Supervised Disparity Estimation

Patrick Vandewalle[a] and Chris Varekamp[b]

[a]Philips Research, High Tech Campus 36, 5656AE Eindhoven, The Netherlands;
[b]Philips Consumer Lifestyle, High Tech Campus 37, 5656AE Eindhoven, The Netherlands

ABSTRACT

We introduce supervised disparity estimation in which an operator can steer the disparity estimation process. Instead of correcting errors, we view the estimation process as a constrained process where the constraints are indicated by the user in the form of control points, scribbles and contours. Control points are used to obtain accurate disparity estimates that can be fully controlled by the operator. Scribbles are used to force regions to have a smooth disparity, while contours create a disparity discontinuity in places where diffusion or energy minimization fail. Control points, scribbles and contours are propagated through the video sequence to create temporally stable results.

Keywords: disparity estimation, stereo, autostereoscopic display

1. INTRODUCTION

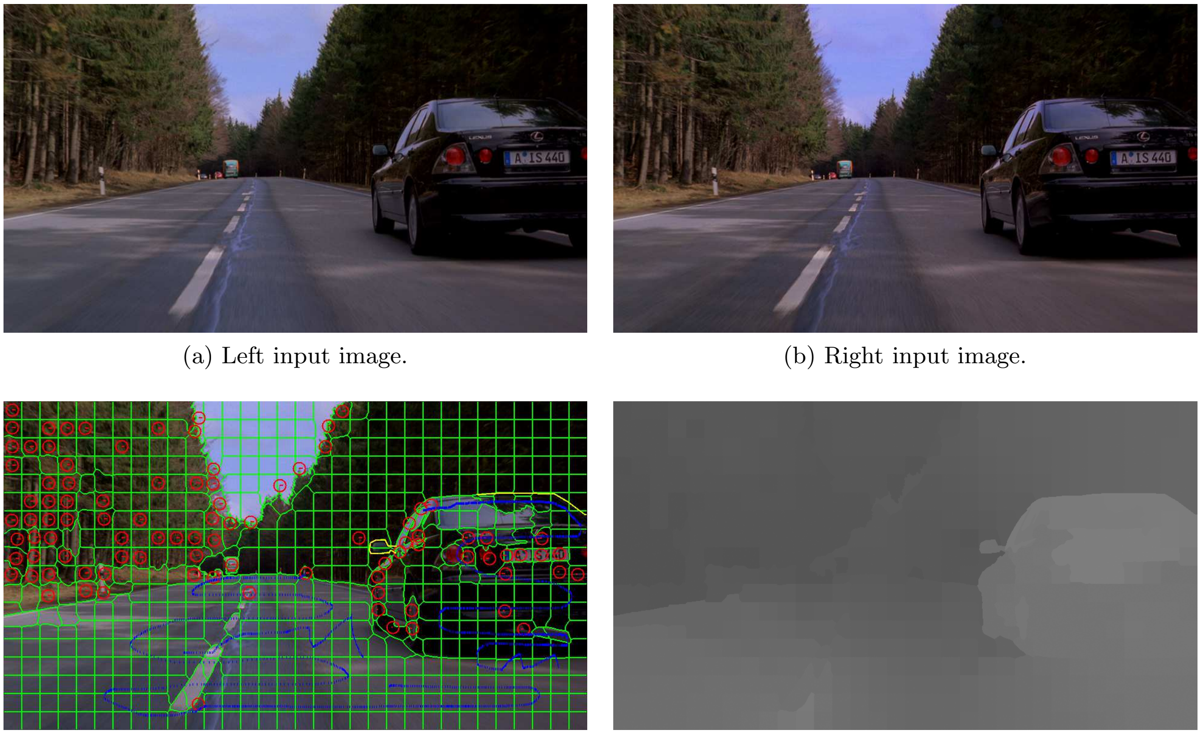
Recently, 3D has received a lot of attention from various sides: more and more movie productions are released in 3D in the movie theaters, the first 3D broadcasting channels are announced, and display manufacturers present various types of 3D displays, both stereoscopic and auto-stereoscopic. On stereoscopic displays, a viewer is presented with two views (one for each eye) using special glasses (shutter-based, polarized, etc.). On auto-stereoscopic displays, a larger number of views is typically generated (e.g. 9 or 15) and viewers do not have to wear special glasses. Each view is sent in a different direction using a lenticular sheet on the display or barriers in the display, such that the viewer's two eyes automatically receive two different views. Such displays generally also offer some (limited) ability to move around the sweet spot and look around objects. In order to display 3D content on an auto-stereoscopic display, a disparity map is typically required. Similarly, if we want to re-purpose stereoscopic 3D content (e.g. for different displays), disparity maps are extremely useful. This is particularly relevant when adapting content created for viewing on a cinema screen to use on a TV or a mobile display.

1.1 Unsupervised disparity estimation (automated)

Automated disparity estimation from stereo input video has been widely investigated (see the work by Scharstein and Szeliski[1] and the corresponding website* for an overview). Various approaches have been tested in the past, ranging from pixel-wise dense estimation to sparse, feature-based matching. A good overview of the various approaches is given by Brown et al.[2] Recently, Markov Random Field approaches have been very popular and successful for disparity estimation.[3] Such methods integrate multiple information sources into the estimation process, such as disparity matching and smoothness using a graph cuts method,[4] or intensity matching and segmentation cues using a belief propagation solution method.[5]

Lang et al. have proposed a method for disparity warping without requiring knowledge of a dense disparity map.[6] They use image warping techniques taking sparse disparity estimates and saliency measures into account. A similar warping-based technique is presented by Chang et al.[7]

While the results using automated algorithms have strongly improved in recent years, they are typically not sufficiently accurate for high quality estimation as it is typically required for broadcasting scenarios.[8] The main causes for failure of such algorithms are homogeneous regions and objects with similar color. In homogeneous regions, a large number of correspondences can be found with similar matching cost but varying disparity. It is therefore difficult to determine the correct disparity. When objects with similar color overlap in the images, it is very difficult to determine precisely the border and thus the disparity discontinuity between the objects.



(a) Left input image.

(b) Right input image.

(c) Left image with annotations.

(d) Resulting disparity map.

Figure 1. Illustration of supervised disparity estimation. A disparity map (d) is estimated from a stereoscopic image pair (a)-(b) and a set of user annotations (c). A super-pixel segmentation (green) forms the basis of the processing and is shown to the user for information. Control points (red), scribbles (blue) and contours (yellow) are added by the user to steer disparity estimation.

1.2 Supervised disparity estimation (semi-automated)

User interaction can be applied to overcome these problems and achieve any desired quality level. Taking user input into account, the problem is changed from 'maximizing quality of the disparity map given all available information in a stereo video' to 'selecting effective user inputs that can lead to any desired quality of a disparity map with a minimum amount of work'. Any quality level can now (in principle) be achieved by increasing the amount of user input. To our knowledge, there is little previous work on supervised disparity estimation for the purpose of view interpolation. Srivastava et al. have extended the learning-based algorithm by Saxena et al.[9] to include interactivity in 3D estimation from monoscopic still images.[10] After an automatic 3D reconstruction, a user can first roughly indicate the foreground object(s) if present. Next, he can place scribbles to indicate regions of the background that belong to the same plane. These constraints are then integrated in a Markov Random Field approach. Lo et al. have developed a 'stereoscopic copy and paste' algorithm that allows a user to interactively segment an object from one stereoscopic sequence and paste it into another one.[11] Special attention is given to disparity errors, occlusion handling and shadow generation. Note also that the warping algorithms by Lang et al.[6] and by Chang et al.[7] also allow interactive editing of the disparity mapping function. This is different however from our focus here, where we try to interactively obtain a dense disparity map.

