# Solving Stereo Matching Problems using Interior Point Methods

Arvind Bhusnurmath
GRASP Laboratory
University of Pennsylvania
bhusnur4@seas.upenn.edu

Camillo J. Taylor
GRASP Laboratory
University of Pennsylvania
cjtaylor@cis.upenn.edu

## Abstract

*This paper describes an approach to reformulating the stereo matching problem as a large scale Linear Program. The approach proceeds by approximating the match cost function associated with each pixel with a piecewise linear convex function. Regularization terms related to the first and second derivative of the disparity field are also captured with piecewise linear penalty terms. The resulting large scale linear program can be tackled using interior point methods and the associated Newton Steps involve matrices that reflect the structure of the underlying pixel grid. The proposed scheme effectively exploits the structure of these matrices to solve these linear systems efficiently.*

## 1. Introduction and Related work

Stereo matching is one of the classic problems in computer vision. Like many other problems in the field, the stereo problem is often rephrased as an optimization problem where the goal is to minimize an objective function which models the costs associated with matching pixels at various disparities and includes terms that seek to reward overall 'smoothness' (except at occlusion boundaries).

Currently, these optimization problems are often solved using discrete optimization techniques such as Belief Propagation or Graph Cuts. In this work we seek to demonstrate that continuous convex optimization techniques can also serve as an effective tool for solving these kinds of optimization problems.

More specifically we describe an approach to tackling these types of optimization problems wherein the original objective function is approximated by a convex variant which can be solved using interior point methods. The resulting approach has a number of useful features. Firstly it allows us to naturally handle problems, like stereo where the decision variables are continuous without requiring any intermediate quantization. Secondly, we are able to naturally incorporate penalty terms involving more complex functions of the disparity values, such as a Laplacian term which allows us to better model slanting surfaces. Recent work by Woodford *et al*. [20] demonstrates the need for such terms. Thirdly this formulation involves solving a set of convex optimization problems with all of the guarantees that that entails.

This paper describes an approach to constructing a convex approximation to the stereo problem and shows how to exploit the structure of the resulting problem to effectively minimize an objective function involving hundreds of thousands of continuous variables.

An excellent survey of the methods of doing stereo was published in Scharstein and Szeliski's taxonomy of stereo algorithms [16]. Currently the standard method of testing these algorithms is by running them on the Middlebury data sets [17]. The website vision.middlebury.edu/stereo keeps track of the algorithms that perform well on these data sets.

As per [16] methods of solving stereo generally fall into either a local framework in which individual pixels or pixel windows are matched across images or global frameworks wherein a global energy function needs to be minimized.

Yoon and Kweon [25] introduce a new similarity measure that is based upon the central assumption that a pixel in the left image and a pixel in the right image that look alike and happen to be distinctive in each image, are more likely to be a match pair. Among local strategies this is the one that performs best.

Almost all of the top ranked algorithms on the Middlebury data set define a global energy function that is minimized for finding the disparities. The methods that are been successful in solving the data sets are dynamic programming, belief propagation and graph cuts.

Dynamic programming optimizes an energy function along a direction, which usually is along each of the scanlines. The most successful employment of this method is that by Hirschmuller in [9, 10] wherein the energy function is optimized in 8 or 16 different directions in what he terms a Semi Global Matching(SGM) method. Cues from segmentation are used for handling untextured areas. Segmentation based ideas are used heavily in the layered approaches (for example [2, 26]).

The current top ranked algorithms all use belief propagation. Usually the belief propagation is done using [7]. Sun *et al*. [19] use a symmetrical energy function which takes into account disparity maps as well as occlusion maps from both images. Klaus *et al*. [12] use oversegmentation and robust plane fitting through the segments to come up with a set of candidate disparity planes. Belief propagation is used to solve an energy function on the segments. An iterative algorithm using hierarchical belief propagation is proposed in Yang *et al*. [21] wherein the data term of the energy function is re-weighted in regions which are occluded or unreliable.

So far linear programming has not figured as a method of solving the energy minimization problem. Yanover *et al*. [23] did a comparative study on the usage of belief propagation and linear programming via Chekuri *et al*. [5] for solving energy minimization and their conclusion was that the linear programming module is usually infeasible for the energy minimization due to the extremely large number of variables involved. Our goal therefore is to show that linear programming is indeed a possible feasible approach with efficiency and can therefore be treated as an alternative to graph cuts [4, 14], belief propagation and the current variant of belief propagation - TRW [13]. The most successful use of linear programming for Energy minimization is that by Komodakis *et al*. [15] wherein duality theory is used to perfom an efficient primal dual optimization involving graph cuts, a combinatorial technique. Our method concentrates on solving the problem in the continuous domain and therefore has the potential to be a better fit for problems like stereo.

