



Coarse-to-Fine Stereo Vision with Accurate 3-D Boundaries

Mikhail Sizintsev

Richard P. Wildes

Technical Report CS-2006-07

June 28, 2006

Department of Computer Science
4700 Keele Street North York, Ontario M3J 1P3 Canada

Coarse-to-Fine Stereo Vision with Accurate 3-D Boundaries

Mikhail Sizintsev
Richard P. Wildes

Department of Computer Science and Engineering
and the Centre for Vision Research
York University
Toronto, Ontario M3J 1P3
Canada

Abstract

This paper presents methods for recovering accurate binocular disparity estimates in the vicinity of 3-D surface discontinuities. Of particular concern are methods that impact coarse-to-fine, local block-based matching as it forms the basis of the fastest and the most resource efficient stereo computation procedures. Several advances are put forth. First, a novel coarse-to-fine refinement that adapts match window support across scale to ameliorate corruption of disparity estimates near boundaries is presented; a detailed analysis of coarse-to-fine 3-D boundary processing is given as well. Second, a novel formulation of half-occlusion cues within the coarse-to-fine block matching framework is described; the relation of the proposed solution to previous methods is extensively discussed. Third, the use of colour or intensity segmentation for better recovery of 3-D boundaries is investigated; a formulation specific to a coarse-to-fine local block-based matching is given. Empirical results show that incorporation of these advances in the standard coarse-to-fine, block matching framework reduces disparity errors by a factor of two, while performing little extra computation and preserving the parallel/pipeline nature of the framework.

Contents

1	Introduction	2
1.1	Motivation	2
1.2	Problem structure	4
1.2.1	Stereo overview	4
1.2.2	Challenges of correspondence	4
1.2.3	Constraints	5
1.3	Previous correspondence methods	8
1.3.1	Matching primitives	8
1.3.2	Local methods	9
1.3.3	Cooperative methods	9
1.3.4	Global methods	10
1.3.5	Coarse-to-fine	11
1.3.6	Challenges revisited	12
1.3.7	Speed-accuracy tradeoff	15
1.4	Contributions	15
1.5	Outline of report	16
2	Technical approach	17
2.1	Block matching algorithm	17
2.2	Adaptive coarse-to-fine stereo for 3-D boundary preservation	19
2.2.1	Basic algorithm	19
2.2.2	Analysis of coarse-to-fine stereo: Boundary deterioration	22
2.2.3	Improving coarse-to-fine block-based stereo	36
2.2.4	Coarse-to-fine non-block-based stereo	43
2.3	Half-Occlusions	45
2.3.1	Geometry of half-occlusions	45
2.3.2	Occlusions and slanted surfaces	48
2.3.3	Cues to half-occlusion detection	50
2.3.4	Occlusions in coarse-to-fine stereo	52
2.3.5	Final half-occlusion detection algorithm	53

2.4	Colour and intensity segmentation in computational stereo	56
2.4.1	Segmentation-driven shiftable windows	56
2.4.2	Relation to other segmentation-based windows	60
2.4.3	Precision versus robustness	61
2.5	Recapitulation	66
3	Experimental evaluation	67
3.1	Methodology	67
3.2	Adaptive coarse-to-fine processing	71
3.3	Half-occlusions	76
3.3.1	Comparison to previous approaches	76
3.4	Colour and intensity segmentation cues	81
3.4.1	Comparison to previous colour-cue formulations	85
3.5	Other variations of stereo algorithm	87
3.6	Final comparison	93
4	Discussion: Relations to alternative disparity estimation frameworks	97
4.1	Speed-accuracy tradeoff	97
4.1.1	Time complexity	98
4.1.2	Memory Complexity	101
4.2	Parallelization	101
4.3	Anytime computation	102
4.4	Additional considerations for practical stereo vision	103
4.4.1	Parameter tuning	103
4.4.2	Sensitivity to noise	103
5	Conclusion	105
5.1	Adaptive coarse-to-fine stereo	105
5.2	Binocular half-occlusions	106
5.3	Colour and intensity cues	106
A	Mutual Information (MI) for stereo correspondence	107

Chapter 1

Introduction

1.1 Motivation

As soon as someone asks the question “How do we perceive the world visually?”, a second question of no less importance arises. Our spatial world is three-dimensional, while images captured by our eyes or cameras are always two-dimensional, as in Figure 1.1. So, how is this third dimension, i.e. *distance*, recovered?

By looking at a painting, drawing or photograph, we perceive the rendered scene as three-dimensional with the help of a variety of single-image depth cues such as perspective, contour, texture, aerial perspective and shading. Successful use of these cues requires a set of assumptions and prior experience and all of them have been replicated in computer algorithms with various level of success – techniques known as “Shape-from-X” [45].

An alternative way to infer depth is to gather more information by taking several images, but from different view-points. Images can be taken from different places (multiple view stereo), different points in time (structure from motion) and even with different focal points (depth from focus). This paper concentrates on the multiple-view approach, and on two-views specifically. Of possible configurations, binocular imaging (i.e. two view stereo) has been a particularly well researched situation as it provides the minimal multiview situation.

Another reason to choose the binocular imaging out of multi-view configurations is that it reflects biological design and there is a potential for cross fertilization between research in artificial and natural binocular stereo. The research in human and animal stereo is immense and useful information discovered in psychophysical labs may be used during design of artificial binocular perception. In complement, computational analysis can suggest potentially fruitful paths of investigation in the study of natural systems. Also, computational realization offers evaluation of the biological models for their ability to mechanistically extract stereoscopic measurements from visual data.

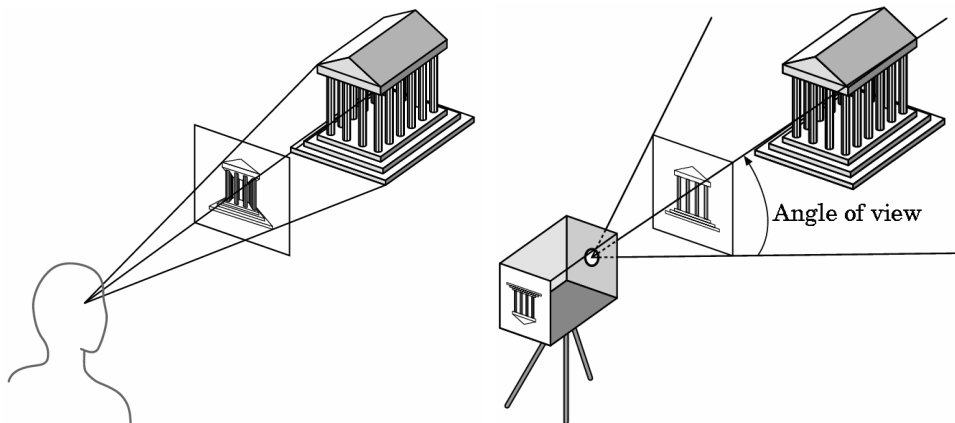


Figure 1.1: 2-D Imaging of a 3-D World.

The ability to automatically perceive (i.e. reconstruct) a scene in 3-D is extremely useful in practice, as many applications critically depend on it. One such application is robotics, where robots must operate in environments that are dangerous or unreachable by humans. Specific examples include space (autonomous planetary exploration, inspection of aircraft on orbit), underground mining, autonomous vehicles for military operations and aids in driving [99, 45]. Moreover, knowledge of 3-D information is required for many consumer applications like portable scene modelers [103] and augmented reality, where dense 3-D information is important for correct rendering of occlusions of virtual objects by real objects [63].

To make these applications practical, the underlying recovery of 3-D measurements must be accurate, rapid and require little in the way of special purpose hardware. No existing technology can respond to these demands. Active sensing technologies (e.g., sonar, lidar, structured light etc.) are based on emitting energy into the environment and analyzing the reflected pattern [99, 45]. They require special purpose hardware (e.g. laser, projector) that is bulky, expensive, and power consuming. Passive sensing approaches, such as computer multi-view stereo vision, are robust and very cheap alternatives, because only cameras (minimum of two in case of binocular stereo) and a computer are required, and no energy emission is involved. However, this technology is hampered by poor speed/accuracy trade-offs and improper reconstructions near object boundaries.

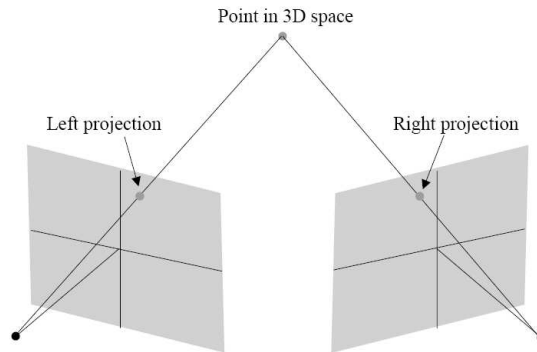


Figure 1.2: Stereo Geometry for Two Perspective Cameras. A 3-D point in space is projected on two spatially displaced cameras.

1.2 Problem structure

1.2.1 Stereo overview

The stereo problem is very easy to state (but, unfortunately, not easy to solve) once one considers the geometry behind it. The situation for a single 3-D point and two perspective cameras [51] is depicted in Figure 1.2. The basis behind the process of inferring actual depth is the search for the projection of the same 3-D point across images and calculation of *disparity* – a difference in image coordinates between those projections. Once the corresponding projections are found, the absolute 3-D coordinates of the world point are completely determined via triangulation, provided that the stereo rig is calibrated [51]. Similarly, we can reconstruct the whole scene point by point. Within this framework, triangulation and calibration are relatively straightforward and well understood; whereas, correspondence remains challenging.

1.2.2 Challenges of correspondence

The recovery of corresponding points across binocular views is a hard problem even when the assumption of Lambertian surfaces [117] is in use¹. More specifically, relying solely on an intensity-based matching function is generally not enough. First, points in correspondence might not look alike, because data contains noise, stereo images are hampered by projective distortions and differences in cameras' settings. Second, points that look alike are not necessarily in correspondence due to repetitive texture, or homogenous regions,

¹Recall that surface is Lambertian if its luminance is the same regardless of the viewing direction and depends on the cosine of the angle between the local surface normal and the illumination direction.

where no distinct points can be identified. Technically speaking, matching is underconstrained in such situations.

Other fundamentally hard regions for establishing correspondences are in the vicinity of 3-D boundaries. This problem is typical for computer vision processing methods that have to deal with noisy data and use low-pass filtering techniques to regularize the solution. While such methods alleviate difficulty with high frequency noise, they also inhibit recovery of high-frequency details, like exact discontinuity locations. The problem of accurate and reliable recovery of 3-D boundaries is very important by itself, as many applications, such as robotic manipulation and 3-D reconstruction, critically depend on accurate depth discontinuity information. Moreover, humans are very sensitive to 3-D boundaries and are able to recover them with precision greater than spacing of photoreceptors on the retina, i.e. they exhibit stereo hyperacuity [57], which proves that nature has a good solution for recovery of 3-D discontinuities, and it is yet to be discovered.

As established so far, computational stereo algorithms try to find points in correspondence. However, for some points in the scene the correspondence cannot be found in principle – those points are called half-occluded, as they are seen only in one of the views of the stereo pair. Thus, a good stereo algorithm must not only find the points in correspondence, but also explicitly say which points have no match. Interestingly, as early as Euclid, the basic geometric relationship that gives rise to half-occlusion was documented [26]. Further, the potential perceptual significance of binocular half-occlusion has been known at least since the time of Leonardo Da Vinci [97]. Much more recently, the fact that humans actually do exploit half-occlusions in making depth inferences was documented [73]. Subsequently, a great number of psychophysical studies of half-occlusion have supported their use by humans (see [57] for review); however, the enabling computations remain unclear.

Many other problems can arise during stereo matching, such as various types of noise, specularities, aerial diffusion, transparencies etc. Some of them (noise, specularities) are partially treated by the design of the appropriate match measures [13, 102, 21, 113]. Others might need the extension of the stereo model to adapt to the specific situation, such as underwater stereo [96]. Still others require a novel insight into the stereo problem from the very beginning, e.g. transparencies [112].

1.2.3 Constraints

To deal with correspondence challenges effectively, various constraints on stereo matching are used.

There exists a fundamental constraint for points in two-view geometry known as the epipolar constraint. Conceptually, it means that corresponding points exist only along the epipolar lines², which changes the general 2-D correspondence search problem to a 1-D

²Given a point in three space and two centres of projection that define their encompassing plane P ,

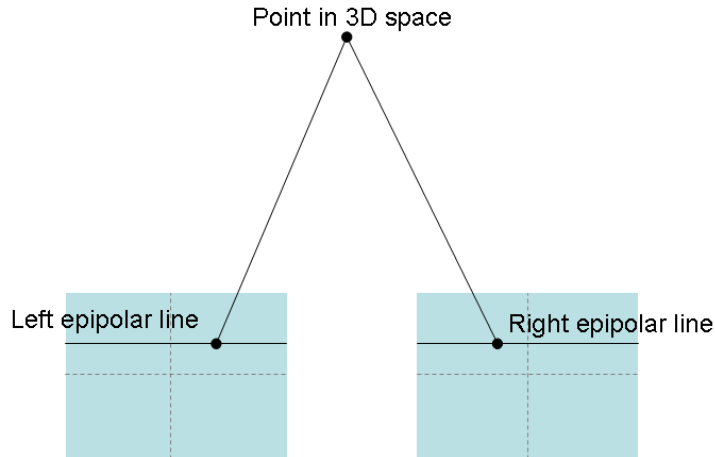


Figure 1.3: Geometry of Nonvergent Stereo. Left and right image planes lie in the same world plane and their axes are aligned. All epipolar lines are parallel to the horizontal axis and the vertical disparity component is always zero.

search problem – a great reduction in possibility of error and processing time. For these reasons, stereo setups with parallel camera axes are usually used (nonvergent geometry) with epipolar lines lying along the horizontal axis (Figure 1.3). Refer to the horizontal axis as the x -axis and the vertical as the y -axis. Note that in case of nonvergent geometry disparity is just the difference in the left and right image x -coordinate and is inversely proportional to depth, while the orthogonal y -coordinate is the same in both images. Alternatively, if the stereo setup is convergent (cameras’ optical axes are not parallel), then images can be pre-warped by homographies to make their epipolar lines lie parallel to the x -axis, i.e. they become *rectified* (Figure 1.4). Today, computational stereo heavily relies on rectification [102, 21] (and hence, on the epipolar constraint) due to existence of fast and robust rectification procedures [21, 51].

A fundamental technique to make the correspondence solution more stable is to assume spatial smoothness, or cohesion, which means that points belonging to a single object tend to reside at a certain near-constant depth. Smoothness is usually enforced by penalizing neighbouring points that have different depths (and hence, disparities) or by assuming that neighbouring points reside at the same depth by aggregation and matching the aggregated regions. Actual mechanisms for application of smoothness can vary, but any contemporary stereo algorithm includes this constraint. In fact, algorithms that only rely on intensity-based pixel matching, epipolar geometry and smoothness are among the state-of-the-art solutions [20, 111].

The uniqueness constraint forces points in both images to have at most one corresponding epipolar lines are defined by the intersection of P with the imaging planes.

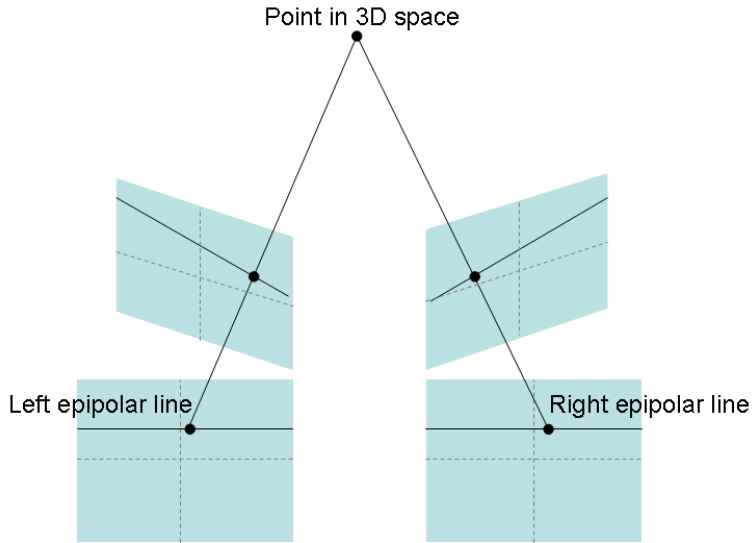


Figure 1.4: Binocular Rectification. Images are warped by homographies such that warped images can be treated as coming from the parallel stereo setup, as in Figure 1.3.

spondence. This constraint comes directly from the geometry, when opaque objects are assumed. However, uniqueness is not easy to apply correctly, because it is stated for points, while stereo algorithms deal with pixels. Trivially, consecutive points on a slanted surface have slightly different disparities, which can easily be quantized to the same pixel disparity value (these difference between disparities is less than 1 pixel), i.e. they will violate uniqueness, which makes non-fronto-parallel surfaces hard to recover.

The ordering constraint states that the order of points along an epipolar line in one view will be the same as in the other. This constraint was initially inspired by the Disparity Gradient Limit [23] formulated for biological stereo vision. The ordering assumption considerably simplifies the matching procedure by significantly pruning the correspondence search space; at the same time, some valid configurations, such as thin foreground structures, violate this constraint, hence will not be recovered correctly.

The occlusion constraint says that half-occlusions in the left-based disparity map correspond to occluded regions in the right-based disparity map and vice versa. This constraint is derived directly from the definition of half-occlusion and has proven to be one of the best methods for half-occlusion detection [39, 110].

The colour or intensity segmentation cue relies on the fact that 3-D object boundaries are very likely to coincide with colour or intensity edges. This cue can be very beneficial for stereo as it gives additional information on how to distinguish between discontinuities and smooth 3-D surfaces [102, 21]. However, this cue cannot be strictly interpreted as a constraint, because two objects may reside at different depths, but be of absolutely

the same colour, i.e. no meaningful intensity edges can be found to separate them in monocular images; alternatively, an object can have strong colour edge elements that do not correspond to any depth changes, i.e. texture edges.

In general, application of these cues and constraints may greatly improve the quality of the recovered stereo disparity [102, 21].

1.3 Previous correspondence methods

Here we want to overview the major stereo correspondence algorithms resulting from previous research, and discuss how each algorithm solves the challenges outlined in Section 1.2.2.

1.3.1 Matching primitives

Section 1.2.1 focused discussion of the stereo problem onto the search for corresponding points. However, nothing was mentioned about what these points might be. Generally, the choice of entity to match depends on the nature of the data. It could be just a plain point in binary images [83], pixel intensities [102], corners [88], edges [9], lines [86], contours [76], phase-based features [42], oriented bandpass filter responses [60], SIFT features [80], colour segments [115] and many others. More complex features are easier to match, as they bear more information and identity, i.e. they are more discriminative; they are also sparse, which substantially reduces the computation time; but, unfortunately, they result in a sparse depth map. Simple features, i.e. pixel intensities, are much more ambiguous, but they do not have to be explicitly extracted and produce disparity maps of maximum density, i.e. depth is estimated for (almost) every pixel.

The algorithms that use simple but dense attributes like pixels have been coined “area-based stereo”, while algorithms that use more distinctive attributes, e.g. edges, corners, etc. are known as “feature-based stereo”. A great deal of early work in stereo was accomplished in the field of photogrammetry, which, e.g., is interested in the automatic reconstruction of three-dimensional terrain models from stereo fly-overs acquired by a plane. There, researchers have extensively exploited correlation methods [70], hence, area-based approaches, to get dense depth maps. In contrast, initial computer vision research concentrated mainly on feature-based algorithms, as computational power was insufficient to perform fast dense stereo and dense stereo itself did not produce satisfying results at that time. An early review of feature-based stereo methods can be found in [10, 36]. As time progressed, computers became faster and the demand for dense depth maps increased, which made people turn most attention back to area-based, or rather pixel-based stereo. Comprehensive reviews of recent advances in pixel-based stereo can be found in [102, 21]. Scharstein and Szeliski have developed a taxonomy for modern stereo algorithms to “allow

the dissection and comparison of individual algorithm components design decisions” [102]; moreover, the authors have organized an interactive website where anyone can evaluate their stereo algorithm on a quite complex dataset and be ranked among other solutions [3]. As this paper is concerned with the recovery of dense depth maps, we review the major approaches to area-based stereo below taking into account the taxonomy proposed in [102].

1.3.2 Local methods

The simplest method to do stereo matching on graylevel images would be to compare each pixel’s intensity in the reference image with pixels in the other image along an epipolar line using some match cost function, and then choose the match (hence, disparity) which minimizes this function – a strategy known as Winner-Take-All (WTA). As matching based on a single pixel is very unlikely to work (due to difficulties outlined in Section 1.2.2), we typically define an aggregation window around each pixel and match the windows instead [117]. The match cost function itself can be as simple as an absolute, or squared intensity difference; it can be normalized to be robust to different camera gains. Alternatively, rank-order statistics of intensity, instead of values themselves can be used. Further, an arbitrary one-to-one mapping between intensity values can be derived during the matching procedure (e.g. using a technique known as Mutual Information [123, 38]). More detailed discussion about match metrics and their comparison can be found in [21].

The described algorithm is a basic stereo algorithm where photometric matching is applied via a similarity cost function, smoothness is applied locally via an aggregation window and disparity decision is made based on a trivial WTA procedure:

$$d = \forall p | : arg \min_{d_p} \left(\sum_{q \in \mathcal{N}(p)} E_{data}(d_q) \right) \quad (1.1)$$

where d is entire disparity map, p is a point on the map, $\mathcal{N}(p)$ is a aggregation region of a point and $E_{data}(d_p)$ is a intensity-based dissimilarity measure for point p and disparity assigned to it, d_p .

While the choice of cost function and window sizes and shapes can vary widely, this kind of algorithm is very easy to code, can be completely parallelized and forms the backbone of today’s fastest methods. In fact, the overwhelming majority of real-time algorithms are local area-based methods [21].

1.3.3 Cooperative methods

The result of the local area-based method outlined in Section 1.3.2 can be improved when disparity estimates are iteratively updated by further enforcement of a smoothness constraint between neighbours. Such algorithms are named “cooperative” and realized by

diffusing reliable matches to neighbours and inhibiting values along the view-directions of the left or right eye, i.e. enforcing the uniqueness constraint.

Historically, cooperative methods were inspired by the computational models of human stereopsis [83]. The algorithm of Zitnik and Kanade [132] is an excellent example of a contemporary cooperative stereo method, which can be seen as an extension to [83]. Later work mainly concentrated on improving various aspects of this algorithm, like better 3-D boundary localization [131, 85, 92, 125].

1.3.4 Global methods

Local (and cooperative) methods employ local aggregation to get meaningful matching results. Considering that aggregation is motivated by smoothing, stereo processing can be essentially seen as pixel-wise intensity matching plus a penalty for neighbouring pixels having different disparities for smoothness. Global algorithms formulate this principle directly as an energy objective function over all pixels, using a Markov Random Field Assumption [102, 21, 116, 122]:

$$E(d) = \sum_p E_{data}(d_p) + \lambda \sum_{q \in \mathcal{N}(p)} E_{smooth}(d_p, d_q) \quad (1.2)$$

where d is the entire disparity map, p is a point in the map, $\mathcal{N}(p)$ is a neighbourhood of a point, λ is a smoothness parameter, $E_{data}(d_p)$ is a photometric-based dissimilarity measure for point p and disparity assigned to it, d_p , and $E_{smooth}(d_p, d_q)$ is a penalty for nearby pixels having different disparity values. Many other terms like colour segmentation, uniqueness, ordering, occlusion can be added to the objective function. In this formulation the disparity map d can be obtained as

$$d = \operatorname{argmin}(E(d)) \quad (1.3)$$

The whole problem now is to minimize this function, i.e. solve (1.3). Fortunately, a number of efficient optimization methods have been developed and applied to global stereo such as dynamic programming [30, 58, 32, 21, 52, 122], loopy belief propagation [111, 41, 110], graph cuts [98, 14, 20, 69, 64, 77, 61, 5, 55, 35], stochastic diffusion [104, 101, 74], PDE [108, 6], genetic algorithms [49] and others [21]. Such optimization methods give extremely good empirical results in comparison to local methods [102, 21]; however, being global, the algorithms are very computationally and memory-intensive and cannot be easily parallelized.

It is worthwhile noting that it is possible to design an algorithm that combines various aspects from each of the classes mentioned above. For example, one can initially aggregate

matches using local support windows, and then make a final disparity assignment based on optimization of a global function (1.2) instead of a simple WTA procedure (e.g. aggregation windows together with dynamic programming [47, 109]). It is also possible to treat the contribution of each point differently based on the reliability of its matching – techniques known as Ground Control Points (GCP) [17] and Unambiguous Matching Components [100].

1.3.5 Coarse-to-fine

In constructing a disparity map, each pixel has a whole range of possible values, and this whole range should be tried to find the optimal assignment. Luckily, the idea of hierarchical processing, or coarse-to-fine, can be applied to stereo.

Coarse-to-fine disparity estimation operates as follows. Initially, images are brought into a pyramid representation where the base level captures the original image, while successive levels capture coarser resolutions with smaller format images via spatial subsampling, applied after low-pass or band-pass filtering (to avoid aliasing). The most widely used pyramids are Quadtree [59], Gaussian and Laplacian [24]. In turn, coarse-to-fine stereo operates by initially estimating disparity for lower resolution images (hence, images of smaller size), then taking these disparities as an offset for refinement using higher resolution (ultimately the original) images. Note that search range can be smaller for low resolution images, because images themselves are smaller. The refinement step is considerably cheaper than calculation from scratch, because the local search range is smaller given the initial disparity offset. This procedure is preformed recursively by doing progressive matching starting from the coarsest pyramid level to the finest level.

Speed is not the only reason to employ coarse-to-fine processing. It also helps to remove local minima in correspondence search by their reduction at the coarse level and allows for variable support aggregation as support region of the same size (in terms of pixels) yields greater smoothness at coarser levels. These properties are exceptionally beneficial for local algorithms, which use fixed support and simple WTA optimization procedures.

Another interesting property of coarse-to-fine stereo is that it can be treated as an anytime algorithm, because intermediate processing results correspond to the final depth map but at coarser resolution. This property is very useful for certain applications, such as hard real-time systems, where it is essential to get at least partial result as soon as the algorithm is interrupted [99].

The coarse-to-fine approach has disadvantages as well. Mistakes made early at the coarse resolution can be difficult to correct. Moreover, coarse-to-fine processing experiences difficulties in recovering thin structures and shows inferior performance near 3-D boundaries, because these are high-frequency details that are unavailable at coarser scales.

Coarse-to-fine stereo appeared almost simultaneously with the first computational stereo algorithms [83, 95]. Since then it is constantly used in real-world applications and new algo-

rithms keep appearing [118, 89, 127, 29]. Interestingly, any disparity estimation procedure can be modified to work in a coarse-to-fine fashion and recent global approaches, mostly the ones that use dynamic programming, even achieved real-time performance [109, 87, 44].

1.3.6 Challenges revisited

Now that the major area-based algorithms for dense stereo have been stated, it is important to discuss how each of them solves the main challenges of the correspondence problem that have been summarized in Section 1.2.2.

Textureless regions

In the case of textureless regions, the calculated disparity is a result of the smoothness constraint entirely – disparity estimates are interpolated between near locations of textured patches for which disparity estimates can be found. In such situations, the result of a computational stereo algorithm depends on the mechanism of smoothness enforcement. No wonder that cooperative and global methods solve this problem reasonably well, as support can be accumulated over the whole image, if necessary.

At the same time, local area-based methods experience difficulties in such areas, because the areas of textureless regions can always be bigger than the size of the support window, while local methods cannot use very big windows because it heavily impacts results near 3-D boundaries (discussed below). In the past, researchers developed various heuristics to attack this problem by explicitly detecting unambiguous regions [100], checking if the minimum selected by WTA procedure is significantly lower than other competing values [54, 48, 107], calculating a curvature of the correlation function [8], adaptively growing the support region by designing more complex rules for aggregation window construction [62, 19, 121] and others [21].

Coarse-to-fine area-based methods significantly improve on single-scale local matchers in textureless regions, as they are able to aggregate greater support at coarse levels, but they cannot solve this problem completely [7].

3-D boundaries

The accurate and reliable recovery of 3-D boundaries is still an outstanding problem for the whole computational stereo community. Performance near depth discontinuities critically depends on how the spatial smoothness is enforced or, alternatively, support is aggregated. Specifically, it is essential that support for a point comes from the same object, i.e. appropriate side of a 3-D boundary, as depicted in Figure 1.5.

Simple local methods, as described in Section 1.3.2, perform unsatisfactorily near depth discontinuities, because the region of support is central and its size is fixed. Thus, when

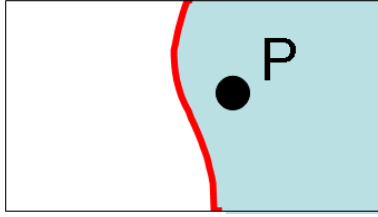


Figure 1.5: Smoothing Near 3-D Boundary (3-D Discontinuity Marked in Dark Grey). Ideally, support aggregation for point P must come from the shaded region only.

used near a depth discontinuity, the window will cross the 3-D boundary and points at different depths will be used to estimate the disparity of a central point – the smoothness assumption within a window is violated. The results near object boundaries worsen when window size grows, which contradicts the desired behaviour in regions with little texture, where greater support is necessary. In response, researchers have developed various techniques for adapting windows during the matching so they are unlikely to cross object boundaries [62, 46, 82, 54, 91, 120, 121] and these techniques prove to perform reasonably well.

Interestingly, cooperative methods also can be hampered by poor recovery of 3-D boundaries, as the initial match estimates are obtained by local window-based methods. Errors may not be so huge, because windows of relatively small size are sufficient for initial disparity estimation.

In contrast, global match methods can be quite keen near 3-D boundaries, if the smoothness cost function $E_{smooth}(d_p, d_q)$ is chosen appropriately. A very good choice is a robust non-convex cost function [15], which penalizes large and small jump discontinuities equally. Convex cost functions may result in a very convenient energy formulation (1.2) which can be optimized efficiently and *exactly*, e.g. with Graph Cuts [58, 119]. Unfortunately this exact solution replicates the real situation quite poorly resulting in ramping of object borders (i.e. natural sharp gradients are “smoothed” by slants of smaller gradient). A simple Pott’s model for smoothness penalty

$$E_{smooth}(d_p, d_q) = \begin{cases} 0 & \text{if } |d_p - d_q| = 0, \\ 1 & \text{otherwise} \end{cases} \quad (1.4)$$

yields particularly good solutions and is used in the majority of global algorithms today [20, 69, 116, 16, 41, 35, 122]. Intuitively, global algorithms have a natural advantage in that they intrinsically construct support on the fly, i.e. not committing to any predefined window shapes. Thus, they have a natural ability to recover borders of arbitrary shape and they have empirically proved to achieve this result [102, 21, 3].

Finally, coarse-to-fine stereo seems to be the worst performer near 3-D boundaries, which has been empirically shown in many papers [74, 109]. Fine spatial details are lost

at coarse scale due to low-pass filtering and coarse disparity estimates do not bear enough information for exact localization of object boundaries during the refinement procedure.

In order to localize depth discontinuities even better, a colour segmentation cue can be used by both local and global algorithms. The former can adapt their match window shape and size according to colour segments [131, 85, 92, 128], while the latter can enforce higher smoothness violation penalties for neighbouring pixels having similar colour values [111, 116, 122]. Moreover, images can be pre-segmented initially and the matching performed directly on segments [115, 5, 35].

Half-occlusions

Early work on computational stereo ignored half-occlusion or treated it as noise in the matching process [10]. Subsequently, a number of approaches to dealing with half-occlusions have emerged (see [39, 21] for reviews and empirical comparison). Several more recent contributions to the literature of half-occlusions can be noted. The use of adaptive spatial support for match windows can ameliorate issues arising in attempts to match half-occluded regions by shaping windows to avoid poorly defined matches [62, 46, 91, 54, 121]. Other recent additions to the literature are based on the expected behavior of disparity gradient in the vicinity of half-occlusions [58, 50, 110], e.g., the fact that occlusions in a left-based disparity map correspond to occluded regions in the right-based disparity map and vice versa (the occlusion constraint). The importance of disparity gradient as a constraint on allowable stereo matches has been known for some time (e.g., [23, 94]); however, its specific interpretation in terms of half-occlusion is relatively recent. Yet another approach rejects matches that are ambiguous (in having rival candidates of similar cost) to diagnose occlusion [100]. Occlusion detection also has been bolstered by constraining occlusion boundaries to align with those of uniform colour segments [35]. Another recent addition to the literature involves interleaved processes of layered disparity estimation and assignment to layers, with the option of pixel assignment to no layer, so that half-occlusions are dealt with as assignment outliers [77]. Interleaved calculation of correspondences and occlusions also has been cast within an expectation maximization framework [34], with high cost matches rejected as arising from occlusion [108].

In terms of empirical performance, some of the most impressive recent results have been demonstrated in conjunction with global methods [69, 90, 16, 110, 35], as they allow for better recovery of initial disparity. In contrast, empirical investigation of half-occlusion detection with local processing underlines shortcomings [39]. Moreover, occlusion handling in a coarse-to-fine framework has not been explicitly studied previously.

It is worthwhile mentioning that half-occlusions are very important in practice since they always arise near 3-D boundaries; thus, correct processing of half-occlusions automatically means better treatment for 3-D boundaries.

