alpha\beta}$ for a 1D image, from [5]. (© 2004 IEEE.)

$\langle V, E - C + t_p^\beta \rangle$ would be connected, which means there exists a path $l_1 : \alpha \leftrightarrow p \leftrightarrow \cdots \leftrightarrow \beta$ in $\langle V, E - C + t_p^\alpha \rangle$, and a path $l_2 : \beta \leftrightarrow p \leftrightarrow \cdots \leftrightarrow \alpha$ in $\langle V, E - C + t_p^\beta \rangle$. If we call the path $p \leftrightarrow \cdots \leftrightarrow \beta$ to be l_1', and the path $p \leftrightarrow \cdots \leftrightarrow \alpha$ to be l_2', then l_1' and l_2' would form a new path l': $\alpha \leftrightarrow \cdots \leftrightarrow p \leftrightarrow \cdots \leftrightarrow \beta$ such that $l' \in \langle G, E - C \rangle$, and l' still connects α and β, which contradicts the fact that C is a cut. As a result, C cannot sever t_p^α and t_p^β simultaneously.

A cut C can induce a new labelling f^C in the following way:

$$f_p^C = \begin{cases} \alpha, & \text{if} \quad t_p^\alpha \in C \text{ for } p \in P_{\alpha\beta} \\ \beta, & \text{if} \quad t_p^\beta \in C \text{ for } p \in P_{\alpha\beta} \\ f_p, & \text{if} \quad p \in P \text{ and } p \notin P_{\alpha\beta} \end{cases}$$

Because of the way the graph is designed, f^C is within one α-β swap of the original labelling f. Boykov et al. [5] gave a rigorous proof that (1) there is a one-to-one mapping between a cut C and the induced labelling f^C, and (2) the total cost of a cut C on $G_{\alpha\beta}$ equals the energy function $E(f^C)$ plus a constant.

Now it is apparent that the optimal α-β swap from f is f^C when C is the minimum cut on $G_{\alpha\beta}$. The energy minimization process has now been transformed to minimum cut problem.

TABLE 1 The weight assignment of the graph $G_{\alpha\beta}$

Edge	Weight	For
t_p^α	$D_p(\alpha) + \sum_{q \in N_p, q \notin P_{\alpha\beta}} V(\alpha, f_q)$	$p \in P_{\alpha\beta}$
t_p^β	$D_p(\beta) + \sum_{q \in N_p, q \notin P_{\alpha\beta}} V(\beta, f_q)$	$p \in P_{\alpha\beta}$
$e_{\{p,q\}}$	$V(\alpha, \beta)$	$p, q \in N, p, q \in P_{\alpha\beta}$

Kolmogorov et al. [15] further improved the graph cut algorithm by introducing an occlusion term:

$$E(f) = E_{\text{smooth}}(f) + E_{\text{occ}}(f) + E_{\text{data}}(f) \qquad (17)$$

where $E_{\text{occ}}(f)$ represents the penalty associated with the pixels that cannot be matched. The results are similar to Boykov et al. [5].

Inspired by a color-segmentation-based stereo framework [26] and [5], Hong et al. [10] used color segmentation to reduce the size of the search space and also enforced disparity smoothness in homogeneous color regions. The graph vertices in [10] are segments instead of single pixels, but the swap and expansion algorithms are identical to [5].

