A Locally Linear Regression Model for Boundary Preserving Regularization in Stereo Matching

Shengqi Zhu[1], Li Zhang[1], and Hailin Jin[2]

[1] University of Wisconsin - Madison
[2] Adobe Systems Incorporated
{sqzhu,lizhang}@cs.wisc.edu,
hljin@adobe.com

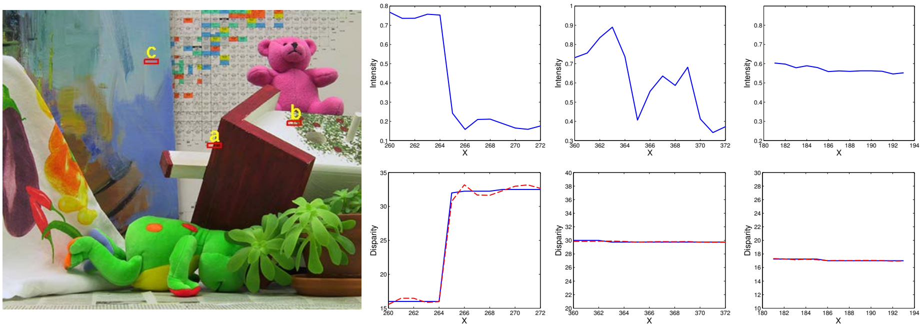
Abstract. We propose a novel regularization model for stereo matching that uses large neighborhood windows. The model is based on the observation that in a local neighborhood there exists a linear relationship between pixel values and disparities. Compared to the traditional boundary preserving regularization models that use adjacent pixels, the proposed model is robust to image noise and captures higher level interactions. We develop a globally optimized stereo matching algorithm based on this regularization model. The algorithm alternates between finding a quadratic upper bound of the relaxed energy function and solving the upper bound using iterative reweighted least squares. To reduce the chance of being trapped in local minima, we propose a progressive convex-hull filter to tighten the data cost relaxation. Our evaluation on the Middlebury datasets shows the effectiveness of our method in preserving boundary sharpness while keeping regions smooth. We also evaluate our method on a wide range of challenging real-world videos. Experimental results show that our method outperforms existing methods in temporal consistency.

1 Introduction

Boundary preserving regularization plays an important role in global stereo matching. In the seminal work [1], Bobick and Intille observed that occlusion boundaries tend to collocate with intensity edges. This observation has since been widely used in most of the top performing global stereo matching methods. The common way of using this observation is to modulate the pairwise regularization with a weight function which is inversely proportional to the pixel difference; that is, the larger the pixel difference is, the weaker the pairwise regularization is [2]. Although being simple and powerful, the pairwise boundary preserving regularization has two limitations. First, between a pair of neighboring pixels, there is no way to tell whether a change in pixel value is due to image noise or not. Second, the term cannot capture potential relationships between pixel values and disparities beyond neighboring pixels.

In this paper, we propose a novel boundary preserving regularization that overcomes these two limitations. In order to reason what is image noise, we go beyond adjacent pixels and consider a large neighborhood window. Using a large neighborhood window also allows us to capture pixel value and disparity interactions at a higher level. Our regularization model is based on the observation

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part V, LNCS 7576, pp. 101–115, 2012.
© Springer-Verlag Berlin Heidelberg 2012



Linear Regression Result

$D(x) = -28.94I(x) + 37.76$ $D(x) = 0.31I(x) + 29.62$ $D(x) = 6.45I(x) + 13.42$

(a) (b) (c)

Fig. 1. Linear regressions between intensities and disparities around three points (marked in the Teddy image) at: (a): disparity discontinuity, (b): dense texture region, and (c): textureless region. Blue solid lines represent the spatial intensity change (1st row) and the spatial disparity change (2nd row). Red dashed lines in the second row show the estimated disparity using the regression model in the equations (3rd row).

that within a local window (*e.g.*, 5×5), the disparity values in the window can be linearly regressed from the intensity values, *i.e.*,

$$D(x, y) = aI(x, y) + b. \tag{1}$$

Figure 1 intuitively illustrates this observation in three typical cases using 1D windows. In the case of Figure 1(a), the window contains a depth discontinuity, and the intensity in the window transits from one level to another. Since the intensity transition collocates with the color transition, the two parameters a and b are sufficient to map the two dominant intensity values to the two dominant disparity levels. In the case of Figure 1(b), the window contains a single disparity but varying intensities (texture), letting $a \approx 0$ and b being the average disparity again linearly maps the disparity levels from the intensity values. In the case of Figure 1(c), the window approximately contains a constant disparity and uniform intensity. This is an ambiguous situation, because a and b follow a linear constraint $\bar{D} \approx a\bar{I} + b$. However, if we add a regularization to penalize a large value of a, then $a \approx 0$ and b being the average disparity leads to a good fit. More general cases are discussed in Section 3.2.

Based on this observation, we propose a new stereo matching energy function that uses local linear regression models as regularization. In this energy function, regression parameters a and b are unknown variables that are adapted to each local window. This energy function is not readily optimized by conventional optimization methods, such as Graph Cuts and Belief Propagation. We develop an iteratively convex relaxation algorithm that minimizes the energy function.

We evaluate our method on all the stereo pairs from Middlebury dataset with ground truth. We achieve the state-of-the-art performance (ranked top 10 on the four benchmark pairs, as of the submission time on March 2012). We also evaluate our stereo method on a wide range of challenging real-world videos used in the literature. Although our stereo method is applied to each pair of frames independently (no temporal constraint is used), our method demonstrates temporally stable disparity estimates and significantly outperforms competitive frame-by-frame methods with regard to temporal consistency. By not depending on motion tracking for temporal consistency, our method is applicable to scenes with fast motion where tracking itself is a challenge.