There have also been several attempts to combine user interaction with computer vision in order to derive either a high quality segmentation or a high-quality 3D map using a monoscopic image sequence as input. For the purpose of automated rotoscoping (the process of tracking contours in a video sequence), Agarwala et al. showed how a user can interactively refine the position of curves, after which an automatic tracker is restarted.[12] Segmentation and matting from (foreground and background) scribbles was presented using geodesic distances by Bai and Sapiro[13] and using minimization of a quadratic cost function by Levin et al.[14] Russell and Torralba

[*] http://vision.middlebury.edu/stereo/

have developed LabelMe3D: a learning-based approach to infer 3D models from rough contours and semantic labels given by a user.[15]

VideoTrace is a system that generates 3D models from a single video using structure from motion analysis.[16] The shape of an object can be 'traced' by the user to produce polygons. VideoTrace does not require pixel-accurate line input since it fits the input curves to local strong superpixel boundaries of a segmentation that is computed in advance.

Efficient methods for semi-automated depth estimation from monoscopic input use manually annotated key-frames and propagate depth between those key-frames.[17]

1.3 This paper

In this paper, we target view-interpolation using stereo video as input. Instead of drawing disparity maps for key-frames, we now draw various annotations to steer the disparity estimation algorithm. The concept is illustrated in Figure 1. Typically, those annotations are applied at the locations that cause problems for a standard disparity estimator. The actual disparity estimation is steered by these annotations. We present two disparity estimation approaches. A first one performs actual disparity estimation only at indicated positions where reliable estimates can be made. The disparity map for an entire frame is then computed using a diffusion approach similar to the one used in our earlier work.[18] However, this time we solve for disparity values instead of object class labels. Our second approach uses an energy minimization using α-expansion and graph cuts.[4]

Our processing is region-based and uses a segmentation into super-pixels as input for the algorithm. Disparity maps are propagated over a sequence using a segment-based motion field. We derive the segment-based motion by diffusing sparse motion vectors obtained using block matching over the segmentation map. This approach differs from the segment-based motion estimation by Ernst et al.,[19] where segments are used for matching. Unlike VideoTrace, we can modify the geometry of super-pixels via the placement of contours. At the same time, these contours modify the constraints between regions. Our contours do not need to be closed polygons, which means that they only need to be placed at positions where disparity estimation fails.

This paper is structured as follows: An overview of our approach is given in Section 2.1. In Section 2.2, we describe our segmentation algorithm. The actual disparity estimation algorithm is presented in Section 2.3, and the different types of user annotation are described in Section 2.4. Section 2.5 discusses the propagation of user annotations between frames. In Section 3, we describe the experiments performed to test our approach and the results on different stereo sequences.

2. APPROACH

2.1 Overview

As already indicated above, various problems can occur when estimating a disparity map. First of all, erroneous matches may be found between left and right images because of similar structures appearing in different parts of the image. To address this, we will use disparity estimates for specific salient points that can be indicated (and removed) by the user to guide the estimation for the rest of the image. These are typically the points on which the disparity can be reliably estimated. Such points will be called *control points* in this paper.

A second issue is the breakup of an object consisting of parts with different colors. We use color similarity as a strong grouping cue to keep pixels or segments together. However, this cue sometimes works counterproductively. For instance, the walls of a house may get different disparity values from its roof, just because the walls are more similar in color to the street than to the roof. We introduce *scribbles* as a solution, indicating that parts of the image belong together and should get similar disparities.

Finally, separate objects with different disparities may get merged together because they have similar colors. This can either lead to errors in the super-pixel segmentation, or to errors in the disparity estimation process. These issues are addressed using object *contours*, which represent a discontinuity in the disparity map.

The operator can annotate a left image from a stereo video with such control points, scribbles and contours (see Section 2.4 for details). After each modification of the annotation, the operator can recalculate and evaluate the disparity map. The estimation of the disparity map is performed using either a diffusion process or a Markov

Random Field approach. In both methods, relations between neighboring regions are based on both image properties and operator supplied annotations. The disparity estimation is explained in detail in Section 2.3, after a description of our super-pixel segmentation algorithm in the next Section.

2.2 Segmentation

The use of super-pixels allows fast response in the editor when the annotation is updated. We use a region fitting approach described by Oliver and Quegan[20] in combination with the global energy minimization algorithm by Duda et al.[21] Initially, the segmentation consists of squares. The boundaries between these squares are iteratively updated by proposing to move pixels that touch a boundary to a neighbor region.[20] A proposed move is accepted only if a total energy measure decreases. The energy consists of a term that measures for each pixel the deviation of the pixel color from the mean region color combined with a term indicating boundary smoothness. More details can be found in the work by Oliver and Quegan.[20] Even with the smoothness constraint, it can happen that long thin regions occur at the boundary between objects, often caused by blur introduced by the imaging process (e.g. due to limited depth-of-field). To avoid long thin regions with multiple edges close together, the region fitting step is done twice with a thin-region removal step in between. This process is described in earlier work.[22] Specifically, after the first region fitting step, a majority filter is applied to the region label map in a local window of 7×7 pixels around each pixel. This step removes thin regions from the segmentation by assigning a pixel the region label that most frequently occurs in the local neighborhood of that pixel. However, the morphological filter displaces edge positions. Region fitting is therefore done a second time to obtain a new local minimum. For the second run, the region smoothness term is weighted more than for the first run to avoid that thin regions can be introduced again.

2.3 Disparity estimation

The disparity estimation operates on the super-pixels of an over-segmentation. The disparity estimation consists of two parts. First, matching is done on the control points. When the application is started, a set of robust control points is automatically initialized. We first select the centers of super-pixel regions where the neighborhood has sufficient contrast. For these points, we perform disparity estimation using block matching on a block of 15×15 pixels from the left image to the right image. We use an L_1 error norm to compute the match costs (sum of absolute differences). Next, we take the corresponding pixel found in the right image, and perform the same disparity estimation back to the left image (as it was also done for example by Fua[23]). Only regions for which the center maps back to the original pixel are kept as robust control points.

More control points can be entered later by the operator, or existing control points can be removed. The disparity value for such manually entered control points is estimated using (only left-to-right) block matching as described above and assigned to the region in which the control point falls. All control points provide a hard constraint D_{cp} for the disparity of a region. If matching goes wrong, the operator can either remove the control point and choose a new one, or manually correct the disparity at a control point, thereby keeping full control of these disparities.

Given the known disparities at the typically small set of regions containing control points, we need to find the unknown disparities at all other regions using the constraints specified between the regions.

