Fast Stereo Matching Using Reliability-Based Dynamic Programming and Consistency Constraints

Minglun Gong and Yee-Hong Yang
Dept. of Computing Science, Univ. of Alberta, Edmonton, AB, Canada
{minglun,yang}@cs.ualberta.ca

Abstract

A method for solving binocular and multi-view stereo matching problems is presented in this paper. A weak consistency constraint is proposed, which expresses the visibility constraint in the image space. It can be proved that the weak consistency constraint holds for scenes that can be represented by a set of 3D points. As well, also proposed is a new reliability measure for dynamic programming techniques, which evaluates the reliability of a given match. A novel reliability-based dynamic programming algorithm is derived accordingly, which can selectively assign disparity values to pixels when the reliabilities of the corresponding matches exceed a given threshold. Consistency constraints and the new reliability-based dynamic programming algorithm can be combined in an iterative approach. The experimental results show that the iterative approach can produce dense (60~90%) and reliable (total error rate of 0.1~1.1%) matching for binocular stereo datasets. It can also generate promising disparity maps for trinocular and multi-view stereo datasets.

1 Introduction

An intensity-based stereo vision algorithm takes two or more images as inputs and produces a dense disparity map for one of the images. Previous works in this area are nicely surveyed by Scharstein and Szeliski [9]. (Due to space limits, only the most recent or relevant references are cited in this paper.) As pointed out in their taxonomy, different stereo algorithms generally perform the following four steps: matching cost computation, cost aggregation, disparity computation, and disparity refinement.

1.1 Motivations

There are mainly two motivations in our work. First of all, we would like to compute consistent disparity maps for multiple source images simultaneously and efficiently. Recent image-based rendering techniques use multiple images and their disparity maps to synthesize novel views. If disparity maps are generated for these images separately using traditional stereo vision algorithms, they may not be consistent with each other. That is, if we calculate a disparity map D_s for view *s* and then warp D_s to view *t*, the disparity value at a given pixel *p* may be different from that of the disparity map D_t, which is calculated directly for view *t*. Our experiments show that the inconsistency problems will cause artifacts in generating novel views.

For binocular stereos, the consistency check can be used to detect some of the inconsistency problems. Previous works show that it helps to reduce mismatches [7]. However, the consistency check can only be used for pixels that are visible in both images, and therefore, cannot be applied in occluded areas. To address this problem, we propose two constraints. The strong consistency constraint is a re-formulation of the consistency check under the multi-view stereo scenario. The weak consistency constraint is an extension of the strong consistency constraint, which allows a pixel to be matched more than once. As a result, the weak consistency constraint can be applied to occluded areas as well.

Like many other approaches, we solve the stereo problem under an optimization framework. Different optimization techniques, including dynamic programming (DP) [1], graph cuts [4], and genetic algorithm [2], have been applied under this framework. These techniques try to find a global optimized solution under some given parameters. However, due to the complex nature of the stereo vision problem, it is difficult, if not impossible, to have a universal set of parameters that can produce good disparity maps for different stereo images. In fact, even within a single image, it is highly possible that different regions should use different parameters since the signal-to-noise ratio may vary within the image. As a result, the best solution in terms of the given parameters may not necessarily be a good solution.

Hence, our second motivation is to incorporate a reliability measure into the optimization technique. The proposed reliability-based dynamic programming (RDP) algorithm is based on DP, which is well known for its efficiency. For each scanline, the algorithm generates the best path in terms of a given discontinuity cost, and simultaneously provides the reliabilities of matches at different locations along the path. As a result, we can choose to only accept matches with reliability measures higher than a given threshold.

The RDP algorithm and the consistency constraints can be integrated in an iterative process, which we call the multi-pass dynamic programming (MDP) approach. The MDP approach uses the RDP algorithm to generate reliable matches for different source images, and then validate the matches obtained from different images using either the strong or the weak consistency constraint. Since the confirmed matches need to pass both the reliability test and the consistency test, the final matches generated by this approach are quite accurate.

1.2 Related works

Our RDP algorithm is most related with the scanline optimization (SO) algorithm [9]. The SO algorithm is, to the best knowledge of the authors, the first DP-based algorithm that enforces the smoothness constraint directly. When searching for the best path that starts from pixel p, instead of limiting the searches using the ordering constraint, this algorithm considers all possible paths at pixel p–1. The smoothness constraint is enforced using a predefined smoothness weight, without which the algorithm becomes a local winner-take-all approach. Our major improvement over the SO algorithm is in the use of the reliability measure, which makes it possible to detect potential mismatches. In addition, we also propose a more efficient implementation of the algorithm.

The proposed weak consistency constraint essentially enforces the same constraint as the visibility constraint used in ref[4]. That is, a match blocks matches that are behind it in all views. However, the visibility constraint is formulated in the 3D space using a set of interactions that contains 3D points occluding each other; while the weak consistency constraint is formulated in image space using the warping relations and the disparity values of pixels. As a result, a more concise form can be defined, and an efficient method can be used to check whether or not a pixel is occluded in one of the source images.

