Semi-automatic Stereo Extraction from Video Footage

Moshe Guttmann, Lior Wolf, Daniel Cohen-Or
The Blavatnik School of Computer Science
Tel Aviv University
wolf@cs.tau.ac.il, dcor@tau.ac.il

Abstract

We present a semi-automatic system that converts conventional video shots to stereoscopic video pairs. The system requires just a few user-scribbles in a sparse set of frames. The system combines a diffusion scheme, which takes into account the local saliency and the local motion at each video location, coupled with a classification scheme that assigns depth to image patches. The system tolerates both scene motion and camera motion. In typical shots, containing hundreds of frames, even in the face of significant motion, it is enough to mark scribbles on the first and last frames of the shot. Once marked, plausible stereo results are obtained in a matter of seconds, leading to a scalable video conversion system. Finally, we validate our results with ground truth stereo video.

1. Introduction

Stereoscopic media augments traditional video technologies with three dimensional perception, thus resulting in a more lifelike viewer experience. However, the adoption of such media is hindered by two main factors: the cumbersome nature of 3D (stereoscopic) displays, and the extra effort required to produce 3D content.

The production of filmed (as opposed to CG) 3D content requires either the capture of stereoscopic or multiple-camera content, or the conversion of 2D to 3D content in post-production. The former has several disadvantages including specialized equipment and production pipeline. Conversion technologies, on the other hand, can be employed to any existing conventional content, inducing the usage of old material. However, despite the conversion advantages, most 3D content today is created by specialized cameras, and not by conversion technologies. A notable exception is the June 2006 release of the movie "Superman Returns", which included 20 minutes of 3D images converted from the 2D original digital footage. It was recently declared that a company called "In-Three" may convert all six "Star Wars" movies to 3D, in a process that seems to be mostly-manual. Other players in the market include DDD® and Philips® - in our results section we compare to both.

The goal of this paper is to explore 2D to 3D conversion technologies which are efficient in terms of both human- and computer-time, and therefore cost-effective and accessible. A semi-automatic method is presented that requires a small amount of user interaction and produces good quality stereoscopic video. Typically, the user needs to mark only a few scribbles on the first and last frames of each video shot to produce a dense disparity map from which a second view is created (see Figure 1).

Our system propagates a sparse set of disparity values (user marked scribbles) across a video, by employing classifiers combined with solving a linear system of equations. The later encourages same disparity values for neighboring pixels, excluding edge separated pixels. Since depth values change over time, smoothness in time is applied judiciously based on motion information. Once the estimated disparities are obtained, a stereoscopic pair is generated by warping the original frame to Left/Right novel-views.

2. Related work

Fully automatic stereoscopic conversion requires the recovery of a three-dimensional structure in the visible part of the underlying scene. 3D structure extraction from a video sequence is a well explored problem. However, no method exists today that can reliably extract 3D information from unconstrained broadcast video.

Structure-from-motion (SfM) methods [4] have been proposed for the conversion problem, e.g., [16]. SfM may be more readily applied if a dense depth map is not required. Rotem *et al.* [20] and more recently Knorr and Sikora [7] create the additional stereo view by employing a planar mapping (via a homography matrix) of one of the existing frames. The availability of suitable frames pose limitations on the underlying shot, these include restrictions on the camera motion and limited, to none, scene motion.

Recently, statistical techniques have been used to infer depth from a single image [3, 6, 22, 23]. Potentially, such statistical techniques may, one day, recover depth accurately

