Image and Vision Computing 28 (2010) 352–366

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Coarse-to-fine stereo vision with accurate 3D boundaries

Mikhail Sizintsev *, Richard P. Wildes

Department of Computer Science & Engineering, and Centre for Vision Research, York University, 4700 Keele Street, CSE 3023, Toronto, Ont., Canada M3J 1P3

ARTICLE INFO

Article history:
Received 10 July 2007
Received in revised form 21 December 2008
Accepted 3 June 2009

Keywords:
Computer vision
Stereo
Coarse-to-fine
Occlusions
Real-time algorithms

ABSTRACT

This paper presents methods for efficient recovery of accurate binocular disparity estimates in the vicinity of 3D 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. A novel coarse-to-fine refinement procedure that adapts match window support across scale to ameliorate corruption of disparity estimates near boundaries is presented. Extensions are included to account for half-occlusions and colour uniformity. Empirical results show that incorporation of these advances in the standard coarse-to-fine, block matching framework reduces disparity errors by more than a factor of two, while performing little extra computation, preserving low complexity and the parallel/pipeline nature of the framework. Moreover, the proposed advances prove to be beneficial for CTF global matchers as well.

1. Introduction

Significant strides have been made in the investigation of artificial binocular stereo. This research is driven by the wide applicability of stereo vision, e.g. in robot manipulation, vehicle guidance and augmented reality. In turn, these applications impose strict requirements on developed technology to be computationally efficient, accurate and precise. Of current outstanding problems in stereo, of particular concern are speed-accuracy tradeoffs and poor reconstruction in the vicinity of 3D object boundaries. The reliable recovery of 3D boundaries is crucial and remains a shortcoming of todays most efficient stereo algorithms, e.g. block matching-based approaches. While computer processing power keeps increasing and thereby provides the ability to utilize complex and computationally expensive solutions, sensor resolution increases even faster [1], which necessitates continued development of low-complexity algorithms.

In this paper we are concerned with the design of algorithms with potential for rapid execution on compact, readily available computational platforms even while exhibiting robust performance in the vicinity of 3D boundaries. We acknowledge that recent efforts have yielded a number of sophisticated, non-linear optimization algorithms that produce remarkable results for both textureless regions and near 3D discontinuities [2]. Nevertheless, we instead concentrate our efforts on an alternative class of algorithms, coarse-to-fine stereo matchers, as such methods remain important when rapid execution is at a premium. Here, if accuracy can be improved significantly without adversely impacting complexity and efficiency, then speed-accuracy trade-offs will be improved correspondingly and thereby increase the real-world deployment of machine stereo vision. As this paper mainly concerns coarse-to-fine stereo correspondence procedures, the abbreviation CTF will be used to denote it throughout.

CTF processing is of interest for several remarkable properties. It helps remove local minima in correspondence search by their reduction at coarser resolutions. As commonly embedded in image pyramids (where image sampling is commensurate with scale) ensuing processing can reduce match ambiguities, as large match windows at fine resolution are covered by smaller windows at coarse resolution. Also, processing speed increases as large disparities at fine resolution can be recovered at coarse resolution with smaller search ranges (subject to refinement at finer resolution). While recent advances in global methods improve efficiency [3,4], block matchers, often with CTF, remain preferred when speed is a concern; such procedures inherently entail lower processing demands, map well to current hardware and software architectures [3,5,6] and are suitable for parallel and pipeline computation [7].

For both local and global methods of disparity estimation, reliable recovery in the vicinity of 3D surface boundaries remains a matter of concern. This problem is of particular note in conjunction with CTF approaches, which tend to resolve poorly such geometries as they are not well represented at coarser resolutions. In the past, much research has considered recovery of binocular disparity near 3D boundaries [2,3,8]. For local methods, the use of adaptive spatial support for match windows can ameliorate issues arising in attempts to match near 3D discontinuities by shaping

* Corresponding author. Tel.: +1 416 736 2100x33411; fax: +1 416 736 5857.
E-mail addresses: sizints@cse.yorku.ca, msizintsev@gmail.com (M. Sizintsev), wildes@cse.yorku.ca (R.P. Wildes).
URL: http://www.cse.yorku.ca/vision (M. Sizintsev).

0262-8856/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2009.06.008

windows to avoid poorly defined matches [9–12]. Many recent advances in disparity estimation near 3D boundaries explicitly consider half-occlusions, where one view sees portions of a background surface that are occluded to the other view by a foreground surface. Some of the most impressive recent results have been demonstrated in conjunction with global methods [13–16]. In comparison, empirical investigation of half-occlusion detection with local processing underlines shortcomings [8]. Moreover, occlusion treatment within CTF processing schemes is rarely discussed in the literature.

The organization of this paper is as follows. Section 1 has motivated our research in CTF, block-matching stereo; further discussion of related research is delayed until our approach is detailed to better expose similarities and differences. Section 2 presents a novel analysis of CTF stereo errors and leads to a simple, yet effective solution to 3D boundary preservation within this matching paradigm. This section also addresses the colour segmentation cue and the problem of half-occlusions, including our novel approach to dealing with such points within the CTF block-matching framework. In Section 3 we document thorough, systematic empirical evaluation of the proposed algorithmic solutions. Finally, Section 4 discusses the proposed advances and their importance in the scope of other binocular stereo approaches.

2. Adaptive coarse-to-fine stereo for 3D boundary preservation

2.1. Basic algorithm and its properties

The basic elements of CTF block binocular matching can be outlined as follows (see [2,3] and references cited therein). Initially both images are brought into image pyramid representations [17,18] 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. At each level, disparity is estimated using any local stereo method. We can describe this procedure mathematically as follows. Let I_1 and I_2 be a pair of images, l_{max} be the number of pyramid levels, ρ be the pixel-wise match cost function (e.g. squared difference (SD) or other [3]), w be the support window function, G be the smoothing kernel applied via the convolution operation $\otimes$, $\downarrow_2$ be subsampling by a factor of two and $\uparrow_2$ be upsampling by a factor of two. We begin by constructing pyramids for the two input images

$$I_1^0 = I_1,\, I_2^0 = I_2, \quad \forall(j)|1 \leqslant j \leqslant l_{max} : I_i^j = \left(G \otimes I_i^{j-1}\right)\downarrow_2, \tag{1}$$

and initializing a corresponding disparity pyramid

$$\forall(x,y)| : disp^{l_{max}+1}(x,y) = 0. \tag{2}$$

Disparity estimation then operates coarse-to-fine over the defined pyramids

$$\forall(k)|l_{max} > k > 0 : \Big[\forall(x,y)| : disp^k(x,y) = 2 \cdot disp^{k+1}(x,y)\uparrow_2 + \arg\min_{d_i} \sum_{(u,v)\in w(x,y)} \rho\Big(I_1^k(u,v), I_2^k(u + 2\cdot disp^{k+1}(x,y)\uparrow_2 + d_i, v)\Big)\Big]. \tag{3}$$

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 the same size (in terms of pixels) support region constitutes to larger support at coarser levels. Further, large disparities in the high-resolution images correspond to small disparities in low-resolution subsampled images; hence, large disparity search space is covered by smaller searches at higher pyramid levels. The last fact makes CTF very efficient because the algorithm can search over smaller disparity ranges at each level. As the original disparity search range can be on the order of a hundred for big images, the gain of CTF in terms of speed may be crucial, especially when real-time performance is needed. Unfortunately, 3D boundaries can be severely degraded during CTF estimation, a side-effect that we proceed to ameliorate.

2.2. 3D boundary deterioration

Intuitively, certain errors introduced by CTF processing 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 the subsampling procedure. Typically, the filtering is realized via a Gaussian kernel as it is causal in scale space [19] and yields an efficient implementation.

At the same time, the low-pass filtering operation effectively blurs the depth discontinuities, i.e. a point near a 3D boundary becomes a mixture of foreground and background surface colours. The actual proportion of the mixture will depend on the ratio of the surfaces' areas covered by the convolution window and the surface properties themselves (Fig. 1 shows a few examples). Thus, pyramid-based CTF procedure, (3), cannot be expected to reliably recover pixels in the vicinity of discontinuities, which means that errors will be accumulated and propagated to finer levels in the computation.

The above-described CTF discontinuity blur is quite similar to issues in match window aggregation when applied near 3D boundaries and the assumption of a single surface in depth is violated. This resulting overreach flaw, typical for window-based matching, is well-researched and a number of efficient remedies have been proposed, e.g. shiftable, overlapping, adaptive windows [9–12,20]. At the same time, the combined effects of CTF projection and match window aggregation across 3D boundaries lead to especially severe boundary deterioration. An associated technical report [21] presents a rigorous analysis of these combined effects.

Furthermore, in the examples of Fig. 1, low-pass filtering attributes a large portion of pixels near discontinuities to the wrong surface. Thus, coarse disparity estimates for such points and in their vicinity tend to be incorrect and these erroneous results are to be upsampled and refined at finer levels propagating and increasing the amount of errors even further. These observations show that 3D boundary preserving CTF disparity estimation must have two key properties. First, CTF upsampling must prevent error accumulation incurred by combining information from surfaces at

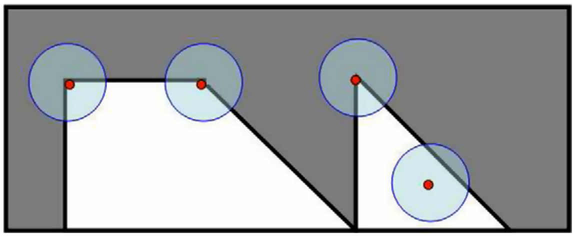


Fig. 1. Blurring of depth discontinuities via low-pass filtering. Gray and white are two different surfaces. Red points are various locations on the white surface close to the discontinuities (black lines). Large semitransparent circles are the support kernel for low-pass Gaussian blur. Note that in most cases, especially corners, the support kernel covers significant proportion of the other surface, which means that it is very likely to attribute those points to the wrong surface as a result of low-pass operations.

different depths. Second, adaptive match window aggregation must prevent support from crossing 3D boundaries.

2.2.1. Disparity upsampling: adaptivity in scale

Assume for a moment that we can precisely recover the disparity map at a current level k and wish to refine this estimate for level $k-1$. As a part of refinement, the coarse estimate must be projected (upsampled) to finer spatial resolution; the procedure is not uniquely defined and various alternative exist, e.g. Nearest Neighbour, Linear, Gaussian interpolations and others [18]. Logically, it should depend on the pyramid construction procedure – Nearest Neighbour is the most suitable for Quadtree pyramids, while Gaussian upsampling is the best for the Gaussian and Laplacian pyramid [17,18]. However, such reasoning does not quite work for pyramids of discontinuous disparity maps.