We compare two approaches of finding a disparity for each region: disparity estimation by diffusion and energy minimization using so-called α-expansion.[4]

2.3.1 Disparity estimation using diffusion

In the first approach we only use the disparities at control points and extrapolate from these regions using only color similarity as a cue. This approach is simple, efficient and does not require explicit disparity estimation in regions that do not contain control points. We can write the problem as a *Dirichlet problem*.[24] Following the notation used by Bendito et al.,[24] let $\Gamma = (V, E)$ denote the region adjacency graph with vertex set V consisting of the super-pixel regions, and with edge set E for the edges between pairs of neighboring regions. Let $\mathcal{L}$ denote the *weighted Laplacian matrix* of graph Γ. This matrix of size $|V| \times |V|$ has non-zero entries $\mathcal{L}_{ij}$ if region i is a neighbor of region j ($(i, j) \in E$). The diagonal elements of such a matrix satisfy

$$\mathcal{L}_{ii} = -\sum_{j \neq i} \mathcal{L}_{ij}. \tag{1}$$

We now seek a predictor that satisfies

$$(\mathcal{L}D)_i = \frac{\sum_{j \neq i} \mathcal{L}_{ij} D_j}{\sum_{j \neq i} \mathcal{L}_{ij}} - D_i = 0 \qquad \forall i \in F, \tag{2}$$

$$D_i = D_{\text{cp},i} \quad \forall i \in F^C, \tag{3}$$

where F is the subset of V containing the regions i for which the disparity D_i is unknown, and $F^C = V \setminus F$. In other words, the regions in F^C are the regions containing control points. The predictor honors the true disparities at control points, while making the disparity field spatially consistent by requiring that the prediction error in (2) is zero. This results in a set of linear equations (2)-(3). In order to solve these equations, we use an iterative algorithm for updating the current solution for D_i:

$$D_i^k \leftarrow \frac{\sum_j \mathcal{L}_{ij} D_j^{k-1}}{\sum_j \mathcal{L}_{ij}} \qquad i \in F. \tag{4}$$

The disparities D_i of regions containing a control point ($i \in F^C$) remain fixed at $D_{\text{cp},i}$. Initially, we set the disparities for all regions in F to zero. We then use 50 iterations of the above diffusion twice, where we run through the disparity vector D in forward and reverse directions.

The off-diagonal entries depend both on image properties and on the constraints imposed by the operator via annotations. Color similarity between regions is used to spatially propagate disparity at control points to other regions in the graph. Initially there are no annotations by the operator and the elements $\mathcal{L}_{ij}$ are set to depend on color similarity only:

$$\mathcal{L}_{ij} = g(i, j) \equiv e^{-\alpha(|r_i - r_j| + |g_i - g_j| + |b_i - b_j|)}, \tag{5}$$

where (r_i, g_i, b_i) is the mean color vector for region i with each component taking values from [0, 255] and α is a parameter (we use a value of 0.1). In the absence of control points, scribbles and contours, the elements of $\mathcal{L}_{ij}$ are never modified. However, as soon as the operator draws an annotation, $\mathcal{L}_{ij}$ is modified accordingly (see Section 2.4).

2.3.2 Disparity estimation using α-expansion

In the second disparity estimation approach we also compute disparity measurements in regions that do not contain control points. Following the notation of Szeliski et al.[3] we minimize the following energy function:

$$E = E_{\text{data}} + \lambda E_{\text{smooth}}. \tag{6}$$

Our data term E_{data} consists of the per-region data cost of matching a block of 15×15 pixels between the left and right image for a certain disparity. We use an L_1 error norm to compute the match cost, and normalize it by the block size and maximum pixel values. The smoothness term is the sum over all pairs of neighboring regions of a smoothness energy between the pair of regions. The smoothness energy between two regions is computed as the difference between their estimated disparity values, normalized by the maximum difference. In our experiments, we set $\lambda = 0.1$. We use α-expansion as energy minimization method. Specifically we have implemented the algorithm described by Boykov et al.[4] for α-expansion, and the algorithm described by Boykov and Kolmogorov[25] for graph-cuts. In our implementation, we adopted the table by Boykov et al.,[4] p. 1228 for the weights assigned to the edges of the graph based on data energy and smoothness energy terms.

2.4 Annotation

We will now describe how user annotations (control points, scribbles and contours) can steer the disparity estimation for both estimation methods.

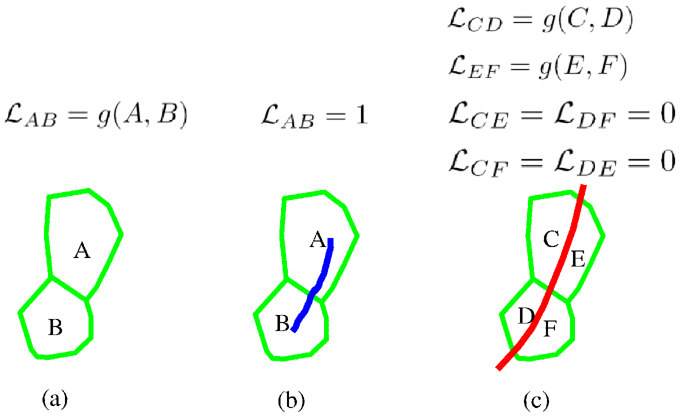

$\mathcal{L}_{AB} = g(A,B)$
$\mathcal{L}_{AB} = 1$
$\mathcal{L}_{CD} = g(C,D)$
$\mathcal{L}_{EF} = g(E,F)$
$\mathcal{L}_{CE} = \mathcal{L}_{DF} = 0$
$\mathcal{L}_{CF} = \mathcal{L}_{DE} = 0$
A
B
A
B
C
E
D
F
(a)
(b)
(c)

Figure 2. Illustration on how the scribble and contour annotations modify the region segmentation and elements of the *weighted Laplacian matrix* in the diffusion-based disparity estimation algorithm. (a) Default situation where edge weights result from color differences between adjacent regions (these weights range from 0 to 1). (b) A scribble connects two regions with the result that weights are set to 1. (c) A contour is drawn that splits both regions A and B in two, creating four new regions C,D,E and F. The region pairs (C,E), (D,F) (C,F) and (D,E) receive weights of 0 thereby avoiding disparity leakage across the contour. The region pairs (C,D) and (E,F) receive new color dependent weights.

2.4.1 Adding a control point

By placing a control point, the operator can select a position in the image that is likely to give a reliable disparity estimate. Placing a control point in region i applies disparity estimation at that position and sets the disparity value for that region correspondingly. In the diffusion algorithm, it also sets all the elements on row i of the Laplacian matrix $\mathcal{L}$ to 0. In the α-expansion algorithm, we set the link t_p^α between any proposed disparity α and the region node i to infinity, such that no new proposed disparity can be accepted anymore (using the same notation as Boykov et al.[4]).

Most of the times the disparity is reliably estimated. However, it can happen that disparity estimation at the control point fails. The operator can therefore use keyboard arrows to increase or decrease the disparity of a control point after it has been selected by clicking with the mouse in its vicinity. Visual feedback is provided by plotting the disparity vector.

2.4.2 Adding a scribble

A scribble connects regions together, such that they get similar disparities. If two neighboring regions i and j contain the same scribble, our diffusion approach sets elements $\mathcal{L}_{ij}$ and $\mathcal{L}_{ji}$ of $\mathcal{L}$ to 1 (see Figure 2). Similarly, in our α-expansion algorithm, the smoothness links between regions i and j (which are applied through an auxiliary node if they have different disparities) are increased with a factor 1000. Disparity values can thus diffuse easily and surfaces may be produced with constant or gradually evolving disparities.

