A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms

Daniel Scharstein
Dept. of Math and Comp. Science
Middlebury College
Middlebury, VT 05753
schar@middlebury.edu

Richard Szeliski
Microsoft Research
Microsoft Corporation
Redmond, WA 98052
szeliski@microsoft.com

Ramin Zabih
Dept. of Computer Science
Cornell University
Ithaca, NY 14850
rdz@cs.cornell.edu

Abstract

Stereo matching is one of the most active research areas in computer vision. While a large number of algorithms for stereo correspondence have been developed, relatively little work has been done on characterizing their performance. In this paper, we present a taxonomy of dense, two-frame stereo methods designed to assess the different components and design decisions made in individual stereo algorithms. Using this taxonomy, we compare existing stereo methods and present experiments evaluating the performance of many different variants. In order to establish a common software platform and a collection of data sets for easy evaluation, we have designed a stand-alone, flexible C++ implementation that enables the evaluation of individual components and that can be easily extended to include new algorithms. We have also produced several new multi-frame stereo data sets with ground truth, and are making both the code and data sets available on the Web.

1. Introduction

Stereo correspondence has traditionally been, and continues to be, one of the most heavily investigated topics in computer vision. However, it is sometimes hard to gauge progress in the field, as most researchers only report qualitative results on the performance of their algorithms, and the last exhaustive stereo surveys date back about a decade [24, 19]. This paper provides an update on the state of the art in the field, with particular emphasis on stereo methods that (1) operate on two frames under known camera geometry, and (2) produce a *dense* disparity map, i.e., a disparity estimate at each pixel. Our goals are two-fold:

1. to provide a **taxonomy** of existing stereo algorithms that allows the dissection and comparison of individual algorithm components design decisions, and
2. to provide a **test bed** for the quantitative evaluation of stereo algorithms. Towards this end, we are placing sample implementations of correspondence algorithms along with test data and results on the Web at `www.middlebury.edu/stereo`.

We emphasize calibrated two-frame methods in order to focus our analysis on the essential components of stereo correspondence. However, it would be relatively straightforward to generalize our approach to include many multi-frame methods, in particular multiple-baseline stereo [54] and its plane-sweep generalizations [21, 71].

The requirement of dense output is motivated by modern applications of stereo such as view synthesis and image-based rendering, which require disparity estimates in all image regions, even those that are occluded or without texture. Thus, sparse and feature-based stereo methods are outside the scope of this paper, unless they are followed by a surface-fitting step, e.g., using triangulation, splines, or seed-and-grow methods.

Our work is motivated by a similar study of optical flow algorithms by Barron *et al.* [5]. In stereo correspondence, two previous comparative papers have focused on the performance of sparse feature matchers [35, 15]. Two recent papers [69, 49] have developed new criteria for evaluating the performance of dense stereo matchers for image-based rendering and tele-presence applications. This work is a continuation of the investigations begun by Szeliski and Zabih [72], which compared the performance of several popular algorithms, but did not provide a detailed taxonomy or as complete a coverage of algorithms.

We begin this paper with a discussion of the assumptions and representations of stereo algorithms. In Section 3, we present our taxonomy of dense two-frame correspondence algorithms. Sections 4 and 5 discuss our implementation and our evaluation methodology. We highlight and discuss the most interesting subset of our results in Section 6 and conclude with a discussion of planned future work.

2. Assumptions and representations

Any vision algorithm, explicitly or implicitly, makes assumptions about the physical world and the image formation process. For example, how does the algorithm measure the evidence that points in the two images *match*, i.e., that they are projections of the same scene point? Common assumptions are Lambertian surfaces, i.e., surfaces whose appear-

ance does not vary with viewpoint. Some algorithms also model specific kinds of camera noise, or differences in gain or bias.

Equally important are assumptions about the world or scene geometry, and the visual appearance of objects. Assuming that the physical world consists of piecewise-smooth surfaces, algorithms have built-in smoothness assumptions (often implicit) without which the correspondence problem would be underconstrained and ill-posed. Our taxonomy of stereo algorithms, presented in Section 3, examines both matching assumptions and smoothness assumptions in order to categorize existing stereo methods.

Finally, most algorithms make assumptions about camera calibration and epipolar geometry. This is arguably the best-understood part of stereo vision; we therefore assume in this paper that we are given a pair of rectified images as input.

A second critical issue in understanding an algorithm is the representation used internally and output externally by the algorithm. Most stereo correspondence methods compute a univalued disparity function $d(x, y)$ with respect to a reference image, which could be one of the input images, or a "cyclopian" view in between some of the images.

Other approaches, in particular multi-view stereo methods, use multi-valued [71], voxel-based [65, 45], or layer-based [79, 3] representations. Still other approaches use full 3D models such as deformable models [75], triangulated meshes [27], or level-set methods [25].