While stereo has not seen the successful usage of linear programming, it is worthwhile noting that a problem domain like motion has been solved by linear programming [11, 1]. The work by Jiang *et al*. [11] in matching feature points is similar to ours in that the data term for each pixel is approximated by a convex combination of points on the lower convex hull of match scores. However, this reduces the formulation into optimization for the interpolants associated with these convex hull points. Also this method uses the simplex method for solving the LP. By using interior point methods we are able to exploit the structure in a more efficient manner.

The remainder of the paper is organized as follows: Section 2 describes the novel LP formulation. Section 3 lays out the algorithm and Section 4 lists the key decisions to be made while solving the LP. Results on the Middlebury data set are provided in section 5. We conclude in Section 6 and describe future work in 7.

## 2. Formulation of the linear program

Assume a pair of stereo images labeled $L$ and $R$ for left and right image respectively are given. The sizes of the images are $W \times H$ and let $N = WH$ denote the total number of pixels that have to labeled with disparities. In the sequel we describe an approach to casting this labeling problem as an energy minimization problem that can solved via linear programming. The formulation will be explained in a 1 dimensional setting (per scanline) for ease of understanding. It is easy to extend it to include smoothness in both vertical and horizontal directions which is done during our actual implementation of the algorithm.

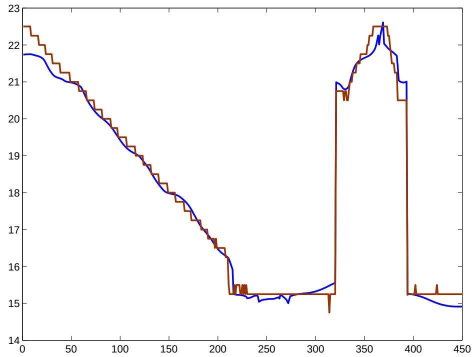


Figure 1. Example of the optimization in 1 dimension. The blue plot shows ground truth disparities along a scanline and the green shows the solution obtained by our interior point method.

Assume a labeling $d$ is given. Given a pixel location $i$ we will let $d_i$ will denote the corresponding disparity value. An energy function is needed that will quantitatively evaluate this labeling. Referring to Figure 1 which shows the ground truth disparities of a particular scanline in blue our goal is to get a solution which is close to it. Our current effort is shown as the dashed green plot.

As is the case with most energy minimization formulations, this involves the creation of a data term and a smoothness term. Let $E_{\text{data}}$ be the data term. We split the smoothness term into energy related to disparity gradient - $E_{\text{grad}}$ and energy related to disparity Laplacian - $E_{\text{lap}}$. This is in order to account for objects that are planar but not frontoparallel. We wish to allow for disparities that change in a linear manner over the image which is achieved by incorporating a penalty term that is related to the Laplacian. Then the energy of the labeling along any scanline can be expressed as

$$E(d) = \sum_{i=1}^{W} E_{\text{data}}(d_i) + \sum_{i=1}^{W-1} E_{\text{grad}}(d_i, d_{i+1}) + \sum_{i=2}^{W-1} E_{\text{lap}}(d_i, d_{i-1}, d_{i+1}) \qquad (1)$$

The additional term of the Laplacian in the energy function is a significant difference from most conventional algorithms which take into account only pairwise interactions of the pixels. The notion of smoothness can be thus be incorporated in different ways in this algorithm. The gradient term appeals to a piecewise constant model of the disparity solution while the Laplacian term appeals to a piecewise linear model of the solution.

For the gradient based energy term a natural choice is just $|d_i - d_{i+1}|$, but to allow for small changes in disparity we choose

$$E_{\text{grad}}(d_i, d_{i+1}) = \begin{cases} 0 \quad \text{if } |d_i - d_{i+1}| \geq \epsilon \\ w_{g_i}(|d_i - d_{i+1}| - \epsilon) \quad \text{otherwise} \end{cases} \tag{2}$$

To take into account disparity discontinuities this penalty term gets weighted differently at different pixel locations. $w_{g_i}$ is the weight corresponding to pixel $i$.

For the Laplacian based energy term we just use the discrete approximation of the Laplacian.

$$E_{\text{lap}}(d_i, d_{i+1}, d_{i-1}) = w_{l_i}|2d_i - d_{i-1} - d_{i+1}| \tag{3}$$

Again in a manner similar to the gradient term, $w_{l_i}$ represents the weight of this energy term for pixel $i$.