2.4.3 Adding a contour

Adding a contour changes both the region segmentation and the region neighbor relations. Figure 2 (c) shows how the contour modifies the adjacency graph in our diffusion algorithm. Note that in practice, we do not set the indicated weights to zero, but instead remove the relations from the graph, which has the same effect. Since new regions are formed, the disparity vector D and the weighted Laplacian matrix $\mathcal{L}$ grow in size. As a contour indicates a separation between objects at different distances, the weights between regions separated by a contour should be set to 0. Let R_i be an element from the set of new regions that result from cutting regions in the segmentation. Let R_j be a neighbor region of R_i. Note that region R_j can be either a new or an existing region. Further, let $L^{(old)}(i)$ denote for new region R_i, the corresponding old region. The algorithm for updating the region adjacency graph and propagation weights is given in Figure 3.

```
for all $i$ do                                   ▷ Loop over all new regions
    $k \leftarrow L^{(old)}(i)$                  ▷ Get old region
    for all $j$ do                               ▷ Loop over all neighbors
        $l \leftarrow L^{(old)}(j)$              ▷ Get old region of neighbor
        if $k = l$ then                          ▷ New contour
            $c_{ij} \leftarrow true$
            $\mathcal{L}_{ij} \leftarrow 0$
        else
            if $c_{kl} = true$ then              ▷ Inherit old contour
                $c_{ij} \leftarrow true$
                $\mathcal{L}_{ij} \leftarrow 0$
            else                                 ▷ Color dependent weights
                $c_{ij} \leftarrow false$
                $\mathcal{L}_{ij} \leftarrow g(i,j)$
            end if
        end if
    end for
end for
```

Figure 3. Algorithm for adding a contour in the diffusion approach.

Similar operations are applied in our α-expansion algorithm, such that the segments are split into new segments, just like for the diffusion approach. Neighbor relations across the contour are also removed, such that they are not connected anymore in the graphs.

Adding a contour changes the segmentation geometry since regions are split. It is therefore necessary to locally reprocess control points and scribbles each time a contour is added or removed.

2.5 Temporal propagation

Control points, scribbles and contours are the constraints on the basis of which the disparity map is produced. To obtain temporally consistent disparity maps, the constraints need to follow the true motion of objects. For clarity of the description, we will present here only propagation in the case of the diffusion-based disparity estimation presented above. Temporal propagation for the α-expansion approach can be derived along the same lines.

Once the operator has entered enough annotations to obtain a disparity map with sufficient quality, disparity for consecutive frames in a video shot can be obtained fully automatically if the annotations are propagated without error. This is the ideal situation. In practice, operator interaction is still needed depending on the success of the motion compensation step. To propagate the annotations we need a dense motion field (i.e. a motion vector per region). To obtain this motion vector field we use exactly the same approach as for the disparity estimation: The region-based motion field is also obtained via a diffusion process thereby re-using the weighted Laplacian matrix $\mathcal{L}$. So instead of diffusing disparities given at control points we diffuse motion vectors that are given at control points. The process is detailed in the next subsections.

2.5.1 Propagation of control points

Motion compensation works best for control points. First, the automatically generated control points are not likely to lie close to object occlusions due to the robust selection process used for the disparity estimator. Second, if a point becomes occluded, we can easily remove the entire track and start a new control point track. For each control point, block matching is used to find the motion vector $\boldsymbol{v}_i \equiv (u_i, v_i)$ that results in the smallest error when matching with the next frame. We apply matching over a block of 15×15 pixels and search 81 pixels horizontally and vertically.

2.5.2 Propagation of scribbles

With the motion compensation of scribbles we face two problems. First, matching pixels that touch scribble points may be difficult since scribbles, unlike control points, can cover homogeneous color regions. Second, a scribble can become partially occluded with the result that local foreground and background objects are wrongly required to have the same disparity. To solve the second problem we need a motion compensated scribble and a motion compensated dense disparity map. This will allow us to check whether some points on a scribble are covered in the next frame. Such covered parts are removed from the scribble. To our rescue comes the fact that the constraints that the operator has supplied to produce a high-quality disparity map for frame t are directly useful to obtain a high-quality motion field for the frame pair $(t, t+1)$. In fact, the weighted Laplacian matrix $\mathcal{L}$ is directly re-used to obtain estimates for the motion vector $\boldsymbol{v}_i$ of region i:

$$\boldsymbol{v}_i = \frac{\sum_j \mathcal{L}_{ij} \boldsymbol{v}_j}{\sum_j \mathcal{L}_{ij}} \qquad i \in F. \tag{7}$$

As for disparities, in F^C, the motion vector from control points is assumed fixed and known:

$$\boldsymbol{v}_i = \boldsymbol{v}_{\text{cp}} \qquad i \in F^C. \tag{8}$$

These motion vectors $\boldsymbol{v}_{\text{cp}}$ are estimated using a block matching process, as described above.

Motion compensation of the scribble works as follows. We first create a warped disparity map D_{warp} in frame $t+1$ by warping the points (x, y) of the disparity map in frame t onto points (x', y') in D_{warp} using the coordinate transformation:

$$\begin{aligned} x' &= x + u_i; \\ y' &= y + v_i, \end{aligned} \tag{9}$$

where $i = S(x, y)$ denotes the region number at position (x, y) in the segmentation map. In case of occlusion, we need to determine whether a point (x, y) becomes covered or not in the next frame. We therefore take for each pixel (x', y') the maximum disparity (minimum depth) to ensure that the disparity of the closest object is present in the warped disparity map. Next to warping the dense disparity map to the next frame, we also warp each scribble to the next frame. Each scribble point with location (x, y) is motion compensated using the region based motion vector $\boldsymbol{v}_i$ that follows from the segmentation map S (with $i = S(x, y)$). A warped scribble point with coordinates (x', y') is only added to the motion compensated scribble if it remains un-occluded in frame $t+1$. To check this we compare the disparity $D_{\text{scribble}}(x', y')$ for each warped scribble point with the disparity $D_{\text{warp}}(x', y')$ of the warped disparity map at the same location. A warped scribble point is thus only added if

$$D_{\text{scribble}}(x', y') \geq D_{\text{warp}}(x', y'). \tag{10}$$

2.5.3 Propagation of contours

We introduced contours to achieve high-quality disparity especially at object boundaries. Over time, contours need to track the local foreground object boundary. We use a similar approach as for the propagation of scribbles. However, this time we cannot simply use motion vector $\boldsymbol{v}_i$ to compensate the position of a contour point positioned in region i to the next frame. We need to make sure that the correct motion vector is taken, i.e. the motion vector of the local foreground object. To ensure that this is the case, we search in a 4-connected neighborhood around each contour point (with coordinates (x, y)) the neighboring point that has maximum disparity (minimum depth) in the disparity map:

$$(x_{min}, y_{min}) = \arg \max_{(x'', y'') \in N(x,y)} (D(x'', y'')), \tag{11}$$

where $N(x, y)$ denotes the 4-connected neighborhood of the contour point at pixel location (x, y). Finally, each point of the contour is motion compensated using the motion vector $\boldsymbol{v}_i$ (with $i = S(x_{\text{min}}, y_{\text{min}})$) to make sure that the contour moves with the foreground object. We do not add the occlusion handling for contours, as the contours themselves remain typically visible.

3. EXPERIMENTAL RESULTS

To evaluate the use of our tool we experimented with different realistic stereo videos and with different amounts of interaction. First, we show its behavior on a standard image pair taken from the Middlebury dataset.[26] We used the 'Teddy' image at quarter resolution, which is available online[†]. The results using as disparity estimator

[†] http://vision.middlebury.edu/stereo/data/scenes2003/

the presented diffusion approach (second row) and the α-expansion approach (third row) are shown in Figure 4. From the initial results (without manual annotations) in Figure 4(c)-(f), we can see that each algorithm has different problem areas. The diffusion approach performs bad if regions with distinct color have no control points (e.g. the roof of the house), while the α-expansion approach has difficulties around disoccluded areas (e.g. the area to the left of the house). Therefore, they also require different annotations. From Figure 4(e)-(h), we can see though that with the right annotations, both algorithms provide accurate results.