The snapshot of CTF estimation in Fig. 2 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 nearest neighbour 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 3D boundaries.

In the example of Fig. 2, assume a 3D discontinuity between a, c and b, d. Point x can belong to either surface and it is impossible to distinguish between different cases based on the low resolution disparity map alone. In particular, 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.

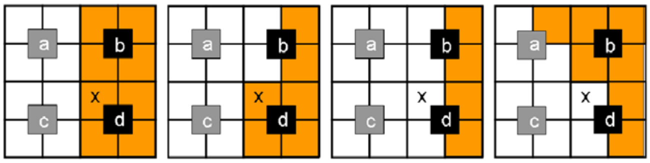


Fig. 2. Snapshot of the coarse-to-fine (CTF) disparity estimation procedure. White and orange cells are pixels at the fine level, gray and black pixels are from the coarse level. Disparity offset for pixel x can be one of disparities at points a, b, c or d (scaled by 2). Disparity discontinuity is between a, c (gray) and b, d (black) with two different surfaces shaded white and orange, respectively. Various cases for the exact discontinuity position are shown.

A straightforward solution to overcome such problems is to use multiple disparity offsets for each fine level pixel so as to encompass all initializations available at neighbouring coarse locations, rather than a single offset provided by standard upsampling procedures. In essence, such an adaptive procedure would try to find the best (according to matching metric) coarse disparity value that is the locally most distant from the 3D boundary, i.e. the points where disparity estimation is more reliable and less-affected by low-pass filtering used in the pyramid construction (1).

2.2.2. Adaptation in space and scale

Adaptive windowing, which we refer to as adaptation in space, has received consideration in stereo matching to avoid aggregation across surfaces at different depths. However, its appropriate combination with the disparity projection procedure is specific to CTF refinement and has not been given adequate attention, i.e. adaptation in scale, as discussed in the previous section, is required as well.

In principle, it is possible to achieve simultaneous space and scale adaptation by combining any adaptive windows technique, e.g. [9], with the adaptive offsets as in Section 2.2.1. This entails additional correspondence search at each finer level for each combination of offset and window configuration, with final disparity assignment taken as that yielding the best score under the block-matching metric.

A closer look suggests a more efficient approach. In Fig. 3, disparity for point x is to be refined in the vicinity of various 3D discontinuity profiles. The red line presents the 3D boundary, the corresponding optimal, e.g. 3 × 3, aggregation window is shaded in orange (centered at point y) and the best offset is marked with a yellow circle. It is seen that the best offset always derives from the coarse resolution parent of the central pixel of the best window. This conclusion is supported by intuition that the optimal window should not cross a 3D boundary, i.e. the centre point of the optimal matching window does not lie on a 3D discontinuity and it makes the nearest neighbour upsampling procedure sufficient.

Thus, the best initialization, match window and refinement for x are achieved via nearest neighbour 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 the central pixel implicitly encompasses all possible initializations! Essentially, we extend the observation of [9] to CTF refinement: "The disparity profile itself drives the selection of an appropriate window" *and disparity offset*.

Mathematical formulation capturing the essential notions is as follows. Using the same notational conventions as in our original CTF formulations (1)–(3), we begin by constructing image pyramids for the input stereo pair and initializing a corresponding disparity pyramid as specified in formulas (1) and (2). Next, at each pyramid level, k, an initial disparity map, $disp_0^k$, is recovered

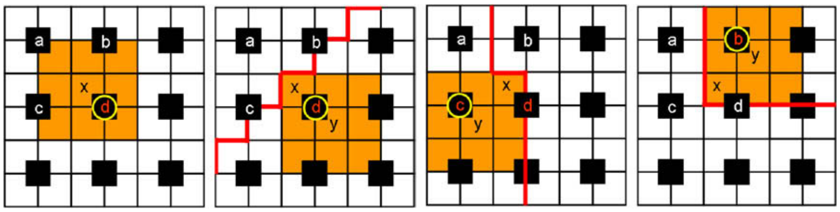
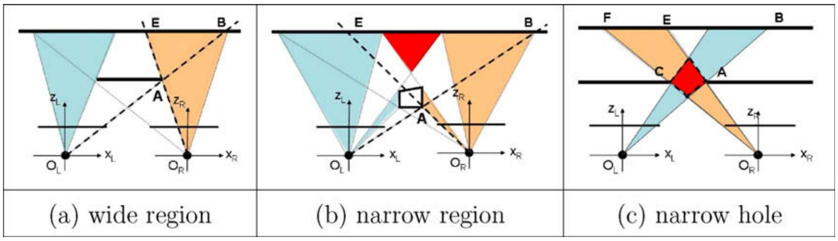


Fig. 3. Snapshot of the coarse-to-fine (CTF) disparity estimation procedure adaptive in scale and space for various configurations of depth discontinuity. White and orange cells are pixels at the fine level, black pixels are from the coarse level. Disparity offset for pixel x can be one of disparities at points a, b, c or d (scaled by 2). 3D boundary is shown in red and the corresponding optimal 3 × 3 window (with centre point y is as shaded in orange). The correct disparity offset is labeled with yellow circle.



(a) wide region (b) narrow region (c) narrow hole

Fig. 4. Three cases of half-occlusion formation. Shown are top-down views of projection to left and right imaging systems centered at O_L and O_R, respectively. Bold lines through points **A** and **B** depict imaged foreground and background surfaces, respectively. In all cases point **A** occludes point **B** in the left view. (a) Front surface occludes back surface creating a single half-occlusion region for each view (shaded on the sketch). (b) Narrower front surface creates configuration of multiple occlusions and binocular visibility regions for back surface. Further interposed surfaces in the red region allow for half-occlusion relations to occur *recursively*. (c) A narrow hole may leave the observed background surface binocularly invisible, i.e. left and right cameras see completely disjoint regions of the background. This case is particularly hard for computational stereo, as disparity of the background surface cannot be detected in principle and no occluder-occluded interrelationship can be stated in terms of disparity per se.

$$\forall (x,y)| : disp_0^k(x,y) = 2\cdot disp^{k+1}(x,y)\uparrow_2 + \arg\min_{d_i} \sum_{(u,v)\in w(x,y)} \rho\left(I_1^k(u,v), I_2^k(u+2\cdot disp^{k+1}(x,y)\uparrow_2 + d_i, v)\right) \qquad (4)$$

and associated with a confidence map based on its match scores

$$conf_0^k(x,y) = \sum_{(u,v)\in w(x,y)} \rho\left(I_1^k(u,v), I_2^k(u+disp_0^k(x,y), v)\right). \qquad (5)$$

The final disparity estimate at each level, $disp^k$, is adapted by taking the disparity within the local match window support, w, that has the highest confidence (lowest match cost)

$$disp^k(x,y) = disp_0^k\left(\arg\min_{(u,v)\in w(x,y)} conf_0^k(u,v)\right). \qquad (6)$$

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; and (ii) finalize the disparity map at each pixel by choosing the disparity of the neighbouring 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 SSD and dilation for NCC match measures [3]) using the aggregation window as a structural element to simulate shiftable windows in single-scale matching [2]. Note that the proposed approach is not identical to estimating disparity estimates at each level via shiftable windows, as proposed in [9,22] 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 **Adaptive CTF** or **ACTF**.

2.3. Half-occlusions

Half-occlusions, where one view in a pair sees surface points that are obscured to the other, occur at 3D boundaries and must be accounted for in accurate reconstruction [3]. Representative cases of such configurations are shown in Fig. 4. Similarly to its predecessors, i.e. variable window approaches [9–12,20], the ACTF algorithm has the capability to avoid half-occluded regions by positioning the support window and choosing the right offset such that occlusions are covered in the least possible way. Nevertheless, it does not explicitly detect and label half-occluded regions as such.

To augment the ACTF algorithm to encompass half-occlusions, we combine disparity and occlusion estimation in a single cooperative scheme that accounts for their mutual dependence, i.e. the fact that disparity information is needed for reasoning about occlusion and vice versa. In particular, at each resolution the algorithm recovers initial disparity and half-occlusion maps; however, prior to refinement, the disparities of background surfaces are extrapolated into the occluded regions of the disparity map. Refinement is then executed at the next finer resolution, i.e. recovery of increased resolution disparity and half-occlusion maps. This approach allows interpolated disparity values to guide estimation at the next level for more precise disparity recovery that, in turn, supports more accurate recovery of half-occlusions.

The described approach allows the ACTF disparity estimation procedure to operate essentially as in its original formulation, except that half occlusion detection must be performed at each resolution level followed by disparity extrapolation. For half-occlusion detection, any local method could be employed [3,8]; here, we use an approach that complements geometric analysis of foreground/background surface disparities with consideration of match quality, which is detailed elsewhere (including comparison to alternatives) [21,23] and is briefly documented next. For disparity extrapolation, we make use of piecewise constancy.

The employed geometric constraint for local half-occlusion detection derives from conditions under which two 3D scene points, **A** and **B**, both visible in the reference image, I_1, occlude one another in the match image,[1] I_2. To characterize such situations, we rely on violations of match uniqueness, as background points are mapped to the same position as foreground points in the occluded view [3,13]. In particular, let $d(\cdot)$ and $x(\cdot)$ specify the disparity and image coordinate along the (rectified) horizontal axis of a 3D point, then violations of uniqueness are captured as

$$d(\mathbf{A}) + x(\mathbf{A}) = d(\mathbf{B}) + x(\mathbf{B}), \qquad (7)$$

i.e. both $x(\mathbf{A})$ and $x(\mathbf{B})$ map to the same location in the match image. 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 be of low quality, at least for areas having distinctive texture. So, given two or more points satisfying (7), the point with the best match score is taken as binocularly visible, and the others as half-occluded.

In practice, straightforward use of uniqueness, (7), is not robust to slanted surfaces [13] and continuous disparity, as integer disparity quantization can cause multiple pixels in one image to map to a

[1] Since the formulation makes use of points **A** and **B** both being visible in one of the views, the constraint is applicable to all typical half-occlusion regions where a foreground surface, wide or narrow, occludes a background surface (as in Fig. 4a and b); however, it is not applicable to binocular viewing through a slit (as in Fig. 4c), as neither view can see the occluded point, **B**. Slit viewing is not dealt with explicitly in the present work; however, empirical results reported in Section 3.2 show that the approach is reasonably robust to such situations.

single pixel in the other. We deal with this situation as follows. Integer disparity values are interpolated to subpixel precision (e.g. parabola fitting around the match peak [10]). 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). The criterion for grouping adjacent pixels as arising from a single surface is $\|\Delta d\| < 1$, with Δd the interpixel disparity difference. This criterion is based on the widely used *occlusion* ($\Delta d \geqslant 1$) and *ordering* constraints ($\Delta d \leqslant -1$), as they both imply depth discontinuities [3,8].