Depending on its applications, the proposed MDP approach are also related with the unambiguous stereo matching [5,7] and the multi-view stereo matching [3,4,8] techniques. These papers are cited in relevant sections in the following.

2 Consistency of Disparity Maps

Here we introduce the two consistency constraints. First of all, several concepts in stereo vision are re-formulated and generalized to facilitate the definitions of the constraints. The definitions and comparison with previously used constraints are given later.

2.1 The Stereo Vision Problem

Like many stereo vision algorithms, we assume that all given source images share the same image plane. The disparity of a pixel is defined using the inverse of the distance between the corresponding 3D point and the shared image plane [6]. As a result, pixels from different source images have the same disparity value if they are projections of the same 3D point in the scene. For two pixels with different disparity values, the one with a larger disparity value is the projection of a 3D point that is closer to the shared image plane.

Let $\mathbf{D}$ denote the value domain of disparity. Let $\mathbf{F}_s$ be the set that contains all the pixels in source image s (s=1…n), and let $\mathbf{G}_s$ be the set containing only pixels that have disparity values assigned. By definition, we know $\mathbf{G}_s \subseteq \mathbf{F}_s$. A disparity map defined on image s is called a full solution, if $\mathbf{G}_s = \mathbf{F}_s$, and is called a partial solution otherwise.

The disparity map is defined as a function, $d^s:\mathbf{G}_s \rightarrow \mathbf{D}$, which assigns each pixel in $\mathbf{G}_s$ a disparity value. Assigning a disparity value d to pixel p in image s actually defines a point M_p^d in the 3D scene. It also defines a set $\mathbf{H}_p^d$, which contains the projections of M_p^d in all source images. Here, we call the set $\mathbf{H}_p^d$ a match and q ($q \neq p \wedge q \notin \mathbf{H}_p^d$) the corresponding pixel of p under disparity value d.

For any given image pair s and t, a warping function, $w^{s,t}:\mathbf{F}_s \times \mathbf{D} \rightarrow \mathbf{F}'_t$, is defined, which maps a pixel p in s to its corresponding pixel q in t under a given disparity value d. Here, $\mathbf{F}'_t \supseteq \mathbf{F}_t$ since q may be outside the boundary of t. Please note that we make no assumption on whether or not the epipolar lines coincide with image rows (columns). Instead, we assume that the warping functions defined for different image pairs and different disparity values satisfy the following two properties:

- One-to-one property: $w^{s,t}(w^{t,s}(p,d),d)=p$;
- Transitive property: $w^{s,t}(w^{r,s}(p,d),d)=w^{r,t}(p,d)$.

Consequently, the match $\mathbf{H}_p^d$ is a complete set for warping operation under disparity value d. This means that $\mathbf{H}_p^d = \mathbf{H}_q^d$ provided q is one of the corresponding pixel of p under disparity value d. We also assume that $M_p^d = M_q^d$, even though the 3D coordinates of the two may not exactly be the same. In the rest of this paper, a match $\mathbf{H}$ and its corresponding 3D point M are used interchangeably.

It can be shown that the above properties hold for most of the previously used binocular and multi-view stereo datasets [3,6,8,9]. A detailed discussion of this is outside the scope of this paper.

2.2 The Definitions of Constraints

Definition 1 (Strong Consistency Constraint): *A disparity map defined on $\mathbf{G}_s$ satisfies the strong consistency constraint if the following hold:*

$$p \in \mathbf{G}_s \Rightarrow \forall t, \left(q \in \mathbf{G}_t \wedge d^s(p) = d^t(q)\right)$$

where $q=w^{s,t}(p,d^s(p))$.

Basically this constraint requires that if the disparity value of pixel p in image s is assigned, the disparity value of the corresponding pixel q in any image t must also be assigned. In addition, q must have the same disparity value as that of p. This indicates that the same 3D point M is visible from all these pixels.

Definition 2 (Weak Consistency Constraint): *A disparity map defined on $\mathbf{G}_s$ satisfies the weak consistency constraint if the following hold:*

$$p \in \mathbf{G}_s \Rightarrow \forall t, \left(q \in \mathbf{F}_t \Rightarrow q \in \mathbf{G}_t \wedge d^s(p) \leq d^t(q)\right)$$

This constraint states that if the disparity value for pixel p is assigned, then the disparity value of the corresponding pixel q in any image t should also be assigned provided q is within the image boundary. Furthermore, q should either have the same disparity value as or a larger value than that of *p*.

Intuitively, the weak consistency constraint can be explained as follows: Assigning a disparity value to pixel *p* in image *s* defines a 3D point M_p^d. Since the corresponding pixel *q* is the projection of M_p^d in image *t*, M_p^d should be either visible at *q* or occluded by another 3D point, which is closer to the shared image plane. Either way, the disparity value at *q* should not be smaller than that of *p*. More formally, we prove that the weak consistency constraint holds for all scenes that can be represented by a set of 3D points.