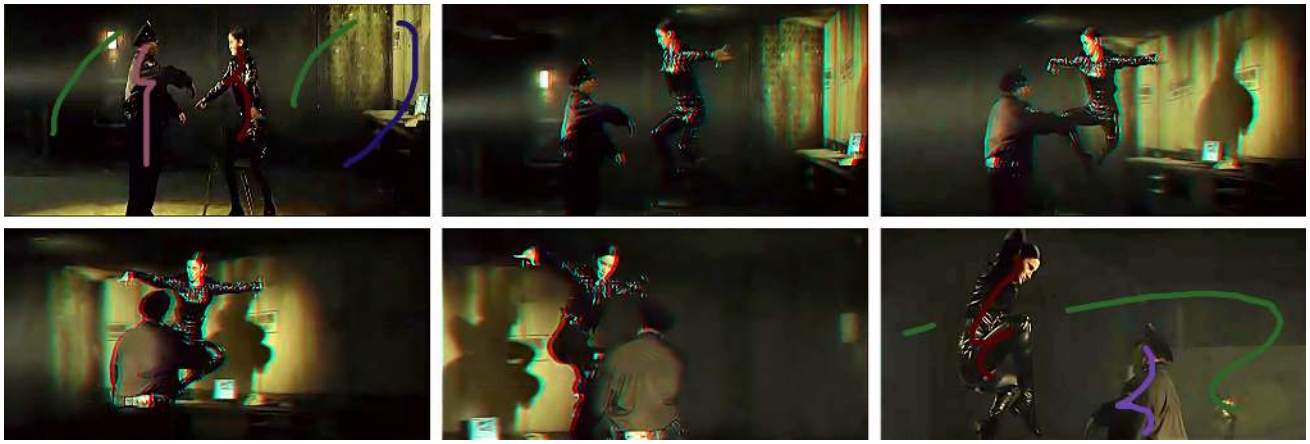


Figure 1. An illustration of the input and output of our conversion system. (from top-left clockwise) original frame from ”Matrix” marked by input scribbles; Output, Red/Cyan anaglyph frames form the shot; last frame marked by input scribbles. Note the significant motion of the camera and characters (the full stereo video is provided in the supplementary material).

enough to allow a fully-automatic video conversion system. However our experiments with the code published by the authors of [23] did not produce the needed results for frames taken from unconstrained video. Instead, we use classifiers that relate local appearance to disparity estimation within a global optimization scheme. To alleviate the unreliability of local classification, we employ a simple alternative to the statistical modeling of across-patch depth links, and incorporate into the optimization only high-confidence predictions.

Unable to perform fully-automatic conversion, existing commercial systems rely on user marked key frames. The system of Philips, at the time of submission in closed beta, seems to employ fully marked key frames and interpolates the depth within the frames. This interpolation may be based on motion estimation (see [25]) and seems to require perfect motion vectors (as in some of the experiments of [25]), or a large number of keyframes. The publicly available system of DDD$^{®}$ employs heuristics in order to approximate depth from live video. We have no information regarding their offline system.

Our system requires less manual labor than the system of Philips by employing scribbles. The use of scribbles was made popular by work by Boykov and Jolly [2], Levin *et al*. [10], Li *et al*. [14], and the refinement stage of Rother *et al*. [21]. We share with the work of [10] the basic set of weighted linear constraints, optimized in a least squares manner, that encourage neighboring pixels to have similar values. Such constrains have been in use for two decades [18, 24, 12]. Video segmentation [26, 13] is a related task to ours. An efficient method was recently presented in which geodesic distances are used in order to segment up to 100 frames based on few scribbles [1]. Recently, scribbles and classifiers were used jointly for image and video editing [11].

3. Algorithm

The basic units of broadcast video are frames, shots and scenes. A shot is a continuous block of frames taken from a single point of view, which may move continuously. A scene is a collection of consecutive shots taking place in one location and in a distinct time. While depth information is largely shared between the shots of each scene, and between all scenes taking place at the same location, in this work we only integrate information across one shot at a time.

Given an input shot, our system carries out the following stages: (1) User scribbles are marked on some of the frames to indicate desired disparity values. (2) The marked disparities are propagated on the frames on which they were drawn. (3) A classifier is trained for every disparity value marked by the user. (4) The classifier is applied to the entire shot, and high confidence predictions are recorded. (5) The disparity map of the entire shot is recovered in an optimization process which is constrained by the original scribbles and the high confidence predictions. Below, we address the design of each of these steps.

3.1. Manual marking

Depth and Disparity. Depth and perceived disparity are two related quantities. Assuming a simple model of two aligned sensors, and a perspective model (see Figure 2(a)), depth (Z) is related to disparity between the two perceived images (δ), through the formula

$$Z = \frac{fP}{\delta} \quad , \tag{1}$$

where P is the interpupillary distance, which measures the distance between the center of projections, of the two sen-