The formulation therefore currently incorporates terms involving absolute values of the variables. All these terms can be removed by adding variables and constraints as shown in [3]. Define a set of variables $y_g$ as proxies for the gradient terms. To capture the effect of 2 introduce the following constraints

$$\begin{array}{rcl} y_{g_i} & \geq & d_i - d_{i+1} - \epsilon \\ y_{g_i} & \geq & d_{i+1} - d_i - \epsilon \\ 0 & \leq y_{g_i} \leq & g_{\text{Max}} \end{array} \tag{4}$$

Similarly introduce variables $y_l$ as proxies for the Laplacian terms. To capture the effect of 3

$$\begin{array}{rcl} y_{l_i} & \geq & -d_{i-1} + 2d_i - d_{i+1} \\ y_{l_i} & \geq & d_{i-1} - 2d_i + d_{i+1} \\ 0 & \leq y_{l_i} \leq & l_{\text{Max}} \end{array} \tag{5}$$

Thus at this stage, the only component of the energy function that is not convex is the data term. We address this issue in the following subsection.

## 2.1. Convex approximation of the data term

Conventionally in stereo the data term associated with assigning the label $d$ to a pixel $(x_L, y_L)$ is given by some function of the intensity differences between a patch around $L(x_L, y_L)$ and a patch around $R(x_L - d, y_L)$. Assume we have a discrete set of potential disparity values $d_{\text{min}} \leq \ldots \leq d_{\text{max}}$. Corresponding to each disparity value $d_i$ we obtain a score that indicates the likelihood of the pixels matching up. Given these scores, the disparity value that should be chosen based only on this information is the one that provides the minimum score. Our aim is to lower bound the score values of each pixel with a continuous, convex function of the disparity.

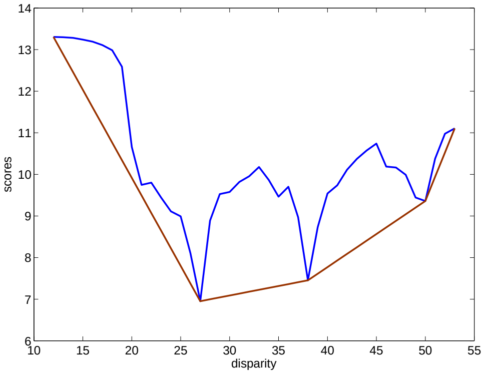


Figure 2. Example of the proposed convex approximation of a score function. The blue plot is that of the scores corresponding to different disparities for a particular pixel and the brown line segments show the approximation. Essentially a lower convex hull is computed.

To do this we construct a piecewise linear function in a manner similar to computing the convex hull of the entries in the score array as shown in Figure 2.

Note that we do not require the disparity values in the (disparity, score) pairs to correspond to integral disparity values. This allows us to easily incorporate subpixel disparity scores directly into the score function.

The constraints associated with the score values are that they need to lie above the line segments. Let variable $y_{m_i}$ be associated with the match score for pixel $i$. Assume there are $S_i$ segments that approximate the score function for the $i^{th}$ pixel, then we have the following constraints corresponding to the match variables.

$$\begin{array}{rcl} y_{m_i} & \geq & a_{ji}d_i + b_{ji},\ j \in \{1, 2, \ldots S_i\} \\ 0 & \leq & y_{m_i} \leq y_{Max_i} \end{array} \tag{6}$$

where variables $a_{ji}$ and $b_{ji}$ represent the slope and intercept for the $j^{\text{th}}$ segment. The data term $E_{data}$ can now simply be written as a weighted sum of these $y_m$ score values. The weights are needed in order to take care of the occlusions. Pixels that are occluded have no match and therefore their match weights need to be 0. However, they still show up in the gradient and Laplacian terms and through this process, the background disparities usually get filled in.

Therefore the final formulation of the objective function is

$$E(d) = \sum_{i=1}^{W} w_{m_i} y_{m_i} + \sum_{i=1}^{W-1} w_{g_i} y_{g_i} + \sum_{i=1}^{W-2} w_{l_i} y_{l_i} \qquad (7)$$

And this objective function is optimized under the matching constraints of 6, the gradient constraints of 4 and the Laplacian constraints of 5.

## 3. Using interior point method to solve the LP

The optimization problem in equation 7 can now be minimized using the interior point log barrier method [3]. The linear programming problem can be stated more compactly as

$$\begin{aligned} &\min w^T x \\ \text{st} \quad & Ax \le b \end{aligned} \qquad (8)$$

where $x$ is a vector formed by concatenating all of the decision variables, $x = \begin{bmatrix} d & y_m & y_g & y_l \end{bmatrix}^T$, and $w$ is the vector formed by the corresponding weights, $w = \begin{bmatrix} 0 & w_m & w_g & w_v \end{bmatrix}^T$.