Since our goal is to compare a large number of methods within one common framework, we have chosen to focus on techniques that produce a univalued *disparity map* $d(x, y)$ as their output. Central to such methods is the concept of a *disparity space* (x, y, d). In computer vision, disparity is often treated as synonymous with inverse depth [16, 54]. More recently, several researchers have defined disparity as a three-dimensional projective transformation of 3-D space (X, Y, Z)[21, 71]. In general, we favor this more generalized interpretation of disparity, since it allows the adaptation of the search space to the geometry of the input cameras. In this study, however, all of our images are taken on a linear path with the optical axis perpendicular to the camera displacement, and the classical inverse-depth interpretation will suffice [54]. Furthermore, in order to be able to compare the disparity estimates produced by different pairings of images, we define disparity to be the horizontal displacement between *successive* images (our images are taken left-to-right at regular displacements). The (x, y) coordinates of the disparity space are taken to be coincident with the pixel coordinates of a *reference image* chosen from our input data set. The correspondence between a pixel (x, y) in reference image r and a pixel (x', y') in matching image m is then given by

$$x' = x + (r - m)d(x, y), \quad y' = y. \tag{1}$$

Once the disparity space has been specified, we can introduce the concept of a *disparity space image* or DSI [81, 14]. In general, a DSI is any image or function defined over a continuous or discretized version of disparity space (x, y, d). In practice, the DSI usually represents the confidence or log likelihood (i.e., *cost*) of a particular match implied by $d(x, y)$.

The goal of a stereo correspondence algorithm is then to produce a univalued function in disparity space $d(x, y)$ that best describes the shape of the surfaces in the scene.

3. A taxonomy of stereo algorithms

In order to support an informed comparison of stereo matching algorithms, we develop in this section a taxonomy and categorization scheme for such algorithms. We present a set of algorithmic "building blocks" from which a large set of existing algorithms can easily be constructed. Our taxonomy is based on the observation that stereo algorithms generally perform (subsets of) the following four steps [63, 62]: (1) matching cost computation; (2) cost (support) aggregation; (3) disparity computation / optimization; and (4) disparity refinement. The actual sequence of steps taken depends on the specific algorithm.

For example, *local* (window-based) algorithms, where the disparity computation at a given point depends only on intensity values within a finite window, usually make implicit smoothness assumptions by aggregating support.

On the other hand, *global* algorithms make explicit smoothness assumptions and then solve an optimization problem. Such algorithms typically do not perform an aggregation step, but rather seek a disparity assignment (step 3) that minimizes a global cost function that combines data (step 1) and smoothness terms. The main distinction between these algorithms is the minimization procedure used, e.g., simulated annealing [47, 4], probabilistic (mean-field) diffusion [63], or graph cuts [18].

In between these two broad classes are certain iterative algorithms that do not explicitly state a global function that is to be minimized, but whose behavior mimics closely that of iterative optimization algorithms [46, 63, 83]. Hierarchical (coarse-to-fine) algorithms resemble such iterative algorithms, but typically operate on an image pyramid [80, 58, 7].

3.1. Matching cost computation

The most common pixel-based matching costs include *squared intensity differences* (SSD) [34, 1, 48, 68], and *absolute intensity differences* (SAD) [39].

More recently, robust measures, including truncated quadratics and contaminated Gaussians have been proposed [11, 12, 63]. These measures are useful because they limit the influence of mismatches during aggregation.

Other traditional matching costs include normalized cross-correlation [34, 60, 15], which behaves similar to sum-of-squared-differences (SSD), and binary matching costs

(i.e., match / no match) [46], based on binary features such as edges [33] or the sign of the Laplacian [51]. Binary matching costs are not commonly used in dense stereo methods, however.

Some costs are insensitive to differences in camera gain or bias, for example gradient-based measures [61], and non-parametric measures, such as rank and census transforms [82]. Of course, it is also possible to correct for different camera characteristics by performing a preprocessing step for bias-gain or histogram equalization [32, 23]. Other matching criteria include phase and filter-bank responses [43, 37, 38]. Finally, Birchfield and Tomasi have proposed a matching cost that is insensitive to image sampling [8].

The matching cost values over all pixels and all disparities form the initial disparity space image $M_0(x, y, d)$. While our study is currently restricted to two-frame methods, the initial DSI can easily incorporate information from more than two images by simply summing up the cost values for each matching image m, since the DSI is associated with a fixed reference image r (Equation (1)). This is the idea behind multiple-baseline SSSD and SSAD methods [54, 42, 50]. As mentioned in Section 2, this idea can be generalized to arbitrary camera configurations [21, 71].

3.2. Aggregation of cost

Local and window-based methods aggregate the matching cost by summing or averaging over a *support region* in the DSI $M(x, y, d)$. A support region can be either two-dimensional at a fixed disparity, or three-dimensional in x-y-d space. Two-dimensional evidence aggregation has been implemented using square windows or Gaussian convolution (traditional), multiple windows anchored at different points (shiftable windows) [2, 14], windows with adaptive sizes [53, 40, 78, 41], and windows based on connected components of constant disparity [17]. Three-dimensional support functions that have been proposed include limited disparity difference [33], limited disparity gradient [56], and Prazdny's coherence principle [57].

Aggregation with a fixed support region can be performed using 2D or 3D convolution, or, in the case of rectangular windows, using efficient (moving average) box-filters. Shiftable windows can also be implemented efficiently using a separable sliding min-filter. A different method of aggregation is *iterative diffusion*, i.e., an aggregating (or averaging) operation that is implemented by repeatedly adding to each pixel's cost the (weighted) values of its neighboring pixels' costs [66, 63].

3.3. Disparity computation and optimization

Local methods. In local methods, the emphasis is on the matching cost computation and on the cost aggregation steps. Computing the final disparities is trivial: simply choose at each pixel the disparity associated with the minimum cost value. Thus, these methods perform a local "winner-take-all" (WTA) optimization at each pixel.