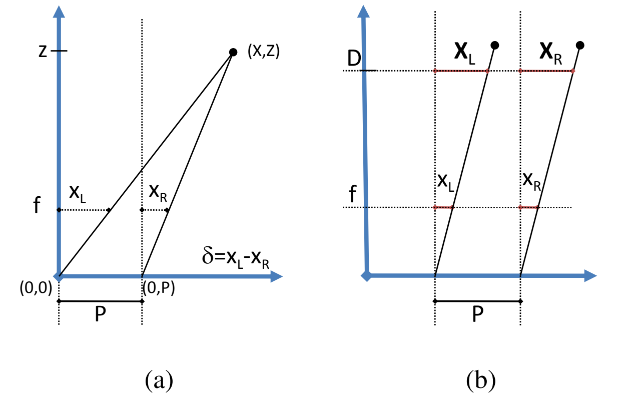


Figure 2. An illustration of stereo geometry according to a simple model. Eq. 1 stems from combining the two similar triangles equalities $\frac{Z}{f} = \frac{X}{X_L}$, $\frac{Z}{f} = \frac{X-P}{X_R}$, to obtain $X = \frac{X_L P}{\delta}$, where the symbols are as marked on (a), and then substituting it back to the first equality. The symbols for Eq 2 are illustrated in (b).

sors. The distance of the image plane from these centers is defined by f.

In stereo displays, the perceived disparity is a result of projecting two different images one per eye, these two images are planar image originating from the same screen. Assuming that the screen is parallel to the image plane of the two sensors, from Figure 2(b) we can observe that

$$\frac{D}{f} = \frac{X_L}{x_l} = \frac{X_R}{x_r} \quad , \tag{2}$$

where D is the distance of the screen, and X_L (resp. X_R) is the distance between the point on the screen closest to the left (right) sensor, and the screen point being projected to the left (right) eye. Note that the on-screen disparity (the distance between the two points on the screen) is given by $D_x = X_R + P - X_L$.

Combining Eq. 1 and Eq. 2, one obtains that in order to provide a depth perception equivalent to that of an object at a distance of Z units from the sensors, it is necessary to provide a screen disparity D_x which satisfies

$$Z = \frac{PD}{P - D_x} \quad . \tag{3}$$

Eq. 3 reveals that the interrelation between the amount of disparity in the video and the perceived depth, depends on the distance of the screen from the viewer, the size of the screen (same video on a larger display means larger on-screen disparity) and the distance between the pupils of the viewer. A stereo pair prepared for one set of viewing conditions may not be appropriate for another, and in order to provide good viewing experience one needs to be familiar with the conditions in which the 3D video is to be observed. Note that it is not enough to enlarge the screen by the ratio of distance-to-screen change, since P remains constant.

In our system we choose to estimate disparities directly without estimating depth first. As noted above, perceived depth is connected to disparity only through limited amount of parameters. Working directly with disparities has the numerical advantage of dealing with values that are not larger in magnitude than a few tens of pixels. Depth on the other hand, can range anywhere between zero and infinity and is inversely proportional the disparity, which quantifies the final observed change in the images.

User Interface. First, an automatic shot detection segments the video into consecutive shots by a simple, yet effective, color histogram change detection.

For each shot, the user is prompted to manually mark scribbles, each depicting a constant disparity value to all its pixels. The user selects a color for the scribble which encodes via the hue channel various disparity values. The default behavior, and the one which was applied in all reported experiments, is to mark scribbles on the first and last frames of each shot.

A natural question to ask is "how can the user know the correct disparity values to mark?". Surprisingly enough, accurate disparity values are not needed in order to create an immersive 3D viewing experience. As cognitive studies by Koenderink and his colleges [8] show, the human visual system is more tuned into depth order and precedence than to absolute depth.

3.2. Disparity propagating through optimization

The optimization problem we solve amounts to a sparse linear system, where the disparity values at each video location ($d(x, y, t)$) serve as the unknowns. We solve the entire shot at once in a least squares manner and in order to make the solution more efficient, we typically solve for a low resolution video, reduced by a factor of 4 in each dimension, and then upsample with a joint bilateral technique [9] exploiting the existence of high res frames.

There are four types of equations. First, we apply soft constraints that encourage the disparity at each location to be similar to the disparity of its spatial neighbors. Second, we encourage continuity over time with relation to the change in disparity expected from the local motion field. Third, we strongly encourage the system to adhere to the scribbles, and finally, we encourage the system to respect the results of the classifiers on anchor points where the confidence values of the classifier are high. As can be seen in Figure 3,5 , classification by itself is not reliable, but it considerably improves the final results.