The matrix $A$ concatenates all of the linear constraints and is given by

$$A = \begin{bmatrix} A_m & -I_m & 0 & 0 \\ B_d & 0 & 0 & 0 \\ 0 & B_m & 0 & 0 \\ G & 0 & -I^{W-1} & 0 \\ -G & 0 & -I^{W-1} & 0 \\ 0 & 0 & B_g & 0 \\ L & 0 & 0 & -I^{W-2} \\ -L & 0 & 0 & -I^{W-2} \\ 0 & 0 & 0 & B_l \end{bmatrix} \qquad (9)$$

where $B_d, B_m, B_g, B_l$ correspond to the bounds on the $d, y_m, y_g, y_l$ variables.

$$A_m = \begin{bmatrix} a_{11} & 0 & \cdots & \cdots & 0 \\ a_{21} & 0 & \cdots & \cdots & 0 \\ \vdots & 0 & \cdots & \cdots & 0 \\ a_{S_1 1} & 0 & \cdots & \cdots & 0 \\ 0 & a_{12} & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ 0 & \vdots & 0 & \cdots & 0 \\ 0 & a_{S_2 2} & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & a_{1M} \\ 0 & \cdots & \cdots & 0 & \vdots \\ 0 & \cdots & \cdots & 0 & a_{S_M M} \end{bmatrix} \qquad (10)$$

The matrix $I_m$ is a sparse matrix with the same fill pattern as $A_m$, but has 1s in place of the $a_{ij}$ entries.

$G$ represents the gradient constraints

$$G = \begin{bmatrix} 1 & -1 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & 0 & 1 & -1 \end{bmatrix} \qquad (11)$$

$L$ represents the Laplacian constraints

$$L = \begin{bmatrix} -1 & 2 & -1 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & -1 & 2 & -1 \end{bmatrix} \qquad (12)$$

We now use interior point barrier method [3] to solve the problem.

Let the Newton step direction be $\delta x = [\delta d \delta y_m \delta y_g \delta y_l]^T$. Let $S = b - Ax$.

Then the Newton step involves solving

$$[A^T \text{diag}(S_i^{-2}) A] \delta x = g \qquad (13)$$

where $\text{diag}(S_i^{-2})$ is a diagonal matrix with the squared inverse of each entry of $S$ along the diagonal.

$$g = -tw - A^T e \qquad (14)$$

where $e$ is a column vector with each entry $e_i = 1/S_i$.

Split $g$ into 4 blocks corresponding to the columns of $A$. $g = \begin{bmatrix} g_d & g_m & g_g & g_l \end{bmatrix}^T$

It can be seen that $A^T \text{diag}(S_i^{-2}) A$ takes on the following form

$$\begin{bmatrix} H_d & H_m^T & H_g^T & H_l^T \\ H_m & D_m & 0 & 0 \\ H_g & 0 & D_g & 0 \\ H_l & 0 & 0 & D_l \end{bmatrix} \qquad (15)$$

where

$$\begin{aligned} H_d &= A_m^T D_1 A_m + B_d^T D_2 B_d + G^T D_g^+ G + L^T D_l^+ L \\ H_m &= -A_m^T D_1 I_m \\ H_g &= G D_g^- \\ H_l &= L D_l^- \end{aligned} \qquad (16)$$

and $D_g^+, D_g^-, D_l^+, D_l^-, D_1, D_g, D_l, D_m, D_2$ are all diagonal matrices.

Solve for $\delta y_m$, $\delta y_g$ and $\delta y_l$ in terms of $\delta d$ to get the following equation

$$\begin{aligned} &(H_d - H_m^T D_m^{-1} H_m - H_g^T D_g^{-1} H_g - H_l^T D_l^{-1} H_l) \delta d \\ &\quad = (g_d - H_m^T D_m^{-1} g_m - H_g^T D_g^{-1} g_g - H_l^T D_l^{-1} g_l) \end{aligned} \qquad (17)$$

which can be written more concisely as

$$H'_d \delta d = g'_d \tag{18}$$

This is the fundamental step of the whole algorithm and needs to be done efficiently.

Let us examine the structure of $H'_d$. $H'_d$ contains scaled versions of $G^T G$, $L^T L$ and $A_m^T A_m$. $A_m^T A_m$ is diagonal, $G^T G$ is tridiagonal and $L^T L$ is pentadiagonal. This means that $L^T L$ is the dominant contributor to the sparsity structure of the matrix in question. Therefore the main step of the algorithm is just the solution of a system of symmetric and positive definite pentadiagonal equations.

As noted before the actual implementation of the algorithm involves terms in both horizontal and vertical directions. This results in block pentadiagonal systems which have pentadiagonal blocks along the diagonal and diagonal blocks on the off diagonal.

Essentially we have a structure that looks like

$$\begin{bmatrix} A_1 & B_1^T & C_1^T & 0 & 0 \\ B_1 & \ddots & \ddots & \ddots & 0 \\ C_1 & \ddots & \ddots & \ddots & C_{n-2}^T \\ 0 & \ddots & \ddots & \ddots & B_{n-1}^T \\ 0 & 0 & C_{n-2} & B_{n-1} & A_n \end{bmatrix} \tag{19}$$

where the $A_i$ are pentadiagonal and the $B_i$ and the $C_i$ are diagonal.