To summarize, our technical contributions include:

- We propose a novel regularization by modeling the local correlation between disparity variations and intensity patterns. Each regularization term involves more than two pixels and we show how to formulate both first order and second order smoothness terms.
- We propose a progressive convex-hull filter that relaxes the data cost in each iteration. We demonstrate that such iterative relaxation helps to escape local minima in optimizing non-convex function in stereo.
- We propose an iterative algorithm that alternates between finding a quadratic upper bound of the relaxed energy function at the current solution and updating the solution by minimizing the upper bound using iterative re-weighted least squares.

2 Related Work

Stereo matching has a large body of literature. Most recent papers can be found and are evaluated on the Middlebury website [3]. Reviewing all the papers is beyond the scope of this paper. We only discuss closely related work.

Our method falls into the category of global methods which formulate stereo matching using a random field model and infer disparity maps using energy minimization methods like Graph Cuts [2] or Belief Propagation [4]. Most global methods are designed to handle energy functions with pairwise terms. High-order terms are usually approximated as pairwise terms for optimization [5, 6]. In addition, image segmentation is often combined with GC and BP based stereo matching to achieve better results [7–10]. However, when applied to video in a frame-by-frame fashion, segmentation based methods may generate artifacts that are temporally inconsistent [5].

The closest previous work to ours is Bhusnurath and Taylor [11], in which they proposed a convex formulation for binocular stereo matching. Their key idea is to find a piecewise linear lower convex hull of each data term and use a weighted L_1 penalty for the first and second order finite differences as regularization. Our work differs from [11] in three major ways. First, our regularization models the local correlation between disparity variation and intensity pattern while the previous work does not. Second, we use a progressive convex hull filter to iteratively tighten the relaxation used in their data term. Third, their energy function is non-smooth, requiring a linear programming solver that introduces a large number of slack variables (to handle inequality constraints); our energy function is smooth, and we designed an iterative re-weighted least squares (IRLS) technique to search for minimum. We demonstrate that our method produces better boundary-preserving results on the Middlebury benchmarks and better temporal consistency on many videos.

Our regularization model is inspired by Rhemann *et al.* [12], where they utilize a local linear model for edge-preserving cost aggregation. Previous works have also successfully applied local linear model in other applications, such as image matting [13] and image filtering [14]. Our method differs from [12] in that ours is an energy-based method that incorporates the local linear model into our minimization objective, while local methods like [12] use linear model as filter for the cost volume.

3 Problem Formulation

We formulate stereo matching as a global optimization. Given a left-right pair I_l and I_r, we seek to estimate the disparity map $\mathbf{D}$ between them. Our formulation is not symmetric: the disparity map is from the left (reference) to the right image. The objective function Φ is a sum of three terms: data terms Φ_0, first order smoothness terms Φ_1, and second order smoothness terms Φ_2:

$$\Phi(\mathbf{D}) = \Phi_0(\mathbf{D}) + \beta_1\Phi_1(\mathbf{D}) + \beta_2\Phi_2(\mathbf{D}), \tag{2}$$

where β_1 and β_2 are the combination coefficients.

3.1 Data Term Φ_0

$\Phi_0(\cdot)$ is a weighted sum of data terms over all pixels as

$$\Phi_0(\mathbf{D}) = \sum_{i\in\mathcal{I}} u_i\gamma_i(d_i), \tag{3}$$

where d_i is the disparity for pixel i, γ_i is the data term function, u_i is the confidence weight for pixel i, and $\mathcal{I}$ is the set of all pixels. Data term is not the contribution of this paper; we used Guided Filtered cost volume [12]. Although the cost volume is defined only on integer disparity levels, we interpolate these discrete values so that the data term function is defined continuously from 0 to the maximum disparity. The calculation of u_i will be described in Section 5.1.

3.2 First Order Smoothness Term Φ_1

Following our observation made in Section 1, we model the linear regression from color intensity to disparity values as follows. For pixel j in the neighborhood of i, the disparity d_j is regressed from its own intensity value I_j as:

$$d_j \approx a_i I_j + o_i, \forall j \in \mathrm{Nbr}(i), \tag{4}$$

where a_i and o_i are the unknown coefficients.

As shown in Figure 1, if the disparity is flat around i, then a_i is close to 0 and o_i is the average disparity within the neighborhood. If there is a sharp transition in the disparity in this neighborhood, *e.g.*, a step edge, then most likely there will be a sharp intensity transition collocated in the same neighborhood, and the coefficients a_i and o_i will transform the intensity profile into the disparity profile. Therefore, this model is valid for images with piecewise smooth intensity patterns where the disparity discontinuity boundaries collocate with the intensity edges.

To set up the regularization for the whole disparity map $\mathbf{D}$, we could, as in Levin *et al.* [13], sum up the sum of squared residual errors for each $[a_i, o_j]$ in all local neighborhoods as

$$\sum_i \sum_{j \in \mathrm{Nbr}(i)} (d_j - a_i I_j - o_i)^2 + \lambda a_i^2, \tag{5}$$

where λa_i^2 is the regularization term. However, using Eq. (5) for disparity regularization has a problem. Specifically, if a local neighborhood contains more than two dominant disparity values, it will likely contain more than two dominant intensity values. As a result, the affine model in Eq. (4) may not be a good fit between the intensity and the disparity. Furthermore, the squared penalty in Eq. (5) is susceptible to outliers that cause misfit. To make the regularization robust, we propose the following form

$$\sum_i \sum_{j \in \mathrm{Nbr}(i)} w_{ij} \rho_\sigma(d_j - a_i I_j - o_i) + \lambda a_i^2, \tag{6}$$

where w_{ij} is the color similarity weight between pixels i and j and ρ_σ is the Huber function with parameter σ.

$$\rho_\sigma(\epsilon) = \begin{cases} \epsilon^2/(2\sigma) & \texttt{for } |\epsilon| \leq \sigma, \\ |\epsilon| - (\sigma/2), & \texttt{otherwise}. \end{cases} \tag{7}$$

Using Eq. (6) as regularization has two benefits. First, the Huber function reduces the effect of large residuals due to model misfit. Second, w_{ij} assigns higher weights to pixels that are more similar in color to pixel i. If pixel i's neighborhood patch has more than two (*e.g.*, three) dominant disparity values, the patch will contain approximately three dominant colors, one for each disparity value, assuming that pixels with similar disparity values have similar colors. w_{ij} will favor two groups of pixels in the regression whose colors are more similar to pixel i's (one group will contain pixel i). Since a_i and o_i can map two dominant colors to two dominant disparities, the regularization model holds.