In summary, the half-occlusion detection procedure is as follows: For each scanline, form the surface equivalence classes by considering interpixel disparity difference and obtain the sets of points that violate uniqueness (7). Within each set, find the point with the best match score and mark all others in the set that are not in the same surface class as occluded. Given an initial disparity map calculated at any given level in the ACTF algorithm, this occlusion detection procedure is executed and corresponding regions in the disparity map are extrapolated from the background surface. Refinement is then executed at the next resolution.

2.4. Colour segmentation

It is a straightforward matter to augment the proposed ACTF method with a colour segmentation cue. So far, we have used shiftable windows of fixed square size based on the intensity driven match score. Now, the locally best window is chosen based not on the match score alone, but also on a measure that maximizes the number of pixels within the support window belonging to the same surface based on colour uniformity. To achieve this end, we largely adapt the recently proposed Gestalt-based aggregation window formulation [12], which constructs window support via consideration of both colour similarity and geometric pixel proximity. For a point (x,y) in the reference view to be matched, the colour similarity component is defined in terms of Euclidean distance across the colour channels, e.g. R(ed), G(reen), B(lue), or alternative colour spaces like CIE LAB or HSV,

$$\varrho_c(x,y) = \sum_{(u,v)\in w(x,y)} \sqrt{(R(u,v)-R(x,y))^2+(G(u,v)-G(x,y))^2+(B(u,v)-B(x,y))^2} \tag{8}$$

with $w(x,y)$ being the aggregation window around the point (x,y). The geometric proximity component also is defined in terms of Euclidean distance as

$$\varrho_{\mathscr{P}}(x,y) = \sum_{(u,v)\in w(x,y)} \sqrt{(u-x)^2+(v-y)^2}. \tag{9}$$

Finally, the colour augmented confidence of match is defined as

$$conf(x,y) = \varrho(x,y) + \lambda\left(\frac{\varrho_c(x,y)}{\alpha} + \frac{\varrho_{\mathscr{P}}(x,y)}{\beta}\right) \tag{10}$$

with $\varrho(x,y)$ the match measure (e.g. SSD) computed with respect to the other view as in (5) and λ, α, β weighting parameters.

Of the various colour segmentation formulations available for stereo matching, we have selected this particular approach for three main reasons. First, it maps easily to our ACTF estimation procedure, (5), by taking it as the definition of local match confidence. Second, by incorporating the proximity cue it discourages break-up of the aggregation window into sets of isolated pixels. Third, it degrades gracefully as colour becomes less important; e.g. for operation with gray-level only images, colour similarity, (8), is simply reduced to $\rho_c(x,y) = \sum_{(u,v)\in w(x,y)} |I(u,v) - I(x,y)|$.

2.5. Relation to other approaches

In general, it is well known that CTF disparity estimation corrupts 3D boundaries. In non-CTF block matching, use of shiftable or otherwise adaptive windows to conform to disparity discontinuities is well established [9–12,20]; however, the link to improving CTF disparity refinement seems not to have been stated previously. In our formulation, we can use quite small windows for better resolution of 3D boundary structure, as larger aggregation is made intrinsically available by CTF. This allows us to achieve the 3D boundary fitting robustness of overlapping windows [10], and avoid complicated construction of variable-sized windows [11]. Interestingly, the modification of shiftable windows used in [2] (referred there as Min Filter, an efficient implementation of [9]) is a special case of ACTF when the pyramid is degraded to a single level: In this case, each point has the same zero offset and ACTF automatically 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 [2,4,24–26], has yielded strong results; however, the importance of considering multiple offsets in projecting CTF has not been addressed clearly. Ideally, according to Section 2.2, 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.

The idea of using multiple offsets is not entirely new. In overlapped pyramid projection strategies they are used to overcome problems of nearest-neighbour interpolation [27]. Also related is recent application of CTF to dynamic programming [28], 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 implement in the dynamic programming framework, albeit with increased processing requirements. However, that work does not discuss multiple offsets, explicitly motivate their solution as we do or relate their approach to standard upsampling.

It is important to recognize the deeper problem of discontinuity pixel recovery as they are a blend of foreground and background surfaces. One way to ameliorate the attendant difficulties in disparity estimation is to make combined use of block-based and feature-based stereo, whereby feature-based processing is exploited to support accurate delineation of discontinuities even as block-based processing provides dense estimates in the intervening regions [29]. More explicit analysis of how different proportions of foreground, background and occluded regions within left and right match windows lead to contaminated match statistics has shown their ramifications for block-based matching, including foreground thickening [30]; this analysis led to the use of statistically robust match measures to combat the corruptions. For pixel-based stereo, various sample-insensitive measures have been introduced in the past [31,32] and explicit recovery of foreground and background mixture proportions has been attempted as well [33,34]. As Section 2.2 discusses, the problem is more serious for pyramid-based CTF methods as blur over relatively large spatial extent occurs. In this light, adaptive local aggregation would implicitly allow grouping of surfaces separated in depth. In fact, it will be shown in Section 3 that adaptivity in scale and space is beneficial even for global CTF stereo methods that theoretically do not require such aggregation.

The proposed approach to detecting half-occlusions emphasizes their processing within a CTF framework, a topic that appears to have received little previous consideration. Within this framework, we use two complementary sources of information to drive the processing. One source of information derives from explicit consideration of disparity relationships between occluded and occluding surfaces, e.g. as previously considered in terms of the occlusion [8,15], ordering [2,3,8] and visibility (an extension of uniqueness) [14,16] constraints. Indeed, our statement and use of the uniqueness constraint (7) also can be used to define the "forbidden zone" where match ordering is violated [35]. Moreover, our use of disparity equivalence classes that do not compete with one another under uniqueness as they are credited to a single continuous surface, is related to previous notions of disparity component matching, which relies on the assumption that images consist of connected sets of pixels with the same [36] or very similar [37] disparities. The other source of information that we employ derives from consideration of match quality, e.g. as previously considered when global methods set occlusion cost to depend on match scores [2,3] and inconsistent bidirectional matches (left-right checks) are used for occlusion detection [8,10]. While these two sources of information have been extensively researched with regard to half-occlusions, it appears that the particular combination that we employ and especially their cooperative CTF instantiation with disparity estimation is novel. In comparison to CTF use of left-right-checking, our proposed method yields superior results [21,23].

Finally, a number of state-of-the-art stereo correspondence algorithms use colour segmentation in a hard or soft manner to define match support [12,15,16]. Here, we largely incorporate a previous approach that maps easily to our adaptive CTF framework and provides it with the ability to guide match widow support according to colour uniformity [12,21]. Here, colour segmentation can guide match support, yet, is robust to situations when colour segmentation fails, e.g. highly-textured regions, as aggregation windows will not degenerate to constellations of disjoint pixels. In contrast to the original Gestalt-based colour-guided aggregation [12], which uses colour to guide construction of otherwise arbitrary pixel groupings, our formulation merely uses colour similarity to guide adaptive block matching.

3. Empirical evaluation

3.1. Experimental methodology

The adaptive CTF processing advances have been implemented in software and tested on a standard test set *Tsukuba*, *Venus*, *Teddy* and *Cones* [38] shown in Fig. 5 and a dataset with naturalistic imagery (albeit no ground truth), *Flower Garden* [39], *Rock* [40], *Stephen1* and *Stephen2* shown in Fig. 6 (Stephen images are novel stereo pairs). Overall, six different algorithmic instantiations are evaluated:

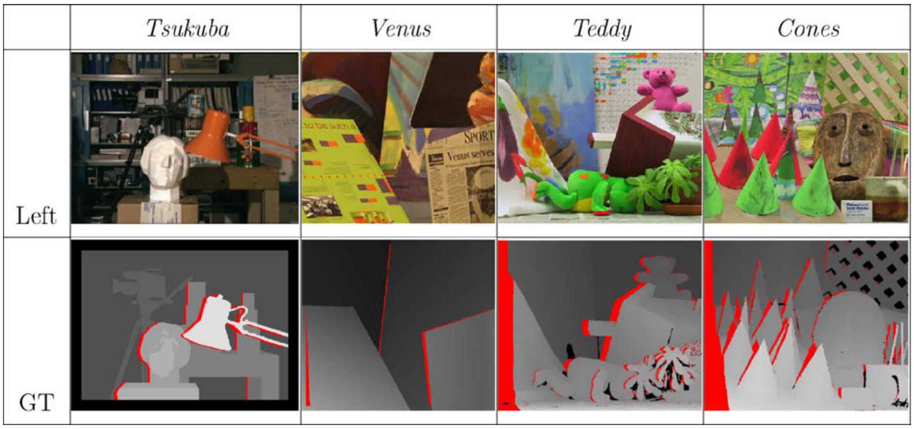


Fig. 5. Lab scenes from the Middlebury database [38]. From top to bottom: left image, disparity ground truth with half-occlusions. Disparity and occlusion ground truth are given with respect to left image. In disparity GT, brighter pixels mean larger disparity; in occlusion GT, red pixels denote half-occlusions.

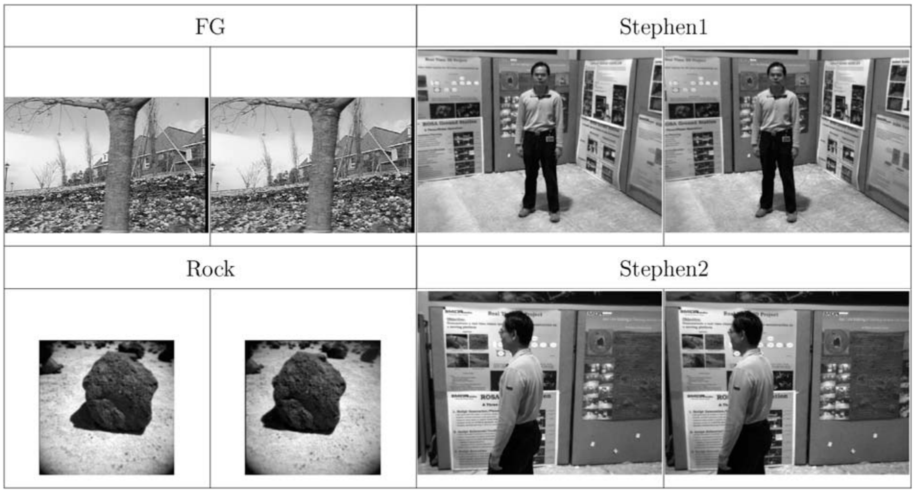


Fig. 6. Real world scenes: *Flower Garden (FG)*; *Rock*; *Stephen1*; *Stephen2*.