1.3.7 Speed-accuracy tradeoff

The brief overview of pros and cons of major dense stereo methods reveals the vivid tendency of global algorithms to be qualitatively superior to local ones. Indeed, global algorithms perform better in textureless regions and near 3-D boundaries due to their ability to construct arbitrary match support on the fly. Also, they surpass local algorithms in half-occlusion detection. On the contrary, no global algorithm can compete with local stereo procedures in terms of speed, storage requirements, and, finally, computational complexity. That is why when it comes to practical, especially real-time, depth measurements, researchers still rely on local and coarse-to-fine methods, as they require less processing and are easily parallelizable [21]. Thus, a speed-accuracy tradeoff is critical to computational stereo – it is possible to get good quality disparity maps at the expense of greater processing time, memory and amount of computation; faster and cheaper stereo gives significantly inferior results. The overwhelming majority of today’s research is concentrated on global algorithms [102, 21], but local and coarse-to-fine stereo still has its own niche in computer vision as being a simple and very fast alternative. Thus, improving the quality of the latter methods is vitally important for many practical applications.

1.4 Contributions

Strongly motivated by the practical applicability of binocular stereo, we have chosen coarse-to-fine block matching algorithms as a cornerstone, as such procedures inherently entail lower processing demand, map well to current hardware and software architectures and are suitable for parallel and pipeline computation. Moreover, the results of local and coarse-to-fine stereo demonstrate constant chronological improvement [62, 46, 54, 27, 93, 128]; some local procedures seem to even outperform many global algorithms when other cues like colour segmentation are used [128].

In the light of previous research and the motivation of the current research, the main contributions of this paper are as follows:

- A detailed analysis of errors near 3-D boundaries arising during coarse-to-fine block matching procedures is given and a simple yet very effective solution to significantly reduce these errors is proposed.
- A detailed analysis of computational half-occlusion detection is presented and a novel method for matching in the vicinity of such regions with respect to local stereo computation is described.
- Special attention is given to half-occlusion treatment in the coarse-to-fine framework, which allows for cooperative disparity and occlusion estimation

- The colour segmentation cue in area-based stereo is revisited and a novel formulation for local coarse-to-fine methods is proposed.
- All proposed advances have been implemented in C and combined in an integrated algorithm. The algorithm has been evaluated using standard datasets such as Middlebury College [3], CMU SRI [2], images from Brown University [1], as well as a set of naturalistic scenes acquired by MacDonald, Dettwiler & Associates Corporation (MDA, former MDRobotics). Qualitative and quantitative analysis show that incorporation of the proposed advances in coarse-to-fine block matching reduces disparity errors by a factor of two, while performing little extra computation, in comparison to previous local coarse-to-fine formulations. Moreover, the proposed algorithm is comparable to state-of-the-art solutions, while being more efficient and having very few parameters to tune.

1.5 Outline of report

This paper is subdivided into four chapters. Chapter 1 has motivated the research, provided background and stated the outstanding problems in stereo vision, some of which are attacked in this paper. Chapter 2 describes in detail the coarse-to-fine stereo framework and proposes improving modifications. It investigates half-occlusion phenomena and proposes a novel half-occlusion detection algorithm. This chapter also describes the colour-segmentation cue for stereo matching and formulates it for the investigated coarse-to-fine framework. Chapter 3 documents experimental evaluation of all proposed advances separately and as combined in a final, cumulative algorithm. Chapter 4 discusses in depth the features of the proposed coarse-to-fine computations. Chapter 5 summarizes our research results and outlines possible directions for future development.

Chapter 2

Technical approach

2.1 Block matching algorithm

The choice of match measure is a central issue in designing a correspondence algorithm, as it allows the quantitative evaluation of similarity between entities. In essence, it is one of the most critical parts for the area-based matching method, which boils down to comparison of two blocks of pixels.

Of the existing match metrics, in the following we will concentrate on sum of squared differences (SSD), sum of absolute differences (SAD), normalized cross-correlation (NCC) [21] and mutual information MI (Appendix A) for the following reasons. For formal developments in this chapter we emphasize SSD as it yields best to analysis. In empirical evaluation we will concentrate on the closely related SAD, which yields to efficient implementation and offers increased robustness to outliers. Significantly, because we will make use of bandpass images in matching, which remove intensity bias (more generally, DC signal component) such non-normalized match measures can perform well [7, 21]. For the sake of empirical comparison, we also will investigate NCC (to observe the effects of explicit normalization) and MI (to allow for matching in the presence of extreme violation of brightness constancy).

In the light of this discussion, we are ready to summarize the local block-based stereo algorithm that is to be used in later investigation. Mathematical encapsulation can be given as

$$\forall(x, y) : disp(x, y) = arg \min_{d_i} \sum_{(u,v) \in w(x,y)} \rho(im_1(u, v), im_2(u + d_i, v)) \quad (2.1)$$

where im_1 and im_2 are matching are reference images, (x, y) is the point in the reference image, w is the aggregation window around the point, and ρ is the cost function which is to be minimized. Specifically, $\rho(a, b) = |a - b|$ in case of SAD, $\rho(a, b) = (a - b)^2$ in case

of SSD, and $\rho \propto -cost_{ncc}$ when the NCC match measure is used (we put a minus sign in front, because NCC needs to be maximized).

Formula (2.1) can be converted into a pseudo-code algorithm:

Module A

```
disp(x,y) - disparity for pixel (x,y)
conf(x,y) - confidence for pixel (x,y)
  For each pixel (x,y) in the reference image
    For each d_i from disparity search range
      calculate cost(x,y,d_i) over
        central square window of size w
    End loop
    disp(x,y) = argmin(cost(x,y,d_i))
    conf(x,y) = cost(x,y,disp(x,y))
  End loop
End loop
```

2.2 Adaptive coarse-to-fine stereo for 3-D boundary preservation

As this paper mainly concerns coarse-to-fine stereo correspondence procedures, the abbreviation **CTF** will be used to denote it throughout the manuscript.

2.2.1 Basic algorithm

The basic elements of CTF block binocular matching can be outlined as follows (see [102, 21] and references therein). Initially both images are brought into image pyramid representations [24, 59] via repeated filtering to remove higher spatial frequency components, followed by commensurate subsampling. The disparity map is estimated for the coarsest level k , and then upsampled and scaled (implicitly or explicitly) to the next finer pyramid level $k - 1$ where it serves to provide an initial estimate for refined matching. The procedure continues until the finest resolution level $k = 0$ is reached and is portrayed by Figure 2.1. At each level disparity is estimated using any local stereo method, such as formulated by (2.1) and outlined as **Module A**. We can describe this procedure mathematically:

$$[(\forall(i)|1 \leq i \leq 2 : im_i^0 = im_i), (\forall(j)|1 \leq j \leq l_{max} : im_i^j = (g \otimes im_i^{j-1}) \downarrow_2)] , \quad (2.2)$$

$$[\forall(x, y) : disp^{l_{max}+1}(x, y) = 0] , \quad (2.3)$$

$$[\forall(k)|l_{max} > k > 0 : [\forall(x, y) : disp^k(x, y) = 2 \cdot disp^{k+1}(x, y) \uparrow_2 + \quad (2.4)$$

$$+ \arg \min_{d_i} \sum_{(u,v) \in w(x,y)} \rho(im_1^k(u, v), im_2^k(u + 2 \cdot disp^{k+1}(x, y) \uparrow_2 + d_i, v))]]$$

where im_1 and im_2 are stereo pair images, l_{max} is the number of pyramid levels, ρ is the match cost function, as in (2.1), g is the smoothing kernel and \otimes is the convolution operation, \downarrow_2 is the subsampling by the factor of two procedure and \uparrow_2 is the upsampling by the factor of two procedure. Note that term (2.2) describes the pyramid construction procedure, (2.3) states that initial coarsest disparity offset is initialized to all zeros, and (2.4) describes the actual CTF disparity estimation procedure.

In turn, this mathematical encapsulation can be summarized into the algorithm:

Module B

```
Reference and matching images are initially
brought into pyramid representation
disp(k,x,y) - disparity for pixel x, y on scale k
conf(k,x,y) - confidence for pixel x, y on scale k
```

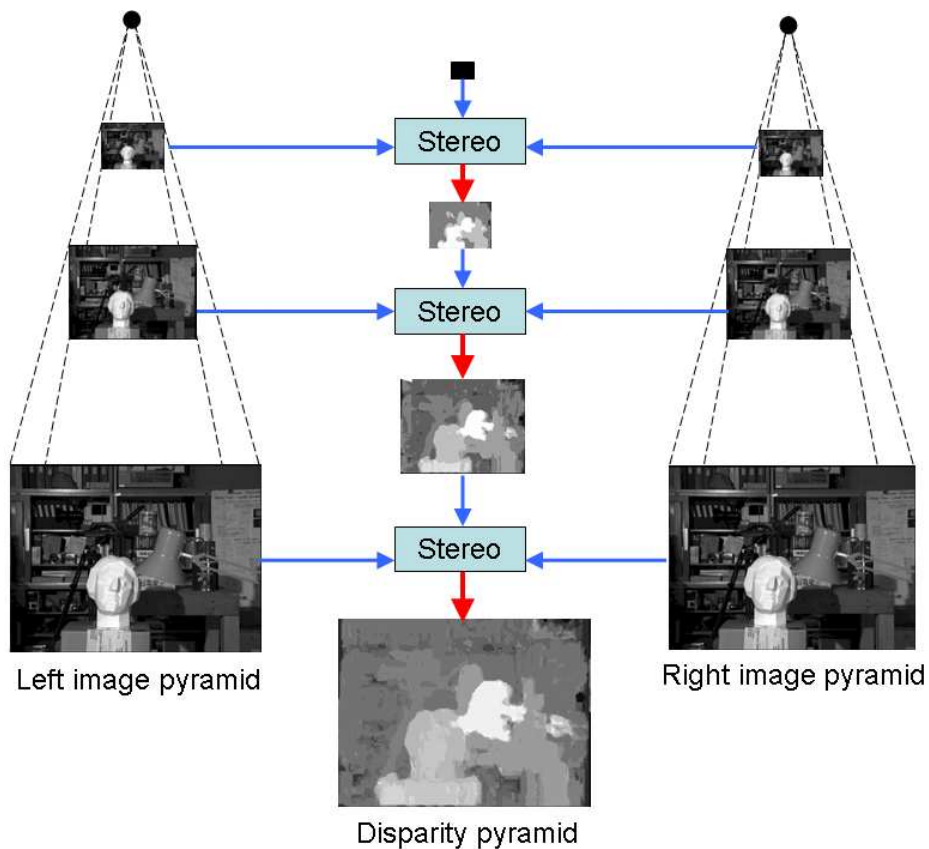


Figure 2.1: Coarse-to-Fine Disparity Estimation Procedure. Left and right images are initially brought into pyramid representations. Next, stereo correspondence for the coarsest level is computed. Initial coarse disparity map is zero everywhere. Using images from the next finer level and upsampled coarse disparity map, stereo correspondence is refined. This procedure is repeated until the base pyramid level (original image resolution) is reached.

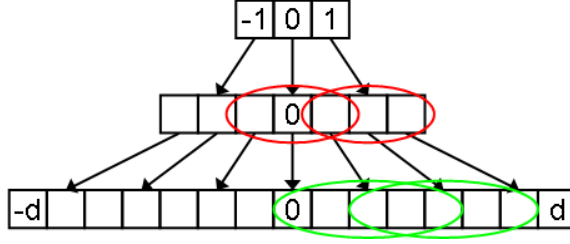


Figure 2.2: Coarse-to-Fine Disparity Search Space: Linear disparity search in single-scale matching vs. tree-like search in coarse-to-fine matching. Dark ovals symbolize the local search range of $\Delta x = \pm 1$; light ovals symbolize the local search range of $\Delta x = \pm 2$. Note the greater overlap in the search space, and hence greater computation redundancy, for greater local search range.

```

Initialize ref_disp(:, :) to all zeros
For each level k from level_max to 0
  For each pixel (k, x, y)
    Run Module A with search range
      [-delta_d+ref_disp(x, y), delta_d+ref_disp(x, y)];
  End loop
  ref_disp = 2*upsample(dis(k, :, :))
End loop

```

The outlined CTF processing has many useful characteristics. It helps to remove local minima in correspondence search by removal of small details at the coarse level. CTF also allows for variable support aggregation as support region of the same size (in terms of pixels) constitutes to larger smoothness at coarser levels. Large disparities in the high-resolution images correspond to small disparities in low-resolution subsampled images; hence large disparity search space is covered by minimal searches at higher pyramid levels, as in Figure 2.2. The last fact makes CTF very fast because the algorithm is essentially independent of the disparity search range. Specifically, if the complexity of the algorithm at a single level is $O(Nd)$, where N is the number of pixels and d is the disparity search range, the coarse-to-fine implementation has the complexity

$$O(NO(1)) + O\left(\frac{N}{4}O(1)\right) + O\left(\frac{N}{16}O(1)\right) + \dots < \sum_{i=0}^{\infty} \frac{O(N)O(1)}{4^i} = \frac{O(N)O(1)}{1 - 1/4} = O(N) \quad (2.5)$$

Taking into account that d can be on the order of a hundred for big images, the gain of coarse-to-fine in terms of speed may be crucial, especially when real-time performance is needed.

Considering the pyramids with resolution that halves going from level to level and local disparity search range $\Delta d \geq 1$, it is not hard to derive the relationship between Δd , the

number of levels l_{max} , and the maximum recoverable disparity d_{max} :

$$d_{max} = 2^{l_{max}+1}\Delta d - 1 \quad (2.6)$$

It is easy to notice from (2.6) that it is much more efficient to cover disparity search range by introducing more levels as (i) d_{max} increases exponentially, while Δd increases linearly, (ii) Δd can be kept small to reduce the amount of computation on each level and minimize the unnecessary overlap of intermediate search spaces while going from coarse to fine levels, which results in redundant computations (light and dark ovals in Figure 2.2).

Ultimately, the most computationally efficient configuration is to use a complete pyramid representation for maximum coverage of possible disparity values (the highest level is just a single row or column of pixels) and disparity search range being $\Delta d = 1$. In practice, such a configuration might not yield the best results and the combination of pyramid levels and search range should be found empirically. This has to do with the loss of distinctive patterns at higher levels leading to poorly constrained matching in terms of image structure. In practice, loss of spatial detail at coarse levels results in heavy deterioration of 3-D boundaries, inability to recover fine geometric structures and difficulty in recovering from errors made at the coarse level, because the algorithm is essentially greedy.

2.2.2 Analysis of coarse-to-fine stereo: Boundary deterioration

In this section we look at the process of boundary deterioration in a CTF SSD block-based algorithm that uses a Gaussian or Laplacian pyramid. SSD is chosen for its convenience in mathematical analysis, while Laplacian pyramids, which consist of bandpassed images, are generally more useful in practical stereo and motion estimation than Gaussian pyramids, (2.2) [7, 8]. This analysis will help us understand weak points of **Module B** and reveal what should be done to improve it. We start by establishing intuition. Next we give an analytic formulation. Finally, we provide numerical simulations.

Noise-free SSD matching

Intuitively, certain errors introduced by CTF arise when operating at coarse levels and estimating coarse disparities, i.e. when the images are low-passed and subsampled. Appropriate low-pass or band-pass filtering avoids aliasing caused by subsampling procedure. Typically the filtering is realized via a Gaussian kernel as it is causal in scale space [78] and yields an efficient implementation. At the same time, depth discontinuities will be blurred, which means that pixels on 3-D boundaries will be a mixture of foreground and background surfaces. The actual proportion of the mixture will depend on the shape of the 3-D discontinuity or, rather, the ratio of the surfaces' areas covered by the convolution window.

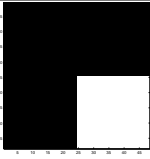
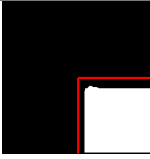

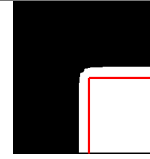
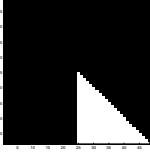
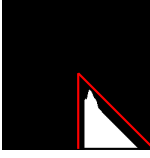
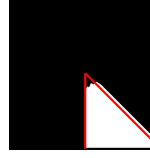
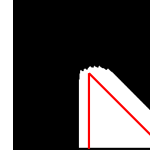
Border config	$\log_2 FBR = -5$	$\log_2 FBR = 0$	$\log_2 FBR = 5$
			
			

Figure 2.3: Examples of 3-D boundary Deterioration in Conventional Coarse-to-Fine Stereo. Boundary deterioration examples for 90° and 45° configurations, top and bottom rows, respectively. In the first column, black denotes background texture, white denotes foreground texture. In the last three columns, sketches show the recovered disparity (foreground-background) for three different $\log FBR$ when three pyramid levels are used, where $FBR = \sigma_f^2/\sigma_b^2$ is a foreground-background ratio defined as the ratio of pixel values variances for foreground and background surfaces. Aggregation window is 5×5 in these examples.

For detailed illustration, we consider two particular cases: a common rectangular-shape boundary (90°) and a harder sharp-corner boundary (45°)¹. Both configurations are shown in Figure 2.3, first column. For simplicity of analysis we assume that the scenes are noise-free, fronto-parallel and textured. These assumptions make a simple SSD measure return 0 in case of correct structure alignment and some other positive numbers for incorrect assignment (except accidental repetitive texture).

To make matters more precise, let arbitrary image texture patterns be characterized in terms of intensity mean, μ , and variance, σ^2 , so that foreground and background surface patterns are parameterized by μ_f, σ_f^2 and μ_b, σ_b^2 , respectively.

The SSD score over an aggregation window can be computed by convolving the squared difference map between reference I_{ref} and disparity shifted match image I_{other} with the kernel W , which corresponds to the shape of window (usually $w \times w$ matrix with all entries being 1):

$$ssd_d = W \otimes (I_{ref} - shift_d(I_{other}))^2 \quad (2.7)$$

$$[\forall(x, y) : shift_d(I(x + d, y)) = I(x, y)] \quad (2.8)$$

where I is an image.

¹Intermediate angle cases are harder to analyze due to the digital nature of the images.

Considering the necessary processing operations, the application of the same operation on an arbitrary pyramid level k can be described similarly as

$$ssd_d = W \otimes [I_{ref}^k - shift_d(I_{other}^k)]^2 \quad (2.9)$$

where I^k denotes an image at pyramid level k .

Let

$$g = \frac{1}{16} [1 \ 4 \ 6 \ 4 \ 1]^T \quad (2.10)$$

be the binomial approximation of a one-dimensional Gaussian with unit variance. I^k is defined recursively in terms of

$$I^{\mathcal{G}(k)} = (gg^T \otimes I^{\mathcal{G}(k-1)}) \downarrow_2 \quad (2.11)$$

and

$$I^{\mathcal{L}(k)} = I^{\mathcal{G}(k)} - 4gg^T \otimes (I^{\mathcal{G}(k+1)}) \uparrow_2 \quad (2.12)$$

with $I^{\mathcal{G}(k)}$ and $I^{\mathcal{L}(k)}$ being k^{th} levels of Gaussian and Laplacian pyramids ($I^{\mathcal{G}(0)} = I$), respectively. Additionally, \downarrow_2 and \uparrow_2 denote factor of two down- and up-sampling, respectively².

The application of the Laplacian operator will result in each pixel near a 3-D boundary being a blend of foreground and background textures, and the mixture proportion will be determined by the spatial position of the pixel with respect to the depth discontinuity. For example, the proportion of the foreground for a pixel can be calculated as the sum of the values in the Laplacian kernel which cover the foreground surface when the kernel is positioned at the centre of the pixel. The proportion of the background is calculated analogously. Thus, if a random variable p represents some pixel's intensity, then its intensity value after the smoothing would be

$$p = \sum_{i \in f} w_i p_{fi} + \sum_{j \in b} w_j p_{bj} \quad (2.13)$$

where w_i are the kernel coefficients, p_{fi} and p_{bi} are samples drawn from foreground and background intensity distributions, which are assumed to be Gaussians for simplicity.

²The factor of 4 in the Laplacian pyramid specification is needed as 3/4 of the samples in the upsampled image are newly inserted zeros, as in this formulation upsampling is accomplished by inserting new rows and columns of zeros between all original rows and columns.

For the subsequent discussion we introduce the following notation:

$$\sum_{i \in f} w_i = f_1, \quad (2.14)$$

$$\sum_{j \in b} w_j = b_1 = -f_1, \quad (2.15)$$

$$\sum_{i \in f} w_i^2 = f_2, \quad (2.16)$$

$$\sum_{j \in b} w_j^2 = b_2, \quad (2.17)$$

where f and b denote the foreground and background patches, respectively. Note that $f_1 + b_1 = 0$, because we have only two surfaces and the sum of elements in the Laplacian kernel is always 0. Analogously, $f_2 + b_2 = \zeta$, where ζ is the constant equal to the sum of squares of the kernel values.

Finally, it is useful to derive the first and the second moments for the pixel intensity distributions p . In doing so, we assume that p_{fi} is independent of p_{fk} and p_{bi} is independent of p_{bk} when $i \neq k$; as before, p_{fi} is independent of p_{bk} for all i and k . Under these assumptions, the first moment evaluates as

$$\begin{aligned} E[p] &= E \left[\sum_{i \in f} w_i p_{fi} + \sum_{j \in b} w_j p_{bj} \right] \\ &= E \left[\sum_{i \in f} w_i p_{fi} \right] + E \left[\sum_{j \in b} w_j p_{bj} \right] \\ &= \sum_{i \in f} w_i E[p_{fi}] + \sum_{j \in b} w_j E[p_{bj}] = \mu_f \sum_{i \in f} w_i + \mu_b \sum_{j \in b} w_j \\ &= f_1 \mu_1 + b_1 \mu_b \end{aligned} \quad (2.18)$$

Similarly, the second moment evaluates as

$$E[p^2] = E \left[\left(\sum_{i \in f} w_i p_{fi} + \sum_{j \in b} w_j p_{bj} \right)^2 \right]$$

here, it is useful to separate terms depending on foreground/background interactions to

yield

$$\begin{aligned}
&= E \left[\sum_{i \in f} w_i^2 p_{fi}^2 + \sum_{j \in b} w_j^2 p_{bj}^2 + \sum_{i \neq k} w_i w_k p_{fi} p_{fk} \right. \\
&\quad \left. + \sum_{j \neq k} w_j w_k p_{bj} p_{bk} + \sum_{i \neq j} w_i w_j p_{fi} p_{bj} \right] \\
&= \sum_{i \in f} w_i^2 E [p_{fi}^2] + \sum_{j \in b} w_j^2 E [p_{bj}^2] + \sum_{i \neq k} w_i w_k E [p_{fi}] E [p_{fk}] \\
&\quad + \sum_{j \neq k} w_j w_k E [p_{bj}] E [p_{bk}] + \sum_{i \neq j} w_i w_j E [p_{fi}] E [p_{bj}] + \sum_{i \neq j} w_j w_i E [p_{bj}] E [p_{fi}]
\end{aligned}$$

appropriate definition of the expectation operation then yields

$$\begin{aligned}
&= \sum_{i \in f} w_i^2 (\sigma_f^2 + \mu_f^2) + \sum_{j \in b} w_j^2 (\sigma_b^2 + \mu_b^2) \\
&\quad + \sum_{i \neq k} w_i w_k \mu_f^2 + \sum_{j \neq k} w_j w_k \mu_b^2 + 2 \sum_{i \neq j} w_i w_j \mu_f \mu_b
\end{aligned}$$

considering the notations (2.14)-(2.17) introduced earlier and the identities $\sum_i w_i \sum_i w_i = \sum_i w_i^2 + \sum_{i \neq k} w_i w_k$ and $\sum_{i,j} w_i w_j = \sum_i w_i \sum_j w_j'$, we get

$$\begin{aligned}
&= f_2 (\sigma_f^2 + \mu_f^2) + b_2 (\sigma_b^2 + \mu_b^2) + \mu_f^2 \left(\left(\sum_{i \in f} w_i \right)^2 - \sum_{i \in f} w_i^2 \right) \\
&\quad + \mu_b^2 \left(\left(\sum_{j \in b} w_j \right)^2 - \sum_{j \in b} w_j^2 \right) + 2 \mu_f \mu_b \sum_{i \in f} w_i \sum_{j \in b} w_j \\
&= f_2 (\sigma_f^2 + \mu_f^2) + b_2 (\sigma_b^2 + \mu_b^2) + \mu_f^2 (f_1^2 - f_2) + \mu_b^2 (b_1^2 - b_2) + 2 \mu_f \mu_b f_1 b_1 \\
&= f_2 \sigma_f^2 + b_2 \sigma_b^2 + (f_1 \mu_f + b_1 \mu_b)^2
\end{aligned}$$

Thus, the second moment is computed as

$$E [p^2] = f_2 \sigma_f^2 + b_2 \sigma_b^2 + (f_1 \mu_f + b_1 \mu_b)^2 \quad (2.19)$$

The SSD score for an individual pixel is calculated as

$$E [(p - q)^2] = E [p^2] + E [q^2] - 2E [pq] \quad (2.20)$$

where p and q are pixels' intensities in reference and other images; p is defined as in (2.13) and q is defined analogously.

Equation (2.20) gives a closed-form expression to calculate cost for an arbitrary disparity assignment, once the cross-term $E[pq]$ is elaborated. We can distinguish 3 different types of assignments: fixation on foreground (SSD_f), fixation on background (SSD_b) and all other assignments (SSD_o).

SSD_f : Foreground structure component is aligned ($p_f = q_f$, $f_{1p} = f_{1q}$, $f_{2p} = f_{2q}$, $b_{1p} = b_{1q}$, $b_{2p} = b_{2q}$), but background structure is not; thus, p_b , q_b and are independent of each other. Hence, the cross-term is elaborated as

$$\begin{aligned}
E[pq] &= E \left[\left(\sum_{i \in f} w_i p_{fi} + \sum_{j \in b} w_j p_{bj} \right) \left(\sum_{i \in f} w_i p_{fi} + \sum_{j \in b} w_j q_{bj} \right) \right] \quad (2.21) \\
&= E \left[\left(\sum_{i \in f} w_i p_{fi} \right)^2 \right] + E \left[\sum_{i \in f} w_i p_{fi} \sum_{j \in b} w_j q_{bj} \right] \\
&\quad + E \left[\sum_{i \in f} w_i p_{fi} \sum_{j \in b} w_j p_{bj} \right] + E \left[\sum_{j \in b} w_j p_{bj} \sum_{j \in b} w_j q_{bj} \right] \\
&= E \left[\sum_{i \in f} w_i^2 p_{fi}^2 + \sum_{i \neq k \in f} w_i p_{fi} w_k p_{fk} \right] + E \left[\sum_{i \in f} w_i p_{fi} \sum_{j \in b} w_j q_{bj} \right] \\
&\quad + E \left[\sum_{i \in f} w_i p_{fi} \sum_{j \in b} w_j p_{bj} \right] + E \left[\sum_{j \in b} w_j p_{bj} \sum_{j \in b} w_j q_{bj} \right]
\end{aligned}$$

using our independence assumptions (p_b , q_b and p_f are independent of each other)

$$\begin{aligned}
&= \sum_{i \in f} w_i^2 E[p_{fi}^2] + \sum_{i \neq k \in f} w_i E[p_{fi}] w_k E[p_{fk}] + \sum_{i \in f} w_i E[p_{fi}] \sum_{j \in b} w_j E[q_{bj}] \\
&\quad + \sum_{i \in f} w_i E[p_{fi}] \sum_{j \in b} w_j E[p_{bj}] + \sum_{j \in b} w_j E[p_{bj}] \sum_{j \in b} w_j E[q_{bj}]
\end{aligned}$$

using the identity $\sum_i w_i \sum_i w_i = \sum_i w_i^2 + \sum_{i \neq k} w_i w_k$ and the previously introduced independence assumptions we get

$$\begin{aligned}
&= E[p_f^2] \sum_{i \in f} w_i^2 + E[p_f] E[p_f] \left(\left(\sum_{i \in f} w_i \right)^2 - \sum_{i \in f} w_i^2 \right) \\
&\quad + E[p_f] E[q_b] \sum_{i \in f} w_i \sum_{j \in b} w_j + E[p_f] E[p_b] \sum_{i \in f} w_i \sum_{j \in b} w_j \\
&\quad + E[p_b] E[q_b] \sum_{j \in b} w_j \sum_{j \in b} w_j
\end{aligned}$$

application of (2.14)-(2.17) and (2.19) and definition of expectation operation yield

$$\begin{aligned}
&= f_2\sigma_f^2 + f_2\mu_f^2 + f_1^2\mu_f^2 - f_2\mu_f^2 + f_1b_1\mu_f\mu_b + f_1b_1\mu_f\mu_b + b_1^2\mu_b^2 \\
&= f_2\sigma_f^2 + (f_1\mu_f + b_1\mu_b)^2
\end{aligned}$$

For the actual SSD score we combine the results of (2.19) and (2.21) to find

$$\begin{aligned}
SSD_f &= E[p^2] + E[q^2] - 2E[pq] \\
&= f_2\sigma_f^2 + b_2\sigma_b^2 + (f_1\mu_f + b_1\mu_b)^2 + f_2\sigma_f^2 + b_2\sigma_b^2 \\
&\quad + (f_1\mu_f + b_1\mu_b)^2 - 2f_2\sigma_f^2 - 2(f_1\mu_f + b_1\mu_b)^2 \\
&= 2b_2\sigma_b^2
\end{aligned} \tag{2.22}$$

SSD_b : In this case, only the background component is aligned, i.e. $p_b = q_b$ and p_b, p_f and q_f are mutually independent. Thus, the cross-term becomes

$$\begin{aligned}
E[pq] &= E\left[\left(\sum_{i \in f} w_i p_{fi} + \sum_{j \in b} w_j p_{bj}\right) \left(\sum_{i \in f} w_i q_{fi} + \sum_{j \in b} w_j p_{bj}\right)\right] \\
&= E\left[\sum_{i \in f} w_i p_{fi} \sum_{i \in f} w_i q_{fi}\right] + E\left[\sum_{i \in f} w_i p_{fi} \sum_{j \in b} w_j p_{bj}\right] \\
&\quad + E\left[\sum_{i \in f} w_i q_{fi} \sum_{j \in b} w_j p_{bj}\right] + E\left[\sum_{j \in b} w_{pj} p_{bj} \sum_{j \in b} w_{qj} p_{bj}\right]
\end{aligned} \tag{2.23}$$

again, under our independence assumptions (introduced earlier)

$$\begin{aligned}
&= \sum_{i \in f} w_i E[p_{fi}] \sum_{i \in f} w_i E[q_{fi}] + \sum_{i \in f} w_i E[p_{fi}] \sum_{j \in b} w_j E[p_{bj}] \\
&\quad + \sum_{i \in f} w_i E[q_{fi}] \sum_{j \in b} w_j E[p_{bj}] + E\left[\sum_{j \in b} w_{pj} p_{bj} \sum_{j \in b} w_{qj} p_{bj}\right]
\end{aligned}$$

considering the notations (2.14)-(2.17) and splitting the last term

$$\begin{aligned}
&= f_{1p}f_{1q}\mu_f^2 + f_{1p}b_{1q}\mu_f\mu_b + b_{1p}f_{1q}\mu_b\mu_f \\
&\quad + E\left[\sum_{j \in b} w_{pj}w_{qj}p_{bj}p_{bj} + \sum_{j \neq k \in b} w_{pj}w_{qk}p_{bj}p_{bk}\right]
\end{aligned}$$

applying identity $\sum_j w_{pj} \sum_j w_{qj} = \sum_{j \neq k} w_{pj}w_{qk} + \sum_j w_{pj}w_{qj}$

$$\begin{aligned}
&= f_{1p}f_{1q}\mu_f^2 + f_{1p}b_{1q}\mu_f\mu_b + b_{1p}f_{1q}\mu_b\mu_f \\
&\quad + \sum_{j \in b} w_{pj}w_{qj}E[p_b^2] + E[p_b]E[p_b] \left(\sum_{j \in b} w_{pj} \sum_{j \in b} w_{qk} - \sum_{j \in b} w_{pj}w_{qj}\right)
\end{aligned}$$

applying (2.14)-(2.17) and $\sum_{j \in b} w_{pj}w_{qj} = \min(b_{2p}, b_{2q})$ (explained further in the text below)

$$\begin{aligned}
&= f_{1p}f_{1q}\mu_f^2 + f_{1p}b_{1q}\mu_f\mu_b + b_{1p}f_{1q}\mu_b\mu_f \\
&\quad + \min(b_{2p}, b_{2q}) (\sigma_b^2 + \mu_b^2) + (b_{1p}b_{1q} - \min(b_{2p}, b_{2q})) \mu_b^2 \\
&= f_{1p}f_{1q}\mu_f^2 + f_{1p}b_{1q}\mu_f\mu_b + b_{1p}f_{1q}\mu_b\mu_f \\
&\quad + \min(b_{2p}, b_{2q})\sigma_b^2 + b_{1p}b_{1q}\mu_b^2 \\
&= \min(b_{2p}, b_{2q})\sigma_b^2 + (f_{1p}\mu_f + b_{1p}\mu_b)(f_{1q}\mu_f + b_{1q}\mu_b)
\end{aligned}$$

Special attention should be paid to the step $\sum_{j \in b_p} w_{pj}w_{qj} = \min(b_{2p}, b_{2q})$. By the above assumption for SSD_b , the background surface components in two views should be identical, which makes $p_b = q_b$ and $w_{pj} = w_{qj}$. However, because foreground components are not aligned, part of the background in either of the images will be occluded, as illustrated in Figure 2.4. This phenomenon of so-called half-occlusion is discussed in detail in Section 2.3. Here, the implication is that some of w_{pj} terms will not have corresponding w_{qj} , or vice versa.

$$\begin{aligned}
\sum_{j \in b} w_{pj}w_{qj} &= \begin{cases} \sum_{j \in b} w_{pj}w_{pj}, & \text{if p is occluded} \\ \sum_{j \in b} w_{qj}w_{qj}, & \text{if q is occluded} \end{cases} \\
&= \begin{cases} b_{2p}, & \text{if p is occluded} \\ b_{2q}, & \text{if q is occluded} \end{cases} \\
&= \min(b_{2p}, b_{2q})
\end{aligned} \tag{2.24}$$

because $b_{2p} < b_{2q}$ when p is occluded (the summation b_{2q} has more positive terms) and, analogously, $b_{2q} < b_{2p}$ when q is occluded.

