Stereo Matching—State-of-the-Art and Research Challenges

Chapter · January 2013

DOI: 10.1007/978-1-4471-5520-1_6

2 authors:

Michael Bleyer
Microsoft
39 PUBLICATIONS **2,336** CITATIONS
SEE PROFILE

Christian Breiteneder
TU Wien
114 PUBLICATIONS **998** CITATIONS
SEE PROFILE

Stereo Matching – State-of-the-Art and Research Challenges

Michael Bleyer and Christian Breiteneder

Abstract Stereo matching denotes the problem of finding dense correspondences in pairs of images in order to perform 3D reconstruction. In this chapter, we provide a review of stereo methods with a focus on recent developments and our own work. We start with a discussion of local methods and introduce our algorithms: geodesic stereo, cost filtering and PatchMatch stereo. Although local algorithms have recently become very popular, they are not capable of handling large untextured regions where a global smoothness prior is required. In the discussion of such global methods, we briefly describe standard optimization techniques. However, the real problem is not in the optimization, but in finding an energy function that represents a good model of the stereo problem. In this context, we investigate data and smoothness terms of standard energies to find the best-suited implementations of which. We then describe our own work on finding a good model. This includes our combined stereo and matting approach, Surface Stereo, Object Stereo as well as a new method that incorporates physics-based reasoning in stereo matching.

1 The Stereo Matching Problem

This chapter concentrates on the stereo matching problem, which is one of the oldest, but still yet unsolved problems in computer vision. Solving this matching problem is the central step in a shape from stereo method. Figure 1 outlines the pipeline of this approach. In analogy to human depth perception that uses two eyes, there are two cameras. These cameras are slightly displaced such that they see the same scene from different viewpoints.[1] Note that throughout this chapter we will assume that the stereo pair has been rectified such that corresponding points lie on the same horizontal scanlines in left and right views.[2]

Michael Bleyer
Vienna University of Technology e-mail: bleyer@ims.tuwien.ac.at

Christian Breiteneder
Vienna University of Technology e-mail: breiteneder@ims.tuwien.ac.at

[1] The distance between the cameras is thereby referred to as the stereo baseline.

[2] Rectification can be accomplished using standard methods (e.g., [29]), once the stereo camera system has been calibrated.

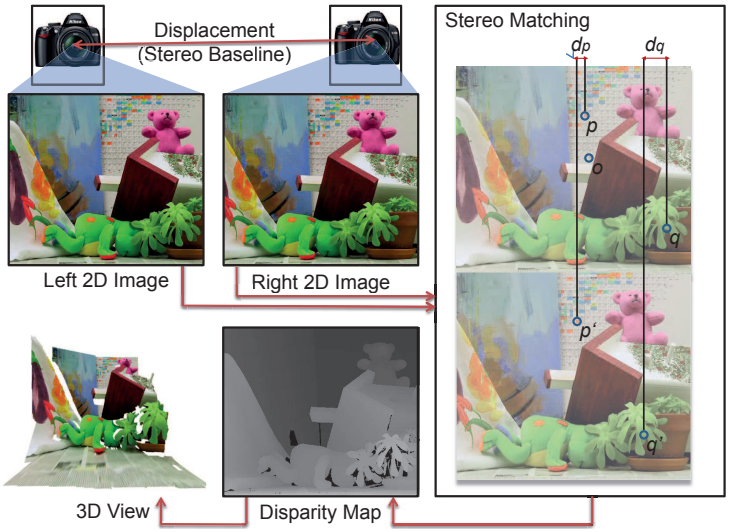


Fig. 1 Depth reconstruction via stereo. Two slightly displaced cameras record the scene. The stereo matching module finds dense correspondences between left and right images and encodes them in a disparity map. This disparity map is sufficient to reconstruct the 3D coordinates of each pixel.

Left and right images then form the input for the stereo matching module. In stereo matching, the task is to find a corresponding pixel in the right image for each pixel of the left image. Let us look at figure 1 where we have placed the left image on top of the right one to understand why this allows depth reasoning. We have marked the background pixel p of the left image as well as its corresponding pixel p' in the right view. Note that p and p' are displaced in horizontal direction due to the different perspectives under which left and right images have been recorded. The amount of this displacement (in pixels) is called disparity (d_p in the figure). We have also marked a pixel q and its correspondence q' of a foreground object. The important observation is that the disparity of the foreground pixel is larger than that of the background pixel (compare d_p and d_q in figure 1) and it is easy to show (e.g., in [29]) that disparity is inversely proportional to the distance of a pixel to the camera. Note that also the human brain perceives depth using disparity information.

In computational stereo, disparity information is typically stored in an intensity image (a so-called disparity map) where dark pixels encode low disparity values (high distances from the camera) and bright intensities encode large disparities (close distances to the camera). See figure 1 for an example. In a calibrated stereo system, the disparity map is sufficient to reconstruct a metric 3D model of the recorded scene, which is the final goal in a shape from stereo approach.

The challenge in shape from stereo is to solve the stereo correspondence problem. This problem is permanently and unconsciously solved in the human brain, but turns out to be very challenging for a computer. This is for various reasons. Firstly, as common in computer vision, images are usually contaminated by sensor noise. Secondly, in the absence of texture, stereo matching becomes highly ambiguous.

Note that also a human is not able to perceive the correct depth if, for example, standing in front of a completely white wall without any texture. Thirdly, not every pixel of one image has a correspondence in the other image, because due to the different image perspectives, a pixel can be occluded. To illustrate this, we have also marked a pixel o in figure 1. Note that this pixel cannot be found in the right view, i.e., it is occluded.

Being able to solve the stereo matching problem is important in two respects. Firstly, it may help to get a better understanding of how human depth perception works. Secondly, there are various applications in computer vision. For example, 3D reconstruction of cities from internet photography has recently gained popularity [1, 71]. Depth maps can also be fused to automatically generate high-quality 3D models of persons or even whole rooms as demonstrated by KinectFusion [56]. Stereo reconstructions can be applied for robot navigation (e.g., autonomously driving car), but also in human motion capture where Microsoft Kinect has recently demonstrated that depth information is vital [69]. There is also potential in next generation television where depth maps enable novel view synthesis, which allows the user to interactively control its viewing perspective [90]. Other applications include 3D tracking (surveillance, pose estimation, augmented reality, human-computer interaction), depth segmentation (z-keying) and industrial applications (quality assurance), to name just a few of them. Basically, whenever one needs to infer geometric information from the surrounding world, stereo vision represents a low-cost and non-intrusive alternative to active devices such as range finders.

In the following, we give a review of stereo methods with a focus on recent developments. This review provides the context for our own algorithms. We follow Scharstein and Szeliski [68] by dividing stereo algorithms in local and global ones.

2 Local Methods

A Naive Approach Let us start by describing the simplest possible stereo algorithm. It is reasonable to assume that corresponding pixels in left and right images have similar colors, which is known as the photo consistency assumption. Furthermore, we know that corresponding pixels lie on the same horizontal scanline as we assume that our images have been rectified. Hence a simple algorithm works as follows. For each pixel p of the left image, we search along the corresponding scanline in the right image. We then select the pixel p' whose color is most similar to that of p as a matching point.

Figure 2c shows the disparity map of this naive approach which is very noisy. The problem is the high ambiguity of the data, i.e., if we want to find the correspondence of a red pixel of the left image (e.g., on the Tsukuba lamp of figure 2a) there is usually a relatively large candidate set of red pixels in the right image. The common approach taken in computer vision (and in all stereo algorithms) is to regularize the problem by imposing a smoothness assumption in order to cope with this ambiguity. This smoothness assumption means that spatial neighboring pixels are likely to

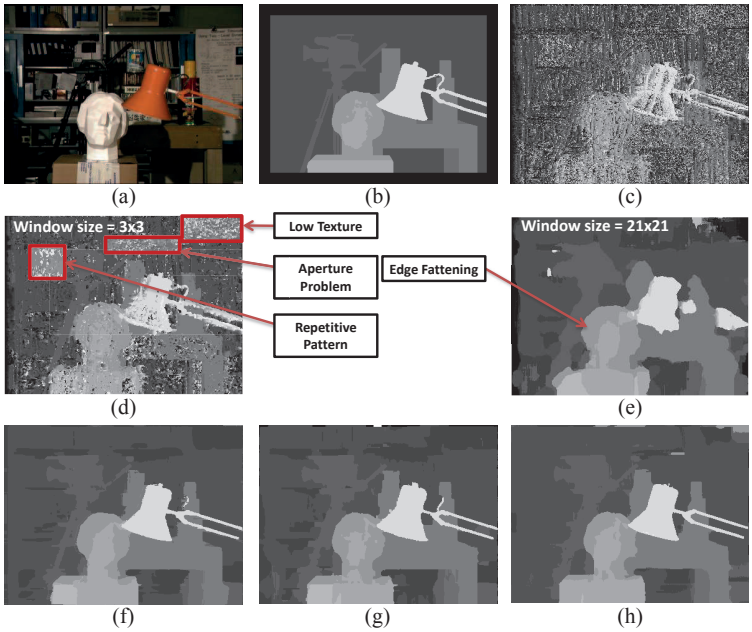


Fig. 2 Local Methods. (a) Left image of the Middlebury Tsukuba image. (b) Ground truth disparity map. (c) Result of a naive algorithm. (d) Window-based aggregation with a 3×3 window. (e) Aggregation with a 21×21 window. (f) Result of the adaptive support weight approach [86]. (g) Our geodesic approach [37]. (h) Our cost filter method [62].

have similar disparities. Stereo algorithms differ in the way how this assumption is implemented, which defines the difference between local and global methods [68].

The Principle of Local Algorithms Let us now come back to the naive approach above. Instead of matching single pixels, we can match small image areas.[3] It is important to note that by using areas we have made use of the smoothness assumption in an implicit way, i.e., we assume that all pixels of the area have exactly the same disparity. Let us now formulate the corresponding algorithm. For each pixel p of the left image, we compute its disparity d_p as

$$d_p = \underset{0 \leq d \leq d_{max}}{\operatorname{argmin}} \sum_{q \in W_p} c(q, q-d). \tag{1}$$

Here, d_{max} is a parameter defining the maximum allowed disparity. W_p denotes a square window centered on p. The function $c(p,q)$ computes the color dissimilarities between a pixel p of the left and a pixel q of the right image (e.g., summed-up absolute differences in RGB values). We write $q-d$ to denote the pixel coordinate that is derived by subtracting d from q's x-coordinate.

[3] This is the reason why local algorithms are also sometimes referred to as area-based methods in literature.

Selecting an Appropriate Window Size The local algorithm above has one important parameter, i.e., the size of the window W. From a computational point of view, the algorithm's runtime complexity is $\mathcal{O}(N \cdot d_{max} \cdot |W|)$ where N is the number of pixels in the image and $|W|$ denotes the number of pixels in the window. Hence, larger windows would lead to higher run times if the algorithm was implemented in a naive way. However, in an efficient implementation, one can take advantage of high redundancy in the computation to reduce runtime complexity to $\mathcal{O}(N \cdot d_{max})$ so that run time no longer depends on the size of the support window. The trick is to use a so-called sliding window technique [21, 55]. However, the size of the support window has a large effect on the quality of the disparity map as discussed next.

The problem of small support windows is that they may not capture enough texture variation to resolve matching ambiguities. In particular, the algorithm fails in untextured regions, areas with only horizontal texture (aperture problem) and repetitive image regions. This is illustrated in figure 2d where we have used a 3×3 window to compute the disparity map. However, note that disparity discontinuities are well preserved as a consequence of the small window size.

A remedy to overcome the ambiguity problem is the use of large support windows. A matching result using a large 21×21 window is shown in figure 2e. While this disparity map is considerably smoother than that of figure 2d, it is evident that object borders are badly preserved. The problem is our implicit smoothness assumption, i.e., pixels within the window are supposed to have constant disparity. In the proximity of depth discontinuities, this assumption is broken as the window captures a mixture of foreground and background disparities. Whether the foreground or the background disparity leads to lower color dissimilarity depends on the texturedness of objects. In many cases the foreground's texture is dominant, which leads to the well-known foreground fattening problem (see figure 2e).

