Presented at 52th Int Symp ELMAR 2010

How To Teach Stereoscopic Matching?

(Invited Paper)

Radim Šára
Center for Machine Perception
Department of Cybernetics
Czech Technical University in Prague, Czech Republic
Email: sara@cmp.felk.cvut.cz

***Abstract*—This paper describes a simple but non-trivial semi-dense stereoscopic matching algorithm that could be taught in Computer Vision courses. The description is meant to be instructive and accessible to the student. The level of detail is sufficient for a student to understand all aspects of the algorithm design and to make his/her own modifications. The paper includes the core parts of the algorithm in C code. We make the point of explaining the algorithm so that all steps are derived from first principles in a clear and lucid way. A simple method is described that helps encourage the student to benchmark his/her own improvements of the algorithm.**

I. Introduction

Teaching stereovision in computer vision classes is a difficult task. There are very many algorithms described in the literature that address various aspects of the problem. A good up-to-date comprehensive review is missing. It is thus very difficult for a student to acquire a balanced overview of the area in a short time. For a non-expert researcher who wants to design and experiment with his/her own algorithm, reading the literature is often a frustrating exercise.

This paper tries to introduce a well selected, simple, and reasonably efficient algorithm. My selection is based on a long-term experience with various matching algorithms and on an experience with teaching a 3D computer vision class. I believe that by studying this algorithm the student is gradually educated about the way researchers think about stereoscopic matching. I tried to address all elementary components of the algorithm design, from occlusion modeling to matching task formulation and benchmarking.

An algorithm suitable for teaching stereo should meet a number of requirements:

Simplicity: The algorithm must be non-trivial but easy to implement.

Accessibility: The algorithm should be described in a complete and accessible form.

Performance: Performance must be reasonably good for the basic implementation to be useful in practice.

Education: The student should exercise formal problem formulation. Every part of the algorithm design must be well justified and derivable from first principles.

Development potential: The algorithm must possess potential for encouraging the student to do further development.

Learnability: There should be a simple recommended method for choosing the algorithm parameters and/or selecting their most suitable values for a given application.

Besides the basic knowledge in computer science and algorithms, basics of probability theory and statistics, and introductory-level computer vision or image processing, it is assumed that the reader is familiar with the concept of epipolar geometry and plane homography. The limited extent of this paper required shortening some explanations.

The purpose of this paper is not to give a balanced state-of-the-art overview. The presentation, however, does give references to relevant literature during the exposition of the problem.

II. The Basic Structure of Stereoscopic Matching

Stereoscopic matching is a method for computing disparity maps from a set of images. In this paper we will only consider pairs of cameras that are not co-located. We will often call them the left and the right camera, respectively. A disparity map codes the shift in location of the corresponding local image features induced by camera displacement. The direction of the shift is given by *epipolar geometry* of the two cameras [1] and the magnitude of the shift, called *disparity,* is approximately inversely proportional to the camera-object distance. Some disparity map examples are shown in Fig. 10. In these maps, disparity is coded by color, close objects are reddish, distant objects bluish, and pixels without disparity are black. Such color-coding makes various kinds of errors clearly visible.

This section formalizes object occlusion so that we could derive useful necessary constraints on disparity map correctness. We will first work with spatial points. The 3D space in front of the two cameras is spanned by two pencils of optical rays, one system per camera. The spatial points can be thought of as the intersections of the optical rays. They will be called *possible correspondences*. The task of matching is to accept some of the possible correspondences and reject the others. Fig. 1 shows a part of a surface in the scene (black line segment) and a point p on the surface. Suppose p is visible in both cameras (by optical ray r_1 in the left camera and by optical ray t_1 in the right camera, both cameras are located above the 'scene'). Then every point on ray r_1 that lies in front of p must be transparent and each point of r_1 behind p must be occluded. A similar observation holds for ray t_1. This shows that the decision on point acceptance and rejection are not independent: If the correspondence given by point p is accepted, then a set of correspondences $X(p)$

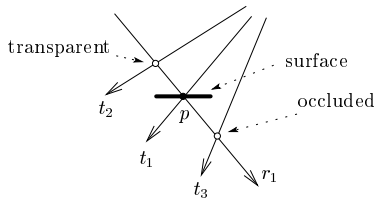


Fig. 1. Occlusion induces a symmetric relation among all possible correspondences: Point p given by the intersection of optical rays r_1 and t_1 occludes the point given by the intersection (r_1, t_3) and implies that point (r_1, t_2) is transparent. Hence (r_1, t_3) and (r_1, t_2) are rejected by accepting (r_1, t_1).

must be rejected. The set $X(p)$ is formed by two optical rays passing through p, from which the p is excluded, i.e. $p \notin X(p)$. Notice the symmetry: If $q \in X(p)$ then $p \in X(q)$. Hence, occlusion induces a symmetric relation among all possible correspondences (3D points). Relation X constrains the admissible selection of correspondences (matching) among all possible correspondences.

We can represent the two systems of optical rays by a grid such as in Fig. 2: Horizontal lines represent optical rays of the left camera and vertical lines represent optical rays of the right camera. A possible correspondence $p = (r_i, t_j)$ is represented by a node of the grid. We will call the grid *matching table* and its nodes *pairs*. We then do not talk about optical rays but about pairs $p = (i, j)$ of the matching table, where i is the pixel index in the left image and j is the pixel index in the right image. The set $X(p)$ is now the union of column j and row i of the table, with pair (i, j) excluded. Relation X then dictates that no two accepted pairs must lie in the same row or the same column of the table. This is the *uniqueness constraint* [2], [3]. Hence, an admissible matching is only allowed to contain pair subsets M such that for each $p \in M$ it holds $X(p) \cap M = \emptyset$.

If we want to encode additional information about the scene, we will need a formal definition of disparity. Let $q = (i, j)$ be a matching table pair. Disparity of q is then the difference $d = i - j$. Pairs on the main diagonal of the table in Fig. 2 have zero disparity, pairs under the main diagonal have positive disparity and pairs above the main diagonal have negative disparity.