Algorithm	Properties
A1	Standard CTF
A2	Proposed ACTF
A3	**A2** with occlusion detection
A4	**A3** with colour-guided windows
A5	Single scale with shiftable windows
A6	Gestalt-based matcher [12] in CTF

Our major comparison is to A1 to show our improvements over standard CTF. All algorithms work on a Gaussian pyramid obtained from grayscale images (A5 works on the finest level only) and use the Normalized Cross-Correlation (NCC) match measure [3]; for match windows, A5 uses 17×17 shiftable windows (which gave the best result in initial tests), A1–A4 use 5×5 windows and work over all attainable pyramid levels for a given image size (i.e. coarsest level auto-selected when one image dimension becomes unity) and search ±1 pixel at each level; A5 searched the maximum disparity range for each test case. Note that parameters for CTF instantiations are selected based on theoretical considerations. The top pyramid level is the maximum that can be computed; search range is the smallest possible; 5x5 match windows are matched to the support in the standard kernel used in pyramid construction, $(1/16^2)[1 \quad 4 \quad 6 \quad 4 \quad 1]^\top[1 \quad 4 \quad 6 \quad 4 \quad 1]$.

We also compare CTF matchers A1–A4 to A5, as it is a strong single scale block matcher [2]. Additionally, comparison to A5 shows the benefits of CTF itself, because A5 can be seen as a special case of A2 where the pyramid is degenerated to a single level with maximum disparity search range. Moreover, we include a version of the adaptive support-weight approach [12] that is based on Gestalt principles but implemented in coarse-to-fine fashion (with adaptive offsets as described in Section 2.2.1) as A6. The original stereo matcher [12] is widely recognized as one of the best local matchers to date (exceptions outlined in Section 3.2) and comparison to this instance will exemplify the importance of CTF

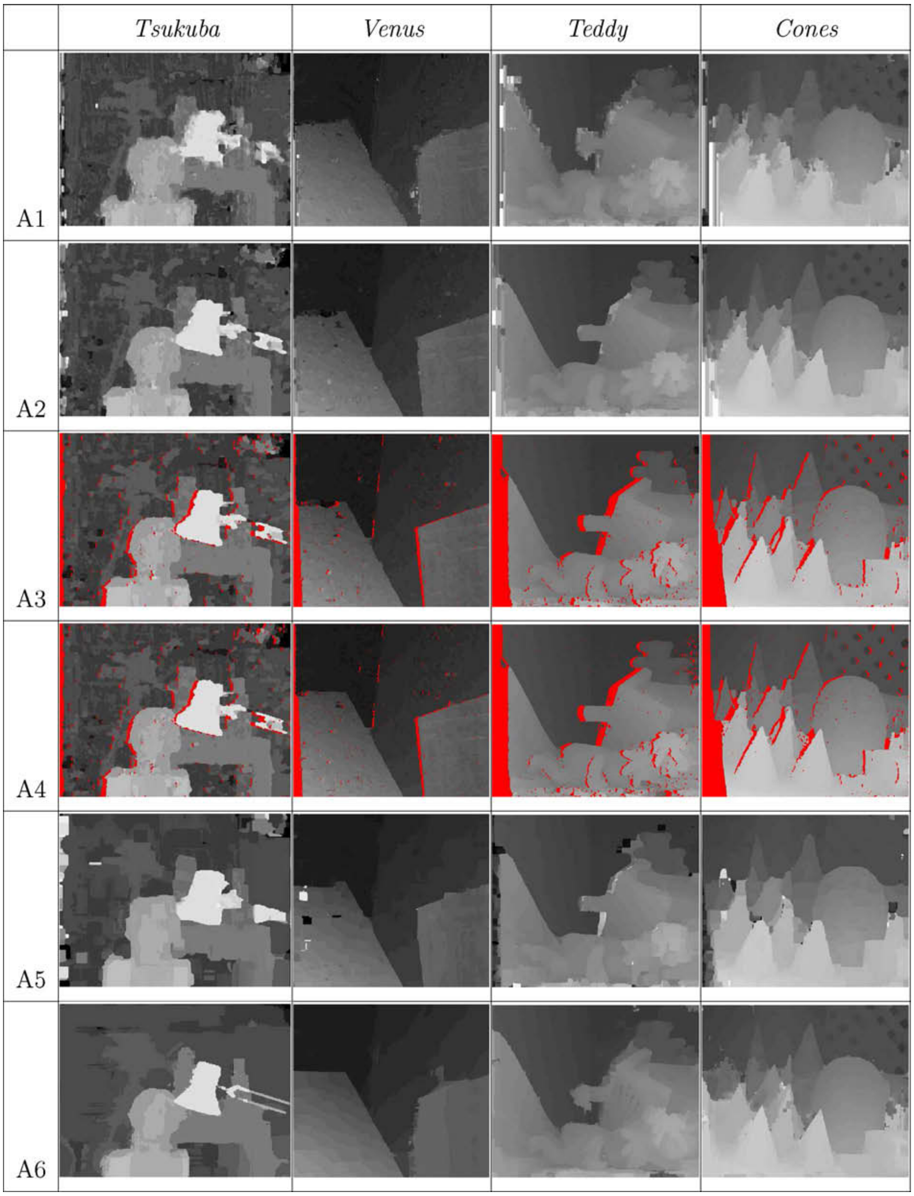


Fig. 7. Middlebury scenes disparity maps of algorithms A1–A6. Half-occlusions detected in A3 and A4 are extrapolated for disparity error analysis by taking disparity value from the background (occluded) surface.

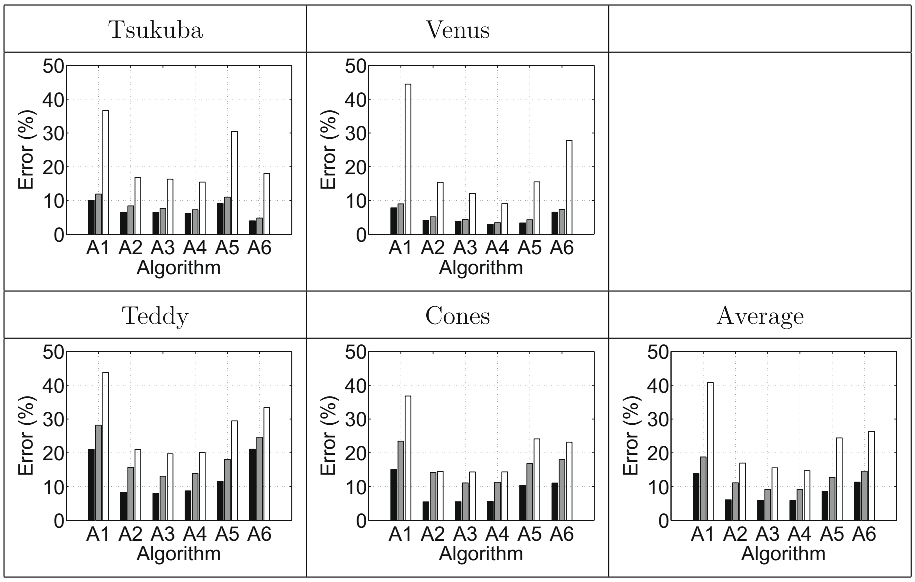


Fig. 8. Error statistics across algorithms A1–A6. Triplet bars represents error statistics for non-occluded (black), all (gray), and discontinuity (white) pixels, as defined in [38].

processing considerations raised in Section 2.2. Instantiation A6 uses ±1 pixel search range at each pyramid level and all other parameters are the same as in [12].[2] Recall that while [12] and A6 use colour to construct match windows support, the proposed approach, A4, uses colour only to weight pixel contributions during square window adaptation (6).

For data sets with ground truth, three kinds of error statistics have been collected [38]: errors for nonoccluded pixels, all pixels including occluded and pixels near discontinuities. Average statistics for each class of errors are given by taking the weighted average over all four stereo pairs, with weights proportional to the number of image pixels. Fig. 7 shows disparity maps while Fig. 8 shows the error statistics. Fig. 9 shows the disparity maps for real world images.

3.2. Adaptive CTF processing

Comparing A1 and A2, the introduction of the proposed ACTF processing results in considerable improvement. It is expected that the adaptive approach bests standard CTF, as it was designed for exactly that purpose. Interestingly, ACTF also bests single scale shiftable windows (A5), especially near discontinuities (white bars in Fig. 8 and qualitative improvement in real scenes, especially crispness of 3D boundaries in *Stephen1* and *Stephen2*); this can be explained by the fact that A2 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.

Nevertheless, ACTF (A2) cannot completely eliminate all disadvantages of CTF processing. For example, thin structures are still hard to recover precisely, e.g. lamp arms in *Tsukuba* and pencils in *Cones*; however, use of small windows results in better recovery of depth discontinuities that in non-CTF A5. Another apparent weakness of the CTF processing is the possible image border effect (disparity for the lower region on *Teddy*, region above the head in *Stephen1* and the whole upper-right corner of *Rock* are recovered incorrectly), when thin regions near image boundaries do not have enough spatial support and become lost at coarser scales.

The explicit consideration of the half-occlusion information (A3) shows even further improvement over ACTF (A2) alone by reducing errors in half-occluded regions, as expected, and, more generally, near 3D discontinuities (gray and white bars in Fig. 8, respectively). These results support the importance of the explicit occlusion refinement and the necessity of cooperation between disparity estimation and the half-occlusion detection in the CTF procedure. The benefit of half-occlusion detection is most evident when large disparity jumps are considered, especially in the real scenes (*Rock*, *Stephen1* and *Stephen2*), but also in the lab settings (*Teddy*, *Cones*). Interestingly, the developed approach also exhibits reasonable robustness to binocular slit viewing, as shown by the generally correct disparity estimates in the vicinity of the background lattice work in *Cones*. Here, even though the occlusion detection based on the disparity cue (7) is not always possible (background surface usually assigned the disparity of the foreground), neighbouring visible points are still assigned correctly, as local errors are not propagated. Still, it is worth noting that the major improvement of A3 over A1 is due to the adaptive window and offset approach (A2). See [21,23] for further related half-occlusion experiments.