The traditional problem of local algorithms is that there is no ideal setting for the parameter defining the window size such that the algorithm gives correct results in low textured areas and regions close to object borders at the same time. Almost all work on local stereo matching focuses on the use of adaptive windows.

Adaptive Windows The idea of adaptive window algorithms is to select an individual window at each pixel such that the support region of a pixel remains large (in order to capture enough intensity variation), but does not overlap a disparity discontinuity (in order to avoid the edge fattening problem). For example, Fusiello et al. [24] test nine different square windows of constant size per pixel. These windows differ from each other in that they are centered at different positions, and the hope is that at least on one of these positions the window does not overlap a depth discontinuity. Hirschmüller et al. [33] divide the search window into a set of nine sub-windows. Only five of these sub-windows are used to compute the aggregated matching costs, i.e., if the other four subwindows capture a different disparity than the center pixel they do not have any influence. As a final example, Veksler [79] estimates an arbitrarily sized and shaped window per pixel by optimizing over a large class of compact windows. In practice, none of the above methods has been able to compete with the quality of global methods (described in section 3). Hence the local stereo approach has been believed to be obsolete for some time.

Adaptive Support Weights In recent years, local stereo has experienced a renaissance due to the introduction of adaptive support weights [86].[4] The key idea is to assign an individual weight to each pixel that determines the pixel's influence in the matching process. Let us reformulate equation (1) accordingly as

$$d_p = \underset{0 \leq d \leq d_{max}}{\operatorname{argmin}} \sum_{q \in W_p} w(p,q) \cdot c(q, q-d). \tag{2}$$

We have now introduced a weight function $w(p,q)$ which should ideally return a value of 1 if pixel q lies on the same disparity as the center pixel p and 0 otherwise. Since disparities are not known in advance, defining $w(p,q)$ is challenging, i.e., leads to a chicken-and-egg problem. Adaptive support weight algorithms differ in the way how they define $w(p,q)$ and we discuss this below. To our knowledge, all of them use the color cue for computing this function.

The Weight Function The original adaptive support weight paper [86] makes the assumption that spatially close pixels that are similar in color are likely to originate from the same scene object. Hence they are also likely to share the same disparity. We define the corresponding function $w_{BL}(p,q)$ as

$$w_{BL}(p,q) = \exp\left(-\left(\frac{color(p,q)}{\gamma_c} + \frac{spatial(p,q)}{\gamma_s}\right)\right). \tag{3}$$

Here, $color(p,q)$ computes the color dissimilarity of p and q as the Euclidean distance between p's and q's color values in RGB space. The function $spatial(p,q)$ computes the Euclidean distance between p's and q's image coordinates. γ_c and γ_s are user-defined parameters. Note that a pixel q obtains high weight only if it is similar to the center pixel p in terms of color and spatial position (see assumption above). Also note that the function above is equivalent to the weighting function of the bilateral filter.

Yoon and Kweon's work [86] has been a breakthrough in local stereo matching as (at least in the Middlebury benchmark) results on-par with global methods could be achieved for the first time. To illustrate the good quality of disparity maps, we plot the Tsukuba result of [86] in figure 2f. The downside of this approach is that these good-quality results come at the price of considerably increased run times. Due to the weighting function the sliding window technique (mentioned above) can no longer work and the algorithm's run time depends on the size of the support window. This is particularly bad as adaptive support weight techniques typically operate on large windows (e.g., 33×33 pixels in [86]). We will discuss later on one of our techniques [62] that has a considerably better runtime property, i.e., runs in real time, and even outperforms [86] in terms of quality of results.

Our Contribution - Geodesic Stereo [37] Our paper [37] proposes a new weight function. The idea is to introduce a connectivity property, i.e., two pixels are likely to lie on the same object (and disparity, respectively) if they are connected by a path

[4] It is interesting to note that the majority of recent submissions to the Middlebury benchmark [68] are adaptive support weight techniques.

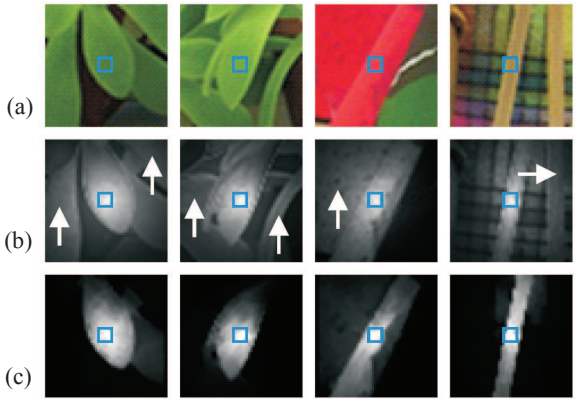


Fig. 3 Our geodesic approach [37]. We show adaptive support windows for the Middlebury Teddy and Cones images. (a) Color patch. The window's center pixel is marked by a blue rectangle. (b) Support weights computed by [86]. Bright values mean high weight. As marked by the white arrows, high weights are assigned to pixels that lie on a different disparity than the center pixel (e.g., the background leaves in the two left-most images). (c) Our geodesic support weights [37]. We avoid these wrong high weights by enforcing connectivity.

of approximately the same color. Let us look at figure 3 to explain the advantage of this connectivity property. Figure 3a shows four image crops taken from the left images of the Middlebury Teddy and Cones pairs. Figure 3b shows that the weights computed by the method of [86] are not ideal. By only looking at color and spatial differences, the method assigns high weights to pixels that lie on a different disparity than the center pixel, which happens, for example, at the background leaves in the two left-most images of figure 3b. In contrast, our method [37] (see figure 3c) avoids these wrong high weights. In the weight computation at each pixel q, we check if q is connected to the center pixel p by a path of constant color. This is not the case in the leaves example as the foreground and background leaves are separated by a color edge. More formally, our weighting function $w_{GEO}(p,q)$ is inversely proportional to the geodesic distance:

$$w_{GEO}(p,q) = \exp\left(-\frac{geo(p,q)}{\gamma}\right) \tag{4}$$

where the parameter γ controls the strength of the segmentation. $geo(p,q)$ is defined as

$$geo(p,q) = \min_{P \in \mathscr{P}_{p,q}} d(P). \tag{5}$$

Here $\mathscr{P}_{p,q}$ denotes all possible paths $< p_1, p_2, \cdots, p_n >$ that connect p and q within the support window, i.e., $p_1 = p$ and $p_n = q$. The costs of a path $d(P)$ are computed as

$$d(P) = \sum_{i=2}^{i=n} color(p_i, p_{i-1}). \tag{6}$$

In our experiments on the Middlebury data, we demonstrate that our geodesic approach outperforms the original adaptive support weight method [86]. At the time of publication (2009) our method has been the top-performer among all local methods in the Middlebury benchmark. The disparity map generated for the Tsukuba images is shown in figure 2g. In a follow-up work [36], we have shown how to speed up our geodesic approach such that near real time frame rates can be achieved without considerable loss of quality.

Adaptive Support Weight Stereo via Image Filtering As stated above, the weight function of the original adaptive support weight function [86] corresponds to the filter weights of a bilateral filter. It is known that the aggregation step of Yoon and Kweon's method [86] can be understood as filtering the cost volume with a joint bilateral filter (see our cost filter paper [62] and [63]). This insight forms the basis for speeding up the adaptive support weight approach. To be more precise, the cost volume is a three-dimensional array that is derived by computing the costs for matching each pixel (x,y) at each allowed disparity d. The joint bilateral filter is then applied on each individual xy-slice of this volume where filter weights are computed from the left color image. Finally, a disparity map is obtained by selecting the disparity of minimum costs in the filtered volume at each pixel.

Being able to implement the original adaptive support weight approach with runtime independent of the window size boils down to the question whether it is possible to implement joint bilateral filtering with runtime complexity independent of the filter kernel size. According to the current state of research, a so-called O(1) implementation only works for approximations of the joint bilateral filter. Several authors [42, 63, 87] have used such approximations to derive fast implementations of Yoon and Kweon's algorithm [86]. In [42] and [87] the joint bilateral filter is approximated by using integral histograms as described in [60], while [63] uses an approximation based on the bilateral grid of [59]. The problem of these approximative approaches is that they sacrifice quality of disparity maps for speed.

A different strategy to derive an O(1) implementation of adaptive support weight matching is to replace the joint bilateral filter with a different filter that shares the joint bilateral filter's edge-preserving property, but can innately be implemented with runtime independent of the filter kernel size. In this line of research, [53, 88] use a cross-shaped filter. Due to using a cross-shaped support region, the algorithm fails at fine structures that are neither horizontal nor vertical. A better alternative is discussed next.

Our Contribution - Cost Filter [62] In [62], we propose to use the very recent guided filter [31] for cost filtering. This filter is similar to the bilateral one in that it shares its edge-preserving property. However, the advantage is that an exact (non-approximative) implementation can be accomplished by running a series of box filters so that the filter's runtime does not depend on the filter kernel size. We show in [39] that a GPU-based implementation runs at 33.3 frames per second for 640×480 pixel images and 40 allowed disparity levels. This makes it the fastest available adaptive support weight algorithm. Apart from that, our method has also been the best-performing method among all local algorithms in the Middlebury table [68] at the time of publication in 2011. In particular, this also means that it outperforms

(a) (b) (c)

Fig. 4 Our cost filter approach [62] applied on problems outside of stereo. (a) Optical flow. We use the color coding of [2] for visualization of the flow field. Our approach accurately reconstructs flow borders and thin structures. (b) Large displacement optical flow. In contrast to many competing algorithms, our method handles large motions. (c) Interactive segmentation. User input is given by drawing the yellow rectangle (see left image). Our method then computes a binary segmentation using cost filtering (see right image).

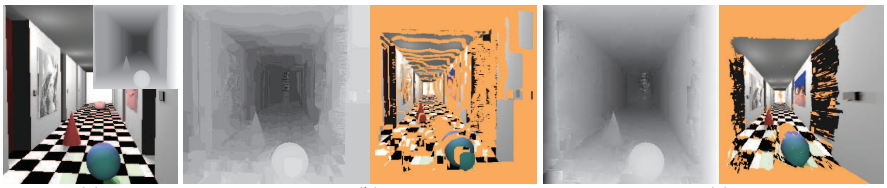
Yoon and Kweon's algorithm [86]. An example result is shown in figure 2h. In [38], we show how to extend the cost filter approach in order to compute temporally consistent disparity maps given a stereo video as an input.

An important contribution of our paper [62] is to show that the concept of cost filtering is not restricted to stereo, but can be applied to other computer vision tasks that can be formulated as labeling problems. We first instantiate our framework for optical flow computation and show that our algorithm can reach a top position in the Middlebury optical flow benchmark [2] using almost identical parameter settings as for the stereo problem. Figure 4a shows a flow map for the Middlebury Schefflera test set. As seen from figure 4b, an advantage of our optical flow method is that it can handle large displacements, which is challenging for most competing methods.

We then instantiate our framework for interactive image segmentation, which is a very different problem from stereo or optical flow computation. Note that in this case, the cost volume does not store color dissimilarities, but the likelihood to which a pixel belongs to a foreground and a background color model, respectively. The other steps, i.e., filtering of the cost volume and minimum selection, remain the same. We show that our segmentation method can compete with a state-of-the-art method, i.e., GrabCut [65], but runs considerably faster. It takes 5 milliseconds to segment a 1000×1000 pixel image. An example result is shown in figure 4c.

Slanted Surfaces and Sub-Pixel Precision Let us now come back to the implicit smoothness assumption of local algorithms, i.e., all pixels within the support window have the same disparity. We have already discussed that this assumption is broken at disparity borders and that adaptive support weight algorithms represent a good remedy to this problem. However, there are two additional problems. (1) Disparity values of pixels within the support window will be different if the window captures a slanted (non-fronto-parallel) surface. (2) So far we have only spoken about integer-valued disparities and ignored that we should ideally match at continuous sub-pixel disparity values.