Hence, the corresponding SSD score is

$$\begin{aligned}
SSD_b &= E[p^2] + E[q^2] - 2E[pq] \\
&= f_{2p}\sigma_f^2 + b_{2p}\sigma_b^2 + (f_{1p}\mu_f + b_{1p}\mu_b)^2 \\
&\quad + f_{2q}\sigma_f^2 + b_{2q}\sigma_b^2 + (f_{1q}\mu_f + b_{1q}\mu_b)^2 \\
&\quad - 2\min(b_{2p}, b_{2q})\sigma_b^2 - 2(f_{1p}\mu_f + b_{1p}\mu_b)(f_{1q}\mu_f + b_{1q}\mu_b) \\
&= (f_{2p} + f_{2q})\sigma_f^2 + (b_{2p} + b_{2q} - 2\min(b_{2p}, b_{2q}))\sigma_b^2 + (f_{1p}\mu_f + b_{1p}\mu_b)^2 \\
&\quad + (f_{1q}\mu_f + b_{1q}\mu_b)^2 - 2(f_{1p}\mu_f + b_{1p}\mu_b)(f_{1q}\mu_f + b_{1q}\mu_b)
\end{aligned} \tag{2.25}$$

applying identity $a + b - 2\min(a, b) = \begin{cases} b - a, & \text{if } a < b \\ a - b, & \text{if } a \geq b \end{cases} = |a - b|$

$$\begin{aligned}
&= (f_{2p} + f_{2q})\sigma_f^2 + |b_{2p} - b_{2q}|\sigma_b^2 + (f_{1p}\mu_f + b_{1p}\mu_b)^2 + (f_{1q}\mu_f + b_{1q}\mu_b)^2 \\
&\quad - 2(f_{1p}\mu_f + b_{1p}\mu_b)(f_{1q}\mu_f + b_{1q}\mu_b) \\
&= (f_{2p} + f_{2q})\sigma_f^2 + |b_{2p} - b_{2q}|\sigma_b^2 + ((f_{1p}\mu_f + b_{1p}\mu_b) - (f_{1q}\mu_f + b_{1q}\mu_b))^2
\end{aligned}$$

p				
w_1	w_6	w_{11}	w_{16}	w_{21}
w_2	w_7	w_{12}	w_{17}	w_{22}
w_3	w_8	w_{13}	0	0
w_4	w_9	w_{14}	0	0
w_5	w_{10}	w_{15}	0	0

q				
w_1	w_6	w_{11}	w_{16}	w_{21}
w_2	w_7	w_{12}	w_{17}	w_{22}
w_3	w_8	w_{13}	w_{18}	w_{23}
w_4	w_9	w_{14}	w_{19}	w_{24}
w_5	w_{10}	w_{15}	w_{20}	w_{25}

Figure 2.4: Calculating Match Score for the Background Surface: Some Points are Occluded. Low- or band-pass kernel in the reference image is applied to p which is close to foreground region (shaded region with $w_i = 0$, left image). The same kernel is applied to the corresponding point q in the other image. Note that some of the background points in the vicinity of q do not have a match (shaded region, right image), as they are occluded by the foreground surface in the other image (shaded region, left image).

remembering that $f_1 + b_1 = \xi$, where $\xi = 0$ and $\xi = 1$ for the Laplacian and Gaussian kernels, respectively

$$\begin{aligned}
&= (f_{2p} + f_{2q})\sigma_f^2 + |b_{2p} - b_{2q}|\sigma_b^2 + (f_{1p}\mu_f + \xi\mu_b - f_{1p}\mu_b - f_{1q}\mu_f - \xi\mu_b + f_{1q}\mu_b)^2 \\
&= (f_{2p} + f_{2q})\sigma_f^2 + |b_{2p} - b_{2q}|\sigma_b^2 + (f_{1p}(\mu_f - \mu_b) - f_{1q}(\mu_f - \mu_b))^2 \\
&= (f_{2p} + f_{2q})\sigma_f^2 + |b_{2p} - b_{2q}|\sigma_b^2 + (f_{1p} - f_{1q})^2(\mu_f - \mu_b)^2
\end{aligned}$$

SSD_o : All parameters are different, as nothing is in alignment; hence, p_b , p_f , q_b and q_f are mutually independent. The cross term is simple to calculate as p and q become independent:

$$E[pq] = E[p] E[q] = (f_{1p}\mu_f + b_{1p}\mu_b)(f_{1q}\mu_f + b_{1q}\mu_b) \quad (2.26)$$

and the final SSD score is calculated as

$$\begin{aligned}
SSD_o &= E[p^2] + E[q^2] - 2E[pq] \\
&= f_{2p}\sigma_f^2 + b_{2p}\sigma_b^2 + (f_{1p}\mu_f + b_{1p}\mu_b)^2 \\
&\quad + f_{2q}\sigma_f^2 + b_{2q}\sigma_b^2 + (f_{1q}\mu_f + b_{1q}\mu_b)^2 \\
&\quad - 2(f_{1p}\mu_f + b_{1p}\mu_b)(f_{1q}\mu_f + b_{1q}\mu_b) \\
&= (f_{2p} + f_{2q})\sigma_f^2 + (b_{2p} + b_{2q})\sigma_b^2 \\
&\quad + ((f_{1p}\mu_f + b_{1p}\mu_b) - (f_{1q}\mu_f + b_{1q}\mu_b))^2
\end{aligned} \quad (2.27)$$

again, considering $f_1 + b_1 = \xi$, where $\xi = 0$ and $\xi = 1$ for the Laplacian and Gaussian kernels, respectively

$$\begin{aligned}
&= (f_{2p} + f_{2q})\sigma_f^2 + (b_{2p} + b_{2q})\sigma_b^2 \\
&\quad + (f_{1p}\mu_f + \xi\mu_b - f_{1p}\mu_b - f_{1q}\mu_f - \xi\mu_b + f_{1q}\mu_b) \\
&= (f_{2p} + f_{2q})\sigma_f^2 + (b_{2p} + b_{2q})\sigma_b^2 + (f_{1p} - f_{1q})^2(\mu_f - \mu_b)^2
\end{aligned}$$

In (2.25) and (2.27), observe the term $(f_{1p} - f_{1q})^2(\mu_f - \mu_b)^2$, which can be called “contrast term”, as it depends on the intensity difference between two surfaces. To simplify the following analysis, we fix this contrast term by assuming that foreground and background textures have the same means (i.e. $\mu_f - \mu_b = 0$), which effectively make the contrast term zero.

Finally, we divide each of (2.22), (2.25) and (2.27) by σ_b^2 (under the assumption that $\sigma_b^2 > 0$), and introduce a foreground to background ratio $FBR = \sigma_f^2/\sigma_b^2$, as it captures the necessary information about the properties of foreground and background surfaces. The derived expected values for SSD_f (2.22), SSD_b (2.25) and SSD_o (2.27) can be used to evaluate the aggregated SSD match measure (2.7) in CTF disparity estimation (2.2)-(2.4) using either Gaussian or Laplacian pyramids. The results yields an analytic formulation that calculates the cost for any given disparity assignment at a foreground/background surface discontinuity. For present concerns, the critical parameters are discontinuity geometry (line, corner, etc.), disparity jump, which is tightly coupled to number of pyramid levels, and ratio of foreground/background texture variance (FBR). Unfortunately, the complexity of the overall formulation has prevented discovery of a closed-form solution for the cost minimizing disparity. In response, we have studied the solution space via numerical simulations. Prior to presenting the results, three additional points are noted, as follows.

First, it is important to note that SSD_o is never less than either of SSD_f or SSD_b , at least in the noise free case, because no structure covered by the aggregation window is aligned in this case. Thus, disparity assignment for each point tends to get attached either to the foreground or background surface. This observation has been confirmed by our computer simulation, during which SSD_o was never less than SSD_f and SSD_b for all pixels under all tested configurations.

Second, considering CTF search space as in Figure 2.2, it is seen that assignment of each fine disparity has a certain path from coarse to fine levels – from top to bottom of the “tree” many values start with the same route and branch off at finer resolution. Hence, there is a certain resolution level when the foreground and the background disparities would first be distinguished (paths are split), and a pixel will get the disparity assignment of either foreground, or background surface. The next resolution level will use either of these assignments as an offset, but also a small search range that will *not* cover the other assignment. That means that the calculations on the finer level can only get a better SSD estimate for the already committed disparity of foreground or background surfaces,

because all other disparities ought to have bigger SSD values. In other words, if, while doing coarse-to-fine estimations, we keep fetching a single coarse disparity value from the previous level as an offset, we just carry on the disparity assignment made on the crucial coarse level, when foreground and background surfaces can be first distinguished. This observation greatly simplifies the CTF analysis by considering the disparity decision made at the crucial coarse level only. The subsequent refinement will just improve the SSD estimate and finer disparity value for the committed surface.

Third, we specify how the coarse disparity offset is determined for each pixel, i.e. the disparity upsampling procedure. Here we concentrate on the Nearest Neighbour as it is widely used in practice and greatly simplifies our analysis. The upsampling procedure itself is defined in the notation of (2.4) as the following:

$$disp^{k+1}(x, y) \uparrow_2 = disp^{k+1}(\lfloor x/2 \rfloor, \lfloor y/2 \rfloor) \quad (2.28)$$

Note that this procedure does not create disparity values in the upsampled disparity map, other than foreground and background disparity values (these are the only values that appear in the coarse disparity map, as discussed in the paragraph above).

As a set of numerical examples, Figure 2.5 shows the expected boundary recovery statistics for horizontal 3-D boundary, vertical 3-D boundary, rectangular-corner (90°) and sharp-corner (45°) cases for disparity estimated over 1, 2, 3 and 4 Laplacian pyramid levels (disparity jumps 1, 3, 7 and 15 respectively were used) under different values of FBR (we employ the \log_2 scale for FBR to capture greater range of values). As we are investigating local block-based matching under the assumption that points can be matched unambiguously, it will perform best near 3-D boundaries when the aggregation window is the smallest possible, as discussed in Section 1.3.6. Hence, the solution with unit window size, $w = 1$, can be treated as an upper bound on the best stereo performance when one can perfectly recover the disparity map on each level using a single coarse disparity offset interpolated by NN.

Based on this simulation, a number of conclusions can be highlighted.

- Boundary overreach for standard CTF block matching is a serious problem and it spreads at a rate that is exponential in the number of pyramid levels. That is illustrated by consistently higher overreach for coarser pyramid levels on Figure 2.5.
- Fine structures suffer the most – in general, 3-D discontinuities become smoothed while corners become rounded. That is clearly demonstrated by foreground shrinking statistics of the corners.
- Boundary overreach behaviour is fundamentally different for horizontal and vertical boundaries. The reason is that vertical 3-D boundaries are complemented by half-occluded regions in one of the images (Section 2.3 will discuss this phenomenon in

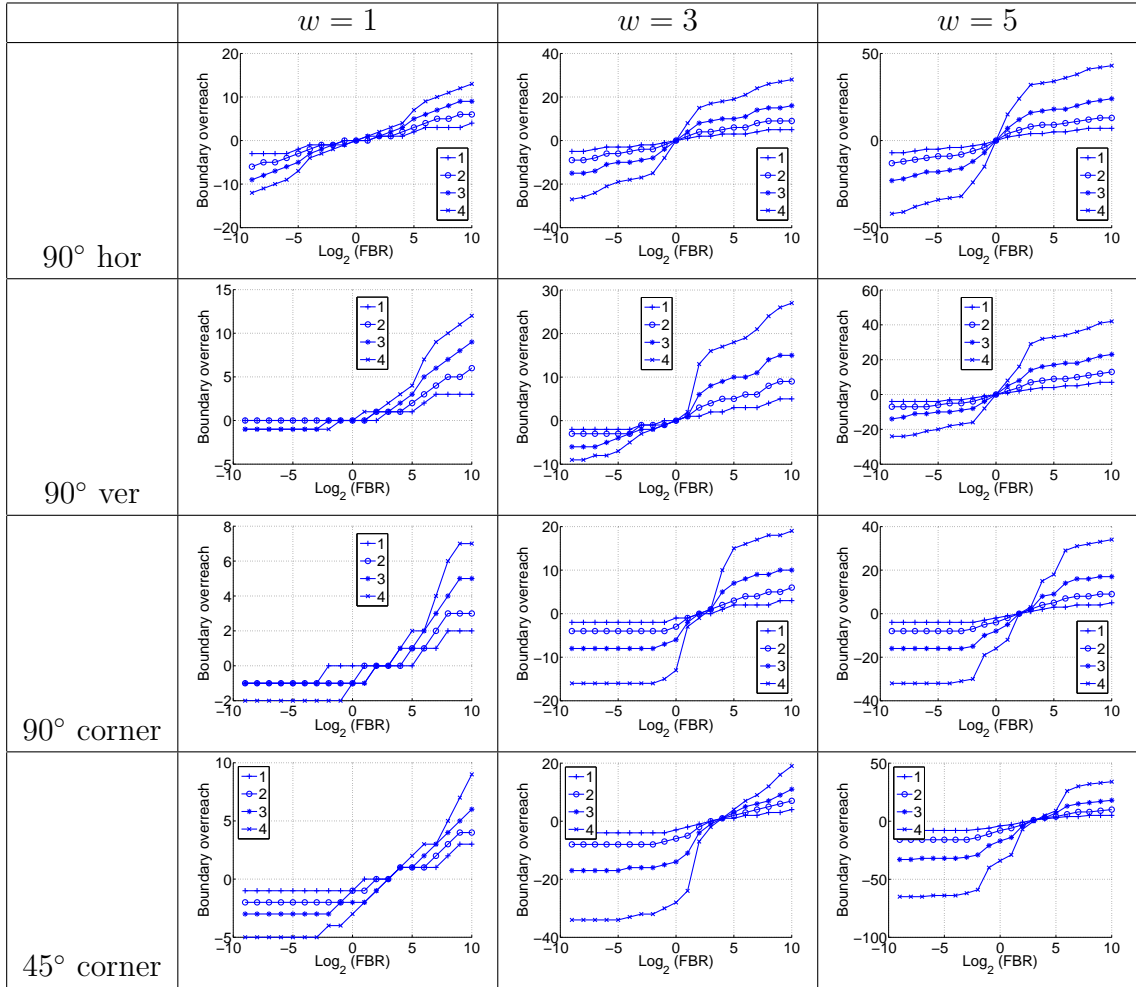


Figure 2.5: Statistics for 3-D Boundary Deterioration in Conventional Coarse-to-Fine Stereo: Simulation Results. Boundary deterioration statistics for corner pixel, pixel on horizontal and pixel on vertical boundaries (as depicted in the first column of 2.3). Negative boundary overreach values (units of pixels) denote foreground shrinking and positive values denote fattening respectively. Each curve in the sketch describes the overreach profile for a certain number of pyramid levels used in the disparity estimation. Aggregation window size is w pixels.

depth), while horizontal 3-D boundaries are not. The error profile with respect to foreground shrinking/fattening is symmetric for geometrically simpler horizontal 3-D boundaries.

- One of the causes for boundary overreach is the fixed aggregation window, as windows of greater size yield bigger errors. This foreground fattening/shrinking effect of rectangular windows has been analyzed before [54, 91].
- CTF processing yields boundary degradation at a rate higher than would be caused through use of analogous fixed size aggregation windows at a single fine level. In single scale, SSD aggregation (2.7) is performed by kernel W only. In CTF at level k , the extra smoothing (and subsampling) from the pyramid construction contributes, and the aggregation with W is performed on low-passed or band-passed images, as in (2.9). Greater implicit aggregation results in greater boundary overreach effect.
- Another cause for boundary deterioration is the use of single disparity offset in CTF projection, i.e. the upsampling procedure. That phenomenon is distilled by “ideal stereo” simulation, with window size $w = 1$, as it does not introduce any new errors in the disparity estimation process on each pyramid level, e.g. caused by fixed square windows. Thus, errors arise from the projection of coarse disparity estimates to finer levels.

Noisy SSD matching

The analytic framework developed in Section 2.2.2 can be trivially extended to include Gaussian white noise in either of the images for more realistic modeling. The introduction of noise will allow us to highlight a principle difference between stereo matchers that use Gaussian and Laplacian pyramids.

Assume that the reference image is corrupted by Gaussian noise with mean μ_n and variance σ_n that is independent of foreground and background samples and independent for all pixels in the image. Then, each pixel of the low- or band-passed image has the noise component being equal to

$$n = \sum_k w_k p_{nk} \quad (2.29)$$

In turn, SSD between two pixels p and q can be described as

$$\begin{aligned} & E [(p - q + n)^2] \quad (2.30) \\ &= E [(p - q)^2] + E [n^2] + E [n] E [p - q] \\ &= E [(p - q)^2] + E \left[\left(\sum_k w_k p_{nk} \right)^2 \right] + E \left[\sum_k w_k p_{nk} \right] (E [p] - E [q]) \end{aligned}$$

using the identity $\sum_i w_i \sum_i w_i = \sum_i w_i^2 + \sum_{i \neq k} w_i w_k$, we get

$$\begin{aligned}
&= E[(p - q)^2] + E \left[\sum_k w_k^2 p_{nk}^2 + \sum_{k \neq m} w_k w_m p_{nk} p_{nm} \right] \\
&\quad + \sum_k w_k E[p_{nk}] (E[p] - E[q]) \\
&= E[(p - q)^2] + E[p_n^2] \sum_k w_k^2 + E[p_n] E[p_n] \sum_{k \neq m} w_k w_m \\
&\quad + \mu_n (E[p] - E[q]) \sum_k w_k
\end{aligned}$$

applying the identity $\sum_i w_i \sum_i w_i = \sum_i w_i^2 + \sum_{i \neq k} w_i w_k$ again

$$\begin{aligned}
&= E[(p - q)^2] + (\sigma_n^2 + \mu_n^2) \sum_k w_k^2 + \mu_n^2 \left(\left(\sum_k w_k \right)^2 - \sum_k w_k^2 \right) \\
&\quad + \mu_n (E[p] - E[q]) \sum_k w_k \\
&= E[(p - q)^2] + \sigma_n^2 \sum_k w_k^2 + \mu_n^2 \left(\sum_k w_k \right)^2 + \mu_n (E[p] - E[q]) \sum_k w_k
\end{aligned}$$

If Gaussian convolution is used (i.e. a Gaussian pyramid is used for stereo matching), then $\sum_k w_k = 1$, which results in

$$E[(p + n - q)^2] = E[(p - q)^2] + \sigma_n^2 \sum_k w_k^2 + \mu_n^2 + \mu_n (E[p] - E[q]) \quad (2.31)$$

The first term is the SSD match score of the ideal case, i.e. what we want to compute. The next two terms are just the numbers that are independent of the entities to be matched, as they come from the noise distribution properties; thus, they will not unpredictably alter the behaviour of SSD matching, aside from adding extra noise. The last term, however, can cause serious trouble. If noise has a bias, i.e. $\mu_n \neq 0$, then the redundant $(E[p] - E[q])$ will be introduced to the calculations, and the SSD score will not compute what it should. It exemplifies the well known vulnerability of SSD measure to the camera gain difference when performed on raw intensity images.

If Laplacian convolution is used (i.e. Laplacian pyramid is used for stereo matching), then $\sum_k w_k = 0$, which makes

$$E[(p + n - q)^2] = E[(p - q)^2] + \sigma_n^2 \sum_k w_k^2 \quad (2.32)$$

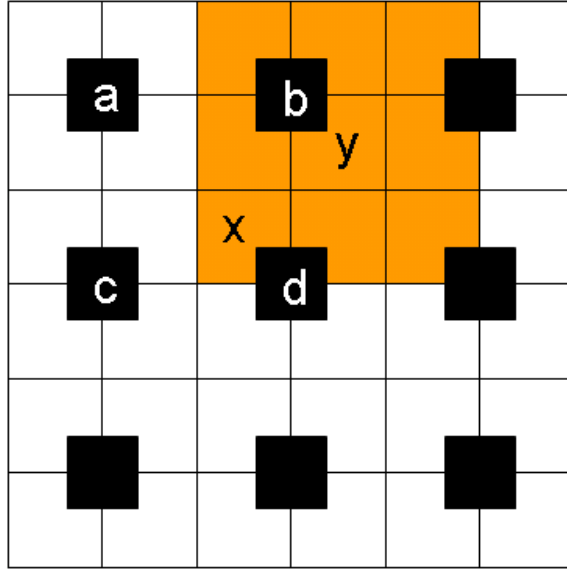


Figure 2.6: Snapshot of the Coarse-To-Fine (CTF) Disparity Estimation Procedure. White cells are pixels at the fine level, black pixels are from the coarse level. Window size is 3×3 . Disparity offset for pixel x can be one of disparities at points a , b , c or d (scaled by 2). If x belongs to the surface described by b , then the correct aggregation window would be centered around point y (shaded in the sketch) and the correct disparity offset comes from point b .

Here our SSD measurement is the desired quantity plus the extra uncertainty that comes from the original noise n . Thus, SSD matching on bandpassed images can be expected to work reliably, even when the corrupting noise is not zero-mean.

2.2.3 Improving coarse-to-fine block-based stereo

The analysis of the previous section reveals the main paths to improving CTF disparity estimation algorithms – more elaborate disparity upsampling procedures must be used and techniques which deal with block-based boundary overreach must be employed. The boundary overreach flaw of block-based matchers is well-researched and a number of efficient remedies were proposed, e.g. shiftable/ overlapping/ adaptive windows [62, 46, 54, 121, 92, 93, 128]. In contrast, the disparity upsampling procedure is specific to CTF refinement and has not been given enough attention, especially in recent stereo research. Hence, in the following, we primarily concentrate on improving the upsampling procedure.

Assume for a moment that we can precisely recover the disparity map at current level k and wish to refine this estimate for level $k - 1$. Hence, consider “ideal stereo” case ($w = 1$), where the only place the errors in CTF processing are introduced is the upsam-

pling procedure of coarsely estimated disparities. This procedure is not uniquely defined and various alternative exist – Nearest Neighbour, Linear, Gaussian interpolations and others [59]. Logically, it should depend on the pyramid construction procedure – Nearest Neighbour is the most suitable for Quadtree pyramids, while Gaussian upsampling is the best of the Gaussian and Laplacian pyramid [24, 59]. The problem is that this reasoning does not quite work for pyramids of (discontinuous) disparity maps.

The snapshot of CTF estimation in Figure 2.6 makes matters more precise. If some point x belongs to a uniform disparity surface, then it makes no difference which upsampling procedure is used, as all coarse level disparity points a , b , c and d would have the same disparity. In contrast, initialization via any of the standard upsamplings of the disparity map recovered at the coarse level leads to difficulties in the vicinity of disparity discontinuities. In this case, disparities for a , b , c and d could be different and, depending on specifics of the situation, upsampled disparities near discontinuities can be incorrectly initialized from the wrong side of the discontinuity (in case of NN interpolation) or come as an average across the discontinuity (in case of Linear or Gaussian interpolation). In either case, subsequent refinement often cannot correct for the poor initialization and recovered surface geometry is compromised near 3-D boundaries. A simple reason for this phenomenon occurring is that high-frequency information, which provides exact discontinuity position, is unavailable at the coarser levels, and hence accurate reconstruction of depth discontinuities is *not* possible based solely on the standard upsampling procedure.

A reasonable solution to overcome such problems would be to use multiple disparity offsets for each fine level pixel, rather than a single offset proposed by standard upsampling procedures. Then, in notation of Figure 2.6, if x belongs to a constant disparity region, then disparity values at neighbouring black points would be the same, which results in a single offset. In contrast, if x is near a 3D boundary (i.e. boundary between regions with distinct disparities), then it is appropriate to search for finer disparities at x using each possible initialization separately, as obtained from a , b , c , d (or even broader areas, if larger windows are used).

Brute-force realization of the above observations entails additional correspondence search at each finer level (one search for each initialization), with final disparity assignment taken as that yielding the best score under the block-matching metric. A closer look suggests a more efficient approach and one that also selects for the best shifted match window about each point. After all, we need to deal with the foreground fattening/shrinking effect as well, and shiftable windows are one of the most efficient and effective remedies. In Figure 2.6, if initialization from b gives the best match for finer level refinement at x , then x and b derive from the same surface; whereas, a , c and d derive from elsewhere. Correspondingly, the best (e.g. 3x3 in Figure 2.6) shifted match window for x would be as shaded. Significantly, the selected window is centered about point y and y gets correct initialization from b via nearest-neighbor upsampling. Analogous conclusions are drawn assuming the best initialization for x derives from a , c or d . In general, the best initialization, match window

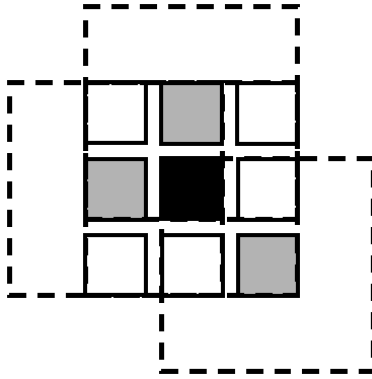


Figure 2.7: Shiftable Window: The effect of trying all 3×3 shifted windows around the black pixel is the same as taking the minimum matching score across all centered windows in the same neighbourhood. Adapted from [102].

and refinement for x are achieved via nearest neighbor (NN) upsampling of the coarser disparity map and subsequent selection of the best disparity estimate derived across all shifted windows that cover x at the finer level. Importantly, it is not necessary to try all window shifts for all initializations: Consideration of possible window shifts with coarse disparity offset taken for central pixel implicitly encompasses possible initializations! Essentially, we extend the observation of [46] to CTF refinement: “The disparity profile itself drives the selection of an appropriate window *and disparity offset*”.

In practice, the desired shiftable window+offset computations for each pyramid level can be realized efficiently in two steps: i) obtain an initial disparity map via central window block matching using Nearest Neighbour upsampled coarse disparity as offset; ii) finalize the disparity map at each pixel by choosing the disparity of the neighboring pixel that has best match score; here, the neighbourhood is that covered by the match window. The latter step is similar to morphological operation on the match score map (erosion for the SAD and SSD match measures) using the aggregation window as a structural element to simulate shiftable windows in the single-scale matching [102]. Note that the proposed approach is not identical to estimating disparity estimates at each level via shiftable windows, as proposed in [46, 17, 91] (shown in Figure 2.7) applied at each pyramid level, because, for each pixel, each shifted window should correspond to a different disparity offset. In the following, we refer to this technique as **CTF shiftable windows**. Mathematical formulation capturing the essential notions is as follows:

$$[im_1^0 = im_1, im_2^0 = im_2, (\forall(j)|1 \leq j \leq l_{max} : im_i^j = (g \otimes im_i^{j-1}) \downarrow_2)] , \quad (2.33)$$

$$[\forall(x, y) : disp^{l_{max}+1}(x, y) = 0] , \quad (2.34)$$

$$[\forall(k)|l_{max} > k > 0 : [\forall(x, y) : disp_0^k(x, y) = 2 \cdot disp^{k+1}(x, y) \uparrow_2 + \quad (2.35)$$

$$arg \min_{d_i} \sum_{(u,v) \in w(x,y)} \rho(im_1^k(u, v), im_2^k(u + 2 \cdot disp^{k+1}(x, y) \uparrow_2 + d_i, v)) ,$$

$$conf_0^k(x, y) = \sum_{(u,v) \in w(x,y)} \rho(im_1^k(u, v), im_2^k(u + disp_0^k(x, y), v)) , \quad (2.36)$$

$$disp^k(x, y) = disp_0^k \left(arg \min_{(u,v) \in w(x,y)} conf_0^k(u, v) \right) \Big] \Big] \quad (2.37)$$

where $disp_0^k$ is the initial disparity at level k , $conf_0^k$ is associated with its match score map, and $disp^k$ is the finalized disparity at level k , as consistent with the two-step procedure described in the paragraph above. All other notation is consistent with mathematical definition of **Module B** (2.2), (2.3), and (2.4). Note that the upsampling procedure \uparrow_2 in this case is Nearest Neighbour, which is defined in (2.28).

The corresponding pseudocode is outlined in **Module C** below:

Module C

```

Reference and matching images are initially
brought into pyramid representation
disp(k,x,y) - disparity for pixel x, y on scale k
conf(k,x,y) - confidence for pixel x, y on scale k
Initialize ref_disp(:, :) to all zeros
For each level k from level_max to 0
  For each pixel (k,x,y)
    Run Module A with search range
      [-delta_d+ref_disp(x,y), delta_d+ref_disp(x,y)];
  End loop
For each pixel (k,x,y)
  In the neighbourhood w of point (x,y)
    find (x_0,y_0) such that conf(k,x_0,y_0) is the best
    and assign disp(k,x,y) = disp(k,x_0,y_0);
End loop
ref_disp = 2*upsampleNN(disp(k, :, :)) /* nearest-neighbour interpolation*/
End loop

```

Carrying on the simulation of Section 2.2.2, we add the simulation of the shiftable window+offset step summarized above. The results of applying CTF shiftable windows are shown in Figure 2.8, from which several conclusions can be drawn:

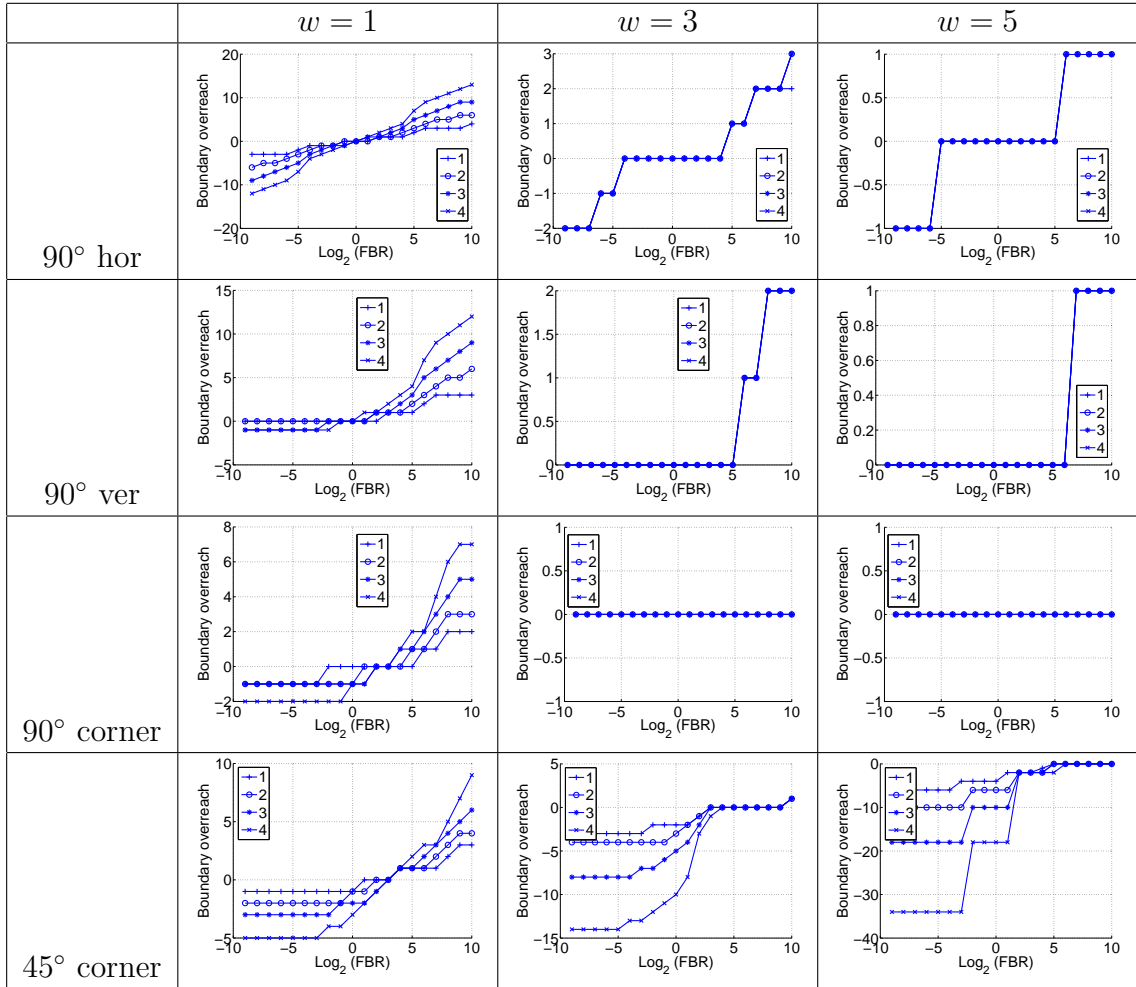


Figure 2.8: Adaptive CTF simulation. Boundary deterioration statistics for corner pixel, pixel on the horizontal and pixel on the vertical boundaries. Negative boundary overreach values (units of pixels) denote shrinking and positive values denote fattening respectively. Each curve in the sketch describes the overreach profile for a certain number of pyramid levels used in the disparity estimation. Aggregation window size is w pixels.