Lemma 1: *The disparity maps for any given set of points in 3D space satisfy the weak consistency constraint, as long as the disparity maps share the same image plane and only one point is visible from any given pixel.*

Proof: Given any set of 3D points, we can generate the disparity maps for all images simultaneously by projecting the points in the order such that the one closer to the shared image plane is projected first. When projecting 3D point *M* to image *s*, one of the following three situations happens:

- $p \notin \mathbf{F}_s$, which means that the projection of *M* is not within the image boundary of image *s*. No change will be made to the disparity map of image *s*.
- $p \in \mathbf{F}_s \wedge p \notin \mathbf{G}_s$, which means that no previously handled point has been projected to pixel *p*, and therefore, *M* is visible at *p*. In this case, *p* is added to $\mathbf{G}_s$, and $d^s(p)$ is assigned according to the depth of *M*. Hence, assume that the projection of *M* in any other image *t* is *q*, we have $q=w^{s,t}(p,d^s(p))$. In addition, if *M* is also visible in image *t*, we have $d^s(p)=d^t(q)$. Otherwise, we have $d^s(p) \leq d^t(q)$ since all previously projected points should not be further away from the image plane than *M*.
- $p \in \mathbf{G}_s$, which means that at least one previously handled point has been projected to pixel *p*. Since only the closest point is visible from *p*, *M* will be occluded. No change will be made to the disparity map of image *s*.

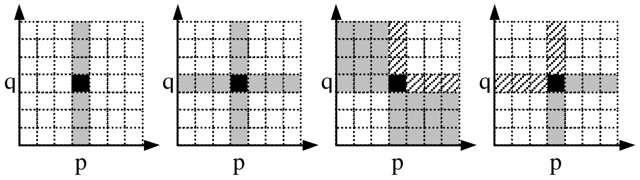
As a result, after all points are projected, the generated disparity maps satisfy the weak consistency constraint. ■

2.3 Comparison with Other Constraints

Both the uniqueness and ordering constraints are widely used in previous works [1,10]. In this section, we highlight the similarities and differences between these constraints and the weak consistency constraint using a binocular matching scenario. For binocular stereo, a match $\mathbf{H}_p^d$ contains two pixels, pixel *p* from source image and *q* from reference image. It can also be expressed as a pair <*p*,*q*>.

Each of the sub-figures in Figure 1 shows an un-skewed disparity space image for a pair of corresponding scanlines. Black denotes a match, gray denotes the match's inhibition zone, and striped denotes the match's occlusion zone.

The uniqueness constraint states that the disparity maps have a unique value per pixel [10]. As shown in Figure 1(a), if we enforce uniqueness in the source image only, a match <*p*,*q*> will forbid pixel *p* to be involved in any other match. Consequently, all other matches in the same column are inhibited. If we enforce uniqueness in both the source and the reference images, as shown in Figure 1(b), a match will forbid any other matches in the same row or the same column. This is the same as the strong consistency constraint under our formulation.



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

Figure 1: Effects of using different constraints.

The ordering or monotonicity constraint states that if an object is to the left of another in one stereo image, it is also to the left in the other image [1]. In practice, we know that this constraint does not hold when thin foreground objects exist in the scene. As shown in Figure 1(c)[1], if we enforce the ordering constraint, a match <*p*,*q*> inhibits matches <*u*,*v*> if $u \geq p$ and $v<q$ or if $v \geq q$ and $u<p$. Matches in the occlusion zone are allowed, but will be penalized by a predefined occlusion cost.

Finally, the effect of the weak consistency constraint is shown in Figure 1(d). The figure shows that under the weak consistency constraint, a match <*p*,*q*> forbids matches <*p*,*v*> ($v<q$) and matches <*u*,*q*> ($u>p$) since these matches will occlude <*p*,*q*>. However, both *p* and *q* can be involved in another match that <*p*,*q*> occludes, as long as the occlusion cost is applied. Since occlusion is modeled explicitly, the weak consistency constraint can be applied to the whole image, while uniqueness and ordering constraints do not hold in either occluded areas or areas that contain thin foreground objects.

3 Reliability-based Dynamic Programming

Here we propose an efficient DP-based algorithm that assigns disparity value *d* to pixel *p* only when the reliability of the corresponding match $\mathbf{H}_p^d$ exceeds a given threshold. First of all, a reliability measure is defined for DP-based approaches in general. A match $\mathbf{H}_p^d$ is denoted as a pair <*p*,*d*> for better readability in this section.

Definition 3 (Reliability): *The reliability R(p,d) of match <p,d> is defined as the cost difference between the best path that does not pass through <p,d> and the best path that passes through <p,d>.*

Obviously, if <*p*,*d*> is on the best path that is found under no constraint, we have $R(p,d) \geq 0$. The higher the value of $R(p,d)$, the more likely it is that the true disparity

[1] Please refer to figure 9 in ref[1], which visualizes the inhibition zone in a skewed disparity space image.

value of pixel p is d.

Comparing to the SO algorithm, our RDP algorithm has several improvements, which are elaborated below. It is noteworthy that since no attempt is made to determine occlusions within a scanline, both algorithms do not require the scanlines to coincide with the epipolar lines. Please also note that we do not explicitly enforce the weak or strong consistency constraint in the RDP algorithm. However, as we show in the next section, the consistency constraints can be applied by using this algorithm as a solver in an iterative process.