Compared to the conventional pairwise regularization using color similarity weight, such as $w_{ij}(d_i - d_j)^2$ or $w_{ij}|d_i - d_j|$, our new regularization is more robust. For example, when w_{ij} takes the form that is used in the bilateral filter [15], the conventional regularization weights depend heavily on the color bandwidth parameter, because large parameters may cause over smoothing of object boundaries and small parameters may cause isolated erroneous disparity regions. In our model, w_{ij} is used to pick up the first one or two dominant colors, and therefore it does not need to be too small to preserve sharp boundaries. The boundary sharpness is encouraged by the linear regression between intensity and disparity. The exact weight setup is described later in Section 5.

To simplify notation, let $\mathbf{a}_i = [a_i; o_i]$, $\mathbf{f}_j = [I_j; 1]$, and $\Lambda_i = \texttt{diag}([\lambda_i; 0])$. Further letting $\mathbf{D} = \{d_i\}$ be the vector of all disparities and $\mathbf{A} = \{\mathbf{a}_i\}$ be that of all coefficients, we then write the regularization as

$$\Phi_1(\mathbf{D}, \mathbf{A}) = \sum_i \sum_{j \in \mathrm{Nbr}(i)} w_{ij} \rho_\sigma(d_j - \mathbf{a}_i^\mathrm{T} \mathbf{f}_j) + \mathbf{a}_i^\mathrm{T} \Lambda \mathbf{a}_i. \tag{8}$$

Note that ρ_σ is a convex function that operates on a linear function of unknown variables $\mathbf{D}$ and $\mathbf{A}$, so $\Phi_1(\mathbf{D}, \mathbf{A})$ is convex. Further note that, the bias variables $\{o_i\}$ in the linear regression model is not regularized in Eq. (6) and Eq. (8); therefore, even for completely dark image regions, *i.e.*, $I_j = 0$, our regularization model is still effective.

3.3 Second Order Smoothness Term Φ_2

The regularization in Eq. (8) encourages pixels with similar colors to have similar disparities, which can be considered as the first order smoothness. We can use the same idea to encourage pixels with similar colors to have similar disparity derivatives, *i.e.*, the second order smoothness.

Specifically, let δ_x and δ_y be pixel index increments for spatial neighbors in the x and y directions, respectively. We approximate the disparity derivatives in x and y directions using finite differences as $d_j - d_{j+\delta_\mathrm{x}}$ and $d_j - d_{j+\delta_\mathrm{y}}$. Replacing the disparity values $\{d_i\}$ in Eq. (8) by the disparity finite differences, our second order smoothness $\Phi_2(\cdot)$ is defined as

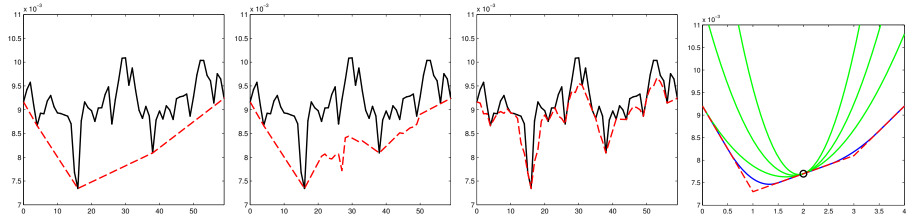
$$\begin{aligned} \Phi_2(\mathbf{D}, \mathbf{B}, \mathbf{C}) = &\sum_i \sum_{j \in \mathrm{Nbr}(i)} w_{ij}^\mathrm{b} \rho_{\sigma'}(d_j - d_{j+\delta_\mathrm{x}} - \mathbf{b}_i^\mathrm{T} \mathbf{f}_j) + \mathbf{b}_i^\mathrm{T} \Lambda^\mathrm{b} \mathbf{b}_i \\ + &\sum_i \sum_{j \in \mathrm{Nbr}(i)} w_{ij}^\mathrm{c} \rho_{\sigma'}(d_j - d_{j+\delta_\mathrm{y}} - \mathbf{c}_i^\mathrm{T} \mathbf{f}_j) + \mathbf{c}_i^\mathrm{T} \Lambda^\mathrm{c} \mathbf{c}_i, \end{aligned} \tag{9}$$

where w_{ij}^b, w_{ij}^c, $\mathbf{b}_i$, and $\mathbf{c}_i$ are counterparts of w_{ij} and $\mathbf{a}_i$ in Eq. (8); $\mathbf{B}$ and $\mathbf{C}$ are vectors of all coefficients $\{\mathbf{b}_i\}$ $\{\mathbf{c}_i\}$, respectively. The setup of w_{ij}^b and w_{ij}^c are different from w_{ij} and will be described later in Section 5.

4 Optimization Algorithm

We present an iterative algorithm to minimize our model in Eq. (2). Our objective function is difficult to optimize using discrete search methods such as Graph Cuts [2] or Belief Propagation [4], because it involves both disparity variables and auxiliary variables—the number of discretized states is huge for each pixel. Therefore we resort to a continuous optimization approach.

Our model in Eq. (2) is also challenging for continuous optimization because the data term energy Φ_0 is highly non-convex, although Φ_1 and Φ_2 are convex. Our strategy is to iteratively relax Φ_0 to be a locally convex function so that the chance of being trapped in a local minimum is reduced. The initial relaxation is loose, which transforms Φ_0 into a global convex function. During the iteration, the relaxation becomes tighter and tighter. In the last few iterations, the algorithm optimizes the original cost function. We discuss our iterative relaxation procedure and how to optimize each relaxed energy function in this section.



