Extended Dynamic Programming and Fast Multidimensional Search Algorithm for Energy Minization in Stereo and Motion

Mikhail G. Mozerov

Abstract—This paper presents a novel extended dynamic programming approach for energy minimization (EDP) to solve the correspondence problem for stereo and motion. A significant speedup is achieved using a recursive minimum search strategy (RMS). The mentioned speedup is particularly important if the disparity space is 2D as well as 3D. The proposed RMS can also be applied in the well-known dynamic programming (DP) approach for stereo and motion. In this case, the general 2D problem of the global discrete energy minimization is reduced to several mutually independent sub-problems of the one-dimensional minimization. The EDP method is used when the approximation of the general 2D discrete energy minimization problem is considered. Then the RMS algorithm is an essential part of the EDP method. Using the EDP algorithm we obtain a lower energy bound than the graph cuts (GC) expansion technique on stereo and motion problems. The proposed calculation scheme possesses natural parallelism and can be realized on graphics processing unit (GPU) platforms, and can be potentially restricted further by the number of scanlines in the image plane. Furthermore, the RMS and EDP methods can be used in any optimization problem where the objective function meets specific conditions in the smoothness term.

Index Terms—Minimum search, fast algorithm, dynamic programming, discrete energy minimization, stereo matching, motion estimation.

✦

1 INTRODUCTION

STEREO and motion matching are used in many applications and remains one of the most challenging open problems in computer vision [1], [2]. The most successful algorithms apply a global correspondence strategy based on an energy-minimization framework [3], [4].

Historically, DP was one of the first attempts to solve the stereo matching problem by minimizing the global energy [5], [6]. It was called scanline optimization when the global 2D problem was split into a many one dimensional sub-problems [7], [8]. Practically, the same DP approach was applied for the motion correspondence problem when a a multidimensional disparity space was considered [9], [11]. In spite of state-of-the-art results in dense disparity map reconstruction [12], [13], [14], [15], [16] the level of global energy minimization of the DP optimization was not satisfactory due to the loss of several essential prior dependencies in the smoothness term of the discrete energy function, e.g. the vertical dependencies in stereo matching in scanline optimization.

To obtain an exact solution of the energy minimization problem, the GC method for linear prior dependencies was proposed by Roy and Cox in [17]. This approach was extended for convex prior dependencies by Ishikawa in [18]. The computational complexity of both algorithms

in case of an exact solution is rather high, especially the Ishikawa approach. Therefore, many GC techniques that aim to obtain an approximate solution have been proposed [19], [20], [21], [22], [23], [24].

Among the GC techniques, we give special consideration to the GC expansion method [19] for two main reasons. The first is that the expansion algorithm still is very popular in computer vision community because it is relatively fast and obtains excellent approximate solutions to the energy minimization problem (only a few algorithms were reported to achieve lower energy level [24], [25], [32], [33]; but the computational complexity in these cases is higher than in [19]). The second is that Middlebury provides an open C++ code for the expansion algorithm which we use to make comparisons with our EDP approximation. In general, GC techniques cannot be implemented for parallel calculation schemes and there is no theoretical guarantee that the parallelization will be faster for every problem instance [34]. In contrast, belief propagation (BP) [25], [26], [27], [28] and DP approaches possess a natural parallelism and can be realized on the GPU platform, as done in [35], for example.

The general stereo and motion correspondence problem assumes a multidimensional disparity space (usually 2D or 3D domain). In this case, the computational complexity of straightforward search of DP algorithms is extremely high [9], [10], [11], [13], and a fast algorithm is highly desirable. For this purpose, we have designed a RMS algorithm that reduces the computational complexity of a straightforward search.

The proposed algorithm results in a considerable com-

• *M. Mozerov is with the Computer Vision Center of Department Informatics, Universitat Autnoma de Barcelona, Barcelona, Spain, 08193.*
E-mail: mozerov@cvc.uab.es

plexity reduction - that is, $O\left(NQ^2\right) \rightarrow O\left(NWQ\right) \rightarrow O\left(NQ\right)$, for an image with N pixels, and a search domain cardinality Q. A constant W is the cardinality of a constrained search subdomain of the RMS algorithm: $W << Q$. We have to note that the solution of the fast RMS algorithm is the same as the solution of the straightforward search. To show that the DP approach still remains competitive even for energy minimization problem, we propose in this paper an improved extension of DP. The so-called EDP algorithm aims to obtain an approximate solution and the method inherits ideas that were proposed in [29], [30]. However, using this new modification of the DP approach we obtain a lower energy than the expansion algorithm [19] on stereo and motion problems. We have to note that the EDP is a kind of DP, which ignores the uniqueness constraint like other approaches based on energy minimization do (GC, BP, etc.), and unlike DP with ordering does.

The paper is organized as follows: In Section 2 the problem is formulated. In Section 3 we describe the fast RMS algorithm. The EDP algorithm is described in Section 4. Computer experiments of the proposed techniques are discussed in Section 5. Section 6 summarizes our conclusions.

2 PROBLEM DEFINITION

The general stereo and motion matching approach aims to find correspondence between pixels of images $I_t\left(\mathbf{x}\right)$ and $I_{t+1}\left(\mathbf{x}\right)$ as it is shown in Fig. 1, where $\mathbf{x}$ is a coordinate of a pixel in the image plane, t is an index of the considered image in the sequence. A vector $\mathbf{v}\left(\mathbf{x}_t\right)$ in Fig. 1 denotes the disparity of two corresponding pixels $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$. In general, the image point coordinate $\mathbf{x}$ is a multidimensional vector, it is 2D in the case of stereo and dynamic images, 3D in the case of a tomography image. The disparity vector $\mathbf{v}$ has usually the same dimensionality as the domain of the image except in certain special cases: for instance, if stereo matching is considered the disparity vector domain becomes one dimensional due to the additional epipolar constraints. Simply expressed, if the stereo or motion matching problem is considered a dense disparity map $\mathbf{v}(\mathbf{x})$ has to be obtained. The question is how to solve such a problem in the most appropriate way. The global energy minimization approach intends to find a desired disparity function $\mathbf{v}(\mathbf{x})$, which minimizes the energy function $E(\mathbf{v}(\mathbf{x}))$ in the disparity space image (DSI) $C(\mathbf{x}, \mathbf{v})$, see Fig. 2. The DSI represents a collection of correspondence cost. For example, if two compared pixels $\left(\mathbf{x}_1\right)_t$ and $\left(\mathbf{x}_1 + \mathbf{v}_1\right)_{t+1}$ have the same luminance value (which means that these pixels are a potential match) the cost value $C\left(\mathbf{x}_1, \mathbf{v}_1\right)$ might be minimal. Vice versa, if the luminance values differ, the related cost value increases. The global energy usually contains two terms, the data term and the smoothness term

$$E\left(\mathbf{v}(\mathbf{x})\right) = \sum_{\mathbf{x} \in \Omega} C\left(\mathbf{x}, \mathbf{v}(\mathbf{x})\right) + \sum_{\mathbf{x} \in \Omega} G\left(\mathbf{v}(\mathbf{x})\right), \quad (1)$$
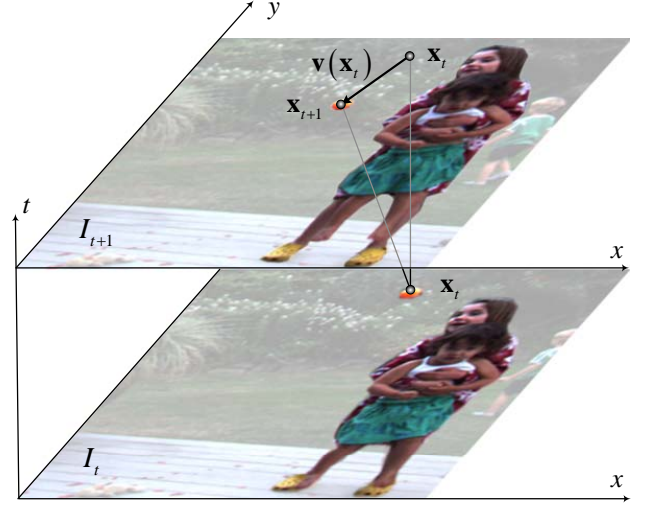

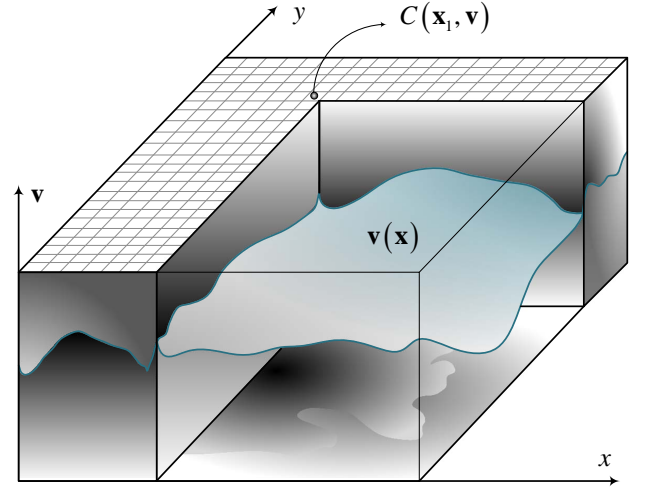Fig. 1. Scheme of correspondence matching between two images.