Such structured matrices are ideal candidates for iterative solvers like conjugate gradients [8]. Also by performing incomplete Cholesky decomposition of these matrices in a manner similar to [6], good preconditioning can be achieved which significantly reduces the number of conjugate gradient iterations needed for convergence.

The algorithm for solving the optimization problem is stated below in Algorithm 1.

**Algorithm 1** Calculate the optimal disparity($d$) for the problem in 8

```
choose an initial set of disparities and $y_m$, $y_g$ and $y_l$
choose an initial $t$ value and a $\mu$ value and set newton decrement = $\infty$
while $t \leq t_{\text{max}}$ value do
    while Newton decrement is greater than a threshold do
        Compute the Hessian from equation 15
        Compute the gradient $(g)$ from equation 14
        Compute $\delta d$ from equation 18
        Use $\delta d$ to obtain $\delta y_m$,$\delta y_g$ and $\delta y_l$.
        Compute Newton decrement = $g^T \delta x$
        Compute the step size $\beta$ using line search
        Update $d$, $y_m$, $y_g$, $y_l$ by $\beta\delta d$, $\beta\delta y_m$, $\beta\delta y_g$ and $\beta\delta y_l$ respectively.
    end while
    $t = \mu * t$
end while
```

## 4. Using the solution framework for Stereo

The first step in our stereo matching procedure is to compute a correlation volume which encodes for each pixel the costs associated with all legal disparities. For this purpose we employed the Adaptive Matching Score method described by Yoon and Kweon in [24]. In this implementation we used 33x33 windows and the following parameter values $\gamma_p = 36$, $\gamma_c = 7$. Prior to computing the support weights in each window, the color image was smoothed using a 5x5 median filter to enhance color homogeneity while respecting image edges.

Once the match scores had been computed for the integer disparities, a quadratic fit was used in the vicinity of local minima to establish subpixel offset and subpixel score values. This resulted in a set of (disparity, score) values for each pixel where the disparity values are not all integral.

The convex approximation procedure described in section 1 was then applied to the score function associated with every pixel to produce a piecewise linear convex lower approximation for use in the optimization procedure.

The structure of the constraint system associated with the LP is determined by these convex approximations and the gradient and laplacian equations which are fixed.

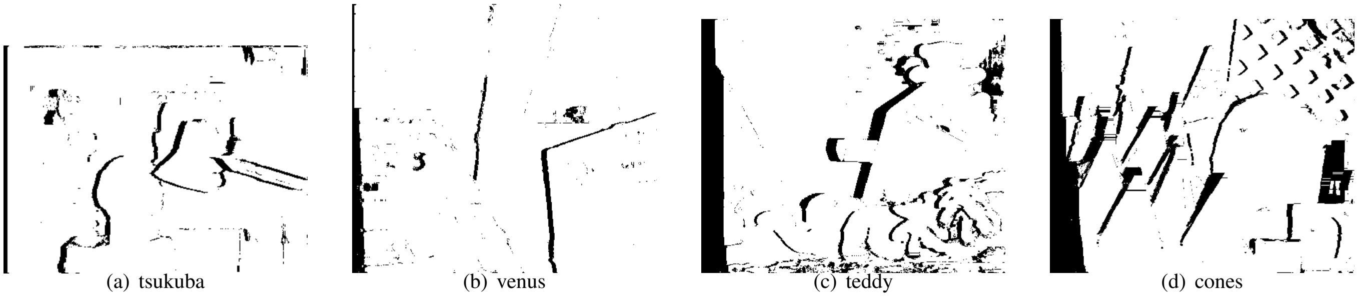
What remains then is to specify the weights associated with the match terms, $w_m$ , the gradient terms and the laplacian terms.

The match weights associated with each pixel should represent the degree of confidence in that pixels match score. The reasoning here is that pixels that are occluded or otherwise suspicious should have low $w_m$ values reflecting the fact that their match costs should be devalued while the weights pixels whose disparities are more certain should be enhanced.

We begin by using the initial disparity estimates provided by the Yoon and Kweon matcher to decide on which pixels appear reliable by using a simple left right check. Additionally, we use the disparity discontinuities found in the right image to predict occlusions in the left image and vice versa in a manner similar to the approach described in Sun *et al.* [19].

The resulting match weights $w_m$ essentially represent an approximation for the occluded regions in the image. Figure 3 shows what these match weight images look like for various images.

