CENTER FOR
MACHINE PERCEPTION

CZECH TECHNICAL
UNIVERSITY

RESEARCH REPORT

The Class of Stable Matchings for Computational Stereo

Radim Šára

sara@cmp.felk.cvut.cz

№ CTU–CMP–1999–22

November 29, 1999

Available at
ftp://cmp.felk.cvut.cz/pub/cmp/articles/sara/Sara-TR-1999-22.pdf

This research has been supported by the Grant Agency of the Czech Republic under the grants 102/97/0480 and 102/97/0855, and by the Czech Ministry of Education under the grant VS96049.

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Karlovo náměstí 13, 121 35 Prague 2, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

The Class of Stable Matchings for Computational Stereo

Abstract

In this paper the notion of stability of a bipartite matching is introduced and studied. It is a strong but overlooked constraint very useful in computational stereo. The resulting algorithms require no prior continuity model, are parameter-free, very fast, and accurate in terms of mismatches. A whole class of stable matchings is presented. It is shown that some of the popular matching algorithms belong to this class and are subsets of *stable complete matching*. The existence and uniqueness theorem for stable complete matching is given in this paper.

The main result of the paper deals with incorporating the ordering constraint into this framework. The corresponding existence theorem is given and a simple sub-optimal polynomial algorithm for maximum-cost stable monotonic matching is suggested.

It is shown on standard stereo pairs that stable monotonic matching performs very well in the stereo matching task. It is also demonstrated that the algorithm is well suited for efficient disparity map fusion.

Keywords: stereoscopic vision, stereo matching, area-based matching, stable matching, disparity map fusion.

1 Introduction

Recently, the fields of computer vision and computer graphics witnessed a surge of interest in computational stereo spurred by applications, mostly in virtual reality. Most of the recent effort focuses on implementation issues and the question of what the fundamental choices are when posing the matching problem is not studied too often. Modern stereo matching algorithms are based on optimality conditions preferring *continuity and smoothness.* Our aim is to contribute to the following question: *In what sense should stereo matching be optimal?* We bring into focus a non-standard view of area-based stereo matching optimality and explore its advantages and limitations thoroughly and systematically.

Traditionally, the stereo matching problem has been posed as an optimization task that essentially either minimizes the sum of match residuals over all assigned matches or maximizes the sum of match correlations. The former approach is related to Bayesian estimation framework (which usually includes a prior continuity model). There is no such clear framework in the latter case and maximizing the sum of costs is seemingly an arbitrary choice. We call both types the *functional-based matching*. The popular implementations include the dynamic programming methods [5, 6, 12] and the network optimization methods [11, 15]. Other modern implementations include general MAP estimators and related approaches [7, 4, 16, 14, 2, 1, 17, 3].

From the practical point of view, the maximization type of functional-based matching tends to work better when ordering constraint is imposed while the continuity term is omitted. This is due to the fact that the 'trivial' solution is the matching table diagonal (i.e., the zero-disparity solution along the epipolar line) as opposed to the minimization type where the trivial solution is the table corner (i.e., a single match at high disparity). The maximization version thus interpolates naturally, as opposed to the minimization which requires a prior continuity model (with an *a priori* known weighting coefficient). The data term and the continuity term in the minimization functional represent two processes competing against each other, which results in heavy artifacts in areas (1) where data is either missing or ambiguous (the prior model prevails) and (2) where the prior model is violated (on a discontinuity).

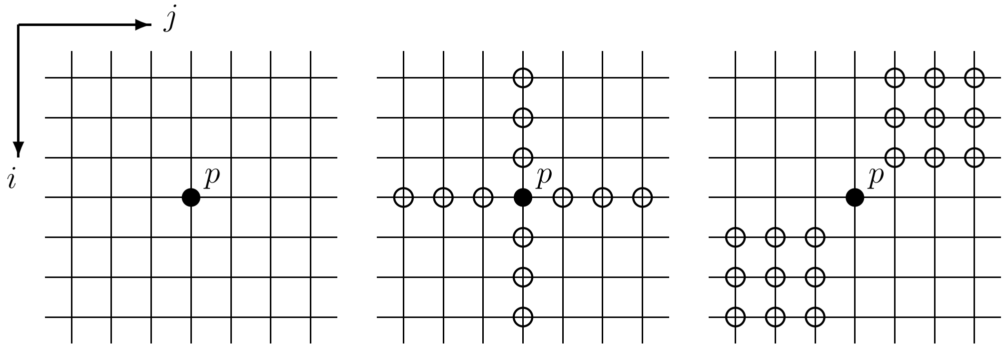
It has been observed by many researchers that, especially in binocular matching, there are often jagged contours in disparity maps, with long streaks stretching across featureless areas in the image, no matter how large the areas are. This happens in both types of functional-based algorithms: In the minimization ones it is due to the additive continuity term, in the maximization ones because of a non-zero probability that a large-cardinality matching of low-correlated pairs has a higher aggregate cost than a small-cardinality matching of high-correlated pairs.[1] In the minimization type, the problem may be alleviated by imposing non-additive matching constraints (like one that the disparity jump be always smaller than a threshold), which in turn increases the computational complexity of the matching task. In the maximization type, the problem may be relieved by a proper choice of the correlation function: A good choice leads to a remarkable improvement in performance; this topic is to be discussed in our forthcoming paper.