3.1 Efficiency Improvement

The first improvement is at the implementation level. In Scharstein and Szeliski's implementation, the complexity of calculating each scanline is $O(L\times D^2)$, where L is the number of pixels per scanline, and D the disparity range. In the result reported by Scharstein and Szeliski [9], depending on the disparity range, the running time of the SO algorithm is 10%~60% slower than that of the conventional DP algorithms.

In our implementation, we assume that the same non-negative discontinuity cost λ is applied whenever neighboring disparity values are different, no matter how large the difference is. With this assumption, at most two possible paths need to be considered when searching the best path that starts from match $<p,d>$. The first one connects to the path at $<p-1,d>$ so that the discontinuity cost will not be incurred. The second path, if differs from the first one, connects directly to the best path for the sub-problem that consists of the first $p-1$ pixels.

Since we only need to search for the best path for the sub-problem once, the complexity of the algorithm is cut down to $O(L\times D)$. Our experimental results show that the speed of the new implementation is comparable with conventional DP algorithms.

3.2 Reliable Matching

According to definition 3, calculating $R(p,d)$ for each match $<p,d>$ on the best path requires running the DP algorithm again under the "do not pass" constraint. For efficiency concern, in the RDP algorithm, the approximate reliability $R'(p,d)$ is used instead, which can be computed using the following algorithm.

First, we assume that array $\mathbf{C}[p,d]$ is used to keep the matching cost of $<p,d>$. When solving the sub-problem for the first p pixels, array $\mathbf{S}[p,d]$ keeps the cost of the best path that starts from $<p,d>$. For every pixel p, $\mathbf{m}[p]$ keeps the value of d that has the smallest value of $\mathbf{S}[p,d]$, and $\mathbf{m}'[p]$ keeps the value of d that gives the second-to-the-smallest value of $\mathbf{S}[p,d]$.

As shown in Figure 2, the same as in the traditional algorithm, the tracing starts from the rightmost pixel z. Assuming $a=\mathbf{m}[z]$ and $b=\mathbf{m}'[z]$, we simultaneously trace the best path (shown in solid lines) from "a" and an alternate path (shown in dashed lines) from "b". It is highly possible that the alternate path may merge with the best path as we trace. We can proof that, using our algorithm, if the two paths merge at pixel p they will merge at $<p,\mathbf{m}[p]>$. This is because at least one of the two paths comes from $<p+1,d>$, where $d\neq\mathbf{m}[p]$. This path always connects to $<p,\mathbf{m}[p]>$ since it is the only choice other than $<p,d>$.

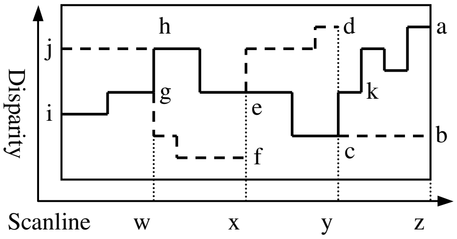


Figure 2: Trace the best and alternate paths.

As shown in the figure, assume that the two paths merge at pixel y and $c=\mathbf{m}[y]$, $d=\mathbf{m}'[y]$. We can then start a new alternate path from d and continue the process. At the end of tracing, we have both the best path and several alternate paths, one on each segment. The approximate reliability R' of matches on the best path within each segment is then calculated using the cost difference between the corresponding alternate path and the best path at the end of the segment. For the above example, the approximate reliabilities of matches on ka is $\mathbf{S}[z,b]-\mathbf{S}[z,a]$, and for those on ec is $\mathbf{S}[y,d]-\mathbf{S}[y,c]$.

We can prove that the inequalities $R'-\lambda\leq R\leq R'+\lambda$ hold. However, the details are not included in this paper due to space limits. Therefore, when the discontinuity cost is small (normally within [0,4] in our experiments), using the approximate reliability measure instead of the reliability measure will not introduce too much bias.

It is noteworthy that when $\lambda=0$, the best path always passes through $\mathbf{m}[p]$ at pixel p, and the alternate paths always start from $\mathbf{m}'[p]$ and merge with the best path after one pixel. Both $R(p,\mathbf{m}[p])$ and $R'(p,\mathbf{m}[p])$ degenerate into measuring $\mathbf{S}[p,\mathbf{m}'[p]]-\mathbf{S}[p,\mathbf{m}[p]]$, which is also equal to $\mathbf{C}[p,\mathbf{m}'[p]]-\mathbf{C}[p,\mathbf{m}[p]]$ when $\lambda=0$.

Since the best path and the alternate paths are traced simultaneously, the approximate reliabilities of all the matches on the best path can be calculated within the same pass as we trace. Therefore, very little computation overhead is added to the conventional algorithm.

3.3 Ground Control Points

Previous research has shown that pre-calculated ground control points (GCPs) can help to eliminate mismatches by reducing the search space [1]. Here, we also use them in a way to improve efficiency. Whenever a pixel is selected as a GCP in previous calculations, the RDP algorithm will skip though it without any redundant computations.