The introduction of the colour-guided component in ACTF (A4) results in further improvement, specifically for Middlebury scenes. The set of parameters producing the best result has been chosen as $\alpha = 5$ and $\beta = 17.5$ (as in [12]) and $\lambda = 2 \times 10^{-4}$. Surprisingly, no visible gain of colour/intensity segmentation cue is observed for the natural scenes (Fig. 9). These results are 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 the lab scenes exemplified in the latest Middlebury dataset. A drastic counter-example, is the *Map* stereo pair from the old Middlebury dataset [38] shown in Fig. 10. This example is particularly easy for adaptive block-based matchers like our approach. At the same time, algorithms that rely too heavily on monocular colour segmentation, including the instantiation of [12] considered here, perform poorly in such situations [12,16,38], even though they are fundamental to multi-image matching.

[2] The original formulation of [12] calculates the match cost of a pixel by performing straight absolute colour difference for pixels in the aggregation neighbourhood (weighted by the colour similarity with the central pixel). For real stereo pairs that have significant radiometric differences, we calculate this matching cost contribution on band-passed images (Laplacian pyramid), while adaptive constructions of the support, i.e. weights, are still computed on full colour images (Gaussian pyramid).

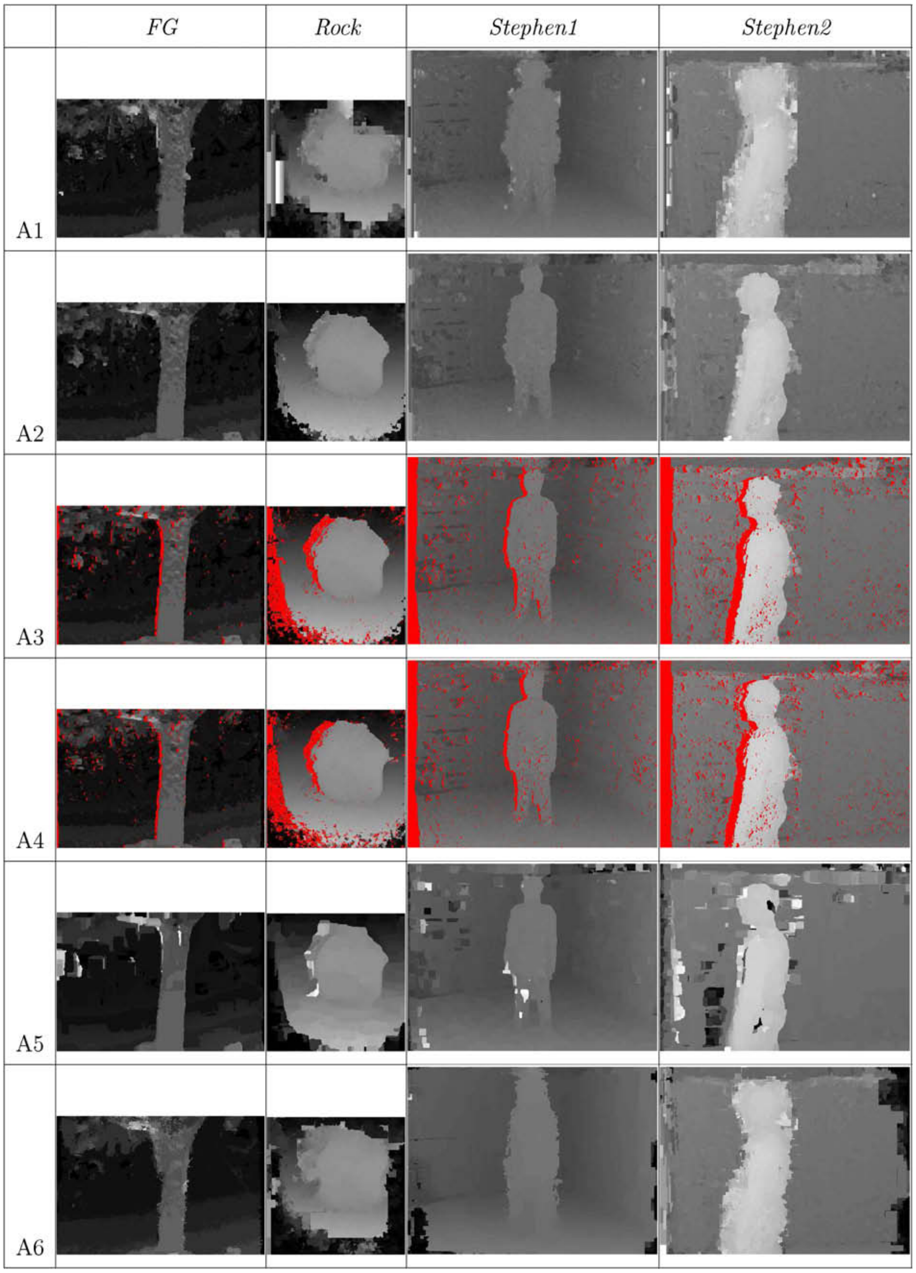


Fig. 9. Disparity recovered for real scenes using algorithms A1–A6.

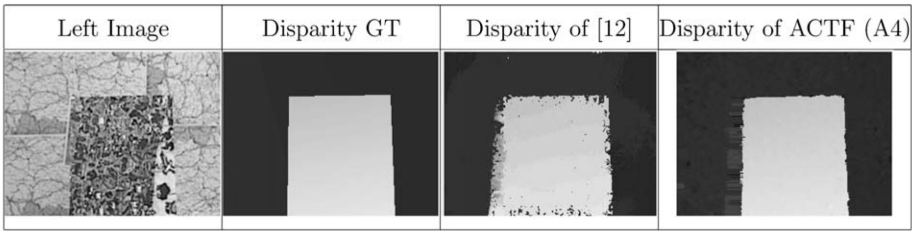


Fig. 10. Colour cue difficulties. From left to right: left image of *Map* dataset from the old version of [38]; Disparity ground truth; disparity recovered by the adaptive weight approach [12] (adapted from [38]); disparity map recovered by the ACTF.

Interestingly, ACTF (A4) exhibits superior performance to the CTF version of the adaptive support-weight approach [12], i.e. A6, in real scenes and even in the majority of colourful lab scenes. This fact supports the necessity of joint adaptation in scale and space that is embodied in A4 to combat the 3D boundary errors that are propagated during CTF processing. Still, A6 behaves better in the textureless regions because of its large window size. Finally, the approach of [12] and even its CTF version A6, though being local, are significantly more computationally intensive than the simpler block-based engine of A2–A4, which is a practically important consideration for this class of algorithms.

Finally, the running times of these local algorithms summed across all four Middlebury image pairs are presented below:

Algorithm	A1	A2	A3	A4	A5
Time (s)	0.989	1.193	1.246	2.851	14.636

The adaptive method (A2) adds a very modest extra burden to the basic CTF processing (A1); at the same time, the disparity results are improved significantly, as documented above. Half-occlusion processing (A3) increases the processing time even less in comparison to (A2). The introduction of the colour segmentation in (A4) is more computationally involved in comparison to basic shiftable windows and offsets method of A2 and A3, but even in this case the algorithm is only twice slower than A3, as it has the same computational complexity. Finally, all CTF algorithms (A1–A4) are an order of magnitude faster than the single-scale counterpart A5, because the latter is an algorithm of well-known higher computational complexity. Formal complexity analysis is considered further in Sections 3.4 and 4.2.1.

3.3. CTF global stereo

Though the major effort of the current work was devoted to efficient block-based CTF matching, the proposed advances are also beneficial for global methods formulated in CTF fashion. As an illustrative example, the well-known Graph Cuts stereo matcher with occlusion detection [13] is considered; its performance results are shown when implemented in CTF with and without adaptive offsets and windows.

Algorithm	Properties
G1	GC CTF with NN disparity upsampling
G2	GC CTF with adaptive offsets
G3	GC ACTF with adaptive 3×3 windows and offsets
G4	Original single-scale GC

Similarly to Section 3.1, we consider the algorithmic instantiations G1–G4 outlined above and test them on the Middlebury dataset. For G1, G2 and G4 the sample-insensitive measure [31] is used, as it is essential to reduce the CTF boundary blurring phenomenon discussed in Section 2.2; for G3, SAD is used as 3×3 adaptive windows serve to combat boundary degradation. CTF instantiations G1–G3 are computed with $\Delta d = 1$ search range. All other parameters related to the GC computation are kept the same for all algorithms G1–G4.

Figure 11 shows the obtained maps, while Fig. 12 shows the corresponding error statistics. As expected, qualitatively and quantitatively, the use of multiple offsets (G2) is superior to standard NN upsampling (G1). However, introduction of small adaptive aggregation windows (G3) produce even better results in comparison to pixel-based matching especially near 3D discontinuities, which supports the claims made in Section 2 regarding the superiority of simultaneous adaptation in scale and space. In general, G3 yields 3D boundary outline quality similar to the traditional single-scale version (G4) and produces fewer spurious matches. The only apparent disadvantage of G3 in comparison to G4 is the inferior performance near thin structures (e.g. pencils in *Cones*), which is a standard flaw of CTF processing that requires further enhancements for significant improvements. Importantly, progress has been made improving CTF recovery of thin structures by exploiting novel image pyramid representations [41].

Finally, the running times of these global algorithms summed across all four Middlebury image pairs are presented below:

Algorithm	G1	G2	G3	G4
Time (s)	76	78	77	259

As expected, the difference between CTF (G1–G3) and single-scale (G4) versions is the most substantial, as CTF versions, having fundamentally lower complexity, run only a fraction of the time needed to process the full disparity space. At the same time, the performance of the proposed adaptive CTF processing (G3) adds little computation burden[3] to the standard G1, but significantly improves the results (as discussed above). Interestingly, G3 with adaptive offsets and windows is also slightly faster than G2 with adaptive offsets only, which is explained by the efficient implementation of ACTF procedure (embedded in G3), as detailed in Section 2.2.2.

3.4. Variation of parameters

The results shown in Figs. 7 and 9 were obtained using the set of parameters theoretically motivated in Section 3.1: ±1 local search range, full pyramid, 5×5 aggregation windows. Here, we investigate the behaviour of the ACTF algorithm (A4) while varying those parameters. In particular, we show in Fig. 13 the quantitative results of varying window size, local disparity search range, match measure and parameters for the colour cue averaged over the four scenes in the Middlebury dataset. Experimentally, no significant variation in performance is noticed. Specifically, increasing the window size has the predicted behaviour of increasing errors near discontinuities, while not reducing overall error, because the CTF procedure intrinsically allows greater aggregation on coarse levels.[4] Increasing the local search range also shows the well-understood tendency of reducing discontinuity errors (CTF is becoming more robust in recovering finer details), while increasing the overall error rating (due to increasing danger of converging to a false match). The choice of the correct match measure is usually guided by properties of the sensor that acquires the data: in our case NCC is superior to SSD (using Laplacian pyramid to ameliorate the need for normalization) and Mutual Information (MI) (the MI-matching is similar to the procedure described in [42]; details are provided in [21]). Finally, choosing a reasonable parameter for colour segmentation, λ, is not difficult, as its performance varies predictably: errors are higher if the colour cue is weighted too little (small λ) or too much (large λ).[5] Overall, the algorithm is very stable with respect to the theoretically motivated choice of parameter values, which also yield the best overall results.