Figure 5b shows the effect of these two problems on the Corridor test set that contains highly slanted surfaces. To derive the disparity map, we have used fronto-parallel windows that are matched at integer-valued disparities. Note that this is exactly what all algorithms described above do. As can be seen from figure 5b,



(a) (b) (c)

Fig. 5 Our PatchMatch Stereo algorithm [13]. (a) Left image and ground truth disparities of the Corridor pair that contains highly slanted surfaces. (b) The disparity map computed with fronto-parallel windows approximates the slanted surfaces via many fronto-parallel ones. (c) PatchMatch Stereo correctly reconstructs the scene as a collection of slanted planar surfaces via using slanted support windows and continuous sub-pixel disparities.

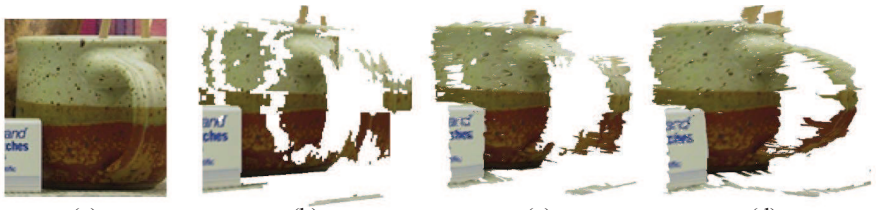
the 3D model derived from the computed disparity map is relatively poor, as the slanted planes are reconstructed via many fronto-parallel surfaces. It is known that this problem can be overcome by using slanted planar support windows matched at continuous sub-pixel disparities. However, this model leads to a difficult optimization problem as it is not known in advance which slanted windows occur in the scene and the number of candidate planes is infinite. After discussing previous methods, we will present our algorithm [13] that can effectively handle this complex optimization problem. The result of our algorithm on the Corridor scene is shown in figure 5c.

Previous Sub-Pixel / Slanted Window Algorithms Some local methods (e.g., [85]) obtain sub-pixel precision in a post-processing step by fitting a parabola in the cost volume. From our experience, this method leads to relatively noisy sub-pixel information. A better option is to account for sub-pixel displacements directly in the matching process. This can be accomplished by extending the label space, i.e., some fractional disparity values (half- or quarter-pixel) are considered in addition to the integer-valued ones (e.g., in [26]). Note that this is still a discrete approach and that run time doubles (quadruples) if half-pixel (quarter-pixel) precision is used. The same principle works for slanted windows, i.e., in addition to fronto-parallel windows a set of slanted windows can be included in the label space (e.g., in [25]). This is known as plane sweeping in the literature. As before, this is a discrete approach, i.e., the chances for not having the correct plane in the label space are high, and this strategy leads to considerably longer run times.

Finally, we also want to mention the approach of [89]. The authors first compute an initial disparity map and then apply plane fitting at each pixel. The extracted slanted planes are then used in a second matching round. The problem is that this method fails for highly slanted surfaces as the disparity results of the first round are too poor to extract the correct planes.

Our Contribution - PatchMatch Stereo [13] Let us reformulate equation (2) such that optimization is now performed over the set of all possible 3D planes, i.e.,

$$d_p = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \sum_{q \in W_p} w(p,q) \cdot c(q, q - (a_f q_x + b_f q_y + c_f)). \tag{7}$$



(a) (b) (c) (d)

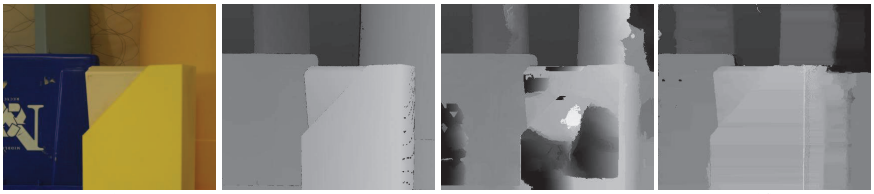
Fig. 6 3D reconstructions of our algorithm [13] and its competitors. (a) Crop of the Middlebury Cones image. (b) Matching using fronto-parallel windows and integer disparities. (c) Matching using fronto-parallel windows and continuous disparities. (c) Our PatchMatch stereo algorithm that uses slanted windows and continuous disparities. Note that the rounded shape of the cup's handle is well reconstructed.

Here $\mathscr{F}$ is the set of all disparity planes and a_f, b_f and c_f are the three parameters of a plane f. We write q_x and q_y to denote pixel q's x- and y-coordinates. Note that we also use an adaptive weight function $w(p,q)$ to handle the edge fattening problem. For simplicity, we use the one of [86] (see equation (3)). As stated above, minimizing equation (7) is difficult as the set $\mathscr{F}$ has infinite cardinality. Hence, the approach of checking all possible labels (implemented by all algorithms above) can no longer work. We propose an algorithm based on PatchMatch [4] to approximate the minimum of equation (7) at each pixel.

The basic observation is that relatively large regions of an image can be modeled by approximately the same plane. In the initialization step of the algorithm, each pixel is assigned to a plane with random parameters. It is relatively obvious that most of these random planes will be wrong, but the hope is that at least one pixel of a region carries a plane that is close to the optimal one. Note that this is very likely, since we have many guesses. For example, in the Corridor scene of figure 5a the ground plane consists of approximately 25000 pixels, i.e., we have 25000 guesses to find the correct plane for this object. If there is at least a single correct guess, then this is already sufficient, since our algorithm propagates this plane to neighboring pixels. We implement three different forms of propagation, i.e., spatial propagation, view propagation and temporal propagation (see our paper [13] for details).

Our PatchMatch stereo algorithm is currently the top performer among all local algorithms when sub-pixel precision is considered (Middlebury error threshold 0.5). It is even the top performer among all algorithms for the challenging Teddy set (Middlebury default error threshold 1.0) that has a highly slanted ground plane where competing methods run into problems. Figure 6 demonstrates the high amount of disparity detail that our method achieves. In contrast to competing methods (figures 6b and 6c), our method successfully reconstructs the handle of the cup in figure 6d.

Limitations of Local Methods Despite of the high research attention that local methods have gained over the last couple of years, they do not render global approaches useless. The problem is that even large support windows are not sufficient for handling highly ambiguous data, i.e., very untextured image regions. To illustrate this, we have used our PatchMatch stereo algorithm [13] that is considered to represent the state-of-the-art in local matching to compute the disparity map for the Middle-



(a) (b) (c) (d)

Fig. 7 Limitations of local methods. (a) Left image of the Middlebury Plastic pair. (b) Ground truth disparity map. (c) Results of our local algorithm PatchMatch stereo [13]. The yellow platic surface is poorly reconstructed due to the lack of texture. (d) Result of a global method. The global smoothness prior is essential to correctly reconstruct the yellow surface.

bury Plastic image. As can be seen from figure 7c, the algorithm fails to handle the completely untextured yellow surface, as this high ambiguity cannot be resolved completely locally. In such cases, a global smoothness prior is essential. To demonstrate this, we have embedded PatchMatch stereo as a data term in a variant of the second order stereo algorithm of [82] (see our paper [13] for details on how this is accomplished). As can be seen from figure 7d, the yellow plastic surface can now be handled successfully. Note that while local stereo approaches have started to take over the top positions in the Middlebury benchmark [68] from global methods, this is not the case on more realistic imaginary such as that of the very recent KITTI street scene benchmark [27] that is dominated by global methods.

The important point is that the aggregation schemes of local methods can supplement global methods in the form of a stronger (less ambiguous) data term. Due to the use of adaptive support weights, aggregation also improves the performance of global methods near depth discontinuities. Slanted support windows may also improve the results of global algorithms at highly slanted surfaces. Finally it is important to note that many robust match measures require a window to be computed (e.g., Census or Normalized Cross Correlation). As described later, such window-based measures are specifically important on realistic images with illumination differences, e.g., the left image is darker than the right one.

Finally, we would also like to mention another limitation of local algorithms, i.e., it is difficult to incorporate occlusion handling directly in the matching process. Typically occlusions are treated in a post-processing step by left-right consistency checking [23]. This means that the disparity map for the right image is computed in addition to that of the left view. The check then invalidates pixels of the left view whose disparity value is different in the right image. In the final step, these invalidated pixels are filled by replicating disparities of spatially close valid pixels (e.g., see our filling procedure in [62]). As we will see in the next section, global methods allow for occlusion handling directly in the matching process by modeling the occlusion problem in their energy functions.

Fig. 8 Disparity maps for increasing values of λ in equation (8). The left-most image shows the result that is optimal according to the data term ($\lambda := 0$), while the right-most image shows the disparity map optimal according to the smoothness term ($\lambda := \infty$).

3 Global Methods

The Principle of Global Algorithms Global methods differ from local ones in that they express the smoothness assumption in an explicit form via a so-called smoothness term. Global algorithms define an energy function $E(D)$ that measures the quality of a disparity map D. In the subsequent optimization step, the goal is to find the disparity map of lowest energy, i.e., of highest quality according to our energy function. The typical form of a stereo energy function is:

$$E(D) = E_{data}(D) + \lambda \cdot E_{smooth}(D). \tag{8}$$

Here, λ is a user-defined parameter that balances the influence of E_{data} and E_{smooth}. The data term E_{data} measures photo consistency and is defined as

$$E_{data}(D) = \sum_{p \in I_l} c(p, p - d_p) \tag{9}$$

where I_l denotes all pixels of the left image and $c(p,p')$ computes the pixel dissimilarity between pixel p of the left view and pixel p' of the right image (e.g., absolute difference of intensity values). d_p represents the disparity value of pixel p in disparity map D. Let us now focus on the smoothness term E_{smooth}. It is responsible for preferring disparity maps that are spatially smooth over such that are not by assigning lower energy. E_{smooth} is defined as

$$E_{smooth}(D) = \sum_{<p,q> \in \mathcal{N}} s(d_p, d_q) \tag{10}$$

where $\mathcal{N}$ denotes all pairs of spatially neighboring pixels in the left image. Note that since the term is defined on pairs of pixels, it is often referred to as pairwise term, whereas the data term is referred to as unary term. The function $s(d_p, d_q)$ assigns a penalty if p's disparity is different from that of q. As discussed below, there are different options for implementing the smoothness function $s()$, which also defines the complexity of the optimization problem. To show the influence of E_{data} and E_{smooth}, we plot disparity maps for the Tsukuba images using different settings of λ in equation (8) in figure 8. Note that if we set $\lambda := 0$, i.e., switch off the smoothness term, then our method "degenerates" to a purely local one.

Optimization Let us now focus on the problem of finding a disparity map that minimizes equation (8). This problem is difficult as the disparity assignment of each

pixel influences the disparity assignment of every other pixel of the image via the smoothness term. This is known as the "Knock-on" effect in literature. It is known that even for the simplest discontinuity-preserving smoothness function, i.e., the Potts model where $s(d_p, d_q) = 0$ if $d_p = d_q$ and $s(d_p, d_q) = 1$ otherwise, finding the global minimum of equation (8) is an np-complete problem (see [18] for a formal proof). However, there exist powerful optimization strategies that approximate the energy minimum. In the following, we will discuss three of them, i.e., (1) dynamic programming, (2) graph-cuts and (3) message passing. Note that there is also an online competition [75] that compares the effectiveness of optimization strategies.

Dynamic Programming As stated above, exact minimization of equation (8) is an np-complete problem in general. However, there exists a special case. If the smoothness interactions ($\mathcal{N}$ in equation (10)) form a tree in the image grid, the global optimal solution can efficiently be computed via dynamic programming (DP). The simplest method to modify $\mathcal{N}$ such that this set does not contain cycles is to remove all vertical smoothness interactions and this is exactly what most DP-based algorithms do (e.g., [6, 17, 58] or the scanline optimization method of [68]). In this case, each horizontal scanline is optimized independently from the others and this leads to the well-known scanline streaking problem (see figure 9a). This streaking problem is the reason why such methods clearly fall behind the state-of-the-art in stereo matching [68].