Weighted least squares modeling All our constraints are soft and are expressed as weighted equations, which are optimized in a least squares manner. When the cost function

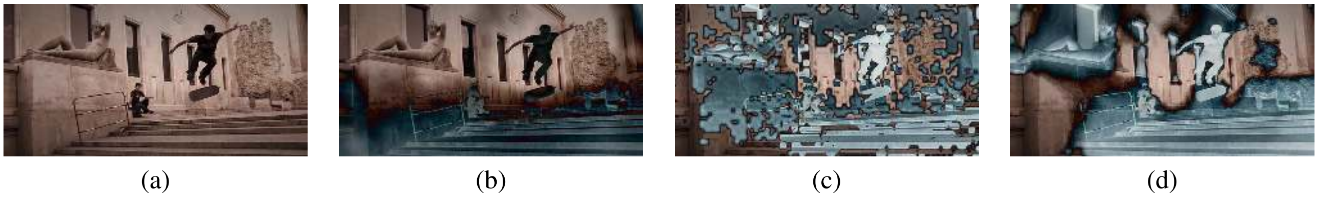


Figure 3. Disparity maps obtained by parts of our system (Hue values encode depth). (a) The original frame. (b) Disparity map obtained by propagating the marked disparities (scribbles). (c) The disparity map obtained by the classifiers alone. (d) The disparity map obtained by combining both smoothness based constraints and classification results as described in Section. 3.2.

is being optimized, error in an equation of weight w_1 is $(\frac{w_1}{w_2})^2$ times more significant than the error in an equation of weight w_2.

This way of modeling is similar to what is done in [27] for the task of video retargeting. It differs from the cost function employed by [10] in that it does not couple neighboring pixels together, and in that normalizing local correlations to a sum of 1 is not needed. However, care should be taken so that all weights are ranged reasonably.

Specifically, the cost function of [10] has elements of the form:

$$\left(U(r) - \sum_{s\in N(r)} w_{r,s}U(s)\right)^2 = \left(\sum_{s\in N(r)} w_{r,s}(U(r) - U(s))\right)^2 \quad (4)$$

where U are the unknown values, r is an index and $N(r)$ is the group of indices neighboring r (i.e. connected pixels), and the equality holds from the fact that the weights were normalized such that for all r $\sum_{s\in N(r)} w_{r,s} = 1$.

The cost function we employ, requires a simpler weighting scheme (no normalization required), and seems to produce comparable results in our experiments. It has analog elements that are of the form:

$$\sum_{s\in N(r)} w_{r,s}\left(U(r) - U(s)\right)^2 \quad , \quad (5)$$

where we set $w_{r,s} = w_r$ to depend only on r.

Spatial and time smoothness constraints The spatial constraints are weighted to reflect the local edge energy at each pixels'location. Pixels which lie on edges are less constrained to be similar to their neighboring pixels.

The edge energy weight is related by the L_2-Norm of the image gradient (images range [0..1]):

$$W_E = \sqrt{2 - (\frac{\partial}{\partial x}I)^2 + (\frac{\partial}{\partial y}I)^2} \quad (6)$$

The linear equations encouraging local smoothness are defined for every video location (coordinates x,y, and frame t) and are of the form:

$$\begin{aligned} c_1 W_E(x,y,t)\left(d(x,y,t) - d(x-1,y,t)\right) = 0 \\ c_1 W_E(x,y,t)\left(d(x,y,t) - d(x,y-1,t)\right) = 0 \end{aligned} \quad (7)$$

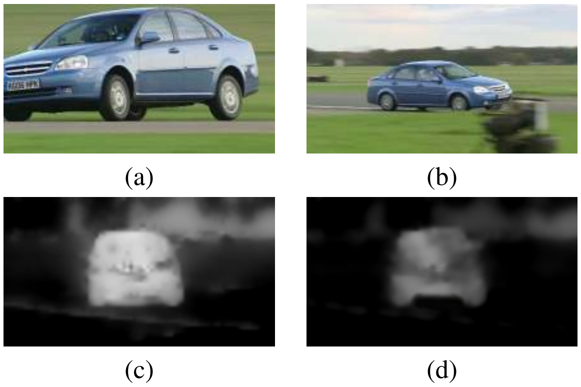


Figure 4. A shot with significant zoom out. (a) first frame, (b) last frame. (c) the disparity map without motion-based weighting (d) with weighting. Note that the car disparity is lowered (darker) i.e. it is farther away from the camera. See the full red/cyan anaglyph video provided in the supplementary material.

where $c_1 = 1$ is a constant in all of our experiments.