- The strategy of searching for the best offset implemented via CTF shiftable windows pays off well, while the artifacts of rectangular-shaped fixed size windows persist – they cannot reliably recover sharp-shaped boundaries, e.g. in case of 45° .
- The CTF shiftable windows take care of both window shape (in this case, position) and the disparity offset. This is demonstrated by the superior results of $w = 3$ and $w = 5$ adaptive window-and-offset method over $w = 1$ “ideal stereo”.
- Window size $w = 5$ has certain advantages over smaller $w = 3$. Clearly, the former allows greater aggregation and a more stable solution. Moreover, $w = 5$ has somewhat greater ability to recover precise (non-acute) boundaries. The reason is that a 5×5 Gaussian kernel (2.10) is used for pyramid construction, which means that support aggregation should be at least as big as 5×5 in order to have comparable ability to recover from blur to be caused by low-pass filtering.
- Some things remain unrecovered. Specifically, the hardest spots are small objects with large disparity jump between foreground and background. This observation also suggests that CTF advantages might be limited for large baseline stereo with large depth discontinuities. In this case, large disparities should be recovered by using more pyramid levels, and it puts an upper bound on the resolution of details. The general coarse-to-fine tradeoff is to reduce pyramid levels and increase search range in order to get potentially better details at the expense of slower speed and increased match ambiguity.

To complete the discussion, we briefly overview previous work regarding adaptive windows and explain our choice of square shiftable windows similar to [46].

In general, it is well known that CTF disparity estimation corrupts 3-D boundaries. In non-CTF block matching, use of shiftable or otherwise adaptive windows to conform to disparity discontinuities is well established [62, 46, 54, 121, 128]; however, the link to improving CTF disparity refinement seems not to have been stated previously.

One of the first introductions of adaptive aggregation windows for dense stereo can be attributed to Kanade and Okutomi [62], where the authors developed a model of local variations in intensity and disparity and chose the support window in such a way that the produced estimate of disparity had the least uncertainty for each pixel of an image. However, this algorithm was iterative and rather slow.

A few years later, Fusiello et al. [46] developed extremely simple shiftable windows which were fixed in shape but locally shifted in such a way that match score would be maximized. The simplicity and speed of this technique made it widely-used in window-based stereo [102]; moreover it has shown superior performance to earlier work [46]. Shiftable windows somewhat similar to [46] appeared in [79, 47, 17].

Recently, Hirschmuller et al. [54] have analyzed the source of errors near 3-D boundaries and proposed their extension of shiftable windows, which can be called overlapping windows. The idea is to choose the best k small windows for each pixel and construct the final aggregation window by taking the union of chosen small windows. While the developed formulation is slightly more computation-intensive than the previous [46], the latter has the advantage of not restricting the windows to the squared shape, which tends to better adapt to various 3-D boundary outlines and yields even better performance. This advantages has been reflected in experiments [54].

Another recent advance with respect to adaptive windows is Veksler’s variable windows [121]. She makes use of square windows of adaptive size and position. The window cost is composed of the average intensity matching error in the window, biased to larger windows, and biased to smaller match error variance within the window. The final algorithm is computationally intensive and author makes use of dynamic programming and integral images (i.e. sliding window computation) for speed.

Finally, many other approaches to shiftable and otherwise adaptive windows exist, such as model-based windows [82], minimum-ratio cycle windows [19] etc. Additional discussion can be found in [21].

In our formulation, we can use quite small windows for better resolution of 3-D boundary structure, as larger aggregation is made intrinsically available by CTF. This allows us to achieve the 3-D boundary fitting robustness of overlapping windows [54], and avoid complicated construction of variable-sized windows [121]. Interestingly, the modification of shiftable windows used in [102] (referred there as Min Filter, an efficient implementation of [46]) is a special case of CTF shiftable windows **Module C** when the pyramid is degraded to a single level – in this case, each point has the same zero offset and shiftable window+offset simply becomes shiftable windows.

It is also of interest to note that recent work that exploits CTF processing for disparity estimation beyond block matching, e.g. with global methods [74, 49, 87, 109, 6, 44, 41], has yielded strong results; however, the importance of considering multiple offsets in projecting CTF has not been addressed clearly. Ideally, these methods should explicitly try multiple offsets; whereas, the proposed method is naturally more efficient – window placement and disparity offset are tied to eliminate extra search.

Finally, it is interesting to note that use of multiple offsets has been noted in the earlier optical flow literature. A vivid example is Anandan’s framework for computation of visual motion [7], where he used block-based matching and multiple offsets while calculating optical flow in a CTF manner. Nevertheless, he did not use shiftable windows, as they were developed after his work.

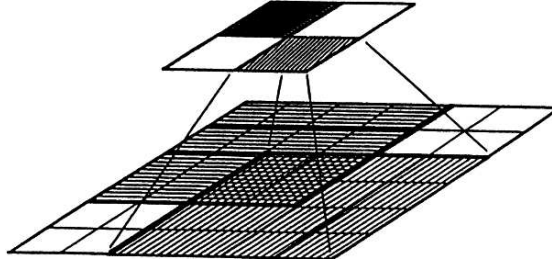


Figure 2.9: The Overlapped Pyramid Projection Scheme. Adapted from [7].

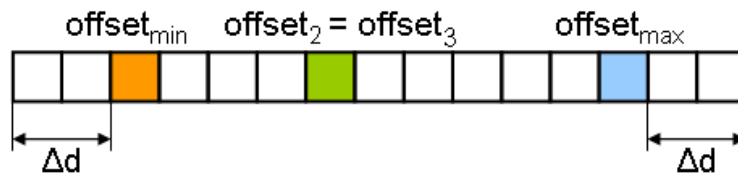


Figure 2.10: Covering Multiple Offsets. An example of extending local search range to cover all possible offsets. Δd is the local search range. Two of four offsets are equal in the example shown here.

2.2.4 Coarse-to-fine non-block-based stereo

As has been established, consideration of multiple offsets is essential for robust performance of CTF block matchers near 3-D boundaries. Global methods are not an exception to this observation. Based on the analysis of Section 2.2.2, Figure 2.5 shows the simulated errors for windows size $w = 1$, which can be treated as running an “ideal stereo” matcher on each resolution level. Note that even in the case of an ideal matcher, but with single disparity offset, errors occur. Unlike block matchers where multiple offsets can be tied to the window configuration, pixel matchers (e.g. global algorithms) require explicit consideration of each possible offset.

The idea of using multiple offsets is not a new one and is reflected in Burt’s overlapped-pyramid projection strategy [25], where it is used to overcome problems of nearest-neighbour interpolation. In Anandan’s words [7], “...disparity of a pixel at the coarse level is transmitted to all the pixels in a 4×4 area at the next finer level; thus, each pixel at the fine level obtains information from four pixels at the coarse level”. Schematically, this approach is depicted in Figure 2.9. Note that in many cases (regions of near-constant disparity) some or all these four different offsets will be identical. Similar to [7], the overlapping pyramid can be directly applied to any CTF stereo algorithm.

A slight twist to the previous solution would be to extend the local search range to include all four possible offsets, as depicted in Figure 2.10. The new offset and search

range for each pixel could be calculated using (2.38), where there could be up to four different coarse disparity offsets.

$$\begin{aligned} offset &= \left\lfloor \frac{offset_{max} + offset_{min}}{2} \right\rfloor \\ SearchRange &= \left\lceil \frac{offset_{max} - offset_{min} + 2\Delta d}{2} \right\rceil \end{aligned} \tag{2.38}$$

Interestingly, this kind of calculation has appeared in application of CTF to dynamic programming [75], where minimum and maximum search range maps are eroded and dilated, respectively, at each CTF level for improved 3D boundaries. The use of single, longer search ranges instead of multiple discontinuous short ones is easier to handle in the dynamic programming framework, albeit with increased processing requirements. However, [75] does not discuss multiple offsets, does not explicitly motivate their solution and does not relate their approach to standard upsampling.

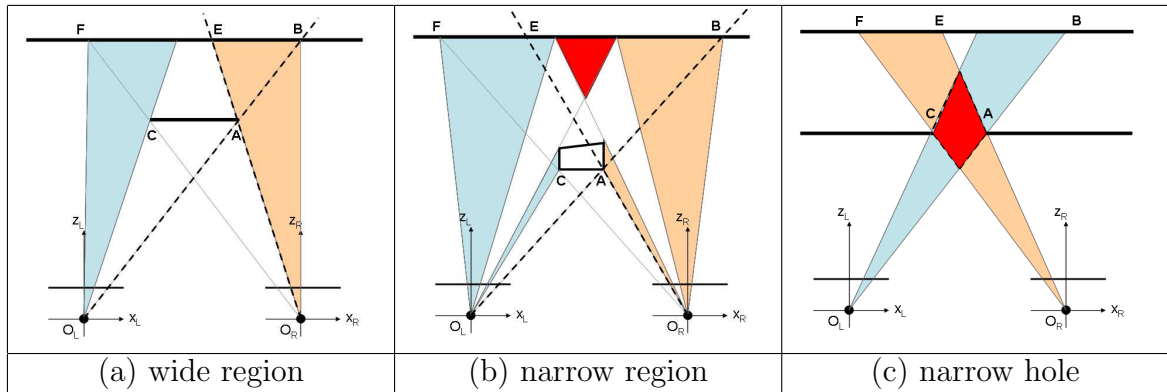


Figure 2.11: Various Cases of Half-Occlusion Geometry. (a) The simplest case occurs when all points on the back surface that are within the “forbidden zone” of the boundaries of the front surface are half-occluded, e.g., A is the right boundary point of the front surface. (b) More complicated situations occur when narrower front surfaces allow portions of the back surface within the forbidden zone of the front surface boundaries to be binocularly visible. Further interposed surfaces in the red (dark grey) region allow for half-occlusion relations to occur *recursively*. (c) Half-occlusions also occur in viewing back surfaces through a narrow hole in a front surface; the back surface is binocularly invisible.

2.3 Half-Occlusions

Points which are visible in only one of the binocular images are called half-occluded. All monocular information, e.g. colour and texture, is available for them, but correspondence cannot be established in principle. That is why such points should be explicitly classified by a stereo algorithm as having depth measurement undefined.

2.3.1 Geometry of half-occlusions

The operative geometric model of image formation is expressed in terms of Figure 2.11a, which shows a top down view of parallel axis (or otherwise rectified) binocular images formed under perspective projection with, e.g., left and right Euclidean coordinate systems defined at the centers of projection, O_l and O_r , respectively, and separated by baseline, b . The Z -axes are taken parallel to the optical axes and increasing toward the orthogonal image planes, located at distance $f = 1$ along these axes. X -axes are parallel to the stereo baseline, increasing to the right and Y -axes are mutually orthogonal to the X and Z axes to yield right handed systems. Let world points be given as $\mathbf{A} = (X, Y, Z)$ and subscripts l and r used to reference points to the left and right coordinates systems, respectively, e.g., \mathbf{A}_l references \mathbf{A} to the left system. Image coordinates are similarly denoted using lower case letters; further, since ensuing developments concentrate on relationships along horizontal

scan lines, image coordinates will be restricted correspondingly, so that perspective yields, e.g., $\mathbf{a}_l = \frac{X_l}{Z_l}$ as the left image coordinate of \mathbf{A} . Given the binocular imaging model, the right image coordinate for \mathbf{A} is given as $\mathbf{a}_r = \frac{X_r}{Z_r} = \frac{X_l - b}{Z_l}$. Correspondingly, disparity (right-based) is given as

$$d_r(\mathbf{A}) = \mathbf{a}_l - \mathbf{a}_r = \frac{b}{Z}. \quad (2.39)$$

Notice that for surfaces of constant Z (fronto-parallel surfaces), disparity is constant.

Half-occlusions always arise near 3-D boundaries when a foreground surface occludes a background surface. Three different configurations are outlined and sketched in Figure 2.11. A typical configuration when foreground surfaces partially occludes the background surface (Figure 2.11a) results in two single half-occluded regions on the left and right sides of the foreground object for right and left cameras respectively (shaded in the sketch). The case with narrow foreground object may give rise to quite complex half-occlusion geometry (Figure 2.11b). In this case, a single foreground object creates multiple disjoint half-occluded regions for both eyes. Note that putting an object in the dark shaded region may give rise to recursive half-occlusion formation. The last case is when a hole in the foreground surface is so small, that the background surface becomes completely binocular invisible (Figure 2.11c). This case is exceptionally hard for computational stereo, as disparity for the entire background object cannot be determined in principle, hence any occluder-occluded interrelationships can not be stated in terms of disparities per se.

It is essential to note that half-occlusions arise only in the “forbidden zones”, i.e. regions where points will appear as violations of the ordering constraint [72], of the foreground point near the occluding boundary. Thus, the relationship between points in terms of corresponding forbidden zones is essential to detection of half-occluded points. In Figure 2.11, angles EAB and O_LAO_R encompass the forbidden zone for point \mathbf{A} .

For present purposes, a useful constraint for half-occlusion processing comes by considering the difference in disparity on either side of the occlusion region and region width. Consider the shaded region on the right side of Figure 2.11a. Let world point \mathbf{A} be the left-most point that is binocularly visible, while world point \mathbf{B} is the right-most half-occluded point (visible only to the right image); let their right image coordinates along a scanline be \mathbf{a}_r and \mathbf{b}_r , respectively. The width of the half-occluded region projected to the right image is

$$\Omega_r^w(\mathbf{B}, \mathbf{A}) = \mathbf{b}_r - \mathbf{a}_r. \quad (2.40)$$

The disparity values for points \mathbf{A} and \mathbf{B} are

$$\begin{aligned} d_r(\mathbf{A}) &= \mathbf{a}_l - \mathbf{a}_r \\ d_r(\mathbf{B}) &= \mathbf{b}_l - \mathbf{b}_r = \mathbf{a}_l - \mathbf{b}_r, \end{aligned} \quad (2.41)$$

with $\mathbf{b}_l = \mathbf{a}_l$ because \mathbf{A} and \mathbf{B} lie along the same line through O_l , the left optical center, by construction. Correspondingly, the change in disparity across the half-occluded region

is given as

$$\begin{aligned}
\Delta d_r(\mathbf{B}, \mathbf{A}) &= d_r(\mathbf{B}) - d_r(\mathbf{A}) \\
&= \mathbf{a}_l - \mathbf{b}_r - (\mathbf{a}_l - \mathbf{a}_r) \\
&= \mathbf{a}_r - \mathbf{b}_r
\end{aligned} \tag{2.42}$$

Now, taking the ratio of disparity change (2.42) to occlusion width (2.40) it is found that

$$\frac{\Delta d_r(\mathbf{B}, \mathbf{A})}{\Omega_r^w(\mathbf{B}, \mathbf{A})} = \frac{\mathbf{a}_r - \mathbf{b}_r}{\mathbf{b}_r - \mathbf{a}_r} = -1. \tag{2.43}$$

It is seen that this ratio is equal to the disparity gradient limit [23]. Further consideration of the geometry illustrated in Figure 2.11 shows that relationship (2.43) between disparity change and occlusion width also holds for regions visible only to the left view of a binocular pair. In this form, the derived constraint will be referred to as the *disparity-change/width constraint* in the following. Note that ‘‘occlusion width’’ refers to the region where occlusion *can* appear. Depending on the situation, the whole area can be occluded (Figure 2.11a), or it can have gaps of binocular visibility (Figure 2.11b).³

The loci of points that yield the value of -1 for the disparity gradient limit lie along a boundary of the forbidden zone [129], e.g. the line through \mathbf{A}, \mathbf{O}_l (and hence \mathbf{B}) in Figure 2.11a. The disparity-change/width constraint captures a subset of a foreground point’s (e.g. \mathbf{A} ’s) forbidden zone as delimited by a background point (e.g. \mathbf{B}) that lies along the forbidden zone boundary. In Figure 2.11a, the constraint captures the portion of the forbidden zone relevant to labeling the segment \mathbf{AB} as potentially half-occluded.

Disparity-change/width can be related to the ‘‘uniqueness constraint’’, i.e. that each point in one image should match to only one in the other: Rearrangement of the terms in (2.43) with substitution from (2.40) and (2.42) yields

$$d_r(\mathbf{A}) + \mathbf{a}_r = d_r(\mathbf{B}) + \mathbf{b}_r, \tag{2.44}$$

where, it is seen that the disparity-change/width constraint constitutes a violation of the uniqueness constraint, as both \mathbf{a}_r and \mathbf{b}_r map to the same location in the left image.

In theory, either of the derived formulae, (2.43) or (2.44), can be used to detect half-occlusions. If the constraint equations for a set of points are satisfied, then there must be one point which is visible (i.e. it is *unique*) and all the rest are half-occluded (i.e. they fall into the forbidden zone boundary of the visible point). In the following, we emphasize (2.44) as it yields a convenient algorithm (**Module D**, see below).

³While definition of the disparity-change/width constraint appeals to the disparity of a half-occluded point, e.g., \mathbf{B} , this should not pose a problem in practice: Let subscript + applied to a point refer to a point immediately to the right, e.g., \mathbf{B}_+ refers to the point immediately to the right of \mathbf{B} . If the surface about \mathbf{B} is taken as locally fronto-parallel, then its disparity is constant in that local region and can be estimated from, e.g., \mathbf{B}_+ , which by definition is binocularly visible.

To arbitrate further between visibility and occlusion a second cue to half-occlusion is employed. Since matches in occluded areas have no physically defined match (corresponding points are not imaged to the other view), any attempted match is expected to have a poor match score, at least for areas having distinctive texture. So, given two or more points satisfying (2.44) or, alternatively (2.43), the point with the best match score is taken as binocularly visible, and the others as half-occluded. We refer to this cue as the *poor match score cue*. Interestingly, the application of this cue does not require the commitment to a certain match measure, as a visible pixel must have *relatively* the best match score in comparison with pixels that violate (2.44). Nevertheless, it is worth mentioning that even such a general approach might be ambiguous when some sample-insensitive or robust match measures are used: The former, e.g. Interval Difference as in [113], tends to return strictly zero for good intensity matches; the latter may return a fixed cost value when the match is bad. Both of these cases should not pose a problem in practice, as cases when rivalrous matches are equally good or equally bad correspond to ambiguous situations anyway.

The proposed approach to half-occlusion processing is able to deal appropriately with Figure 2.11a and 2.11b, but not always with 2.11c. When disparity of some point between **C** and **A** place it in the region shaded with red (dark grey), then the point is not in the forbidden zone of **A** or **C** and the disparity-change/width constraint is never violated⁴. The hole in the disparity map would be smoothed. Any algorithm that relies on visibility constraints will suffer here, as the only peculiarity the region might have is a poor match score.

In our previous work [106] we have applied the *disparity-change/width constraint* and *poor match score cue* in a different fashion by combining them in a Bayesian framework, which allowed us to obtain the *probability* of a pixel being half-occluded, rather than a binary occlusion map. It was even able to solve hard cases like one in Figure 2.11c, although it required an offline learning procedure. We prefer the present formulation to the Bayesian instantiation, as the present formulation is much more efficient and experimentally exhibits greater precision in delineating foreground and background objects.

2.3.2 Occlusions and slanted surfaces

In practice, straightforward use of uniqueness, (2.44), is not robust to slanted surfaces [69] and continuous disparity: Integer quantized disparity values, as recovered by standard block matching, can cause multiple pixels in one image to map to a single pixel in the other. In the current context, the noted problem with uniqueness can be dealt with efficiently as follows. Integer disparity values are interpolated to subpixel precision ([105] used in re-

⁴More generally, points in the dark shaded region of Figure 2.11c will not fall on the forbidden zone boundary of any of the binocularly visible points.

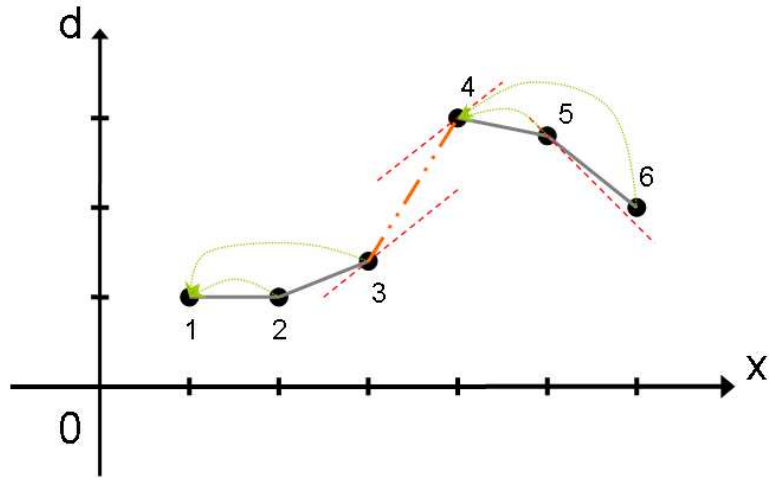


Figure 2.12: Between-Pixel Interpolation. Solid lines indicates that pixels are not in the forbidden zone of one-another; dashed arrows point to the surface ID, being left-most pixel of the surface.

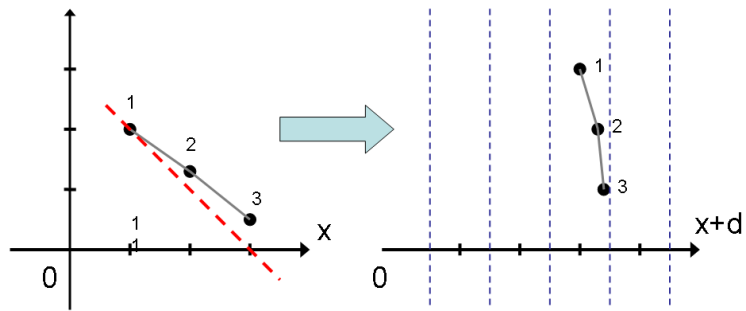


Figure 2.13: Finding Oclusions of Interpolated Surfaces. Uniqueness constraint warps points into the same bin, while interpolation links are preserved.

ported experiments, Chapter 3). Subsequently, disparity relations between adjacent pixels on a scanline are used to group pixels into equivalence classes according to whether or not they are consistent with a single continuous surface. Given this grouping: Pixels consistent with a single surface cannot engage in half-occlusion relationships (violation of uniqueness credited to disparity quantization issues); in contrast, pixels that violate uniqueness and are not consistent with a single surface are considered for half-occlusion.

The disparity relations that yield the desired pixel groupings derive from the widely used *occlusion* and *ordering* constraints [39, 21]. Following the notation style used to derive (2.43) in conjunction with Figure 2.11, consider two scene points \mathbf{F} and \mathbf{C} that project to *adjacent* right image pixels with coordinates \mathbf{f}_r and \mathbf{c}_r , i.e. $\mathbf{c}_r - \mathbf{f}_r = 1$. The condition $\Delta d_r(\mathbf{C}, \mathbf{F}) = d_r(\mathbf{C}) - d_r(\mathbf{F}) \geq 1$ captures the occlusion constraint [39], i.e. there is a half-occluded region between the points in the *other* image; hence, points \mathbf{f}_r and \mathbf{c}_r arise from distinct surfaces. Alternatively, for adjacent image points where one arises from the forbidden zone of the other, ordering along scanlines will be violated in the left vs. right images. In particular, for any point, \mathbf{P} in the forbidden zone of \mathbf{A} , $\Delta d_r(\mathbf{P}, \mathbf{A}) \leq -1$ [129], with equality when points lie along a forbidden zone boundary (2.43). In any case, $\Delta d_r \leq -1$ indicates that the involved points arise from different surfaces; although, they might both be visible as in Figure 2.11b.

Combining the given disparity relations, it is seen that: $\|\Delta d_r\| \geq 1$ implies the presence of a discontinuity between adjacent pixels; otherwise, the pixels are consistent with a single continuous surface. Consideration of $\|\Delta d_r\|$ between adjacent pixels allows all pixels along a scanline to be grouped into the desired equivalence classes (each class consistent with a single continuous surface). The grouping process is visualized in Figure 2.12 and can be easily implemented by carrying an extra pointer to the head of the chain with each pixel (shown in Figure 2.12 with dotted arrows), while detecting half-occlusions.

Subsequently, in determining half-occlusion relationships based on uniqueness and match score, pixels in the same class cannot compete for visibility: They are consistent with a single underlying surface, even if they map to identical integer disparities. As schematically illustrated in Figure 2.13, although points belonging to the same slanted surface fall in one bin, the enforcement of equivalence class relationships will prevent them from being marked as occluded. A linked list could be a good implementation for bins, as they will have few occupants, most often only one pixel. Note that bins with no pixels correspond to half-occlusions for the other view, and are essentially described by the “occlusion constraint” [39, 110].

2.3.3 Cues to half-occlusion detection

Prior to converting the analysis of Sections 2.3.1 and 2.3.2 into a working algorithm, it is essential to relate the introduced *disparity-change/width constraint* and *poor match score cue* to previous approaches for half-occlusion detection.

The proposed approach to half-occlusion is most similar to others that also explicitly consider disparity of occluded and occluding surfaces. The “occlusion constraint” says a discontinuity in right-based disparity corresponds to a half-occluded region in left-based disparity and vice versa, e.g., [39, 58, 50, 110]. An advantage of the current approach is that it is defined with respect to a single view, making it more natural to use without two-way matching. Moreover, we showed that our method can yield the half-occlusions for the other view as a byproduct.

The “ordering constraint” also considers disparity of occluder and occluded, as it imposes strict ordering of matched points in left and right images [129, 58, 102, 61] (essential to many dynamic programming-based matchers, e.g., [47, 30, 17, 32]) and as a result can disallow matching in half-occlusion regions [39, 21]. However, ordering can be violated in physically realizable view conditions that do not involve half-occlusion, like thin foreground objects [71], e.g. Figure 2.11b. In contrast, disparity-change/width is just a limiting case of ordering (i.e. it corresponds to a *boundary* of the forbidden zone, as discussed above) and more specific to half-occlusion than ordering.

Disparity-change/width can be recast to match uniqueness (2.44), widely used in binocular matching for detection of half-occlusions [69, 21] and more consistent disparity maps in general [132, 107]. Match uniqueness explicitly enforces a one-to-one mapping between points in the images. To deal with uniqueness violations from physically realizable situations subject to discretization (e.g. slanted surfaces), recent approaches use a “generalized visibility constraint”, enforcing one-to-one mapping between continuous *intervals* by affine parameterized matching on segments, rather than individual pixels [90, 16, 35]. While these methods are robust to slanted surfaces, they are expensive and usually rely on prior segmentation. In contrast, the emphasis of the proposed approach is on methods that can directly impact local block-matching.

Other approaches that explicitly consider both surfaces involved in half-occlusion are “bimodality tests” [39], which rely on the observation that histogrammed disparity in the vicinity of half-occlusions can show bimodal distributions as both foreground and background surfaces are captured. Again, the disparity-change/width constraint is tighter, explicitly stating the relationship between disparity values of the surfaces which are covered by the aggregation window; moreover, it is faster and easier to apply as no arbitrary intrinsic parameters are to be chosen. A potential shortcoming of all bimodality test approaches arises when noise in matching or local surface geometry yields disparity patterns that mimic those of half-occlusion, e.g. steep surfaces with respect to the views.

The current approach also makes use of match scores in deciding which points are binocularly visible vs. half-occluded. Previously, match scores have been used in diagnosing half-occlusion in two ways. First, unidirectional match scores are examined for patterns indicative of match failure; in some cases patterns of interest involve rapid change in match score [39]. More straightforwardly, poor match quality is used by many dynamic programming and graph cut implementations, where the occlusion cost term depends on

match quality [102, 21]. A recent cooperative matcher [132] also uses poor matches to filter out half-occlusions (as well as other matching errors). Poor matches defined by colour differences at aligned image locations also have been used to diagnose half-occlusion [108]. In summary, the poor match constraint used in this paper is an instance of this type of approach as it simply looks for locally bad matches. Second, inconsistencies between bidirectional matches are detected, i.e. “left- right checking” [39, 48, 54, 52], a method that requires disparity maps for both views. While such approaches can detect half-occlusions, they are not specific to this situation; rather, they more generally diagnose problems in matching, e.g. from various noise sources.

Significantly, the two broad classes of approach to half-occlusion detection discussed in the previous paragraphs are complementary: methods based on analysis of half-occlusion geometry predict the relationship between disparities that arise on either side of a half-occluded region; whereas, methods based on considerations of match quality predict what is expected within a region of half-occlusion. From this perspective, the present work encompasses a wide range of approaches (including all methods outlined and compared in [39]), even as it yields a method that is more specific to half-occlusion than other approaches, which often are more generally aimed at diagnosing errors in matching.

2.3.4 Occlusions in coarse-to-fine stereo

As outlined in Section 2.2, CTF stereo is based on refining initial coarse disparity estimates using images of higher resolution – coarse disparity value is taken as an offset and new values within a small search range are tried. But what is to be done when more complete information is available, i.e. coarse disparities *and* half-occlusions are supplied? More specifically, how should one refine half-occlusions in CTF? Significantly, the answer to this question will allow for a cooperative occlusion-disparity estimation procedure in local block matchers – a very useful characteristic that only cooperative and global algorithms truly possess. Indeed, cooperative estimation of disparity and half-occlusions is essential as disparity information is needed for reasoning about half-occlusions and occlusion information is needed to construct support and define the disparity search space correctly. In this light, it is surprising that the problem of half-occlusion detection in a coarse-to-fine framework has not been clearly addressed before.

Several solutions can be considered:

1. Detect possible half-occlusions only at the finest pyramid level. This means that we refuse the ability for cooperative disparity and occlusion estimation, i.e. do not address this specific the problem. Our previous work [106] pursued this line of attack in preparation for the more complete solution now considered.

2. Consider coarse half-occlusion as a special offset and declare finer resolution pixels as half-occluded, if their coarse resolution parents are. This kind of approach might have the significant drawback that half-occlusion boundaries will be poorly recovered (by analogy with disparities, see Section 2.2.2). A more serious problem might be the inability to recover from coarse error, as occluded pixels at coarser levels may turn out to be a thin structure or a slanted surface at a finer level.
3. Complete the coarser disparity map by extrapolating neighboring background surface disparity values into the occluded regions, i.e. explicitly incorporate half-occlusion information in the disparity map⁵. Following extrapolation, upsample the resulting disparity map and repeat the entire process at each finer level.

We pursue the last approach: We extrapolate the background surface disparity into half-occlusions by constant disparity propagation under a constant depth assumption. This yields better ability to initialize disparity estimation at finer levels, especially in the vicinity of half-occlusions, and the actual CTF estimation procedure can be left essentially unchanged. Moreover, extrapolation forces occluded pixels to have disparity values of neighbouring surfaces, which will make pixels converge to a correct estimate if they really come from this surface, or will worsen their match score if the disparity assignment is incorrect, i.e. pixels are really half-occluded – this will allow for truly half-occluded pixels to be re-detected at the finer level.

In conclusion, even if a slanted surface has not been recovered correctly at the coarser level, i.e. half-occlusion is falsely detected, that should not pose a problem in practice. In this case, the slanted surface is approximated by piecewise frontoparallel patches in a staircase fashion, i.e. the disparity differences between neighbouring patches is at most 1; hence, the extrapolation procedure will modify the disparity in the falsely-detected occluded region by at most 1, which means that reliable refinement is still possible.

2.3.5 Final half-occlusion detection algorithm

Overall, the proposed approach to detecting half-occlusions at any given pyramid level that employs *disparity gradient/occlusion width* and *poor match score* cues can be encapsulated

⁵It is worth mentioning that any extrapolation procedure (be it assuming constant disparity, or constant disparity gradient of the background surface, or any other) is just an ad-hoc solution, as actual depth of occluded points could be arbitrary. Refer to [57] for specific examples.

as follows.

$$[\forall y] : sId(1, y) = 1 \quad , \quad (2.45)$$

$$(\forall x \mid x > 1 \wedge |disp(x-1, y) - disp(x, y)| < 1 : sId(x, y) = sId(x-1, y)) \quad , \quad (2.46)$$

$$(\forall x \mid x > 1 \wedge |disp(x-1, y) - disp(x, y)| \geq 1 : sId(x, y) = x) \quad , \quad (2.47)$$

$$[\forall x] : [\exists x' : sId(x, y) \neq sId(x', y) \wedge disp(x', y) + x' = disp(x, y) + x \wedge \quad (2.48) \\ conf(x, y) < conf(x', y)] \equiv occl(x, y)]$$

where *disp* is a (subpixel) disparity map, *conf* is a match score map (higher score signals better match) for the calculated disparity map, *sId*(*x*, *y*) is the surface equivalence class identifier for each point *x*, *y* (as described in Section 2.3.2), and *occl* is a binary half-occlusion map (*false*, or 0, denotes visible and *true*, or 1, denotes half-occluded). Note that (2.45)-(2.47) specify the construction of surface equivalence classes, while (2.48) describes the half-occlusion inference procedure itself.

The corresponding pseudocode for operating at a single pyramid level is outlined below.

Module D

```

For each scanline
  Define pixel equivalence classes via interpixel
    disparity differences |d(x+1)-d(x)|
  Map each point x to cell x+d(x) in a 1D array
  /* points in a single cell violate uniqueness */
  For each cell in the array
    Find the point with highest match score
    Mark it and all other points in the cell coming
      from the same surface as visible
    Mark all other points as occluded
  end loop
end loop

```

It is essential to note that **Module D** has no free parameters and the procedure is local (subject to match window) at each pyramid level. More attention can be given to the computational complexity of the implementation. The algorithm is fast and runs in $O(N)$ time, where N is the number of pixels, but the current instantiation in the form of **Module D** can be parallelized only up to a scanline. The reason is that pixel equivalent classes must be formed by traversing the pixels in each scanline sequentially. At the same time, the case of several consecutive pixels belonging to the same surface and, hence, falling into the same bin, corresponds to highly inclined surfaces with the disparity gradient being very close to the forbidden zone boundary. Taking into account the facts that such configurations are not very common and block matching stereo has troubles in recovering highly-slanted surfaces anyway (due to local fronto-parallel surface assumption in the aggregation window), we

can consider the equivalence surface relationship only between consecutive pixels along the scanline. This modified algorithm is formulated below:

Module D-modified

```
For each scanline
  Map each point x to cell x+d(x) in a 1D array
  /* points in a single cell violate uniqueness */
  For each cell in the array
    Find the point with highest match score
    and mark it as visible /* vis */
    For each other point /* cur */
      If |d(x_vis)-d(x_cur)| >= 1
        Mark current point as occluded
      Else
        Mark current point as visible
      End if
    End loop
  End loop
End loop
```

Module D-modified requires only adjacent pixel comparisons and hence yields to greater parallelization than the original **Module D**.