In this paper, we answer the question whether we can step back from the functional-based matching and still obtain satisfactory results while doing away with the streak issue. As it turns out we indeed can: At the cost that image regions of weak texture or repetitive pattern remain unmatched, which results in sparse disparity map. But this is the *desired* behavior as overmuch false correspondences in these areas make it virtually impossible to *fuse* disparity maps from more views *in a post-processing step*. Indeed, it is very hard to fuse outliers with correct entities. If disparity map happens to be sparse, it only means we need to take another look from a different viewpoint or to use an independent modality (like Shape-from-X) to acquire the missing information. On the other hand, if we are unable to take another measurement, the sparse map could be used as the automatic 'ground control points' [10, 8] for a dense matching algorithm on condition that *the percentage of false positives among the automatic ground control points remains very low.*

In the past, *non-functional stereo matching* has been attempted too because it leads to very simple algorithms. This is especially appealing for real-time applications. The simple-minded algorithm of this sort works as follows: for each pixel on the left-image epipolar line, select the highest-correlation corresponding pixel on the right-image epipolar line. This can indeed be done fast but the result is not a (bipartite) matching, since several pixels in the left image may be assigned to the same pixel in the right image, which happens quite often in real images. The correct matching algorithm, which we call the X*-dominant matching* (for reasons that become clear later), searches for corresponding pixels in such a way that the correlation maximum is attained from the right image point of view *simultaneously*. The resulting disparity map is usually rather sparse, unless there were strong texture everywhere in the images.

We will show that the X-dominant matching is a subset of a more general *stable complete matching*, which assigns matches to *all* pixels on both epipolar lines (if they are of the same

[1] By cardinality of matching M we mean the number of matched pairs in M.



(a) $p \in P$ (b) $X(p) \subset P$ (c) $F(p) \subset P$

Figure 1: The correlation table P, the X-zone $X(p)$ (empty circles), and the F-zone $F(p)$.

length). Stable complete matching can be considered a basis for non-functional stereo matching algorithms, since it is a superset to all useful non-functional matchings. As will be seen, the monotonic subset of stable matching (which we call the *stable monotonic matching*) provides the densest possible set of automatic ground control points for a subsequent dense stereo algorithm.

It will be shown that stable complete matching can be found in $O(n^2)$ time and $O(n^2)$ space for an epipolar line of length n. Stable complete matching is a special case of *stable marriage problem,* which uses asymmetric rankings; the interested reader is referred to the recent monograph [9] that covers the topic almost exhaustively.

This paper is organized as follows: We first introduce the concept of stability in matching and give the description of stable matching algorithm in Section 2. We also show that all stable matchings are subsets of stable complete matching. In Section 3, we suggest how ordering constraint can be incorporated into this framework. Section 4 then briefly presents results for standard test examples and demonstrates disparity map fusion on a controlled experiment example. Section 5 concludes the paper.

2 The Stable Complete Matching

2.1 Basic Concepts

Let I and J be two index sets representing the set of pixels on the left-image and right-image epipolar line, respectively. Let P be a subset of $I \times J$. The element $p = (i, j) \in P$ will be called a *pair.* Let $c\colon P \to Q$ be a mapping onto an interval of real (or rational) numbers Q assigning each pair p a value called its *correlation*. In the stereo correspondence problem the value of $c(i, j)$ is a measure of how much binocular measurements taken over an area of support at respective positions i and j correspond to the images of the neighborhood of the same spatial point. There is an array of functions that can serve as a correlation measure but we will not distinguish among them here. For the purpose of this paper we convert all correlations c to respective *ranks,* the highest correlation on P having the highest rank r. We will assume there are no ties in the ranking, this helps us simplify the proofs. For given P and r, the tuple (P, r) will be called the *matching problem instance.* It can be arranged in a *correlation table,* which

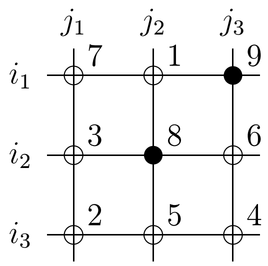


Figure 2: An X-dominant matching represented by the full circles. Numbers are ranks $r(i, j)$.

we will depict as a diagram, in which each crossing (circled or not) represents a pair from P, see Fig. 1(a). Among all pairs in P we will often distinguish *assigned pairs*, which will be denoted by full circles in the diagrams.