The time smoothness equation takes a similar form:

$$c_2 W_M\left(d(x,y,t) - d(x,y,t-1)\right) = 0 \quad (8)$$

The constant c_2 is set to 3 in all reported results. W_M is set to exploit common characteristics of interplay between depth and motion in the scene. This is a unique characteristic of the disparity estimation problem. In a general segmentation problem, for example, the property is constant over time, as it is oblivious to the distance of the object from the camera. Specifically, we notice that lateral motion seldom changes the depth of the objects, while objects moving in the vertical direction may change depth. Moreover, a zoom-in or a zoom-out motion results in motion in both directions, and is likely to result in a depth change.

To apply these observations in our system, we compute optical flow using the method of [17]. W_M is set to 1 if there is no motion or horizontal motion only; to 0.5 if the vertical motion is above a threshold of 2 pixels/frame, and to 0.2 if there is motion above this threshold in both directions. As can be seen in Figure 4, weighting the time constraints have a substantial effect on the results.

Scribble and classifier constraints. Another set of equations arise from the user scribbles. Taking into account inaccuracies that may be introduced by the user, we apply these as soft-constraints as well, albeit with a high weight, given by the constant $c_3 = 10$. Define $V(x, y, t)$ as the value of the scribble in a marked coordinate (x,y) on frame t (in our system, either the first frame or the last frame). The following equation is added for each marked triplet:

$$c_3 d(x, y, t) = c_3 V(x, y, t) \tag{9}$$

In our system a classifier (see Section. 3.3) is employed to predicted the depth value based on appearance. We define anchor points as points where the prediction is assigned a high confidence value. Let the prediction be T, and set the constant $c_4 = 0.5$, the added equations are of the form:

$$c_4 d(x, y, t) = c_4 T(x, y, t) \tag{10}$$

3.3. Classification

One of the design requirements we impose on our system is to have only a sparse set of user scribbles. The user marks only two frames out of hundreds of consecutive frames. This is much sparser, for example, than the results presented in [10], and somewhat sparser than what was presented in [1], nevertheless we chose to challenge our system with a more complex set of video shots containing significant motion.

To achieve this sparsity, we employ a Support Vectors Machine (SVM) classifier that is trained on the frames marked by the user and then applied to the entire video. The SVM is trained as follows: first the scribble data is propagated on the frames on which they were drawn. Then for each frame a multiclass classifier is trained (see below). The classifier is then applied to the other frame(s) and the confidence values are calibrated.

Training the classifiers The local appearance at an image point can be represented in many ways. We have experimented with using the gray values themselves, color histograms, SIFT [15], and SIFT+gray descriptors. As Figure 5 shows, it seems that SIFT+gray values is the most suitable for our problem.

The mapping between local image appearance and desired disparity values is a continuous regression problem, however, we obtain much better results treating it as a multiclass classification problem. The underlying reason might be that local appearance by itself does not tell us much on disparity, and therefore one cannot expect a patch which is similar in appearance to a blend of two patches to have their interpolated disparity value.

Given the user marked scribbles on the first and last frames of the shot. To obtain a training set, we first propagate the data of marked scribbles onto their frames by solving the spatial equations described in Section 3.2. We look at the unique set of values that the user scribbles are composed of[1], and assign each pixel in the frame to the class that is associated with the value which is closest to the propagated disparity value.

Then, for the first frame and separately for the last frame, we train SVM classifiers in a One-Vs-All scheme. The input space for the classifiers is the local appearance descriptor at each point, and the target space is the set of unique user-scribble values.

Calibrating the classifiers Classification based on appearance alone can be unreliable. We therefore only take into account high confidence classifications. In order to do so we need to transform the vector of One-Vs-All outputs (a vector of signed distances from each SVM hyperplane) to likelihood of correct classification. This is the classifier calibration problem in the multiclass SVM setting. Previous work include [19] for the binary case, which can be combined with methods such as coupling [5] for the multiclass case. Here we propose a simple binning method[28].

We learn, for each scribble color in the first frame a classifier, which has all pixels belonging to that color as positive examples, and all pixels belonging to the other colors as negative examples. To calibrate our classifier, we apply the classifiers learned on the first frame to the pixels of the last frame. Ground truth is provided to marked pixels on the last frame by assigning each scribble the color of the closest color of the first frames‘ scribbles.