2.4 Colour and intensity segmentation in computational stereo

As established in Chapter 1, 3-D boundaries delineating different objects are essential for practical applicability of stereo. At the same time, 3-D boundaries usually coincide with colour and intensity discontinuities. This natural phenomenon is significant in the current context and many recent stereo algorithms have benefited by using some form of colour segmentation. Specific examples include: Initial segmentation of the images is performed and correspondence is established directly on the segmented patches [115, 16, 35]; disparity and colour segmentation are performed simultaneously in a single energy minimization framework [77]; intensity gradient is used as a line process in diffusion to get sharper boundaries [74]; the smoothness term of the global cost function is relaxed for regions which are different in colour and vice versa [69, 111, 116, 110]; pixels in the support window are weighted according to their colour similarity with the central pixel [131, 92, 93, 128]. Further, there is always an option to post-process the disparity map based on edges detected in the original images; here, virtually any stereo algorithm could be used to get the initial disparity.

2.4.1 Segmentation-driven shiftable windows

Our strategy to introduce the colour cue in the adaptive CTF procedure **Module C** is based on the following idea. We used shiftable windows of fixed square size, which is essential to have the ability to search for the best CTF disparity offset and alleviate the problem of boundary overreach (refer to Section 2.2.3 for details). Now, the locally best window is chosen based not just on the match score alone, but also on some measure that maximizes the number of pixels within the support window belonging to the same surface based on the colour cue. By doing this, we essentially want our window to maximally cover the correct surface patch, i.e. maximize the presence of the correct surface in the window.

For the sake of exposition, we will formulate our strategy for intensity-based segmentation and describe a generalization to the full colour cue later.

Inspired by segmentation-based windows introduced by Yoon and Kweon [128], we introduce the intensity similarity and proximity cues for pixels. We can outline the procedure from the maximum likelihood (ML) point of view

$$P(d_{x,y}|I(x,y)) \propto P(I(x,y)|d_{x,y}) = \mathcal{L}(x,y,d) \quad (2.49)$$

where $P(d_{x,y}|I(x,y))$ is the probability of point (x,y) having disparity $d_{x,y}$ given point's intensity $I(x,y)$, and $P(I(x,y)|d_{x,y})$ is the likelihood of point (x,y) with disparity $d_{x,y}$ to have intensity $I(x,y)$. For brevity in subsequent calculations, let $P(I(x,y)|d_{x,y}) = \mathcal{L}(x,y,d)$ with $d_{x,y} = d$.

Consider an arbitrary point (x, y) at pyramid level k for which the best window configuration is to be found, and, hence, disparity is to be determined. Specification of a likelihood model, $\mathcal{L}(x, y, d)$, that combines disparity estimation, colour segmentation and proximity requires definition of three corresponding component likelihoods, $\mathcal{L}_\rho(x, y, d)$, $\mathcal{L}_\mathcal{I}(x, y, d)$ and $\mathcal{L}_\pi(x, y, d)$, respectively.

First, assuming independence of pixel intensities within the aggregation window⁶ Ω , the likelihood of assigning disparity d to point (x, y) at level k can be modeled as an exponential distribution for simplicity, in particular

$$\mathcal{L}_\rho(x, y, d) = \prod_{x_i, y_i \in \Omega} \exp\left(-\frac{1}{\lambda} \rho(I_{ref}(k, x_i, y_i), I_{other}(k, x_i + d, y_i))\right). \quad (2.50)$$

Looking ahead, the minimization of the negative log-likelihood (2.50) is equivalent to minimizing the match measure (e.g. SAD) as in **Module A**.

Second, the likelihood of the intensity similarity cue also can be described by an exponential. Experimentally, we have found that intensity differences between arbitrary pixels in an image are reasonably well approximated by an exponential; moreover, previous research uses this model [128]. Additional assumption of pixel intensity independence (as used in calculation of (2.50)) results in the likelihood

$$\mathcal{L}_\mathcal{I}(x, y, d) = \prod_{x_i, y_i \in \Omega} \exp\left(-\frac{1}{\alpha} |I_{ref}(k, x, y) - I_{ref}(k, x_i, y_i)|\right). \quad (2.51)$$

Third, the proximity cue is a simple heuristic that closer points are more important in the calculation of disparity of a point (x, y) . The likelihood for a proximity cue is similarly modeled as

$$\mathcal{L}_\pi(x, y, d) = \prod_{x_i, y_i \in \Omega} \exp\left(-\frac{1}{\beta} \sqrt{(x - x_i)^2 + (y - y_i)^2}\right). \quad (2.52)$$

Note that λ , α and β are free parameters of the corresponding model distributions. Finally, taking independence of goodness of match, ρ , intensity cue, \mathcal{I} , and proximity cue, π , the final likelihood of a point (x, y) having disparity d can be expressed as a multiplication of

⁶Independence of pixel intensities in the aggregation window is the most widely used assumption in correlation-based matching. As an example, both SAD and SSD match measures are derived using this assumption. Finally, note that independence of intensities is used only in calculation of match correlation score, while disparities are definitely not independent.

From the point of signal corruption, independence can be motivated by assuming that the noise in the image is independently and identically distributed.

the three corresponding likelihood terms (2.50), (2.51) and (2.52) to yield

$$\begin{aligned}
\mathcal{L}(x, y, d) &= \mathcal{L}_{\mathcal{I}}(x, y, d)\mathcal{L}_{\rho}(x, y, d)\mathcal{L}_{\pi}(x, y, d) & (2.53) \\
&= \prod_{x_i, y_i \in \Omega} \exp\left(-\frac{1}{\lambda}\rho(I_{ref}(k, x + x_i, y + y_i), I_{other}(k, x + x_i + d, y + y_i))\right) \\
&\quad \times \prod_{x_i, y_i \in \Omega} \exp\left(-\frac{1}{\alpha}|I_{ref}(k, x, y) - I_{ref}(k, x_i, y_i)|\right) \\
&\quad \times \prod_{x_i, y_i \in \Omega} \exp\left(-\frac{1}{\beta}\sqrt{(x - x_i)^2 + (y - y_i)^2}\right).
\end{aligned}$$

In theory, maximization of the overall likelihood, $\mathcal{L}(x, y, d)$, yields the desired disparity. Following common practice, we instead eliminate the exponentials by minimizing the negative log-likelihood multiplied by λ :

$$\begin{aligned}
-\lambda \log \mathcal{L}(x, y, d) & & (2.54) \\
&= \sum_{x_i, y_i \in \Omega} \rho(I_{ref}(k, x + x_i, y + y_i), I_{other}(k, x + x_i + d, y + y_i)) + \\
&\quad \lambda \left(\frac{1}{\alpha} \sum_{x_i, y_i \in \Omega} |I_{ref}(k, x, y) - I_{ref}(k, x_i, y_i)| + \frac{1}{\beta} \sum_{x_i, y_i \in \Omega} \sqrt{(x - x_i)^2 + (y - y_i)^2} \right)
\end{aligned}$$

The negative log-likelihood formulation (2.54) allows us to have a better understanding of what each term of (2.54) means. The first term is the actual match score, which motivated modeling the corresponding likelihood component (2.50) with an exponential distribution. This term should be weighted high and play an important role in the final calculation (i.e. λ should not be very big), as it is able to intrinsically search the right disparity offset while going CTF. The second term is the cue of how each pixel in the aggregation window is similar to the pixel which is to be matched – the segmentation cue. The third term after a close examination is just a bias toward a central window – it can be calculated offline, because it depends only on the position of the window’s centre; its value is maximum when the matching pixel is exactly in the centre and falls off as the pixel moves to the border of the window. Usually $\beta > \alpha$, because this bias should not be very strong, but some bias is beneficial to avoid the blocky artifact of shiftable windows, as exhibited in [46, 102] and Section 3.2. Thus, the second and third terms (colour segmentation and proximity bias) provide guidance in the choice of best window – such a window still must yield a good match score thanks to the presence of the first term.

There are a number of ways to improve the model (2.53) even further by exploiting distributions that suit real data better (although, it might be very hard to determine these distributions in general) or introducing priors and upgrading the procedure to a Maximum a

where $-\lambda \log \mathcal{L}(x, y, d)$ is defined as in (2.54) or (2.56) and all other notation is consistent with the one for **Module C**. The corresponding pseudocode statement can be given as follows.

Module E

```

Reference and matching images are initially
  brought into pyramid representation
disp(k,x,y) - disparity for pixel x, y on scale k
conf(k,x,y) - confidence for pixel x, y on scale k
Initialize ref_disp(k,,:) to all zeros
Loop for level k to 0
  For each pixel (k,x,y)
    Run Module A with search range
      [-delta_d+ref_disp(x,y), delta_d+ref_disp(x,y)]
  End loop
  For each pixel (k,x,y)
    For each point (k,x_i,y_i) no further than aggregation
      window size
      Evaluate (2.56) if greylevel images or (2.58) if colour images
        for each rectangular window centered at (k,x_i,y_i)
      Choose the window with minimum cost
        /* let it be centered at (k,x_min, y_min) */
    End loop
    disp(k,x,y) = disp(k, x_min, y_min)
    conf(k,x,y) = conf(k, x_min, y_min)
  End loop
  ref_disp = 2*upsampleNN(disp(k,,:)) /* nearest-neighb. interp.*/
End loop

```

2.4.2 Relation to other segmentation-based windows

We now relate our approach to other work that has used colour and intensity segmentation cues in the context of local matching. Zhang and Kambhamettu [131] pre-segment the images and construct windows such that they consist of pixels of the same segments only. Park et al. [92] proposed a formulation where the neighbour pixels not similar to the centre one are excluded when computing the local correlation value. Similarity is calculated using the L_2 norm and the decision of whether to include the pixels in the support region is based on a pre-defined threshold. Patricio et al. [93] developed a very similar formulation, but calculated the threshold in an adaptive manner (set as the average colour difference within a window). Yoon and Kweon [128] communicated roughly the same idea in a more formal way – pixels in the window were weighed based on colour similarity with the central pixel and based on the spatial distance between the pixels – the authors claim to have encoded

the basic Gestalt grouping principles (proximity and similarity) in a simple and straightforward fashion. Experimental analysis has shown that this class of methods outperforms all previous shiftable/adaptive/overlapping window techniques when images are colour-rich and relatively easy to segment[128], as they are able to construct windows which *exactly* shapes to the 3-D boundary and can use windows of large size (which improves performance in textureless regions).

Finally, it is important to say that none of these approaches were used in the context of CTF processing, and their behaviour across scale has not been investigated previously.

2.4.3 Precision versus robustness

In this subsection, we further motivate our particular approach to weighting the contribution of match-measure driven window shifting and colour segmentation as a solution to CTF boundary preserving stereo. We reconsider alternative formulations such as robust match measures, pure shiftable and pure colour segmentation windows and show that they all compromise either precision or robustness. In this light we motivate our choice as a solution that provides both precision and robustness.

As stated in Section 1.2.2, the major problem for good 3-D boundary recovery is the correct aggregation of match support (recall Figure 1.5 in Section 1.3.6). In the context of a block-based matcher, the tricky part is to find correct aggregation windows for each point in the scene. If a point lies far from a 3-D boundary, support could be correspondingly large (bigger size generally results in more reliable matches) and a symmetric square window is a good choice in terms of implementation efficiency. If a point lies very close to a 3-D boundary, then the support window should cover only the object to which the point belongs – call these object points *inliers*. Points which do not belong to the same object as the point of interest, should not be included in the disparity calculations (i.e. aggregation) as the fundamental assumption of the uniform disparity within aggregation window does not hold – call these points *outliers*.

The concept of inlier-outlier is intuitive and useful in this situation. It allows us to cast the aggregation step in terms of robust statistics and all difficulties in matching such as occlusions, specularities, non-Gaussian noise can be treated as outliers. In classical stereo, aggregation is done by calculating the first moment (mean) of the similarity distribution within the window, e.g. SAD, SSD, NCC etc. omitting normalization by the window size. Every point is treated as an inlier, which results in the inevitable corruption of 3-D boundaries, as investigated in Section 2.2.

Several strategies can be exploited in order to treat the outliers:

- Make the match metric itself robust
- Choose another support window to avoid outliers (bad match scores) – a technique known as shiftable/overlapping/variable windows (Section 2.2.3).

- Use some other knowledge, e.g. colour, to explicitly label the pixel as an outlier either in terms of probabilities or simple binary labeling.

Note that for many well-known match measures the corresponding similarity distribution is unimodal and outliers are characterized by being very far from this mode. More specifically, for SAD/SSD, outliers have very high dissimilarity values.

We give a brief overview to each method, identify their pros and cons and conclude with a proposal to include the colour (intensity) cue in this framework in the best possible way.

Robust match measures

Widely-used area-based match measures such as SAD, SSD, NCC, etc. [21] are intensity based, and are optimal for Gaussian noise distributions. Meanwhile, real image matching is usually characterized by outliers, and heavy-tailed error distributions suit the purpose of robust estimation better. Moreover, matching can be done not directly on intensity information, but on the rank-order statistics of the intensity values in the windows (Rank, Census transforms [130], and other ordinal measures [12]).

$$cost(x, y, d)_{SAD_{robust}} = \sum_{(u,v) \in w(x,y)} \min(|im_1(u, v) - im_2(u + d, v)|, \tau) \quad (2.60)$$

Here, we discuss the benefits of robust match measures in the vicinity of 3-D boundaries. Consider truncated SAD (2.60) for clarity. If an aggregation window crosses a 3-D boundary, the match score component for the points coming from different surfaces will have high values that would be clamped to some smaller number, i.e. threshold τ . Thus, the outliers will be forced to have smaller values and their cumulative influence will be diminished. Note though, that outliers are not eliminated from the match score computation, which means that if their proportion is high, e.g. a pixel of interest on a 3-D corner, the disparity estimate cannot be expected to be reliable.

Shiftable windows

Here, the window is constructed, or rather chosen, in a way that the match score would be the best possible – low matches that characterize the outliers are avoided. The main driving force of this method is that central-based windows yield bad estimates near the boundaries, as the window covers two or more surfaces – alignment is never good in this case, so the confidence of match will be poor. By choosing a window configuration which yields a better confidence of match we essentially choose a window which results in the best possible alignment, i.e. it covers only one surface. Such a setup is identical to minimizing the presence of outliers and could be performed as a morphological operation (erosion in

case of SAD, SSD and dilation in case of NCC) on the disparity search space, as discussed in Section 2.2.3. Again, we have just described the simplest shiftable window principle [102], which works very well in practice and possess these main characteristics:

- Windows are of the restricted shape. Usually they are rectangular, so the problem is tractable and implementation is easy.
- Smaller windows result in higher accuracy of 3-D boundary estimation. Because windows are rectangular, very fine structures like sharp corners will be damaged, as aggregation with uniform window is a low-pass operation.
- Implementation is both easy and fast in the case of parallel and sequential computations, as no prior information is used in aggregation – pixelwise cost is reused by the neighbourhoods in the aggregation step.

Overall, this type of window was originally designed to behave better near 3-D boundaries and it succeeds by being a robust calculation that does not rely on any extra assumptions [102]. It behaves quite well in the presence of small specularities, non-Gaussian noise and occlusions. So, these windows are not spatially precise (they have fixed shape), but they are robust. Importantly, empirical investigations have shown that shiftable windows behave better than the use of robust measures with fixed windows [102]. Hence, we can declare that shiftable windows largely subsume the robust cost calculation for the block-based matcher, as described above.

Segmentation-based windows

How wonderful it would be if we knew the inlier-outlier membership in advance of disparity estimation! Here we try to predict this membership based on colour segmentation, i.e. by introducing extra information. As overviewed in Section 2.4.2, the shape of the window is constructed based on the assumption that pixels of similar colours come from the same surfaces. The following outlines the main characteristics of such methods.

- Windows of arbitrary shape are constructed. This is definitely good when the scene possesses complex boundary outlines, like sharp angles, which are destroyed by square windows.
- Windows can be quite large. Before, square windows should have been of the smallest possible size to recover boundaries as precisely as possible – small size, on the contrary, is disadvantages when poorly textured surfaces are present. Segmentation-based windows are constructed in such way that they do not cross 3-D boundaries by definition, and hence, their window size can be made as large as possible to recover the disparity for textureless regions unambiguously.

It is worth saying, that such reasoning does not apply in the case of non-fronto-parallel surfaces, as the constant depth assumption is used within the window. This restriction does not allow support windows to be *arbitrarily* large, unless affine or more sophisticated matching is used.

- Implementation is easy, but slow in the case of sequential computation – each pixel requires unique treatment for custom window design (of potentially large size) and cannot be sped up with sliding window techniques [102].

Theoretically, this type of window behaves perfectly in both cases (3-D boundaries and textureless regions), but under a strong segmentation assumption. For example, this assumption is not reasonable when surfaces are heavy textured (luckily, conventional stereo methods behave well in such situations), or when there are photometric errors, unmodelled noise, and occlusions. Interestingly, regions near occlusions suffer the most – the constructed local support will include the occluded regions, as they might be perfectly consistent with monocular segmentation, and this constructed aggregation region will not be adequate for the matching image. Another example of when the segmentation cue may fail is when similarly coloured patches coming from different surfaces are projected closely to each other in one of the images – the constructed aggregation window can be composed of these two patches and, hence, will not be matched correctly, because spatial location of these two patches is different in the other image.

Thus, if the segmentation cue fails, the matching is done over a big incorrect window, which might have severe consequences. Hence, this type of window is spatially very precise (as they have ability to adapt to exact shape of 3-D boundary), but not robust.

Adaptivity and segmentation in coarse-to-fine processing

The outlined tradeoff between the major classes of local matching windows is of even bigger concern when coarse-to-fine computations are used. In CTF, robustness is very important, as we must have an ability to recover from coarse level errors; large support is not as crucial as in single-level matching, because it is aggregated at coarser scales; segmentation is tricky on a coarse level, because low-passed filtering is performed to obtain coarse resolution images, which blurs 3-D boundaries (as analyzed in Section 2.2.2).

These observations are confirmed by our experiments, which have shown the better overall performance of the shiftable windows over segmentation-based windows within the coarse-to-fine framework, especially near 3-D boundaries (see Section 3.4).

As a result of this analysis, we developed our approach to keep the advantages of CTF shiftable windows (adaptive offset, substantial reduction of boundary overreach, robust performance near occlusions) even while exploiting colour segmentation to guide shifting (2.54, 2.56). This approach possess three main advantages. First, colour segmentation can

guide match support, yet, is robust to situations when colour segmentation fails, e.g. highly-textured regions, as aggregation window will not degenerate to constellations of disjoint pixels. Second, windows can shift to define best support in the absence of the colour cue. Third, the essential ability to handle multiple offsets is preserved.

2.5 Recapitulation

This chapter started with the statement of basic, single scale block matching algorithm **Module A** and a basic CTF extension, **Module B**. As a result of the analysis of how CTF corrupts 3-D boundaries (Section 2.2), an improved version of the CTF stereo algorithm, Adaptive CTF has been developed, as embodied in **Module C**. Section 2.3 has been devoted to the investigation of half-occlusion phenomena and culminated with the half-occlusion detection procedure **Module D** (or its alternative **Module D-modified**), which can be used in conjunction with any of **Module A**, **Module B**, or **Module C**. The particular combination of **Module C** and **Module D** is of our prime interest, as it is expected to perform robustly near 3-D boundaries and recover half-occlusions and disparity in a CTF cooperative manner.

Section 2.4 has introduced additional monocular cues to stereo processing and proposed improvements to the Adaptive CTF **Module C**: **Module E** that handles the intensity/colour information. This variation on Adaptive CTF can be combined with **Module D** to gain the benefit of half-occlusion detection analysis.

Chapter 3

Experimental evaluation

3.1 Methodology

The algorithmic instantiations of **Module A**, **Module B**, **Module C**, **Module D** and **Module E** have been implemented in C. The resulting code can be compiled and executed on any compatible environment as it does not make use of special-purpose instructions or libraries.

To test the proposed algorithmic advances in a variety of situations, we use two classes of data: images of lab scenes, for which the ground truth has been obtained, and images of natural scenes.

Testing on lab scenes is very informative, because the ground truth is supplied. In this paper we use the Middlebury College Stereo dataset [3], which consists of scenes of various complexity. Moreover, most other stereo algorithms are tested on this data set, which gives us an ability to compare our results to the state-of-the-art solutions. The scenes themselves and the associated disparity and occlusion ground truth are shown in Figure 3.1.

Testing on real, naturalistic scenes is vital, as the algorithm must operate in real world situations. Our real scene database (Figure 3.2) consists of the *Flower Garden (FG)* scene from the Brown university database [1] and two scenes *Stephen1* and *Stephen2* provided by an industry collaborator MacDonald, Dettwiler & Associates Corporation (MDA).

Quantitative performance evaluation is done on the Middlebury dataset with associated ground truth. Similar to [3, 102], we calculate three kinds of error statistics: errors for nonoccluded pixels, all pixels including occluded and pixels near discontinuities. Pixels are considered to be near discontinuities if they are not farther than 9 pixels apart from the 3-D boundary (disparity jump in ground truth). A pixel is considered to be erroneous when the absolute difference between its assigned and true disparities is more than some predefined threshold – similar to [3, 102], we use the threshold of 1.

To understand how the advances proposed in Chapter 2 impact the performance, they





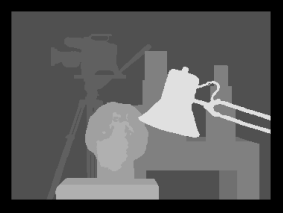

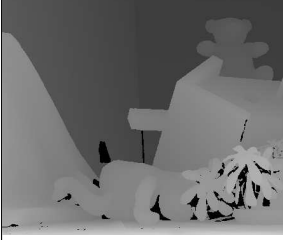


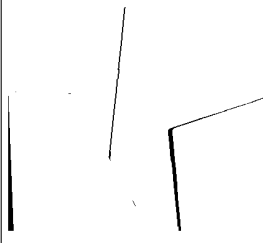


	<i>Tsukuba</i>	<i>Venus</i>	<i>Teddy</i>	<i>Cones</i>
Left				
GT				
GTocc				

Figure 3.1: Lab scene from the Middlebury Database [3]. From top to bottom: Left image, disparity ground truth, half-occlusion ground truth. Disparity and Occlusion Ground Truth are given with respect to left image. In Disparity GT, brighter pixels mean larger disparity; in occlusion GT, black pixels denote half-occlusions.





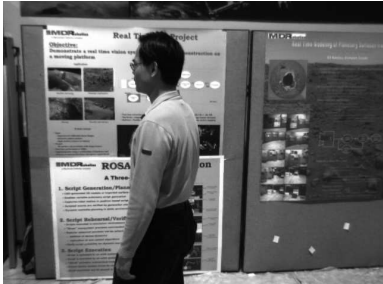
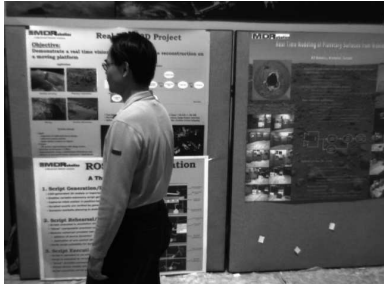
	Left image	Right image
FG		
Stephen1		
Stephen2		

Figure 3.2: Real Scenes: *Flower Garden (FG)* from Brown university [1]; *Stephen1* and *Stephen2* obtained from MDA.

Tag	Algorithm
A1	Single-scale 17x17 SW, SAD
A2	Single-scale, SAD
A3	Coarse-to-fine (CTF), SAD
A4	Adaptive CTF, SAD
A5	Adaptive CTF, SAD + occl
A6	A5 with colour cue
A7	[128] in CTF with 11x11 windows without multiple offsets
A8	[128] in CTF with 11x11 windows with multiple offsets, as in (2.38)
A9a	A5 with SSD
A9b	A5 with NCC
A9c	A5 with MI
A10	A5 with 7×7 aggregation window
A11	A5 with $\Delta d = \pm 2$
A12	A9b with colour cue

Table 3.1: Summary of Algorithms Compared Empirically.

have been added incrementally to standard CTF block matching. All algorithmic instantiations are outline in Table 3.1. Later, while proceeding with experimental evaluation, each algorithm will be described in more detail.

3.2 Adaptive coarse-to-fine processing

We evaluated the proposed adaptive CTF processing using the lab and real scenes outlined in Section 3.1. To see the effect of conventional CTF and adaptive CTF, we evaluate the performance of the following algorithms:

- A1 – single-scale block matcher **Module A** which operates on 17×17 shiftable [46] square windows and Laplacian-bandpassed images (level 0 of the Laplacian pyramid) with maximum disparity range for each test case.
- A2 – single-scale block matcher **Module A** which operates on 5×5 square windows and Laplacian-bandpassed images (as above) with maximum disparity range for each test case.
- A3 – coarse-to-fine block matcher **Module B** which operates on 5×5 square windows and Laplacian pyramid over all attainable levels (i.e. coarsest level auto-selected when one image dimension becomes unity) and search range ± 1 at each level.
- A4 – adaptive coarse-to-fine block matcher **Module C** which operates on 5×5 square windows and Laplacian pyramid over all attainable levels (i.e. coarsest level auto-selected when one image dimension becomes unity) and search range ± 1 at each level.

All algorithms use the SAD match measure.

The purpose of A1 is to exhibit the best performance of a standard single-scale block-matching technique with shifting, an analogue of the one used in Scharstein and Szeliski comparison [102]. The purpose of A2 is to show the effect of introducing CTF in A3. Finally, A4 embodies the CTF processing advance proposed in Section 2.2.3. Note that A1 is a degenerate version of A4 that operates on an image pyramid with a single level, maximum search range, but with bigger window size (the proposed adaptive CTF becomes ordinary shiftable windows when used over a single scale, because disparity offset is implicitly zero for each window configuration).

The qualitative and quantitative analysis with respect to Middlebury lab scenes is shown in Figures 3.3, 3.4. Comparing A2 and A3 (that use the same support window), CTF results in fewer errors overall (because coarser level matching is analogous to using a larger window at finer levels), but greater boundary error, as discussed in Section 2.2.2. When the proposed approach to adaptive CTF processing, **Module C**, is introduced (A4), considerable improvement is had over single scale (A2) and standard CTF (A3). It is expected that the adaptive approach bests standard CTF, – adaptive CTF significantly improves both the recovery of 3-D boundaries (as it was designed for exactly that purpose) and overall errors (as adaptive processing has some ability recover from errors made on the coarse level and not propagate/amplify them). It also is of interest to see that adaptive



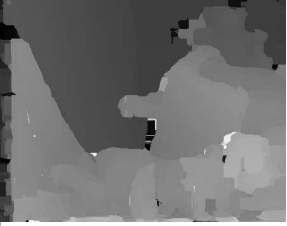
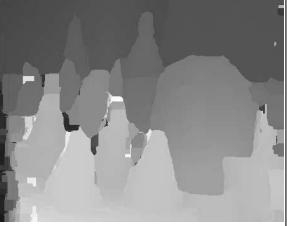
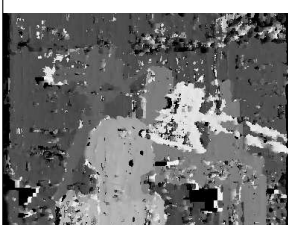
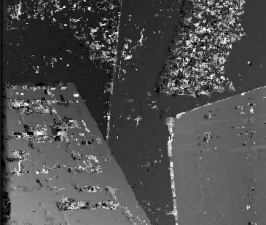


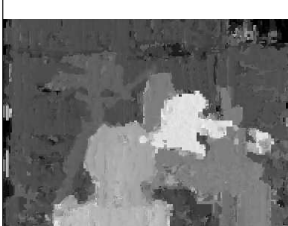

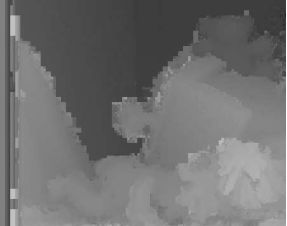
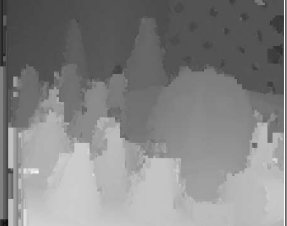

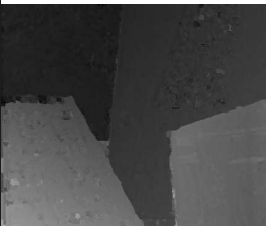
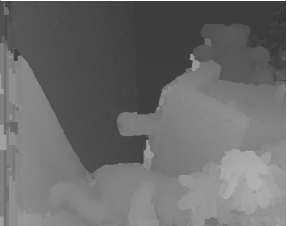
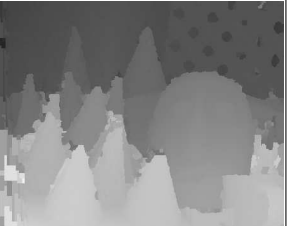
	<i>Tsukuba</i>	<i>Venus</i>	<i>Teddy</i>	<i>Cones</i>
A1				
A2				
A3				
A4				

Figure 3.3: Disparity Recovered for Middlebury scenes using algorithms A1-A4 from Table 3.1.

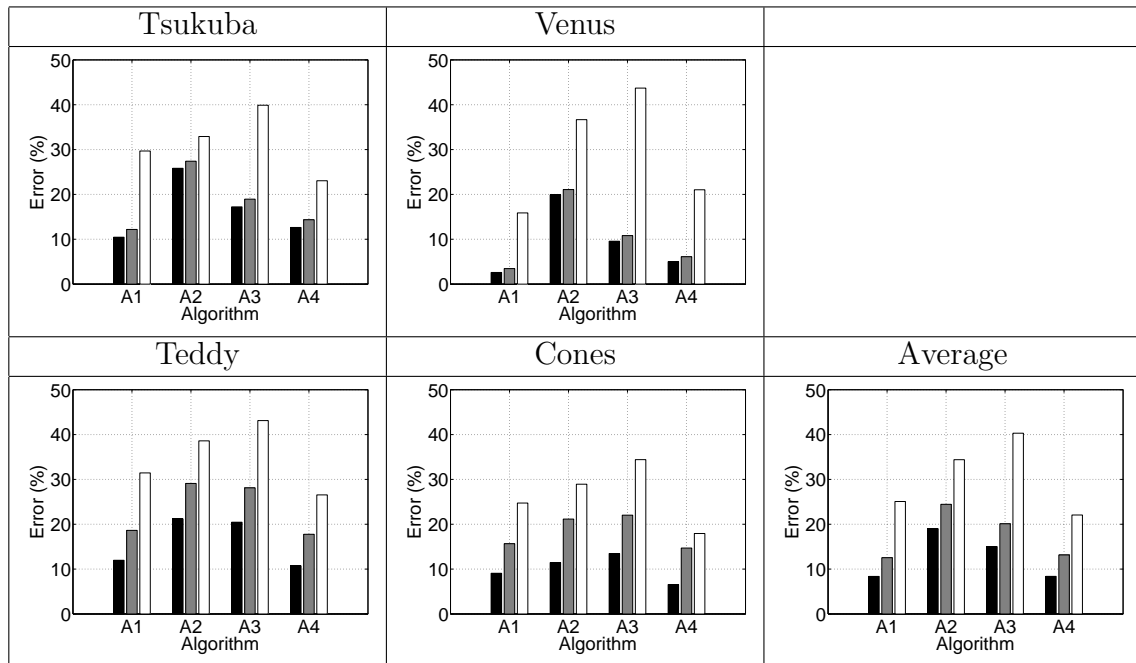


Figure 3.4: Error Statistics Across Algorithms A1-A4. Triplet bars represents error statistics for non-occluded (black), all (gray), and discontinuity (white) pixels, as defined in [3]. Algorithm indices are given in Table 3.1.

CTF bests single scale shiftable windows (A1), especially near discontinuities (white bars in Figure 3.4); this can be explained by the fact that A4 can use smaller windows (5×5 vs. 17×17) to yield more precise boundary-fitting and search over small ranges (i.e. ± 1 at each resolution) for less ambiguous matches.

From error statistics and visual inspection of the recovered disparity maps, adaptive CTF (A4) completely outperforms the standard CTF (A3) and single-scale matcher with fixed windows of the same size (A2), especially near 3-D discontinuities. The overall performance of adaptive CTF (A4) is better than adaptive single-scale matcher (A1) for more complex scenes *Cones* and *Teddy* scenes. For *Tsukuba* the results near 3-D boundaries are significantly improved; however, textureless regions are better handled by A1 as windows of bigger size (e.g. 17×17) aggregate more information to resolve ambiguity. A1 results for *Venus* are better than of A4 mainly for the same reasons – *Venus* has few very simple 3-D boundaries and many textureless regions, thus large support shiftable windows are able to reconstruct the disparity quite well. Along these lines, it is worth mentioning that out of the four Middlebury images, *Venus* is the worst representation of a naturalistic scene.

The results of algorithms A1-A4 from Table 3.1 run on naturalistic scenes are shown in Figure 3.5. While the ground truth and, hence, error statistics are not available for these images, some conclusions can be made based on visual inspection of the recovered disparity maps. All estimations were obtained using absolutely the same parameters as for the Middlebury test dataset.