Let us define the *X-zone* $X(p)$ and the *F-zone* $F(p)$ of a pair p as follows. The X-zone of $p = (i, j)$ consists of all pairs with the same row index i or column index j, except for the pair p itself, see Fig. 1(b). Formally,

$$X(p) = \big\{q = (i, l) \mid l \in J,\, l \neq j\big\} \cup \big\{r = (k, j) \mid k \in I,\, k \neq i\big\}, \tag{1}$$

The F-zone is formed by two opposite quadrants around (i, j), as is apparent from Fig. 1(c):

$$F(p) = \big\{q = (k, l) \mid (k > i \wedge l < j) \vee (k < i \wedge l > j)\big\}. \tag{2}$$

The union of the X-zone and F-zone of a pair will be called the FX-zone:

$$FX(p) = X(p) \cup F(p). \tag{3}$$

In the rest of the paper it is assumed that we are given a matching problem instance (P, r). If not stated otherwise, pairs p, q, etc. are always drawn from P, this assumption is used silently not to overload the text.

Definition 1 *A bipartite matching $M \subseteq P$ is a one-to-one mapping from I to J.*

We may observe that for each pair p in a bipartite matching M it holds that $X(p) \cap M = \emptyset$ (where $\emptyset$ is an empty set). For the sake of brevity, we omit the word 'bipartite' in the following text.

A perfect matching[2] M is complete in the sense that no pair p can be added to M without violating the X-zone of a pair $q \in M$:

Definition 2 *A matching $M \subseteq P$ is complete iff for each $p \in P$, $p \notin M$ there is a pair $q \in M$ such that $q \in X(p)$.*

In Fig. 4 we can see three examples of a complete matching.

Since what we introduced above as the X-dominant matching has become so popular, we may define it here using the introduced terminology:

[2]In a perfect matching every $i \in I$ is assigned to a unique $j \in J$ and vice versa.

(a) $M_1 = \{(i_2, j_1), (i_1, j_2)\}$ (b) $M_2 = \{(i_2, j_1)\}$

Figure 3: Two examples of an unstable 2×2 matching.

Definition 3 *A matching $M \subseteq P$ is X-dominant iff the rank $r(p)$ of each $p \in M$ is higher than the rank of all pairs q from $X(p)$.*

It is easy to see that X-dominant matching is not necessarily complete in the sense of Def. 2. An example is given in Fig. 2: Note that there is no pair $p \in M$ in $X(i_3, j_1)$.

We will therefore weaken the precedence of high-cost pairs in forming a matching such that it guarantees the existence of a complete matching. Such a matching will be called *stable*:

Definition 4 *Matching M is called stable iff for each pair $p \in M$ and every pair $q \in X(p)$ of higher rank $r(q) > r(p)$ there is a pair $s \in M \cap X(q)$ such that $r(s) > r(q)$.*

Represented by full circles, two examples of an unstable matching are shown in Fig. 3. In Fig. 3(a) the condition is violated because of the pair (i_1, j_1) which is of higher rank than both $r(i_1, j_2)$ and $r(i_2, j_1)$. Pairs such as (i_1, j_1) in this example are called *blocking pairs*. In Fig. 3(b), there is no pair $r \in M$ in $X(i_1, j_1)$ other than (i_2, j_1) and the matching is therefore unstable as well.

The X-dominant matching is stable because for each pair $p \in M$ there is no pair $q \in X(p)$ of higher rank and the stability condition thus holds trivially.

2.2 A Simple Algorithm for Stable Complete Matching

Matching that is both stable and complete will be called *stable complete matching*. Where we want to stress a matching is complete, we put a bar above the letter, e.g., $\bar{M}$. We suggest the following simple algorithm:

Algorithm 1 (Stable Complete Matching)

1. *Form a list L of all pairs $p \in P$ and sort it in descending order according to $c(p)$. Initialize $\bar{M}$ to an empty set.*
2. *If L is empty, terminate. The set $\bar{M}$ is a stable complete matching.*
3. *Let p be the first element in L. Add p to $\bar{M}$ and remove p together with all $q \in X(p)$ from L.*
4. *Go to step 2.*

For an $n \times n$ correlation table, this algorithm requires $O(n^2 \log n)$ time for sorting L and at most $O(n^2)$ time to execute the loop (with the help of two auxiliary arrays that are used to keep track of the deleted pairs). A more efficient implementation uses heaps and works in $O(n^2)$ time, which is the optimum time complexity, since there are $O(n^2)$ pairs in the correlation table and all of them must be examined.