Fig. 2. The DSI as a collection of correspondence costs $C\left(\mathbf{x}, \mathbf{v}\right)$ and a desired function of disparity value $\mathbf{v}(\mathbf{x})$.

where G is a smoothness function and $\mathbf{\Omega}$ is the domain of the vector $\mathbf{x} = \{x_1, x_2, .., x_D\}$ or $\{x, y, .., z\}$ with the cardinality $|\mathbf{\Omega}| = N$. The domain of the vector $\mathbf{v} = \{v_1, v_2, .., v_R\}$ is denoted as $\mathbf{V}$ with the cardinality $|\mathbf{V}| = Q$. The image domain that we have to work with is a discrete vector space, and it is reasonable to consider vector components x_d and v_r as proportional to the natural numbers (e.g. $\alpha\left(v_r + v_r^0\right), \beta\left(x_d + x_d^0\right) \in \mathbb{N}$). Since α, β, v_r^0 and x_d^0 can be any arbitrary real constants, our discretization model allows subpixel accuracy. If the discretization step in the disparity domain is less than in the image domain, the accuracy of the disparity estimation is obviously at subpixel level. Thus, a substitution of such variables for the natural scale is trivial, and further in this paper we will assume that the natural numeration of the indexes in the multidimensional vector grid coincide with the values of the discrete vectors $\mathbf{x}$ and

$\mathbf{v}$. Consequently, the conditions of the mentioned vector domain is

$$v_r \in \{1, 2, .., v_r^{\max}\}, v_r^{\max} \in \mathbb{N}, \\ x_d \in \{1, 2, .., x_d^{\max}\}, x_d^{\max} \in \mathbb{N}. \quad (2)$$

The cost values $C(\mathbf{v}(\mathbf{x}))$ that form a DSI might be computed as follows

$$C(\mathbf{x}, \mathbf{v}) = |I_{t+1}(\mathbf{x} + \mathbf{v}) - I_t(\mathbf{x})|^{1,2} \wedge C_{\max}, \quad (3)$$

where I_{t+1} and I_t are luminance values of two neighboring images in stereo or dynamic sequences, $\wedge$ is a sign of the pairwise **min** value operation, $C_{\max}$ is a cost truncation constant. We also use $\bigwedge$ as a sign of **min** value operation defined on a set of variables. The cost truncation constant $C_{\max}$ is introduced in our paper for some practical reasons: this parameter is referred in the experimental section. If $C_{\max} = \infty$ then the comparison metric (3) is the most popular L_1 or L_2 norm. Nonetheless, there are many other cost metrics (see e.g. [1], [2]) that could be used for the DSI formation.

The function G in the smoothness term of (1) is given by

$$G(\mathbf{v}(\mathbf{x})) = \lambda \sum_{d \in D} \sum_{r \in R} f(|\mathbf{v}_r(x_d + 1) - \mathbf{v}_r(x_d)|), \quad (4)$$

where λ is a constant used to penalize motion vector discontinuities, R and D are the dimensionalities of motion vector and image spaces respectively. Later on in this paper a shortcut

$$f(|\mathbf{v}(x_d + 1) - \mathbf{v}(x_d)|) = \\ \sum_{r \in R} f(|\mathbf{v}_r(x_d + 1) - \mathbf{v}_r(x_d)|) \quad (5)$$

is used for the distance measure. A positive definite increasing function f is usually proportional to the gradient of the motion vector or its squared value

$$G(\mathbf{v}(\mathbf{x})) = \lambda \sum_{d \in D} |\mathbf{v}(x_d + 1) - \mathbf{v}(x_d)|^{1,2}, \quad (6)$$

To prevent over-penalizing discontinuity a more flexible smoothness function is used

$$\bar{G}(\mathbf{v}(\mathbf{x})) = \lambda \sum_{d \in D} f(|\mathbf{v}(x_d + 1) - \mathbf{v}(x_d)|) \wedge \lambda f(g), \quad (7)$$

where g is a truncation threshold and $\lambda(\mathbf{x})$ is locally adaptive in such away that $\lambda(\mathbf{x}) = 2\lambda$ if a value of a local gradient of an image $I_t(\mathbf{x})$ is less than 10 and $\lambda(\mathbf{x}) = 2\lambda$ otherwise. The meaning of this threshold becomes clearer if we consider disparity map estimation in stereo for the most popular truncated linear prior, e.g. the case of $f(|a|) = |a|$ in (7). Indeed, the smoothness rate is proportional to the disparity difference and matters only inside an object segment, where this difference is supposed to be relatively small. In contrast, there is no reason to penalize a solution more for being 20 instead of 5, for example, if it is known that both values belong to the same class of discontinuity: say foreground object - background surface. We suppose that the positive
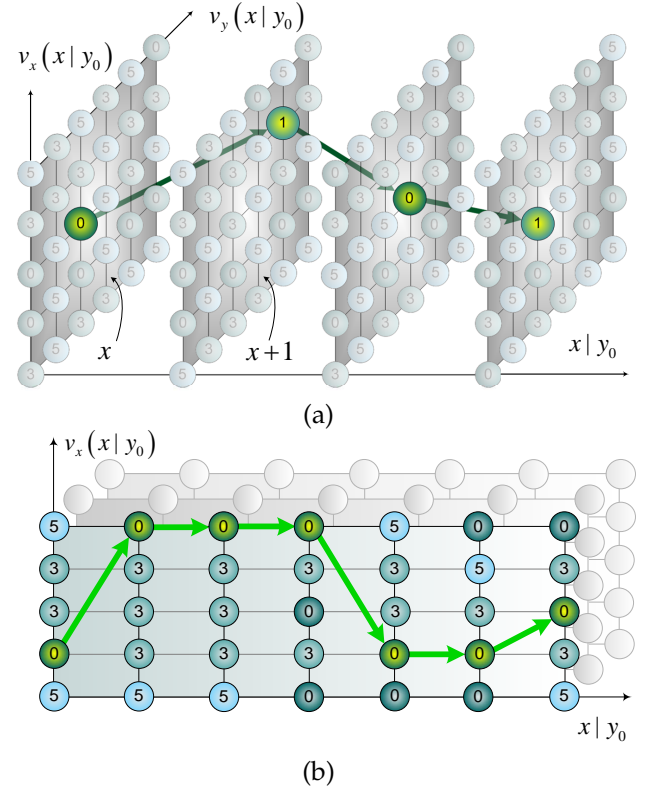

Fig. 3. The optimal path through the DSI trellis, (a) for 2D motion, (b) for stereo.

definite threshold g belongs to the domain as $|v_r|$, in other words is a natural number.

The general 2D problem of the global energy minimization can be reduced to several mutually independent sub-problems of the one-dimensional minimization if several additional constraints of the smoothness term are considered, e.g. the vertical prior dependencies are equal to zero. Then, the solution is given by

$$\bar{\mathbf{v}}(x|y_0) = \arg \min_{\mathbf{v}} \left\{ \begin{array}{l} \sum_{x \in \Omega_x} C(x, y_0, \mathbf{v}(x|y_0)) + \\ \lambda \sum_{x \in \Omega_x} (f(|\mathbf{u}(x|y_0)|) \wedge f(g)) \end{array} \right\}, \quad (8)$$

where y_0 is the fixed index of the considered sub-problem and $f(|\mathbf{u}(x|y_0)|) = f(|\mathbf{v}(x+1|y_0) - \mathbf{v}(x|y_0)|)$. Each discrete function $\mathbf{v}(x|y_0)$ coincides with the optimal (minimal energy) path through the 3D trellis (Fig. 3(a)) for motion, which is a slice of the initial 4D DSI. The optimal path for stereo is illustrated in Fig. 3(b). Hereafter, the fixed index y_0 of (8) is omitted for notational simplicity.

Such problems can be solved by means of the DP algorithm. The method consists of step-by-step control and optimization. Let $S(x, \mathbf{v}(x))$ be the optimal path cost to vertex $(x, \mathbf{v}(x))$, see Fig. 3. Then the optimal path cost to vertex $(x + 1, \mathbf{v}(x + 1))$ is given by the recurrence
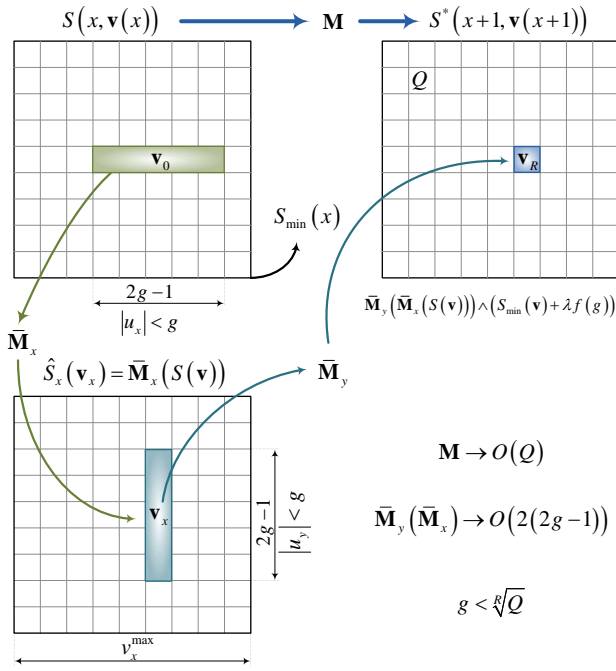
Fig. 4. The straightforward minimum search versus the RMS algorithm.

relation

$$S(x+1, \mathbf{v}(x+1)) = C(x+1, \mathbf{v}(x+1)) + \bigwedge_{\mathbf{v}(x)} S(x, \mathbf{v}(x)) + \lambda f(|\mathbf{v}(x+1) - \mathbf{v}(x)|) \wedge \lambda f(g). \quad (9)$$

$$\underbrace{\qquad\qquad\qquad\qquad}_{S^*(x+1, \mathbf{v}(x+1))}$$

Equation (9) is an essential part of any DP based technique. It is obvious that the calculation of the sum $S^*(x+1, \mathbf{v}(x+1))$ in (9) is the most time consuming operation. Therefore, let $\mathbf{M}\left(S(x, \mathbf{v}(x)), \bar{G}(x, \mathbf{v}(x))\right) = \mathbf{M}(S(x, \mathbf{v}(x)))$ be the operator that transforms the full array of optimal cost $S(x, \mathbf{v}(x))$ at the step x (Fig. 3 and Fig. 4) to the additional sum $S^*(x+1, \mathbf{v}(x+1))$ at the step $x+1$

$$S^*(x+1, \mathbf{v}(x+1)) = \mathbf{M}(S(x, \mathbf{v}(x))) \quad (10)$$

Then, the computational complexity of the operator $\mathbf{M}$ that is used in step-by-step optimization (9) via (10) is proportional to $O(Q)$ per each vertex $(\mathbf{x}, \mathbf{v})$ in the DSI space, see illustration in Fig. 4, where Q is the number of labels in the multidimensional disparity space for the energy minimization problem.