Global optimization. In contrast, global methods perform almost all of their work during the disparity computation phase, and often skip the aggregation step. Many global methods are formulated in an energy-minimization framework [74]. The objective is to find a disparity function d that minimizes a global energy,

$$E(d) = E_{\mathit{data}}(d) + \lambda E_{\mathit{smooth}}(d). \tag{2}$$

The data term, $E_{\mathit{data}}(d)$, measures how well the disparity function d agrees with the input image pair. Using the disparity space formulation,

$$E_{\mathit{data}}(d) = \sum_{(x,y)} M(x, y, d(x, y)), \tag{3}$$

where M is the (initial or aggregated) matching cost DSI.

The smoothness term $E_{\mathit{smooth}}(d)$ encodes the smoothness assumptions made by the algorithm. To make the optimization computationally tractable, the smoothness term is often restricted to only measuring the differences between neighboring pixels' disparities,

$$E_{\mathit{smooth}}(d) = \sum_{(x,y)} \rho(d(x, y) - d(x+1, y)) + \rho(d(x, y) - d(x, y+1)), \tag{4}$$

where ρ is some monotonically increasing function of disparity. In regularization-based vision [55], ρ is a quadratic function, which makes d smooth everywhere, and may lead to poor results at object boundaries. Energy functions that do not have this problem are called *discontinuity-preserving*, and are based on robust ρ functions [74, 12, 63]. Geman and Geman's seminal paper [31] gave a Bayesian interpretation of these kinds of energy functions and proposed a discontinuity-preserving energy function based on Markov Random Fields (MRFs) and additional *line processes*. Black and Rangarajan [12] show how line processes can be often be subsumed by a robust regularization framework.

The terms in E_{smooth} can also depend on the intensity differences, e.g.,

$$\rho_d(d(x, y) - d(x+1, y)) \cdot \rho_I(\|I(x, y) - I(x+1, y)\|). \tag{5}$$

This idea [28, 26, 14, 18] encourages disparity discontinuities to coincide with intensity/color edges, and appears to account for some of the good performance of global optimization approaches.

Once the global energy has been defined, a variety of algorithms can be used to find a (local) minimum. The traditional approaches associated with regularization and Markov Random Fields include continuation [13], simulated annealing [31, 47, 4], highest confidence first [20], and mean-field annealing [29].

More recently, *max-flow* and *graph-cut* methods have been proposed to solve a special class of global optimization problems [59, 36, 18, 77, 44]. Such methods are more efficient than simulated annealing, and have produced good results.

Dynamic programming. A different class of global optimization algorithms are those based on *dynamic programming*. While the 2D-optimization of equation (2) can be shown to be NP-hard for common classes of smoothness functions [77], dynamic programming can find the global minimum for independent scanlines in polynomial time. Recent approaches have focused on the dense (intensity-based) scanline optimization problem [6, 30, 22, 14, 9]. These approaches work by computing the minimum cost path through the matrix of all pairwise matching costs between two corresponding scanlines. Partial occlusion is handled explicitly by assigning a group of pixels in one image to a single pixel in the other image.

Problems with dynamic programming stereo include the selection of the right cost for occluded pixels and the difficulty of enforcing inter-scanline consistency, although several methods propose ways of addressing the latter [52, 14]. Another problem is that the dynamic programming approach requires enforcing the *monotonicity* or *ordering constraint*. This constraint requires that the relative ordering of pixels on a scanline remain the same between the two views, which may not be the case in scenes containing narrow foreground objects.

Cooperative algorithms. Finally, *cooperative* algorithms, inspired by computational models of human stereo vision, were among the earliest methods proposed for disparity computation [46]. Such algorithms iteratively perform local computations, but use nonlinear operations that result in an overall global behavior similar to global optimization algorithms. In fact, for some of these algorithms, it is possible to explicitly state a global function that is being minimized [63]. Recently, a promising variant of Marr and Poggio's original cooperative algorithm has been proposed [83].

3.4. Refinement of disparities

Most stereo correspondence algorithms compute a set of disparity estimates in some discretized space, e.g., for integer disparities (exceptions include continuous optimization techniques such as optic flow [7] or splines [70]). For applications such as robot navigation or people tracking, these may be perfectly adequate. However for image-based rendering, such quantized maps lead to very unappealing view synthesis results (the scene appears to be made up of many thin shearing layers). To remedy this situation, sub-pixel disparity estimates can be computed in a variety of ways, including iterative gradient descent and fitting a curve to the matching costs at discrete disparity levels [76, 48, 40]. This provides an easy way to increase the resolution of a stereo algorithm with little additional computation. However, to work well, the intensities being matched must vary smoothly, and the regions over which these estimates are computed must be on the same (correct) surface.

Recently, some questions have been raised about the advisability of fitting correlation curves to integer-sampled matching costs [67]. This situation may even be worse when sampling-insensitive dissimilarity measures are used [8]. This is an area we plan to investigate in the future.

3.5. Other methods

Not all dense two-frame stereo correspondence algorithms can be described in terms of our basic taxonomy and representations. The algorithms described in this paper first enumerate all possible matches at all possible disparities, then select the best set of matches in some way. This is a useful approach when a large amount of ambiguity may exist in the computed disparities. An alternative approach is to use methods inspired by classic (infinitesimal) optic flow computation. Here, images are successively warped and motion estimates incrementally updated until a satisfactory registration is achieved. These techniques are most often implemented within a coarse-to-fine hierarchical refinement framework [58, 7, 5, 70].