Veksler [80] proposes a smarter way to transform $\mathcal{N}$ into a tree that contains horizontal as well as vertical smoothness edges. The idea is that some smoothness edges are more important than others. In particular, if two neighboring pixels are similar in color, they are likely to lie on the same disparity. Hence the corresponding smoothness edge is more important than one connecting two pixels of different color. This prioritization is then transformed into weights and the tree is built using a minimum spanning tree algorithm. While this approach reduces horizontal streaks, it, however, introduces vertical streaks (see figure 9b).

One of the most popular stereo algorithms (e.g., implemented in the Open Computer Vision library) is the semi-global matching approach [32]. Note that we will describe it from the viewpoint of tree DP, which is very different from the description given in [32].[5] Each pixel p represents the root of a tree. This tree has a star-shaped form, i.e., captures all pixels that lie on p's horizontal, vertical and diagonal scanlines. The root p is then assigned to the disparity of optimal costs on this tree structure. As can be seen from figure 9c, the approach effectively overcomes the scanline streaking problem, but produces isolated pixels. The problem is that the star-shaped tree is composed only of a subset of all image pixels. This subset may not capture enough texture to derive the correct disparity at the tree's root.

Our Contribution - Simple Trees [11] Our Simple Tree approach [11] also constructs individual trees per pixel. However, in contrast to [32], our trees contain all pixels of the reference view and hence the problem of missing texture (see above) is avoided. As shown in figure 9d, we use two different trees at each pixel p. The

[5] Hirschmüller stresses a relationship to local methods. He calls the method semi-global matching, as he uses the aggregation step of local methods, but aggregates (global) DP path costs instead of pixel dissimilarities of spatially close pixels.

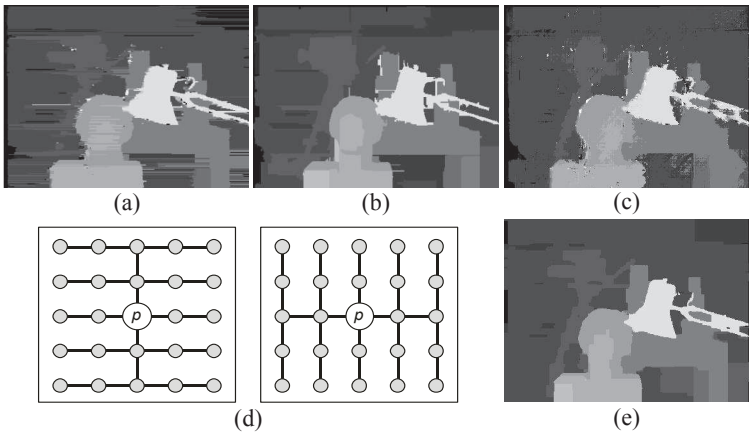


Fig. 9 Dynamic programming-based methods. (a) The scanline optimization method of [68]. The disparity map suffers from horizontal streaks. (b) The tree-based method of [80]. The streaking problem is reduced, but in addition to horizontal streaks, vertical streaks are introduced. (c) Our reimplementation of [32]. The streaking problem is eliminated, but the algorithm produces isolated pixels. (d) The tree structures used in our work [11]. (e) Corresponding disparity map. Our disparity result does not suffer from streaks or isolated pixels.

horizontal tree (figure 9d left) captures all horizontal smoothness edges as well as the vertical smoothness links of p's scanline. The vertical tree (figure 9d right) represents the complementary tree, i.e., captures all vertical smoothness links and the horizontal edges of p's scanline. Note that there is high redundancy in computation of all tree's optima. This can be accomplished with a few DP scanline passes, which takes less than a second on standard images and in a CPU-based implementation.[6] Hence the algorithm shows an excellent speed versus quality trade-off. We show our Tsukuba result in figure 9e. As can be seen, this result is free of scanline streaks and isolated pixels.

Graph-Cuts Graph-cuts have been used in stereo vision for some years (e.g., [67]). In the following, we will speak about "modern" graph-cut based algorithms, so-called move making algorithms. These algorithms are capable of effectively minimizing equation (8) and have been the winner in the optimization benchmark of [75] on some problem instances. The idea is to start with an arbitrary disparity map[7] and then to iteratively apply moves. Here, a move leads to a new disparity map of lower (or at least the same) energy. There are two well-known moves, i.e., $\alpha\beta$-swaps and α-expansions [18]. We will focus on α-expansions, because they are known to outperform $\alpha\beta$-swaps. To apply an α-expansion, a discrete disparity label d_α is selected. The α-expansion move changes the disparity map such that each pixel either keeps its current disparity or changes it to d_α. The challenge is to compute the α-expansion that leads to the largest decrease of energy (8) among all

[6] It is likely that the approach would run in real time if implemented on a modern GPU.

[7] Note that although we speak about disparity, these move making algorithms can be applied to arbitrary computer vision labeling problems outside stereo vision.

possible α-expansions. This problem can be solved exactly by computing the minimum cut / maximum flow in a special purpose graph.[8] Note that this is only true if the energy is sub-modular [18], which is e.g., the case for the Potts model. There are many stereo algorithms that apply α-exapansions as their optimization engine (e.g., [35, 46, 47, 52] or our algorithm [10]). Note that graph-cuts can optimize more complex energies than that of equation (8) such as that discussed below.

Lempitsky et al. [50] have recently proposed a new move, i.e., the fusion move that is interesting because it enables graph-cut-based optimization over continuous disparity values. The idea is to have have two proposal disparity maps. In a fusion move, each pixel either takes the disparity of proposal 1 or that of proposal 2. Note that if proposal 2 only has a single disparity for all pixels, then this fusion move is identical to an α-expansion. Hence fusion moves are a generalization of α-expansions. The problem is that in the general case the optimization graph contains non-submodular edges and hence computing the optimal fusion move becomes an np-complete problem. However, there are graph-cut based algorithms, i.e., quadratic pseudo-boolean optimization (QPBO) [45] that can handle non-submodular energies.[9] The fusion move framework has been used in the second order stereo algorithm [82] and our algorithms [15, 16].

Message Passing The most prominent algorithm based on message passing is Belief Propagation (BP) [22]. BP is an iterative procedure where each pixel communicates with its four spatial neighbors via sending messages. A message is thereby a vector that has an entry for each allowed disparity. A message $m^t_{pq}(d_q)$ sent from pixel p to pixel q at iteration t encodes p's belief that q should be assigned to a particular disparity d_q and is computed as

$$m^t_{pq}(d_q) = \min_{0 \leq d_p \leq d_{max}} \left(s(d_p, d_q) + c(p, p - d_p) + \sum_{r \in \mathcal{N}(p) \setminus \{q\}} m^{t-1}_{rp}(d_p) \right) \tag{11}$$

where d_{max} is the maximum allowed disparity and $\mathcal{N}(p)\setminus\{q\}$ denotes p's spatial neighbors excluding q. After T iterations the final disparity d^*_q at pixel q is computed as

$$d^*_q = \operatorname*{argmin}_{0 \leq d_q \leq d_{max}} \left(c(q, q - d_q) + \sum_{p \in \mathcal{N}(q)} m^T_{pq}(d_q) \right). \tag{12}$$

Note that by using the strategy of [22] the computational complexity of calculating the messages (equation (11)) for all image pixels can be reduced to $\mathcal{O}(N \cdot d_{max})$ where N is the number of pixels. This makes BP fast and GPU-based real time implementations for stereo are available [61]. BP forms the optimization engine for many stereo algorithms (e.g., [43, 72, 73, 76]).

---

[8] This is why these algorithms are referred to as graph-cuts.

[9] Due to the np-hardness of the problem, QPBO can only guarantee to find a part of the global optimal solution and will, in general, leave a subset of pixels unlabeled. The autarky property of QPBO guarantees that by assigning unlabeled pixels to the disparities of proposal 1 the energy of the fusion result will be lower or the same as that of proposal 1. Alternative ways for handling these unlabeled pixels are QPBO-I and QPBO-P [66].

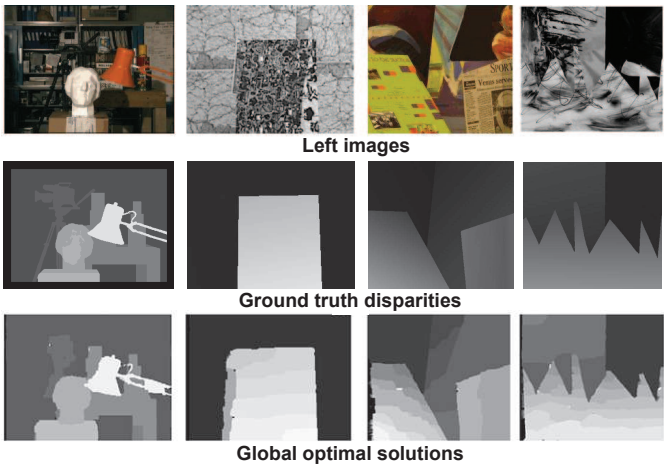


Fig. 10 Global optimal solutions for energy (8) on the old Middlebury set. (The Potts model is used to implement the smoothness function.) Images are taken from [54]. Note that even the global energy minimum leads to disparity maps that are relatively far off from the ground truth solution (e.g., see handle of the lamp or the camera in the Tsukuba image). This suggests that the energy formulation does not represent an ideal model for the stereo problem.

Finally, we also mention tree-reweighted message passing (TRW) [81], which is similar to BP, but uses a different message update rule. TRW allows to compute a lower bound on the energy. Comparison of the current energy against this lower bound gives an estimate of how close the current solution is to a global optimal one [54, 75], i.e., if the energy is equal to the lower bound, a global energy optimum has been found.

The Problem is the Model In general, when an energy minimization approach delivers suboptimal results, there are two reasons: (1) The energy function does not represent a good model of the stereo problem or (2) the optimization technique fails to effectively minimize the energy. The problem is that one often does not know whether bad performance is for the first or the second reason (or a combination thereof), since the global energy optimum is usually unknown.

Recent studies give indication that with the use of modern optimization schemes (graph-cuts or TRW) the formulation of the energy function has become the limiting factor. For example, in the above mentioned optimization algorithm comparison [75], the authors have shown that the best-performing optimization algorithms applied on the energy of equation (8) are capable of computing local energy minima extremely close to the exact solution. However, it has been pointed out that, due to flaws in the energy formulation, this does not necessarily translate into improved reconstruction performance (with respect to ground truth data). A specifically interesting result of this study (also noted in [78]) is that the energy of the ground truth image is considerably higher than that produced by modern optimization algorithms. This clearly shows that the energy function represents an unsatisfactory model for the stereo problem.

This is also consistent with the findings of Meltzer et al. [54], who have shown that, despite the np-hardness of the optimization problem, an exact optimum can be obtained for some standard benchmark stereo pairs (the old Middlebury set [68]) using tree-reweighted message passing. The corresponding results are shown in figure 10. Even the global optimal solution has not led to a better disparity reconstruction in comparison to standard energy minimization approaches. Consequently, the authors have concluded that *"the problem is not in the optimization algorithm, but rather in the energy function"*. In other words, progress in stereo can only be achieved by improving the energy function. In the remainder of this chapter, we will discuss options to improve the stereo model.

Data Term Let us start by investigating the data term (equation (9) of our energy (8)). Note that from our experience, the match measure has a very large influence on the quality of disparity maps (also see our papers [7, 8]), which is oftentimes larger than the influences of smoothness terms and optimization algorithms.

There are various options to implement the pixel dissimilarity function $c()$. The simplest strategy is to compute the absolute difference (AD) of intensity values, i.e., $c(p,q) = |I_p - I_q|$ where I_p denotes the intensity at pixel p. An alternative approach is to calculate the squared difference (SD), i.e., $c(p,q) = (I_p - I_q)^2$. According to the experiments of [68], there is relatively little performance difference between AD and SD, although SD is potentially more sensitive to outliers. It makes sense to truncate AD and SD values to reduce the influence of outlier pixels. While this improves matching performance [68], the selection of the threshold value that should be chosen according to the images' noise level represents a problem. Birchfield and Tomasi [5] present a match measure that reduces the influence of image sampling. The idea is to perform linear interpolation to derive the intensity at sub-pixel locations and to check whether these sub-pixel intensities lead to lower matching costs. From our experience, this match measure does not necessarily improve quality.