The weights associated with the gradient and laplacian term are computed by considering the color differences be-



(a) tsukuba (b) venus (c) teddy (d) cones

Figure 3. Match weights being used in the algorithm

tween neighboring pixels. Similar weighting terms are used in most modern stereo matching algorithms. In this particular case the weight associated with the difference between disparity i and disparity i+1 is computed as follows

$$W_{gi} = s_g \exp(-(\Delta c_i/\sigma)^2) \tag{20}$$

Where $\Delta c_i$ represents the Euclidean difference between the colors associated with pixel $i$ and pixel $i+1$ in Lab space.

The weight associated with the Laplacian term $y_{li} \geq |d_{i-1} - 2d_i - d_{i+1}|$ is computed as follows

$$W_{li} = s_l \exp(-((\Delta c_{i-1}/\sigma)^2 + (\Delta c_i/\sigma)^2)/2) \tag{21}$$

Note that the Laplacian weight considers the difference between pixel $i$ and $i-1$ as well as the difference between pixels $i$ and $i+1$. In this implementation the parameter values were set as follows $s_g = 1, s_l = 2, \sigma = 6$. Once again the image is smoothed with a 5x5 median filter before the color differences deltaic are computed.

## 5. Results

The proposed method was applied to the Middlebury data set and the results are summarized in figure 4. The first two columns show the images and ground truth disparities respectively while the last two columns show the results of our implementation of the Adaptive Support weight method which is used to construct the correlation volume and the final results of the optimization procedure.

We also show the pixels that have errors exceeding a 0.5 pixel threshold in figure 5. Note that this is the strictest possible threshold in the Middlebury dataset.

The entire procedure was implemented in Matlab and the time required to perform the LP optimization was 9 minutes on a Intel Core 2Duo system with 2GB of memory. During this procedure the system performed 40 Newton iterations where each step took approximately 13.4 seconds. Again the vast majority of the computational effort is spent inverting the sparse system in equation 18. As is typical of a barrier method, the system converges to an answer quite rapidly.

Table 1 shows the percentage of erroneous pixels with a 0.5 pixel threshold and compares it against other state of the art methods. It also shows the improvement we get by using our method on top of our implementation of the Yoon and Kweon matcher.

## 6. Conclusion

In this paper we have described a novel approach to employing convex optimization methods to computer vision problems. The approach proceeds by approximating the original energy function with a piecewise linear convex lower approximation and then tackling the resulting convex problem.

There are two main observations underlying this approach. The first is that the proposed convex functions provide a reasonable model of the true objective function in most cases and effectively capture the ambiguities inherent in the data. The second is that the Hessian matrices associated with the resulting optimization problems have special structure which mirrors the clique structure of the underlying objective function. By exploiting this structure we are able to effectively solve problems involving hundreds of thousands of variables. Problems that were previously beyond the reach of generic LP solvers

We have demonstrated this approach in the context of stereo vision and have described results obtained by applying this method to standard data sets. However, the approach is quite general and could be applied to a number of problems including motion estimation, homography estimation and image deformation estimation.

## 7. Future Work

There are a number of directions that could be taken to further improve the results on the given data sets. The current system performs a single pass to arrive at the final result. One can imagine a system that proceeds in several rounds where the results of one round are used to refine the weights used in the next round.