(a) $r = 60$ (b) $r = 11$ (c) $r = 2$ (d) Upper bound

Fig. 2. Illustration of the progressive convex hull filtering with three different radius. Initially a large r is used to compute the lower convex hull in (a). As iteration continues, r becomes smaller so that the relaxation of the data term becomes tighter ((b) and (c)). (d) shows an exaggerated quadratic B-spline (solid blue) and three parabola upper bounds (solid green) for the relaxed data term like (a). We search for the upper bound with the lowest curvature to approximate the relaxed data term in our optimization.

4.1 Progressive Convex Hull Filtering

At the first iteration, we compute the lower convex hull of each date term $\gamma_i(\cdot)$ function, as shown in Figure 2(a), and we use the hull as the relaxed energy function. This relaxation turns the original problem into a convex optimization problem. We use its optimum solution to initialize the search of a less relaxed energy function in the next iteration.

To make the relaxation tighter, we reduce the range over which the lower convex hull is computed. Specifically, for each disparity level d, we consider a range $[d - r : d + r]$ of radius r. We compute the lower convex hull of $\gamma_i(\cdot)$ over this reduced range and define the value of the relaxed function at d to be the value of this local lower convex hull at d. We sequentially evaluate the relaxed function value at each d to obtain the relaxed function for this iteration.

Note that in the first iteration, r is set to be the maximum disparity, so the relaxation is the global lower convex hull. r reduces as the iteration continues. In the last few iterations, $r = 0$ and the relaxation becomes the original data term function. Figure 2 shows a few examples of relaxation with different r, indicating that a bigger r eliminates more local minimum at the expense of a poor approximate of the original energy function.

4.2 Optimizing Relaxed Objective Function

Although the relaxed Φ_0 has less number of local minima, it is still difficult to minimize together with Φ_1 and Φ_2, because the relaxed Φ_0 is non-smooth (piecewise linear, as shown in Figure 2) and the optimization problem involves a large number of variables: disparity values and regression coefficients defined for every pixel. Methods that handle non-smooth functions using sub-gradients would be prohibitively slow. Instead, we choose to approximate the relaxed Φ_0 by smoothing its corners and use an EM-like method to optimize the approximated relaxation. Specifically we iteratively find an exact quadratic upper bound and update the solution by minimizing the upper bound.

Since the regularization $\varPhi_1$ and $\varPhi_2$ are defined using Huber function, which is smooth and convex, no relaxation and smoothing approximation is needed. Only quadratic upper bound is needed, which is shown below.

$\bar{\boldsymbol{\varPhi}}_0$: Upper Bound for Relaxed Data Term $\boldsymbol{\varPhi}_0$. Our main idea is illustrated in Figure 2(d). Specifically, we approximate the relaxed data term using quadratic B-spline, as shown in solid blue line. This approximation is equivalent to convolve the relaxed data term with a box filter of support $[-0.5, 0.5]$, so the approximation error is negligible. (The B-spline in Figure 2(d) is exaggerated.) Then at the current solution (black circle in Figure 2(d)), we find a parabola with minimum curvature that is both tangent to the spline and above the whole spline, as shown in solid green lines. This lowest parabola serves as a tight upper bound of the relaxed data term. The details of how to find this parabola is provided in the supplementary material. Its computational cost is linear to the number of disparity levels.

In summary, given the current disparity map $\mathbf{D}^*$, the parabola upper bound for the data terms summed over all pixels can be written as a quadric form as

$$\bar{\varPhi}_0(\mathbf{D}) = (\mathbf{D} - \mathbf{D}^*)^{\mathrm{T}} \mathbf{H_0} (\mathbf{D} - \mathbf{D}^*) + (\mathbf{D} - \mathbf{D}^*)^{\mathrm{T}} \varPhi_0'(\mathbf{D}^*), \tag{10}$$

where $\mathbf{H_0}$ is a diagonal matrix whose diagonal elements correspond to the curvature of each parabola; $\varPhi_0'(\mathbf{D}^*)$ is a vector whose elements correspond to the tangent of the parabola. A constant is dropped in Eq. (10) because it does not alter the optimization result.

$\bar{\boldsymbol{\varPhi}}_1$: Upper bound for the First Order Term $\boldsymbol{\varPhi}_1$. Given the current estimates of $\mathbf{D}$ and $\mathbf{A}$, we can construct an exact upper bound for the regularization $\varPhi_1(\cdot)$ in Eq. (8) by finding the exact upper bound of the Huber function at the current estimates.

Precisely, we first compute the absolute value of the current residual as

$$\epsilon^*_{ij} = |d^*_j - \mathbf{a}^{*\mathrm{T}}_i \mathbf{f}_j|, \tag{11}$$

where d^*_j and $\mathbf{a}^*_i$ are the values of the current estimates. At this ϵ^*_{ij}, the Huber function $\rho(\cdot)$ in Eq. (7) has an exact upper bound $\bar{\rho}_\sigma(\cdot)$ as

$$\bar{\rho}_\sigma(\epsilon) = \begin{cases} \epsilon^2/(2\sigma) & \texttt{for } \epsilon^*_{ij} \le \sigma, \\ \epsilon^2/(2\epsilon^*_{ij}) + (\epsilon^*_{ij} - \sigma)/2 & \texttt{otherwise.} \end{cases} \tag{12}$$

Applying Eq. (12) to each term in Eq. (8), we obtain an exact upper bound $\bar{\varPhi}_1(\cdot)$ for $\varPhi_1(\cdot)$ as

$$\bar{\varPhi}_1(\mathbf{D}, \mathbf{A}) = \sum_i \sum_j v_{ij} (d_j - \mathbf{a}_i^{\mathrm{T}} \mathbf{f}_j)^2 + \mathbf{a}_i^{\mathrm{T}} \varLambda \mathbf{a}_i, \tag{13}$$

where $v_{ij} = \frac{w_{ij}}{2\max(\sigma, \epsilon^*_{ij})}$ and the constant $\sum\limits_{ij} \frac{w_{ij} \max(\epsilon^*_{ij} - \sigma, 0)}{2}$ is dropped in Eq. (13) since it does not alter the result.