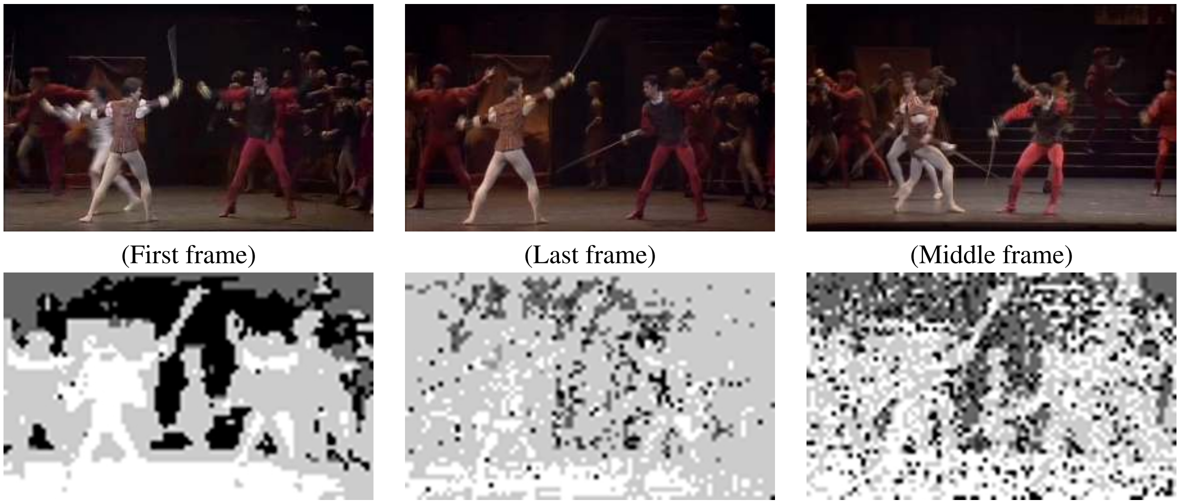
Then, for each classifier, we consider only those pixels in the last frame on which the classifier returned the highest positive value among all classifiers (with accordance to the One-Vs-All scheme). Based on the ground truth labels of those pixels we set a threshold for the classifier at hand such that the false positive rate is less than 5%.

This process is repeated when inverting the roles of the first and last frames. The two multiclass classifiers are then applied to the entire shot, and pixels, which are classified above the low-false-positive threshold, are marked as anchor points in the optimization process.

3.4. Synthesizing the output video

The recovered disparity is of a lower resolution. In order to increase its resolution while respecting the edges of the original image, we apply the Joint Bilateral Upsampling method[9]. Once the disparity is estimated, we create two novel views by shifting the left and right views each by half of the disparity value. A simple forward warping method is used to synthesize the two novel views.

[1] Typically this set is smaller than the number of scribbles. If this set is large, e.g., when the user marks points on the same scribble with different tones, clustering can be first applied.



(First frame) (Last frame) (Middle frame)

(SIFT+Gray values) (Color histogram) (SIFT)

Figure 5. Disparity estimation results, obtained on the middle frame, with either SIFT + gray values patches, color histograms, or SIFT as the local image descriptor. From our tests it seems that SIFT + gray level patches perform better than the alternatives implemented.

4. Results

We applied our method on a variety of broadcast videos. Figure 6,1 shows a collection of our results. As can be seen our method works well on sport videos with long shots, extreme scene and camera motion, on animation sequences, on documentation (music concert, ballet) and on feature films (”Matrix”, ”Batman - The Dark Knight” etc).

Note that although the user scribbles are sketchy, a real depth experience emerges, and the dynamic range of the disparity map is quite large (Table 4 User-survey).

We notice limitations mostly in cases where the scene changes within a shot. It is evident in a scene from “The Dark Knight”, we used for testing. This specific scene is composed out of many highly dynamic short shots. In one of the shots, a glass shattered reveals another room. The depth perception in the middle of the shot is lost (Fig 7).

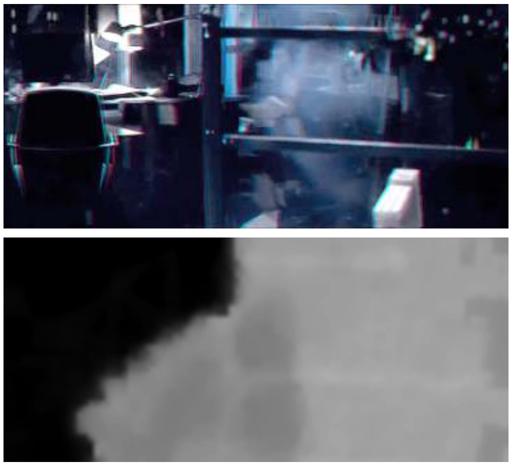


Figure 7. Depth Failure in ”Batman - The Dark Knight”. Due to an abrupt change of scenery during the shot, the disparity map lost its continuity and a flat map is obtained. (top) original frame. (bottom) disparity map.