The proof of the algorithm correctness will be stated by Theorem 1, based on the following two lemmas:

Lemma 1 *Let (P, r) be a matching problem instance. The stable complete matching $\bar{M} \subseteq P$ contains the highest-rank pair in P.*

Proof Suppose there is a stable complete matching $\bar{M} \subseteq P$ and the highest-rank pair p in P is not a member of $\bar{M}$. From the definition of completeness, it follows there must be pairs $\{q, s\} \subseteq X(p) \cap \bar{M}$ such that $r(p) > r(q)$ and $r(s) > r(p)$. But the highest rank is $r(p)$, so there is no such s, which is a contradiction. $\square$

The following lemma suggests how a stable complete matching can be decomposed into a series of nested subproblems:

Lemma 2 *Let (P_1, r) be a matching problem instance. Let a matching subproblem (P_2, r) be constructed such that $P_2 = P_1 \setminus (X(p_1) \cup \{p_1\})$, where p_1 is the highest-rank pair in P_1. Then the following holds:*

1. *If $\bar{M}_1$ is a stable complete matching for (P_1, r) then $\bar{M}_2 = \bar{M}_1 \setminus \{p_1\}$ is a stable complete matching for (P_2, r).*

2. *If $\bar{N}_2$ is a stable complete matching for (P_2, r) then $\bar{N}_1 = \bar{N}_2 \cup \{p_1\}$ is a stable complete matching for (P_1, r).*

Proof The first statement holds, since after deleting $X(p_1)$ from P_1, both 'stable' and 'complete' properties of $\bar{M}_2$ are inherited from $\bar{M}_1$. The second statement holds, since rank values in $X(p_1)$ are always smaller than $r(p_1)$, thus the 'stable' property holds, and the completeness is obvious. $\square$

This means that for a given matching problem there is always a unique stable complete matching:

Theorem 1 *For any matching problem instance, Algorithm 1 terminates, and returns a unique stable complete matching.*

Proof We construct a series of strictly nested problems as in Lemma 2. After n steps, $n = \min(|I|, |J|)$, we end up with a trivial matching problem (P_n, r), for which stable complete matching is $\bar{M}_n$. By Lemma 2, the $\bar{M}_1$ is constructed as $\bar{M}_n \cup \{p_n\} \cup \{p_{n-1}\} \cup \cdots \cup \{p_1\}$ and is a stable complete matching. Since the process always succeeds in solving the problem P_n, there is always a stable matching. Because of the nesting strictness, there is at most one such matching. $\square$

Fig. 4 shows a few examples of stable matchings. Note that the sum of costs for the stable complete matching in 4(c) is 14 which is less than the sum of costs for a maximum-cost bipartite

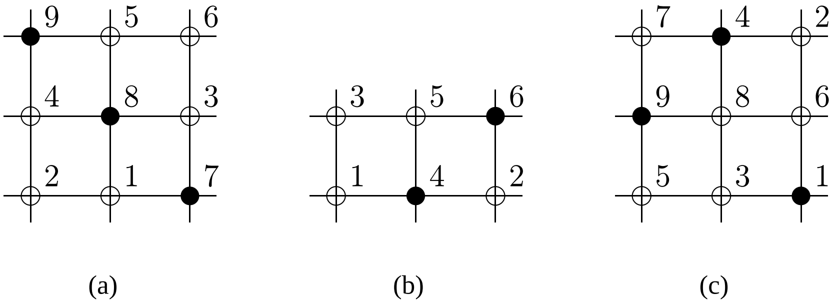


Figure 4: Three examples of stable complete matching (full circles). Numbers are pair ranks.

matching for the same problem (consisting of pairs of rank 1,7,8 or 7,6,3; whose aggregate cost is 16).

2.3 The Set of All Stable Matchings

For the development that will be presented mostly in Section 3, we need to see what is the relation between all stable matchings for a given matching problem instance (P, r). The next two lemmas are therefore crucial.

Lemma 3 *Let (P, r) be a matching problem instance, p be the highest-rank pair in P and let M be a stable matching. Then $X(p) \cap M = \emptyset$.*

Proof Assume that $q \in M \cap X(p) \neq \emptyset$. Then, from the definition of stability, there must exist $s \in X(p)$ such that $r(s) > r(p)$. But this is not possible, since $r(p)$ is the highest rank. $\square$

Lemma 4 *For a given matching problem instance (P, r), any stable matching N on (P, r) is a subset of stable complete matching $\bar{M}$ on (P, r).*

Proof Let $\bar{M}$ and N be the stable complete matching and a stable matching on (P, r), respectively. Let (P_i, r), $P_1 = P$, $i = 1, \ldots, n$ be the nested series of matching problems as in Lemma 2. Let p_i be the highest-rank pair in P_i, and $\bar{M}_i$ be the stable complete matching on (P_i, r). Let

$$N_1 = N,$$
$$N_{i+1} = \begin{cases} N_i \setminus \{p_i\} & \text{if } p_i \in N_i, \\ N_i & \text{otherwise.} \end{cases}$$

Then

(1) By construction of N_i and by Lemma 3, the intersection $N_i \cap (X(p_i) \cup \{p_i\})$ is either empty or $\{p_i\}$. Since $\bar{M}_i \cap (X(p_i) \cup \{p_i\}) = \{p_i\}$, it holds that there is no $q \in (X(p_i) \cup \{p_i\})$ such that $(q \in N_i) \wedge (q \notin \bar{M}_i)$.