Qualitatively, adaptive CTF (A4) yields better disparity maps overall. While single-scale matchers yield predominantly very noisy disparity maps (both A1 and A2), especially for *Stephen1* and *Stephen2*, CTF processing (A3 and A4) tends to produce smoother results; moreover, adaptive CTF (A4) produces much better 3-D boundaries as well. Note that use of adaptive windows (A1 and A4) helps to reduce significantly the foreground fattening/shrinkage effect, as predicted in Section 2.2.3. Moreover, adaptive CTF (A4) demonstrates the consistent ability to recover from errors made on coarse level, unlike standard CTF (A3). The latter can be concluded by noticing of removal of many gross disparity errors: right side of the tree trunk, bottom-left corner for *Flower Garden*, upper-left corner and left shoulder of *Steven1*, chest of *Steven2*.

Nevertheless, even the proposed Adaptive CTF (A4) cannot eliminate all disadvantages of CTF processing. For example, thin structures are still hard to recover precisely, e.g. arm lamps in *Tsukuba* and pencils in *Cones*; however, use of small windows results in better recovery of depth discontinuities that in non-CTF A1. Another apparent weakness of the CTF processing is the possible image border effect (disparity for the lower region on *Teddy* and region above the head in *Stephen1* are recovered incorrectly), when thin regions near image boundaries do not have enough spatial support and become lost at coarser scales.

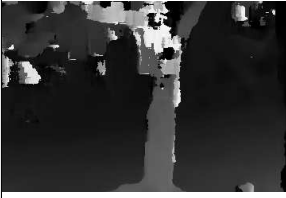











	<i>FG</i>	<i>Stephen1</i>	<i>Stephen2</i>
A1			
A2			
A3			
A4			

Figure 3.5: Disparity Recovered for real scenes using algorithms A1-A4 from Table 3.1.

3.3 Half-occlusions

Now we concentrate on the algorithm version that utilizes the half-occlusion processing proposed in Section 2.3.5:

- A5 – an extension of A4 that uses coarse-to-fine half-occlusion detection **Module D**. While doing CTF processing, A5 detects half occlusions and extrapolates disparity values from the back surface into the half-occluded regions, as described in Section 2.3.4. As we are determining disparity for the left image, filling from the left is done, because the background surface will be always to the left of the half-occlusion, as shown in Figures 2.11a and Figures 2.11b.

Results of running A4 and A5 on the Middlebury dataset are given in Figure 3.6. For comparison purposes, we have shown the case of detecting half-occlusions on the finest level only by post-processing the results of A4.

Introduction of half-occlusion processing further reduces errors, especially in occluded areas and near discontinuities (Figure 3.7 gray and white bars). Note that slanted surfaces are correctly recovered without false occlusions, as in *Teddy*. Occlusion detection results are isolated in Table 3.2, showing *Hit Rate (HR)* (percentage of pixels correctly labeled as half-occluded) and *False Positives (FP)* (percentage of pixels incorrectly labeled as half-occluded). Inspecting the qualitative results, we can conclude that the majority of half-occlusion false positives arise in the large textureless regions, where local methods (and CTF as well) are least robust. Note that the problem of textureless regions has not been explicitly addressed in this paper; we have just relied on CTF estimation for implicit improvement in such areas.

Finally, the results of Figure 3.6 show that occlusion detection in a CTF, i.e. cooperative, manner (A5) is more beneficial compared to simple post-processing of the finest level (A4). Qualitatively, this benefit is clear from the results of *Tsukuba* and *Teddy*, which show less hazy outlines of major half-occluded regions. Quantitative results for A4 post-processing and A5 are shown in Table 3.2, which exhibits consistently higher hit rate and consistently lower false positives rate for half-occlusion detection in A5.

Similarly, the same instantiations of A4 and A5 are run on our naturalistic dataset, the results of which are shown in Figure 3.8. As in the case with Middlebury, the major half occlusions are detected reliably, and CTF half-occlusion detection is superior to half-occlusion as post-processing. This is especially noticeable with images having large disparity jump discontinuities, e.g. *Stephen2*.

3.3.1 Comparison to previous approaches

As a comparison with the local half-occlusion detection methods, Table 3.2 shows results based on the often used left-right checking (LRC) [39, 54, 48] applied to A4 disparity. LRC

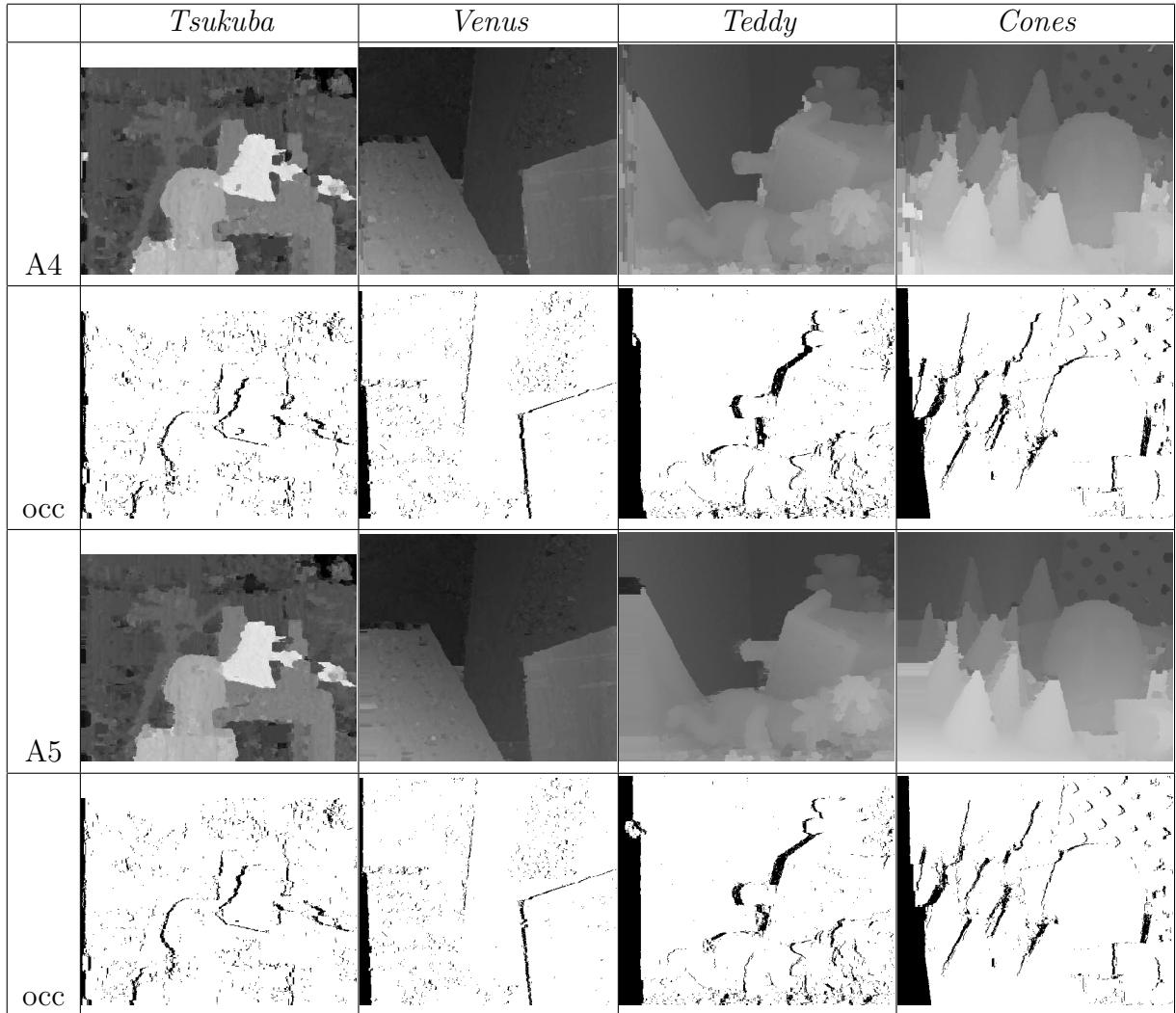


Figure 3.6: Disparity Recovered for Middlebury scenes using algorithms A4-A5 from Table 3.1 and the half-occluded regions. Black in half-occlusion maps denote half-occluded points. Half-occlusions detected in A5 are extrapolated by taking disparity value from the background (occluded) surface.

Algorithm		Tsukuba	Venus	Teddy	Cones	Average
A4	HR (%)	43.62	61.03	81.62	77.27	68.01
	FP (%)	3.20	1.83	2.92	2.76	2.63
A5	HR (%)	46.63	63.56	81.53	77.92	69.39
	FP (%)	2.31	1.27	2.27	2.21	1.99
LRC	HR (%)	59.87	70.3	87.71	82.82	76.65
	FP (%)	8.74	3.76	6.64	5.07	5.75

Table 3.2: Half-Occlusion Detection Statistics. From top to bottom: **Module D** applied as a postprocessing on result of A4; cooperative disparity and half-occlusion detection A5; Left-Right-Checking procedure applied on result of A4. Hit rate (HR) and false positive rate (FP) as percent of pixels correctly and incorrectly marked occluded.

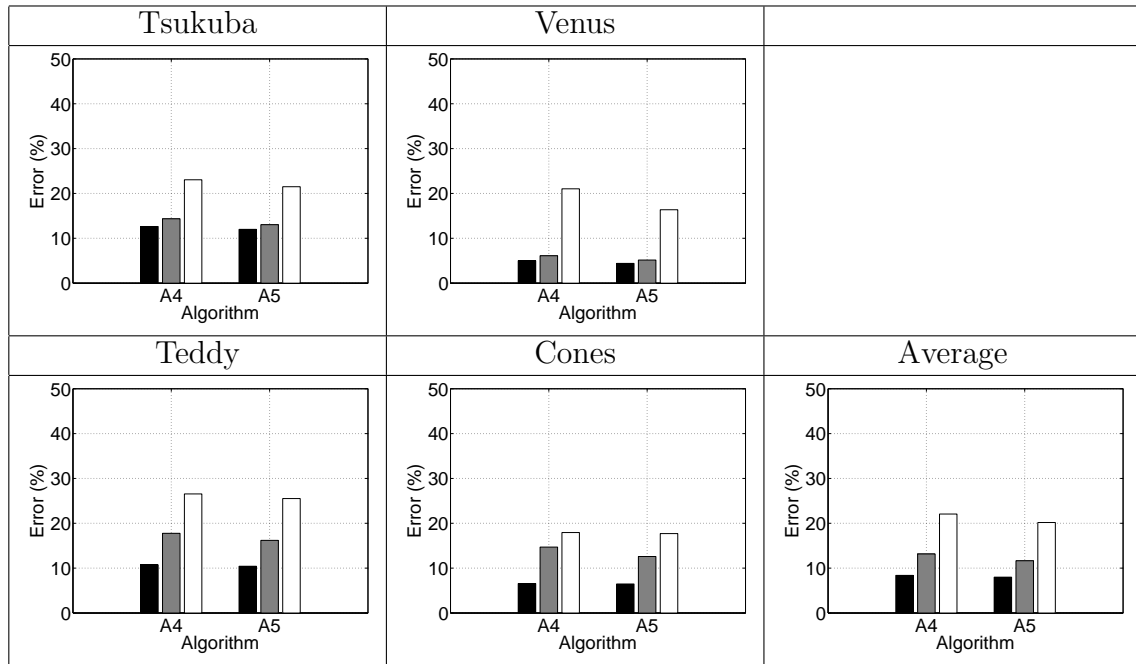


Figure 3.7: Error Statistics Across Algorithms A4-A5. Triplet bars represents error statistics for non-occluded (black), all (gray), and discontinuity (white) pixels, as defined in [3]. Algorithm indices are given in Table 3.1.

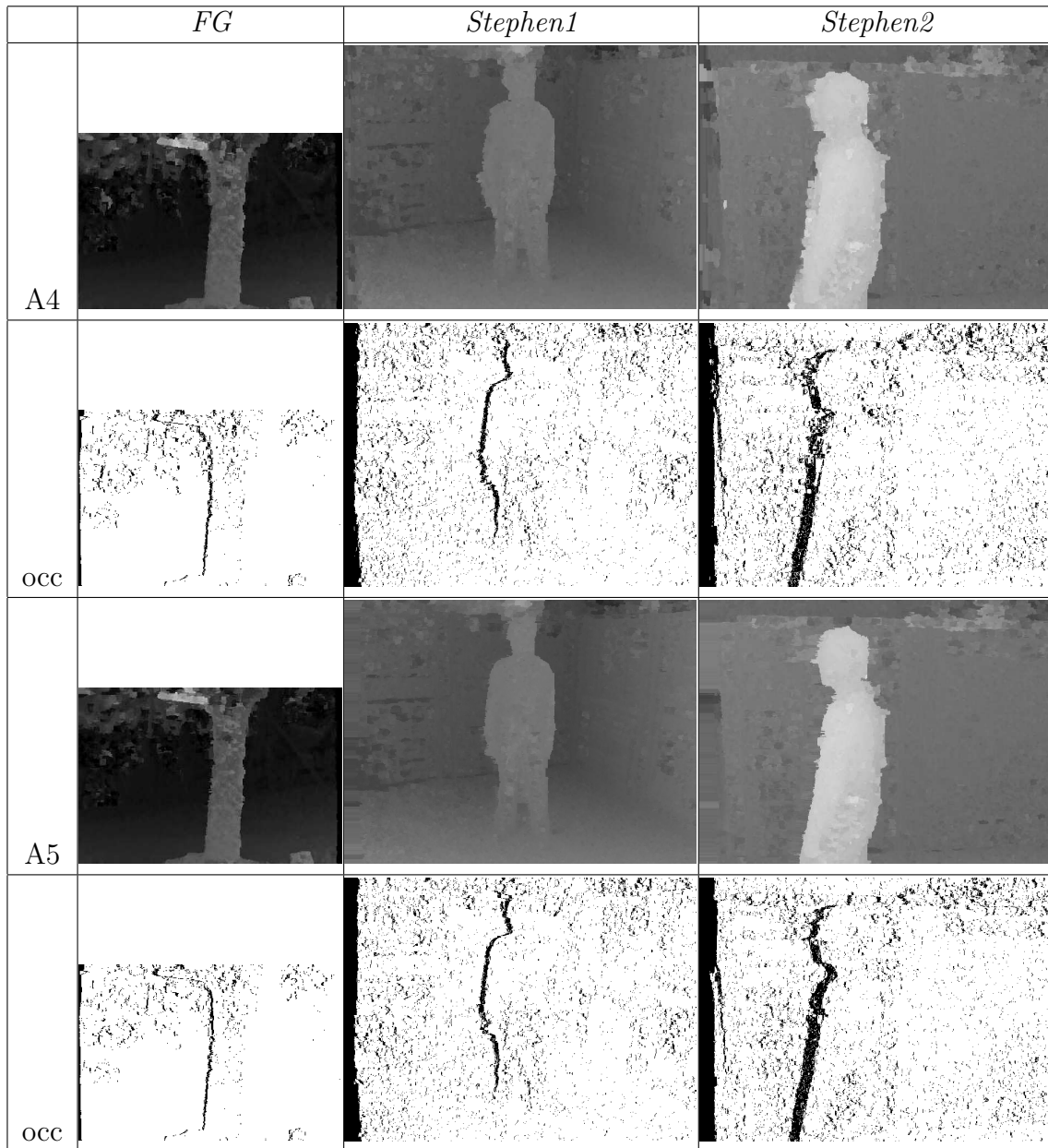


Figure 3.8: Disparity Recovered for real scenes using algorithms A4-A5 from Table 3.1 and the half-occluded regions. Black in half-occlusion maps denote half-occluded points. Half-occlusions detected in A5 are extrapolated by taking disparity value from the background (occluded) surface.

yields a 7% average hit rate increase and 2.5 times higher false positive rate, supporting the claim that **Module D** is more specific to half-occlusions, as discussed in Section 2.3.3, and therefore better suited when seeking to distinguish 3-D boundaries from other sources of match error. This distinction might be useful in some specific procedures, like segmentation of 3-D objects, as half-occlusions always arise near object boundaries.

A complimentary comparison between a half-occlusion detection method combining the same *disparity-change/width constraint* and *poor match score cue* (Section 2.3) and a wider variety of standard methods is presented in our earlier work [106].

3.4 Colour and intensity segmentation cues

Now we investigate the performance of the proposed use of colour (intensity in the case of grayscale images) segmentation in the window-based, coarse-to-fine matching procedure, **Module E**, as described in Section 2.4.1. Here we have one free parameter, λ , which controls the level of guidance for shiftable windows by colour and proximity cues. There are also α and β parameters that represent the mixture proportions between the power of similarity and proximity cues. We fix $\alpha = 7$ and $\beta = 36$ (same parameters as in [128]), which gave good results in our experiments. In the experiments of this section we will vary parameter λ only, in order to demonstrate the effect of the colour cue.

In this experiment, the following algorithmic instantiations have been evaluated:

- A5 – as described in previous sections. Results are shown with background surface disparities being interpolated into detected half-occluded regions.
- A6 – implementation of **Module E** together with CTF occlusion detection **Module D**; essentially, an extension of A5. The value $\lambda = 0.3$ is used for best performance, and results for the most representative three different values for λ ($\lambda = 0.1$; $\lambda = 0.3$ and $\lambda = 0.75$) are discussed.
- A7 – CTF implementation of segmentation-based windows [128] with window size 15×15 . A7 uses the nearest-neighbour upsampling procedure while going CTF (refer to Section 2.2.2 for more details).
- A8 – same as A7 but employs multiple offsets as described by (2.38) in Section 2.2.4. A7 and A8 comparison to previous colour segmentation-guided stereo.

Note that the actual window-based matching is still performed on the Laplacian pyramid constructed from grayscale images, so SAD is still “normalized”.

According to the qualitative and quantitative results on the Middlebury dataset shown in Figures 3.9, 3.11, we are able to see the consistent behaviour of the introduced colour segmentation cue. The best overall results are achieved when $\lambda \approx 0.3$ (A6) – all bigger and smaller values yield greater error. Note that A5 essentially corresponds to case when $\lambda = 0$.

Interestingly, if we visually inspect the disparity maps, we can notice the sharper boundaries (lamp arms in *Tsukuba*, tips of the cones in *Cones*, teddy’s head in *Teddy*, upper part of the leftmost plane in *Venus*) as well as occasional error introduced in some regions (upper part of rightmost plane in *Venus*, leftmost plane in *Teddy*), and these errors can grow quite large and yield inferior results when λ is too big. As an example, Figure 3.10 shows the results of CTF with colour-driven windows on *Teddy* for different values of λ , which apparent artifacts introduced by large values of λ . Thus, we have shown empirically that overreliance on the segmentation cue may be dangerous, confirming our concerns raised

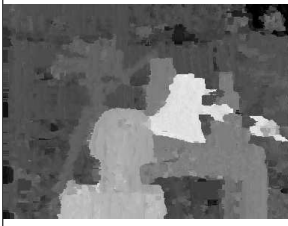


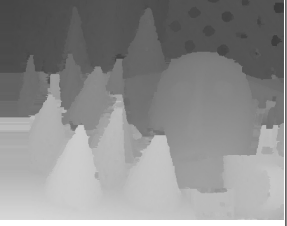




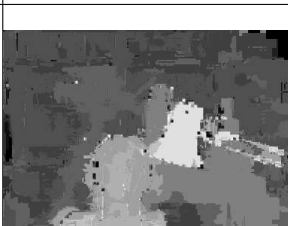
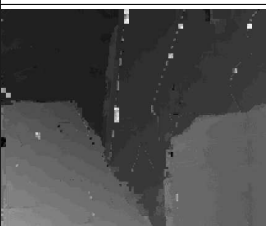
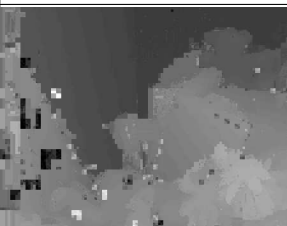
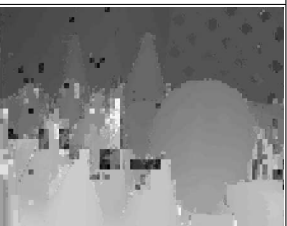

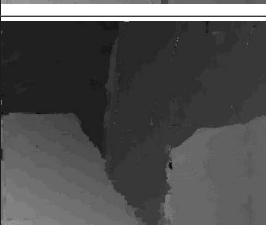
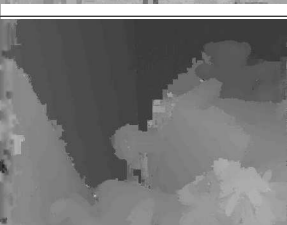

	<i>Tsukuba</i>	<i>Venus</i>	<i>Teddy</i>	<i>Cones</i>
A4				
A6				
A7				
A8				

Figure 3.9: Disparity Recovered for Middlebury scenes using algorithms A1-A4 from Table 3.1 and the half-occluded regions. Black in half-occlusion maps denote half-occluded points. Half-occlusions detected in A5 are extrapolated by taking disparity value from the background (occluded) surface.

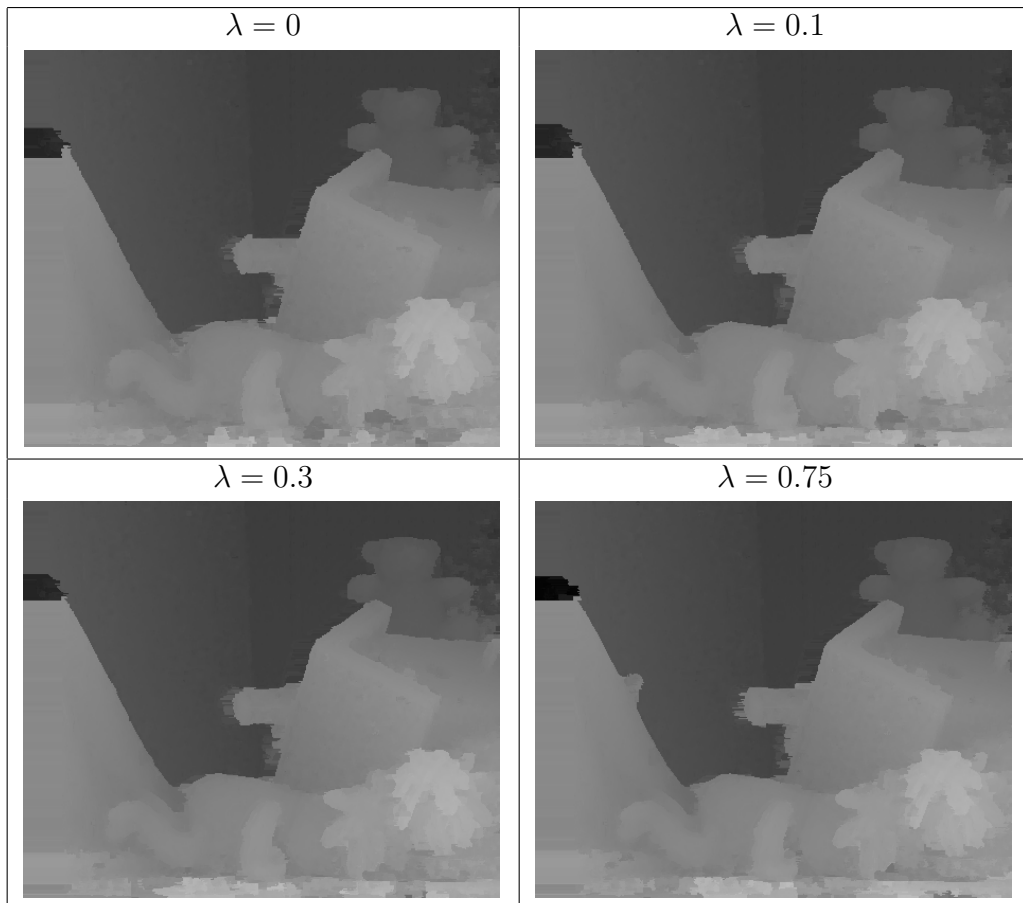


Figure 3.10: Disparity Recovered for *Teddy* using algorithm A6 from Table 3.1 with different parameter values.

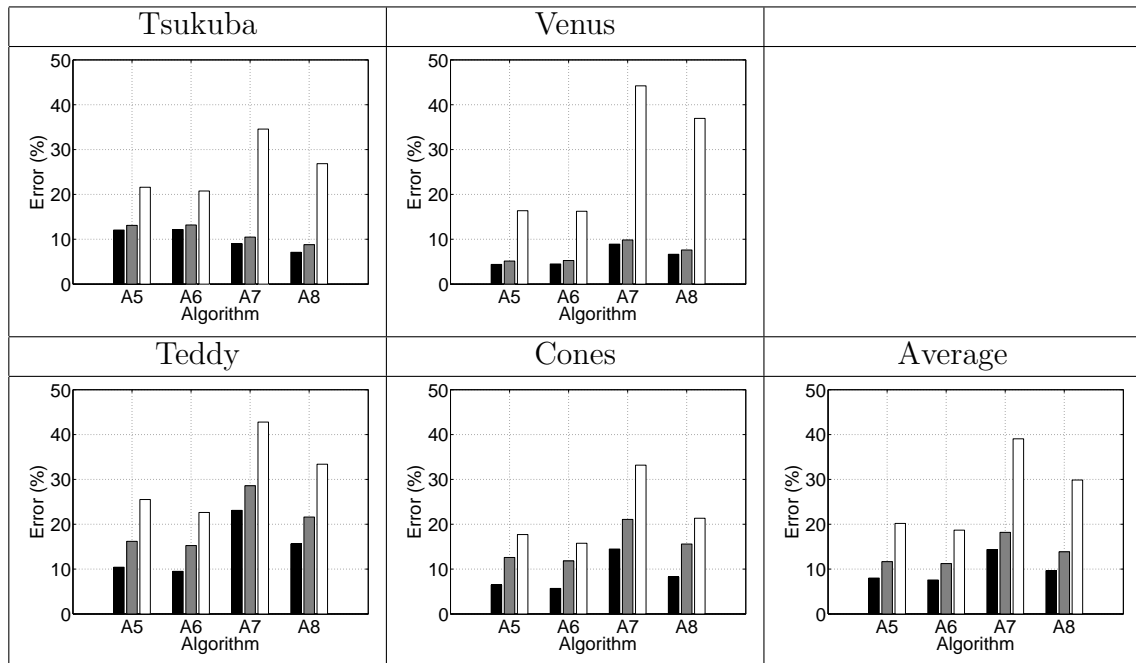


Figure 3.11: Error Statistics Across Algorithms A6-A7. Triplet bars represents error statistics for non-occluded (black), all (gray), and discontinuity (white) pixels, as defined in [3]. Algorithm indices are given in Table 3.1.

in Section 2.4: The ability to guide the best window should not only be based on colour, but also on match score, which provides resilience to occlusions and implicitly chooses the correct disparity offset.

With respect to our naturalistic images it is significant to note that only *Stephen1* and *Stephen2* are colour, while the others are grayscale. Results are shown in Figure 3.12. They are quite interesting in terms of no visible gain of colour/intensity segmentation cue. These results might be explained by the fact that the most useful information, especially in outdoor scenes, are coming from texture, rather than drastic change in intensity profile – colourful homogeneous objects are much more common in lab scenes, as exemplified in the Middlebury dataset.

3.4.1 Comparison to previous colour-cue formulations

To emphasize the necessity of being both robust and precise using adaptive windows guided by colour cue segmentation, we have implemented the coarse-to-fine version of Yoon and Kweon Gestalt-based stereo [128]. All parameters were the same as for A5, except window size, which was set to 15×15 ([128] requires big windows to operate reliably). We used an RGB colour space, not *CIELab* colour space as in [128], to be on a same foot with our colour-guided shiftable windows. Two versions are inspected – conventional CTF implementation (A7), and implementation with multiple offsets (A8), where multiple offsets are realized as described in Section 2.2.4.

According to quantitative and especially qualitative results depicted on Figures 3.9-3.12, the use of Gestalt windows (A7) did not improve the 3-D boundary estimation even in comparison to A5, which does not use any segmentation whatsoever. Importantly, use of multiple offsets (A8) improves results near 3-D boundaries, which again supports their necessity. However, the use of multiple offsets did not dramatically change the situation, especially in comparison to (A6). This confirms the previous concerns about the difficulty of precise segmentation at coarse levels. Another reason for the lack of benefit might be the necessity to search for better disparity offsets explicitly, and extra search, especially when performed in ambiguous cases, always has more chance to choose the wrong result. Recall that, on the other hand, the proposed CTF shiftable windows take care of variable offsets implicitly. Finally, it is worth mentioning that overall error for A8 is not greater (and even smaller in case of *Tsukuba*) than for A5, which could be attributed to the bigger aggregation window and, hence, better recovery of disparity in the textureless regions.

	<i>FG</i>	<i>Stephen1</i>	<i>Stephen2</i>
A5			
A6			
A7			
A8			

Figure 3.12: Disparity Recovered for real scenes using algorithms A1-A4 from Table 3.1 and the half-occluded regions. Black in half-occlusion maps denote half-occluded points. Half-occlusions detected in A5 are extrapolated by taking disparity value from the background (occluded) surface.

3.5 Other variations of stereo algorithm

In previous sections of Chapter 3 we have demonstrated how the incremental addition of the proposed enhancements has systematically improved the results of basic coarse-to-fine stereo processing. In doing so, we have used the SAD match measure, Laplacian pyramid with maximum number of levels, minimum disparity search range of ± 1 and window size of 5, all motivated in Chapter 2 from theoretical considerations. In this section we investigate the change in behaviour of the proposed algorithm by varying the underlying match measure, as well as by considering a different aggregation window and a broader local disparity search range.

The last portion of Table 3.1 summarizes the explored algorithms:

- A9a – version of A5 using SSD.
- A9b – version of A5 using Normalized Cross Correlation (NCC) and Gaussian pyramid (as match measure itself is normalized).
- A9c – version of A5 that uses Mutual Information (MI) a in coarse-to-fine fashion. We have used only 5 pyramid levels, not the maximum attainable, as MI estimation on very coarse images is very unreliable due to insufficient number of data points, i.e. pixels. We used base kernel of size $\sigma_1 = 32$ for Parzen window approximation. We used a Gaussian pyramid, as the MI-based measure determines the intensity mapping function; so, intensity values themselves are required. See Appendix A for discussion of MI-based matching.
- A10 – version of A5 with 7×7 windows.
- A11 – version of A5 with local disparity search range $\Delta d = \pm 2$.

Figures 3.13 show the results for the Middlebury dataset, while Figures 3.15 show results for the naturalistic dataset.

As expected, no superior results are observed with respect to which simple match measure is use SAD or SSD – as SAD is calculated faster, it remains a better choice. Surprisingly, NCC used in conjunction with Gaussian pyramids yielded noticeably better results for lab scenes, especially near discontinuities. In contrast, NCC showed no visible gain with respect to naturalistic scenes. Such behaviour could be attributed to the fact that lab images are of better quality (hence, the brightness constancy assumption is more reasonable) and by keeping more information of each level, i.e. using Gaussian instead of Laplacian pyramid¹, we increase the discriminatory power of our matching. In any case,

¹Gaussian pyramids contain lower resolution details in all images, while Laplacian pyramids contain the details from the restricted image frequency band.

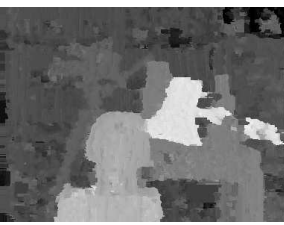

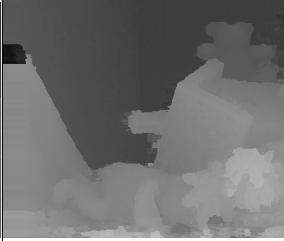
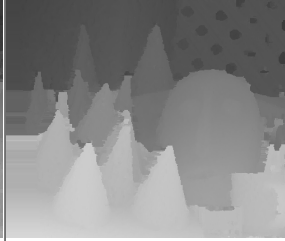



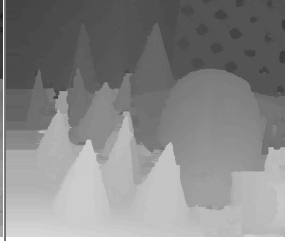



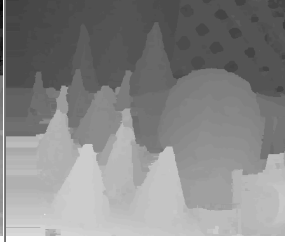



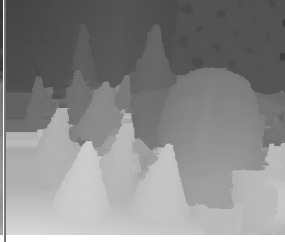



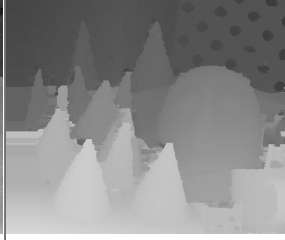
	<i>Tsukuba</i>	<i>Venus</i>	<i>Teddy</i>	<i>Cones</i>
A9a				
A9b				
A9c				
A10				
A11				

Figure 3.13: Disparity Recovered for Middlebury scenes using algorithms A9-A11 from Table 3.1.