A univalued representation of the disparity map is also not essential. Multi-valued representations, which can represent several depth values along each line of sight, have been extensively studied recently, especially for large multi-view data set. Many of these techniques use a *voxel-based* representation to encode the reconstructed colors and spatial occupancies or opacities [71, 65, 45]. Another way to represent a scene with more complexity is to use multiple layers, each of which can be represented by a plane plus residual parallax [3, 10, 73]. Finally, deformable surfaces of various kinds have also been used to perform 3D shape reconstruction from multiple images [75, 27, 25].

4. Implementation

We have developed a stand-alone, portable C++ implementation of several stereo algorithms. The implementation is closely tied to the taxonomy presented in Section 3, and currently includes window-based algorithms, diffusion algorithms, as well as global optimization methods using dynamic programming and graph cuts. While many published methods include special features and post-processing steps to improve the results, we have chosen to implement the basic versions of such algorithms, in order to assess their respective merits most directly.

The implementation is modular, and can easily be extended to include other algorithms or their components. We plan to add several other algorithms in the near future, and we hope that other authors will contribute their methods to

our framework as well. Once a new algorithm has been integrated, it can easily be compared with other algorithms using our evaluation module, which can measure disparity error and reprojection error (Section 5). The implementation contains a sophisticated mechanism for specifying parameter values that support recursive script files for easy performance comparisons on multiple data sets.

Due to space limitations, we can only include a brief discussion of the optimization module. For a detailed discussion of the matching cost computation, aggregation, and disparity refinement modules, please see the full version of this paper [64].

Given (optionally aggregated) costs, the optimization module computes the winning disparities using one of several algorithms. In this paper we report on results of the following optimization methods:

- winner-take-all (WTA);
- dynamic programming (DP);
- scanline optimization (SO);
- graph cut (GC).

The winner-take-all method simply picks the lowest (aggregated) matching cost as the selected disparity at each pixel. The other methods require (in addition to the matching cost) the definition of a smoothness cost. Prior to invoking one of the optimization algorithms, we set up some tables reflecting the values of ρ_d in (5) and precompute the spatially varying weights $\rho_I(x, y)$. These two tables are controlled by the parameters λ, which controls the overall scale of the smoothness term, and γ, which controls the dependence on the intensity gradient (difference). We currently use

$$\rho_I(\Delta I) = \frac{1}{1 + \gamma|\Delta I|}. \tag{6}$$

We can thus ensure that all of the optimization algorithms are minimizing the same objective function, while enabling a more meaningful comparison of their performance.

Our first global optimization technique, DP, is a dynamic programming method similar to the one proposed by [14]. The algorithm works by computing the minimum-cost path through each x-d slice in the DSI . Every point in this slice can be in one of three states: M (match), L (left-visible only), or R (right-visible only). Assuming the ordering constraint is being enforced, a valid path can take at most three directions at a point, each associated with a deterministic state change. Using dynamic programming, the minimum cost of all paths to a point can be accumulated efficiently. Points in state M are simply charged the matching cost at this point in the DSI. Points in states L and R are charged a fixed *occlusion cost* `opt_ocst`.

Our second global optimization technique, *scanline optimization* (SO), is a simple (and, to our knowledge, novel) approach designed to assess different smoothness terms. Like the previous method, it operates on individual x-d DSI slices and optimizes one scanline at a time. However, the method is asymmetric and does not utilize visibility or ordering constraints. Instead, a d value is assigned at each point x such that the overall cost along the scanline is minimized. (Note that without a smoothness term, this would be equivalent to a winner-take-all optimization.) The global minimum can again be computed using dynamic programming; however, unlike in traditional (symmetric) DP algorithms, the ordering constraint does not need to be enforced, and no occlusion cost parameter is necessary. Thus, the SO algorithm solves the same optimization problem as the graph-cut algorithm described below, except that vertical smoothness terms are ignored.

Both DP and SO algorithms suffer from the well-known difficulty of enforcing inter-scanline consistency, resulting in horizontal "streaks" in the computed disparity map. Bobick and Intille's approach to this problem is to detect edges in the DSI slice, and to lower the occlusion cost for paths along those edges. This has the effect of aligning depth discontinuities with intensity edges. In our implementation, we achieve the same goal by using an intensity-dependent smoothness cost (Equation (5)), which, in our DP algorithm, is charged at all L-M and R-M state transitions.

Our final global optimization method, GC, implements the α-β swap move algorithm described in [18, 77]. We randomize the α-β pairings at each (inner) iteration, and stop the algorithm when no further (local) energy improvements are possible.

5. Evaluation methodology

To evaluate the performance of a stereo algorithm or the effects of varying some of its parameters, we need a quantitative way to estimate the quality of the computed correspondences. Two general approaches to this are to compute error statistics with respect to some ground truth data [5] and to evaluate the synthetic images obtained by warping the reference or unseen images by the computed disparity map [69].

In the current version of our software, we compute the following two quality measures based on known ground truth data:

1. RMS (root-mean-squared) error (measured in disparity units) between the computed depth map $d_C(x, y)$ and the ground truth map $d_T(x, y)$, i.e.,

$$E = \left(\frac{1}{N} \sum_{(x,y)} |d_C(x, y) - d_T(x, y)|^2 \right)^{\frac{1}{2}}, \tag{7}$$

   where N is the total number of pixels.

2. Percentage of bad matching pixels,

$$P = \frac{1}{N} \sum_{(x,y)} (|d_C(x,y) - d_T(x,y)| > \delta_d), \qquad (8)$$

where δ_d is a disparity error tolerance. In our current set of experiments, we use $\delta_d = 1$.