The geometric meaning of disparity can be derived from the

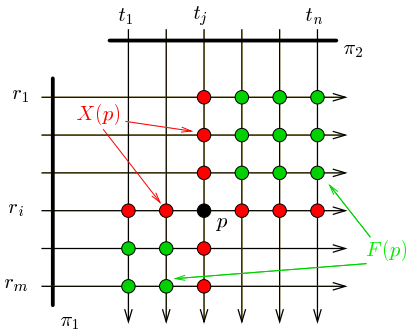


Fig. 2. Matching table representation of the correspondence problem. The π_1 and π_2 represent image lines in the left and right image, respectively, and the r_i, t_j are optical rays. The sets $X(p)$ and $F(p)$ are explained in the paper.

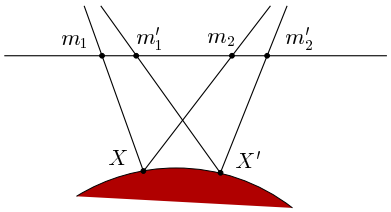


Fig. 3. Ordering constraint.

camera pair geometry [1]. Here we will use the fact that the camera-object distance z is a bijective function of disparity d (in a camera setup with parallel optical rays it is $z \sim \frac{1}{d}$).

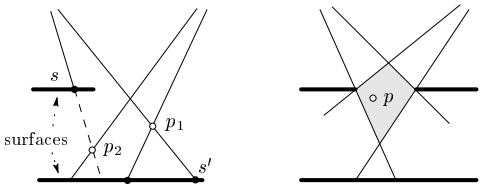
The fact that disparity is a basis for stereopsis has been demonstrated by the stereoscope of Sir Charles Wheatstone [4] which laid the foundations of the modern study of stereopsis. We can use the fact to 'translate' our knowledge about the shape of the geometric object on a disparity map constraint. A more informed model we are seeking needs to express our natural expectation that matter is cohesive and therefore the depths of neighboring image pixels should not differ much. This means their disparity should not differ much. This expectation is often called *continuity constraint* [2].

The first step to representing the continuity constraint is the following observation. Let there be a continuous surface that is visible to both cameras, as shown in Fig. 3. We select two points X a X' on the surface. They project to image points m_1, m_1' (in the left camera) and m_2, m_2' (in the right camera). If m_1' is on the right of m_1 then m_2' must on the right of m_2, as shown in Fig. 3. This is a necessary condition for continuity expressed in a form which can be represented in the matching table in a similar way as the uniqueness constraint. It is called the *ordering constraint* [5]: Let p be a matching table pair as in Fig. 2 and let $F(p)$ be a set of those pairs that violate the ordering constraint with respect to p. Then these pairs form two opposite quadrants around p, as shown by the green nodes in Fig. 2, where we define $p \notin F(p)$. Again, the relation is symmetric: If $q \in F(p)$ then $p \in F(q)$. As in the uniqueness constraint, the ordering constraint can be thought of as a dependence between pair acceptance and rejection. In fact, it is an occlusion model for piecewise continuous surfaces, in the same sense as the uniqueness constraint is an occlusion model for a set of points. In practice we work with the union of $FX(p) = X(p) \cup F(p)$, which is called the *forbidden zone* [6]. This notion is strongly related to the *disparity gradient limit* [7], that has been used in one of the first successful stereoscopic matching algorithms PMF [8].

Interestingly, the ordering constraint is satisfied even in non-continuous scenes. Consider Fig. 3: The point X' need not lie on the same surface as X for the constraint to hold. The constraint is violated in cases when there are very thin objects close to the camera or when the distance between the cameras (stereoscopic baseline) is large. This does not happen very often, even the scene in Fig. 4 satisfies the ordering constraint almost everywhere.

Clearly, the ordering constraint is informative: There are

Fig. 4. A simple scene with both types of occlusions.



(a) half-occlusion (b) mutual occlusion

Fig. 5. In half-occlusion (a) each off-surface spatial point (e.g. p_1 or p_2) is rejected by some surface point which is visible in both cameras (e.g. s or s'; full points). In mutual occlusion (b) there is a region (gray) that is not rejected by binocularly visible points; this region can neither be accepted nor rejected by any matching algorithm that does not include image interpretation.

considerably fewer matchings that satisfy ordering than all matchings.[1] If the constraint holds in our stereo setup, it should be included in our model, since selection from a smaller set of admissible matchings results in a smaller chance of getting a 'wrong' result when image similarity is essentially a random variable and there are several matchings of very similar cost (more on matching cost will come next).

What we have just said suggests that texture is important for successful stereopsis: Image similarity is the basis for any stereoscopic matching algorithm. The similarity must distinguish between good and bad correspondences well. In an ideal case, each surface point is uniquely identified by the appearance of its local surface neighborhood. Texture features are the most often used features for surface point identification. A more spatially varying texture typically results in a more discriminable feature. If the texture has a small contrast with respect to image quantization we say it is a weak texture. A local (let alone global) presence of weak textures is a typical problem of stereovision.

Before we formulate the stereoscopic matching problem let us return back to the notion of accepting and rejecting possible correspondences. We will test this on two examples in Fig. 5(a) and 5(b). Let the set $X(p)$ be extended for p, this is denoted as $X^*(p) = X(p) \cup \{p\}$. The scene in Fig. 5(a) consists of a plane which is partially occluded by another planar object. Consider the union of all those points on both planes that are visible in both cameras, for instance s or s' belong there. If we take the union of the sets $X^*(\cdot)$ for all these binocularly visible points, the whole space will be filled in. This means that each spatial point is either a member of the matching or it is rejected by a matched point. In other words, there is either a corresponding point in the left camera for every right image point t or the intersection of optical ray t with an object in the scene is occluded by some other point visible in both cameras (we are ignoring image borders). The same holds for left image points. This configuration is called *half-occlusion.* In this case every image point in either camera can be matched or explained as half-occluded.

This is not the case in the scene in Fig. 5(b). The scene is formed by a plane in front of which there is another plane with a slit in it. The slit is so narrow and the planes are so apart that no point of the rear plane is visible in both cameras. The union of the sets $X^*(\cdot)$ for all points on the front plane fill the space up to the region that is shown in grey in Fig. 5(b). This region projects to both images. The cameras see the image of the background there. In case when the background has a weak texture it is not easy to decide if the region can be explained as a slit or as a featureless object anywhere within the gray region.