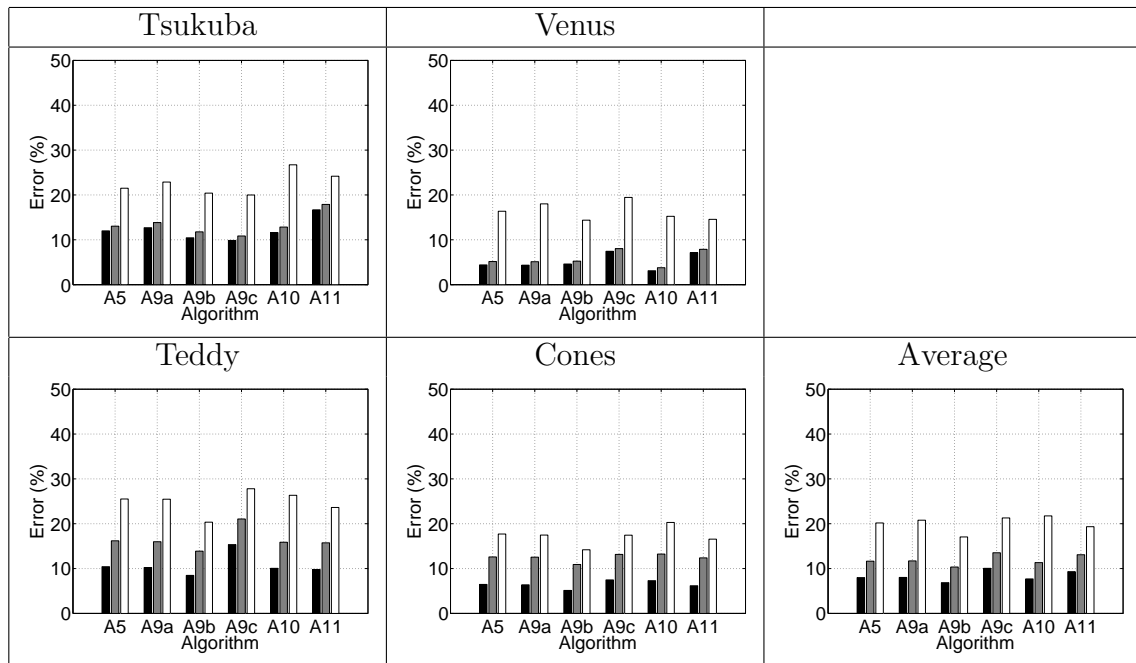


Figure 3.14: Error Statistics Across Algorithms. Triplet bars represents error statistics for non-occluded (black), all (gray), and discontinuity (white) pixels, as defined in [3]. Algorithm indices are given in Table 3.1.

NCC is more computationally intensive than SAD, so improvement of lab scene results under NCC is another example of the speed-accuracy tradeoff.

The use of bigger window size has predictable effects as well – the results for *Venus* have improved, as it has many homogeneous regions, while introducing slightly more errors near 3-D boundaries for other datasets, because the window structural element has grown in size. Use of bigger disparity search range slightly helped in disambiguation of very fine details, like pencils in *Cones* and the person’s outline in *Stephen2*, while introducing more gross errors, as in the background of *Tsukuba* and the upper left corner of *Stephen2*. Moreover, it takes almost twice more computation time. Importantly, the purpose of this experimental setup was to demonstrate the robustness of A5 to the choice of parameters – the initial choices guided by theoretical considerations seem to yield the best overall results.

The algorithm exploiting Mutual Information (A9c) deserves extra attention. For better discussion, Figure 3.16 shows the power of MI when one of the stereo images (left in our case) is distorted by a non-trivial transformation. On this basis, A9c is able to solve very hard cases when images are inverted in colour space, or a non-linear transformation is applied to their intensity values. However, when brightness constancy assumption (or normalization) is viable, MI consistently shows inferior results (Figures 3.13-3.14). This behaviour is consistent with previous research findings where superiority of MI in all cases was not achieved [64, 52]. These results can be explained as examples of the overfitting phenomenon. The recursive (or coarse-to-fine) procedure to estimate the one-to-one intensity mapping function is done using non-parametric techniques with use of little a priori information. In contrast, the brightness constancy assumption match measures (e.g. SSD) correspond to specific forms of this function (e.g. Gaussian cylinder running across the main diagonal in the case of SSD, as exemplified in Figure 3.16 first row, last column). Results for SAD (run on bandpassed Laplacian images) are better in cases when normalization works, as we force the intensity mapping function to be of a specific form.



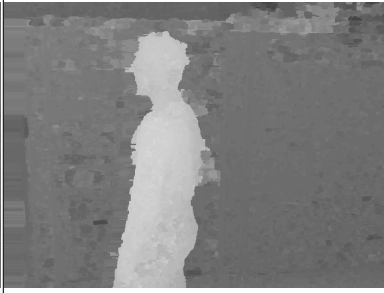
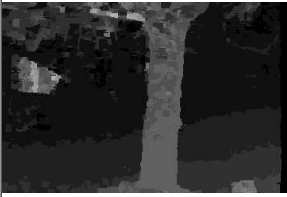

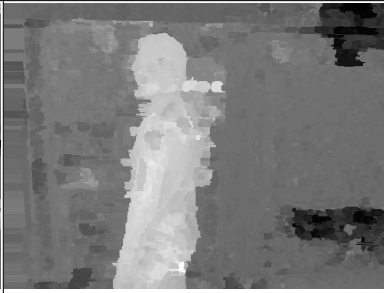



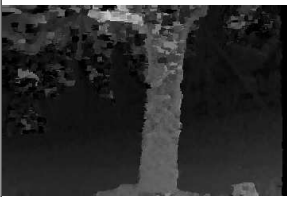

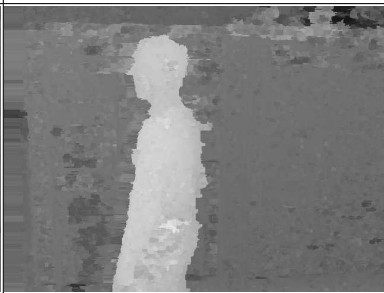
	<i>FG</i>	<i>Stephen1</i>	<i>Stephen2</i>
A9b			
A9c			
A10			
A11			

Figure 3.15: Disparity Recovered for real scenes using algorithms A9-A11 from Table 3.1 and the half-occluded regions. Black in half-occlusion maps denote half-occluded points. Half-occlusions detected in A5 are extrapolated by taking disparity value from the background (occluded) surface. Note: results of A9a (SSD) are very similar to A5 (SAD) and not shown here.

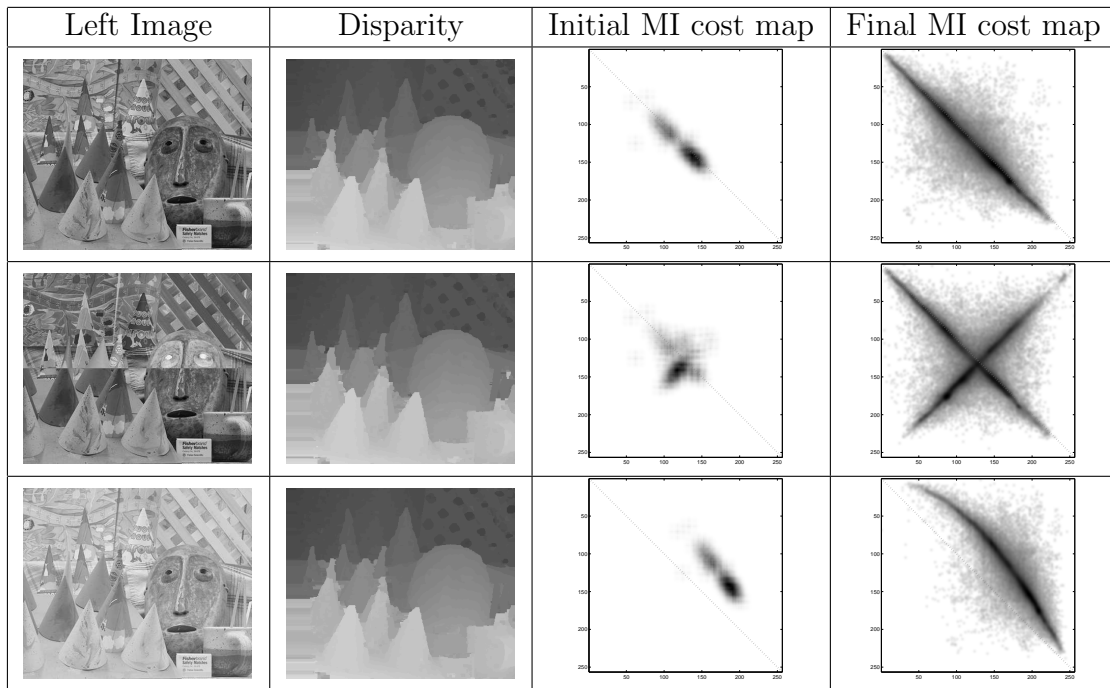


Figure 3.16: Demonstration of Mutual Information in Stereo Processing : *Cones*. Right stereo image stays unchanged, while left image has been synthetically modified to violate brightness constancy assumption. Third and fourth columns depict the intensity mapping function at the coarsest (after the first estimation) and the finest level (after the last estimation). If the brightness constancy assumption is true, then the function forms the ridge along the dashed line. Three various perturbations to the left image of the stereopair are investigated (from top to bottom): first row – original image; second row – upper half of the image is intensity inverted; third row – transformation $i = 255 * \sqrt{i/255}$ to every pixel's intensity.

3.6 Final comparison

We have extensively shown the performance and benefits of the proposed improvements to CTF stereo described in Chapter 2. According to experimental results of the previous sections, an algorithm with special potential to provide a strong speed/accuracy trade-off comes via adaptive coarse-to-fine stereo **Module C** using full Laplacian pyramid, local disparity search range ± 1 , window size 5, CTF occlusion detection **Module D**. It corresponds to A5 from Table 3.1. At increased computational cost, an NCC match measure operating on Gaussian pyramids, i.e. A9b, can be used for modest accuracy improvement. Finally, if we wish to get even better results at the expense of even longer processing time, the colour segmentation cue, A6b, can be incorporated.

Speed is an important advantage of any CTF algorithm including the proposed algorithm. For image and match window sizes $m \times n$ and w^2 , respectively, the theoretical complexity is $O(mndw^2) = O(mnw^2)$ (i.e. search range at each pyramid level, $d = 1$, for A3-A10 in all reported experiments), and can be decreased to $O(mn)$ via a running box filter implementation for window cost aggregation [109, 75]. The advances over standard CTF that are embodied in A5 do not degrade this complexity (e.g. implementation of shiftable windows as in [102] via morphological operation).

As a runtime example, it takes approximately 1 second to process *Teddy* with A5 as realized in unoptimized C without any special instructions (as outlined in Section 3.1) and (relatively) expensive $O(mnw^2)$ implementation on a 3.0 GHz P4. Since the developed approach is consistent with the CTF, block-matching framework, there is great potential for improved software runtimes and real-time performance, when mapped to appropriate processing architecture and/or hardware. Nevertheless, our work has been concentrated not on the fast implementation of the coarse-to-fine algorithm, but rather on analysis and improvement of its performance.

The implementation of segmentation-driven shiftable windows is considerably slower. In this paper we have explored a straight implementation of **Module E** for colour cue, which is very inefficient, as many redundant computations are performed. However, a very efficient formulation using distance transform is possible to speed the computations.

As we have extensively compared all algorithmic instantiations outlined in Table 3.1 using the Middlebury dataset, we can also compare our performance in accuracy with current state-of-the-art solutions. For this comparison we use a final variation of our work:

- A12 – the version of A9b augmented with colour segmentation cue ($\lambda = 0.0002$ being chosen to give the best overall results).

The results of running A12 on the Middlebury dataset are shown in Figure 3.18.

The snapshot of ranking of the proposed CTF processing scheme with colour segmentation, A12, on the Middlebury website [3] is depicted in Figure 3.17. A12 is currently

Error Threshold = 1		Sort by nonocc			Sort by all			Sort by disc					
Error Threshold... ▾		▾			▾			▾					
Algorithm	Avg.	<u>Tsukuba</u> ground truth			<u>Venus</u> ground truth			<u>Teddy</u> ground truth			<u>Cones</u> ground truth		
	Rank ▾	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc
AdaptingBP [17]	1.7	<u>1.11</u> 3	1.37 2	5.79 3	<u>0.10</u> 1	<u>0.21</u> 1	<u>1.44</u> 1	<u>4.22</u> 2	7.06 2	11.8 2	<u>2.48</u> 1	<u>7.92</u> 1	<u>7.32</u> 1
Double-BP [15]	1.8	0.88 1	1.29 1	4.76 1	<u>0.14</u> 2	0.60 3	2.00 2	<u>3.55</u> 1	8.71 3	9.70 1	<u>2.90</u> 2	9.24 3	7.80 2
Segm+visib [4]	4.2	<u>1.30</u> 6	<u>1.57</u> 3	6.92 7	<u>0.79</u> 6	1.06 4	6.76 7	<u>5.00</u> 3	6.54 1	12.3 3	<u>3.72</u> 4	8.62 2	10.2 5
SymBP+occ [7]	4.4	<u>0.97</u> 2	1.75 4	5.09 2	<u>0.16</u> 3	0.33 2	2.19 3	<u>6.47</u> 5	10.7 4	17.0 5	<u>4.79</u> 9	10.7 8	10.9 6
AdaptWeight [12]	5.8	<u>1.38</u> 8	1.85 5	6.90 6	<u>0.71</u> 4	1.19 5	6.13 5	<u>7.88</u> 7	13.3 6	18.6 9	<u>3.97</u> 6	9.79 5	8.26 3
SemiGlob [6]	7.3	<u>3.26</u> 13	3.96 11	12.8 16	<u>1.00</u> 6	1.57 6	11.3 12	<u>6.02</u> 4	12.2 5	16.3 4	<u>3.06</u> 3	9.75 4	8.90 4
Layered [5]	9.3	<u>1.57</u> 9	1.87 6	8.28 9	<u>1.34</u> 8	1.85 7	6.85 8	<u>8.64</u> 10	14.3 8	18.5 8	<u>6.59</u> 13	14.7 13	14.4 12
GC+occ [2]	9.3	<u>1.19</u> 4	2.01 8	6.24 4	<u>1.64</u> 11	2.19 10	6.75 6	<u>11.2</u> 13	17.4 13	19.8 11	<u>5.36</u> 11	12.4 11	13.0 10
MultiCamGC [3]	9.8	<u>1.27</u> 5	1.99 7	6.48 5	<u>2.79</u> 15	3.13 13	3.60 4	<u>12.0</u> 14	17.6 14	22.0 13	<u>4.89</u> 10	11.8 10	12.1 8
TensorVoting [9]	10.8	<u>3.79</u> 14	4.79 14	8.86 10	<u>1.23</u> 7	1.88 8	11.5 13	<u>9.76</u> 11	17.0 12	24.0 15	<u>4.38</u> 7	11.4 9	12.2 9
CostRelax [11]	11.5	<u>4.76</u> 16	6.08 16	20.3 18	<u>1.41</u> 10	2.48 11	18.5 16	<u>8.18</u> 9	15.9 10	23.8 14	<u>3.91</u> 5	10.2 6	11.8 7
RealTime-GPU [14]	11.6	<u>2.05</u> 12	4.22 13	10.6 13	<u>1.92</u> 13	2.98 12	20.3 17	<u>7.23</u> 6	14.4 9	17.6 6	<u>6.41</u> 12	13.7 12	16.5 14
YOUR METHOD	12.3	<u>5.86</u> 19	7.23 19	16.0 17	<u>4.11</u> 17	4.78 16	11.1 11	<u>8.03</u> 8	13.3 7	18.5 7	<u>4.74</u> 8	10.6 7	13.0 11
Reliably-DP [13]	13.0	<u>1.36</u> 7	3.39 10	7.25 8	<u>2.35</u> 14	3.48 15	12.2 15	<u>9.82</u> 12	16.9 11	19.5 10	<u>12.9</u> 19	19.9 19	19.7 16
TreeDP [8]	13.2	<u>1.99</u> 11	2.84 9	9.96 12	<u>1.41</u> 9	2.10 9	7.74 9	<u>15.9</u> 17	23.9 17	27.1 18	<u>10.0</u> 16	18.3 16	18.9 15
GC [1d]	13.7	<u>1.94</u> 10	4.12 12	9.39 11	<u>1.79</u> 12	3.44 14	8.75 10	<u>16.5</u> 18	25.0 19	24.9 16	<u>7.70</u> 14	18.2 15	15.3 13
DP [1b]	16.4	<u>4.12</u> 15	5.04 15	12.0 14	<u>10.1</u> 21	11.0 21	21.0 18	<u>14.0</u> 15	21.6 15	20.6 12	<u>10.5</u> 17	19.1 17	21.1 17
SSD+MF [1a]	17.7	<u>5.23</u> 18	7.07 17	24.1 19	<u>3.74</u> 16	5.16 17	11.9 14	<u>16.5</u> 19	24.8 18	32.9 19	<u>10.6</u> 18	19.8 18	26.3 19
STICA [16]	18.4	<u>7.70</u> 20	9.63 21	27.8 20	<u>8.19</u> 19	9.58 19	40.3 21	<u>15.8</u> 16	23.2 16	37.7 20	<u>9.80</u> 15	17.8 14	28.7 20
SO [1c]	18.9	<u>5.08</u> 17	7.22 18	12.2 15	<u>9.44</u> 20	10.9 20	21.9 19	<u>19.9</u> 21	28.2 21	26.3 17	<u>13.0</u> 20	22.8 21	22.3 18
Infection [10]	20.1	<u>7.95</u> 21	9.54 20	28.9 21	<u>4.41</u> 18	5.53 18	31.7 20	<u>17.7</u> 20	25.1 20	44.4 21	<u>14.3</u> 21	21.3 20	38.0 21

Figure 3.17: Snapshot from the Middlebury Comparison Table [3]. Proposed algorithm (A12) is labeled “YOUR METHOD”. Dated April 1, 2006

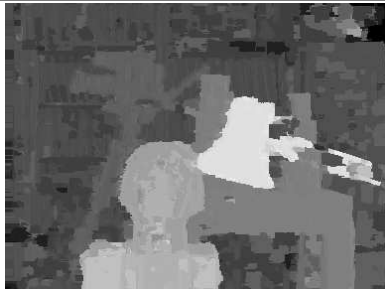


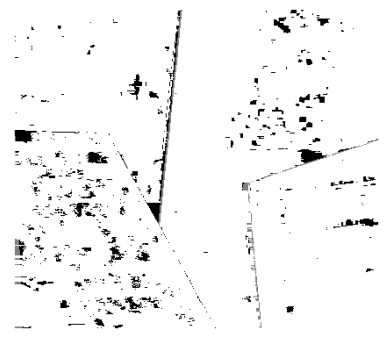
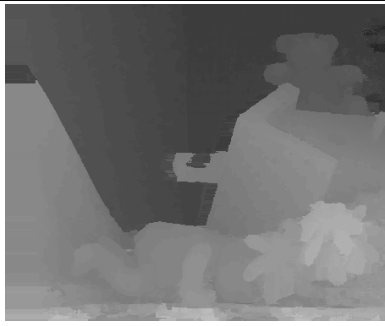
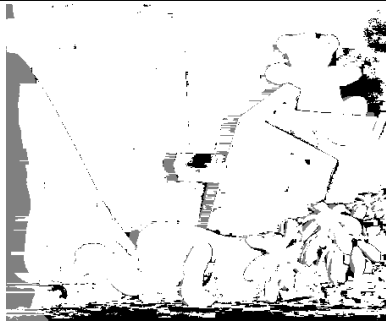

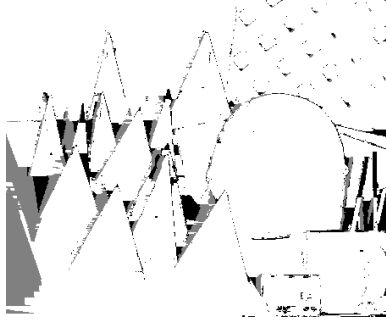
	Disparity	Error map
Tsukuba		
Venus		
Teddy		
Cones		

Figure 3.18: Disparity Maps of the Algorithm A12 (Table 3.1) for Middlebury Dataset. Disparity of the background surfaces are interpolated in the detected half-occluded regions. Left column: disparity maps; Right column: error maps. For error maps: black pixels denote pixels for which recovered disparity differs from ground truth disparity by more than 1. Gray pixels denote half-occluded pixels for which disparity has been inferred incorrectly.

ranked in 13th out of 21 places considering the four test sets overall and with error threshold set to 1. For particular test sets relative performance is improved – ranking improves to 7th for *Teddy* and *Cones*. The ranking for *Tsukuba* is 19 (but 17 near discontinuities), which exposes the relative weakness of the proposed algorithm (like any local algorithm) when operating in regions with little texture, as present in many areas of this data set. Nevertheless, introduction of the colour cue halved the errors for the *Tsukuba* scene. Another apparent weakness that is revealed with respect to *Tsukuba* and *Cones* is the lack of resolution for thin structures. Importantly, our Adaptive CTF formulation with occlusion detection outperforms single-scale shiftable windows algorithm, basic dynamic programming stereo algorithms, and is competitive to the basic graph cuts solution, which again proves the effectiveness of the combined best window and disparity offset search procedure and the necessity of proper half-occlusion handling.

It is important to inspect the actual error maps for the Middlebury lab scenes *Tsukuba*, *Venus*, *Teddy* and *Cones*, which are shown in Figure 3.18 for A12. Most of the errors are concentrated in the textureless regions, like *Tsukuba* and *Venus*, and thin structures, like lamp arms in *Tsukuba* and pencils and thin background patches in between cones in *Cones*. However, tips of the cones in *Cones* are recovered reliably, because they are extensions of bigger structures in the scene. Finally, the general 3-D boundaries are reliably recovered, which has been the main purpose of this paper. Moreover, the introduction of the colour cue (A12 vs. A9b) is mostly beneficial near the 3-D boundaries.

In any case, a critical comparison is that of adaptive CTF with occlusions A5 to standard CTF A3, as a major goal of the present work is improved disparity estimates for this style of efficient processing; such improvement is clearly demonstrated in Sections 3.2 and 3.3 on the set of lab and naturalistic images. The major gain of A5 with respect of A3 is the improvements near 3-D boundaries, which has been accompanied by significant reduction of overall errors as well. Analyzing the qualitative results for A3 and A5 presented in Figures 3.4 and 3.7, we observe the average reduction of errors by a factor of two. The major improvement in the disparity estimation per se comes with introduction of A4, while the upgrade to A5 also identifies half-occluded regions as such.

Chapter 4

Discussion: Relations to alternative disparity estimation frameworks

4.1 Speed-accuracy tradeoff

There are two major dimensions along which we can quantify an algorithm – its accuracy and the amount of work it performs. In the case of stereo, the first dimension can be the error percentage, as has already been used in most stereo evaluations [102, 114, 68], e.g. percentage of pixels in the image where recovered disparity value differs by more than 1 from the ground truth. For the second dimension, we chose computational and memory complexity. These complexity measures are independent of implementation details, which is fair given that the algorithms must be optimized differently. Moreover, they are designed and implemented by different people. Note that the complexity is not a sufficient measure of performance, as various computations can be run in parallel, which would significantly lower the final computational time – an issue which we will address later.

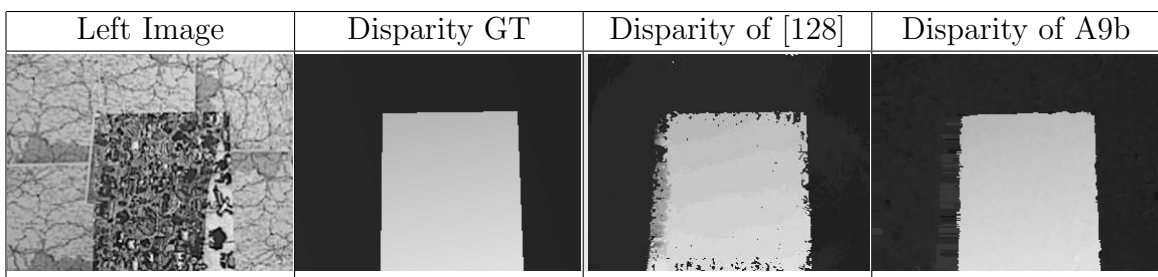


Figure 4.1: From left to right: Left image of *Map* dataset from the old version of [3]; Disparity ground truth; Disparity recovered by the adaptive weight approach [128] (adapted from [3]); Disparity map recovered by the proposed CTF algorithm A9b (Section 3.5).

While investigating the speed-accuracy tradeoff, we consider only formulations that perform pixel-based matching. Recently, methods that perform colour segmentation-based region matching [115, 16, 55, 125, 128, 65] became quite popular for their very good results of the lab scenes, like Middlebury dataset. However, such lab images are characterized by regions of highly contrasting colours; thus, strong performance of these algorithm on such data set owes much to colour segmentation apart from stereo matching per se. Indeed it is likely that most matches can benefit from consideration of colour segmentation. For example, benefits to adaptive CTF were demonstrated in Chapter 2.4 of this report. On the other hand, many natural image scenarios (e.g. outdoor terrain) may contain little in the way of colour differences, yet will be sufficiently textured to drive stereo matching. A laboratory example that is representative of such situation is the *Map* test pair that was available in an earlier version of the Middlebury test suite [3] and shown in Figure 4.1. Interestingly, algorithms that rely too heavily on colour segmentation perform poorly in such situations [110, 3] even though they are fundamental to multi-image matching.

More generally, for fair comparison with adaptive CTF we primarily consider other pixel-based (intensity, not colour) matchers. While some recent matchers employ various additional sources of information (e.g. colour) and/or postprocessing (e.g. plane-fitting), here we concentrate on methods that exploit similar image information to the proposed algorithm. We take the version A9b from Section 3.5 as a representative of the adaptive CTF stereo matcher described in Section 2.2 and abbreviate it **ACTF** for brevity.

4.1.1 Time complexity

Representative algorithms and their time complexities are outlined in Table 4.1. The complexity itself is expressed in n (number of pixels), d (disparity levels), k (number of iterations). Note that global message-passing optimization methods (DP, BP) use the Potts model as a smoothing prior (alternatively, linear or quadratic truncated cost functions), which allows for fast computation via a distance transform [40] and, hence, reduction of the complexity in comparison to the original formulations. Here we give more details about each framework:

- The proposed adaptive CTF algorithm (ACTF): Section 2.2 derived the complexity as $O(n)$.
- Conventional block-matching with shiftable windows (Block-SW) as in [102]: Its complexity is trivially $O(nd)$, because it makes a single pass over the whole disparity image space (DSI).
- Dynamic programming (DP) as in [102]: The naive implementation of DP is $O(nd^2)$ but only $O(nd)$ when the distance transform is used. Originally DP was organized along scanlines only and the ordering constraint was employed, while the recent

advance termed Semi-Global matching uses DP in numerous directions without assuming ordering [52, 53].

- Belief Propagation (BP) as in [116]: The original BP for stereo is $O(nd^2k)$ [111], but is reducible to $O(ndk)$ by application of a distance transform. We used the version of [116] since the typical performer as its energy formulation is the most closely related to widespread GC and DP.
- Graph Cuts (GC) as in [20]: The worst-case complexity of GC is quite bad, and depending on the algorithm can be, for example, $O(\text{Vertex} \times \text{Edges}^2) = O(nd(nd)^2) = O(n^3d^3)$ if push-relabel maximum flow algorithm is adopted [18]. In turn, the average complexity is rather hard to predict, so we take the only reported in the computer vision literature average complexity of the Roy and Cox GC formulation [98] as $O(n^{1.2}d^{1.3})$ [21]. Nevertheless, it is worth saying that GC developed by [20] and further enhanced and tested in [18, 114] (more specifically, expansion-move and swap-move algorithms) are quite fast in practice, and significantly more efficient than non-hierarchical versions of BP [114].
- Graph Cuts with occlusions (GCoc) as in [69]: We explicitly consider the version with occlusions as this formulation is more appropriate for binocular stereo and considers more highly-connected graphs [68]. The complexity of GCoc is slightly higher than of GC as a graph with more connections has to be solved. More specifically, we consider the worst-case complexity as $O(\text{Vertex} \times \text{Edges}^2) = O(nd(nd)^2) = O(n^3d^5)$ and expected as $O(n^{1.2}d^{1.3\frac{5}{3}}) = O(n^{1.2}d^{2.2})$. However, note that the same graph can be solved by message-passing optimization techniques, like BP, in only $O(ndk)$ complexity, not $O(nd^2k)$ as can be implied by construction, though yielding inferior results [68].

Currently, we do not include the newly-proposed tree-reweighted message passing scheme [66] here as it is not yet widely used, its performance is rather similar to GC and implementations are slower.

In order to be able to rank the algorithms based on complexity, it is desirable to express the complexity measurement in terms of a single variable, horizontal image size N . To do it, we have to recast $O(n)$, $O(d)$ and $O(k)$ in terms of $O(N)$.

First, we assume that our images have a standard height/width ratio, and the width should be of the same order N as the height, which would make the number of image pixels $n = O(N^2)$.

The number of possible disparity values d is smaller than the horizontal image size, but it is clear that it is proportional to the size of the image and we assume that it depends linearly on N . Thus, $d = O(N)$.

Algorithm	Complexity (reported)	Complexity (adapted)
ACTF	$O(n)$	$O(N^2)$
Block-SW	$O(nd)$ [102]	$O(N^3)$
DP	$O(nd)$ [53]	$O(N^3)$
BP	$O(ndk)$ [41]	$O(N^3) : O(N^4)$
GC	$O(n^{1.2}d^{1.3}) : O(n^3d^3)$ [21]	$O(N^{3.7}) : O(N^9)$
GCocc	$O(n^{1.2}d^{2.2}) : O(n^3d^5)$	$O(N^{4.6}) : O(N^{11})$

Table 4.1: Major Stereo Algorithms and Their Complexity. Note that complexity for belief propagation, and dynamic programming are taking under assumption that a distance transform is used in the computations.

Finally, belief propagation is an iterative algorithm¹ and the number of iterations k depend on the nature of the scene. It can be few iterations when the environment is highly textured, or on the order of $O(N)$, if there are large textureless regions and the information from a structure regions has to be propagated over large image areas. Moreover, the iteration component is highly dependent on the message update schedule. For example, the efficient hierarchical belief propagation presented in [41] can be considered as constant overhead, independent on image size. Thus, the number of iterations $k \in [O(1), O(N)]$, i.e. it may introduce a constant overhead or require a lot of computations to converge to a reasonable estimate.

The adapted complexity functions \mathcal{C} in terms of argument N are shown in the third column of Table 4.1. Figure 4.2 shows the plot of complexity versus performance for the major computational algorithms. The performance is measured in terms of error percentage, and we have used the numbers reported by authors directly either in the corresponding papers or in the benchmark website [3]. Note that abscissa has the logarithmic scale. Algorithms which are closer to the origin are desirable as they provide accurate results at a reasonable time (low error rates with low computational complexity).

Analyzing Figure 4.2, all standard algorithms (marked with blue squares) exhibit a consistent tendency of better performance (lower error rates) at the expense of higher computational complexity. The proposed ACTF (marked with green asterisk) lies to the lower left side of the cloud of standard algorithms, which signals its good combination of low error rate and very low complexity. For completeness, we also show one of the best stereo solutions that are based on DP [52] and BP [110] (marked with red squares). While their error rates are significantly lower than of original formulations, complexity is still at

¹Graph cuts for stereo is iterative too, but the number of iterations is few (only two or three iterations is enough [20, 69]). The variable complexity of the algorithm is attributed to the Maximum-Flow algorithm itself, as its execution time highly depends on data.

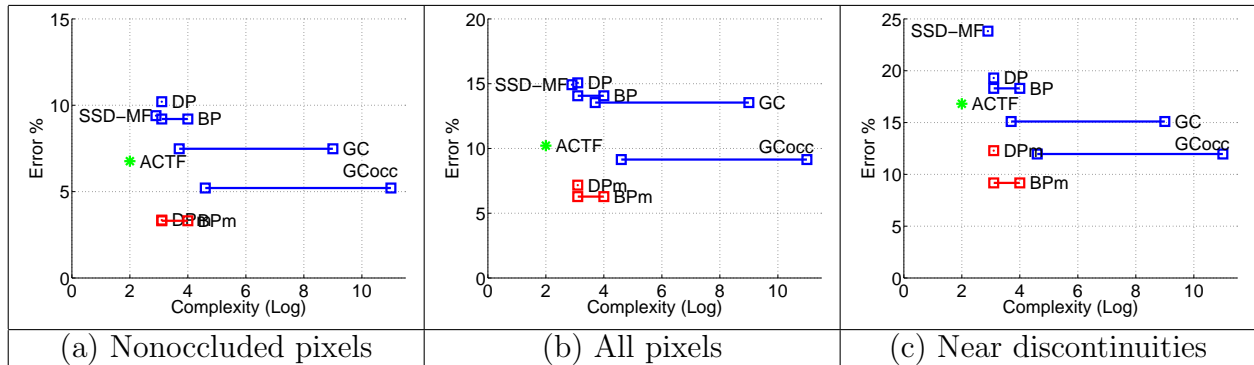


Figure 4.2: Complexity-Accuracy Tradeoff of Major Dense Stereo Algorithm. Plot shows the order of complexity ($\log \mathcal{C}$) on abscissa versus percentage of erroneous pixels on the ordinate for three classes of errors: (a) Nonoccluded pixels (b) All, including occluded (c) Pixels near discontinuities. The proposed algorithm (ACTF) is marked with green asterisks and lies to the lower left side of the region formed by other basic algorithms (blue squares), which signals a very good speed-accuracy tradeoff characteristics of ACTF. The performance of the best stereo algorithm that are based on DP and BP are shown in red.

least as high and running times are slower, because more computation is to be performed.

4.1.2 Memory Complexity

In addition to computational complexity, a few words should be said about memory complexity. All single-scale algorithms, including the ones shown in the plot of Figure 4.2, require the construction of the whole disparity image space (DSI) [102], which means that they must have at least $O(nd) = O(N^3)$ memory complexity to store the DSI and operate on it². On the contrary, the pyramid-based CTF approach, as the one presented here, requires only $O(n) = O(N^2)$ space, which results from the fact that CTF does not construct the complete DSI. Recently, some interest has been expressed in non-CTF methods that do not explicitly estimate the whole DSI, because the exact determination of disparities per se might not be so important for some applications as opposed to efficiently segmenting the scene into coarse depth layers [67, 31, 4].