Numerical results In order to obtain a quantitative performance measure we collect a set of 11 demo videos from the Philips® WowVx© project website. Each video contains a 2D input video along side with a high quality disparity map created manually or by using a 3D camera. Monochrome scribbles are drawn on the first and last frames of each shot. Disparity values are assigned to the scribbles based on the mean disparity value along the scribble. Each tested conversion system is then applied, producing right stereo view and estimate disparity.

We compare the following systems: (1) A simple scribble based, disparity propagation method marked as “Motion Scribble”. In this method the disparity is propagated from the scribbles to each frame individually, using a least squares system similar to ours. The location of each scribble is tracked from one frame to the next, and from the last frame to the first one. The contribution of each scribble is weighed by the distance from its original frame (first or last frame). (2) Our implementation of the system of [25]. This system uses the entire depth information of the first and the last frames. Depth information from the first frame is propagated until the middle of the shot. Depth information from the last frame is propagated in reverse through the second half of the shot. (3) Tri-Def® DDD© system, which works automatically on the raw video without using the scribble information. (4) Our method with the classification module turned off. (5) Our complete method. Table 4 presents the

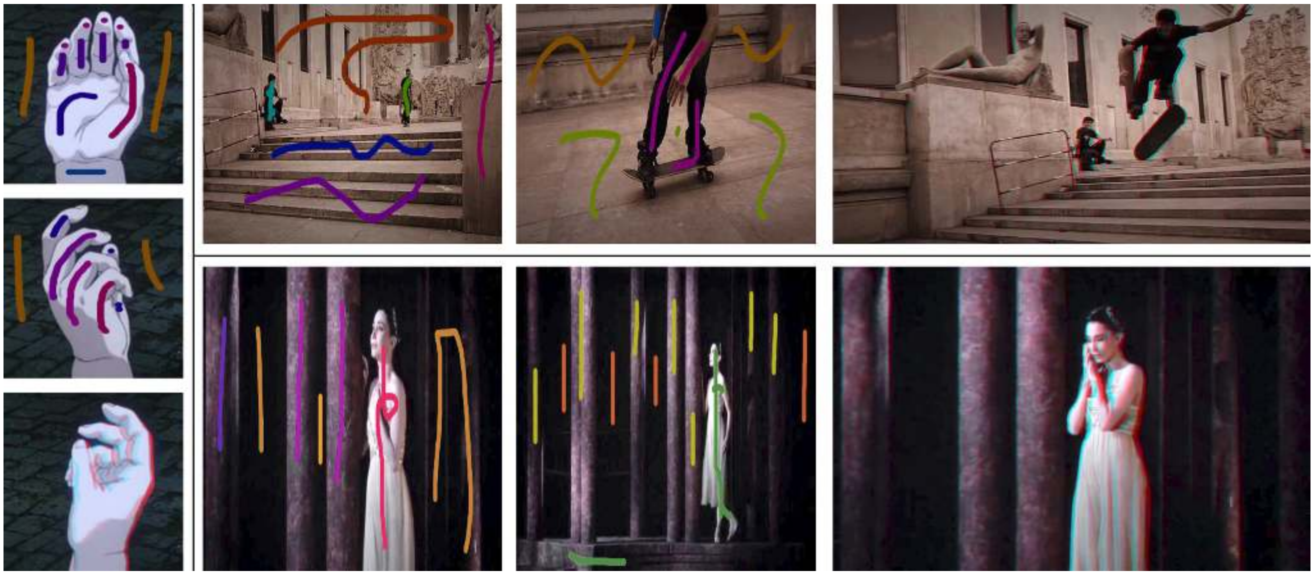


Figure 6. Depth experience emerges from video, as demonstrated on the middle frame of each shot. Best viewed with Red/Cyan glasses. (left) Disparity reconstruction for a rotating animated hand, (top right) Disparity reconstruction in a high motion skateboard sequence. (bottom right) "Romeo and Juliet" – although the scribbles are of discrete values, a dynamic range of disparities is obtained. The full stereo videos are provided in the supplementary material.