We will call this configuration *mutual occlusion.* The possibility of mutual occlusion means that it is not possible to reach a unique decision on all image points as in the case of half-occlusion. A good model must capture this possibility to be able to explain a segment of the image as unmatchable. An example of a scene with mutual occlusion is shown in Fig. 4. The mutual occlusion region is between the first and the second tree from the left. The sky represents the weakly textured background. Of course, if we knew which image segment corresponded to the sky and which to the foreground trees then we could have resolved the non-uniqueness. But this means that a low-level process like stereopsis requires image interpretation!

The next section formalizes the matching problem so that our knowledge about stereoscopic image formation we have just developed is employed properly.

III. Stereoscopic Matching Problem Formulation

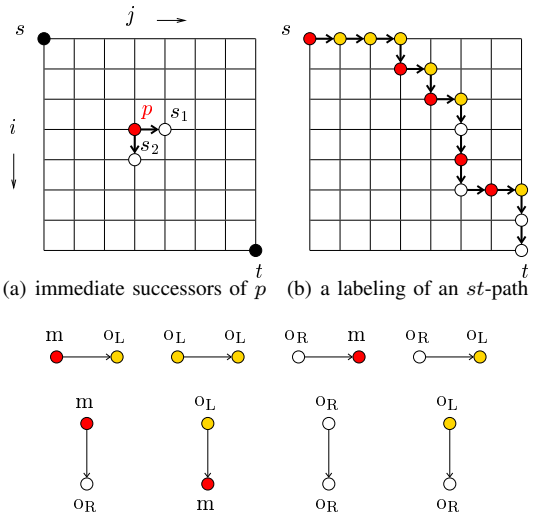
This section describes a simple matching algorithm that satisfies the uniqueness and ordering constraints. The matching table can be thought of as a graph $G = (U, E)$. Its nodes U are pairs of the table. Edges E of this graph connect each pair with its four immediate neighbors in the table.

The graph G will contain two special nodes: the source node $s = (1, 1)$ and the sink node $t = (m, n)$. Edges of G are oriented so that each node $p \in U$, $p = (i, j)$ has two successors: one to the right, i.e. to $s_1 = (i, j + 1)$ and one to the down, i.e. to $s_2 = (i + 1, j)$, see Fig. 6(a).[2] An oriented path from node s to t will be called an *st-path.* Note the important property that the lengths of all st-paths are equal. An example of st-path is in Fig. 6(b), where the arrows follow the graph edge orientation (the colors are immaterial at this point).

Next, the uniqueness constraint and occlusion must be represented. Each node of an st-path receives one of three

[1] The number of all maximal matchings in an $n \times n$ matching table is $n!$, while the number of all maximal ordered matchings is smaller than $2\binom{2(n-1)}{n-1}$.

[2] The nodes at the bottom and the right sides of the table have a single successor and the sink t has no successor.



(a) immediate successors of p (b) a labeling of an st-path

(c) allowed pairs of neighboring labels for horizontal/vertical edges, respectively

Fig. 6. The construction of st-path in oriented graph G representing the matching table (a,b). The construction of admissible labeling (c).

auxiliary labels: m – the node represents a match (accepted correspondence), $\mathrm{o_L}$ – the node is half-occluded by an m-node in the left image, and $\mathrm{o_R}$ – the node is half-occluded by an m-node in the right image. Labels in neighboring nodes cannot be arbitrary: The allowed label pairs differ for the horizontal and the vertical edges, as listed in Fig. 6(c), where the labels are color-coded for clarity. An example of an st-path with an admissible labeling is shown in Fig. 6(b).

It is not difficult to verify the following statements for each given admissibly labeled st-path C:

1) Labeling of C is typically not unique.
2) No two m-nodes on C are immediate neighbors.
3) Along C, there is at most a single transition $\mathrm{m} \to \mathrm{o_L}$ or $\mathrm{o_R} \to \mathrm{m}$ in each matching table row and at most a single transition $\mathrm{m} \to \mathrm{o_R}$ or $\mathrm{o_L} \to \mathrm{m}$ in each matching table column.
4) The m-nodes of C satisfy the uniqueness and ordering constraints, hence they form a matching.
5) For each matching M there is an admissibly labeled st-path such that M is the set of its m-nodes.

The labeling rules that involve labels $\mathrm{o_L}$ or $\mathrm{o_R}$ model both occlusion types:

1) Every transition $\mathrm{o_L} \to \mathrm{o_L}$ or $\mathrm{o_R} \to \mathrm{o_R}$ along an st-path corresponds to 'skipping' one pixel in either image, which represents half-occlusion of this pixel.
2) Transitions $\mathrm{o_L} \to \mathrm{o_R}$ and $\mathrm{o_R} \to \mathrm{o_L}$ represent switching from left-occlusion to right-occlusion and vice-versa, which helps represent mutual occlusion.

The stereoscopic matching problem is now reduced to finding an optimal labeled st-path C. In this paper we select solution of maximal posterior probability which leads to finding

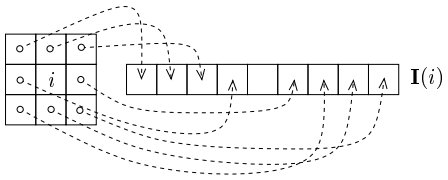


Fig. 7. A 3×3 neighborhood of image pixel i and the construction of $\mathbf{I}_L(i)$.

$$(C^*, L^*) = \operatorname*{arg\,min}_{C \in \mathcal{C}, L \in \mathcal{L}(C)} \Big(V(D \mid C, L) + W(C, L) \Big), \quad (1)$$

in which the data term $V(D \mid C, L)$ models image similarity in m-nodes given an st-path C together with its labeling L, and the structural prior term $W(C, L)$ penalizes deviations from the prior disparity model. In our case we choose the most popular prior model: Equality of disparity in neighboring m-nodes along an st-path. The $\mathcal{C}$ is the set of all st-paths in graph G and $\mathcal{L}(C)$ is the set of all admissible labelings of C.