4.2 Parallelization

A question of computational complexity and performance is usually complemented by the ability of computations to run in parallel. This property results in much more efficient

²Note that the naive implementation of the block-based matching algorithm can be only $O(n)$ in space, though at the expense of numerous redundant computations.

utilization of hardware capabilities and ultimately allows faster and, in many cases, real-time processing.

Parallelization is always useful in the realm of special purpose hardware, where a custom processing chip can be designed for cameras of certain resolution and guarantee real-time processing. Recently, a research effort has been directed toward performing stereo processing on commodity graphics hardware [127, 29, 124, 126] and FPGA [33, 84].

The ability to make the algorithm run efficiently in parallel depends on the nature of computations. Ideally, the computations require only local access to data so that they can be correspondingly parallel to individual processors. For stereo, block-based matching is readily parallelized as it is truly local, and the presented CTF block-based matching naturally possess this property too. The complication of the coarse-to-fine scheme is that it is sequential in scale processing. Nevertheless, the parallelization is very efficient because number of scales is logarithmic with respect to the image size. Additionally, the pipeline architectures [118] are more plausible for CTF processing, and existing systems already provide real-time performance.

In contrast, the DP approaches, though possessing the same theoretical complexity as block matchers, are parallelizable up to a scanline (or corresponding assumed Markov Chain). As an example for BP approaches, the messages in a single iteration can be computed in parallel (the message computation is a local operation), but they depend on the previous iteration. Nevertheless, authors of [124, 126] demonstrated that both DP and BP can yield real-time performance using graphics hardware (together with CPU processing in parallel), albeit on rather small images and coarsely quantized disparity (320x256 with 16 disparity levels).

4.3 Anytime computation

Coarse-to-fine processing uses lower resolution disparity obtained in the previous iteration to compute the final result via refinement. Interestingly, if the refinement step is dropped, the lower resolution result is directly available. That simple fact makes the coarse-to-fine algorithm an anytime algorithm. This property is critically important in hard real-time systems, when some solution must be available in the middle of computation [99]. In the case of stereo, which typically uses factor of two pyramids, the time required to obtain a low resolution solution is four times less than to refine it, which means that an initial coarse approximation to the disparity map can be obtained very fast.

All iterative algorithms, including GC, BP, PDE can also be perceived as anytime algorithms to some extent, because they can be stopped during the processing and solution can be extracted; moreover, the solution is getting progressively better with time. Note, however, that in most cases the quality of the intermediate solution is hard to assess meaningfully – instead of smaller resolution when all points have stable results, as in CTF

computation, intermediate iterative solutions usually consist of regions of correctly and incorrectly inferred measurements, with no certain indicator which is which. Luckily, any of these algorithms can be brought into CTF to exemplify the anytime property, and there already exist multiresolution implementations of BP, DP, anisotropic diffusion and others [74, 49, 87, 109, 6, 44, 41].

4.4 Additional considerations for practical stereo vision

Scharstein and Szeliski [102] made a substantial contribution in organizing a test bed for stereo algorithms and their evaluation. Their comparative study gives a clear idea about accuracy and performance of major stereo frameworks. Nevertheless, several questions with respect to utilization of the algorithm in real situations require further investigation.

4.4.1 Parameter tuning

Most recent algorithms employ rather complex models of disparity maps with occlusions, colour segmentation, plane fitting etc. All these require the introduction of various parameters the majority of which are arbitrary and hand-tuned³. Even basic global formulations require certain vital parameters to be specified: smoothness cost for prior, parameter value for robust datacost (e.g. threshold for truncated basic match measures) and occlusion cost, if there are occlusions in the formulation. Moreover, the same parameter values typically are incapable of producing uniformly superior solutions for all datasets. The sensitivity to choice of parameter values can become a significant obstacle to bringing the algorithm from the lab to the real world.

In contrast, the proposed ACTF has virtually no parameters to tune. Window size, which is typically the major and critical choice for stereo algorithms, is kept small (5x5) to allow precise boundary localization, while greater support aggregation is available by using coarser resolutions. Moreover, this configuration was able to produce the best results both for lab and real scenes, datasets which are very different in nature.

4.4.2 Sensitivity to noise

Interestingly, this issue has been more rigorously addressed in the optical flow literature (e.g. [11, 22]). The overall conclusion was that local aggregation methods like Lucas/Kanade [81] are more noise-resilient than purely-global formulation like Horn/Schunk

³As an example, one of the state-of-the-art stere systems by Sun et al. [110] has 5 free parameters without colour segmentation + 1 when segmentation is used, not including the free parameters needed to obtain the segmentation itself.

[56]. Thus, in addition to global regularization, local aggregation is desirable for more robust performance. Additionally, the use of normalized match measures also requires some support region around a point. Finally, the local aggregation introduces errors in discontinuity localization; for this reason it is tried to be avoided in the global formulation on the first place.

From this perspective, the proposed adaptive CTF computation is completely justified, as it allows for accurate computation of disparity information near discontinuities while employing local aggregation. Global regularization very desirable to be added on top, but without removing adaptive local aggregation step, which has an additional important feature of correctly treating coarse disparity offset, i.e. removing upsampling uncertainty near 3D discontinuities.

Chapter 5

Conclusion

5.1 Adaptive coarse-to-fine stereo

This paper has presented extensive analysis of CTF stereo processing with specific emphasis on block based matching. As the result of this analysis, the main sources of error are identified – a well known foreground fattening/shrinking artifact of block-based matching [54, 105], and the necessity of multiple offsets that has not been given enough attention by the stereo community. A simple combination of adaptive windows and adaptive offset is put forth, which has empirically shown significant improvement over standard CTF block matching stereo and single-scale stereo matching with shiftable windows [102]. Moreover, we summarize various alternatives of how to introduce multiple offsets in non-block-based stereo algorithms. While the proposed CTF shiftable-windows approach is simple and straightforward, no previous explicit mentioning of such an algorithm has appeared in the literature. Moreover, the analysis, explanation and comprehensive evaluation of adaptive CTF has not been presented previously. The results document significant advantage in using the proposed methods for improved CTF disparity estimation in the vicinity of 3-D boundaries.

The disparity produced by the algorithm can be exploited directly, or can be used as a reliable initial estimate for any global optimization procedure to improve results even further. Also, the algorithm is directly extendible to the computation of optical flow (it would extend [7] in this case), where disparity is a two-dimensional vector.

Our thorough investigation of CTF stereo processing fills a gap in recent stereo research. Recently, much effort has been applied to improve local and global algorithms, especially near 3-D boundaries, [54, 121, 21, 110, 35], but CTF stereo, while widely-used in practice, receives relatively little attention. This paper helps to fill the gap in stereo research by providing a better understanding of CTF stereo processing power and showing its competitiveness in accuracy to many state-of-the-art solutions, together with its com-

putational and storage requirements superiority. The idea of CTF refinement will always be among the most useful tools of computer vision algorithm designers, because even as computer processing power increases, the volume of data and search space increases as well. Therefore, CTF processing is likely to remain one of the best alternatives to get at least some solution, when computational abilities are limited. A better understanding of limitations as well as fundamental improvements of CTF stereo will only make this tool more useful.

5.2 Binocular half-occlusions

As the half-occlusion phenomenon is one of the toughest source of errors for computational stereo, we thoroughly investigate it and complement the knowledge gained from previous computational investigations [39, 90, 110, 35, 106]. We explicitly formulate the basics of half-occlusion formation in relation to the binocular forbidden zone [71], which has not been done before. Further, we distill the geometry and match cues, which have been used by previous algorithms [39], reveal their complementarity, and combine them in an efficient algorithm for half-occlusion detection. Importantly, the proposed half-occlusion processing is formulated for CTF block matching algorithms to yield cooperative coarse-to-fine processing of disparity and half-occlusions. The benefit of this approach is documented extensively.

5.3 Colour and intensity cues

Monocular colour and intensity cues are quite heavily used by recent state-of-the-art computational stereo formulations [16, 110, 35, 128]. We discuss the application of intensity segmentation in local block matching stereo from the robust statistics point of view. We came to the conclusion that both robustness and precision must be retained during CTF processing and augment the adaptive stereo processing of Section 2.2 with an intensity similarity cue. Experimental results in Section 3.4 show that our approach is more appropriate for CTF local stereo processing, than the ones suggested previously, e.g. [128].

Appendix A

Mutual Information (MI) for stereo correspondence

In the realm of stereo matching, the normalized match measures are robust to overall intensity difference in both images (bias) and local difference in luminance variance (gain). But what can be done when the settings are totally obscured, or, even worse, data to match comes from different modalities?

Historically, the ability to match images obtained from different modalities is motivated by the registration of the medical images that depict different data, e.g. PET and fMRI. Viola and Wells III [123] and Collignon et al. [28] independently proposed solutions that used mutual information (MI) for registration. Their results far surpassed the correlation-based registration of multimodal images and, in fact, very closely approached the ground truth.

In stereo, MI matching can become useful when data quality is bad, camera settings are very different, or in exceptional stereo situations when the robot head is equipped with usual and infrared cameras; then an extra-verification step can be performed by calculating the depth using the usual and infrared cameras. Recently, a number of stereo algorithms which use mutual information appeared [38, 43, 64, 52].

The aim of MI approaches is to reconstruct a general one-to-one mapping function of intensity values in one image to the intensity values in the other using non-parametric technique like histograms, or, more generally, Parzen windows [37].

Mutual information is defined from the entropy of two images and their joint entropy.

$$MI_{im_1, im_2} = H_{im_1} + H_{im_2} - H_{im_1, im_2}, \text{ where} \quad (\text{A.1})$$

$$H_{im} = - \int_0^1 P_{im}(i) \log P_{im}(i) di \quad (\text{A.2})$$

$$H_{im_1, im_2} = - \int_0^1 \int_0^1 P_{im_1, im_2}(i_1, i_2) \log P_{im_1, im_2}(i_1, i_2) di_1 di_2 \quad (\text{A.3})$$

Assuming pixel independence, the joint entropy of two images can be calculated as a sum of joint entropies between pixels in correspondence.

$$H_{im_1, im_2} = \sum_p h_{im_1, im_2}(I_{im_1p}, I_{im_2p}) \quad (\text{A.4})$$

Kim et al. [64] transformed the calculation of joint entropy (A.4) into the sum of two data terms using a Taylor series expansion. Algorithmically, (A.4) is calculated directly from the probability distribution of intensities P_{im_1, im_2} that has been obtained by Parzen windows with a Gaussian kernel:

$$h_{im_1, im_2}(i, k) = -\frac{1}{n} \log (P_{im_1, im_2}(i, k) \otimes g(i, k) \otimes g(i, k)) \quad (\text{A.5})$$

where $g(i, k)$ is a Gaussian kernel used by Parzen windows and n is the number of non-occluded pixels. Note that only non-occluded pixels must be used here, as occluded pixels have no match and cannot contribute to the estimation of the matching function.

Once the distribution for $P_{im_1, im_2}(i, k)$ has been obtained (and stored in the non-parametric Parzen window form), single entropies H_{im_1} and H_{im_2} can be calculated by marginalization, i.e. $P_{im_1} = \sum_k P_{im_1, im_2}(i, k)$, and manipulations similar to (A.4) and (A.5).

The resulting working definition of Mutual Information is

$$mi_{im_1, im_2}(i, k) = h_{im_1}(i) + h_{im_2}(k) - h_{im_1, im_2}(i, k) \quad (\text{A.6})$$

and its negative can be directly treated as a cost function for matching individual pixels, i.e. we want to minimize cost by maximizing the mutual information:

$$cost_{MI}(i, k) = -mi_{im_1, im_2}(i, k) \quad (\text{A.7})$$

Once the calculation framework is ready, we can start the actual search of this function, which maximizes the mutual information. The first step is to obtain the distribution for $P_{im_1, im_2}(i, k)$, which can be stored as a simple 256×256 histogram smoothed by a Gaussian kernel to simulate the Parzen window. However, some initial disparity assignment is needed to obtain the data points. In practice, it suffices to start with random disparity assignment to construct the mi_{im_1, im_2} function by the described-above procedure, and re-calculate the stereo disparity using this cost function. Doing these steps iteratively, we approach better estimates of disparity and the intensity mapping function with each iteration.

A more detailed derivation and explanation of MI in stereo matching can be found in [64] and [52].

We formulate the MI match measure for the local algorithm as recursive calculation of disparity based on a *single global* intensity mapping function – the cost is aggregated by

trivial summation of MI terms for individual pixels and disparity is assigned based on a WTA decision. Note that this is different from the local stereo method with MI pioneered by Egnal [38], where the author estimated the mapping function for each window separately, which is both inefficient, as gradient descent search is performed in each window, and error-prone as the number of samples taken from the local window is insufficient to estimate the mapping function reliably. Later, Fookes et al. [43] augmented the local MI calculation by introducing the prior which came from MI calculated for the whole image, while Kim et al. [64] showed that it is enough to use the global intensity mapping only calculated for the whole image.

Following the examples of [43, 52], we calculate MI using coarse-to-fine instead of iteratively. Interestingly, such an approach organically fits into coarse-to-fine stereo with pyramids, where the intensity mapping function will converge to its correct distribution with increasing resolution. As before, we use zero disparity map as the initial coarsest disparity map¹. Note, that matching with MI involves relatively little processing overhead – only filling and smoothing the histogram for Parzen windows on each resolution level, as described earlier.

The outlined procedure has one vital parameter – the width σ of the Gaussian kernel $g(i, k)$ used for Parzen windows. Unfortunately, no golden number exists, and its choice is very case-dependent [37]. In general, the choice of σ is governed by the amount of sample points that describe the distribution. When there are many points, σ can be rather small as the data itself is able to define a quite sharp and dense distribution; when the number of points is too few, greater smoothing is essential to get a more reasonable approximation to the original distribution. In coarse-to-fine stereo with MI, these considerations imply that σ cannot be fixed and should change with pyramid level, because the size of the images (hence, the number of datapoints) changes from level to level – the kernel variance σ should be quite high at coarse levels and shrink gradually while proceeding to finer levels.

From theoretical considerations, the volume cell (a d -dimensional unit of space over which the distribution is defined), should decrease at a rate slower than the increase of data size [37]. Considering that our intensity-mapping function is two-dimensional, a reasonable choice for the relationship between kernel window size for a single point, σ_1 (to be initially defined manually), number of data points, n , and the corresponding kernel width for Parzen window σ_n is given as

$$\sigma_n^2 = \frac{\sigma_1^2}{\sqrt{n}}, \tag{A.8}$$

¹Recall that some initial disparity assignment is required to obtain an initial MI matching cost function. While a random disparity map is usually used in this case, in the current implementation, zero initial disparity map yields the same performance as the random disparity map.

[37]. Thus, by choosing some specific value for σ_1 , the Parzen window Gaussian kernel for an image with n pixels can be deterministically derived as

$$\sigma_n = \frac{\sigma_1}{\sqrt[4]{n}} \quad (\text{A.9})$$

In conclusion, in the following pseudocode we summarize the augmentation of adaptive coarse-to-fine stereo algorithm **Module C** (as outlined in Section 2.2.3) to use Mutual Information as a match measure:

Module C-MI

```

Reference and matching images are initially
  brought into pyramid representation
disp(k,x,y) - disparity for pixel x, y on scale k
conf(k,x,y) - confidence for pixel x, y on scale k
Initialize ref_disp(:, :) to all zeros
For each level k from level_max to 0
  Create 256x256 MI cost function between intensity values
    of reference and matching image, using ref_disp as warping disparity
  For each pixel (k,x,y)
    Run Module A with search range
      [-delta_d+ref_disp(x,y), delta_d+ref_disp(x,y)]
    using the obtained MI cost function as a match measure
  End loop
  For each pixel (k,x,y)
    In the neighbourhood w of point (x,y)
      find (x_0,y_0) such that conf(k,x_0,y_0) is the best
      and assign disp(k,x,y) = disp(k,x_0,y_0);
  End loop
  ref_disp = upsampleNN(disp(k, :, :)) /* nearest-neighbour interpolation*/
End loop

```

Bibliography

- [1] Brown University, Image Sequences, <http://www.cs.brown.edu/people/black/images.html>. Current 2006.
- [2] Carnegie Mellon University, VASC image database, <http://www.vasc.ri.cmu.edu/idb>. Current 2006.
- [3] *Middlebury College, Stereo Vision Page*. <http://www.middlebury.edu/stereo/>. Current 2006.
- [4] A. Agarwal and A. Blake. The Panum Proxy algorithm for dense stereo matching over a volume of interest. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.
- [5] M. Agrawal and L. S. Davis. Window-based, discontinuity preserving stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 66–73, 2004.
- [6] L. Alvarez, R. Deriche, J. Weickert, and J. Sanchez. Dense disparity map estimation respecting image discontinuities: A PDE and scale-space based approach. *Journal of Visual Computation and Image Representation*, 13:3–21, 2002.
- [7] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–301, 1989.
- [8] A. Ansar, A. Castano, and L. Matthies. Enhanced real-time stereo using bilateral filtering. *Proceedings of the IEEE Symposium on 3D Data Processing, Visualization and Transmission*, pages 455–462, 2004.
- [9] H. H. Baker. Edge-based stereo correlation. In *Proceedings of the DARPA Image Understanding Workshop*, pages 168–175, 1980.
- [10] S. T. Barnard and M. A. Fischler. Computational stereo. *ACM Computing Surveys*, 14(4):553–572, 1982.
- [11] J. L. Barron, D. J. Fleet, and S. Beauchemin. *International Journal of Computer Vision*, 12:43–77, 1994.
- [12] D. N. Bhat and S. K. Nayar. Ordinal measures for image correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):415–423, April 1998.
- [13] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401–406, 1998.

- [14] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 489–495, 1999.
- [15] M. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, 19(1):57–92, July 1996.
- [16] M. Bleyer and M. Gelautz. A layered stereo algorithm using image segmentation and global visibility constraint. In *Proceedings of the IEEE International Conference on Image Processing*, pages 2997–3000, 2004.
- [17] A. F. Bobick and S. S. Intille. Large occlusion stereo. *International Journal of Computer Vision*, 33(3):181–200, September 1999.
- [18] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.
- [19] Y. Boykov, O. Veksler, and R. Zabih. A variable window approach to early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1283–1294, 1998.
- [20] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.
- [21] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):993–1008, August 2003.
- [22] A. Bruhn, J. Weickert, and C. Schnorr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal on Computer Vision*, 63:211–231, 2005.
- [23] P. Burt and B. Julesz. A disparity gradient limit for binocular fusion. *Nature*, 208:615–617, 1980.
- [24] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, April 1983.
- [25] P. J. Burt, T. H. Hong, and A. Rosenfeld. Segmentation and estimation of image region properties through cooperative hierarchical computation. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(12):802–809, 1981.
- [26] H. Burton. The optics of Euclid. *Journal of Optical Society of America*, 35:357–372, 1945.
- [27] O. Choi, K.-J. Yoon, and I.-S. Kweon. A hierarchical window based approach for correspondence problem in vision. In *Proceedings of the International Consortium for Medical Imaging Technology*, pages 590–594, 2003.
- [28] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal. Automated multi-modality image registration based on information theory. In *Information Processing in Medical Imaging, Y. Bizais and C. Barillot and R. Di Paola eds.*, pages 263–274. Kluwer Academic Publishers, Dordrecht, 1995.

- [29] N. Cornelis and L. V. Gool. Real-time connectivity constrained depth map combination using programmable graphics hardware. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1099–1104, 2005.
- [30] I. J. Cox, S. L. Hingorani, S. Rao, and B. M. Maggs. A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567, 1996.
- [31] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. Bilayer segmentation of live video. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, 2006.
- [32] A. Criminisi, J. Shotton, A. Blake, and P. Torr. Gaze manipulation for one-to-one teleconferencing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 191–198, 2003.
- [33] A. Darabiha, W. J. MacLean, and J. Rose. Reconfigurable hardware implementation of a phase-correlation stereo algorithm. *Machine Vision and Applications Journal*, 17:116–132, 2006.
- [34] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 34:1–38, 1977.
- [35] Y. Deng, Q. Yang, X. Lin, and X. Tang. A symmetric patch-based correspondence model for occlusion handling. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 1316–1322, 2005.
- [36] U. R. Dhond and J. K. Aggarwal. Structure from stereo - A review. *IEEE Transactions on System, Man and Cybernetics*, 19(6):1489–1510, 1989.
- [37] R. Duda, P. Hart, and D. Stork. *Pattern Classification (Second Edition)*. John Wiley & Sons, Inc., New York, USA, 2001.
- [38] G. Egnal. Mutual information as a stereo correspondence measure. Technical Report MS-CIS-00-20, University of Pennsylvania, 2000.
- [39] G. Egnal and R. P. Wildes. Detecting binocular half-occlusions: Empirical comparisons of five approaches. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1127–1133, August 2002.
- [40] P. F. Felzenszwalb and D. P. Huttenlocher. Distance transforms of sampled functions. Technical Report TR2004-1963, Cornell Computing and Information Science, 2004.
- [41] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 261–268, 2004.
- [42] D. J. Fleet, A. D. Jepson, and M. R. M. Jenkin. Phase-based disparity measurement. *Journal of Computer Vision, Graphics, and Image Processing*, 53(2):198–210, 1991.
- [43] C. Fookes, M. Bennamoun, and A. Lamanna. Improved stereo image matching using mutual information and hierarchical prior probabilities. In *Proceedings of the IEEE International Conference on Pattern Recognition*, pages 937–940, 2002.

- [44] S. Forstmann, Y. Kanou, J. Ohya, S. Thuring, and A. Schmitt. Real-time stereo by using dynamic programming. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Workshop*, volume 3, pages 29–35, 2004.
- [45] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ, USA, 2003.
- [46] A. Fusiello and V. Roberto. Efficient stereo with multiple windowing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 97)*, pages 885–863, 1997.
- [47] D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. *International Journal of Computer Vision*, 14:211–226, 1995.
- [48] S. B. Goldberg, M. W. Maimone, and L. Matthies. Stereo vision and rover navigation software for planetary exploration. In *Proceedings of the IEEE Aerospace Conference*, volume 5, pages 2025–2036, 2002.
- [49] M. Gong and Y.-H. Yang. Multi-resolution stereo matching using genetic algorithm. *Proceedings of the IEEE Workshop on Stereo and Multi-Baseline Vision*, pages 21–29, 2001.
- [50] N. Grammalidis and M. G. Strintzis. Disparity and occlusion estimation in multiocular systems and their coding for the communication of multiview image sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(3):328–334, June 1998.
- [51] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision. Second Edition*. Cambridge University Press, Cambridge, UK, 2004.
- [52] H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 807–814, 2005.
- [53] H. Hirschmuller. Stereo vision in structured environments by consistent semi-global matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2006.
- [54] H. Hirschmuller, P. R. Innocent, and J. Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision*, 47:229–246, 2002.
- [55] L. Hong and G. Chen. Segment-based stereo matching using graph cuts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 74–81, 2004.
- [56] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, pages 185–203, 1981.
- [57] I. Howard and B. Rogers. *Seeing in Depth*. I. Porteus, Thornhill, Ontario, Canada, 2002.
- [58] H. Ishikawa and D. Geiger. Occlusions, discontinuities and epipolar lines in stereo. In *Proceedings of the European Conference on Computer Vision*, pages 1–14, 1998.
- [59] B. Jahne. *Digital Image Processing: Concepts, Algorithms and Scientific Applications*. Berlin: Springer-Verlag, 1993.

- [60] D. G. Jones and J. Malik. A computational framework for determining stereo correspondence from a set of linear spatial filters. In *Proceedings of the European Conference on Computer Vision*, pages 395–410, 1992.
- [61] P.-J. Kack. Robust stereo correspondence using graph cuts. Master’s thesis, School of Computer Science and Engineering, Royal Institute of Technology, Stockholm, 2004.
- [62] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:920–932, 1994.
- [63] M. Kanbara, T. Okuma, H. Takemura, and N. Yokoya. A stereoscopic video see-through augmented reality system based on real-time vision-based registration. In *Proceedings of the 2d IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 255–262, 2000.
- [64] J. Kim, V. Kolmogorov, and R. Zabih. Visual correspondence using energy minimization and mutual information. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1033–1040, October 2003.
- [65] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *Proceedings of the IEEE International Conference on Pattern Recognition*, 2006.
- [66] V. Kolmogorov. Convergent tree-reweighted message passign for energy minimization. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2006.
- [67] V. Kolmogorov, A. Criminisi, G. C. A. Blake, and C. Rother. Bi-layer segmentation of binocular stereo video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.
- [68] V. Kolmogorov and C. Rother. Comparison of energy minimization algorithm for highly connected graphs. In *Proceedings of the European Conference on Computer Vision*, May 2006.
- [69] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 508–515, July 2001.
- [70] G. Konecny and D. Pape. Correlation techniques and devices. *Photogrammetric Engineering and Remote Sensing*, pages 323–333, 1981.
- [71] J. Krol and W. van der Grind. The double nail illusion. *Perception*, 9:651–659, 1980.
- [72] J. Krol and W. van der Grind. Rehabilitation of a classical notion of Panum’s fusional area. *Perception*, 11:615–619, 1982.
- [73] R. B. Lawson and D. C. Mount. Minimum condition for stereopsis and anomalous contour. *Science*, 158:802–804, November 1967.
- [74] S. H. Lee, Y. Kanatsugu, and J.-I. Park. Hierarchical stochastic diffusion for disparity estimation. In *Proceedings of the IEEE Workshop on Stereo and Multi-Baseline Vision*, pages 111–120, 2001.

- [75] C. Leung, B. Appleton, and C. Sun. Fast stereo matching by iterated dynamic programming and quadtree subregioning. *Proceedings of the British Machine Vision Conference*, pages 97–106, 2004.
- [76] G. Li and S. W. Zucker. A differential geometrical model for contour-based stereo correspondence. In *Proceedings of the IEEE Workshop on Variational, Geometric, and Level Set Methods in Computer Vision, Nice, France, 2003*.
- [77] M. Lin and C. Tomasi. Surfaces with occlusion from layered stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1073–1078, 2004.
- [78] T. Lindeberg. *Scale-space theory in computer vision*. Boston: Kluwer Academic Publishers, 1994.
- [79] J. J. Little. Accurate early detection of discontinuities. In *Proceedings of the Conference Vision Interfaces*, pages 97–102, 1992.
- [80] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [81] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, page 674679, 1981.
- [82] R. Maas, B. M. ter Haar Romeny, and M. A. Viergever. Area-based computation of stereo disparity with model-based window size selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 106–112, 1999.
- [83] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194:283–287, 1976.
- [84] D. K. Masrani and W. J. MacLean. Expanding disparity range in an fpga stereo system while keeping resource utilization low. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition - Workshops*, pages 132–139, 2005.
- [85] H. Mayer. Analysis of means to improve cooperative disparity estimation. In *ISPRS Conference on Photogrammetric Image Analysis, Technical university of Munich, Germany, September 2003*.
- [86] G. G. Medioni and R. Nevatia. Segment-based stereo matching. *Computer Vision, Graphics, and Image Processing*, 31(1):2–18, July 1985.
- [87] G. V. Meerbergen, M. Vergauwen, M. Pollefeys, and L. V. Gool. A hierarchical symmetric stereo algorithm using dynamic programming. *International Journal of Computer Vision*, 47:275–282, 2002.
- [88] H. Moravec. Rover visual obstacle avoidance. In *Proceedings of the IEEE International Joint Conference on Artificial Intelligence*, pages 785–790, 1981.
- [89] J. Mulligan, V. Isler, and K. Daniilidis. Trinocular stereo: a real-time algorithm and its evaluation. *International Journal of Computer Vision*, 47:51–61, 2002.

- [90] A. S. Ogale and Y. Aloimonos. Stereo correspondence with slanted surfaces: critical implications of horizontal slant. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 568–573, 2004.
- [91] M. Okutomi, Y. Katayama, and S. Oka. A simple stereo algorithm to recover precise object boundaries and smooth surfaces. *International Journal of Computer Vision*, 47:261–273, 2002.
- [92] S. Y. Park, S. H. Lee, and N. I. Cho. Segmentation based disparity estimation using color and depth information. In *Proceedings of the IEEE International Conference on Image Processing*, pages 3275–3278, 2004.
- [93] M. P. Patricio, F. Cabestaing, O. Colot, and P. Bonnet. A similarity-based adaptive neighborhood method for correlation-based stereo matching. In *Proceedings of the IEEE International Conference on Image Processing*, pages 1341–1344, 2004.
- [94] S. Pollard, J. Mayhew, and J. Frisby. PMF: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470, 1985.
- [95] L. Quam. Hierarchical warp stereo. *Proceedings of the DARPA Image Understanding Workshop*, pages 149–155, 1984.
- [96] J. P. Queiroz-Neto, R. Carceroni, W. Barros, and M. Campos. Underwater stereo. In *Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing (SIB-GRAPI'04)*, pages 170–177, 2004.
- [97] J. Richter (Ed.). *Selections from the Notebooks of Leonardo da Vinci*. Oxford, U.K.: Oxford University Press, 1977.
- [98] S. Roy and I. J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 492–502, 1998.
- [99] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ, USA, 2003.
- [100] R. Sara. Finding the largest unambiguous components of stereo matching. In *Proceedings of the European Conference on Computer Vision*, volume 3, pages 900–914, 2002.
- [101] D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. *International Journal of Computer Vision*, 28(2):155–174, 1998.
- [102] D. Scharstein and R. Szeliski. Taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47:7–42, 2002.
- [103] S. Se and P. Jasiobedzki. Instant scene modeler for crime scene reconstruction. In *Proceedings of the IEEE Workshop on Advanced 3D Imaging for Safety and Security (A3DISS)*, San Diego, USA, june 2005.
- [104] J. Shah. A nonlinear diffusion model for discontinuous disparity and half-occlusions in stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 34–40, 1993.

- [105] M. Shimizu and M. Okutomi. Precise subpixel estimation on area-based matching. *Systems and Computers in Japan*, 33, July 2002.
- [106] M. Sizintsev and R. Wildes. Computational analysis of binocular half-occlusions. Technical Report CS-2005-12, York University, 4700 Keele street, Toronto, Ontario, Canada, 2005.
- [107] L. D. Stefano, M. Marchionni, S. Mattoccia, and G. Neri. A fast area-based stereo matching algorithm. *Image and Vision Computing*, 22:983–1005, 2004.
- [108] C. Strecha, R. Fransens, and L. van Gool. Wide-baseline stereo from multiple views: A probabilistic account. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 718–725, 2004.
- [109] C. Sun. Fast stereo matching using rectangular subregioning and 3D maximum-surface techniques. *International Journal of Computer Vision*, 47:99–117, May 2002.
- [110] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum. Symmetric stereo matching for occlusion handling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 399–406, June 2005.
- [111] J. Sun, N.-N. Zheng, and H.-Y. Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):787–800, 2003.
- [112] R. Szeliski and P. Golland. Stereo matching with transparency and matting. *International Journal of Computer Vision*, 32:45–61, 1999.
- [113] R. Szeliski and D. Scharstein. Sampling the disparity space image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3):419–425, 2004.
- [114] R. Szeliski, R. Zabih, D. Scharstein, O. Veskler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. Comparative study of energy minimization methods for markov random fields. In *Proceedings of the European Conference on Computer Vision*, 2006.
- [115] H. Tao, H. S. Sawhney, and R. Kumar. A global matching framework for stereo computation. In *Proceedings of IEEE International Conference on Computer Vision*, pages 532–539, 2001.
- [116] M. F. Tappen and W. T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 900–907, 2003.
- [117] E. Trucco and A. Verri. *Introductory techniques for 3-D computer vision*. Prentice Hall, Upper Saddle River, NJ, USA, 1998.
- [118] G. van der Wal, M. Hansen, and M. Piacentino. The Acadia vision processor. In *Proceedings of the International Workshop on Computer Architecture for Machine Perception, Padua, Italy*, pages 31–40, September 2000.
- [119] O. Veksler. *Efficient Graph-Based Energy Minimization Methods in Computer Vision*. PhD thesis, Cornell University, August 1999.
- [120] O. Veksler. Stereo correspondence with compact windows via minimum ratio cycle. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1645–1660, 2002.

- [121] O. Veksler. Fast variable window for stereo correspondence using integral images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 556–561, 2003.
- [122] O. Veksler. Stereo correspondence by dynamic programming on a tree. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 384–390, 2005.
- [123] P. Viola and W. M. Wells III. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24:137–154, 1997.
- [124] L. Wang, M. Liao, M. Gong, R. Yang, and D. Nister. High-quality real-time stereo using adaptive cost aggregation and dynamic programming. In *Proceedings of the Third International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, 2006.
- [125] Y. Wei and L. Quan. Region-based progressive stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 106–113, 2004.
- [126] Q. Yang, L. Wang, R. Yang, S. Wang, M. Liao, and D. Nister. Real-time global stereo matching using hierarchical belief propagation. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2006.
- [127] R. Yang and M. Pollefeys. Multi-resolution real-time stereo on commodity graphics hardware. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 211–217, 2003.
- [128] K.-J. Yoon and I.-S. Kweon. Locally adaptive support-weight correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 924–931, 2005.
- [129] A. Yuille and T. Poggio. A generalized ordering constraint for stereo. *AI Lab Memo 777, MIT, Cambridge, MA*, 1984.
- [130] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proceedings of the European Conference on Computer Vision*, pages 151–158, 1994.
- [131] Y. Zhang and C. Kambhamettu. Stereo matching with segmentation-based cooperation. In *Proceedings of the European Conference on Computer Vision*, pages 556–571, 2002.
- [132] L. Zitnick and T. Kanade. A cooperative algorithm for stereo matching with occlusion detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:675–684, 2000.