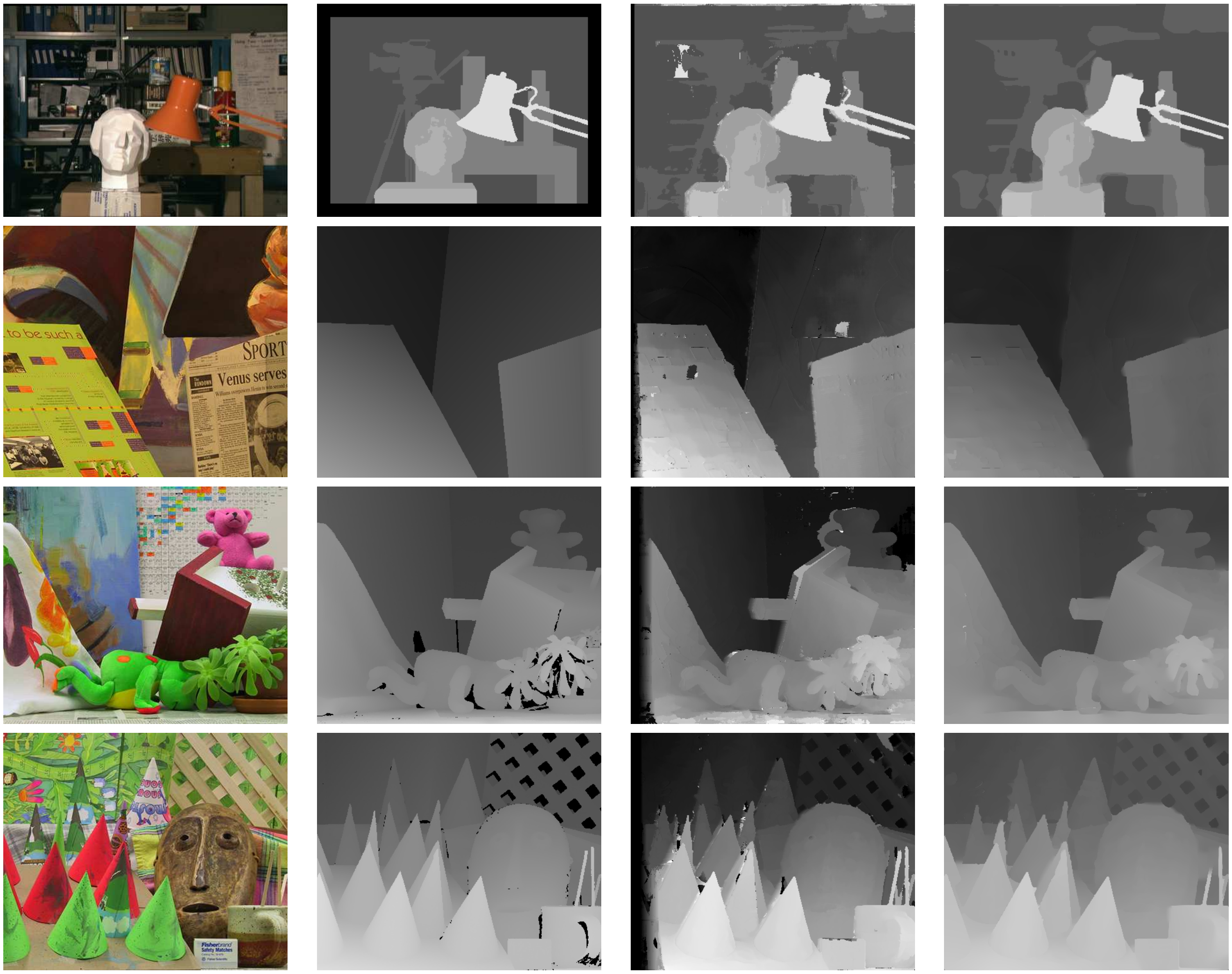
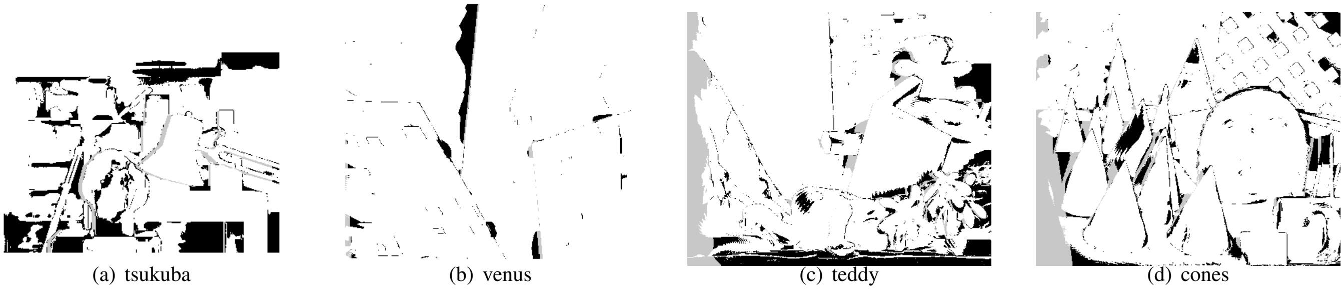


Figure 4. Results on the Middlebury data set. The first column shows the ground truth, the second shows the results obtained by us using Yoon and Kweon Adaptive weighting scheme [24]. The cost volume obtained from this algorithm is sent in as input to our optimization scheme and the final result after the optimization is shown in column three.



(a) tsukuba

(b) venus

(c) teddy

(d) cones

Figure 5. Bad pixels with a 0.5 threshold

Other methods make use of explicit color segmentations and plane fitting methods. These approaches are particularly effective on this data set where planar surfaces predominate. This type of analysis could easily be incorporated into the overall system. Explicit segmentations could also serve to improve performance on and around occlusion