$\bar{\varPhi}_1(\cdot)$ is a quadratic regularization that involves $\mathbf{D}$ and $\mathbf{A}$. If $v_{ij} = 1$, Eq. (13) has the same form as the regularization in [13]. To simplify the optimization, we can adopt the technique in [13] to eliminate $\mathbf{A}$ as intermediate variables. (Details are given in our supplementary material.) The result is

$$\bar{\Phi}_1(\mathbf{D}) = \mathbf{D}^{\mathtt{T}}\mathbf{H_1}\mathbf{D}, \tag{14}$$

where

$$\begin{aligned}
\mathbf{H_1} &= \mathtt{diag}(v_j) - \sum_i \mathbf{G}_i^{\mathtt{T}}\mathbf{F}_i^{-1}\mathbf{G}_i \\
v_j &= \sum_i v_{ij} \\
\mathbf{F}_i &= \Lambda + \sum_j v_{ij}\mathbf{f}_i\mathbf{f}_i^{\mathtt{T}} \\
\mathbf{G}_i &= [v_{i1}\mathbf{f}_1, v_{i2}\mathbf{f}_2, \cdots, v_{iN}\mathbf{f}_N].
\end{aligned} \tag{15}$$

Eq. (14) is a positive definite regularizer for the disparity map $\mathbf{D}$ that does not involve $\mathbf{A}$.

$\bar{\Phi}_2$: Upper bound for the Second Order Term Φ_2. Since $\Phi_2(\cdot)$ in Eq. (9) has the same form as Φ_1 in Eq. (8), except that it is operated on the finite difference of the disparity map rather than the disparity values, we can follow the same procedure as in Section 4.2 to obtain the quadratic upper bound for $\Phi_2(\cdot)$ as

$$\bar{\Phi}_2(\mathbf{D}) = \mathbf{D}^{\mathtt{T}}\mathbf{H_2}\mathbf{D}, \tag{16}$$

where

$$\mathbf{H_2} = \nabla_{\mathtt{x}}^{\mathtt{T}}\left(\mathtt{diag}(v_j^{\mathtt{b}}) - \sum_i \mathbf{G}_i^{\mathtt{b}^{\mathtt{T}}}\mathbf{F}_i^{\mathtt{b}^{-1}}\mathbf{G}_i^{\mathtt{b}}\right)\nabla_{\mathtt{x}} + \nabla_{\mathtt{y}}^{\mathtt{T}}\left(\mathtt{diag}(v_j^{\mathtt{c}}) - \sum_i \mathbf{G}_i^{\mathtt{c}^{\mathtt{T}}}\mathbf{F}_i^{\mathtt{c}^{-1}}\mathbf{G}_i^{\mathtt{c}}\right)\nabla_{\mathtt{y}}, \tag{17}$$

and $\nabla_{\mathtt{x}}$ and $\nabla_{\mathtt{y}}$ are sparse matrix operators that transform the disparity map $\mathbf{D}$ into its derivative maps in x and y directions, respectively; $v_j^{\mathtt{b}}$, $\mathbf{F}_i^{\mathtt{b}}$, and $\mathbf{G}_i^{\mathtt{b}}$ are computed using Eqs.(11,13,15) with w_{ij} replaced by $w_{ij}^{\mathtt{b}}$; $v_j^{\mathtt{c}}$, $\mathbf{F}_i^{\mathtt{c}}$, and $\mathbf{G}_i^{\mathtt{c}}$ are computed in the same way but with w_{ij} replaced by $w_{ij}^{\mathtt{c}}$.

4.3 Algorithm Summary

Our algorithm is summarized below.

Step0 Initialization
Set $\mathbf{D}$ using a local method and $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ as 0; set r as max disparity.
Step1 Find the relaxed data term using convex hull filter with radius r.
Step2 Evaluate upper bounds
$\bar{\Phi}_0(\mathbf{D})$, $\bar{\Phi}_1(\mathbf{D})$, and $\bar{\Phi}_2(\mathbf{D})$ in Eqs. (10,14,16).
Step3 Update disparity map
$\mathbf{D} = (\mathbf{H_0} + \beta_1\mathbf{H_1} + \beta_2\mathbf{H_2})^{-1}(\mathbf{H_0}\mathbf{D}^* - \Phi_0'(\mathbf{D}^*))$.
Step4 reduce r and return to **Step1**

In our implementation, we alternate between Step 1-4 for 30 iterations. It takes around 30 minutes to calculate the disparity map of a 450x375 image, *e.g.*, Cones, based on our Matlab implementation. The majority of the time is spent on using Matlab `backslash` to solve a sparse linear system; constructing the upper bounds and setting up the sparse matrices takes much less time. We have found the