Both algorithms have been implemented in C++. Note in this respect that we have implemented the α-expansion algorithm as described in Section 2.3.2 and by Boykov et al.,[4, 25] but without the optimizations described in those articles. In their current implementation without optimizations, both algorithms perform disparity estimation in the order of at most a few seconds on the 450×375 pixel Teddy image. In our experience, running times of the disparity estimation should be at most in the order of 1s for easy interactive operation. This provides a user a rapid feedback showing the results of his interactions, avoiding long waiting times. We consider that this can be achieved with both proposed algorithms by performing some further optimization steps.

Next, we illustrate temporal behavior (using diffusion-based disparity estimation) in Figure 5. A first frame with its annotations, motion field and resulting disparity map using diffusion-based estimation is shown in Figure 5(a)-(c)-(e)-(g). These annotations are then propagated over the next frames with some minimal additional annotations, and the result eight frames later is shown in Figure 5(b)-(d)-(f)-(h). From these figures, it is clear that with minimal effort, a coherent sequence of disparity maps can be obtained.

Now, we demonstrate that arbitrary quality levels of the disparity map can be obtained by increasing the amount of user input. Figure 6(a)-(b) shows the disparity map obtained after automated diffusion-based estimation using only automatically detected control points in our disparity estimator. Note the errors around the back of the horse, which are due to the color similarity with the background. Because the colors are similar, the super-pixel segmentation is incorrect. The disparity estimation creates a gradual transition through this region, again because of the similar colors. A similar problem causes errors in the disparity values of the fence behind the horse, where some parts get the same disparity as the horse itself. After removing 2 control points and adding 3 contours, the errors around the back of the horse are corrected (as well as some other errors, see Figure 6(c)-(d)). The horse and background now get separate disparities and are well aligned with the object edges. Adding more annotations results in a disparity map where also the fence and some other issues are corrected (Figure 6(e)-(f) and a detail in Figure 7).

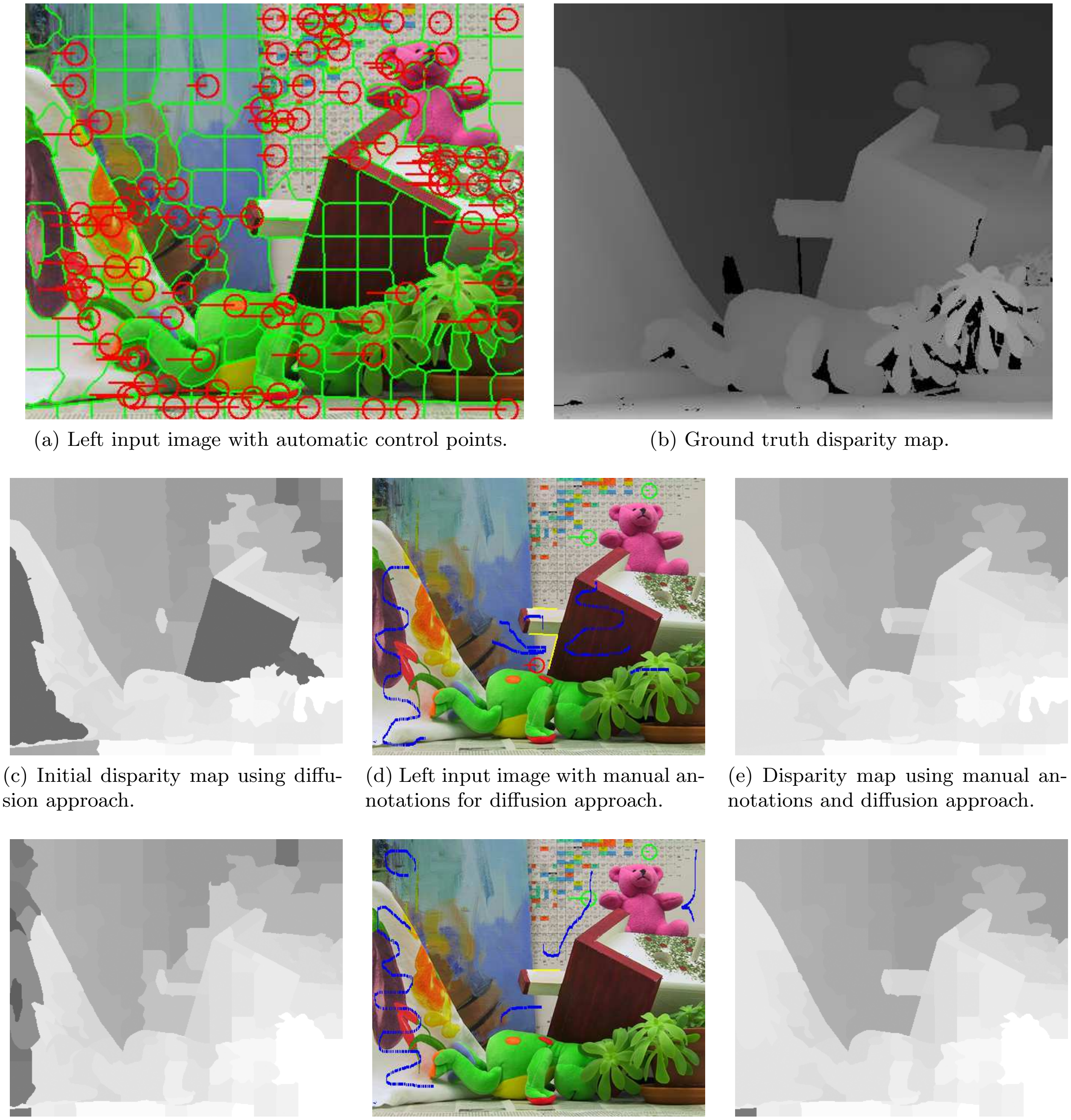
For illustration, we also add the initial and annotated version using the α-expansion approach (see Figure 8). Figure 8(a)-(d) shows the initial result using the automated α-expansion approach. The results using the α-expansion algorithm on the annotations made above for the diffusion algorithm are not good (see Figure 8(b)-(e)). This illustrates the fact that different types of annotations are needed, due to the differences between the algorithms. For the α-expansion method, more annotations are needed around disocclusion areas, while less are needed around color discontinuities. Results with tailored annotations for the α-expansion approach are shown in Figure 8(c)-(f).

Finally, we tested our approach using the diffusion algorithm on a more complex (indoor) scene, containing a large number of objects (see Figure 9). The initial disparity map obtained using only automatically selected control points is very inaccurate. Although the complexity of the scene leads to a large amount of manual annotations, a high quality disparity map can be obtained. Note that mainly fine details that are imprecisely segmented in the initial segmentation (like the plants in the background) require a lot of manual annotation. Sharp depth transitions are achieved in the final result.

As can be seen from the figures, the super-pixel segmentation is also drawn in the user interface. From our own experience, this helps a user when placing annotations.

4. CONCLUSION

We have presented a supervised method to create disparity maps from stereo video material. We use a disparity estimation algorithm that starts from a set of feature points for which disparity can be precisely estimated, and uses these values to estimate a disparity map using either a diffusion process or a Markov Random Field energy minimization with α-expansions. User input is given in the form of (additional) feature points, scribbles that



(a) Left input image with automatic control points.