In addition to computing these statistics over the whole image, we also focus on three different kinds of regions. These regions are computed by pre-processing the reference image and ground truth disparity map to yield the following three binary segmentations:

- textureless regions $\mathcal{T}$ (locations with low horizontal intensity gradient);
- occluded regions $\mathcal{O}$; and
- depth discontinuity regions $\mathcal{D}$ (pixels in the vicinity of a disparity jump).

These regions were selected to support the analysis of matching results in typical problem areas. The statistics described above are computed for each of the three regions and their complements.

The second major approach to gauging the quality of reconstruction algorithms is to use the color images and disparity maps to predict the appearance of other views [69]. For a discussion and results of this *prediction error* methodology, please see the full version of this paper [64].

To quantitatively evaluate our correspondence algorithms, we require data sets that either have a ground truth disparity map, or a set of additional views that can be used for prediction error test (or preferably both).

We have begun to build such a database of images, building upon the methodology introduced in [72]. We take our images at regular intervals with a camera mounted on a horizontal translation stage, with the camera pointing perpendicularly to the direction of motion. We use a digital high-resolution camera (Canon G1) set in manual exposure and focus mode, and rectify the images using tracked feature points. We then downsample the original 2048×1536 images to 512×384 using a high-quality 8-tap filter.

All of the sequences we have captured are made up of piecewise planar objects (typically posters or paintings, some with cut-out edges). Before downsampling the images, we hand-label each image into its piecewise planar components (Figure 1). We then use a direct alignment technique on each planar region [3] to estimate the affine motion of each patch. The horizontal component of these motions is then used to compute the ground truth disparity.

In addition to these novel sequences, we have also been using the University of Tsukuba data set [50], which contains 5×5 arrays of translated images together with hand-labeled integer disparity ground truth images. We plan to extend our acquisition methodology to handle scenes with quadric surfaces (cylinders, cones, spheres, ...).

Figure 1: *Sample (reference) images with planar region labeling and computed ground truth disparities.*

6. Experiments and results

In this section, we highlight a subset of our results. The experimental study we are performing compares all the algorithms we have implemented with the following goals in mind:

1. Focus on common problem areas for stereo algorithms. These include textureless areas, object boundaries and the "foreground fattening" effect, occluded areas, and thin objects that violate the ordering constraint.

2. Isolate the effect of the smoothness cost function in global optimization methods. In particular, we are studying the effect of smoothness functions with and without an intensity gradient term that favors disparity discontinuities to align with intensity edges.

3. Isolate the effect of the matching cost, by comparing different cost functions, including absolute differences, squared differences, truncated versions of both, and Birchfield and Tomasi's [8] measure. These cost functions will be used in several algorithms whose other parameters are held constant.

4. Investigate prediction error as an evaluation criterion, by correlating both forward and backward prediction error with disparity error, to see whether prediction error is a valid evaluation method for data sets where ground truth is not available.

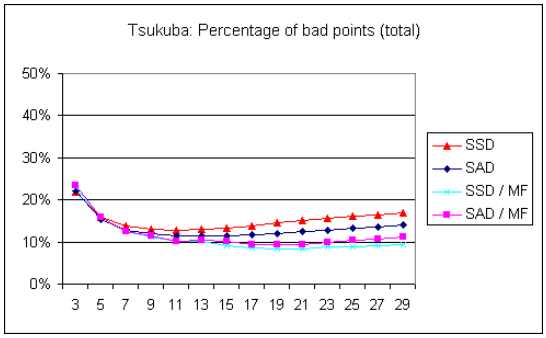
For the complete set of results, please see the tech-report version of this paper [64]. The results, together with our implementation and the data sets we created, are also available on the Web at `www.middlebury.edu/stereo`. Here we report on initial experiments that address goals 1 and 2. We focus first on local window-based algorithms, then on global optimization methods.

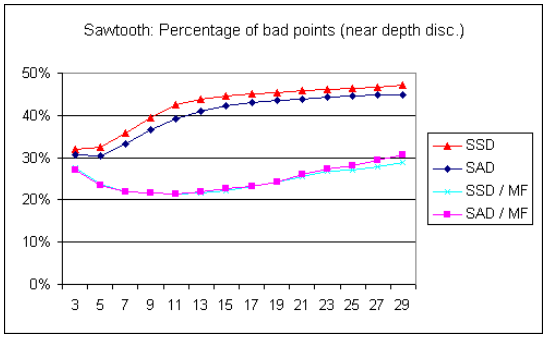
6.1. Window-based algorithms

Our first set of experiments measures the effect of window size for SAD and SSD algorithms, and investigates the use of a min-filter (i.e., shiftable window). Figure 2 shows various plots for the Tsukuba data set. We report the percentage of bad points here; RMS disparity errors behave qualitatively very similar. The first plot compares SSD, SAD, SSD with min-filter, and SAD with min-filter by examining the percentage of bad points over the entire image. The curves do not differentiate clearly among the algorithms, but indicate that the min-filter versions generally yield better results. They also seem to indicate that fairly large window sizes yield best results. In the second plot we evaluate only image regions near discontinuities. In these regions, the better performance of the min-filter versions is clearly noticeable, and also a preference for smaller window sizes. The disparity map for the minimizing window size in the second plot is shown at the bottom. In the third plot we select one algorithm (SAD with min-filter), and show how the error measure depends on the type of image region. It can clearly be seen that error in textureless regions decreases with larger window sizes, while the opposite is true for regions near discontinuities.