Therefore, the main idea of the RMS approach is to make the DP step-by-step operator $\mathbf{M}$ significantly faster, and this idea will be explained in the next section.

3 RECURSIVE SEARCH ALGORITHM

This section is subdivided into two subsections. The first subsection explains the general case of the RMS algorithm, and the second subsection introduces the RMS technique for the truncated linear prior, because in this case a further speedup can be obtained.

3.1 General RMS Algorithm

By the distributive and associative laws of the minimum-addition semi-ring over reals (or integers) one can get from (9)

$$S^*(x+1, \mathbf{v}(x+1)) =$$
$$\underbrace{\underset{\mathbf{v}(x): f(|\mathbf{u}(x)|) < f(g)}{MIN} \begin{pmatrix} S(x, \mathbf{v}(x)) + \\ \lambda f(|\mathbf{u}(x)|) \end{pmatrix}}_{\hat{S}(x+1, \mathbf{v}(x+1))} \wedge$$
$$\underbrace{\underset{\mathbf{v}(x)}{MIN}(S(x, \mathbf{v}(x)) + \lambda f(g))}_{S_{\min}(x) + \lambda f(g)}, \quad (11)$$

in which we used a shortcut $\mathbf{u}(x) = \mathbf{v}(x+1) - \mathbf{v}(x)$.

The term $\hat{S}(x+1, \mathbf{v}(x+1))$ says that one has a natural limit on the minimum search range. The term of $S_{\min}(x) = \bigwedge_{\mathbf{v}(x)}(S(x, \mathbf{v}(x)))$ is easy to obtain as a by-product of the previous phase of the forward pass of the DP algorithm. In fact, assuming $f(g) > 0$, the $f(g)$ in $f(|\mathbf{u}(x)|) < f(g)$ in (11) can be replaced with $f(|\mathbf{u}(x)|) < a f(g)$ where $a \geq 1$ is a real constant. This merely corresponds to enlarging the search range. The value of S^* will not change if we choose some $a > 1$ due to clipping by the value of $S_{\min}(x) + \lambda f(g)$. This helps us decompose the term $\hat{S}$ recursively, per dimension: There is always some finite $a \geq 1$ so that

$$\hat{S}(x+1, \mathbf{v}(x+1)) =$$
$$\underset{f(|\mathbf{u}(x)|) < a f(g)}{MIN} \left\{ S(x, \mathbf{v}(x)) + \lambda \sum_{i=1}^{R} f(|u_i(x)|) \right\} =$$
$$\underset{f(|u_R|) < f(g)}{MIN} \cdots \underset{f(|u_1|) < f(g)}{MIN} \begin{pmatrix} S(x, \mathbf{v}(x)) + \lambda f(|u_1|) + \\ \lambda f(|u_2|) + \ldots + \lambda f(|u_R|) \end{pmatrix} =$$
$$\underset{|u_R| < g}{MIN} \cdots \underset{|u_2| < g}{MIN} \underset{|u_1| < g}{MIN} (S + \lambda f(|u_1|)) + \lambda f(|u_2|)$$
$$\quad (12)$$

in which we used a shortcut $u_i = v_i(x+1) - v_i(x)$ and $f(|u_i|) < f(g)$, thus the f is a positively definite increasing function by our definition.

Therefore, the per dimensional decomposition of the straightforward minimum search in (12) is the key property the RMS algorithm, which gives significant speedup of the sum $S^*(x+1, \mathbf{v}(x+1))$ calculation in the step-by-step optimization in (9). Now, let

$$\hat{S}_r(\mathbf{v}_r) = \bar{\mathbf{M}}_r\left(\hat{S}_{r-1}(\mathbf{v}_{r-1})\right) =$$
$$\bigwedge_{|u_r| < g}\left(\hat{S}_{r-1}(\mathbf{v}_{r-1}) + \lambda f(g)\right), \quad (13)$$

then from (12)

$$S^*(x+1, \mathbf{v}(x+1)) = \mathbf{M}(S(x, \mathbf{v}(x))) =$$
$$\quad (14)$$
$$\bar{\mathbf{M}}_R\left(\ldots \bar{\mathbf{M}}_2\left(\bar{\mathbf{M}}_1(S(x, \mathbf{v}(x)))\right)\right) \wedge (S_{\min}(x) + \lambda f(g)),$$

where the $\mathbf{v}_r$ is an intermediate domain

$$\mathbf{v}_r = \{v_R(x), \ldots v_{r+1}(x), v_r(x+1), \ldots, v_1(x+1)\}, \quad (15)$$

introduced to make sense of the variable u_r in (12), (13) and is used for calculation of the intermediate function
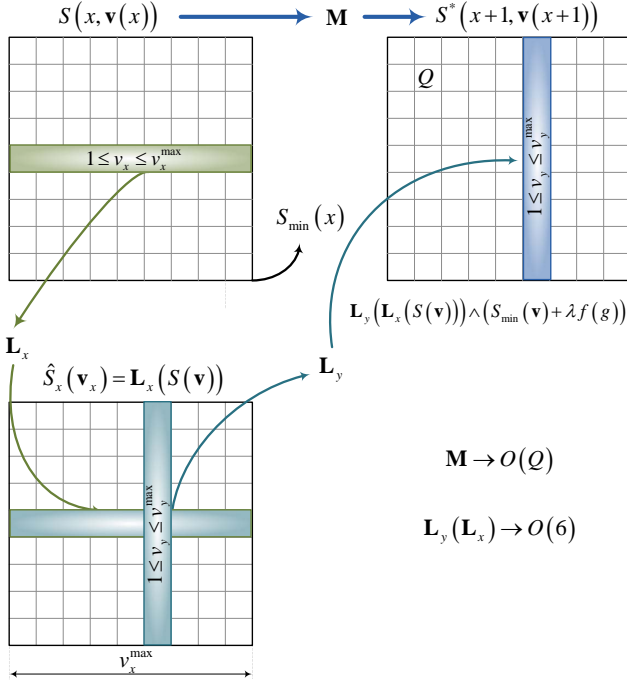
Fig. 5. The straightforward minimum search versus the RMS algorithm for the truncated linear prior.

$\hat{S}_r(\mathbf{v}_r)$. Then obviously $\hat{S}_R(\mathbf{v}_R) = \hat{S}(x+1, \mathbf{v}(x+1))$ and $\hat{S}_0(\mathbf{v}_0) = S(x, \mathbf{v}(x))$.

The application of the fast calculation scheme derived in (11) and (12) for the 2D case is illustrated in Fig. 4, in which the sum $\hat{S}_x$ denotes an intermediate results of the application of the operator $\bar{\mathbf{M}}_x$. The computational complexity of each one-dimensional operator $\bar{\mathbf{M}}_r$ is $2g-1$ operations per each vertex $(\mathbf{x}, \mathbf{v})$ in the DSI space. The application of all R operators gives $R(2g-1)$ per each vertex $(\mathbf{x}, \mathbf{v})$ instead of Q. Note that the maximum value of the threshold g in general is $g << \sqrt[R]{Q}$, and this is a significant speedup especially in the multidimensional case shown in the experimental section of the paper. If the smoothness term is a truncated linear prior and $g > 2$, further speedup can be achieved as it is shown in the next subsection.