As a result, the complexity of the algorithm drops to $O(L_1+L_2\times D)$, where L_1 is the number of GCPs, L_2 the number of pixels to be calculated. This means that most of the calculations are spent on pixels with ambiguities. As more pixels are added into the GCPs, less computational time is needed.

4 Enforcing Consistency through Iterations

In this section, we propose a new MDP approach, which integrates consistency constraints and the RDP algorithm in an iterative process. Each iteration consists of three major steps: match suggestion, match validation, and disparity space update. All these steps can be done in polynomial time.

In the first step, the RDP algorithm takes a reliability threshold and a discontinuity cost as inputs. It suggests some reliable matches for each of the source images. In the second step, the matches found through different images are validated using either the strong or the weak consistency constraint, depending on the application. Each pixel involved in a confirmed match **H** will be added to solutions for the corresponding source image *s*, provided that **H** is visible in *s*. The pixels already added to the solution will be treated as GCPs by the RDP algorithm in future iterations. Finally, in the last step, the disparity space is updated based on the weak consistency constraint using the new matches found in the current iteration. For each new match, the cost of matches in its inhibition zone is set to infinity. The cost of matches in its occlusion zone is set to the predefined occlusion cost, regardless of whether the original cost is higher or lower than the occlusion cost.

The disparity space update ensures that the matches found by the RDP algorithm in future iterations are weakly consistent with the matches already confirmed, even though the new matches may not be weakly consistent with each other. Since we always add matches to and never remove them from the solutions, the algorithm is bounded to converge.

Depending on the application, different strategies can be used to combine the consistency constraints and the RDP algorithm. We will discuss two different applications in the following subsections.

4.1 Unambiguous Stereo Matching

Due to the difficulties of binocular stereo matching problem, some researchers have started to investigate how to find unambiguous components of stereo matching [5,7]. Following their idea, here we discuss how to apply the MDP approach in this application.

The ambiguities tend to appear in noisy areas, occluded areas, and weakly/periodic textured areas. Similar to previous approaches, we can detect occluded areas using consistency check, in particular, through the strong consistency constraint. Also, the RDP algorithm can be used to identify weakly or periodic textured areas by setting up a reliability threshold. However, a unique feature of our approach is that it is possible to propagate the supports from reliable matches to their neighbors through the smoothness constraint. While previous techniques [7] can only increase the *number* of matches by lowering the threshold, which will introduce more errors, our approach can gradually increase the *density* of matches by increasing the discontinuity cost, *without* lowering the reliability threshold. This helps to find correct matches in weakly or periodic textured areas, while still keeps mismatches caused by isolated noises out of the solution.

A simple algorithm can be formulated using the above idea. Given a binocular stereo dataset, in the first stage, the MDP approach is used to compute reliable matches under the strong consistency constraint, using a high reliability threshold, without any discontinuity cost. After the algorithm converges, which normally takes 3~5 iterations, we increase the discontinuity cost and start another stage of iterations. Depending on how dense or how accurate the matches we want, the user can choose how many stages to use and how fast the discontinuity cost to grow.

4.2 Multi-view Stereo Matching

The visibility problem cannot be fully addressed for binocular stereos. To generate full solutions, the best we can do is to detect occluded areas and fill them using heuristic approaches. However, when multi-view stereo data is available, different algorithms [3,4,8] can be applied to better solve the problem.

In the SEA approach [8], 9 cameras, placed in a 3×3 array on a plane, are used to capture the scene. These captured images form 8 pairs, each of which consists of the center image and one of the peripheral images. The occlusion detection algorithms calculate an overall cost for each disparity value based on the matching costs obtained using different pairs. The local winner-take-all approach is then used to find the best disparity value for the center image.

When handling the same dataset, Kolmogorov and Zabih [4] use 5 images (center, top, bottom, left, right), forming either 4 or 10 pairs. Instead of calculating an overall cost, the matching costs obtained from different pairs are used separately. For a given smoothness term, the graph cuts technique is used to find a local minimum in a strong sense.

The local approach [8] is fast but gives noisy results. Graph cuts based approaches [3,4] produce very nice results but are slow. We try to fill the gap. Similar to ref[4], in our approach, a separate disparity space $\mathbf{S}_{st}$ is initialized using the matching cost computed for each image pair <*s*,*t*>. In each iteration, based on $\mathbf{S}_{st}$, the RDP algorithm is used to calculate some reliable matches for both *s* and *t*. Assuming *N* pairs are used, the above calculation will provide *N* suggested solutions for the center image *s* and one suggested solution for each of the peripheral images *t*. Even though there are redundant calculations for image *s*, they make it possible to detect mismatches that may pass both the reliability and the consistency tests.

For this application, the suggested matches for different images are validated by enforcing the strong consistency constraint between each pair and the weak consistency constraint among all images. That is, to confirm a match **H**, which is found for pixel *p* in image *s* through image pair <*s*,*t*>, **H** must be visible in both *s* and *t*. In addition, **H** cannot occlude any existing or suggested matches in any other image *u*, but it is allowed that **H** is occluded in *u*.