We choose the data term as

$$V(D \mid C, L) = \sum_{p \in C} V_p\big(D \mid \lambda(p)\big), \quad (2)$$

where the individual terms in the sum are defined in nodes p given the label $\lambda(p)$. We now select a suitable V_p. Let $p = (i, j)$ be the node of graph G. Let $\mathbf{I}_L(i)$ be the vector of pixel neighborhood values around pixel i in the left image. The order of the elements in $\mathbf{I}_L(i)$ is given by the lexicographic order of listing the $k \times k$ elements in the neighborhood of i, as illustrated in Fig. 7 (typically, $k = 5$). The vector $\mathbf{I}_R(j)$ in the right image is defined analogically. We assume the mean over all elements of $\mathbf{I}_L(i)$ is zero (if not, it can be subtracted from all the elements). Assuming $p = (i, j)$, a suitable choice for V is then

$$V_p(D \mid \lambda(p) \neq \mathrm{m}) = V_\mathrm{o}, \quad (3)$$

$$V_p(D \mid \lambda(p) = \mathrm{m}) = \frac{1}{k^2\, \sigma_I^2(i,j)} \|\mathbf{I}_L(i) - \mathbf{I}_R(j)\|^2, \quad (4)$$

where V_o is a constant occlusion penalty, and $\sigma_I(i, j)$ is a constant for each pair. The $\sigma_I(i, j)$ represents the scale in which we measure the difference $\mathbf{I}_L(i) - \mathbf{I}_R(j)$. These scales are unknown, so we replace them by suitable estimates. Since the optical image pixel value can be thought of as Poisson random variable [9], whose standard deviation mean is equal to its mean, then, assuming ergodicity of the process [10], we can estimate

$$\sigma_I^2(i,j) = \alpha_0 \left(s^2\big(\mathbf{I}_L(i)\big) + s^2\big(\mathbf{I}_R(j)\big) \right), \quad (5)$$

where $s^2(\mathbf{x})$ is the sample variance computed from the elements of vector $\mathbf{x}$ and α_0 is a constant. By substituting (5) to (4), we get

$$V_p(D \mid \lambda(p) = \mathrm{m}) = \frac{1}{\alpha_0} \left(1 - \frac{2\, s\big(\mathbf{I}_L(i), \mathbf{I}_R(j)\big)}{s^2\big(\mathbf{I}_L(i)\big) + s^2\big(\mathbf{I}_R(j)\big)} \right) \stackrel{\mathrm{def}}{=} \alpha_0^{-1} \Big(1 - \mathrm{MNCC}\big(\mathbf{I}_L(i), \mathbf{I}_R(j)\big) \Big), \quad (6)$$

TABLE I
PAIRWISE COMPATIBILITY FUNCTION $W(\lambda_2 \mid \lambda_1)$.

$W(\lambda_2 \mid \lambda_1)$		λ_2		
		m	o_L	o_R
λ_1	m	∞	0	0
	o_L	$\ln \frac{1+\alpha_1+\alpha_2}{2\alpha_2}$	$\ln \frac{1+\alpha_1+\alpha_2}{2}$	$\ln \frac{1+\alpha_1+\alpha_2}{2\alpha_1}$
	o_R	$\ln \frac{1+\alpha_1+\alpha_2}{2\alpha_2}$	$\ln \frac{1+\alpha_1+\alpha_2}{2\alpha_1}$	$\ln \frac{1+\alpha_1+\alpha_2}{2}$

where $s(\mathbf{x}, \mathbf{y})$ is the sample covariance. We recognize the term MNCC as the modification of normalized cross-correlation due to Moravec [11]. Even though we have used just an estimate of σ_I, this normalization determines the success of the resulting algorithm in a fundamental way.

Assuming Markovianity, the structural term W in (1) can be expressed as

$$W(C, L) = W\big(\lambda(s)\big) + \sum_{p \in C, p \neq t} W\Big(\lambda\big(S_C(p)\big) \mid \lambda(p)\Big), \quad (7)$$

where $S_C(p)$ is the successor of p on st-path C. The conditional function $W\big(\lambda(p_2) \mid \lambda(p_1)\big)$ is derived from the joint energy function $W\big(\lambda(p_2), \lambda(p_1)\big)$, which represents the penalty for label incompatibility in neighboring nodes p_1, p_2 on C. Our choice of $W\big(\lambda(p_2), \lambda(p_1)\big)$ will be independent of the location of the label pair on the st-path. We will therefore write it as $W(\lambda_2, \lambda_1)$.

Note that the occluded segments of the st-path, i.e. the series of labels of the form $\mathrm{m} \to \mathrm{o} \to \cdots \to \mathrm{o} \to \mathrm{m}$, $\mathrm{o} \in \{o_L, o_R\}$, contribute to (7) by a factor proportional to the length of this segment. Since all st-paths are of equal length and since every change of disparity induces occlusion, we indirectly minimize the total variation of disparity along the st-path.

A concrete function W is chosen so that $W(\lambda_2, \lambda_1)$ corresponds to the negative logarithm of the probability function $p(\lambda_2, \lambda_1)$ representing the compatibility of neighboring labels. The choice in Tab. I is derived from natural conditions $p(\mathrm{m}, o_L) = p(o_L, \mathrm{m}) = p(m, o_R) = p(o_R, \mathrm{m}) \overset{\text{def}}{=} p(\mathrm{o}, \mathrm{m})$, $p(o_L, o_L) = p(o_R, o_R) \overset{\text{def}}{=} p(\mathrm{o}, \mathrm{o})$, $p(o_L, o_R) = p(o_R, o_L)$ and

$$\alpha_1 = \frac{p(o_L, o_R)}{p(\mathrm{o}, \mathrm{o})}, \quad \alpha_2 = \frac{p(\mathrm{o}, \mathrm{m})}{p(\mathrm{o}, \mathrm{o})}, \quad (8)$$

where $0 \leq \alpha_1 \leq 1$, and $0 < \alpha_2 \leq 1 + \alpha_1$.

Parameter α_1 permits ($\alpha_1 = 1$) or forbids ($\alpha_1 = 0$) mutual occlusion. Parameter α_2 permits ($\alpha_2 = 1 + \alpha_1$) or forbids ($\alpha_2 \to 0$) small objects in the scene. These parameters must be chosen according to the expected character of the scene. The resulting functions $W(\lambda_2 \mid \lambda_1)$ required by (7), are given in Tab. I (after subtracting $W(\mathrm{o} \mid \mathrm{m}) = \ln 2$ from each of them, which is possible since all st-paths have equal length).