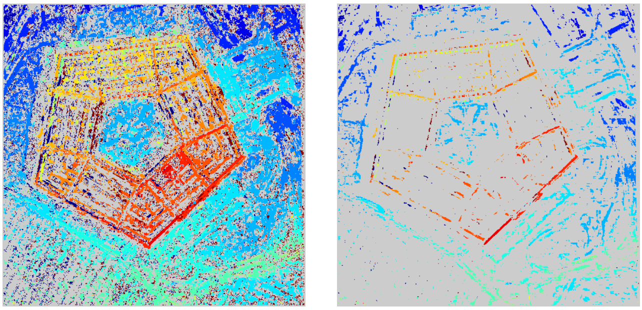


Figure 5: The X-dominant (left) and the FX-dominant matching for the Pentagon stereo pair. See Fig. 6 for input images and the description of color coding.

(2) By observation, N_{i+1} is a stable matching on P_{i+1}. This ensures the validity of (1) for all i.

Since

$$P = \bigcup_i \big(X(p_i) \cup \{p_i\} \big),$$

it holds that there is no $q \in P$ such that $(q \in N) \wedge (q \notin \bar{M})$, which concludes the proof. $\square$

2.4 The X-Dominant and FX-Dominant Matchings

The X-dominant matching has been introduced in Def. 3. We may define:

Definition 5 *A matching $M \subseteq P$ is FX-dominant iff the rank $r(p)$ of each $p \in M$ is higher than the rank of all pairs q from $FX(p)$.*

It is easy to see that both X-dominant and FX-dominant matchings are stable and, by Lemma 4, are therefore subsets of stable complete matching. Figure 5 shows the X-dominant and the FX-dominant matching for the Pentagon standard stereo pair. The FX-dominant matching is very sparse but almost without any false positives. Note the boundaries of the west wing of the building are mismatched (dark colors) even though the match selection constraint is very strong (restrictive).

3 The Stable Monotonic Matching

3.1 Basic Concepts

Definition 6 *Matching M is monotonic iff for each two pairs $p, q \in M$, $p = (i, j)$, $q = (k, l)$ such that $k > i$ it holds that $l > j$.*

For example, the matching in Fig. 4(a) is monotonic, while those in Fig. 4(b) and 4(c) are not.

Definition 7 *Pairs p, q are mutually discordant iff $q \in F(p)$.*

Note that $q \in F(p) \Leftrightarrow p \in F(q)$. We say p, q are *concordant* if they are not discordant. For instance, the pairs of ranks 1 and 9 in Fig. 4(c) are concordant, while the pairs 9 and 4 are discordant. We may observe that a matching M is monotonic iff all couples (p, q) of pairs in M are mutually concordant.

Definition 8 *A monotonic matching M is complete iff for each $p \in P$, $p \notin M$ there is a pair $q \in M$ such that $q \in FX(p)$.*

A matching that is both stable and monotonic will be called *stable monotonic matching* and will be denoted by an asterisk, e.g., M^*. Unfortunately, there is no equivalent of Lemma 1 for stable monotonic matchings, since the largest-cost pair $p \in P$ may be rejected based solely on the existence of a pair $q \in F(p)$. Instead, we have the following:

Theorem 2 *Let (P, r) be a matching problem instance. Then there is a stable monotonic matching M^* for this problem which is complete in the sense of Def. 8 and which is a subset of stable complete matching $\bar{M}$ for the problem.*

Proof We know from Lemma 4 that any stable matching is a subset of stable complete matching. Among all stable matchings on (P, r), there is at least one monotonic matching.[3] The completeness follows from the completeness of stable complete matching: the complete monotonic matching $M^* = \bar{M} \setminus D$ is obtained by removing the smallest possible discordant set D from $\bar{M}$. Such an M^* is complete in the sense of Def. 8. $\square$

Theorem 2 has far reaching consequences for the time complexity of the stable monotonic matching algorithm. Instead of considering all n^2 pairs in P it is sufficient to consider just n pairs, $n = \min(|I|, |J|)$. On the other hand, it also states the stable monotonic matching is obtained by removing pairs from stable complete matching that do not conform to the ordering constraint. The best[4] monotonic matching can thus never be made better than the stable complete matching, even not by including some high-correlation pairs that are not members of the stable complete matching.

3.2 The Maximum-Cost Stable Monotonic Matching

Let $\bar{M}$ be a stable complete matching for a given problem (P, r). We will now focus on removing some pairs from $\bar{M}$ to get a stable monotonic matching M^*, which we know to be a subset of $\bar{M}$.