These experiments expose some of the fundamental limitations of local methods. While large windows are needed to avoid wrong matches in regions with little texture, window-based stereo methods perform poorly near object boundaries (i.e., depth discontinuities). The reason is that such methods implicitly assume that all points within a window have similar disparities. If a window straddles a depth boundary, some points in the window match at the foreground disparity, while others match at the background disparity. The (aggregated) cost function at a point near a depth discontinuity is thus bimodal in the d direction, and stronger of the two modes will be selected as the winning disparity. Which one of the two modes will win? This depends on the amount of (horizontal) texture present in the two regions. Consider first a purely horizontal depth discontinuity. Whichever of the two regions has more horizontal texture will create a stronger mode, and the computed disparities will thus "bleed" into the less-textured region. For non-horizontal depth boundaries, however, the most prominent horizontal texture is usually the object boundary itself, since different objects typically have different colors and intensities. Since the object boundary is at the foreground disparity, a strong preference for the foreground disparity at points near the boundary is created, even if the background is textured. This is the explanation for the well-known "foreground fattening" effect exhibited by window-based algorithms.

Adaptive window methods have been developed to combat this problem. The simplest variant, shiftable windows (min-filters) can be effective as is shown in the above experiment. Shiftable windows can recover object boundaries





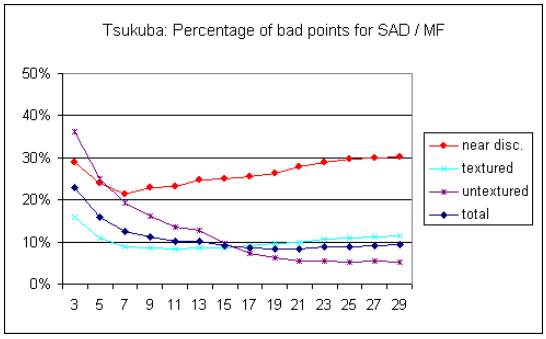


Figure 2: *Evaluation measures for Tsukuba image sequence. All plots report the percentage of bad points vs. the window size. Top plot: overall comparison of SSD and SAD, with and without min-filter (MF). Second plot: same comparison, but only for points near depth discontinuities. Third plot: comparison of points in different image regions for SAD with min-filter. Bottom: best disparity map for SAD / MF (selected based on second plot), with window size 7.*

Algorithm	RMS err. E	bad pixels P	occluded $P_{\mathcal{O}}$	textureless $P_{\mathcal{T}}$
SAD	1.988	12.87 %	89.47 %	17.60 %
SAD / MF	1.983	12.43 %	74.52 %	19.18 %
DP	1.793	9.52 %	33.68 %	9.28 %
SO	1.657	9.76 %	28.25 %	12.63 %
GC	1.537	6.46 %	30.75 %	3.33 %

Table 1: *Error statistics for 5 different algorithms run on the University of Tsukuba data sets. Notice how the errors in occluded and textureless regions are significantly lower for the global algorithms, in particular the graph cut algorithm. Parameters: SAD, SAD/MF: window size $W = 7$; other methods (DP, SO, GC): see Figure 3.*

quite accurately if both foreground and background regions are textured, and as long as the window fits as a whole within the foreground object. The size of the min-filter should be chosen to match the window size. As all local methods, however, shiftable windows fail in textureless areas, and they can even "amplify" bad matches.

6.2. Global algorithms

We have also performed experiments using our implementations of dynamic programming (DP), scanline optimization (SO), and graph cuts (GC). Table 1 summarizes the error statistics for the best run of each algorithm, and compares them to SAD and SAD/MF. It can clearly be seen that the global algorithms outperform the local ones, in particular in textureless regions. Figure 3 shows the corresponding disparity maps. All three global algorithms perform quite well, but both DP and SO show the "streaking" characteristic for scanline-based algorithms. The graph-cut algorithm performs best, both quantitatively and qualitatively. It should be noted, however, that the algorithms are currently fairly sensitive to the tuning of the smoothness cost, in particular to parameters λ and γ in Equation 6.

Figure 3: *"Best" disparity maps computed by the global algorithms. Top: Dynamic programming (DP),* $\lambda = 100$, $\gamma = 1$, `opt_ocst`$= 60$. *Middle: Scanline optimization (SO),* $\lambda = 100$, $\gamma = 0$. *Bottom: Graph cut algorithm (GC),* $\lambda = 1000$, $\gamma = 2$.

7. Conclusion

In this paper, we have proposed a taxonomy for dense two-frame stereo correspondence algorithms. We use this taxonomy to highlight the most important features of existing stereo algorithms, and to study important algorithmic components in isolation. We have implemented a suite of stereo matching algorithm components, and constructed a test harness that can be used to combine these, to vary the algorithm parameters in a controlled way, and to test the performance of these algorithm on interesting data sets. We have also produced some new calibrated multi-view stereo data sets with hand-labeled ground truth. We have started an extensive experimental investigation in order to assess the individual value of the different algorithmic components. The experiments reported here have demonstrated the limitations of local methods, and have started to assess the value of different global techniques.

There are some other open questions we would like to address: How important is it to devise the right cost function in global optimization algorithms vs. how important is it to find a global minimum? What kind of adaptive/shiftable windows work best? Are error measures that are insensitive to integral shifts also good for sub-pixel refinement, or do we need to use quadratic energy measures for these to work?