The function $W(\lambda)$ is obtained by marginalization

$$W(\mathrm{m}) = \ln \frac{1 + \alpha_1 + \alpha_2}{2\alpha_2}, \quad W(o_L) = W(o_R) = 0, \quad (9)$$

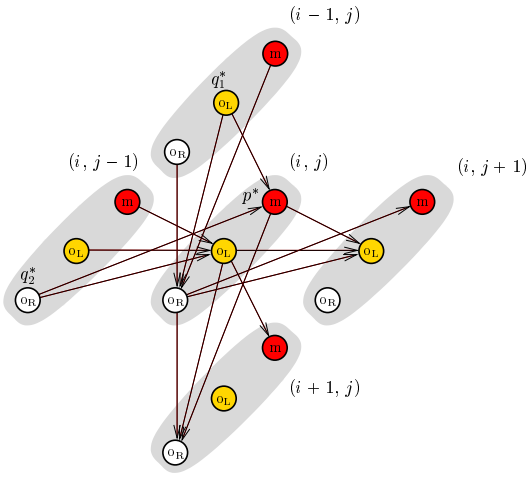


Fig. 8. Part of oriented graph G^*. The gray hyper-nodes are nodes of G, indices (i, j) represent their location in matching table.

from which we subtracted a common term of $\ln p(\mathrm{o}, \mathrm{o}) - W(\mathrm{o} \mid \mathrm{o})$.[3]

IV. THE 3LDP ALGORITHM

We now describe an algorithm that solves (1). The solution is similar to the method published in [12] or in an independently published work [13]. Our variant will be called 3LDP (*3-label dynamic programming stereo*).

The problem of finding an optimal labeled path will now be reduced to a minimum-cost path problem in an oriented circuit-free graph. We construct a new graph G^* from the original graph G by replacing each node of G with three nodes in G^* corresponding to labels m, o_L, o_R. Each edge of G will be replaced with a bipartite graph in G^*, whose edges connect consistent label pairs in neighboring nodes of G. The neighborhood of one of such triple of nodes in G^* is shown in Fig. 8. We are now finding the minimum-cost path in G^* that starts in either of the three nodes $s^*(\lambda)$ and ends in any of the three nodes $t^*(\lambda)$, where $\lambda \in \Lambda = \{\mathrm{m}, o_L, o_R\}$. The cost of path C^* sums all the node costs $V_p\big(D \mid \lambda(p)\big)$ and all the edge costs $W\big(\lambda(S_C(p)) \mid \lambda(p)\big)$ along C^*. Since G^* is circuit-free, there is a topological ordering of its vertices (e.g. by lexicographic ordering of the nodes of G^*) and we can use dynamic programming [14] (which is faster than a general minimum-cost path algorithm).

The elementary step of the forward pass of dynamic programming searches for the min-cost path C^* from the triple of starting nodes $s^*(\lambda)$ to each node $p^*(\lambda)$, $\lambda \in \Lambda$. This path must pass through some of the predecessors of $p^*(\lambda)$ in G^*. Consider, for instance, node $p^*(\mathrm{m})$ at location (i, j) with label m, as in Fig. 8. We assume we know the cost of the min-cost path $C^*(i-1, j, o_L)$ to node $q_1^*(o_L)$ at location $(i-1, j)$ with label o_L and the cost of the min-cost path $C^*(i, j-1, o_R)$ to node $q_2^*(o_R)$ at location $(i, j-1)$ with label o_R. The cost

[3] Note that if we wanted using a similar trick by subtracting α_0^{-1} from (6) then we must not forget subtracting it from V_o in (3) as well.

of the min-cost path to $p^*(\mathrm{m})$ can be obtained by 'solving' a small problem

$$\begin{aligned} C^*(i,j,\mathrm{m}) &= \min(c_1,c_2) + V_{(i,j)}(D \mid \mathrm{m}), \text{ where} \\ c_1 &= C^*(i-1,j,\mathrm{o_L}) + W(\mathrm{m} \mid \mathrm{o_L}), \\ c_2 &= C^*(i,j-1,\mathrm{o_R}) + W(\mathrm{m} \mid \mathrm{o_R}). \end{aligned} \tag{10}$$

If we start in node $s^*(\lambda)$, where we initialize three costs for $\lambda \in \Lambda$

$$C^*(1,1,\lambda) = W(\lambda) + V_{(1,1)}(D \mid \lambda), \tag{11}$$

then we can inductively compute the cost of the min-cost path to each of the nodes $t^*(\lambda)$, $\lambda \in \Lambda$ by repeated application of (10).

If we remember which node (i,j,λ) we came from in each min-cost path extension step (10) then the actual min-cost path C^* can be traced-out by a reverse pass in which we start in the node $t^*(\lambda)$ that received the optimal cost and then follow the stored back-links.

Nodes that received label m on C^* are the members of matching M. Their disparity $d = i - j$ is recorded in the output disparity map. If the matching table is $m \times n$ then the complexity of the algorithm is $O(m \cdot n)$, which does not exceed the complexity of computing V_p for all elements of the matching table. The algorithm is therefore optimal.

The algorithm processes just a single pair of epipolar lines. If the image is transformed so that epipolar lines correspond to image matrix rows, then i are the column indices in the left image and j are the column indices in the right image. Raw input images are easily transformed to this form by a pair of *epipolar rectification homographies*. Images in the top row of Fig. 10 are already rectified (which is the reason they are distorted a little). A suitable epipolar rectification algorithm is described in [15], a more general method that allows epipoles within the image(s) is [16].[4]

Fig. 9 lists the source code of 3LDP in plain C. The argument of the forward pass `DP3LForward` is the matching table, which is a structure consisting of the array of image similarities MNCC from (6) together with the beginning and the end of valid data in this array. Arrays `C_m`, `C_oL`, `C_oR` store the min-costs to each given node. Arrays `P_m`, `P_oL`, `P_oR` store the back-links for the reverse pass. The first three elements of the array `penalty` are the $W(\lambda_2 \mid \lambda_1)$ from Tab. I, multiplied with α_0 from (6). The last element is the $\alpha_0 V_\mathrm{o}$ from (3). Macros `MINi` and `MAXi` compute the beginning and the end of valid data in column `j` of the correlation table, where `tab.drange` is the expected disparity search range. Constraining this range is useful, since it speeds up image similarity computation considerably. Note also the initialization before the main cycle, it helps avoid testing an exception in the start nodes $s^*(\lambda)$.[5]

[4]I do not recommend the general method in cases when we know the epipoles will be outside of the images.