Consider first three pairs p, q, and s from P, such that $r(p) < r(q) < r(s)$, $s \in F(p)$, and $p, s \in \bar{M}$. According to Def. 4, if s is removed from $\bar{M}$, the matching becomes unstable because of the pair q, $r(q) > r(p)$. To restore stability, the pair p must be removed as well. We say the (deleting of the) pair $s \in \bar{M}$ is *conditioned* by the pair $p \in \bar{M}$, which we call the *conditioning pair*. If a matching is to remain stable after a conditioned pair is removed, the corresponding conditioning pair has to be removed too.

[3] For instance, it is any matching of cardinality 1, which is guaranteed to exist, consider just the highest-rank pair in (P, r).

[4] Best in terms of the matching cardinality or the aggregate costs.

Lemma 5 *Let $\bar{M}$ and M^* be the stable complete matching and a stable monotonic matching for a problem (P, r), respectively. Let $\{p, s\} \subseteq \bar{M}$ be the conditioning and conditioned pairs, respectively, such that $s \in F(p)$. Then $p \notin M^*$.*

Proof The conditioning is an implication: $\mathrm{remove}(s) \Rightarrow \mathrm{remove}(p)$. Two discordant pairs cannot belong to M^* simultaneously, which means $\mathrm{remove}(s) \vee \mathrm{remove}(p)$. By combining the two conditions we get the statement. □

Generally, the ordered path between the conditioning and the conditioned pair can have more than just a single intermediate pair. If the endpoints of such a path happen to be mutually discordant, the corresponding conditioning pair has to be removed from $\bar{M}$.

If it were not for the conditioning relation among the pairs of a stable complete matching $\bar{M}$, finding the largest-cost monotonic subset M^* of $\bar{M}$ would be rather simple: by a dynamic programming algorithm which completes in $O(n^2)$ time.

A polynomial complexity algorithm for the maximum-cost stable monotonic matching is the topic of our ongoing research. At this moment, we only have a *sub-optimal* $O(n^2)$ algorithm that proceeds as follows:

Algorithm 2 (Maximum-Cost Stable Monotonic Matching)

1. *Solve the stable complete matching problem on (P, r) to get $\bar{M}$.*
2. *Remove all conditioning pairs to get $\tilde{M} \subseteq \bar{M}$.*
3. *Solve the maximum-cost monotonic path problem through the pairs of $\tilde{M}$ to get $M^+ \subseteq \tilde{M}$. Use dynamic-programming with free ends; the costs to be summed are the original correlations $c(i, j)$. No jump penalty is necessary.*
4. *Add to M^+ those conditioning pairs from $\bar{M}$ that can be returned back without violating the stability or monotonicity of the matching.*

In a modification of this algorithm, the dynamic problem is solved first and the conditioning pairs are removed from the result. The disparity map resulting from the modified algorithm is more dense but there are slightly more erroneous matches. Handling the conditioning pairs *is* important: we have observed that there are fewer false positives in the disparity map as opposed to the case when the conditioning pairs are just ignored.

4 Results

We are preparing a detailed study of how various stereo matching algorithms fail on thin objects and/or scenes of wide range and many objects. Without any thorough analysis, we only show in this paper what results may be expected from Algorithm 2 on standard test sets for stereo matching.

The results are shown in Fig. 6. Matching window was 5×5 pixels in all cases, which is the minimum reasonable size. Note the percentage of mismatches is very low although the algorithm searches the entire disparity range. Holes (shown in grey) appear in extremely low-contrast areas (like in the lower-right image corner in the Parking Meter set or in the featureless ground area in the Birch Tree set) or in ambiguous areas (like in the west wing of the building in the Pentagon set, which is imaged as a very regularly repeated pattern). In the wide-baseline EPI Tree pair, the scene ordering is not preserved in the images, which is the reason why a section of the rightmost tree trunk is missing in the disparity map (missing is the featureless trunk, not the strongly-textured background). In the Birch Tree disparity map, small gaps in the foreground tree trunk are present because the ordering is violated locally as well: the solution is switching between the foreground and the background depending on the relative texture strength.

In the Parking Meter, EPI Tree, and the Birch Tree sets, the matching was repeated for the right image shifted up and down by one line. The resulting three disparity maps were fused by Algorithm 2. This was done to increase the disparity map density, since the images from these sets are rather poorly rectified. In the other two sets (Shrub and Pentagon), the fusion results had slightly more mismatches on thin horizontal objects than in the standard disparity map, the images are therefore well rectified. We conclude that good epipolar rectification is important for the success of stereo matching. We suggest not to use the Parking Meter, EPI Tree, and the Birch Tree sets for testing stereo matching algorithms relying on rectification, since their results are then incomparable. In one of the other datasets which is often used (the Fruit set), the rectification problem is even more pronounced.

In the tested pairs, the total CPU time spent in Algorithm 2 was about 43% of the total matching time including the computation of the correlation values (we used the normalized cross-correlation coefficient). On a 350 MHz Pentium-II processor, the matching requires about $0.92\mu s$ of CPU time per pixel of disparity search volume. This figure includes the correlation coefficient computation and must be multiplied by the size $m \times n \times d$ of the volume to get the total matching time, where $m \times n$ is the image size and d is the disparity search range. We believe the matching could be speeded up several times by a careful implementation.