(b) Ground truth disparity map.

(c) Initial disparity map using diffusion approach.

(d) Left input image with manual annotations for diffusion approach.

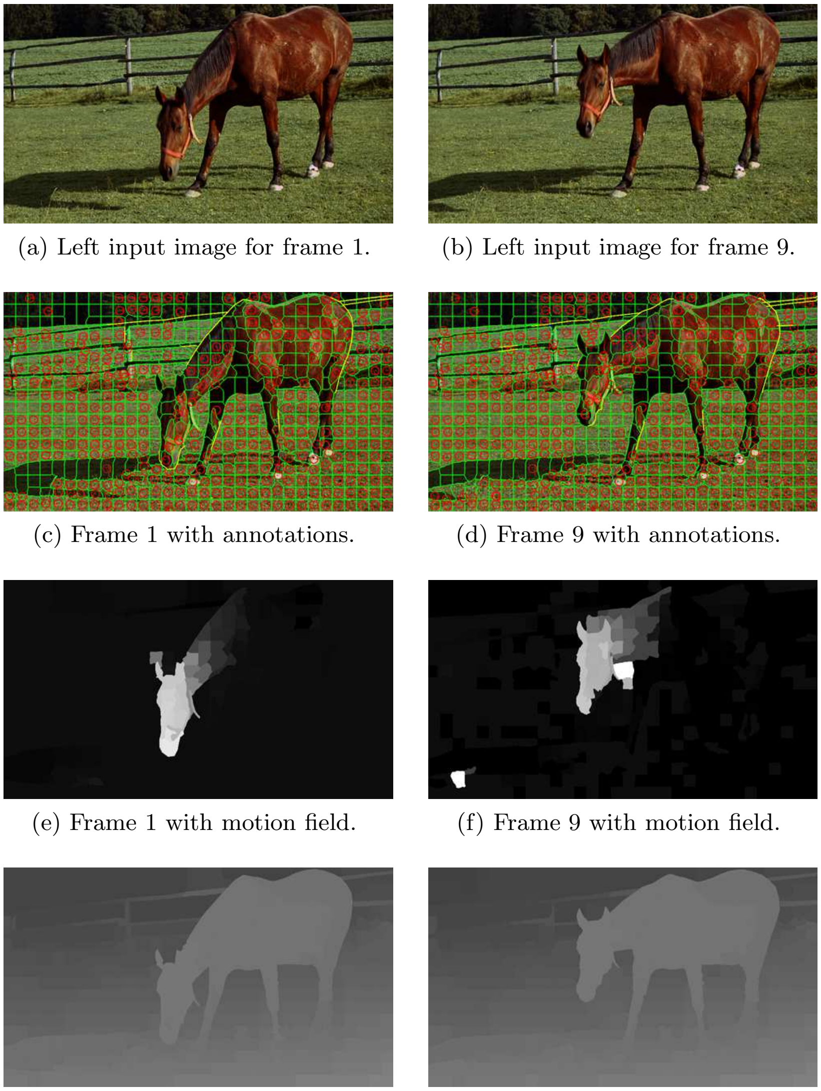
(e) Disparity map using manual annotations and diffusion approach.

(f) Initial disparity map using α-expansion.

(g) Left input image with manual annotations for α-expansion approach.

(h) Disparity map using manual annotations and α-expansion approach.

Figure 4. Results using diffusion approach (second row) and α-expansion approach (third row) for 'Teddy' image from Middlebury dataset.[26] Similar results can be obtained with both approaches, but each approach requires different annotations due to algorithmic differences. Note that large disparities (nearby objects) are encoded with bright gray levels, and correspondingly, small disparities (remote objects) have dark gray levels. Disparity values in the range -40 to 40 pixels of horizontal displacement have been mapped from black to white. A different scaling was used compared to the ground truth map.



(a) Left input image for frame 1.

(b) Left input image for frame 9.

(c) Frame 1 with annotations.

(d) Frame 9 with annotations.

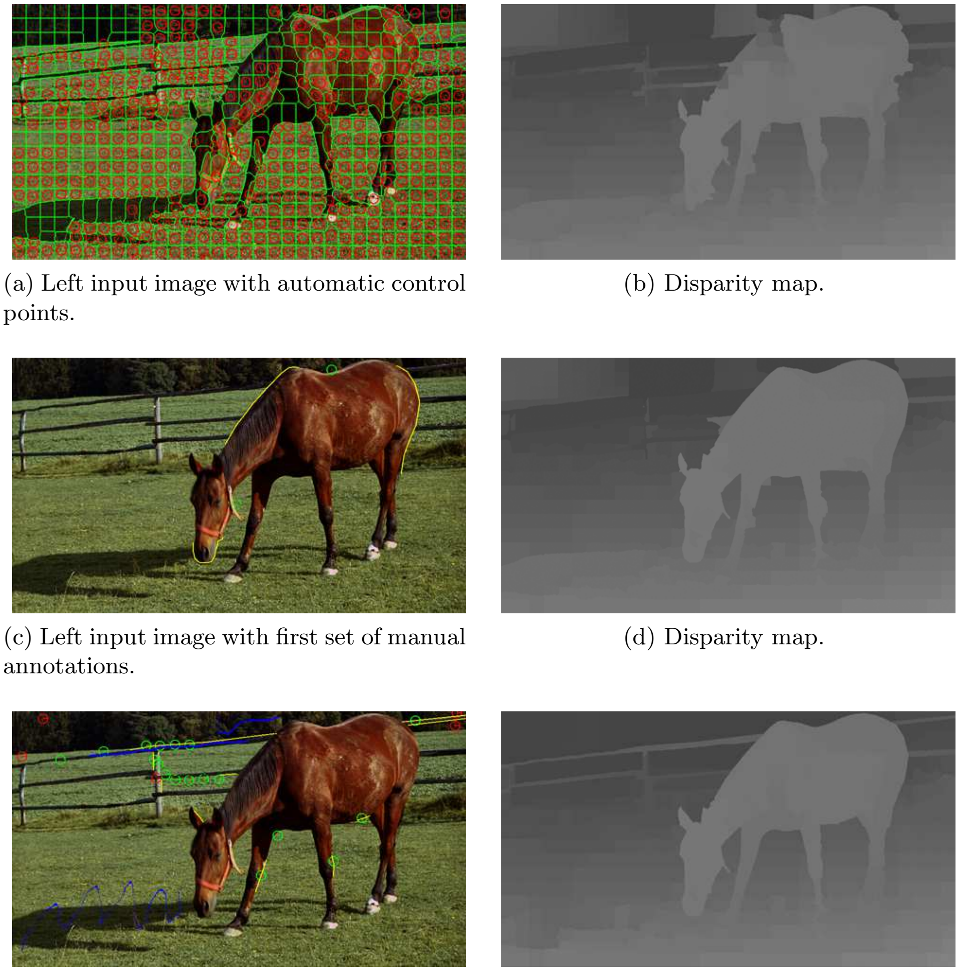
(e) Frame 1 with motion field.

(f) Frame 9 with motion field.

(g) Disparity map for frame 1.

(h) Disparity map for frame 9.