3.5. Final remarks

The *critical comparison* is that of A4 to standard CTF (A1), as a major goal of the present work is improved disparity estimates

[3] The difference in runtimes between G3 and G1 is much less noticeable than in comparison of local matchers A2 and A1 (see Section 3.2), since the running time is dominated by the graph cuts computation.

[4] Note that windows of size 3×3 perform unsatisfactory because support is too small for a local winner-take-all procedure. More importantly, they are smaller than the size of the Gaussian kernel 5×5 and, even in theory, cannot overcome the possible boundary overreach of the intrinsic aggregation.

[5] The parameter λ is argued as of primary importance as it directly controls the strength of the colour cue in the adaptive CTF matching, while α and β control the proportion between colour similarity and proximity components. Refer to [12] for discussion of variation in α and β.

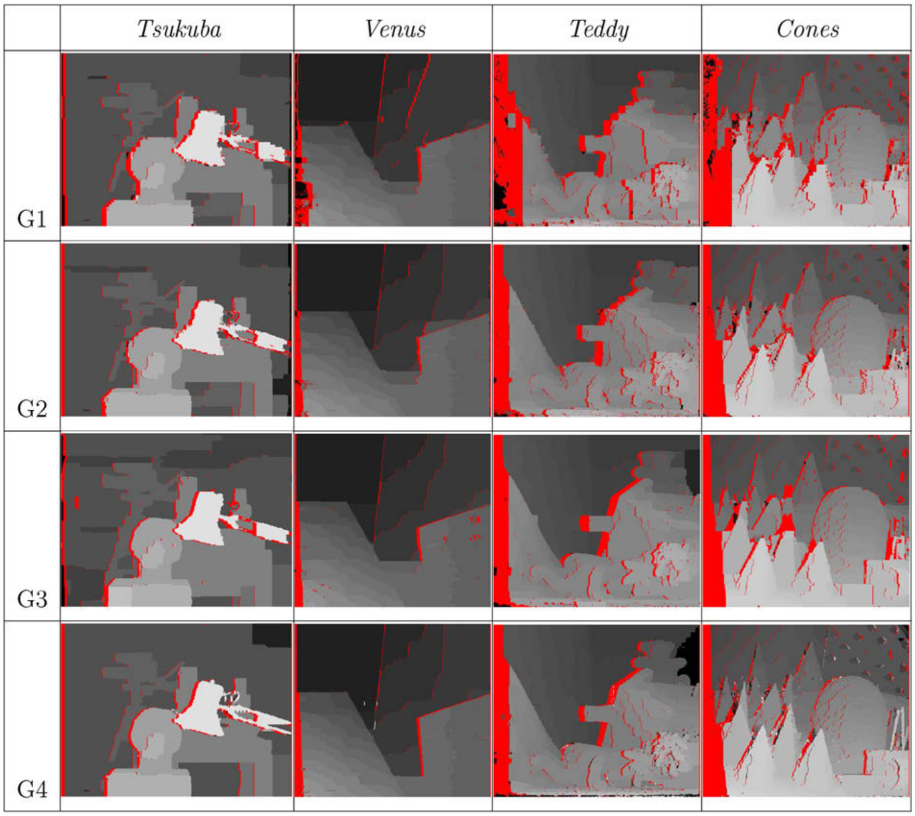


Fig. 11. Middlebury scenes disparity maps of algorithms G1–G4. Red pixels denote half-occlusions.

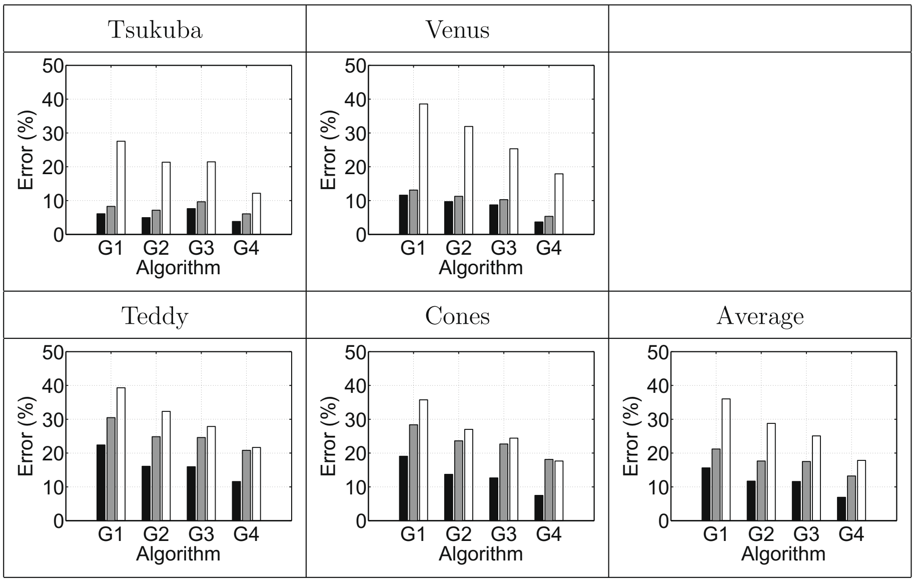


Fig. 12. Error statistics across algorithms G1–G4. Triplet bars represents error statistics for non-occluded (black), all (gray), and discontinuity (white) pixels, as defined in [38].

for this style of efficient processing; such improvement is clear in Figs. 7 and 9, e.g. average errors reduced by a factor of two or more for all error classes plotted in Fig. 8 and dramatic qualitative improvement in the images of real scenes.

Finally, speed is an important advantage of any CTF algorithm including the proposed algorithm. For an image of n pixels, search range d and match window size w^2, the theoretical complexity is

$$O(ndw^2) + O\left(\frac{n}{4}dw^2\right) + O\left(\frac{n}{16}dw^2\right) + \cdots < \sum_{i=0}^{\infty} \frac{O(ndw^2)}{4^i} = \frac{O(ndw^2)}{1-1/4} = O(ndw^2), \quad (11)$$

which in our case reduces to $O(nw^2)$, because search range at each pyramid level is $d = 1$ for A2–A4 in all reported experiments, and

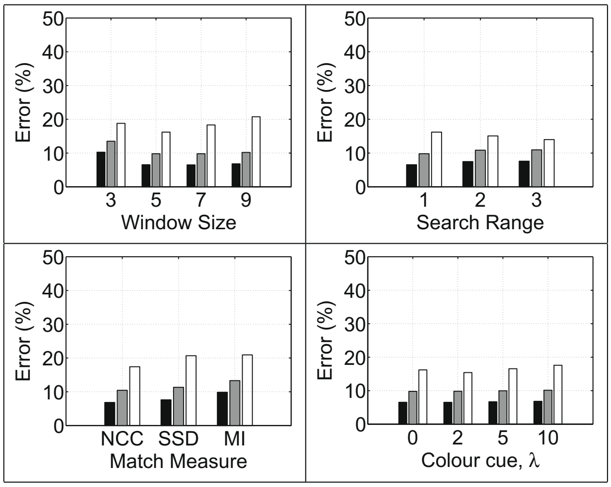


Fig. 13. Error statistics for various parameter variations. Triplet bars represents error statistics for non-occluded (black), all (gray), and discontinuity (white) pixels, as defined in [38].

can be decreased to $O(n)$ via a running box filter implementation for window cost aggregation [25,28]. Importantly, the advances over standard CTF that are embodied in ACTF do not degrade this complexity (e.g. implementation of shiftable windows as in [2] via morphological operation).

The runtimes documented in Sections 3.2 and 3.3 are reported for the current implementation realized in unoptimized C++ without any special instructions on a 3.0 GHz P4. As another example, matching of 1024 × 768 images (*Stephen1* and *Stephen2*) with A4 takes only 0.781 s, vs. 24.73 s needed to execute the single-scale A5 (note that larger images have inherently larger disparity search range, which explains the growing gap between running times of A4 and A5, as they are of different computational complexities). 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 fast implementation of the coarse-to-fine algorithm, but rather on analysis and improvement of its accuracy, while preserving its inherently low complexity.

4. Discussion

4.1. 3D boundary recovery within the CTF framework

This paper has presented extensive analysis of CTF stereo processing with special emphasis on local, block-based matching. As results of this analysis, the main sources of error are identified – a well-known foreground fattening/shrinking artifact of block-based matching [10,43] and the propagation of incorrect disparity estimates near 3D boundaries during coarse-to-fine projection. Further, the analysis has led to a novel combination of adaptive windows and adaptive offsets (adaptation in space and scale), which has empirically shown significant improvement over standard CTF block matching stereo and single-scale stereo matching with shiftable windows [2], especially in the vicinity of 3D boundaries. While the proposed ACTF procedure is simple and straightforward, the extant literature does not provide a similar analysis of coarse-to-fine processing nor document a corresponding algorithm.

The half-occlusion phenomenon is one of the toughest sources of error for computational stereo [3,8]. In response, we formulate matters as a cooperative process that interleaves disparity and half-occlusion estimation across coarse-to-fine refinement. Even though the ACTF scheme itself (Section 2.2.2) is robust in the vicinity of half-occlusions, an explicit treatment of such regions is shown to yield extra benefit, which is documented extensively. Significantly, we find that our approach is of particular benefit in the processing of real world scenes, as shown in Fig. 9.

The disparity produced by our algorithm can be exploited directly, or can be used to provide input to more computationally intensive estimators that can benefit from reliable initial estimates, e.g. various global optimization procedures. Moreover, the presented analyses and techniques can be adapted and incorporated into other CTF disparity estimators, both local and global, as well as extended to optical flow recovery.

4.2. Speed-accuracy tradeoff

A major motivation for the present work is to improve speed-accuracy tradeoffs in computational stereo; correspondingly, we now explicitly compare our proposed algorithm to various alternatives along these dimensions. To quantify accuracy, we follow current practice and rely on error percentage [2,44,45], e.g. percentage of pixels in the image where recovered disparity value differs by more than 1 from the ground truth. To quantify speed, we chose computational and memory complexity. These complexity measures are independent of implementation details, which is appropriate given that the algorithms must be optimized differently and may be implemented by different researchers. Note, however, that complexity is not a sufficient measure of performance, as various computations can be run in parallel, which would significantly lower the final processing time – an issue that we address subsequently.