For disparity map fusion, one needs artifact-free maps, especially when dealing with a large number of stereo pairs in a polynocular setup. If the artifacts prevailed in such a case, they would make the fusion completely impossible. We believe that, because of its properties, stable monotonic matching is suitable not only for computing the primary disparity maps but also as a fusion algorithm.

The utility of stable monotonic matching for disparity map fusion is illustrated in Fig. 7. The real scene consists of a ball sitting on the top of a cylindrical stalk. Behind the ball there is a vertical plane and below, in front of the stalk, there is a plane inclined at about 30 degrees. All surfaces are texture-free (white paper, plastic). The scene is constructed so that it consists of planar, simply, and doubly curved surfaces and of an occluded region of varying geometry. The background plane has approximately zero disparity. The disparity range in the scene is about 140 pixels (largely because of the inclined plane). Images are 330×330 pixels.

Random texture of adjustable contrast was *projected* onto the scene. The four input images under low-contrast texture illumination are shown in Fig. 7(a). The texture contrast was set so that the matching algorithm failed to find a dense disparity map. Disparity maps for the horizontal image pairs are shown in Fig. 7(b) at left, and for the vertical pairs at right (rotated by 90 degrees). Image set taken under strong texture illumination was used to generate the ground-truth disparity map for the top image pair, see Fig. 7(e). The ground-truth map was not cleaned manually. Fig. 7(d) shows the result of fusion of the maps from Fig. 7(b) in the disparity space of the top camera pair. The depression in the sphere's center is due to a strong specularity. For comparison, the top pair disparity map is enlarged in Fig. 7(c). After fusion, the percentage of holes decreased by 45% (holes were counted relative to the ground-truth map).

The Shrub set: `shrub-3` and `shrub-21`

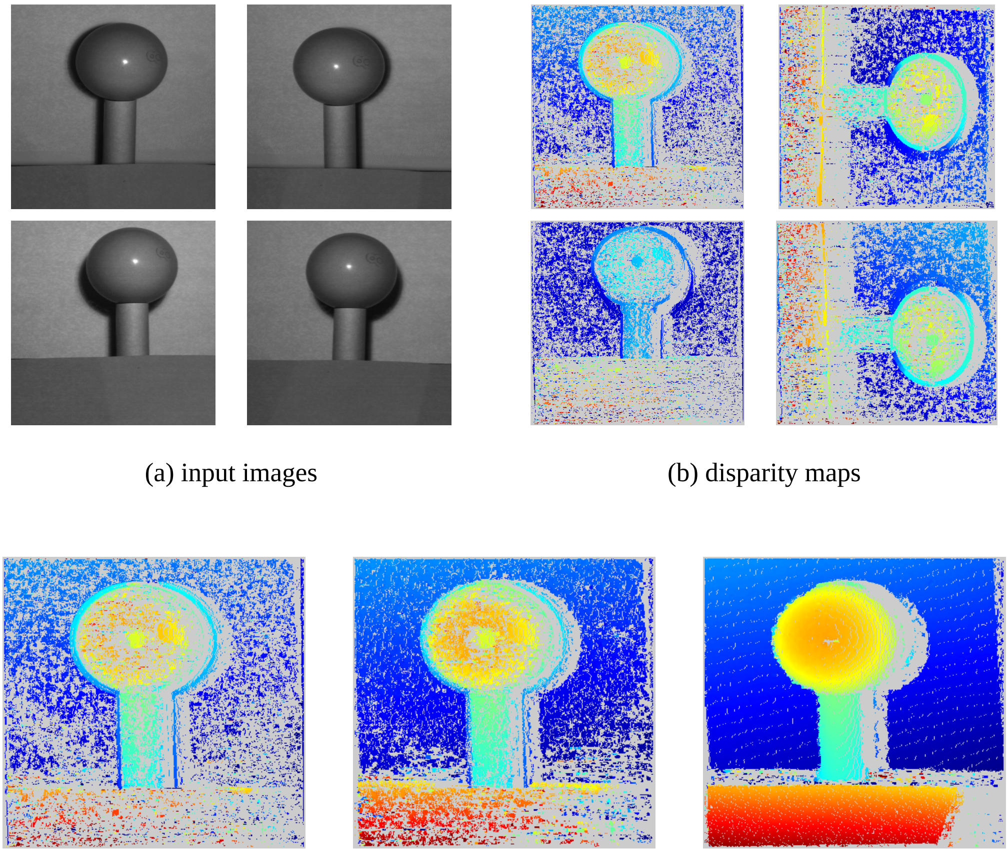
The Parking Meter set: `pm-2` and `pm-14`

The Pentagon set: `left` and `right`

The EPI Tree set: `epi-32` and `epi-16`

The Birch Tree set: `birch-1` and `birch-2`

Figure 6: Results of Algorithm 2 on some standard stereo sets. Disparity is coded by color (small disparity is blue, large disparity red), holes are shown in grey.