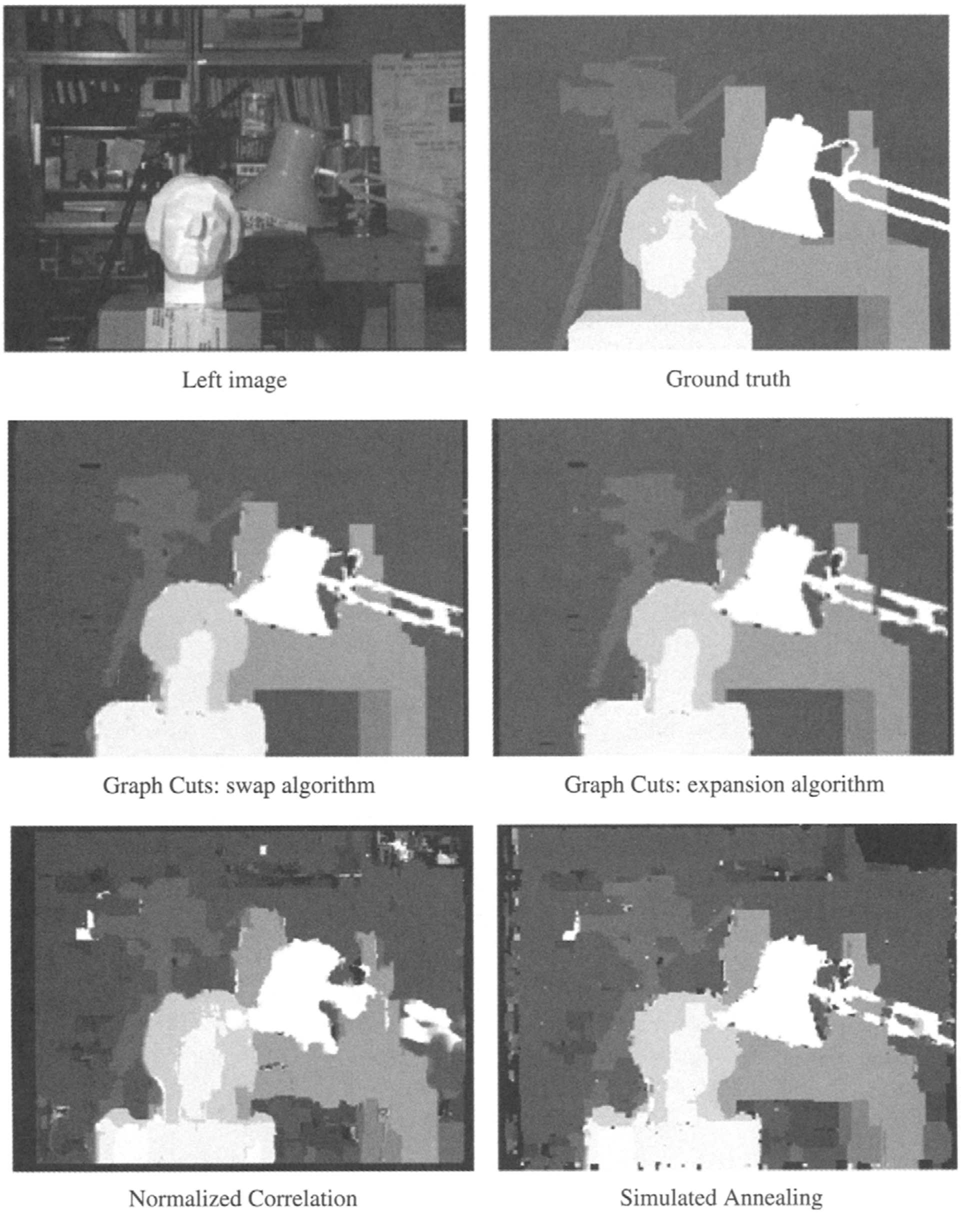
Szelizki et al. [25] compared several techniques using the University of Tsukuba dataset. They defined the error to be ±1 away from the ground truth. For local methods, they tested several correlation-based methods using L_1, L_2 distance, etc. They also plotted the total accuracy against the window radius. Interestingly, they found that window sizes do not play a significant role in the total performance once the windows are sufficiently large. For global methods, the graph cuts algorithm [5] and simulated annealing were tested. Another important non-energy—minimization global method was also compared. This algorithm was proposed by Zitnick et al. [27], which improved the Marr-Poggio [16] cooperative method. The experimental results suggested that the graph cuts algorithm has the most accurate performance, especially in low textured areas.

4 Conclusion

In this paper, we reviewed several important local and global algorithms proposed in the last decade. Local methods are faster, since the search space is relatively limited to a small local area, but they are more sensitive to local ambiguities like low or high texture regions and occlusions. Global algorithms are primarily based on optimization, and make use of existing optimization techniques such as dynamic programming and graph cuts. By quantitative analysis [5] and [24], the global methods have been shown to outperform the local ones in accuracy.

A great number of stereo algorithms, based on different philosophies and methodologies, have been proposed over the last 30 years. One major difficulty is trying to quantitatively evaluate these algorithms. Scharstein et al. [24] tested several popular algorithms on the same dataset (from Tsukuba and Microsoft) with ground truth. Their method is a promising step toward understanding the difference in the behaviors of these different algorithms.

One major unsolved problem of stereo is the combination of efficiency and accuracy. Although global algorithms have impressive accuracy, they cannot run in real time. To build a



Left image

Ground truth

Graph Cuts: swap algorithm

Graph Cuts: expansion algorithm

Normalized Correlation

Simulated Annealing

FIGURE 12 Experiment results of three different techniques (Normalized Correlation, Simulated Annealing and Graph Cuts), from [5]. (© 2004 IEEE.)

machine that can roam the real 3D world still remains a great challenge for all computer vision researchers.

Acknowledgment

Special thanks go to Marco Perez, Murphey Johnson, Qingfeng Yu, Debi Prather, and Al Bovik for their generous help and insightful comments during the preparation of this paper.