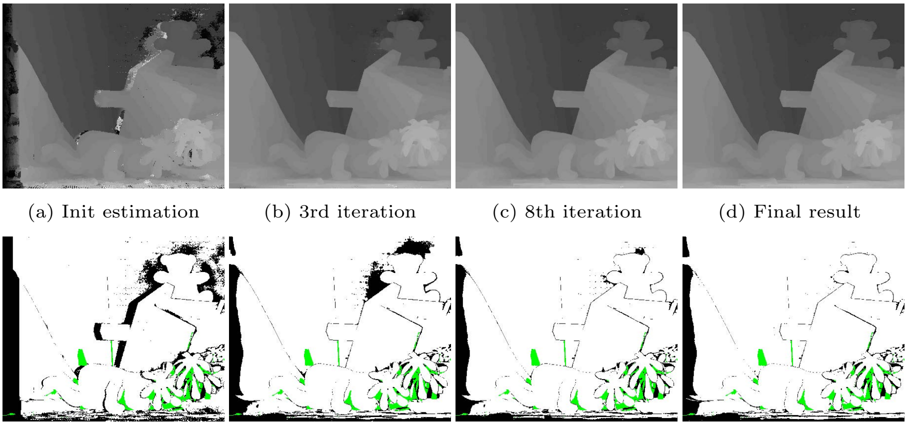

(a) Init estimation (b) 3rd iteration (c) 8th iteration (d) Final result

Fig. 3. Disparity maps (first row) and error maps (second row) of the Teddy dataset after a number of iterations. The initial estimation (a) is incorrect in (i) textureless region (right part of the wall), (ii) occluded regions (left of the roof), and (iii) regions with repetitive structures (left part of the wall). The first two types of error can be corrected in the first few iterations in (b). Our progressive convex hull filter gradually tightens the relaxation and provides more discriminative power to correct the third type of error in (c). (d) shows the final converged result of our method.

running time can be reduced by adopting an over-segmentation representation, but the results in this paper are based on our non-segmented approach. The speed can be further improved by using conjugate gradient method on a GPU, in which $\mathbf{H_0}$, $\mathbf{H_1}$, and $\mathbf{H_2}$ do not need to be constructed explicitly.

5 Implementation Details

5.1 Computing Weight $\{u_i\}$ in Eq. (3)

We use the coefficient u_i to exclude overly-textureless pixels and overly-repetitive pixels, so that they have minimum effect on the objective function. Many choices are available as reviewed in [16]. We find the following procedure works well in our implementation. For each pixel, we calculate two features: *distinctiveness* and *uniqueness*. Let $\hat{d}$ be the disparity corresponding to the minimum cost. Then distinctiveness is defined as the minimum difference of the cost value between $\hat{d}$ and $\hat{d} \pm 1$. Uniqueness is defined as the difference between the first and the second minimum cost value. We rank all the pixels by these two features, then map these two rankings linearly to $[0, 1]$, and finally set u_i to be 1 if the product of the two mapping results is above 0.1, and 0 otherwise.

5.2 Computing Weights $\{w_{ij}, w^{\mathsf{b}}_{ij}, w^{\mathsf{c}}_{ij}\}$ in Eqs. (8,9)

The pixel-pair weights w_{ij}, w^{b}_{ij} and w^{c}_{ij} should capture the likelihood of the corresponding pixels belonging to the same object. We calculate the weights based on color similarity, as: $w_{ij} = \exp\left(-\frac{\delta(i,j)}{\sigma_c}\right)$, $w^{\mathsf{b}}_{ij} = \exp\left(-\frac{\delta(i,j)+\delta(i+\delta_{\mathrm{x}},j+\delta_{\mathrm{x}})}{2\sigma_c}\right)$, and $w^{\mathsf{c}}_{ij} = \exp\left(-\frac{\delta(i,j)+\delta(i+\delta_{\mathrm{y}},j+\delta_{\mathrm{y}})}{2\sigma_c}\right)$, where $\delta(i,j)$ is the average absolute difference of RGB channels and the color bandwidth parameter $\sigma_c = 3$. For each pixel i, these weights are evaluated on its immediate 8 spatial neighbors and the most similar 8 neighbors in a search range of 31 pixels, where the similarity is measured by the distance between the two 5x5 patches that cover pixel i and j.

6 Experimental Results

We evaluate our method on the Middlebury datasets and a variety of real-world videos. A subset of the results are reported here; more are provided in the supplementary document and video.

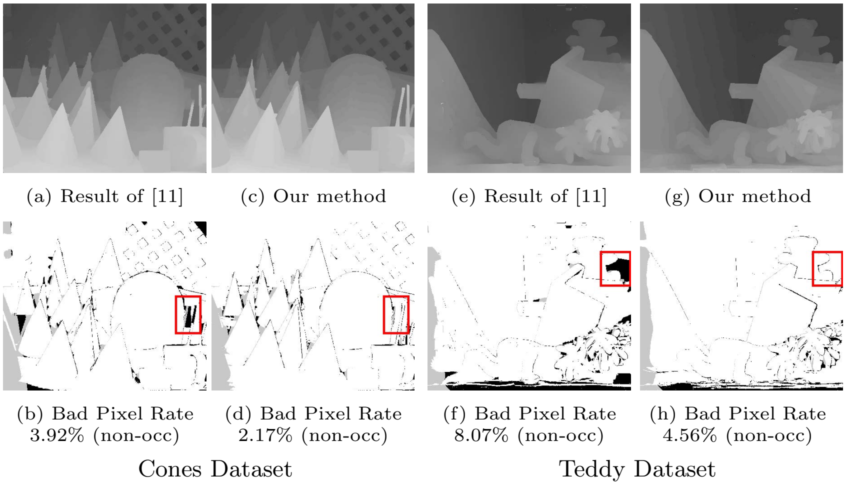
6.1 Effectiveness of Iterative Optimization

We use Teddy image dataset as an example to show the effectiveness of our iterative optimization algorithm. Figure 3 shows our intermediate disparity estimations of the left view after a number of iterations. The initial estimation (Figure 3(a)) is largely incorrect in (i) textureless regions (*e.g.*, right part of the wall), (ii) occluded regions (*e.g.*, left part of the roof), and (iii) regions with repetitive structures (*e.g.*, left part of the wall). After the first few iterations (Figure 3(b)), the occluded regions and textureless areas are corrected by the convex hull relaxation of the data term and our large-neighborhood smoothness term. But the convex hull relaxation lacks sufficient discriminative power to produce the correct disparities for repetitive structures. As iteration continues (Figure 3(c)), the progressive convex filter gradually tightens the amount of relaxation and the disparity estimates become closer to ground truth. Figure 3(d) shows the final converged result of our method.