Our current plan is that, by publishing this study along with our sample code and data sets on the Web, other stereo researchers will run their algorithms on our data and allow us to report their results. Even better, we hope that some researchers take the time to produce implementation of their algorithms compatible with our framework available for oth-

ers to use and to build upon. We would be thrilled if some set of standard data and testing methodology were to become an accepted standard in the stereo correspondence community, so that new algorithms would have to pass a "litmus test" to demonstrate that they improve on the state of the art.

By building on the framework and methodology developed in this paper, we will hopefully reach a deeper understanding of the complex behavior of stereo correspondence algorithms. Only once the representations become rich enough to capture the full complexity of the visual world will image-based modeling fulfill its promise of accurately capturing and replaying the complete appearance of interesting, complex scenes and objects.

Acknowledgements

Thanks to Y. Ohta and Y. Nakamura for supplying the ground-truth imagery from the University of Tsukuba. Thanks also to Padma Ugbabe for helping to label the image regions, and to Fred Lower for providing his paintings for our image data sets. This research was supported in part by NSF CAREER grant 9984485.

References

[1] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *IJCV*, 2(3):283–310, 1989.

[2] R. D. Arnold. Automated stereo perception. Technical Report AIM-351, AI Lab, Stanford University, 1983.

[3] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *CVPR*, pp. 434–441, 1998.

[4] S. T. Barnard. Stochastic stereo matching over scale. *IJCV*, 3(1):17–32, 1989.

[5] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *IJCV*, 12(1):43–77, 1994.

[6] P. N. Belhumeur. A Bayesian approach to binocular stereopsis. *IJCV*, 19(3):237–260, 1996.

[7] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *ECCV*, pp. 237–252, 1992.

[8] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. In *ICCV*, pp. 1073–1080, 1998.

[9] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE TPAMI*, 20(4):401–406, 1998.

[10] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *ICCV*, pp. 489–495, 1999.

[11] M. J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *ICCV*, pp. 231–236, 1993.

[12] M. J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *IJCV*, 19(1):57–91, 1996.

[13] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, MA, 1987.

[14] A. F. Bobick and S. S. Intille. Large occlusion stereo. *IJCV*, 33(3):181–200, 1999.

[15] R. C. Bolles, H. H. Baker, and M. J. Hannah. The JISCT stereo evaluation. In *IUW*, pp. 263–274, 1993.

[16] R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *IJCV*, 1:7–55, 1987.

[17] Y. Boykov, O. Veksler, and R. Zabih. A variable window approach to early vision. *IEEE TPAMI*, 20(12):1283–1294, 1998.

[18] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *ICCV*, pp. 377–384, 1999.

[19] L. G. Brown. A survey of image registration techniques. *Computing Surveys*, 24(4):325–376, 1992.

[20] P. B. Chou and C. M. Brown. The theory and practice of Bayesian image labeling. *IJCV*, 4(3):185–210, 1990.

[21] R. T. Collins. A space-sweep approach to true multi-image matching. In *CVPR*, pp. 358–363, 1996.

[22] I. J. Cox, S. L. Hingorani, S. B. Rao, and B. M. Maggs. A maximum likelihood stereo algorithm. *CVIU*, 63(3):542–567, 1996.

[23] I. J. Cox, S. Roy, and S. L. Hingorani. Dynamic histogram warping of image pairs for constant image brightness. In *ICIP*, vol. 2, pp. 366–369, 1995.

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

[25] O. Faugeras and R. Keriven. Variational principles, surface evolution, PDE's, level set methods, and the stereo problem. *IEEE Trans. Image Proc.*, 7(3):336–344, 1998.

[26] P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 6:35–49, 1993.

[27] P. Fua and Y. G. Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *IJCV*, 16:35–56, 1995.

[28] E. Gamble and T. Poggio. Visual integration and detection of discontinuities: the key role of intensity edges. A. I. Memo 970, AI Lab, MIT, 1987.

[29] D. Geiger and F. Girosi. Mean field theory for surface reconstruction. *IEEE TPAMI*, 13(5):401–412, 1991.

[30] D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. In *ECCV*, pp. 425–433, 1992.

[31] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE TPAMI*, 6(6):721–741, 1984.

[32] M. A. Gennert. Brightness-based stereo matching. In *ICCV*, pp. 139–143, 1988.

[33] W. E. L. Grimson. Computational experiments with a feature based stereo algorithm. *IEEE TPAMI*, 7(1):17–34, 1985.

[34] M. J. Hannah. *Computer Matching of Areas in Stereo Images*. PhD thesis, Stanford University, 1974.

[35] Y. C. Hsieh, D. McKeown, and F. P. Perlant. Performance evaluation of scene registration and stereo matching for cartographic feature extraction. *IEEE TPAMI*, 14(2):214–238, 1992.

[36] H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *ECCV*, pp. 232–248, 1998.

[37] M. R. M. Jenkin, A. D. Jepson, and J. K. Tsotsos. Techniques for disparity measurement. *CVGIP: IU*, 53(1):14–30, 1991.

[38] D. G. Jones and J. Malik. A computational framework for determining stereo correspondence from a set of linear spatial filters. In *ECCV*, pp. 395–410, 1992.

[39] T. Kanade. Development of a video-rate stereo machine. In *IUW*, pp. 549–557, 1994.

[40] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE TPAMI*, 16(9):920–932, 1994.