In the remainder of this section, analysis will focus on correspondence procedures that rely on image intensity-based matching. While some recent matchers employ various additional sources of information (e.g. colour) and/or postprocessing (e.g. plane-fitting), the methods discussed could be augmented to exploit such information as appropriate (e.g. as we discussed with respect to colour and our own approach in Section 2.4), with additional computational cost. We also neglect tree-reweighted message passing [46], as it is not yet widely used, while its performance is rather similar to graph cuts and implementations are slower [45].

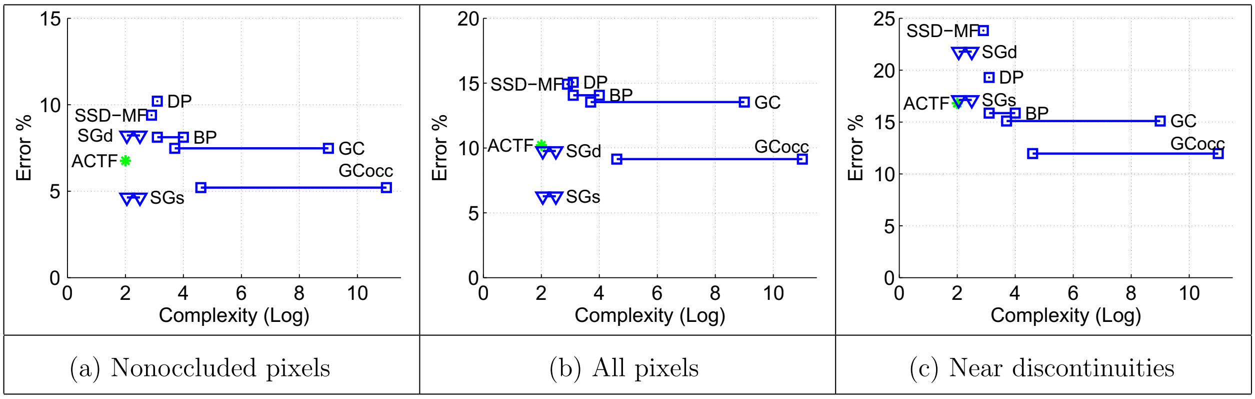
4.2.1. Time complexity

Representative algorithms and their time complexities are outlined in Table 1. The complexity itself is expressed in n (number of pixels), d (disparity levels), k (number of iterations), m (number of processed disparity candidates). Brief explanations follow.

- The proposed ACTF: Section 3 derived the complexity as $O(n)$.
- Conventional block-matching with shiftable windows (Block-SW) as in [2]: Its complexity is trivially $O(nd)$, because it makes a single pass over the whole disparity image space (DSI).

Table 1
Major stereo algorithms and their complexity.

Algorithm	Complexity (reported)	Complexity (adapted)
ACTF	$O(n)$	$O(N^2)$
Block-SW	$O(nd)$ [2]	$O(N^3)$
DP	$O(nd)$ [42]	$O(N^3)$
BP	$O(ndk)$ [4]	$O(N^3) : O(N^4)$
GC	$O(n^{1.2}d^{1.3}) : O(n^3d^3)$ [3]	$O(N^{3.7}) : O(N^9)$
GCocc	$O(n^{1.2}d^{2.2}) : O(n^3d^5)$	$O(N^{4.6}) : O(N^{11})$
SG	$O(nm)$ [52]	$O(N^2) : O(N^3)$



(a) Nonoccluded pixels (b) All pixels (c) Near discontinuities

Fig. 14. Time complexity–accuracy tradeoff of major dense stereo algorithms. Plot shows the order of complexity ($\log_N \mathscr{C}$) on abscissa vs. 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 asterisks and lies to the lower left side of the region formed by other basic dense stereo algorithms (marked with squares with initials defined in Table 1), which shows the outstanding speed-accuracy tradeoff characteristics of ACTF. Seed growing algorithms (marked with triangles and initials SGs vs. SGd for relatively sparse (65–70%) and somewhat denser (85–90%) estimates, respectively) also yield good speed-accuracy tradeoffs, but do so at the expense of density, as they typically produce only sparse disparity estimates.

- Dynamic programming (DP) as in [2]: The naive implementation of DP is $O(nd^2)$ but only $O(nd)$ when the distance transform is used [4].
- Belief Propagation (BP) as in [47]: The original BP for stereo is $O(nd^2k)$ [48], but is reducible to $O(ndk)$ by application of a distance transform [4].
- Graph Cuts (GCs) as in [49]: The worst-case complexity of GC is quite poor, and depending on the algorithm can be, for example, $O(Vertex \times Edges^2) = O(nd(nd)^2) = O(n^3d^3)$ if push-relabel maximum flow algorithm is adopted [50]. In turn, the average complexity is harder to predict; some authors observed close to linear dependence on n and d [50]; we take apparently the tightest result reported in the computer vision literature, $O(n^{1.2}d^{1.3})$ [3,51].
- Graph Cuts with occlusions (GCocc) as in [13]: The complexity of GCocc 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 $O(Vertex \times Edges^2) = O(nd(nd^2)^2) = O(n^3d^5)$ and expected as $O\left(n^{1.2}d^{1.3\frac{5}{3}}\right) = O(n^{1.2}d^{2.2})$.
- Seed growing (SG) as in [52] (the latest advancement of typical representatives [53,54]). Unlike the other algorithms considered, which return dense estimates of disparity, seed growing methods tend to return sparser estimates. This class of algorithms propagates the disparity map from initially computed seed points that typically are taken at loci of distinctive features (e.g. corners [55]) and consider only a small set of disparities, m, out of the possible range, d. Complexity is thus $O(nm)$.

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 so, we 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 $n = O(N^2)$. Second, 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 can assume that it depends linearly on N. Thus, $d = O(N)$. Belief propagation is an iterative algorithm[6] and the number of iterations k depends 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 structured regions has to be propagated over large image areas. Moreover, the iterations are dependent on the message update schedule, e.g. hierarchical belief propagation can yield constant overhead iterations [4]. Thus, the impact of iterations can range from constant overhead to the order of image size, i.e. $k \in [O(1), O(N)]$. Finally, the complexity of the seed growing method is $O(mN^2)$. The actual m will largely depend on the complexity of the scene and might even be proportional to d; typically, it is relatively small and is regarded as a constant,[7] as discussed in [52]. Thus, we consider the complexity of the algorithm as the range $O(N^2) : O(N^3)$.

The adapted complexity functions, $\mathscr{C}$, in terms of argument N are shown in the third column of Table 1. Fig. 14 shows the plot of complexity vs. performance for the major computational algorithms considered. Note that the abscissa has a logarithmic scale. 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 [38], with one exception. The exception is for the case of seed growing, where it appears that quantitative results have not been reported on a standard data set. To fill this void, a publically available algorithmic instantiation of [52] has been executed on the test images shown in Fig. 5. Since this algorithm returns variable densities of disparity, results are reported at two different settings with a resulting trade-off in density vs. accuracy.[8]

In Fig. 14, algorithms that are closer to the origin are desirable as they provide accurate results at a reasonable expense (low error rates with low computational complexity). Analyzing Fig. 14, all standard algorithms (marked with squares) exhibit a consistent tendency of better performance (lower error rates) at the expense of higher computational complexity. The proposed ACTF (marked with asterisk) lies to the lower left side of the cloud of standard algorithms, which signals its strong combination of low error rate and very low complexity. Of the algorithms considered, seed growing provides the only direct competition to ACTF in terms of speed-accuracy trade-off. Indeed, in certain situations, e.g. errors at non-occluded and across all pixels at low density, it can yield superior

[6] Graph cuts for stereo is iterative too, but few iterations are required, e.g. two or three [13,49].

[7] The actual complexity also depends on the particular implementation and data structures in use – the implementation of sparse disparity spaces using binary search trees for each reference point makes the complexity $O(nm \log m)$ [52,53].

[8] Density in the employed seed growing algorithm depends on a correlation growing threshold, which arbitrates whether a local match is propagated according to its correlation value (i.e. parameter τ in [52]).

accuracy at comparable complexity. Significantly, however, these results come at the expense of density. In particular, the densities of the more accurate SGs are only 65.07%, 70.62% and 68.78% for nonoccluded, all and near-discontinuity points, respectively. The corresponding numbers for denser, albeit inferior accuracy, SGd are 91.0%, 84.6% and 85.21%. Given the present emphasis on efficient computation with accurate 3D boundaries, it is especially interesting to note that the denser version of seed growing (SGd) still only provides approximately 85% density near discontinuities; whereas, the proposed ACTF provides notably improved accuracy near discontinuities with maximally dense (100%) estimates. Moreover, while the theoretical complexity of ACTF and seed growing is comparable, ACTF enjoys the advantage of a simpler practical design and implementation as it does not rely on heterogeneous two-stage matching (i.e. seed matching followed by match propagation), which should yield increased practical efficiency for ACTF, less dependence on parameters and easier mapping onto parallel hardware.

4.2.2. Memory complexity

All single-scale algorithms, including the ones shown in the plot of Fig. 14, except seed growing methods, require the construction of the entire disparity image space (DSI) [2], which means that they have at least $O(nd) = O(N^3)$ memory complexity to store the DSI and operate on it.[9] In comparison, pyramid-based CTF approaches, as the one presented here, require only $O(n) = O(N^2)$ space because CTF does not construct the complete DSI. Seed growing methods also are reported to maintain $O(N^2)$ complexity [52,53].

4.2.3. Parallelization

The analysis of computational complexity and performance is usefully complemented by discussion of the degree to which computations can be parallelized. Parallelization results in much more efficient utilization of hardware capabilities and ultimately allows faster and, in certain cases, real-time processing. Of related interest are recent efforts in realizing stereo computations on commodity graphics hardware (see, e.g. [56] and references cited therein) and FPGAs [57].

For stereo, block-based matching is readily parallelized as it is local, and the presented CTF block-based matching procedure naturally possesses 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 the number of scales is logarithmic with respect to the image size. Further, pipeline architectures are well suited to CTF processing, and existing systems already provide real-time performance [7]. In contrast, the DP approaches, though also possessing relatively low theoretical complexity, are parallelizable only up to a scanline (or corresponding assumed Markov Chain). For the case of BP-based 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, DP and BP can yield real-time performance using graphics hardware (together with the CPU working in parallel), albeit on rather small images and coarsely quantized disparity (320×256 with 16 disparity levels) [56,58].

4.3. 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 free and ultimately hand-tuned.[10] Even basic global formulations require certain intrinsic 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.