6.2 Justification for Our Regression Based Regularization Model

Since [11] also uses a convex relaxation of data terms with pairwise regularization for stereo matching, we compared our method with [11] to justify our regularization model in Section 3, as opposed to the conventional weighted L_1 penalties: $w_{ij}|d_i - d_j|$ and its second order version, which are used in [11].

We first compare the disparity maps of these two methods on Teddy and Cones dataset. Figure 4 suggests both of these two methods can provide accurate results in most of the area. Our method preserves sharp object boundaries (see the sticks in Figure 4(b) and (d)) while keeping disparity smooth within other regions (see the wall area in Figure 4(f) and (h)) using the same parameters for these images. In this figure, Cones and Teddy results of [11] are downloaded from the Middlebury evaluation page.



(a) Result of [11]

(c) Our method

(e) Result of [11]

(g) Our method

(b) Bad Pixel Rate 3.92% (non-occ)

(d) Bad Pixel Rate 2.17% (non-occ)

(f) Bad Pixel Rate 8.07% (non-occ)

(h) Bad Pixel Rate 4.56% (non-occ)

Cones Dataset

Teddy Dataset

Fig. 4. Comparison between our method and [11] on Cones and Teddy dataset. Both of these two methods can provide accurate results in most of the area. Our method preserves sharp object boundaries ((b) and (d)) while keeping disparity smooth within background region ((f) and (h)).

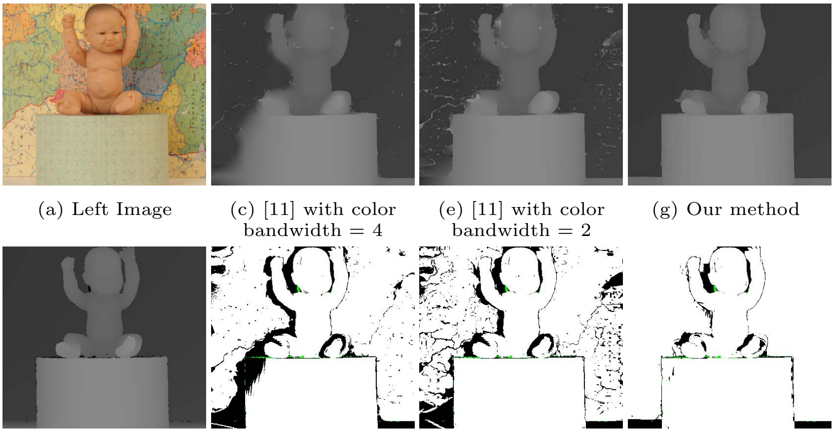
We further show that it is very difficult for [11] to set parameter to keep both sharp boundaries and smooth regions. We implemented [11] in Matlab and compared it with our method on Baby1 dataset. As Figure 5 shows, it is difficult to set the weights for the L_1 penalty that work well for all the pixels in an image: using a large color bandwidth parameter leads to fuzzy weights that cause over-smoothing across boundaries (Figure 5(c) and (d)), while using a small color bandwidth leads to sharp weights that cause disparity discontinuity in textured area (Figure 5(e) and (f)). Our regularization is much less sensitive to the color bandwidth parameter setting (Figure 5(g) and (h)). More comparisons are available in the supplementary material.

6.3 Evaluation on the Middlebury Benchmarks

We have evaluated our method on the Middlebury benchmarks [3]. We set $\beta_1 = \beta_2 = 2.5 \times 10^{-4}$, $\lambda = 10^{-3}$. Table 1 shows the ranking and percentage of bad pixels (error threshold = 1) of our method as well as two other

Table 1. Evaluation of our method on the Middlebury benchmarks

Algorithm	Rank	Tsukuba			Venus			Teddy			Cones			Avg.
		nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	
ADCensus[17]	1	1.07 (15)	1.48 (11)	5.73 (17)	0.09 (2)	0.25 (7)	1.15 (3)	4.10 (5)	6.22 (3)	10.9 (5)	2.42 (6)	7.25 (5)	6.95 (7)	3.97
Our method	10	1.05 (12)	1.65 (19)	5.64 (13)	0.29 (35)	0.81 (45)	3.07 (37)	4.56 (9)	9.81 (18)	12.2 (9)	**2.17** **(1)**	8.02 (15)	6.42 (2)	4.64
InteriorPtLP[11]	56	1.27 (29)	1.62 (15)	6.82 (37)	1.15 (77)	1.67 (72)	12.7 (87)	8.07 (65)	11.9 (36)	18.7 (70)	3.92 (54)	9.68 (49)	9.62 (45)	7.26