An important weakness of all above dissimilarity functions is that they cannot handle radiometric distortions (e.g., the left image is darker than the right one). Note that in practical stereo such radiometric distortions are almost always present and the treatment of this problem is vital. There are various competing radiometric insensitive measures and an evaluation of which has been conducted in [34].

The first option is to transform left and right images via a filter in a pre-processing step. Matching is then performed on the filtered images using a standard measure such as AD or SD. An edge filter (e.g., Sobel) is an obvious choice. Note that the gradient is not affected if radiometric differences are due to a constant intensity offset, e.g., the intensity of corresponding points is always 10 levels higher in the right image.[10] An alternative filter to handle intensity offsets is to subtract the mean computed in a window from the original intensity images. Note that since the mean is computed within a window, this leads to edge fattening. Hence a better option is to subtract the image processed by an edge-preserving filter (e.g., bilateral or guided filter) from the original image. According to the experiments of [34], the filter-based methods fall behind the match measures discussed next in terms of quality.

[10] In practice, it is sufficient to compute the gradient in x-direction, as vertical edges contain more disparity information than horizontal ones.

Mutual Information (MI) as implemented in [32] builds a single model of the intensity change for the whole image, i.e., given intensities I_p and $I_{p'}$ of pixels in left and right images, the MI score (looked up from the global model) gives the likelihood that I_p and $I_{p'}$ match each other. To compute this global model a disparity map is required, which leads to a chicken-and-egg problem. This dilemma can be solved in an iterative fashion. Given an initial disparity map, the MI scores are computed. These MI scores are then used to compute a new disparity map and so on.[11] The advantage is that MI is a pixel-based measure, i.e., there is no window required. Hence edge fattening and problems at slanted surfaces are avoided. This potential advantage should, however, be regarded in the light of new aggregation schemes that use adaptive support weights and slanted windows as discussed in section 2. The main disadvantage is that radiometric distortions are usually not the same all over the image (e.g., only the left bottom image part is darker in the left view). Such local changes cannot be handled by the global distortion model, which may represent the reason why MI has been outperformed by the window-based measures described next in the study of [34].

Zero mean Normalized Cross-Correlation (ZNCC) is a window-based measure than can handle intensity offsets and gains. It is defined as

$$c(p,p-d) = \frac{\sum_{q\in W_p}(I_q - \overline{I_p})(I_{q-d} - \overline{I_{p-d}})}{\sum_{q\in W_p}(I_q - \overline{I_p})^2 \sum_{q\in W_p}(I_{q-d} - \overline{I_{p-d}})^2} \tag{13}$$

where W_p represents a squared window centered at p and $\overline{I_p}$ denotes the mean intensity computed over all pixels inside W_p. A second popular window-based measure is Census. Instead of directly matching intensities (that may be affected by radiometric distortion), Census generates a bit string representation that describes the texture within the window. Each pixel q of the window is thereby compared against the center pixel p. If $I_q < I_p$ then 0 is concatenated to the bit string and 1 otherwise. The Census costs between the reference window and the window in the match frame are then computed as the Hamming distance between the corresponding two bit strings.

Our Contribution - The Role of Color [7] So far we have only spoken about matching intensities and ignored the fact that color information is typically available. Note that although color obviously represents additional information, many researcher still convert the input images to grey-scale when computing the match measure. We have published two evaluation papers [7, 8] that investigate the usefulness of color.

In the first paper [8], we have concentrated on the standard match measures AD and SD (see above) that we compute in eight different color spaces[12] as well as on the intensity images. We apply the Simple Tree method [13] (see above) to optimize energy (8) using the resulting match measures as data terms. We evaluate on a relatively large test set of 30 Middlebury ground truth pairs. We report a considerable

[11] Hirschmüller [32] uses a hierarchical approach to speed up this iterative procedure.

[12] In our study, the extension of AD and SD to color works by computing AD and SD for each color channel separately. We then sum up the differences over the 3 color channels.

quality improvement when using color. The best-performing color space gives 25% less disparity errors in comparison to only using intensity.

In our follow-up paper [7], we take a closer look at the image regions where color improves matching performance over intensity-based matching when using AD or SD. The key insight is that these regions correspond to image areas that are affected by radiometric distortions. The paper shows that the benefit of color is small, given that considerably better performance at radiometric distorted regions is achieved by directly using a radiometric insensitive measure such as ZNCC and Census instead of color-based AD. An interesting observation is that in comparison to the intensity-based versions of ZNCC and Census, performance even decreases if these measures are "enriched" with color information. Therefore, we suggest not to use color at all, but radiometric insensitive measures computed on intensity images.

Occlusion-Aware Data Term The data term defined in equation (9) has a systematical problem in that it does not consider the occlusion problem. It does not make sense to compute a match measure for occluded pixels, as the matching point in the other image does not exist. Note that ignoring the occlusion problem leads to wrong disparity results near disparity discontinuities, i.e., in occluded regions and their proximity. For example, in the second column of figure 10, the disparity to the left side of the foreground object is completely wrong due to this problem.

Let us now redefine equation (9) such that it accounts for occlusions:

$$E_{data}(D) = \sum_{p \in I_l} c(p, p - d_p)(1 - O(p)) + \lambda_{occ} O(p). \tag{14}$$

Here $O(p)$ is a function that returns 1 if pixel p is occluded and 0 otherwise. λ_{occ} is a user-defined penalty for occlusion. This parameter is required to prevent the algorithm from maximizing the number of occluded pixels. Note that according to equation (14) the match measure is only evaluated for visible pixels, while we impose the occlusion penalty for occluded ones. The challenge is to implement the occlusion function $O(p)$ and we follow the definition of [82]:

$$O(p) = \begin{cases} 1 & \text{if } \exists q \in I_l : p - d_p = q - d_q \text{ and } d_p < d_q \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

Let us use figure 11 to explain this equation. We have two pixels p and q. We can use their disparity values d_p and d_q to warp them into the geometry of the right view, i.e., we compute $p' = p - d_p$ and $q' = q - d_q$. In our example, p' and q' both lie on the same x-coordinate in the right view and given that the pixels lie on opaque (non-transparent) objects, only one of them can be seen by the right camera. From figure 11 it is clear that the visible pixel is q' as it has higher disparity, i.e., lies closer to the camera. This is exactly what is written in equation (15), i.e., a pixel p is occluded if there exists a pixel q such that both pixels project to the same pixel in the right image and p's disparity is smaller than that of q. Note that this data term can be optimized via graph-cuts, which has been done in [82].

Other approaches that handle occlusions in their energy functions are based on a symmetrical formulation, i.e., disparity maps are computed for both images and

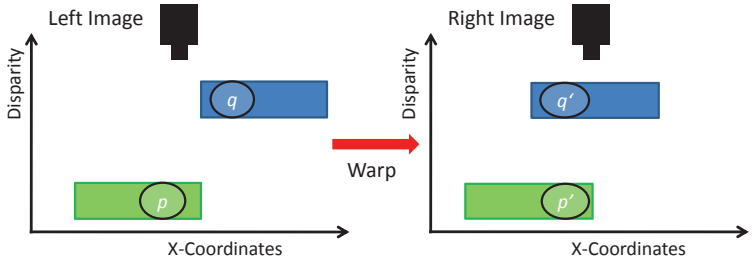


Fig. 11 Occlusion Reasoning. The pixel p is occluded by pixel q, because both of them project to the same pixel in the right image and q has higher disparity than p.

not just for the left one as we have done above. This makes sense as there is no plausible reason for treating the left image different from the right one.[13] Most authors (e.g., [46, 47, 52] and our paper [10]) implement the uniqueness assumption. This assumption enforces that each pixel has at most one matching point. Note that this assumption is broken for highly slanted surfaces where due to surface slant one pixel may have multiple (visible) correspondences in the other view [57]. Sun et al. [72] proposes a so-called visibility constraint to handle this problem and we show that the problem can be overcome using a surface-based representation in [15].

Smoothness Term Let us now concentrate on the smoothness term. As stated above, there are several possibilities to implement the smoothness function $s()$ of equation (10). We will first focus on first order functions that penalize the gradient of the disparity map. Let us start with the linear smoothness function, i.e., $s(d_p, d_q) = |d_p - d_q|$. Note that from the viewpoint of optimization this linear function represents an interesting special case as exact optimization of equation (8) does no longer represent an np-complete problem, but can be accomplished by computing a single cut in a graph [67].[14] The problem of a linear function is that it is not ideal for the stereo problem. Let us assume that we want to reconstruct a disparity border where disparity abruptly changes from 5 to 15. The problem of the linear term is that reconstructing this disparity discontinuity via several small jumps (e.g., from disparity 5 to 10 and then a few pixels later from 10 to 15) is as expensive as the large jump (from 5 to 15) that we prefer. Hence, disparity borders are blurred.

As stated above, the simplest discontinuity preserving function is the Potts model, i.e., $s(d_p, d_q) = 0$ if $d_p = d_q$ and $s(d_p, d_q) = 1$ otherwise. While this term works well for reconstructing depth discontinuities, it is suboptimal for reconstructing slanted surfaces where there is a gradual change of disparity. Note that each of these small disparity transitions receives a large penalty, i.e., the same penalty that we assign to depth discontinuities. The consequence is that slanted surfaces are reconstructed via various fronto-parallel ones.

A compromise between accuracy at disparity borders and accuracy at slanted surfaces is the truncated linear function, i.e., $s(d_p, d_q) = min(|d_p - d_q|, k)$ where k is a user-defined truncation value. Let us assume that we set $k := 3$ and again consider

[13] The stereo problem is symmetrical.

[14] In general, such a construction works if the smoothness term is convex [40].

the example of reconstructing a depth discontinuity where disparity changes from 5 to 15. The costs of using two jumps (from disparity 5 to 10 and from 10 to 15) is 6, while that of the single large jump (from 5 to 15) is 3, i.e, the single large jump is cheaper. This means that the model is discontinuity preserving. Let us now consider a small disparity transition (e.g., from 5 to 6) that occurs at a slanted surface. This transition only obtains a small penalty (1 in our example). The truncated linear term works well in practice and is implemented in many global stereo approaches (e.g., [32, 72, 84] to cite a few).

Note that none of the above first order smoothness functions is ideal for handling slanted surfaces. All of them assign higher energy to slanted surfaces than to fronto-parallel ones. This fronto-parallel bias cannot be justified as slanted surfaces are as likely to occur in natural images as fronto-parallel ones. The solution is to put a penalty on the curvature of the disparity map [51, 82], which is a so-called second order smoothness prior. This term is defined as $s(d_o, d_p, d_q) = |d_o - 2d_p + d_q|$ and penalizes the deviation of the three disparity values from a line (regardless of the line's slant). Note that minimization of energy (8) under second order smoothness becomes complex as the smoothness function has three arguments, i.e., leads to higher-order cliques in the optimization graph. However, for the triple cliques of the second order term, a scheme for transformation into pairwise cliques is known [48] and has provided the basis for the graph-cut-based optimization algorithm used in the second order stereo method of [82].[15]

Edge-Sensitive and Highly-Connected Smoothness Terms In section 2 on local stereo, we have stressed that the color image provides valuable information about the location of depth discontinuities. In particular, in most natural images, a disparity discontinuity is more likely to occur between neighboring pixels of different color than between homogeneously colored neighbors. Let us reformulate the smoothness term of equation (10) such that it incorporates this observation:

$$E_{smooth}(D) = \sum_{<p,q>\in\mathcal{N}} w(p,q) \cdot s(d_p, d_q). \tag{16}$$

This is known as an edge-sensitive smoothness term. The difference to the standard term is the weighting function $w()$. This function can, for example, be implemented using the color term of the bilateral weight function in equation (3), i.e., $w(p,q) = \exp(-color(p,q)/\gamma_c)$. Note that a high weight is assigned if p and q are similar in color and a low weight otherwise. Hence the energy for aligning depth discontinuities with color edges is lower than for placing depth discontinuities inside a homogeneously colored region.