Figure 5. Temporal behavior. A disparity map (g) is estimated from a first frame (a) using user annotations (c). These annotations are then propagated through the next frames. The user can remove (and correct) in each frames the incorrectly propagated annotations, resulting in a temporally very stable sequence of disparity maps. The result after 8 frames is illustrated in (h). The magnitude of the motion is displayed in (e)-(f). This magnitude was clipped on a maximum of 15 pixels and mapped from black to white.



(a) Left input image with automatic control points.

(b) Disparity map.

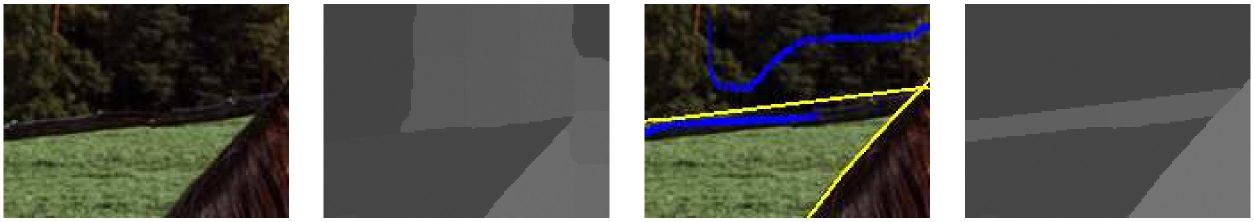
(c) Left input image with first set of manual annotations.

(d) Disparity map.

(e) Left input image with a second set of manual annotations.

(f) Disparity map.

Figure 6. Disparity maps for increasing number of annotations. (a)-(b) Automated result without manual input. (c)-(d) Manual annotations and result after removing 2 control points and adding 3 contours. (e)-(f) Additional manual annotations compared to (c) and their result on the disparity map.



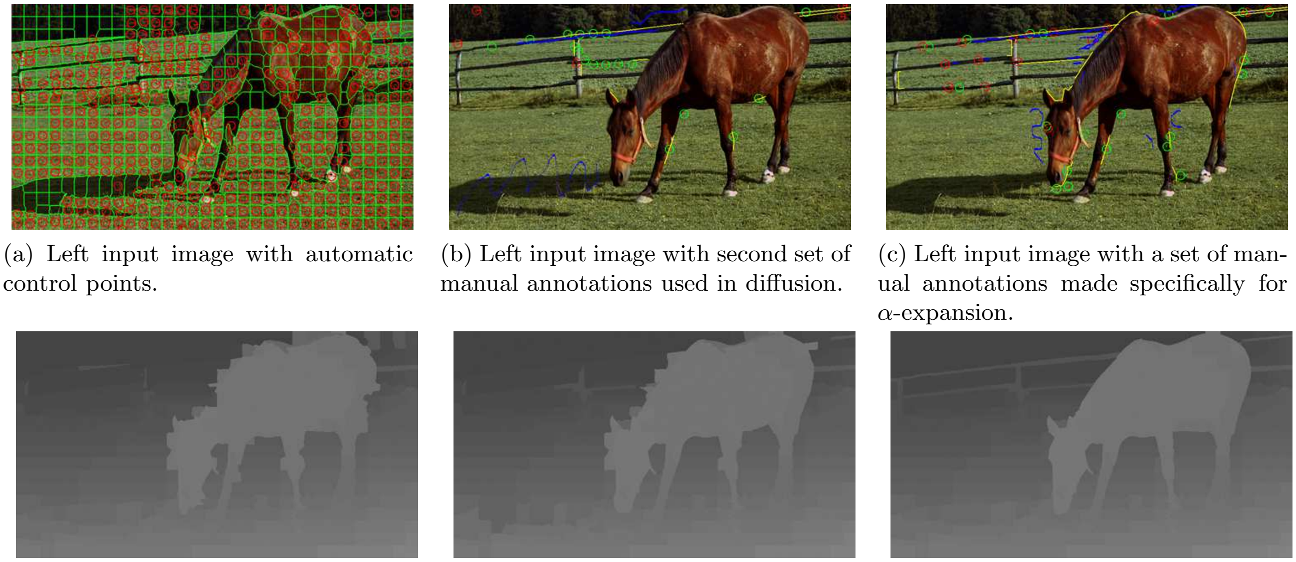
(a) Detail of left input image with automatic control points.

(b) Detail of disparity map.

(c) Detail of left input image with second set of manual annotations.

(d) Detail of disparity map.

Figure 7. Detail of Figure 6 to illustrate the effect of placing contours. Using a contour, the disparity transition at the back of the horse is placed at the right position. The fence is now (correctly) placed at an intermediate depth using another contour and two scribbles.



(a) Left input image with automatic control points.

(b) Left input image with second set of manual annotations used in diffusion.

(c) Left input image with a set of manual annotations made specifically for α-expansion.

(d) Initial disparity map using α-expansion.

(e) Disparity map using α-expansion on diffusion annotations.

(f) Disparity map using α-expansion on specific annotations.

Figure 8. Disparity maps using α-expansion approach. (a)-(b) Automated result using α-expansion without manual input. (c)-(d) Manual annotations used for diffusion approach and result of α-expansion approach on these annotations. (e)-(f) Manual annotations made for α-expansion approach and result of α-expansion approach on these annotations.

indicate regions with the same disparity, and contours indicating discontinuities in disparity. Disparities are propagated through a sequence by propagating the user annotations. We have illustrated the performance of our algorithm on stereo sequences with varying content.

While temporal propagation generally works well, it fails for regions with large motion and around unsharp edges.

Overall, we conclude that the presented interactive disparity estimation algorithm using control points, scribbles and contours is a powerful combination of annotation types, allowing a user to steer the disparity estimation process for a stereo video sequence. Higher quality can be achieved by adding more interaction. We have shown that such annotations are very flexible, and can be incorporated into different disparity estimation algorithms. We have applied it to a diffusion-based method and an α-expansion energy minimization method with good results.