In contrast, the proposed ACTF algorithm has virtually no parameters to tune. Window size, which is typically the major and critical choice for stereo algorithms, is kept small (5×5) to allow precise boundary localization, while greater support aggregation is available by using coarser resolutions. The ability of ACTF to recover from errors made at the coarser levels, allows it to use the full pyramid, which, in turn, allows for restriction to the smallest search range ±1 for the fastest search and least ambiguous matching. Moreover, this specification was able to produce the best results for both lab and real world scenes, datasets with widely varying characteristics.

5. Conclusion

Standard coarse-to-fine, block-matching algorithms form the basis of the most resource efficient stereo correspondence approaches; however, traditionally such methods have been hampered by poor performance in the vicinity of 3D boundaries. In response this state of affairs, this paper has presented simple, effective procedures for improving disparity estimates near 3D boundaries within the coarse-to-fine (CTF), block-matching paradigm. The procedures entail adaptive CTF refinement that avoids corruption of disparities across 3D discontinuities and accurate half-occlusion recovery. Empirical evaluation of an embodiment of these advances in a CTF, block-matcher shows its superior performance in comparison to the same matcher without the proposed advances. Significantly, the enhanced disparity estimator enjoys the same efficient style of computation as does standard CTF, block-matching. Furthermore, we have demonstrated the usefulness of the proposed advances when embedded in global CTF matchers (graph cuts with occlusions [13]). In practice, the proposed advances should have considerable utility owing to their efficient, effective nature, small number of parameters and applicability to any CTF matcher.

References

[1] A. Agarwal, 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, pp. 2339–2346.

[2] D. Scharstein, R. Szeliski, Taxonomy and evaluation of dense two-frame stereo correspondence algorithms, International Journal of Computer Vision 47 (2002) 7–42.

[3] M.Z. Brown, D. Burschka, G.D. Hager, Advances in computational stereo, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (8) (2003) 993–1008.

[4] P.F. Felzenszwalb, D.P. Huttenlocher, Efficient belief propagation for early vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2004, pp. 261–268.

[5] R. Yang, M. Pollefeys, Multi-resolution real-time stereo on commodity graphics hardware, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2003, pp. 211–217.

[6] N. Cornelis, L. van Gool, Real-time connectivity constrained depth map combination using programmable graphics hardware, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 1099–1104.

[7] G. van der Wal, M. Hansen, M. Piacentino, The Acadia vision processor, in: Proceedings of the International Workshop on Computer Architecture for Machine Perception, Padua, Italy, 2000, pp. 31–40.

[8] G. Egnal, R.P. Wildes, Detecting binocular half-occlusions: empirical

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

[10] As an example, one of the state-of-the-art stereo systems by Sun et al. [15] has five free parameters without colour segmentation + 1 when segmentation is used, not including the free parameters needed to obtain the segmentation itself.

comparisons of five approaches, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (8) (2002) 1127–1133.

[9] A. Fusiello, V. Roberto, Efficient stereo with multiple windowing, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1997, pp. 885–863.

[10] H. Hirschmuller, P.R. Innocent, J. Garibaldi, Real-time correlation-based stereo vision with reduced border errors, International Journal of Computer Vision 47 (2002) 229–246.

[11] O. Veksler, Fast variable window for stereo correspondence using integral images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2003, pp. 556–561.

[12] K.-J. Yoon, I.S. Kweon, Adaptive support-weight approach for correspondence search, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (4) (2006) 650–656.

[13] V. Kolmogorov, R. Zabih, Computing visual correspondence with occlusions using graph cuts, in: Proceedings of the IEEE International Conference on Computer Vision, 2001, pp. 508–515.

[14] A.S. Ogale, Y. Aloimonos, Stereo correspondence with slanted surfaces: critical implications of horizontal slant, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2004, pp. 568–573.

[15] J. Sun, Y. Li, S.B. Kang, H.-Y. Shum, Symmetric stereo matching for occlusion handling, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 399–406.

[16] Y. Deng, Q. Yang, X. Lin, X. Tang, A symmetric patch-based correspondence model for occlusion handling, in: Proceedings of the IEEE International Conference on Computer Vision, vol. 2, 2005, pp. 1316–1322.

[17] P.J. Burt, E.H. Adelson, The Laplacian pyramid as a compact image code, IEEE Transactions on Communications 31 (4) (1983) 532–540.

[18] B. Jahne, Digital Image Processing: Concepts, Algorithms and Scientific Applications, Springer, Berlin, 1993.

[19] T. Lindeberg, Scale-Space Theory in Computer Vision, Kluwer Academic Publishers, Boston, 1994.

[20] T. Kanade, M. Okutomi, A stereo matching algorithm with an adaptive window, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (1994) 920–932.

[21] M. Sizintsev, R.P. Wildes, Coarse-to-fine stereo vision with accurate 3D boundaries, Tech. Rep. CS-2006-07, York University, Toronto, Canada, 2006.

[22] M. Okutomi, Y. Katayama, S. Oka, A simple stereo algorithm to recover precise object boundaries and smooth surfaces, International Journal of Computer Vision 47 (2002) 261–273.

[23] M. Sizintsev, R.P. Wildes, Efficient stereo with accurate 3-D boundaries, in: Proceedings of the British Machine Vision Conference, 2006, pp. 237–246.

[24] G.V. Meerbergen, M. Vergauwen, M. Pollefeys, L. van Gool, A hierarchical symmetric stereo algorithm using dynamic programming, International Journal of Computer Vision 47 (2002) 275–282.

[25] C. Sun, Fast stereo matching using rectangular subregioning and 3D maximum-surface techniques, International Journal of Computer Vision 47 (2002) 99–117.

[26] S. Forstmann, Y. Kanou, J. Ohya, S. Thuering, A. Schmitt, Real-time stereo by using dynamic programming, in: Proceedings of the IEEE Computer Vision and Pattern Recognition Workshop, vol. 3, 2004, pp. 29–35.

[27] P. Anandan, A computational framework and an algorithm for the measurement of visual motion, International Journal of Computer Vision 2 (1989) 283–301.

[28] C. Leung, B. Appleton, C. Sun, Fast stereo matching by iterated dynamic programming and quadtree subregioning, in: Proceedings of the British Machine Vision Conference, 2004, pp. 97–106.

[29] S.D. Cochran, G.G. Medioni, 3-D surface description from binocular stereo, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (10) (1992) 981–994.

[30] R. Sara, R. Bajcsy, On occluding contour artifacts in stereo vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1997, pp. 852–857.

[31] S. Birchfield, C. Tomasi, A pixel dissimilarity measure that is insensitive to image sampling, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (4) (1998) 401–406.

[32] R. Szeliski, D. Scharstein, Sampling the disparity space image, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (3) (2004) 419–425.

[33] R. Szeliski, P. Golland, Stereo matching with transparency and matting, International Journal of Computer Vision 32 (1999) 45–61.

[34] W. Xiong, J. Jia, Stereo matching on objects with fractional boundary, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[35] M. Drumheller, T. Poggio, On parallel stereo, in: Proceedings of the IEEE Conference on Robotics and Automation, 1986, pp. 1439–1448.

[36] Y. Boykov, O. Veksler, R. Zabih, Disparity component matching for visual correspondence, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1997, pp. 470–475.

[37] J. Kostkova, R. Sara, Stratified dense matching for stereopsis in complex scenes, in: Proceedings of the British Machine Vision Conference, 2003, pp. 339–348.

[38] Middlebury College Stereo Vision Page, 2008. Available from: <http://www.middlebury.edu/stereo/>.

[39] Brown University Image Sequences, 2006. Available from: <http://www.cs.brown.edu/people/black/images.html>.

[40] S.B. Goldberg, M.W. Maimone, L. Matthies, Stereo vision and rover navigation software for planetary exploration, in: Proceedings of the IEEE Aerospace Conference, vol. 5, 2002, pp. 2025–2036.

[41] M. Sizintsev, Hierarchical stereo with thin structures and transparency, in: Proceedings of the Canadian Conference on Computer and Robot Vision, 2008, pp. 97–104.

[42] H. Hirschmuller, Stereo vision in structured environments by consistent semi-global matching, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2006, pp. 2386–2393.

[43] M. Shimizu, M. Okutomi, Precise subpixel estimation on area-based matching, Systems and Computers in Japan 33 (7) (2002) 1409–1418.

[44] R. Szeliski, R. Zabih, D. Scharstein, O. Veskler, V. Kolmogorov, A. Agarwala, M. Tappen, C. Rother, Comparative study of energy minimization methods for Markov random fields, in: Proceedings of the European Conference on Computer Vision, vol. 2, 2006, pp. 16–29.

[45] V. Kolmogorov, C. Rother, Comparison of energy minimization algorithm for highly connected graphs, in: Proceedings of the European Conference on Computer Vision, vol. 2, 2006, pp. 1–15.

[46] V. Kolmogorov, Convergent tree-reweighted message passing for energy minimization, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2006) 1568–1583.

[47] M.F. Tappen, 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, 2003, pp. 900–907.

[48] J. Sun, N.-N. Zheng, H.-Y. Shum, Stereo matching using belief propagation, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (7) (2003) 787–800.

[49] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (11) (2001) 1222–1239.

[50] Y. Boykov, 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) (2004) 1124–1137.

[51] S. Roy, I.J. Cox, A maximum-flow formulation of the N-camera stereo correspondence problem, in: Proceedings of the IEEE International Conference on Computer Vision, 1998, pp. 492–502.

[52] J. Cech, R. Sara, Efficient sampling of disparity space for fast and accurate matching, in: Proceedings of the BenCOS Workshop CVPR, 2007. Available from: <http://cmp.felk.cvut.cz/~cechj/GCS/>.

[53] M. Lhuillier, L. Quan, Match propagation for image-based modeling and rendering, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (8) (2002) 1140–1146.

[54] R. Sara, Finding the largest unambiguous components of stereo matching, in: Proceedings of the European Conference on Computer Vision, vol. 3, 2002, pp. 900–914.

[55] C. Harris, M. Stephens, A combined corner and edge detector, in: Proceedings of the Alvey Vision Conference, 1988, pp. 147–152.

[56] Q. Yang, L. Wang, R. Yang, S. Wang, M. Liao, D. Nister, Real-time global stereo matching using hierarchical belief propagation, in: Proceedings of the British Machine Vision Conference, 2006, pp. 989–998.

[57] A. Darabiha, W.J. MacLean, J. Rose, Reconfigurable hardware implementation of a phase-correlation stereo algorithm, Machine Vision and Applications Journal 17 (2006) 116–132.

[58] L. Wang, M. Liao, M. Gong, R. Yang, D. Nister, High-quality real-time stereo using adaptive cost aggregation and dynamic programming, in: Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission, 2006, pp. 798–805.