[5]The test for out-of bounds access to elements of array `C` is removed for clarity in Fig. 9.

```
#define clamp(x, mi, ma)\
 ((x) < (mi) ? (mi) : ((x) > (ma) ? (ma) : (x)))
#define MAXi(tab,j)\
 clamp((j)+(tab).drange[1], (tab).beg[0], (tab).end[0])
#define MINi(tab,j)\
 clamp((j)+(tab).drange[0], (tab).beg[0], (tab).end[0])

#define ARG_MIN2(Ca, La, C0, L0, C1, L1)       \
 if ((C0) < (C1)) {                            \
  Ca = C0; La = L0; } else { Ca = C1; La = L1; }

#define ARG_MIN3(Ca, La, C0, L0, C1, L1, C2, L2)      \
 if ( (C0) <= MIN(C1, C2) ) { Ca = C0; La = L0; }     \
 else if ( (C1) < MIN(C0, C2) ) { Ca = C1; La = L1; } \
 else { Ca = C2; La = L2; }

void DP3LForward(MatchingTableT tab) {

 int i = tab.beg[0]; int j = tab.beg[1];
 C_m[j][i-1] = C_m[j-1][i] = MAXDOUBLE;
 C_oL[j][i-1] = C_oR[j-1][i] = 0.0;
 C_oL[j-1][i] = C_oR[j][i-1] = -penalty[0];

 for(j = tab.beg[1]; j <= tab.end[1]; j++)
  for(i = MINi(tab,j); i <= MAXi(tab,j); i++) {

    ARG_MIN2(C_m[j][i], P_m[j][i],
             C_oR[j-1][i] + penalty[2], lbl_oR,
             C_oL[j][i-1] + penalty[2], lbl_oL);
    C_m[j][i] += 1.0 - tab.MNCC[j][i];

    ARG_MIN3(C_oL[j][i], P_oL[j][i], C_m[j-1][i], lbl_m,
             C_oL[j-1][i] + penalty[0], lbl_oL,
             C_oR[j-1][i] + penalty[1], lbl_oR);
    C_oL[j][i] += penalty[3];

    ARG_MIN3(C_oR[j][i], P_oR[j][i], C_m[j][i-1], lbl_m,
             C_oR[j][i-1] + penalty[0], lbl_oR,
             C_oL[j][i-1] + penalty[1], lbl_oL);
    C_oR[j][i] += penalty[3];
  }
}

void DP3LReverse(double *D, MatchingTableT tab) {

 int i,j; labelT La; double Ca;
 for(i=0; i<nl; i++) D[i] = nan; /* not-a-number */

 i = tab.end[0]; j = tab.end[1];
 ARG_MIN3(Ca, La, C_m[j][i], lbl_m,
          C_oL[j][i], lbl_oL, C_oR[j][i], lbl_oR);

 while (i >= tab.beg[0] && j >= tab.beg[1] && La > 0)
  switch (La) {
   case lbl_m: D[i] = i-j;
    switch (La = P_m[j][i]) {
     case lbl_oL: i--; break;
     case lbl_oR: j--; break;
     default: Error(...);
    } break;

   case lbl_oL: La = P_oL[j][i]; j--; break;
   case lbl_oR: La = P_oR[j][i]; i--; break;
   default: Error(...);
  }
}
```

Fig. 9. The core of the 3LDP algorithm: forward and reverse passes.

The reverse pass of the matching algorithm is performed by `DP3LReverse` in Fig. 9. This procedure writes disparity to the output array `D` for each left image pixel i.

V. 3LDP Results

A few example results of the 3LDP algorithm are shown in Fig. 10(d)–(f). Parameters were set to $\alpha_0 = 2.17, \alpha_1 = 1, \alpha_2 = 0.81, V_o = 0.083$. These were chosen so that the

algorithm achieves a chosen error rate and disparity map density (see next). The image neighborhood size for computing V_p was 5×5 pixels.

The result is worse than results from state-of-the art algorithms. For the purpose of this paper we sacrificed perfectness to simplicity of description. But we can say that despite the simplicity (and speed) of the algorithm, its results are relatively good. Fig. 10(f) shows the result for the most difficult reference scene *Tsukuba* [17] in a standard test suite [18], [19]. On this scene the matching algorithm, including MNCC computation, run 0.61 sec (CPU time, a single core of Intel T7500 at 2.2GHz, implementation partly in Matlab®, partly in C). The time is strongly dominated by computing image similarities V_p.

VI. A Simple Extension of 3LDP

An alternative matching method does not search for the min-cost solution but for a *stable* solution. Stability means insensitivity to small data perturbations. The simplest approach to stability in stereo has been proposed in [20]. The algorithm would first use the above dynamic programming only to compute the quality of each pair $p^*(\mathrm{m})$ in G^*. The quality is defined as the cost of the optimal path that passes through the node $p^*(\mathrm{m})$ in graph G^*.[6] The pair $p^*(\mathrm{m})$ is then accepted for the final matching when the min-cost of the optimal path through $p^*(\mathrm{m})$ is by a selected margin smaller (better) than the smallest-cost st-path that does not pass through $p^*(\mathrm{m})$ in G^*. This procedure may be seen as a variant of *cost aggregation* [18].

Better (denser) results can be obtained by taking a more general approach to stability. Instead of the hard thresholding, the individual nodes compete for selection in a stable matching algorithm described in [21]. Results of this modification are shown in Fig. 10(g)–(i).

For comparison, Figs. 10(j)–(l) show the results from maximum-likelihood matching augmented with a final match thresholding based on its cost (ML). The ML problem is obtained from (1) by choosing $W(C, L) = 0$ and not considering ordering. It can be solved by the Hungarian Algorithm [22].[7] The ML matching can be considered as the baseline algorithm.