Table 1. Our place in the Middlebury evaluation table with a 0.5 pixel error threshold. Please note that the row corresponding to Adaptive weighting consits of scores that we get upon implementing the algorithm by ourselves and hence are not completely in agreement with scores in [24]

| | Tsukuba | | | Venus | | | Teddy | | | Cones | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | nonocc | all | disc | nonocc | all | disc | nonocc | all | disc | nonocc | all | disc |
| SubPixDoubleBP [22] | 8.78 | 9.45 | 14.9 | 0.72 | 1.12 | 5.24 | 10.1 | 16.4 | 21.3 | 8.49 | 14.7 | 16.5 |
| C-SemiGlobal [10] | 13.9 | 14.7 | 18.9 | 3.3 | 3.82 | 10.9 | 9.82 | 17.4 | 22.8 | 5.37 | 11.7 | 12.8 |
| ImproveSubPix | 8.96 | 9.66 | 16.2 | 4.62 | 5.41 | 16.9 | 11.0 | 17.8 | 24.1 | 4.90 | 10.8 | 12.2 |
| OverSegBP [26] | 7.75 | 8.17 | 13.8 | 4.33 | 4.73 | 16.8 | 13.2 | 19.3 | 27.5 | 6.53 | 12.6 | 14.0 |
| SemiGlob [9] | 13.4 | 14.3 | 20.3 | 4.55 | 5.38 | 15.7 | 11.0 | 18.5 | 26.1 | 4.93 | 12.5 | 13.5 |
| GenModel [18] | 7.89 | 10.0 | 18.5 | 4.59 | 6.03 | 23.5 | 14.8 | 22.8 | 31.8 | 10.2 | 20.2 | 19.0 |
| **LP** | **18.7** | **19.3** | **20.8** | **3.0** | **3.71** | **18.8** | **13.5** | **19.2** | **29.7** | **9.57** | **16.2** | **20.3** |
| Adaptive Weight (our) | 19.8 | 21.4 | 18.6 | 3.67 | 5.28 | 16.9 | 14.0 | 22.9 | 28.0 | 9.24 | 19.5 | 18.3 |

boundaries.

While our Matlab implementation does a reasonable job of performing a large scale optimization on a quarter of a million continuous variables, we feel that there is ample room for further improvement in this area. More specifically the structure of equation 19 makes it particularly amenable to parallel solution methodologies that could exploit the abundance of floating point performance afforded by GPUs and other parallel architectures.

## References

[1] M. Ben-Ezra, S. Peleg, and M. Werman. Real-time motion analysis with linear programming. In *ICCV*, 1999. 2

[2] M. Bleyer and M. Gelautz. A layered stereo matching algorithm using image segmentation and global visibility constraints. In *ICIP*, 2004. 1

[3] S. Boyd and L. VandenBerghe. *Convex Optimization*. Cambridge University Press, 2004. 3, 4

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11), 2001. 2

[5] C. Chekuri, S. Khanna, J. Naor, and L. Zosin. Approximation algorithms for the metric labeling problem via a new linear programming formulation. In *SODA*, 2001. 2

[6] P. Concus, G. Golub, and G. Meurant. Block preconditioning for the conjugate gradient method. *SIAM Journal of Scientific Computing*, 6(1), 1985. 5

[7] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. In *CVPR*, 2004. 2

[8] G. Golub and C. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996. 5

[9] H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *CVPR*, 2005. 1, 8

[10] H. Hirschmuller. Stereo vision in structured environments by consistent semi-global matching. In *CVPR*, 2006. 1, 8

[11] H. Jiang, M. Drew, and Z.-N. Li. Matching by linear programming and successive convexification. *PAMI*, 29(6), 2007. 2

[12] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *ICPR*, 2006. 2

[13] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28(10), 2006. 2

[14] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *PAMI*, 26(2), 2004. 2

[15] N. Komodakis and G. Tziritas. A new framework for approximate labeling via graph cuts. In *ICCV*, 2005. 2

[16] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47:7–42, 2002. 1

[17] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *CVPR*, 2003. 1

[18] C. Strecha, R. Fransens, and L. V. Gool. Combined depth and outlier estimation in multi-view stereo. In *CVPR*, 2006. 8

[19] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum. Symmetric stereo matching for occlusion handling. In *CVPR*, 2005. 2, 5

[20] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. In *CVPR*, 2008. 1

[21] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister. Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. In *CVPR*, 2006. 2

[22] Q. Yang, R. Yang, J. Davis, and D. Nister. Spatial-depth super resolution for range images. In *CVPR*, 2007. 8

[23] C. Yanover, T. Meltzer, and Y. Weiss. Linear programming relaxations and belief propagation - an empirical study. *Journal of Machine Learning Research*, 7:1887–1907, 2006. 2

[24] K.-J. Yoon and I. S. Kweon. Adaptive support-weight approach for correspondence search. *PAMI*, 28(4), 2006. 5, 7, 8

[25] K.-J. Yoon and I. S. Kweon. Stereo matching with the distinctive similarity measure. In *ICCV*, 2007. 1

[26] C. L. Zitnick and S. B. Kang. Stereo for image-based rendering using image over-segmentation. *IJCV*, 75(1), 2007. 1, 8