[41] S. B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *CVPR*, 2001.

[42] S. B. Kang, J. Webb, L. Zitnick, and T. Kanade. A multi-baseline stereo system with active illumination and real-time image acquisition. In *ICCV*, pp. 88–93, 1995.

[43] M. Kass. Linear image features in stereopsis. *IJCV*, 1(4):357–368, 1988.

[44] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *ICCV*, vol. II, pp. 508–515, 2001.

[45] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *IJCV*, 38(3):199–218, 2000.

[46] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194:283–287, 1976.

[47] J. Marroquin, S. Mitter, and T. Poggio. Probabilistic solution of ill-posed problems in computational vision. *J. Am. Stat. Assoc.*, 82(397):76–89, 1987.

[48] L. Matthies, R. Szeliski, and T. Kanade. Kalman filter-based algorithms for estimating depth from image sequences. *IJCV*, 3:209–236, 1989.

[49] J. Mulligan, V. Isler, and K. Daniilidis. Performance evaluation of stereo for tele-presence. In *ICCV*, vol. II, pp. 558–565, 2001.

[50] Y. Nakamura, T. Matsuura, K. Satoh, and Y. Ohta. Occlusion detectable stereo — occlusion patterns in camera matrix. In *CVPR*, pp. 371–378, 1996.

[51] H. K. Nishihara. Practical real-time imaging stereo matcher. *Optical Engineering*, 23(5):536–545, 1984.

[52] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE TPAMI*, 7(2):139–154, 1985.

[53] M. Okutomi and T. Kanade. A locally adaptive window for signal matching. *IJCV*, 7(2):143–162, 1992.

[54] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE TPAMI*, 15(4):353–363, 1993.

[55] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317(6035):314–319, 1985.

[56] S. B. Pollard, J. E. W. Mayhew, and J. P. Frisby. PMF: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470, 1985.

[57] K. Prazdny. Detection of binocular disparities. *Biological Cybernetics*, 52(2):93–99, 1985.

[58] L. H. Quam. Hierarchical warp stereo. In *IUW*, pp. 149–155, 1984.

[59] S. Roy and I. J. Cox. A maximum-flow formulation of the N-camera stereo correspondence problem. In *ICCV*, pp. 492–499, 1998.

[60] T. W. Ryan, R. T. Gray, and B. R. Hunt. Prediction of correlation errors in stereo-pair images. *Optical Engineering*, 19(3):312–322, 1980.

[61] D. Scharstein. Matching images by comparing their gradient fields. In *ICPR*, vol. 1, pp. 572–575, 1994.

[62] D. Scharstein. *View Synthesis Using Stereo Vision*, LNCS vol. 1583. Springer-Verlag, 1999.

[63] D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. *IJCV*, 28(2):155–174, 1998.

[64] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Technical Report MSR-TR-2001-81, Microsoft Research, 2001.

[65] S. M. Seitz and C. M. Dyer. Photorealistic scene reconstruction by voxel coloring. *IJCV*, 35(2):1–23, 1999.

[66] J. Shah. A nonlinear diffusion model for discontinuous disparity and half-occlusion in stereo. In *CVPR*, pp. 34–40, 1993.

[67] M. Shimizu and M. Okutomi. Precise sub-pixel estimation on area-based matching. In *ICCV*, vol. I, pp. 90–97, 2001.

[68] E. P. Simoncelli, E. H. Adelson, and D. J. Heeger. Probability distributions of optic flow. In *CVPR*, pp. 310–315, 1991.

[69] R. Szeliski. Prediction error as a quality metric for motion and stereo. In *ICCV*, pp. 781–788, 1999.

[70] R. Szeliski and J. Coughlan. Hierarchical spline-based image registration. *IJCV*, 22(3):199–218, 1997.

[71] R. Szeliski and P. Golland. Stereo matching with transparency and matting. *IJCV*, 32(1):45–61, 1999.

[72] R. Szeliski and R. Zabih. An experimental comparison of stereo algorithms. In *Intl. Workshop on Vision Algs.*, pp. 1–19, 1999.

[73] H. Tao, H. Sawhney, and R. Kumar. A global matching framework for stereo computation. In *ICCV*, vol. I, pp. 532–539, 2001.

[74] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE TPAMI*, 8(4):413–424, 1986.

[75] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: Deformable superquadrics. *IEEE TPAMI*, 13(7):703–714, 1991.

[76] Q. Tian and M. N. Huhns. Algorithms for subpixel registration. *CVGIP*, 35:220–233, 1986.

[77] O. Veksler. *Efficient Graph-based Energy Minimization Methods in Computer Vision*. PhD thesis, Cornell University, 1999.

[78] O. Veksler. Stereo matching by compact windows via minimum ratio cycle. In *ICCV*, vol. I, pp. 540–547, 2001.

[79] J. Y. A. Wang and E. H. Adelson. Layered representation for motion analysis. In *CVPR*, pp. 361–366, 1993.

[80] A. Witkin, D. Terzopoulos, and M. Kass. Signal matching through scale space. *IJCV*, 1:133–144, 1987.

[81] Y. Yang, A. Yuille, and J. Lu. Local, global, and multilevel stereo matching. In *CVPR*, pp. 274–279, 1993.

[82] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *ECCV*, vol. II, pp. 151–158, 1994.

[83] C. L. Zitnick and T. Kanade. A cooperative algorithm for stereo matching and occlusion detection. *IEEE TPAMI*, 22(7):675–684, 2000.