The standard smoothness term has a bias towards reconstructing compact objects, because it assigns low energy if the length of a disparity border is small. Hence it is not well-suited for reconstructing complex outlines and thin structures. This is known as the edge shrinking bias in literature. The edge-sensitive smoothness term attenuates this shrinking bias to some extent, but does not work sufficiently well in practice.

[15] Note that in [82], the smoothness term is also truncated to make it discontinuity preserving.

A promising approach to overcome the shrinking bias[16] is to increase the neighborhood, which leads to a so-called highly-connected smoothness term defined as

$$E_{smooth}(D) = \sum_{p \in I_l} \sum_{q \in W_p} w(p,q) \cdot s(d_p, d_q) \tag{17}$$

where I_l denotes all pixel coordinates of the left image. Note that the smoothness function $s()$ is no longer only evaluated between a pixel p's four spatial neighbors, but within a window W_p centered on p. An obvious choice for $w()$ is the bilateral function of equation (3), i.e., the strength of the smoothness connection decreases proportional to the distance and color dissimilarity of connected pixels. Note that this approach can be regarded as the "global version" of the (local) adaptive weight approach [86] and Smith et al. [70] have shown that it achieves impressive results on the stereo problem. However, due to the large number of smoothness edges, optimization (e.g., via graph-cuts) is computationally very expensive. To reduce the number of edges, Smith et al. [70] eliminate "less important" smoothness connections using a strategy similar to Veksler's tree DP [80] (see paragraph on DP above), but their approach is still slow.

Ideally, the window W_p in equation (17) should capture all pixels of the whole image. This leads to a so-called fully-connected smoothness term where each pixel is directly connected to every other pixel via a smoothness link. Optimization of the resulting energy is intractable using standard methods (e.g., graph-cuts or standard BP) even on small images. Krähenbühl and Koltun [49] have recently presented an efficient optimization algorithm for this problem based on message passing. The "trick" is to compute message updates via a fast edge-preserving filter operation.[17]

Segmentation-Based Methods Segmentation-based algorithms (e.g., [20, 35, 77, 90] and our papers [9, 10]) also use the color cue. They apply a color segmentation of the left input image. The assumption of segmentation-based algorithms is that disparity varies smoothly within a region of homogeneous color (i.e., within a segment), while depth discontinuities coincide with segment borders. Hence instead of matching individual pixels, these methods match whole segments at once. Disparity within a segment is thereby assumed to be constant [90] or (in a more sophisticated approach) to lie on a slanted 3D plane [9, 10, 77].

Segmentation-based methods have been very popular for some time, most likely due to their excellent performance on the (new) Middlebury benchmark [68]. However, generalization to other images represents an issue. To illustrate this, we plot results on the Map test set in figure 12 that has traditionally represented a pitfall for segmentation-based methods in the old Middlebury benchmark. When segmenting the image of figure 12a there occur segments that overlap a depth discontinuity. Segmentation-based methods cannot recover from such situations, because they model whole segments via a single plane (see errors of our method [9] along the depth discontinuity in figure 12c). The second problem is the planar model that oversimplifies the 3D shape in the presence of rounded objects. To alleviate both

[16] There are also alternative methods (e.g., cooperative cuts [41]).

[17] It is interesting to note some similarity to cost filter approaches discussed in section 2.

Fig. 12 Limitations of segmentation-based methods. (a) The intensity image of the Middlebury Map pair. (b) Ground truth disparity map. (c) Due to segments that overlap disparity discontinuities, our segmentation-based method [9] produces inaccurate results at depth discontinuities (see red arrows). (d) Our method [15] that uses segmentation only as a soft constraint can successfully handle this image pair.

issues, segmentation-based methods apply a strong oversegmentation, but as seen from figure 12c, this cannot completely overcome these problems. At a later point, we will discuss our Surface Stereo method [15] that can effectively handle both problems. We show the result of this method in figure 12d.

Stereo and the Matting Problem Before presenting our first algorithm that aims at providing a better model for the stereo problem, let us discuss the following problem. Natural images usually contain pixels that receive contributions from more than one real-world object as a consequence of lense blur. Such mixed pixels occur along depth boundaries where a point's color is the composite of colors reflected by background and foreground objects. This so-called matting problem is well-known in photo-montaging. When cutting out an object and pasting it against a new background without considering mixed pixels, the results at the object's border look unnatural as colors are "contaminated" by the color of the old background.

Let us look at figure 13a to understand the relevance of the matting problem in stereo matching. In this example, the goal is to reconstruct the depth of point p. Due to lense blur, the left camera sees a pixel color that is a mixture between the red foreground and the blue background, whereas in the right camera the red color of p is mixed with the green background. Hence p has different colors in left and right images, i.e., the photo consistency assumption is broken. This makes stereo matching difficult.

Although early work [3, 74] on stereo matching in combination with matting dates back 15 years, there has been relative little progress since then. Some authors [30, 90] compute alpha matting information in a postprocessing step, i.e., after computing a disparity map. Xiong and Jia [83] propose a method that can handle the color inconsistency problem of figure 13a using a joint stereo and matting formulation. However, their approach only works for images that consist of exactly two layers (i.e., foreground and background layers) and generalization to other images is an issue. Taguchi et al. [76] propose a method that can handle an arbitrary number of layers. However, they do not use matting information to overcome the color inconsistency problem.

Our Contribution - Combined Stereo and Matting [12] We present an algorithm for solving the stereo matching and alpha matting problems simultaneously. In contrast to the papers above, our method operates on an arbitrary number of layers

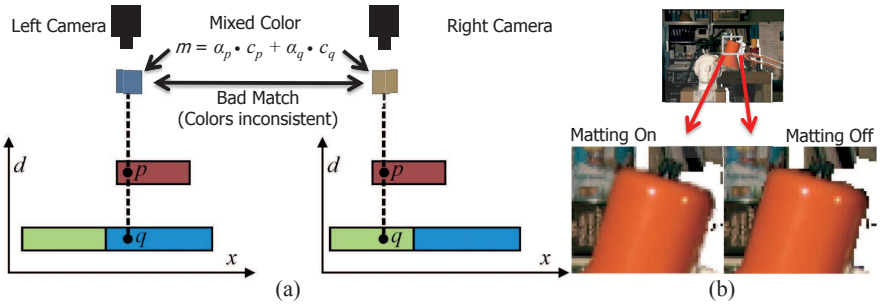


Fig. 13 The matting problem in stereo matching and results of our approach [12]. (a) Due to lense blur, the color of pixels at object borders is a mixture between the colors of foreground and background objects. The problem is that this mixture is different in left and right images, which leads to color inconsistencies. (b) We use our stereo and matting results for novel view synthesis. Due to using our matting information, the lamp naturally melts with its new background in the left image. The right image is generated without matting information. Notice disturbing artefacts at the lamp's border.

and still handles the color inconsistency problem. Our algorithm is based on image segmentation (see discussion of such methods above). Hence, we first apply color segmentation to the left image. Each segment is then enlarged using morphological dilation. The idea is that since mixed pixels typically occur at disparity (segment) borders (see discussion above), we only allow transparencies in a small band around segment borders. In accordance to alpha matting literature, we call this band a segment's unknown region. Note that due to this enlargement, we have many overlapping segments.

We now estimate a disparity plane for each overlapping segment and an alpha value as well as the true color for each pixel via energy minimization. Given alpha values and colors, our energy function computes the mixed colors for the left view. If alpha values and colors were correct, the resulting artificial image should be very similar to the real left view. Our energy function also computes an artificial right view via warping the pixels into the geometry of the right image according to our current disparity solution. This artificial right view is then compared against the real right image to measure the quality of alpha values, colors and disparity planes. The crucial question in generating the artificial right view is how alpha-values are affected by image warping. To correctly perform the warping operation, we introduce the concept of solidity (see our paper).

In the experimental results, we show that our method can compete with the state-of-the-art on the Middlebury benchmark [68]. More importantly, in contrast to all methods listed on Middlebury, our method also provides matting information. This makes it an ideal algorithm for performing depth segmentation and novel view generation. Figure 13 shows a novel view example, i.e., an image that shows the scene from a virtual viewpoint not recorded by the stereo cameras. As can be seen, our matting information leads to realistic results at object borders when a foreground object is pasted against a new background.

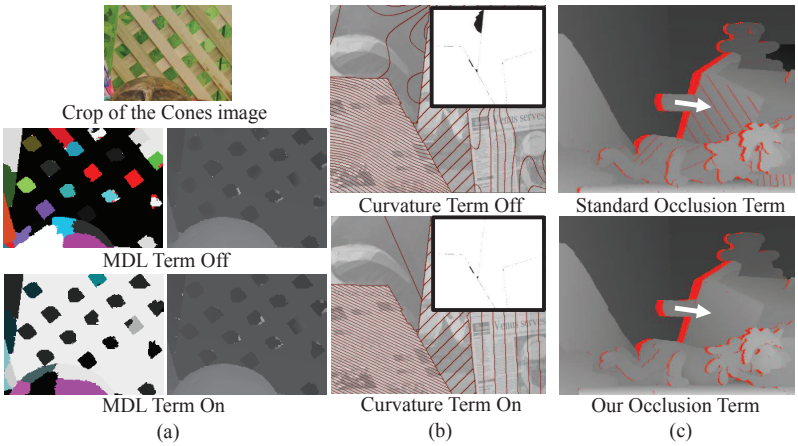


Fig. 14 Effect of different terms in the energy of Surface Stereo [15]. (a) MDL term. (b) Curvature term. Contour lines of the disparity maps are superimposed on the left intensity image. (c) Our improved occlusion handling strategy. See text for explanation.

Let us recall a limitation of segmentation-based methods, i.e., they fail if a segment overlaps a depth discontinuity. This is only partially true for our method, as it can change the shape of segments by adjusting the alpha values within a segment's unknown region. However, the algorithm fails in the presence of large segmentation errors. A more sophisticated solution is discussed next.

Our Contribution - Surface Stereo [15] In our paper [15], we stress the importance of a surface-based representation. Instead of directly assigning pixels to disparity values, we assign them to 3D surfaces. In our implementation, surfaces thereby correspond to planes and B-Splines (to model rounded surfaces). We claim that this surface-based representation is superior in comparison to the commonly-used disparity-based one as it allows incorporating energy terms that are very challenging or even impossible in the disparity-based representation.

Our approach simultaneously infers (1) which surfaces are present in the scene and (2) which pixels belong to which surface. We search an assignment of pixels to surfaces that minimizes an energy. Apart from data and smoothness terms (as in the standard energy of equation (8)), our energy function includes three new terms, i.e., (1) a soft segmentation term, (2) a minimum description length (MDL) term and (3) a curvature term. We will discuss these terms in the following.

As stated above, segmentation-based methods fail if (1) segments overlap disparity discontinuities (see figure 12c) or (2) if the planar model oversimplifies the real 3D shape (e.g., rounded objects). Note that we handle problem (2) by using a more general surface model, i.e., a B-spline model. Problem (1) is handled by our soft segmentation term. This term prefers disparity maps that are consistent with a precomputed color segmentation over such that are not by assigning lower energy. Note that in contrast to most segmentation-based methods, inconsistent disparity maps are allowed (at the price of higher energy) and hence segmentation is only used as a soft constraint. For each pixel p, the soft segmentation term builds the

intersection of $p's$ segment with a squared window centered at p, which gives us p's subsegment. The soft segmentation term imposes 0 costs if all pixels within p's subsegment are assigned to the same surface and a constant penalty, otherwise.[18] Figure 12d demonstrates that our soft segmentation term can handle the Map test set where color segmentation is misleading. The reason is that the data term of our energy overrules the segmentation term and supports the correct disparity solution.