3.2 RMS Algorithm for Truncated Linear Prior

The value of S^* will not change if we replace a domain $v_i : |u_i| < g$ in (12) by $v_i : |u_i| \leq v_r^{\max} < \infty$ due to clipping by the value of $S_{\min}(x) + \lambda f(g)$. Taking the linear prior $f(|u_r|) = |u_r|$ we can replace $\bar{\mathbf{M}}_r$ in (14) by an operator $\mathbf{L}_r$, which is defined as

$$\mathbf{L}_r\left(\hat{S}_{r-1}(\mathbf{v}_{r-1})\right) = \bigwedge_{|u_r| \leq v_r^{\max}} \left(\hat{S}_{r-1}(\mathbf{v}_{r-1}) + \lambda |u_r|\right). \quad (16)$$

Let us rewrite (16) by taking $v' = v_r(x+1)$, $v = v_r(x)$, $u = v' - v$ and omitting some unimportant components $\mathbf{v}_r/v'$ and $\mathbf{v}_{r-1}/v$ of the vector domains $\mathbf{v}_r$ and $\mathbf{v}_{r-1}$, by using the distributive and associative laws of the

minimum-addition semi-ring and property of the absolute value operation

$$\hat{S}_r(v') = \bigwedge_{|u| \leq v_r^{\max}} \left(\hat{S}_{r-1}(v) + \lambda |u|\right) =$$
$$\underbrace{\bigwedge_{0 \leq u}\left(\hat{S}_{r-1}(v) + \lambda u\right)}_{\hat{S}_r^+(v')} \wedge \underbrace{\bigwedge_{u<0}\left(\hat{S}_{r-1}(v) - \lambda u\right)}_{\hat{S}_r^-(v')}. \quad (17)$$

The sums $\hat{S}_r^-(v')$ and $\hat{S}_r^+(v')$ can be calculated recursively

$$
\begin{aligned}
\hat{S}_r^+(v') &= \bigwedge_{v \leq v'} \left(\hat{S}_{r-1}(v) + \lambda v' - \lambda v\right), \\
\hat{S}_r^+(v'+1) &= \bigwedge_{v \leq v'+1} \left(\hat{S}_{r-1}(v) + \lambda v' - \lambda v\right) + \lambda \\
&= \left(\hat{S}_r^+(v') + \lambda\right) \wedge \hat{S}_{r-1}(v'+1),
\end{aligned}
\quad (18)
$$

$$\hat{S}_r^+(1) = \hat{S}_{r-1}(1),$$

and

$$
\begin{aligned}
\hat{S}_r^-(v') &= \bigwedge_{v' < v} \left(\hat{S}_{r-1}(v) - \lambda v' + \lambda v\right), \\
\hat{S}_r^-(v'-1) &= \bigwedge_{v' < v+1} \left(\hat{S}_{r-1}(v) - \lambda v' + \lambda v\right) + \lambda \\
&= \left(\hat{S}_r^+(v') + \lambda\right) \wedge \hat{S}_{r-1}(v'-1),
\end{aligned}
\quad (19)
$$

$$\hat{S}_r^+(v_r^{\max}) = \infty.$$

Finally, we get the fast operator $\mathbf{L}_r$

$$\hat{S}_r(v') = \mathbf{L}_r\left(\hat{S}_{r-1}(v)\right) = \hat{S}_r^+(v') \wedge \hat{S}_r^-(v'), \quad (20)$$

that consists of two pass recursion described in (18) and in (19) and the additional comparison pass in (20).

The process of the fast operator $\mathbf{L}_r$ should be done in the whole subdomain $1 \leq v_r(x) \leq v_r^{\max}$ and the result of the application of the $\mathbf{L}_r$ have to be placed to the intermediate buffer for the application of the next fast operator $\mathbf{L}_{r+1}$ if $R > 1$ like it is illustrated in Fig. 5. In this illustration we have the 2D disparity space with two instances of the application of the fast operator $\mathbf{L}_x$ and $\mathbf{L}_y$. The number of the intermediate buffers in this case is one.

The sum S^* in step-by-step optimization process (9) for the truncated linear prior now is given by

$$S^*(x+1, \mathbf{v}(x+1)) = \mathbf{M}(S(x, \mathbf{v}(x))) =$$
$$\mathbf{L}_R(...\mathbf{L}_2(\mathbf{L}_1(S(x, \mathbf{v}(x))))) \wedge (S_{\min}(x) + \lambda f(g)). \quad (21)$$

The computational complexity of the fast operator $\mathbf{L}_r$ is exactly $O(3)$ per vertex in the DSI space and the speedup of the application of the operator $\mathbf{L}_r$ in (21) makes sense only for the threshold value $g > 2$.

Summarizing the RMS approach we have to note that there are three ways of calculating the sum S^* in the step-by-step optimization process (9):

- SFMS - The straightforward minimum search using directly the expression in the formula (9);

- GRMS - The general RMS algorithm summarized in (14);
- LRMS - The RMS algorithm for the truncated linear prior summarized in (21).

The above abbreviations are referred in comparison tables in the experimental section of the paper.

4   DP AND EDP ALGORITHMS

The advantage of the DP approach is shown for the one-dimensional case by comparison with other methods. For example, the BP algorithms need $x^{\max}$ iterations to achieve the exact solution of the problem described in (8) instead of one in the case of the DP approach application. In this section we describe the EDP algorithm that is an extension of the general DP technique to the multidimensional case. However, in the first subsection we have to explain the computational scheme of the standard realization of the DP algorithm and its modification for the one-dimensional case.

4.1   DP Algorithm

To make our algorithm more flexible, we omit the initialization step of the classical DP technique in (9). Instead, we take the optimal sums in (9) equal to zero for all pixels that are not included into the image domain

$$S(x, \mathbf{v}) \equiv 0, \ and \ \mathbf{M}(S(x, \mathbf{v})) \equiv 0, \forall x \notin \Omega_x, \qquad (22)$$

this condition will hold for all algorithms described in our paper. By using the recurrence relation in (9) the minimal value of the objective function in (8) can be found at the last step of optimization as

$$S_{\min}(x^{\max}, \bar{\mathbf{v}}) = \bigwedge_{\mathbf{v}} S(x^{\max}, \mathbf{v}), \qquad (23)$$

in which $\bar{\mathbf{v}}$ is the value of the disparity where the sum $S(x^{\max}, \mathbf{v})$ reaches its minimum and, in the same time, it is the first value $\bar{\mathbf{v}}(x^{\max})$ of the desired solution. Additionally, it is the starting vertex for the backward optimal path recovery process when the algorithm works in reverse order and recovers a sequence of optimal steps as it is illustrated in Fig. 6(a). Where the forward steps include the storage of the optimal choice indexes, which are used in the backward steps.

The sum $S(x^{\max})$ is the marginal function for the support domain of optimality $\{1, ..., x, x+1, ..., x^{\max}\}$. If we change the direction of the recurrence (starting from the vertex $x^{\max}$ to the vertex 1) the marginal function for the support domain of optimality $\{x^{\max}, ..., x, x-1, ..., 1\}$ will be the $S(1)$. Let the sum $S_1(x-1)$ be the marginal function for the support domain of optimality $\{1, 2, ..., x-1\}$ and the sum $S_{-1}(x+1)$ be the marginal function for the support domain of optimality $\{x^{\max}, x^{\max}-1, ..., x+1\}$, then the marginal function $S_{\Omega_x}(x)$ for the full domain $x \in \Omega_x$ is

$$S_{\Omega_x}(x, \mathbf{v}) = \mathbf{M}(S_1(x-1, \mathbf{v})) + C(x, \mathbf{v}) + \mathbf{M}(S_{-1}(x+1, \mathbf{v})). \quad (24)$$
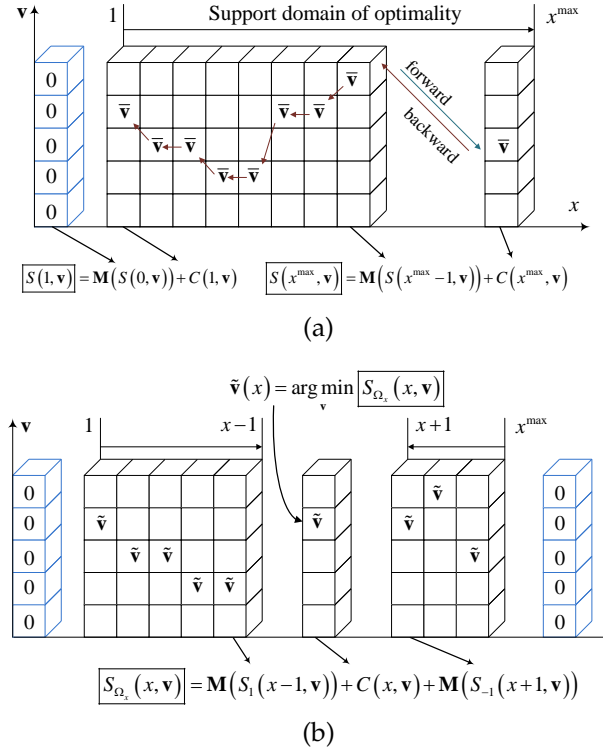

Fig. 6. The scheme of the DP agorithm, (a) - standard realization, (b) - modified.

Then, the desired solution can be obtained in each vertex x by

$$\tilde{\mathbf{v}}(x) = \arg\min_{\mathbf{v}} (S_{\Omega_x}(x, \mathbf{v})). \qquad (25)$$

It was proven in [30] that $\tilde{\mathbf{v}}$ is an exact solution, $\tilde{\mathbf{v}}(x) = \bar{\mathbf{v}}(x)$, if this optimal solution is unique and an approximation otherwise. The process of the modified DP algorithm is illustrated in Fig. 6(b).

The modification of the DP algorithm in (24) and (25) are not useful for the one-dimensional case, because the computational complexity increases twice in this case, however, the idea to divide the image domain into a set of support subdomains becomes a significant advantage for the multidimensional case.