After a match **H** is confirmed, it will be projected to all images. Assuming the projection of **H** is pixel *q* in image *u*. *q* will be added to $\mathbf{G}_u$ if it is not already in there. In addition, the disparity space $\mathbf{S}_{uv}$ for different image pair <*u*,*v*> are also updated using the weak consistency constraint.

After the iteration converges, we will obtain a disparity map for each source image. Disparity values will be assigned to most of the pixels that are visible in both the center image and at least one of the peripheral images. As a result, the disparity map for the center image is normally very dense since very few pixels are not visible in any of the peripheral images. Those for the peripheral images are also denser than the matching results generated using only one pair since reliable matches found from other pairs are used. In our approach, the missing disparity values are filled using median filtering under the weak consistency constraint.

5 Experimental Results

In the first experiment, the RDP algorithm is tested using the “Venus” dataset [9]. To make the results comparable with that reported for the SO algorithm, the same set of parameters is used here.

Figure 3(a) shows the disparity map generated using the RDP algorithm without setting the reliability threshold. The color coding used is shown beside the figure and is the same as that used in ref[7]. Since no GCPs are utilized, the result is similar to that of the SO algorithm. However, the efficiency is improved. On an Athlon 1.5GHz PC with 1GB memory running Windows XP Professional, our implementation can generate the above disparity map in 0.06 sec, while our implementation of SO algorithm takes 0.56 sec. Please note that the time needed for the matching cost computation is not included because it is the same for both algorithms. In addition, the matching costs need to be calculated only once when the RDP algorithm is used in the iterative approach.

Figure 3(b) visualizes the approximate reliabilities calculated for different regions of the disparity map. The brighter the color, the higher the approximate reliability of the corresponding disparity value is. The result shows that the approximate reliabilities are low in textureless areas, such as the top-right corner. Referring to Figure 3(a), we notice that there are many horizontal streaks in this area. On the other hand, in areas where correct and smooth disparity values are produced, such as the top-left corner, the approximate reliabilities are relatively high.

After we set up a reliability threshold, the RDP algorithm can give a partial solution that contains only reliable matches. As shown in Figure 3(c), most of the errors and streaks are removed in the partial solution. However, there are still some errors left in the result, partly because the large smoothness weight used (50 as suggested [9]) enlarges the difference between the reliability and the approximate reliability. Normally a much smaller smoothness weight is used in the RDP algorithm since we do not rely on the smoothness constraint to remove mismatches.

The second experiment compares the unambiguous matches generated for the four datasets used in ref[9] using Sara’s approach [7] and the proposed MDP approach. The results of Sara’s approach are downloaded from http://cmp.felk.cvut.cz/cmp/demos/Stereo/New/Matching/smm-standard.html. Our results are generated by running the algorithm discussed in 4.1 for three stages. The same set of parameters is used for all datasets. The value of the reliability threshold is fixed at 2 throughout the process, and the value of λ steps through 0, 1, and 2 for the three stages. We found that our algorithm is not very sensitive to these parameters. However, questions such as which set of parameters is the best and how to automatically select parameters are worth investigating in the future.

Table 1: Comparison for unambiguous stereo matching

	Tsukuba		Sawtooth		Venus		Map	
	D (%)	*e* (%)	*D* (%)	*e* (%)	*D* (%)	*e* (%)	*D* (%)	*e* (%)
Stage 1	21.7	0.24	26.8	0.11	14.6	0.02	29.2	0.02
Stage 2	36.5	0.33	47.7	0.19	27.5	0.12	43.5	0.03
Stage 3	85.7	1.07	85.0	0.41	67.1	0.51	60.8	0.09
Sara’s	45.7	2.05	61.7	2.15	47.6	1.54	69.9	0.76

The density (*D*) and error rate (*e*) of the matches produced are shown in Table 1. Density is defined as the percentage of matches generated, and error rate the percentage of bad matches (not within correct disparity ±1). Please note that, different from the measure used in ref[9], bad matches within the occluded areas are also counted in the error rate calculation. This generates slightly higher error rates for both approaches. We believe that this is a more fair evaluation for unambiguous stereo matching applications, which are supposed to detect occluded areas.

The results show that with different datasets, our approach gives a lower error rate. In addition, after three stages, in the first three datasets, our approach can produce denser disparity maps as well. The result for the “Map” dataset is not as dense because, as indicated by Scharstein and Szeliski [9], this dataset requires a higher smoothness weight than the other three. If we increase the value of λ to 4 and run for another stage, we can produce a 76.9% density disparity maps with 0.17% error rate.

Due to page limits, here we only show the disparity map generated and the corresponding bad pixels for the first dataset, which has the highest error rate under our approach. Figure 4(a), (b), and (c) are the results after each of the three stages. For comparison, the result of Sara's approach is shown in Figure 4(d).