VII. Which Matching Algorithm is The Best?

Many stereoscopic matching algorithms have been developed over the past four decades. They usually demonstrate the effectivity of some matching mechanism. None of them can be considered 'universally the best.' The most popular evaluation method has been developed in [18]. New results are continuously being published at the www address [19]. Their evaluation method assumes the matching algorithm can deliver

[6]The forward pass computes the min-cost path from the $s^*(\lambda)$ nodes to each node $p^*(\mathrm{m})$. A reverse pass computes the min-cost path from the $t^*(\lambda)$ nodes to $p^*(\mathrm{m})$ in a graph that is constructed from G^* by reversing all arcs. The cost of the optimal st-path passing through $p^*(\mathrm{m})$ is given as the sum of these two terms.

[7]In stereovision literature it is sometimes suggested that that the ML problem is solvable by dynamic programming. But that would not be a correct solution because it would not satisfy the uniqueness constraint.

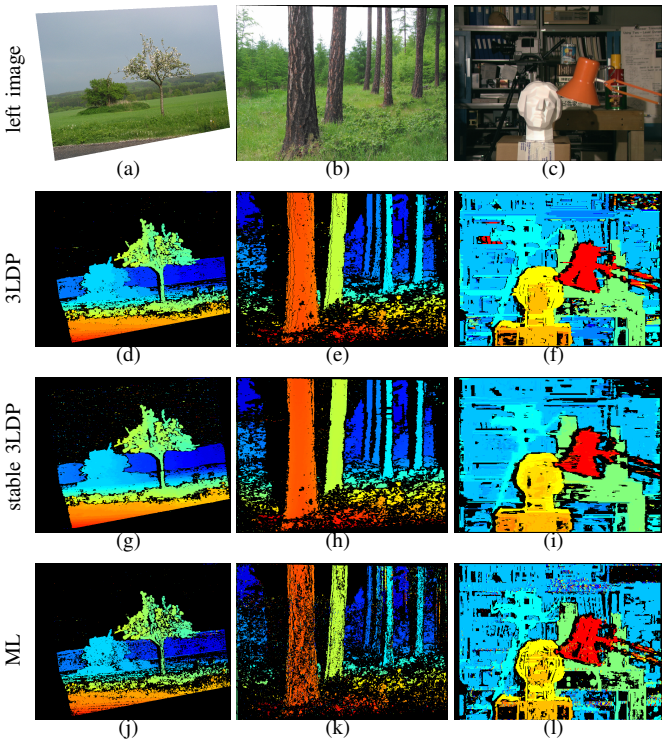


Fig. 10. Matching algorithm results on three non-trivial example scenes (a) – (c). Algorithm 3LDP with parameters set to mean error 3% achieves 76% mean density (d) – (f). The stable modification s3LDP achieves mean density of 81% for the same level of mean error (g) – (h), while the maximum-likelihood matching with thresholding (ML) achieves 61% density for this error level (j) – (l).

100% dense disparity map, including occlusions. One cannot measure disparity in occlusion but its most probable value can be estimated, based on a suitable continuity model (except perhaps in mutually occluded regions).

The 3LDP algorithm cannot be benchmarked using this method since the 100% density requirement is unnatural for an algorithm that is designed to model occlusion. Modeling occlusion is important if matching is to be used for 3D model reconstruction, since it is artifact-free (missing data from occlusions can be obtained 'by looking there'). Currently, there is no universally accepted method for benchmarking semi-dense matching algorithms. In this paper we briefly describe a simple method that is based on [23]. The method captures the entire density-accuracy tradeoff. It is based on a generalization of the concept of *Receiver Operating Characteristic* (ROC).

The ROC method uses ground-truth disparity maps. The maps are segmented to binocularly visible regions (denoted as G_B) and to occlusions (denoted as G_O). Given a disparity map D from an algorithm, we can measure two elementary characteristics: inaccuracy and density (sensitivity). In this paper, inaccuracy is defined as the number of pixels in G_B that are present in D with error greater than 0.75 pixels plus the number of pixels in G_O that received some disparity in D (a mismatch in occlusion). The total is normalized by the number of all image pixels. Density is defined as the number of pixels

in G_B that received some disparity in D (the rest are 'holes'). The total is normalized by the number of pixels in G_B.

The ROC method defines relative inaccuracy $\bar{a}$ and density d in percentage points. The pair $(\bar{a}, d)$ is a point which is obtained with some setting of the parameters of the tested algorithm (for instance by setting some V_o, α_0, α_1, α_2 in 3LDP). The upper envelope curve of all the points $(\bar{a}, d)$ obtained by varying all possible parameter settings is the (generalized) ROC curve. The region above the curve is reachable by no parameter setting of the given algorithm. It could be reachable by another (perhaps better) algorithm.

The ROC curve can be found by a simple genetic programming algorithm in which the points on the curve form the population. The moves include inserting points with random parameter settings, random mutations of the parameters of an individual point or crossover operation on two members of the population. Fitness of the population is defined as the area under the ROC curve. When a point appears above the curve or a point on the curve makes a step upward, the sub-population that appears under the new curve is removed.

If the ROC curve of algorithm A_1 is above the ROC curve of algorithm A_2 then we say that A_1 is strictly more informed than A_2 (A_1 is therefore a better algorithm). The area above the ROC curve is related to the error of a classifier that recognizes correct and incorrect correspondences [24]. Examples of ROC curves for various algorithms are shown in Fig. 11, where we can see, for instance, that 3LDP is strictly more informed than ML. The points $(\bar{a}, d)$ in the curves in Fig. 11 are computed as the mean values over a set of seven different ground-truth disparity maps: *Tsukuba, Venus, Sawtooth, Cones, Teddy* [19] and *Slits, Stripes* [25].

The advantages of a benchmarking based on the area above the ROC curve are that (1) one obtains a parameter-independent assessment of an algorithm that is still meaningful as the expected algorithm error, (2) the influence of all parts of the algorithm can be studied in detail: image similarity, the prior model, and the algorithm implementation itself, (3) the method works for both semi-dense and dense algorithms, (4) the curve can be used to select an optimal performance for a given application by choosing the proper balance between accuracy and density.

VIII. Conclusions

This paper tried to explain the essential components of the stereoscopic vision problem from the computational point of view. Because of space limitations, I could not explain all steps on an even level of detail, the reader necessarily finds some 'leaps in complexity.'

Stereovision is still an area of active research. For instance, there has been interest in finding out if image segmentation boosts matching performance (e.g. [26]). Results are very encouraging, segmentation algorithms occupy top ranks in the Middlebury evaluation [19]. Another interesting research problem is processing continuous stereoscopic video streams with the help of additional constraints on disparity map dynamics (e.g. [27], [28]).