4.2   EDP Algorithm

The main goal of the EDP algorithm is to estimate marginal function $S_{\Omega}(\mathbf{x}, \mathbf{v})$. Unfortunately, the exact estimation of the mentioned marginal function is not possible for the case of the multidimensional variable $\mathbf{x}$, like it can be done for the one-dimensional case and was described in the previous subsection. Nevertheless, an excellent approximation can be done, using the similar approach. Let $S_k(x_1, ..., x_{|k|}, ..x_D, \mathbf{v})$ be a sum of optimal costs that covers a support domain $S_k(\mathbf{x}, \mathbf{v}) \to \mathbf{x}' \in \{\forall \Omega, \text{sgn}(k) x'_{|k|} \leq \text{sgn}(k) x_{|k|}\}$ for all the $k \in K = \{\pm 1, \pm 2, ..., \pm D\}$ as illustrated in Fig. 7, where $D = 2$ and the number of types of the optimal sums S_k is 4. In the general multidimensional case, this number is $2D$, which
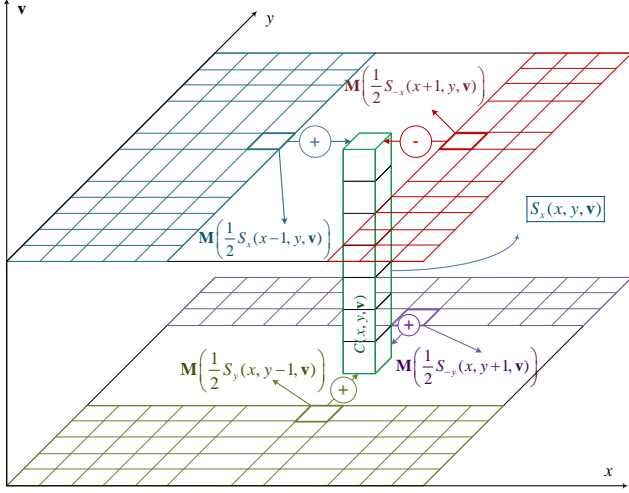
Fig. 7. Recursion of an optimal sum $S_x(x, y, \mathbf{v})$.

is twice as much as the dimensionality of the image space. We also introduce an auxiliary shortcut for the image space variable $\mathbf{x}_k = \{x_1, x_2, ..., x_{|k|} - \text{sgn}(k), ..., x_D\}$, then the marginal sum $S_{\boldsymbol{\Omega}}(\mathbf{x}, \mathbf{v})$ of the whole image space domain $\boldsymbol{\Omega}$ is getting

$$S_{\boldsymbol{\Omega}}(\mathbf{x}, \mathbf{v}) = \sum_{k \in K} \mathbf{M}\left(\frac{1}{D} S_k(\mathbf{x}_k, \mathbf{v})\right), \qquad (26)$$

where the minimum search operator $\mathbf{M}$ has the same meaning as in (9) and (10). The normalization factor $\frac{1}{D}$ in (26) indicates that the integral number of the image space vertex $\mathbf{x}$, which is covered by the $2D$ support domains of the sums $S_k(\mathbf{x}, \mathbf{v})$ in (26), is D times of the $|\boldsymbol{\Omega}|$. The estimation of the marginal function $S_{\boldsymbol{\Omega}}(\mathbf{x}, \mathbf{v})$ in (26) supposes that all k optimal sums $S_k(\mathbf{x}, \mathbf{v})$ are yet calculated. The sums $S_k(\mathbf{x}, \mathbf{v})$, in turn, can be calculated recursively

$$S_{k'}(\mathbf{x}, \mathbf{v}) = \sum_{k \in K/-k'} \mathbf{M}\left(\frac{1}{D} S_k(\mathbf{x}_k, \mathbf{v})\right) \\ - \mathbf{M}\left(\frac{1}{D} S_{-k'}(\mathbf{x}_{-k'}, \mathbf{v})\right). \qquad (27)$$

The process of this stage of the EDP algorithm is illustrated in Fig. 7, where the $S_x(x, y, \mathbf{v})$ is one of the four possible optimal sums to be calculated. The approximation of the desired solution in (1) now is

$$\tilde{\mathbf{v}}(\mathbf{x}) = \arg\min_{\mathbf{v}}(S_{\boldsymbol{\Omega}}(\mathbf{x}, \mathbf{v})). \qquad (28)$$

Note that the solution in (28) has not the same meaning than the optimal solution in (25) because in the case of the EDP the marginals are only estimated, unlike in (25) where they are exact. The equation (27) is the base formula of the EDP algorithm and the case of $S_x(\mathbf{x}, \mathbf{v})$ (or $S_1(\mathbf{x}, \mathbf{v})$) calculation in the 2D image space is illustrated in Fig. 7, where the sum S_x in the vertex (x, y) is calculated recursively using the previously calculated sums $\{S_x, S_y, S_y, S_{-y}\}$ in the vertices $\{(x - 1, y), (x, y - 1), (x + 1, y), (x, y + 1)\}$ respectively. The both formulas (26) and (27) were not strictly

derived and, in fact, are a heuristic. The main reason we chose this forms instead of something else is based on the cardinality preservation for the support domains of the sums $S_k(\mathbf{x}, \mathbf{v})$. Let $\sigma(S_k(\mathbf{x}))$ be the cardinality of the support domain of the sum $S_k(\mathbf{x})$ (for example $\sigma(S_{\boldsymbol{\Omega}}(\mathbf{x})) = N$), then

$$\sigma(S_{\boldsymbol{\Omega}}(\mathbf{x})) \simeq \frac{1}{D} \sum_{k \in K} \sigma(S_k(\mathbf{x}_k)), \qquad (29)$$

and

$$\sigma(S_{k'}(\mathbf{x})) \simeq \frac{1}{D} \sum_{k \in K/-k'} \sigma(S_k(\mathbf{x}_k)) \\ - \frac{1}{D}\sigma(S_{-k'}(\mathbf{x}_{-k'})). \qquad (30)$$

Thus, the comparison of (26) and (27) with (29) and (30) explains why we chose the mentioned heuristic.

If the calculation of the sums S_k in (27) is organized by iteration:

$$S_{k'}^{\tau+1}(\mathbf{x}, \mathbf{v}) = \sum_{k \in K/-k'} \mathbf{M}\left(\frac{1}{D} S_k^{\tau}(\mathbf{x}_k, \mathbf{v})\right) \\ - \mathbf{M}\left(\frac{1}{D} S_{-k'}^{\tau}(\mathbf{x}_{-k'}, \mathbf{v})\right). \qquad (31)$$

Then the result of the energy minimization will be the same as in the loopy belief propagation (LBP) approach [26], [27], which is reported in the evaluation paper [3]: it shows a slow convergence to 110-125% energy level of the GC expansion algorithm.

Therefore, to improve LBP results our algorithm has to inherit the recursive property of the DP approach. Thus, the EDP algorithm is essentially recursive instead of iterative. Of course, we can also use several iterations to improve the level of the energy minimization; however, in this case our algorithm never switches from τ to $\tau+1$ as it is necessary for iterative algorithm, like for instance in (31). Note that recursion and iteration are sometimes considered equivalent, but here we consider iteration as a data transformation from one instance of the whole data set to another instance. In contrast, recursion is considered as a data transformation from one node of the data set to a neighbor. Ideally, in the case of recursion, an impact of one node value reaches any other node of the data set within one iteration, like it is in the case of DP, for instance. In such a way, the EDP recursion gives advantage over pure BP iterations, where information from one node reaches another after several iterations.

Let us consider a vertex scanning procedure (with different scanning trees) of a rectangular image in $2D$. The simplest and the most popular scanning procedure consist of two embedded incremental loops: one intrinsic loop is a set of incremental steps in the x coordinate of the image plane $\text{inc}(x) = (1 \to 2 \to ... \to x \to x + 1 \to ... \to x^{\max})$ with a fixed value of the y; and another extrinsic loop is a set of incremental steps in the y coordinate of the image plane $\text{inc}(y) = (1 \to ... \to y \to ... \to y^{\max})$. The both loops can be also organized in the decrementing order, for example for the x: $\text{dec}(x) = (x^{\max} \to ... \to x \to ... \to 1)$. Finally there are four simple scanning procedures in 2D with four different pairs of starting-ending vertices:

- $P_1 = \text{inc}\,(\text{inc}\,(x)\,,y) = ((1,1) \to (x^{\max}, y^{\max}));$
- $P_2 = \text{inc}\,(\text{dec}\,(x)\,,y) = ((x^{\max},1) \to (1, y^{\max}));$
- $P_3 = \text{dec}\,(\text{inc}\,(x)\,,y) = ((1, y^{\max}) \to (x^{\max},1));$
- $P_4 = \text{dec}\,(\text{dec}\,(x)\,,y) = ((x^{\max}, y^{\max}) \to (1,1)).$

If the scanning order of the image is chosen as, for instance P_1, for the sums S_k calculation in (27), then only the $S_x = S_1$ (see Fig. 7) and the $S_y = S_2$ are calculated recursively. Thus, the main part of the EDP algorithm consists of the four passes of the sums S_k calculation:

- Calculate S_x and S_y with (27) under the scanning procedure P_1;
- Calculate S_{-x} and S_y with (27) under the scanning procedure P_2;
- Calculate S_x and S_{-y} with (27) under the scanning procedure P_3;
- Calculate S_{-x} and S_{-y} (27) under the scanning procedure P_4.

Let us unify all previously described four image scan passes of (27) into one iteration step $S_k\,(\mathbf{x}, \mathbf{v}) = \mathbb{P}\,(S_k\,(\mathbf{x}, \mathbf{v}))$. Then the summarized EDP algorithm consists of the next steps:

Algorithm 1 EDP for Energy Minimization

1: Form the DSI by calculating the cost values $C\,(\mathbf{x}, \mathbf{v})$ with (3).
2: Initialize the function $\bar{G}(\mathbf{v}(\mathbf{x}))$ in (7) for the minimum search operator $\mathbf{M}$ in (10) with experimental parameters λ, g and a chosen function f.
3: Perform j iterations of the procedure $\mathbb{P}$ with (27), where $j > 0$.
4: Calculate the marginal sums S_Ω with (26).
5: Obtain the approximation $\tilde{\mathbf{v}}$ of the desired function of the disparity map with (28).