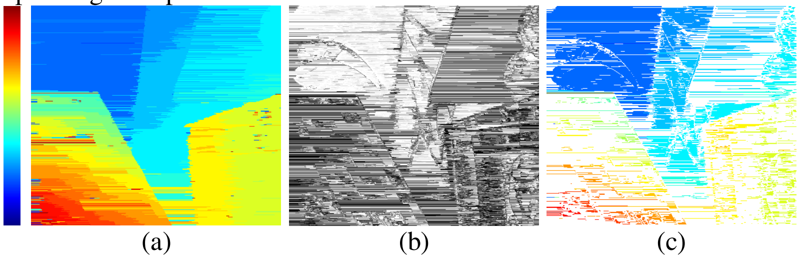


Figure 3: Results of the reliability-based dynamic programming algorithm.

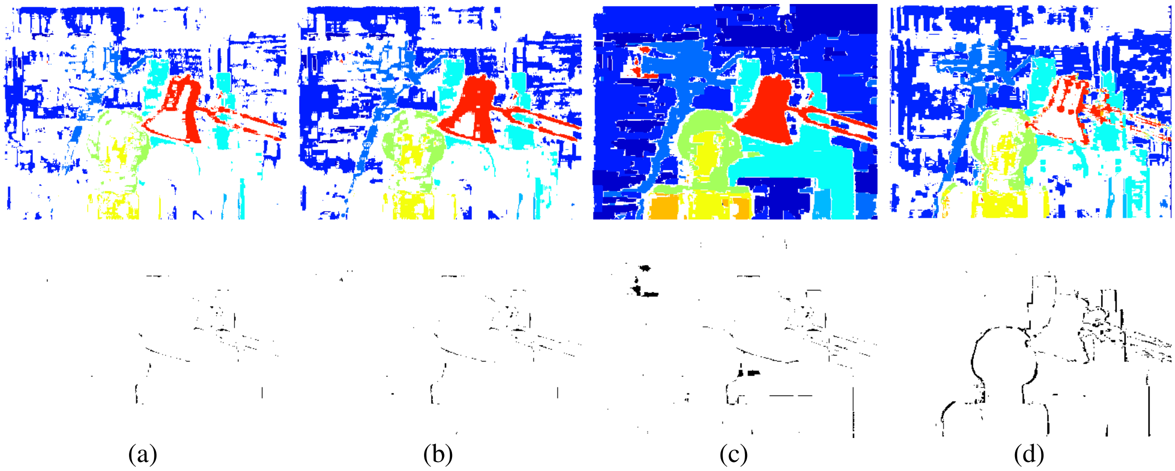


Figure 4: Unambiguous matches found for binocular stereo.

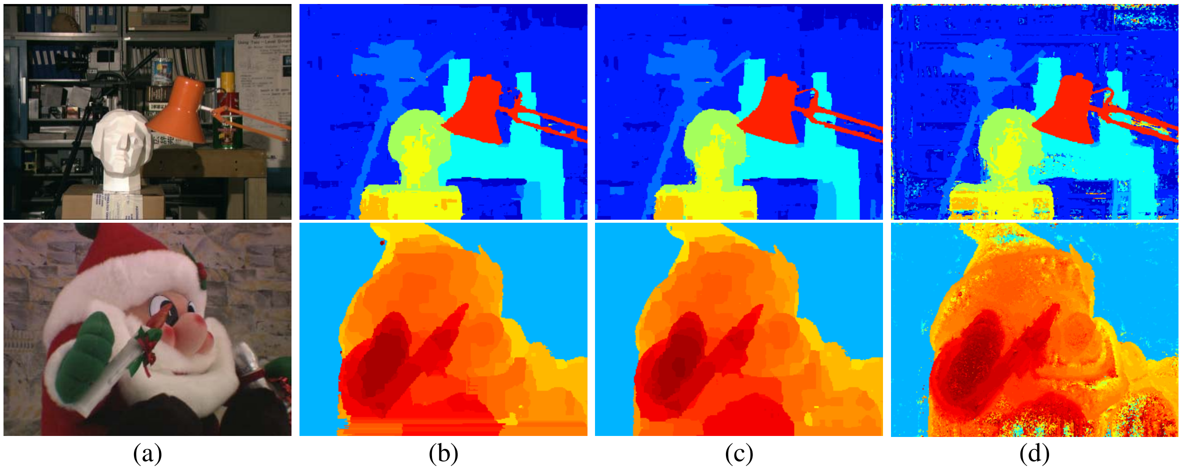


Figure 5: Trinocular and multi-view stereo matching results.

The results show how weakly textured areas are filled up as we increase the discontinuity cost. In addition, as shown in Figure 4(a), because of the high reliability threshold used, the first stage only provides a sparse disparity map. However, some fine details that are not available in Figure 4(d), such as the boom of lamp shade, do show up in the result. The disparity map in Figure 4(b), which has similar density as the one in Figure 4(d), contains much less bad pixels, especially in depth discontinuous regions. This is probably because we use 3×3 matching windows and they use 5×5.