ACKNOWLEDGMENTS

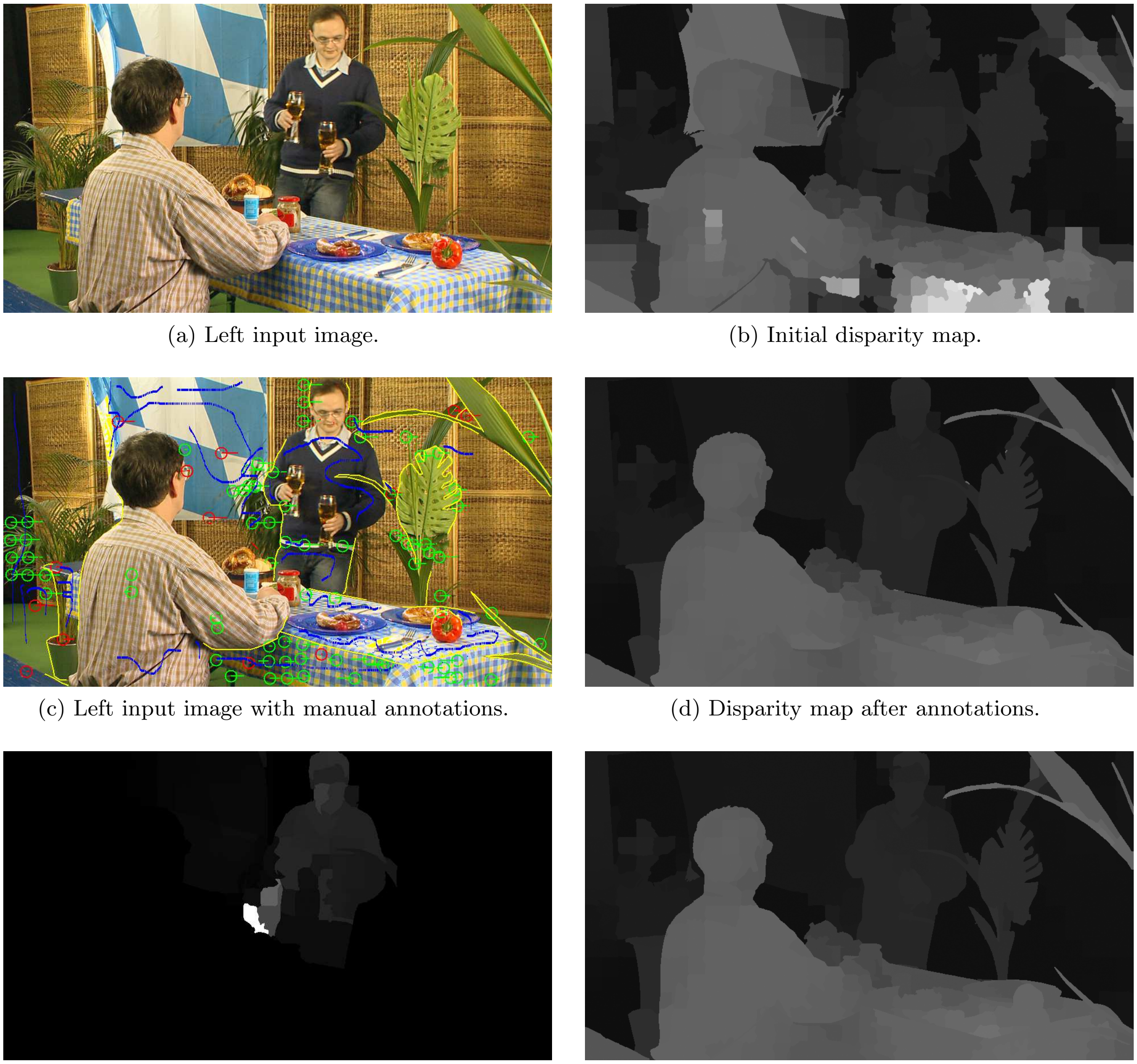
We are grateful to Evgeny Verbitskiy (Philips Research) for pointing out that the disparity estimation problem as put down in this paper is in fact a *Dirichlet problem*, and to Rene Klein Gunnewiek (Philips Research). We thank KUK for providing the stereo image sequences generated in the context of the 3D4YOU project. This work is partially supported by the European Commision in the 3D4YOU project, ICT 215075.

REFERENCES

[1] Scharstein, D. and Szeliski, R., "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision* **47**(1/2/3), 7–42 (2002).

[2] Brown, M., Burschka, D., and Hager, G., "Advances in computational stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**, 993–1008 (August 2003).

[3] Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., and Rother, C., "A comparative study of energy minimization methods for markov random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**, 1068–1080 (June 2008).



(a) Left input image.

(b) Initial disparity map.

(c) Left input image with manual annotations.

(d) Disparity map after annotations.

(e) Motion field to next frame.

(f) Disparity map propagated to the next frame.

Figure 9. Supervised disparity estimation for a complex scene. High quality can be achieved by adding more annotations. Fine details that are inaccurately segmented in the initial segmentation require a lot of input.

[4] Boykov, Y., Veksler, O., and Zabih, R., "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**, 1222–1239 (November 2001).

[5] Sun, J., Zheng, N.-N., and Shum, H.-Y., "Stereo matching using belief propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**, 787–800 (July 2003).

[6] Lang, M., Hornung, A., Wang, O., Poulakos, S., Smolic, A., and Gross, M., "Nonlinear disparity mapping for stereoscopic 3d," in [*Proc. ACM SIGGRAPH*], (2010).

[7] Chang, C.-H., Liang, C.-K., and Chuang, Y.-Y., "Content-aware display adaptation and interactive editing for stereoscopic images," *IEEE Transactions on Multimedia* **13**(4), 589–601 (2011).

[8] Europe, T., "Sky bans 2d to 3d conversions." Online (March 2010). http://www.tvbeurope.com/main-content/full/sky-bans-2d-to-3d-conversions.

[9] Saxena, A., Sun, M., and Ng, A. Y., "Make3d: Learning 3d scene structure from a single still image," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**, 824–840 (May 2009).

[10] Srivastava, S., Saxena, A., Theobalt, C., Thrun, S., and Ng, A. Y., "i23 - rapid interactive 3d reconstruction from a single image," in [*Proc. Vision, Modelling and Visualization (VMV)*], (2009).

[11] Lo, W.-Y., van Baar, J., Knaus, C., Zwicker, M., and Gross, M., "Stereoscopic 3d copy and paste," in [*Proc. ACM SIGGRAPH Asia*], (2010).

[12] Agarwala, A., Hetzmann, A., Salesin, D. H., and Seitz, S. M., "Keyframe-based tracking for rotoscoping and animation," in [*Proc. ACM SIGGRAPH*], **23**(3), 584–591 (2004).

[13] Bai, X. and Sapiro, G., "A geodesic framework for fast interactive image and video segmentation and matting," in [*Proc. IEEE International Conference on Computer Vision (ICCV)*], (2007).

[14] Levin, A., Lischinski, D., and Weiss, Y., "A closed-form solution to natural image matting," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**, 228–242 (February 2008).

[15] Russell, B. C. and Torralba, A., "Building a database of 3d scenes from user annotations," in [*Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], 2711–2718 (2009).

[16] Hengel, A. v. d., Dick, A., Thormählen, T., Ward, B., and Torr, P. H., "Videotrace: Rapid interactive scene modelling from video," *ACM Transactions on Graphics* **26** (July 2007).

[17] Varekamp, C. and Barenbrug, B., "Improved depth propagation for 2d to 3d video conversion using keyframes," in [*4th European Conference on Visual Media Production (IETCVMP)*], 1–7 (2007).

[18] Varekamp, C., Vandewalle, P., and de Putter, M., "Question interface for 3d picture creation on an autostereoscopic digital picture frame," in [*3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2009*], 1–4 (2009).

[19] Ernst, F., Wilinski, P., and van Overveld, K., "Dense structure-from-motion: An approach based on segment matching," in [*Lecture Notes in Computer Science (Proc. ECCV)*], **2351/2002**, 552–554, Springer (2002).

[20] Oliver, C. and Quegan, S., [*Understanding Synthetic Aperture Radar Images*], Artech-House (1998).

[21] Duda, R., Hart, P., and Stork, D., [*Pattern Classification*], 548–549, John Wiley & Sons, Inc., New York (2001).

[22] Varekamp, C., "Method and apparatus for removing false edges from a segmented image," (2004). International Patent Application WO2004/051573.

[23] Fua, P., "A parallel stereo algorithm that produces dense depth maps and preserves image features," *Machine Vision and Applications* **6**(1), 35–49 (1993).

[24] Bendito, E., Carmona, A., and Encinas, A., "Solving dirichlet and poisson problems on graphs by means of equilibrium measures," *European Journal of Combinatorics* **24**, 365–375 (2003).

[25] Boykov, Y. and Kolmogorov, V., "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**, 1124–1136 (September 2004).

[26] Scharstein, D. and Szeliski, R., "High-accuracy stereo depth maps using structured light," in [*Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*], **1**, 195–202 (2003).