---

5 EXPERIMENTAL RESULTS

In this section we consider two different aspects of the technique that is proposed in our paper. First, we show the advantages of using the DP approach in discrete energy minimization with comparative results obtained by our EDP algorithm and the GC expansion. Then, in the second subsection, the computational speedup that can be obtained by application of our RMS algorithm is discussed on the base of the previously presented experimental setup.

5.1 Energy Minimization

In stereo, the most successful methods (see Middlebury stereo evaluation table), use energy minimization approach, such as GC and LBP. In other words, solving the problem in (1) is the core of the contemporary state-of-art stereo matching techniques. On the other hand, to reach the top-ten results method should include a cascade of different image processing techniques , for example in [28], where energy minimization for the MAP matching

model is the core algorithm. So, authors in [28] use hierarchical belief propagation to solve energy minimization problem, and in principle this core algorithm can be replaced by our EDP approach to improve energy minimization accuracy and speedup the calculation process. We propose to compare our EDP approximation with one of the most excellent energy minimization algorithm: GC expansion that was described in [19]. There are two main reasons of our choice. Firstly, the expansion algorithm is relatively fast (among energy minimization algorithms, which can achieve the same approximation level) and allows to obtain excellent approximate solution in the energy minimization problem. Also, we use an open C++ code for the expansion algorithm (taken from the Middlebury data set) to make comparisons with our EDP approximation. However, we also provide a qualitative comparison with the energy minimization approach proposed in [32], which is reported to be the best approximation.

This subsection includes four experimental setups: two in stereo and two in motion with the 2D disparity space. Usually, stereo matching methods include additional cues in the smoothness term of (1), pre- and post-processing [1] to improve the final result. In our experiments we omit these algorithm stages, because it helps explaining the main idea of the paper more clearly.

In our experiments only two kinds of the prior in (7) with $f\,(|\mathbf{v}|) = |\mathbf{v}|^{l_1}$ are used: the case of the truncated linear $l_1 = 1$ and the case of the truncated squared $l_1 = 2$ priors. Two kinds of the cost in (3) also the case of the truncated linear $l_2 = 1$ and the case of the truncated squared $l_2 = 2$ cost calculation with $C_{\max} = (100)^{l_2}$. The parameter λ of the prior function in (7) in all our experiments is taken proportional to the mean value $\langle C\,(\mathbf{x}, \mathbf{v}) \rangle$ of the DSI cost

$$\lambda = \left\lfloor \frac{l_2 \,\langle C\,(\mathbf{x}, \mathbf{v}) \rangle}{l_1 \,|g|^{l_1}} \right\rfloor, \tag{32}$$

To make our comparison with GC expansion feasible we rounded the λ by the floor function $\lfloor\rfloor$ in (32). It means that all the values of the input prior matrix for the GC expansion are integer.

The first experimental setup is based on two stereo images: Cones from the Middlebury data set (the size of the images is 450x375). The maximum disparity in this case is 59 (or 60 labels). The value of power for the prior measure is $l_1 = 1$, which is the case of the truncated linear prior; and the value of power for the cost measure is $l_2 = 2$, which is the case of the truncated squared cost. The truncation threshold is $g = 5$. The disparity map obtained by our EDP algorithm after first iteration is shown in Fig. 8(b), where the left part of this image is the resultant disparity map and the right part is the ground truth disparity map. Fig. 8(a) represents the left and the right images of the Cones stereo pair.

Convergence of the energy value versus number of iteration is shown in Fig. 8(c). We can see that for this
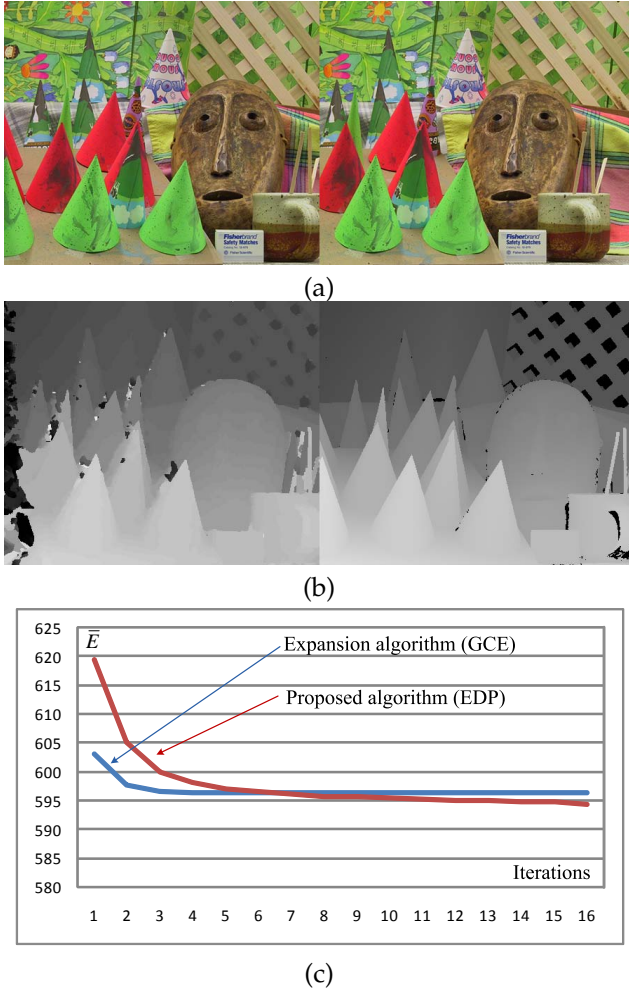
(a)

(b)

(c)

Fig. 8. The results of the first experiment, (a) the left and right images of the Cones stereo pair, (b) - the disparity map obtained by the EDP algorithm after first iteration and the ground truth, (c) the comparison diagram of energy values versus iteration for the EDP and the expansion algorithms.
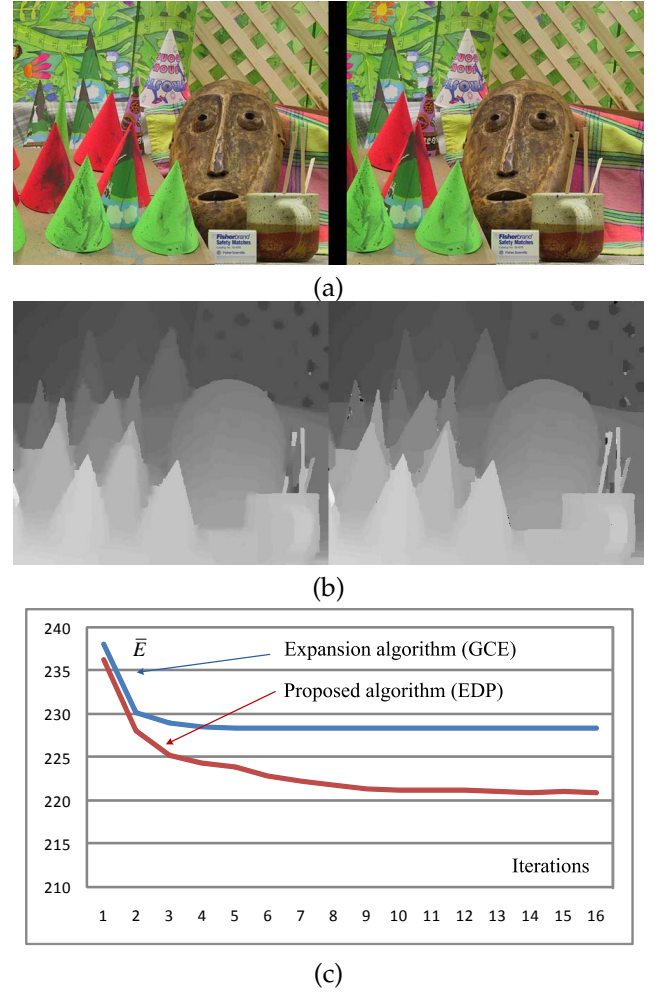

(a)

(b)

(c)

Fig. 9. The results of the second experiment, (a) the middle and the right images of the Cones stereo triplet, (b) - the resultant disparity map obtained by the EDP algorithm after 16 iterations for the truncated square and the truncated linear priors, (c) the comparison diagram of energy values versus iteration for the EDP and the expansion algorithms.

particular setup our algorithm surpasses the result of GC expansion algorithm after six iterations. Here and later on we use the per pixel energy value $\bar{E} = \frac{E}{N}$ instead of the energy value in (9), because this value is closer to the values of the mean squared and the mean absolute error criteria. For example, in the diagram in Fig. 8(c) the value of the $\bar{E}$ is 619.47 and it says that the average value of difference between corresponding color vectors in the left and in the right stereo images is not more than 25.