References

[1] H. H. Baker and T. O. Binford. "Depth from edge and intensity based stereo." In *Proc. 7th Int. Joint Conf. Artificial Intell.*, 631–636, Vancouver, Canada, August 1981.

[2] S. T. Barnard and M. A. Fischler. "Computational stereo." *ACM Computing Surveys*, 14:553–572, 1982.

[3] P. N. Belhumeur and D. Mumford. "A bayesian treatment of the stereo correspondence problem using half-occluded regions." In *Proc. Comp. Vis. and Pattern Rec.*, 1992.

[4] S. Birchfield and C. Tomasi. "Depth discontinuities by pixel-to-pixel stereo." *Int'l J. Computer Vision*, 35(3):269–293, 1999.

[5] Y. Boykov and O. Veksler. "Fast approximate energy minmization via graph cuts." *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, November 2001.

[6] M. Z. Brown, D. Burschka, and G. D. Hager. "Advances in computational stereo." *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 25(8):993–1007, August 2003.

[7] I. J. Cox, S. L. Hingorani, S. B. Rao, and B. M. Maggs. "A maximum likelihood stereo algorithm." *Computer Vision and Image Understanding*, 63(3):542–567, 1996.

[8] G. C. DeAngelis, I. Ohzawa, R. D. Freeman. "Depth is encoded in the visual cortex by a specialized receptive field structure." *Nature*, 352:156–159, July 1991.

[9] U. R. Dhond and J. K. Aggarwal. "Structure from stereo-a review." *IEEE Trans. on Systems, Man, and Cybernetics*, 19(6):1489–1510, 1989.

[10] L. Hong and G. Chen. "Segment-based stereo matching using graph cuts." In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1, 74–81, IEEE, 27 June–2 July 2004.

[11] E. Horowitz, S. Sahni, and S. Rajasekaran. *Computer Algorithms*. W. H. Freeman and Company, New York, 1997.

[12] D. H. Hubel and T. Wiesel. "Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex." *Journal of Physiology*, 160:106–154, 1962.

[13] H. Ishikawa and D. Geiger. "Occusions, discontinuities, and epipolar lines in stereo." In *Fifth European Conference on Computer Vision (ECCV'98)*, Freiburg, Germany, June, 1998. Springer.

[14] T. Kanade and M. Okutomi. "A stereo matching algorithm with an adaptive window: theory and experiment." *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 16(9):920–932, September 1994.

[15] V. Kolmogorov and R. Zabih. "Computing visual correspondence with occlusions using graph cuts." In *International Conference on Computer Vision*, 2:508–515, 2001.

[16] D. Marr and T. Poggio. "A computational theory of human stereo vision." *Proceedings of the Royal Society of London B*, 204:301–328, 1979.

[17] G. V. Meerbergen, M. Vergauwen, M. Pollefeys, and L. V. Gool. "A hierarchical symmetric stereo algorithm using dynamic programming." *Int'l J. Computer Vision*, 47:275–285, 2002.

[18] Y. Ohta and T. Kanade. "Stereo by intra- and inter-scanline search using dynamic programming." *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 7(2):139–154, March 1985.

[19] I. Ohzawa, G. C. DeAngelis, and R. D. Freeman. "Stereoscopic depth discrimination in the visual cortex: Neurons ideally suited as disparity detectors." *Science*, 249:1037–1041, 1990.

[20] M. Okutomi and T. Kanade. "A multiple-baseline stereo." *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 15(4):353–363, April 1993.

[21] N. Qian and Y. Zhu. "Physiological computation of binocular disparity." *Vision Res.*, 37(13):1811–1827, 1997.

[22] S. Roy. "Stereo without epipolar lines: a maximum-flow formulation." *Int'l J. Computer Vision*, 34:147–161, 1999.

[23] S. Roy and I. J. Cox. "A maximum-flow formulation of n-camera stereo correpondence problem, " In *Proceedings of the Sixth International Conference on Computer Vision*, 1998.

[24] D. Scharstein and R. Szeliski. "A taxtomy and evaluation of dense two-frame stereo correspondence algorithms." *Int'l J Computer Vision*, 47(1):7–42, 2002.

[25] R. Szeliski and R. Zabih. "An experimental comparison of stereo algorithms." In *International Workshop on Vision Algorithms*, 1–19, Greece, September 1999, Springer.

[26] H. Tao, H. S. Sawhney, and R. Kumar. "A global matching framework for stereo computation." In *International Conference on Computer Vision*, 2001.

[27] C. Zitnick and T. Kanade. "A cooperative algorithm for stereo matching and occlusion detection." Technical Report CMU-RI-TR-99-35, Robotics Institutes, Carnegie Mellon University, University, Pittsburgh, PA, October 1999.