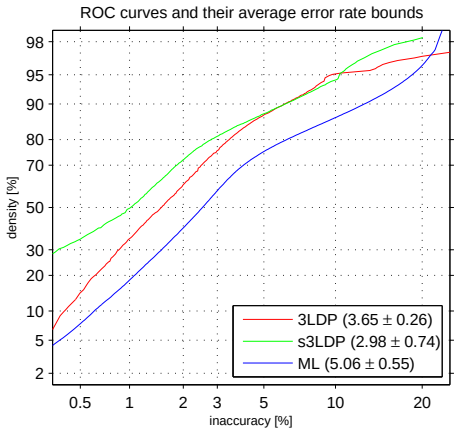


Fig. 11. The generalized ROC curve captures the degree to which various matching algorithms are informed. The ROC curve of a more informed algorithm is higher. The numbers in the legend give the area above the ROC curve in the range of $\langle 0, 100 \rangle$; double probability paper plot.

If the reader plans to experiment with his/her own matching algorithm, he/she should first try setting a reasonable performance goal. For many applications, simple algorithms already achieve good performance, as the disparity maps in Fig. 10 and ROC curves in Fig. 11 show.

Acknowledgment

This work has been supported by the Czech Ministry of Education under project MSM6840770012.

References

[1] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2003.

[2] D. Marr, "A note on the computation of binocular disparity in a symbolic, low-level visual processor," A.I. Lab, MIT, A.I. Memo 327, 1974.

[3] D. Marr and T. Poggio, "A theory of human stereo vision," A.I. Lab, MIT, A.I. Memo 451, November 1977.

[4] C. Wheatstone, "On some remarkable, and hitherto unobserved, phenomena of binocular vision," *Phil Trans Royal Soc London*, vol. 128, pp. 371–394, 1838.

[5] H. H. Baker and T. O. Binford, "Depth from edge and intensity based stereo," in *Proc Int Joint Conf on Artificial Intelligence*, 1981, pp. 631–636.

[6] J. D. Krol and W. A. van de Grind, "Rehabilitation of a classical notion of panum's fusional area," *Perception*, vol. 11, no. 5, pp. 615–619, 1982.

[7] P. Burt and B. Julesz, "A disparity gradient limit for binocular vision," *Science*, vol. 208, pp. 615–617, 1980.

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

[9] G. E. Healey and R. Kondepudy, "Radiometric CCD camera calibration and noise estimation," *IEEE T Pattern Anal*, vol. 16, no. 3, pp. 267–276, 1994.

[10] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th ed. Boston, USA: McGraw-Hill, 2002.

[11] H. P. Moravec, "Towards automatic visual obstacle avoidance," in *Proc Int Joint Conf on Artificial Intelligence*, 1977, p. 584.

[12] G. L. Gimel'farb, "Intensity-based computer binocular stereo vision: Signal models and algorithms," *International Journal of Imaging Systems and Technology*, vol. 3, no. 3, pp. 189–200, 1991.

[13] A. Criminisi, A. Blake, C. Rother, J. Shotton, and P. Torr, "Efficient dense stereo with occlusions for new view-synthesis by four-state dynamic programming," *Int J Computer Vision*, vol. 71, no. 1, pp. 89–110, 2007.

[14] R. E. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

[15] C. Loop and Z. Zhang, "Computing rectifying homographies for stereo vision," in *Proc IEEE Computer Soc Conf Computer Vision and Pattern Recognition*. IEEE Computer Society Press, 1999, pp. 125–131.

[16] M. Pollefeys, R. Koch, and L. V. Gool, "A simple and efficient rectification method for general motion," in *Proc IEEE Int Conf Computer Vision*, vol. 1. IEEE Computer Society Press, 1999, pp. 496–501.

[17] Y. Nakamura, T. Matsuura, K. Satoh, and Y. Ohta, "Occlusion detectable stereo – occlusion patterns in camera matrix," in *Proc IEEE Computer Soc Conf Computer Vision and Pattern Recognition*, 1996, pp. 371–378.

[18] D. Scharstein, R. Szeliski, and R. Zabih, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," in *Proc IEEE Workshop on Stereo and Multi-Baseline Vision*. IEEE Computer Society Press, December 2001.

[19] D. Scharstein and R. Szeliski, "Middlebury stereo vision page," [on-line] http://vision.middlebury.edu/stereo/, 2007.

[20] M. Gong and Y.-H. Yang, "Fast stereo matching using reliability-based dynamic programming and consistency constraints," in *Proc IEEE Int Conf Computer Vision*. IEEE Computer Society Press, 2003.

[21] R. Šára, "Robust correspondence recognition for computer vision," in *Proc COMPSTAT 2006: 17th ERS-IASC Symposium on Computational Statistics*. Physica-Verlag, 2006, pp. 119–131.

[22] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society of Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[23] J. Kostlivá, J. Čech, and R. Šára, "Feasibility boundary in dense and semi-dense stereo matching," in *Proc BenCOS 2007: CVPR Workshop Towards Benchmarking Automated Calibration, Orientation and Surface Reconstruction from Images*. Omnipress, June 2007.

[24] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[25] J. Kostková, J. Čech, and R. Šára, "Dense stereomatching algorithm performance for view prediction and structure reconstruction," in *Proc Scandinavian Conf on Image Analysis*, vol. LNCS 2749. Springer Verlag, 2003, pp. 101–107.

[26] M. Bleyer and M. Gelautz, "Graph-cut-based stereo matching using image segmentation with symmetrical treatment of occlusions," *Signal Processing: Image Communication*, vol. 22, no. 2, pp. 127–143, 2007.

[27] M. Sizintsev and R. P. Wildes, "Spatiotemporal stereo via spatiotemporal quadric element (stequel) matching," in *Proc IEEE Computer Soc Conf Computer Vision and Pattern Recognition*. IEEE Computer Society, 2009, pp. 493–500.

[28] M. Bleyer and M. Gelautz, "Temporally consistent disparity maps from uncalibrated stereo videos," in *Proceedings 6th Int Symp Image and Signal Processing and Analysis*. IEEE Computer Society, 2009.