The second experimental setup is chosen to demonstrate the possibility of occlusion handling. Such a setup is based on three stereo images: Cones from the Middlebury data set (the size is 450x375). The maximum disparity in this case is 59 (or 60 labels). The value $l_1 = 2$, which is the case of the truncated squared prior; and the value $l_2 = 2$, which is the case of the truncated squared cost. The truncation threshold is $g = 3$. The DSI cost in

this case is a superposition of two costs: middle-image-to-right and middle-image-to-left

$$C\left(\mathbf{x}, \mathbf{v}\right) = C_{mdl \rightarrow rgh}\left(\mathbf{x}, \mathbf{v}\right) \wedge C_{mdl \rightarrow lft}\left(\mathbf{x}, -\mathbf{v}\right), \quad (33)$$

in which the $C_{mdl \rightarrow rgh}\left(\mathbf{x}, \mathbf{v}\right)$ and the $C_{mdl \rightarrow lft}\left(\mathbf{x}, -\mathbf{v}\right)$ are calculated by (3). The idea of such superposition was proposed in [36]. Note than in this case the distance between left and middle camera lenses has to be equal to the distance between right and middle camera lenses, and the axes of all the coordinate systems of the cameras have to be collinear. The resultant disparity map obtained by our EDP algorithm after 16 iterations is shown in Fig. 9(b), where the left part of this image is the disparity map obtained with the truncated squared prior and the right part is the disparity map obtained with the truncated linear prior. Fig. 9(a) represents the left and the right images of the Cones stereo triplet. We can see that in case of three stereo images the problem

of the disparity map estimation in the occluded regions is practically solved, compare Fig. 8(b) and Fig. 9(b). However, the edges of the disparity map in Fig. 9(b) are smooth due to the squared prior of the experiment setup in comparison with the right part of Fig. 9(b), where the truncated linear prior is used. Convergence of the energy value versus number of iteration is showed in Fig. 9(c). We can see that in this particular prior function our algorithm surpasses the result of GC expansion algorithm from the first iteration.

The third experimental setup is based on two motion images: Back Yard from the Middlebury data set (the size is 320x265). The maximum disparity in this case is ± 7 in the v_y disparity direction and is ± 13 in the v_x disparity direction (or 405 labels). The value $l_1 = 1$, which is the case of the truncated linear prior; and the value $l_2 = 2$, which is the case of the truncated squared cost. The truncation threshold $g = 3$. The resultant disparity map obtained by our EDP algorithm after first iteration is shown in Fig. 10(b), where Fig. 10(a) is the first and the second images of the Back Yard motion sequence. The right side of Fig. 10(b) is the hue saturation color map related to the size and orientation of the disparity vector. This map is overlapped by a histogram: the darker pixels in the color map correspond to the more frequent value of the disparity vector in the disparity map. Convergence of the energy value versus number of iteration is shown in Fig. 10(c).

The fourth experimental setup is based on three motion images: Back Yard from the Middlebury data set (the size is 320x265). The maximum disparity in this case is ± 7 in the v_x disparity direction and is ± 13 in the v_y disparity direction (or 405 labels). The value $l_1 = 2$, which is the case of the truncated squared prior; and the value $l_2 = 2$, which is the case of the truncated squared cost. The truncation threshold $g = 3$. The resultant disparity map obtained by our EDP algorithm after 16 iterations is shown in Fig. 11(b), where Fig. 11(a) is the second and the third images of the Back Yard motion sequence. The DSI cost in this particular case is calculated by (33). Here we use the assumption that the forward motion vector (in the 2D disparity space) is approximately equal to the negative backward motion vector. This assumption helps us to overcome the occlusion errors. This is understandable if we compare to resultant disparity maps in Fig. 10(b) and in Fig. 11(b). Convergence of the energy value versus number of iteration is shown in Fig. 11(c).

5.2 Computational Speedup

Let us consider more carefully the possible speedup in the case of the implementation of the proposed algorithm general RMS (GRMS) and the linear RMS (LRMS). Theoretically, the slowest straightforward search needs $O(Q)$ operations per vertex in the DSI space where Q is the number of labels. For stereo, the number of labels is $Q = v^{max}$ and for motion is $Q = (2v_x^{\max} - 1)(2v_y^{\max} - 1)$. The application of the GRMS algorithm restrict the num-
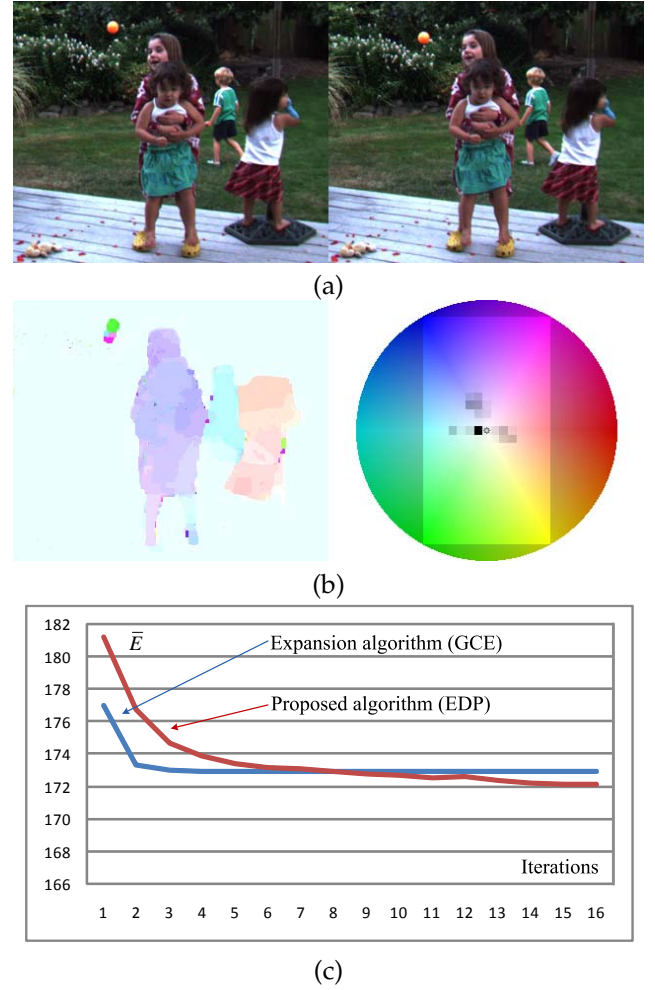

(a)

(b)

(c)

Fig. 10. The results of the third experiment, (a) the first and the second images of the Back Yard motion sequence, (b) - the resultant disparity map obtained by the EDP algorithm after the first iteration with the truncated linear prior, (c) the comparison diagram of energy values versus iteration for the EDP and the expansion algorithms.

ber of the needed operations to $R(2g - 1)$ and the application of the LRMS algorithm restrict the number of need operation to $3R$. In Tab. 1 these theoretically derived numbers are given for the four different experimental setups described in Subsection 5.1. The practical realization of the compared algorithms, which are running along, shows nearly the same relative speedup. However, the speedup in the real program realization of the algorithm is not the same; due to additional operations inside the program body (e. g. stack and cycles control operations). In Tab. 2 we give the result of the computer experiments that was done on a standard desktop PC equipped with Core Duo 3.16 GHz CPU 4 GB, where the number of running seconds is per iteration. The time needed per iteration in the case of the expansion algorithm depends on the DSI data and on the number of iterations. Thus, in Tab. 2, we report the time needed for the first iteration.

Also, our algorithm is running faster than the expan-
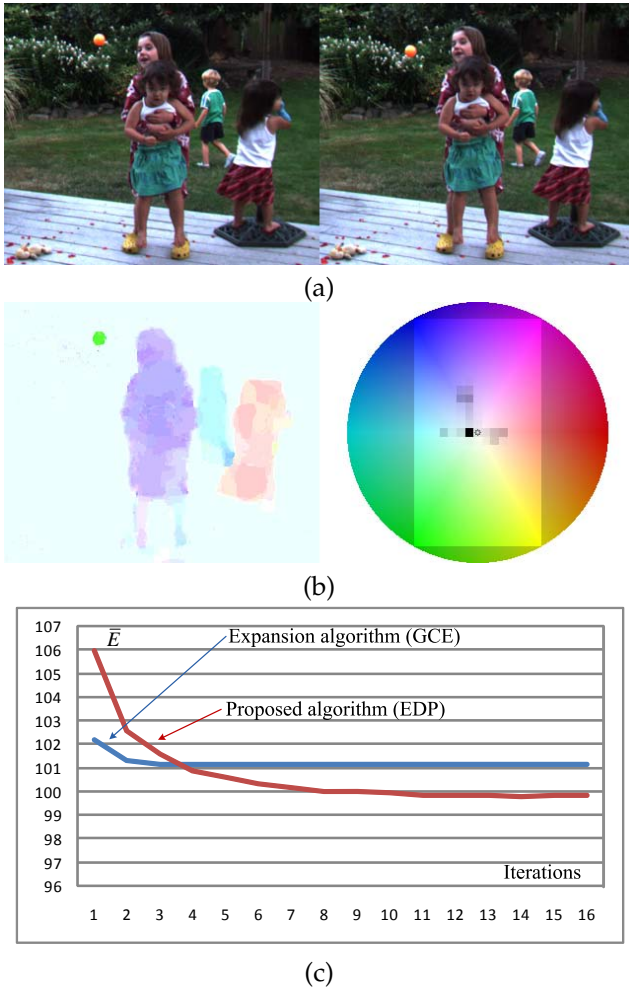
(a)

(b)

(c)

Fig. 11. The results of the third experiment, (a) the second and the third images of the Back Yard motion sequence, (b) - the resultant disparity map obtained by the EDP algorithm after 16 iterations with the truncated squared prior, (c) the comparison diagram of energy values versus iteration for the EDP and the expansion algorithms.

TABLE 1

The number of operations per vertex in the DSI space for SFMS, GRMS and LRMS.

Algorithms	Setup 1	Setup 2	Setup 3	Setup 4
SFMS	60	60	405	405
GRMS	-	9	-	10
LRMS	3	-	6	-

sion algorithm, especially in the case when the composite DSI cost calculated with (33) is used or the prior dependency is the truncated squared.