Algorithm	MSE disparity-map	User survey No.1	User survey No.2
Motion Scribble	2.48	0%	0%
DDD*	n/a	0%	8.7%
[25]	2.00	0%	18.4%
Ours w/o classification	1.59		
Ours w/ classification	1.41	22.7%	59.2%
Ground Truth	0	77.3	13.7%

Table 1. Left: Disparity prediction results. Right: User survey results. * The DDD system takes no depth information as input and produces no depth information.

benchmark's numerical results.

In addition, we conduct a user study on the same videos. 30 viewers were asked to rate the $3D$ clips of the various systems and to select a winner and a first runner-up. The last two columns in table 4 show the average number of times an algorithm was selected for the first and second place. To keep the results clean, and to reduce the viewer's labor, only our full system participates in the survey. Naturally, the ground-truth wins the user-study, however, our system is selected as the next best system in almost all cases. In some of the anaglyphs, users were consistently unable to tell the ground-truth from our system, implying that even when the disparity-map is imperfect the user cannot distinguish between results.

5. Conclusions

In this paper we make several contributions. These contributions are made more pronounced by the shortcomings of existing commercial systems, which we demonstrate on a newly developed video benchmark. (i) We propose a scribble-data propagation scheme that is more long ranged than previous contributions in any vision task and is able to deal with large amount of camera motion. (ii) We demonstrate the utility of classifiers in a video scribble based systems. (iii) We show how to incorporate anticipated depth changes into our framework.

References

[1] X. Bai and G. Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *ICCV*, 2007.

[2] Y. Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *ICCV*, 2001.

[3] F. Han and S.-C. Zhu. Bayesian reconstruction of 3d shapes and scenes from a single image. In *Int. Workshop on Higher-Level Knowledge in 3D Modeling*, 2003.

[4] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[5] T. Hastie and R. Tibshirani. Classification by pairwise coupling. In *NIPS*, 1997.

[6] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, 2005.

[7] S. Knorr and T. Sikora. An image-based rendering (ibr) approach for realistic stereo view synthesis of tv broadcast based on structure from motion. *ICIP*, 2007.

[8] J. J. Koenderink, A. J. van Doorn, A. M. Kappers, and J. T. Todd. Ambiguity and the 'mental eye' in pictorial relief. *Perception*, 30(4):431–448, 2001.

[9] J. Kopf, M. Cohen, D. Lischinski, and M. Uyttendaele. Joint bilateral upsampling. *SIGGRAPH*, 2007.

[10] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *SIGGRAPH*, 2004.

[11] Li, Y., Adelson, E., Agarwala, and A. Scribbleboost: Adding classification to edge-aware interpolation of local image and video adjustments. *Computer Graphics Forum*, 2008.

[12] S. Z. Li. On discontinuity-adaptive smoothness priors in computer vision. *PAMI*, 17:576–586, 1995.

[13] Y. Li, J. Sun, and H.-Y. Shum. Video object cut and paste. *SIGGRAPH*, 2005.

[14] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM Trans. Graph.*, 23(3):303–308, 2004.

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004.

[16] K. Moustakas, D. Tzovaras, and M. Strintzis. Stereoscopic video generation based on efficient layered structure and motion estimation from a monoscopic image sequence. *Circuits and Systems for Video Technology*, 15(8), 2005.

[17] A. S. Ogale and Y. Aloimonos. A roadmap to the integration of early visual modules. *IJCV*, 72(1):9–25, 2007.

[18] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *PAMI*, 12(7):629–639, 1990.

[19] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74, 1999.

[20] E. Rotem, K. Wolowelsky, and D. Pelz. Automatic video to stereoscopic video conversion. In *SPIE*, 2005.

[21] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, August 2004.

[22] A. Saxena, S. H. Chung, and A. Ng. Learning depth from single monocular images. In *NIPS*, 2006.

[23] A. Saxena, M. Sun, and A. Ng. Learning 3-d scene structure from a single still image. *ICCV*, 2007.

[24] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 2000.

[25] C. Varekamp and B. Barenbrug. Improved depth propagation for 2d to 3d video conversion using keyframes. *IETCVMP*, 2007.

[26] J. Wang, P. Bhat, R. A. Colburn, M. Agrawala, and M. F. Cohen. Interactive video cutout. In *SIGGRAPH*, 2005.

[27] L. Wolf, M. Guttmann, and D. Cohen-Or. Non-homogeneous content-driven video-retargeting. *ICCV*, 2007.

[28] B. Zadrozny and C. Elkan. Learning and making decisions when costs and probabilities are both unknown. In *SIGKDD*, 2001.