The MDL term implements the assumption that a simple scene explanation is better than an unnecessarily complex one. We therefore impose a penalty on the number of surfaces, i.e., a solution containing five surfaces is cheaper than one with 100 surfaces. Figure 14a shows the effect of the MDL term on a crop of the Middlebury Cones images. The interesting part is the background surface behind the fence. If the MDL term is switched off, the green squared background regions are all modeled by different surfaces. (We use a color coding in figure 14a where all pixels assigned to the same surface carry the same color.) In contrast, a large portion of the isolated background regions are correctly modeled by a single surface if our MDL term is switched on.

Our curvature term puts a penalty on the second order derivative of surfaces. Note that this term is very similar to the second order smoothness term of [82] (see paragraph about smoothness terms), but has two advantages. Firstly, we can analytically and exactly compute the second order derivative, because we have a surface-based representation. This is in contrast to [82] where curvature is only approximated from the disparity values of three neighboring pixels. Secondly, our curvature term represents a unary potential in optimization and we can therefore avoid the complex and time-consuming triple clique construction of [82]. Figure 14b shows the effect of the curvature term. We switch off the curvature term in the top figure. Due to low texture and no penalization of curvature, the spline in the background (top left part of the image) erroneously adapts to the data. This leads to a large disparity error (see black pixels in the error image). When using the curvature term (see bottom figure) representation of the background via a plane leads to lower energy than modeling it as a spline and hence the correct disparity solution is obtained.

Finally, we also make a contribution in the context of occlusion treatment. As stated in the paragraph on occlusion handling, the uniqueness assumption is broken for slanted surfaces where a pixel of one view may have multiple correspondences in the other view. This leads to wrongly detected occlusions. For example, in the top image of figure 14c, occlusions are erroneously detected on the roof of the toy house. (We use red color to mark occlusions.) We overcome this problem by stating that two pixels must not occlude each other if they originate from the same surface. As seen from the bottom image of figure 14c, this avoids wrongly detected occlusions on the roof.

[18] This term forms a higher-order clique in the optimization step. In general, optimization of such higher-order cliques is intractable. We take advantage of the fact that these cliques are sparse, i.e., only one state generates 0 costs and all other states generate constant costs. There exist graph-cut based algorithms that can "efficiently" optimize such sparse higher-order cliques [44, 64] and we make use of them.

(a) (b) (c)

Fig. 15 Object Stereo [16]. (a) Left image of a stereo pair that forms the input of the algorithm. (b) Computed object labeling. (Pixels of the same color lie on the same object.) (c) Computed disparity map.

Let us now focus on optimization of the resulting energy, which is accomplished using fusion moves [50] (see paragraph on graph-cuts). We start with an initial solution. We have a proposal generator that produces six different types of proposal solutions (see paper for details). We now compute the "optimal" fusion move of our current solution and a proposal obtained from our generator.[19] The fusion result is then fused with the next proposal of our generator and so on.

In the experiments, we show that our method achieves an excellent Middlebury ranking, i.e., rank six out of 74 methods at the time of publication. On the complex Teddy test set, our method even takes the first rank on all error measures. Moreover, we demonstrate that each term of our energy contributes to the quality of results (see paper for more information).

Our Contribution - Object Stereo [16] In [16], we combine stereo matching with the object segmentation problem where the task is to label pixels that belong to the same object. Figure 15 gives an example. The method's input is formed by left and right views of a stereo pair. Our algorithm then jointly computes a segmentation of the image into objects (figure 15b) and a disparity map (figure 15c). The idea is that there exist synergies between the stereo and object segmentation problems. We describe our model that is encoded in an energy function in the following.

Our model considers the scene as a collection of a small number of objects.[20] An object is characterized by two aspects. Firstly, an object contains a color model. We use this color model to implement the assumption that pixels of the same object are similar in appearance, i.e., in color. The distribution of colors is thereby modeled using a Gaussian mixture model. Secondly, an object is characterized by a surface model, which implements the assumption that pixels of the same object lie approximately on the same 3D plane. We use a plane-plus-parallax model to allow deviations from this plane. An obvious aspect of our model is that disparities of pixels shall be estimated such that photo consistency is maximized (which represents a standard stereo data term). Finally, we introduce a constraint stating that an object shall be connected in 3D and describe this in more detail below.

[19] By optimal we mean the solution that leads to the largest energy decrease according to our energy function among all possible fusion moves.

[20] Small number means that we apply the MDL term above to penalize the number of objects.

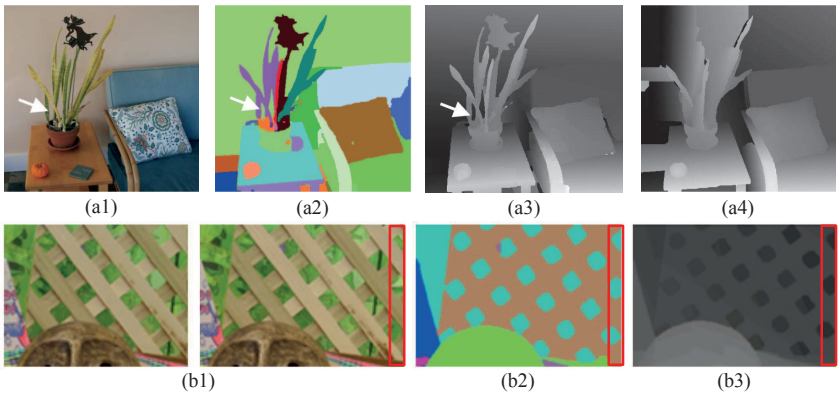


Fig. 16 Results of Object Stereo [16]. We can handle regions of very low texture (a) and regions that are completely occluded (b). (a1) Left image of our Fairy test set. The region marked by the white arrow is difficult to match. (a2) Computed object labeling. (a3) Computed disparity map. Note that the disparity of the background is correctly recovered, whereas the competitor Surface Stereo [15] fails badly in these areas as shown in (a4). (b1) Left and right images of a crop of the Cones images. Note that the red rectangle marks pixels that are completely occluded in the left view. Our method combines all green pixels in a single background object as shown in (b2). This helps to find the correct depth even for the completely occluded image regions (b3).

Our model allows depth reasoning based on the appearance cue. This is because pixels of similar appearance (color) are grouped in the same object and the disparities for these pixels are biased to lie on the object's plane. Advantages of our color-based depth reasoning are twofold. Firstly, similar to Surface Stereo, this allows implementing color segmentation as a soft constraint. Secondly and more importantly, this allows to recover disparity for regions where stereo information is poor (low texture) or not present at all (occlusions).

For example, consider the image of figure 16a1 that has a challenging background of very low texture. Especially, the background region marked by the arrow poses a problem as the data costs for this region are highly ambiguous (no texture) and the pairwise smoothness term motives to assign this isolated region to the foreground disparity. Our algorithm looks at the color of this small region. It finds that the color matches that of the background object "Wall" and biases the disparity towards the plane of the object "Wall" (figure 16a3). Note that our Surface Stereo method described above fails on this challenging pair (figure 16a4), because it does not apply color-based depth reasoning.

Figure 16b shows a challenging occlusion case. We show crops of left and right views of the Middlebury Cones images in figure 16b1. Note that the isolated green background regions inside the red rectangle are only visible in the right image, i.e., they are completely occluded in the left image. This makes correct computation of depth impossible for standard algorithms. Note that standard methods would extrapolate disparity via the smoothness term, i.e., they would assign the occluded background regions to the disparity of the (foreground) fence. In contrast, our method combines all isolated squares into a single background object based on their green

color (figure 16b2). Disparities inside the isolated regions are then biased towards the disparity of the background plane, which gives the correct disparity despite of some green regions being completely occluded (figure 16b3).

Let us now focus on our 3D connectivity constraint mentioned above. This constraint states that disconnected 2D regions in an image may belong to the same object only if they are separated by an occluding object with smaller depth (i.e., closer to the camera). According to this constraint, the green squared background regions of figure 16b1 may all belong to the same object, because they are separated by an occluder of smaller depth, i.e., the fence. In contrast to this, the two green cones of figure 15a must not lie in the same object. Although both cones are similar in color and depth, there is no occluder that separates them. Note that our 3D connectivity constraint demonstrates the usefulness of depth information for the object segmentation task, i.e., this constraint can only work, because we have depth. In practice, it helps to improve object segmentation results.

At the time of publication of the corresponding paper [16], Object Stereo has taken rank 10 on Middlebury and the first rank on the challenging Cones images. Note that the major advantage over other stereo algorithms is that Object Stereo also provides a segmentation of the input images into objects.

Our Contribution - Physics-Based Segmentation and Stereo [14] As in Object Stereo, our paper [14] concentrates on jointly computing a disparity map as well as a segmentation of the input images into objects. One main difference to Object Stereo is that we give objects a third dimension. In contrast to stating that an object is approximately planar (as in Object Stereo), we compute an enclosing 3D bounding box per object where the bounding box represents a proxy for the real 3D extent of the object. One of our 3D reconstructions and bounding boxes of extracted objects are shown in figure 17b. Note that this bounding box representation is more general than the "billboard" world of Object Stereo. For example, in figure 17f, the water kettle is not flat and hence Object Stereo oversegments the kettle using multiple flat objects (see left arrow in figure 17f). In contrast, our bounding box representation allows modeling the water kettle as a single object (see figures 17c and 17d). However, the main argument for using objects with 3D extents is that it enables modeling physical constraints [28], which is discussed in the following.

In particular, we model intersection and gravity constraints. The intersection constraint thereby reasons about occupancy in 3D space, i.e., the spatial extents of 3D objects should not intersect each other. For example, in figure 17f (right arrow), Object Stereo assigns the top and the bottom part of the can to the same object, while the middle part is assigned to a different object. This volume intersection is physically very unlikely, whereas our results in figures 17c and 17d are physically plausible. Our second physics-based constraint, i.e., the gravity constraint, encodes the observation that objects usually do not float in the air, but stand on top of each other due to gravity. Analogously to the intersection constraint, this could not be realized using Object Stereo's surface-based representation.

In the experimental results, we demonstrate that our approach is on par with the state-of-the-art in stereo matching. We also demonstrate that matching accuracy is improved if our physics-based constraints are enabled. A major focus lies on the

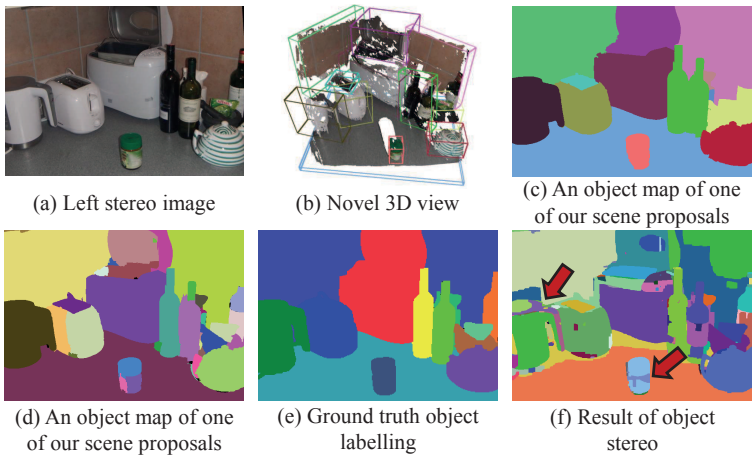


Fig. 17 Our approach [14]. Given a stereo pair (a), our algorithm jointly estimates a 3D reconstruction (b) and object maps (c,d) using physics-based reasoning. The result is considerably closer to ground truth (e) than the one of Object Stereo [16] (f).

object segmentation results. We show that our method can be applied to generate a large set of object proposals. This large set of object proposals has been the key to success in [19], which is the recent winner of the PASCAL object recognition challenge. After automatically ranking the proposals we show experimentally that our results are considerably closer to ground truth than state-of-the-art techniques which either use stereo or monocular images.

4 Summary

In this chapter, we have reviewed the state-of-the-art in stereo matching and highlighted our contributions in this field. We have followed the standard categorization of stereo methods and divided them between local and global algorithms.