The GCE algorithm was first proposed in 2000. The graph cut community now offers new methods that report better results in terms of energy minimization. See for example approach in [32]. However, if Kumar's method in [32] is compared with GCE we can see that it is still time consuming: in general this method works 10-25 times slower than the GTE algorithm to reach almost

TABLE 2

The speedup of the GRMS and the LRMS versus the SFMS and the GCE algorithms,where the number of running seconds is per iteration.

Algorithms	Setup 1	Setup 2	Setup 3	Setup 4
SFMS	19.1 s	19.1 s	336 s	336 s
GRMS	-	3.2 s	-	16.6 s
LRMS	2.3 s	-	10.8 c	-
GCE	4.3 s	5.1 s	12.1 s	15.8 s

the same result. On the other hand, for the real stereo matching experiments, Kumar's method does not show significant improvement in terms of energy minimization in comparison with GCE, especially in the case of the truncated linear prior. Indeed, the difference of the minimum values of two these approaches is less than 0.2%. For example, the difference of the minimum values between GCE and LBP can be more than 15%. Note that such a difference (0.2%) in energy minimization has only theoretical impact for reconstruction accuracy. We would like to stress that our algorithm surpasses the result of GCE by almost the same 0.2% in the case of the truncated linear prior.

6 Conclusion

In this paper a novel algorithm that reduces the computational complexity of the straightforward search in the DP approach is presented. The proposed method has been utilized for stereo and motion problems and has shown a significant speedup. Also a new expansion of the DP technique was proposed. This algorithm showed state-of-the-art results in terms of energy minimization and can be applied directly for many other early vision problems that use the MRF approach and the regular rectangular image grid.

Acknowledgments

References

[1] D. Scharstein, and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," International Journal of Computer Vision, vol. 47, no.1-3, pp. 7-42, 2002.
[2] M.Z. Brown, D. Burschka, and G.D. Hager, "Advances in Computational Stereo," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 25, no. 8, pp. 993-1008, 2003.
[3] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A Comparative Study of Energy Minimization Methods for Markov Random Fields," Proc. European Conference on Computer Vision (ECCV 2006), vol. 2, pp. 16-29, 2006.
[4] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors," IEEE Trans. Pattern Analysis and Machine Intelligence, vol.30, no.6, pp. 1068 - 1080, 2008.
[5] G.L.Gimelfarb, V.B. Marchenko, and V.I. Rybak, "Algorithm of Automatic Matching of Identical Patches in Stereopairs," Kibernetika, vol.2, pp. 118-129, 1972.

[6] Y. Ohta and T. Kanade, "Stereo by Intra - and Intra-Scanline Search Using Dynamic Programming," IEEE Trans. Pattern Analysis and Machine Intelligence, vol.7, no.2, pp. 139-154, 1985.

[7] C. L. Zitnik and T. Kanade. "A Cooperative Algorithm for Stereo Matching and Occlusion Detection,"' IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 7, pp. 675-684, 2000.

[8] V. Veksler, "Stereo correspondence by dynamic programming on a tree," Proc. IEEE Computer Vision and Pattern Recognition (CVPR 2005), vol. 2, pp. 384 - 390, 2005.

[9] C. Sun, "Fast Optical Flow Using 3D Shortest Path Techniques," Image and Vision Computing, vol. 20(13-14), pp. 981-991, 2002.

[10] C. Sun, Y. Li, S.B. Kang, and H-Y. Shum, "Symmetric Stereo Matching for Occlusion Handling," Proc. IEEE Computer Vision and Pattern Recognition (CVPR 2005), vol. 2, pp. 2386-2393, 2005.

[11] M. Gong, and Y. Yang, "Disparity Flow Estimation using Orthogonal Reliability-based Dynamic Programming," Proc. 18th International Conference Pattern Recognition (ICPR 2006), vol. 2, pp. 70-73, 2006.

[12] H. Hirschmller, "Stereo Vision in Structured Environments by Consistent Semi-Global Matching," Proc. IEEE Computer Vision and Pattern Recognition (CVPR 2006), vol. 2, pp. 2386-2393, 2006.

[13] L. Wang, M. Liao, M. Gong, R. Yang, and D. Nister, "High-Quality Real-Time Stereo Using Adaptive Cost Aggregation and Dynamic Programming," Proc. Third International Symposium 3D Data Processing, Visualization, and Transmission, pp. 798-805, 2006.

[14] G. Zhang, J. Jia, T-T.Wong, and H. Bao. "Recovering Consistent Video Depth Maps via Bundle Optimization," Proc. IEEE Computer Vision and Pattern Recognition (CVPR 2008), vol. 2, pp. 1-8, 2008.

[15] S. Mattoccia, F. Tombari, and L. Di Stefano, "Stereo vision enabling precise border localization within a scanline optimization framework," Proc. of Asian Conference on Computer Vision (ACCV 2009), vol. 4844, pp. 517-527, Springer LNCS, 2007.

[16] J.Salmen, M. Schlipsing, J. Edelbrunner, and S. Hegemann, "Real-time stereo vision: making more out of dynamic programming," Proc. of Conference on Computer Analysis of Images and Patterns (CAIP 2009), vol. 5702, pp. 617-626, Springer LNCS, 2009.

[17] S. Roy and I. J. Cox, "A Maximum-Flow Formulation of the N-Camera stereo Correspondence Problem," Proc. International Conference Computer Vision (ICCV 1998), pp. 492-499, 1998.

[18] H. Ishikawa,"Exact optimization for Markov random fields with convex priors," IEEE Trans. Pattern Analysis and Machine Intelligence, vol.25, no.10, pp. 1333 - 1336, Oct. 2003.

[19] Y. Boykov, O. Veksler, and R. Zabih, "Fast Approximate Energy Minimization via Graph Cuts," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no. 11, pp. 1222-1239, 2001.

[20] J. Kim, V. Kolmogorov, and R. Zabih, "Visual Correspondence Using Energy Minimization and Mutual Information," Proc. IEEE International Conference on Computer Vision (ICCV 2003), pp. 1033-1040, 2003.

[21] V. Kolmogorov and R. Zabih,"What Energy Functions can be Minimized via Graph Cuts?" IEEE Trans. Pattern Analysis and Machine Intelligence, vol.26, no.2, pp. 147-159, Feb. 2004.

[22] Y. Boykov and V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," IEEE Trans. Pattern Analysis and Machine Intelligence, vol.26, no.9, pp. 1124-1137, Sept. 2004.

[23] N. Komodakis, and G. Tziritas,"Approximate Labeling via Graph Cuts Based on Linear Programming," IEEE Trans. Pattern Analysis and Machine Intelligence, vol.29, no.8, pp. 1436-1453, Aug. 2007.

[24] O. Veksler, "Graph Cut Based Optimization for MRFs with Truncated Convex Priors," Proc. IEEE Computer Vision and Pattern Recognition (CVPR 2007), vol. 2, pp. 1-8, 2007.

[25] V. Kolmogorov, "Convergent Tree-Reweighted Message Passing for Energy Minimization," IEEE Trans. Pattern Analysis and Machine Intelligence, vol.28, no.10, pp. 1568 - 1583, Oct. 2006.

[26] W. Freeman, E. Pasztor, and O. Carmichael, "Learning Low-Level Vision," Int. J. Computer Vision, vol. 40, no. 1, pp. 25-47, 2000.

[27] P. Felzenszwalb and D. Huttenlocher, "Efficient Belief Propagation for Early Vision," I. J. Computer Vision, vol. 70, no. 1, pp. 41-54, 2006.

[28] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister, "Stereo Matching with Color-Weighted Correlation, Hierarchical Belief Propagation and Occlusion Handling," Proc. IEEE Computer Vision and Pattern Recognition (CVPR 2006), vol. 2, pp. 2347-2354, 2006.

[29] M. Mozerov, V. Kober, and T.S. Choi, "Improved Motion Stereo Matching based on a Modified Dynamic Programming," Proc. IEEE Computer Vision and Pattern Recognition (CVPR 2000), vol. 2, pp. 501 - 505, 2000.

[30] M. Mozerov, "An Effective Stereo Matching Algorithm with Optimal Path Cost Aggregation," Proc. of 28th Annual Symposium of the German Association for Pattern Recognition (DAGM 2006), pp. 617-626, Springer LNCS, 2006.

[31] T. Heibela, B. Glockera, M. Grohera, N. Paragios,N. Komodakis, and N. Navaba,"Discrete Tracking of Parametrized Curves," Proc. IEEE CVPR Workshops 2009, pp.1754-1761, June 2009.

[32] M. P. Kumar and P. H. S. Torr,"Improved moves for truncated convex models," Technical report, Universityof Oxford, 2008.

[33] M. P. Kumar and P. H. S. Torr,"Improved moves for truncated convex models," In NIPS, 2008.

[34] P. Strandmark and F. Kahl, "IParallel and Distributed Graph Cuts by Dual Decomposition," Proc. IEEE Computer Vision and Pattern Recognition (CVPR 2010), pp. 8, 2010.

[35] T. Pock, M. Unger, D. Cremers, and H. Bischof, "Fast and exact solution of Total Variation models on the GPU," Proc. IEEE Computer Vision and Pattern Recognition Workshops(CVPRW 2008), pp. 1-8, 2008.

[36] M. Mozerov, J. Gonzalez, X. Roca, and J. J. Villanueva, "Trinocular stereo matching with composite disparity space image," Proc. IEEE IEEE International Conference on Image Processing (ICIP 2009), pp. 2089 - 2092, 2009.