Using the MDP approach to generate reliable matches is also efficient. The experiments show that the three-stage process normally requires a total of 10~15 iterations. As a result, the RDP algorithm is invoked for about 20~30 times since both the left and the right disparity map need to be calculated in each iteration. In practice, the CPU time needed is between 2~5 sec for each of the above datasets. If necessary, the computation time can be reduced at the cost of lowering the density or the accuracy of the matches by limiting the number of iterations per stage.

Finally, the MDP approach is tested for multi-view stereo matching problems. Figure 5(a) shows the center image of the two datasets used in ref[8]. To test the performance of our approach on trinocular datasets, 3 of the 9 source images (center, top, and right) are used first to generate the disparity maps (shown in Figure 5(b)). The approach is then tested using 5 images (same as in ref[4]), and the results are shown in Figure 5(c).

Table 2: Comparison for multi-view stereo matching

	# of images	# of pairs	*e* (%)	*E* (%)	Time (sec)
Ours	3	2	2.03	13.31	2+6
	5	4	1.86	12.62	4+11
SEA	9	8	4.83	23.45	9+0.01
Graph Cuts	5	4	2.75	6.13	369
	5	10	2.30	4.53	837

The comparison on the first dataset, which has the ground truth available, is shown in Table 2. In the table, the mismatch rate (*E*) is defined as the percentage of pixels that do not match exactly to the ground truth. The running time is reported in the form: matching cost computation time + disparity computation time.

The results of SEA approach are based on our implementation using 3×3 matching windows and "new mask" [8] as the detection algorithm (results shown in Figure 5(d)). The results of graph cuts are reported by Kolmogorov and Zabih [4]. In their work, the time is measured on a 450MHz UltraSPARC II.

Table 2 shows that our results have less than half of the bad pixels as the result of SEA approach does, even though less image pairs are used. In addition, within the reported time, our approach generates the disparity maps for all source images, while the SEA approach only produces the center one. Compared with the results of the graph cuts approach, our results have a slightly lower error rate and a much higher mismatch rate. However, it seems the computation cost of our approach is much lower, even though no comparison on the same platform is available.

6 Conclusion

In this paper, we introduce the strong and the weak consistency constraints by re-formulating and extending the consistency check. The weak consistency constraint explicitly models the visibility in the image space, and can be applied to both occluded areas and areas that contain thin foreground objects.

Another contribution of this paper is to introduce a new reliability measure for DP approaches in general based on the cost difference between the best alternate path and the path under use. In the proposed RDP algorithm, an approximate reliability measure is calculated for all matches on the best path with minimal overhead. As a result, instead of relying on the smoothness constraint to remove mismatches, which may also remove some details, we use the approximate reliability measure to detect mismatches. Fine details with enough reliability can then be preserved in the resulting disparity map.

When using the RDP algorithm in an iterative process, we can choose to increase the discontinuity cost gradually. Consequently, matches of the most distinct features are confirmed first. Those in noisy and textureless areas will not be accepted until there are enough supports from neighboring matches. This gives the effects of automatically adjusting the smoothness weight at different regions of the image. Basically, the higher the signal-to-noise ratio of a region, the sooner the matches in the region will be confirmed, resulting in a smaller discontinuity cost to be applied in the region. Our experimental results show that the new MDP approach can generate promising results for both unambiguous stereo matching and multi-view stereo matching applications.

Since the techniques proposed in this paper are based on DP, they are computationally efficient. Improvements can be done on our implementation, especially on matching cost computation part. Parallel implementations for real-time applications appear feasible and will be investigated in future research.

Acknowledgements:

We would like to acknowledge financial supports from NSERC, the Killam Trust, and the Department of Computing Science at the University of Alberta.

References

[1] A. F. Bobick & S. S. Intille. Large occlusion stereo. *IJCV.* 33(3):181-200. 1999.

[2] M. Gong & Y.-H. Yang. Genetic-Based Stereo Algorithm and Disparity Map Evaluation. *IJCV.* 47(1-3):63-77. 2002.

[3] S. B. Kang, R. Szeliski, & J. Chai. Handling occlusions in dense multi-view stereo. *Proc. of CVPR.* 2001.

[4] V. Kolmogorov & R. Zabih. Multi-camera scene reconstruction via graph cuts. *Proc. of ECCV.* pp. 82-96. 2002.

[5] R. Manduchi & C. Tomasi. Distinctiveness maps for image matching. *Proc. of ICIAP.* pp. 26-31. 1999.

[6] M. Okutomi & T. Kanade. A multiple-baseline stereo. *PAMI.* 15(4):353-363. 1993.

[7] R. Sara. Finding the largest unambiguous component of stereo matching. *Proc. of ECCV, v2.* pp. 900-914. 2002.

[8] K. Satoh & Y. Ohta. Occlusion detectable stereo - systematic comparison of detection algorithms. *Proc. of ICPR, v1.* pp. 280-286. 1996.

[9] D. Scharstein & R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV.* 47(1-3):7-42. 2002.

[10] L. C. Zitnick & T. Kanade. A cooperative algorithm for stereo matching and occlusion detection. *PAMI.* 22(7):675-684. 2000.