Local algorithms operate on windows that are displaced in the right image to find the matching point of lowest color dissimilarity. The implicit assumption that all pixels within the support window have constant disparity is broken at disparity borders, which leads to edge fattening. We have discussed that adaptive weights [86] represent a remedy to the edge fattening problem and have presented our contributions in this domain. We have first presented our geodesic support weight algorithm [36]. A limitation of [36] and [86] is the relatively slow runtime. In this context, we have presented our cost filter approach [62] that allows high-quality adaptive support weight matching in real-time. We have then focused on the problem of reconstructing slanted surfaces where standard local methods show poor performance due to using fronto-parallel windows. We have presented our PatchMatch Stereo work [13] that overcomes this problem by using slanted support windows.

In the discussion of global methods, we have started by describing a standard stereo energy function and then briefly focused on the problem of optimizing this energy. We have discussed the following optimization techniques: dynamic programming (including our Simple Tree method [11]), graph-cuts and message passing. However, the real problem is not in the optimization, but in finding an energy function that represents a good model of the stereo problem. Hence, we have concentrated on improving the model in the remainder of this chapter.

We have discussed the data term of our energy and presented different match measures. In the context of data terms, we have presented our work [7], which investigates the usefulness of color information in stereo matching. We have then explained how to incorporate occlusion treatment into the data terms of global methods. In the discussion of smoothness terms, we have presented first and second order terms and stressed the importance of edge-sensitive and highly-connected smoothness terms. After a brief overview of segmentation-based methods, we have explained our combined stereo and matting method [12]. Finally, we have presented our other papers on improving the stereo model. This has included Surface Stereo [15], Object Stereo [16] and our physics-based approach [14].

Acknowledgements This work was supported in part by the Vienna Science and Technology Fund (WWTF) under project ICT08-019.

References

1. S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building rome in a day. In *ICCV*, 2009.
2. S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 92(1):1–31, 2011.
3. S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *CVPR*, pages 434–441, 1998.
4. C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (SIGGRAPH)*, 2009.
5. S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *TPAMI*, 20(4):401–406, 1998.
6. S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. *IJCV*, 35(3):269–293, 1999.
7. M. Bleyer and S. Chambon. Does color really help in dense stereo matching? In *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, 2010.
8. M. Bleyer, S. Chambon, U. Poppe, and M. Gelautz. Evaluation of different methods for using colour information in global stereo matching. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XXXVII, pages 415–422.
9. M. Bleyer and M. Gelautz. A layered stereo matching algorithm using image segmentation and global visibility constraints. *ISPRS Journal*, 59(3):128–150, 2005.
10. M. Bleyer and M. Gelautz. Graph-cut-based stereo matching using image segmentation with symmetrical treatment of occlusions. *Signal Processing: Image Communication*, 22(2):127–143, 2007.
11. M. Bleyer and M. Gelautz. Simple but effective tree structures for dynamic programming-based stereo matching. In *VISAPP*, volume 2, pages 415–422, 2008.

12. M. Bleyer, M. Gelautz, C. Rother, and C. Rhemann. A stereo approach that handles the matting problem via image warping. In *CVPR*, pages 501–508, 2009.
13. M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo - stereo matching with slanted support windows. In *BMVC*, 2011.
14. M. Bleyer, C. Rhemann, and C. Rother. Extracting 3d scene-consistent object proposals and depth from stereo images. In *ECCV*, 2012.
15. M. Bleyer, C. Rother, and P. Kohli. Surface stereo with soft segmentation. In *CVPR*, 2010.
16. M. Bleyer, C. Rother, P. Kohli, D. Scharstein, and S. Sinha. Object stereo joint stereo matching and object segmentation. In *CVPR*, 2011.
17. A. Bobick and S. Intille. Large occlusion stereo. *IJCV*, 33(3):181–200, 1999.
18. Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, 2001.
19. J. Carreira, F. Li, and C. Sminchisescu. Object recognition by sequential figure-ground ranking. *IJCV*, 2012.
20. Y. Deng, Q. Yang, X. Lin, and X. Tang. A symmetric patch-based correspondence model for occlusion handling. In *ICCV*, pages 542–567, 2005.
21. O. Faugeras, B. Hotz, H. Mathieu, T. Viéville, Z. Zhang, P. Fua, E. Théron, L. Moll, G. Berry, J. Vuillemin, P. Bertin, and C. Proy. Real time correlation based stereo: algorithm implementations and applications. Technical report, RR-2013, INRIA, 1996.
22. P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1), 2006.
23. P.V. Fua. Combining stereo and monocular information to compute dense depth maps that preserve depth discontinuities. In *International Joint Conference on Artificial Intelligence*, pages 1292–1298, 1991.
24. A. Fusiello, V. Roberto, and E. Trucco. Efficient stereo with multiple windowing. In *CVPR*, pages 858–863, 1997.
25. D. Gallup, J. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *CVPR*, 2007.
26. S. Gehrig and U. Franke. Improving sub-pixel accuracy for long range stereo. In *ICCV VRML workshop*, 2007.
27. A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.
28. A. Gupta, A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*, 2010.
29. R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. 2003.
30. S. Hasinoff, S. B. Kang, and R. Szeliski. Boundary matting for view synthesis. *CVIU*, 103(1):22–32, 2006.
31. K. He, J. Sun, and X. Tang. Guided image filtering. In *ECCV*, 2010.
32. H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *CVPR*, volume 2, pages 807–814, 2005.
33. H. Hirschmüller, P. Innocent, and J. Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *IJCV*, 47:229–246, 2002.
34. H. Hirschmüller and D. Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *PAMI*, 31:1582–1599, 2009.
35. L. Hong and G. Chen. Segment-based stereo matching using graph cuts. In *CVPR*, volume 1, pages 74–81, 2004.
36. A. Hosni, M. Bleyer, and M. Gelautz. Near real-time stereo with adaptive support weight approaches. In *3DPVT*, 2010.
37. A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann. Local stereo matching using geodesic support weights. In *ICIP*, 2009.
38. A. Hosni, C. Rhemann, M. Bleyer, and M. Gelautz. Temporally consistent disparity and optical flow via efficient spatio-temporal filtering. In *PSIVT*, 2011.
39. A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *PAMI (under review)*, 2012.

40. H. Ishikawa. *Global Optimization Using Embedded Graphs*. PhD thesis, New York University, 2000.
41. S. Jegelka and J. Bilmes. Submodularity beyond submodular energies: coupling edges in graph cuts. In *CVPR*, 2011.
42. M. Ju and H. Kang. Constant time stereo matching. In *MVIP*, pages 13 – 17, 2009.
43. A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *ICPR*, pages 15–18, 2006.
44. P. Kohli, M. Kumar, and P. Torr. P3 & beyond: Solving energies with higher order cliques. In *CVPR*, 2007.
45. V. Kolmogorov and C. Rother. Minimizing non-submodular functions with graph cuts - a review. *PAMI*, 29(7):1274–1279, 2007.
46. V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *ICCV*, volume 2, pages 508–515, 2002.
47. V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *ECCV*, 2002.
48. V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *TPAMI*, 26(2):147–159, 2004.
49. P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems*, 2011.
50. V. Lempitsky, C. Rother, and A. Blake. Logcut - efficient graph cut optimization for markov random fields. In *ICCV*, 2007.
51. G. Li and S. W. Zucker. Surface geometric constraints for stereo in belief propagation. In *CVPR*, pages 2355–2362, 2006.
52. M. Lin and C. Tomasi. Surfaces with occlusions from layered stereo. In *CVPR*, pages 710–717, 2003.
53. X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang. On building an accurate stereo matching system on graphics hardware. In *GPUCV*, pages 467 – 474, 2011.
54. T. Meltzer, T. C. Yanover, and Y. Weiss. Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation. In *ICCV*, pages 428 – 435, 2005.
55. K. Mühlmann, D. Maier, J. Hesser, and R. Männer. Calculating dense disparity maps from color stereo images, an efficient implementation. *IJCV*, 47(1):79–88, 2002.
56. R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, 2011.
57. A. S. Ogale and Y. Aloimonos. Stereo correspondence with slanted surfaces: critical implications of horizontal slant. In *CVPR*, pages 568–573, 2004.
58. Y. Ohta and T. Kanade. Stereo by intra- and inter- scanline search. *TPAMI*, 7(2):139–154, 1985.
59. S. Paris and F. Durandi. A fast approximation of the bilateral filter using a signal processing approach. In *IJCV*, volume 81, pages 24 – 52, 2009.
60. F. Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *CVPR*, volume 1, pages 829 – 836, 2005.
61. R. Yang S. Wang M. Liao Q. Yang, L. Wang and D. Nister. Real-time global stereo matching using hierarchical belief propagation. In *BMVC*, 2006.
62. C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *CVPR*, 2011.
63. C. Richardt, D. Orr, I. Davies, A. Criminisi, and N.A. Dodgson. Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid. In *ECCV*, volume 6313, pages 510 – 523, 2010.
64. C. Rother, P. Kohli, W. Feng, and J. Jia. Minimizing sparse higher order energy functions of discrete variables. In *CVPR*, pages 1382–1389, 2009.
65. C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23:309–314, 2004.
66. C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary mrfs via extended roof duality. In *CVPR*, 2007.

67. S. Roy and I. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *ICCV*, pages 492–499, 1998.
68. D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1/2/3):7–42, 2002. http://vision.middlebury.edu/stereo/.
69. J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from a single depth image. In *CVPR*, 2011.
70. B. Smith, L. Zhang, and H. Jin. Stereo matching with nonparametric smoothness priors in feature space. In *CVPR*, pages 485–492, 2009.
71. N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, 25:835–846, 2006.
72. J. Sun, Y. Li, S.B. Kang, and H.Y. Shum. Symmetric stereo matching for occlusion handling. In *CVPR*, volume 25, pages 399–406, 2005.
73. J. Sun, N.N. Zheng, and H.Y. Shum. Stereo matching using belief propagation. *PAMI*, 25(7):787–800, 2003.
74. R. Szeliski and P. Golland. Stereo matching with transparency and matting. In *ICCV*, pages 517–525, 1998.
75. R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields. In *ECCV*, volume 2, pages 19–26, 2006.
76. Y. Taguchi, B. Wilburn, and L. Zitnick. Stereo reconstruction with mixed pixels using adaptive over-segmentation. In *CVPR*, pages 1–8, 2008.
77. H. Tao, H. Sawhney, and R. Kumar. A global matching framework for stereo computation. In *ICCV*, pages 532–539, 2001.
78. M.F. Tappen and W.T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *ICCV*, volume 2, pages 900–906, 2003.
79. O. Veksler. Stereo correspondence with compact windows via minimum ratio cycle. *IEEE TPAMI*, 24(12):1654–1660, 2002.
80. O. Veksler. Stereo correspondence by dynamic programming on a tree. In *CVPR*, pages 384–390, 2005.
81. M. Wainwright, T. Jaakkola, and A. Willsky. Tree reweighted belief propagation and approximate ml estimation by pseudo-moment matching. In *AISTATS*, 2003.
82. O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. In *CVPR*, 2008.
83. W. Xiong and J. Jia. Stereo matching on objects with fractional boundary. In *CVPR*, pages 1–8, 2007.
84. Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister. Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. *PAMI*, 2009.
85. Q. Yang, R. Yang, J. Davis, and D. Nister. Spatial-depth super resolution for range images. In *CVPR*, 2007.
86. K.J. Yoon and I.S. Kweon. Locally adaptive support-weight approach for visual correspondence search. In *CVPR*, 2005.
87. K. Zhang, G. Lafruit, R. Lauwereins, and L. Gool. Joint integral histograms and its application in stereo matching. In *ICIP*, pages 817 – 820, 2010.
88. K. Zhang, J. Lu, and G. Lafruit. Cross-based local stereo matching using orthogonal integral images. In *TCSVT*, volume 19, pages 1073 – 1079, 2009.
89. Y. Zhang, M. Gong, and Y. Yang. Local stereo matching with 3D adaptive cost aggregation for slanted surface modeling and sub-pixel accuracy. In *ICPR*, 2008.
90. L. Zitnick, S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. *ACM Transaction on Graphics*, 23(3):600–608, 2004.