(a) input images

(b) disparity maps

(c) top pair disparity map

(d) fused map

(e) ground-truth map

Figure 7: Disparity map fusion from two binocular views of the same object.

5 Conclusions

In this paper, we studied the class of stable matchings including the X-dominant matching, the FX-dominant matching, and the maximum stable monotonic matching. It was shown that all of them are subsets of stable complete matching. It is interesting to note that the idea of stability is present intrinsically in the famous PMF algorithm [13].

We saw that stable complete matching can be found in $O(n^2)$ time where n is the image width in pixels. Stable monotonic matching (SMM) imposes ordering constraint on the solution. We have presented a simple sub-optimal $O(n^2)$ algorithm for SMM (Algorithm 2), which works much like dynamic programming-based matching but is parameter-free (except for matching window size) and differs in that it looks for maximum-cost path through only a subset of pairs of stable complete matching. The CPU time spent in Algorithm 2 is about 43% of the total matching time if normalized cross correlation is used. We are currently working on a polynomial algorithm that finds the exact solution to the maximum-cost stable monotonic matching problem.

As has been argued, the SMM is suitable when low percentage of false positives is required in disparity map. As opposed to functional-based matching algorithms, the SMM does not suffer from the 'streak' effect. The price is that the map is usually sparser, especially in the areas of weak and/or ambiguous texture. But the result can be made denser by taking another look and fusing the binocular maps, at which SMM is very efficient, especially for large polynocular stereo sets. The sparsity of maps to be fused does not matter in SMM. In the two binocular view example described in Section 4 we have seen that the sparsity decreased by 45% after fusion. We have also seen that SMM is suitable for deep-range scenes of thin objects (cf. the EPI and Birch Tree examples shown in Fig. 6).

As long as ordering holds, the SMM rarely misses thin foreground objects that appear against a distant background. This has a lot to do with the principle on which the matching is based. Unlike stable matchings, algorithms that require a prior continuity model or accumulate matching costs along the solution fail quite often in such cases. We are preparing a detailed comparative study analyzing these issues.

Acknowledgements

I am grateful to Ondřej Drbohlav from CMP, who contributed significantly to the clarity of the paper.

Results in this paper are presented on test images from the VASC Image Database at Carnegie Mellon University, the Shrub and ParkingMeter meter data were provided by CMU, the EPI and Birch Tree data were provided by SRI.

References

[1] P. N. Belhumeur. A Bayesian approach to binocular stereopsis. *Int. J. Computer Vision*, 19(3):237–260, 1996.

[2] Y. Boykov, O. Veksler, and R. Zabih. Disparity component matching for visual correspondence. In *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 470–475, 1997.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *Proc. Int. Conf. Computer Vision*, vol. 1, pp. 377–384, 1999.

[4] C. Chang and S. Chatterjee. A deterministic approach for stereo disparity calculation. In *Proc. European Conf. Computer Vision*, pp. 420–424, 1992.

[5] I. J. Cox. A maximum likelihood *n*-camera stereo algorithm. In *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 733–739, 1994.

[6] D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. *Int. J. Computer Vision*, 14:211–226, 1995.

[7] M. A. Gennert. Brightness-based stereo matching. In *Proc. Int. Conf. Computer Vision*, pp. 139–143, 1988.

[8] G. Gimel'farb. Pros and cons of using ground control points to validate stereo and multiview terrain reconstruction. Presented at *Evaluation and Validation of Computer Vision Algorithms*, Schloss Dagstuhl, Wadern, Germany, March 1998.

[9] D. Gusfield and R. W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. The MIT Press, Cambridge, London, 1989.

[10] S. S. Intille and A. F. Bobick. Disparity-space images and large occlusion stereo. In *Proc. European Conf. Computer Vision*, pp. 179–186, 1994.

[11] H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *Proc. European Conf. Computer Vision*, vol. 1, pp. 232–248, 1998.

[12] L. H. Matthies. Stereo vision for planetary rovers: Stochastic modeling to near real time implementation. *Int. J. Computer Vision*, 8(1):71–91, 1992.

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

[14] L. Robert and R. Deriche. Dense depth map reconstruction: A minimization and regularization approach which preserves discontinuities. In *Proc. Int. Conf. Image Processing*, pp. 123–127, 1992.

[15] S. Roy and I. J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *Proc. Int. Conf. Computer Vision*, pp. 492–499, 1998.

[16] D. Scharstein and R. Szeliski. Stereo matching with non-linear diffusion. In *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 343–350, 1996.

[17] R. Szeliski. Stereo algorithms and representations for image-based rendering. In *Proc. British Machine Vision Conf.*, vol. 2, pp. 314–328, 1999.