(a) Left Image

(c) [11] with color bandwidth = 4

(e) [11] with color bandwidth = 2

(g) Our method

(b) Ground Truth

(d) Bad Pixel Rate 12.59%

(f) Bad Pixel Rate 12.40%

(h) Bad Pixel Rate 4.92%

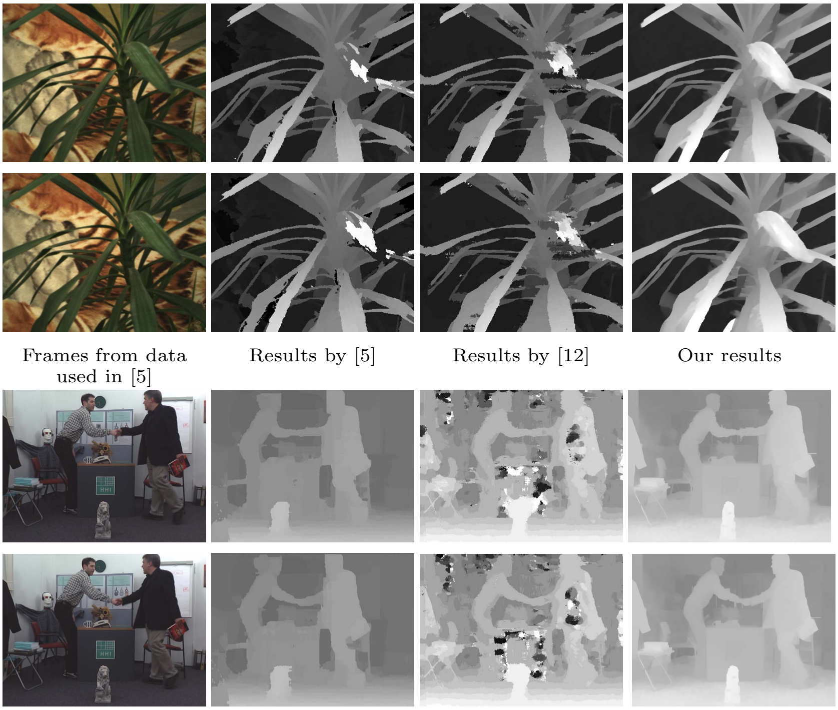
Fig. 5. Comparison between our method and [11] on *Baby1* dataset. The weighted L_1 penalty used in [11] makes its results sensitively depend on the color bandwidth parameter. A large color bandwidth yields over-smoothing across discontinuity boundaries ((c) and (d)) and a small color bandwidth results in many isolated areas in the background due to texture ((e) and (f)). Note that even with such a small bandwidth parameter, there are still over-smoothed regions around the contour of the baby and its base, which suggests it is very difficult to find a parameter that is suitable for all the pixels. Our result ((g) and (h)) preserves the sharp boundary and smooth background.

methods: the current top 1 performer, ADCensus [17], and InteriorPtLP [11], which are compared visually in Sections 6.4 and 6.2 respectively. Our method is ranked 10th and achieves the best non-occluded disparity accuracy in the Cones dataset.

6.4 Temporal Consistency on Real-World Video Data

We have tested our method on publicly available, real-world videos used in previous papers. In particular, we compared with ADCensus [9], the current top 1 performer on the Middlebury website, and Smith et al [5], a frame-by-frame method shown to have better temporal consistency than classic stereo methods. Visual comparison for two consecutive frames is shown in Figure 6. See the supplementary video for better visual comparison of temporal consistency.

On the *Plant* data used in [5], our method produces more consistent results than both [5] and [12]. Note that in [5], the result on this dataset is much better because it is obtained using 5 images as input. Here we only use two of the 5 images as input for fair comparison. Similarly, on the *Book Arrival* data used in [17], our method shows dramatic improvement over [17] and [12], in terms of preserving consistent boundaries across frames. More video examples are provided in the supplementary video.



Frames from data used in [5]

Results by [5]

Results by [12]

Our results

Frames from data used in [17]

Results by [17]

Results by [12]

Our results

Fig. 6. Evaluation on stereo videos used in previous literatures. Each two rows are consecutive frames from real-world stereo videos. Our method noticeably outperforms all the other methods in temporal consistency of boundary preserving performance. The results by [17] are screen captures of their result video.

7 Discussion

In this paper, we present a local linear regression model that is based on the local correlation between disparity values and intensity patterns, and is capable of handling large neighborhood window. Our method produces accurate disparity maps with sharp object boundaries in the Middlebury dataset. It outperforms competitive methods with regard to temporal consistency. In the future, we hope to pursue the following venues. First, we would like to implement the algorithm using iterative linear system solvers in a coarse to fine framework for computation efficiency. Second, we would like to investigate more robust weighting methods for $w_{i,j}$ in the presence of pixel noise and compression artifact. Third, we would like to explore this idea in optical flow estimation.

Acknowledgement. This work is supported in part by NSF EFRI-BSBA-0937847, NSF IIS-0845916, NSF IIS-0916441, a Sloan Research Fellowship, a Packard Fellowship for Science and Engineering, and a gift donation from Adobe Systems Incorporated.

References

1. Bobick, A.F., Intille, S.S.: Large occlusion stereo. IJCV 33, 181–200 (1999)
2. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. TPAMI (2001)
3. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. IJCV 47, 7–42 (2002)
4. Freeman, W., Pasztor, E.: Learning low-level vision. In: ICCV (1999)
5. Smith, B., Zhang, L., Jin, H.: Stereo matching with nonparametric smoothness priors in feature space. In: CVPR (2009)
6. Jung, H., Lee, K.M., Lee, S.U.: Stereo reconstruction using high order likelihood. In: ICCV (2011)
7. Sun, J., Shum, H.Y., Zheng, N.N.: Stereo matching using belief propagation. TPAMI 25, 787–800 (2003)
8. Woodford, O.J., Torr, P.H.S., Reid, I.D., Fitzgibbon, A.W.: Global stereo reconstruction under second order smoothness priors. In: CVPR (2008)
9. Yang, Q., Wang, L., Yang, R., Stewénius, H., Nister, D.: Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. TPAMI 31, 492–504 (2009)
10. Bleyer, M., Rother, C., Kohli, P.: Surface stereo with soft segmentation. In: CVPR (2010)
11. Bhusnurmath, A., Taylor, C.: Solving stereo matching problems using interior point methods. In: 3DPVT (2008)
12. Rhemann, C., Hosni, A., Bleyer, M., Rother, C., Gelautz, M.: Fast cost-volume filtering for visual correspondence and beyond. In: CVPR (2011)
13. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. TPAMI (2008)
14. He, K., Sun, J., Tang, X.: Guided Image Filtering. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 1–14. Springer, Heidelberg (2010)
15. Yoon, K.J., Kweon, I.S.: Adaptive support-weight approach for correspondence search. TPAMI 28, 650–656 (2006)
16. Xu, L., Jia, J.: Stereo Matching: An Outlier Confidence Approach. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 775–787. Springer, Heidelberg (2008)
17. Mei, X., Sun, X., Zhou, M., Jiao, S., Wang, H., Zhang, X.: On building an accurate stereo matching system on graphics hardware. In: GPUCV (2011)