DOI: 10.2352/ISSN.2470-1173.2016.5.SDA-433

Towards Perceptually Coherent Depth Maps in 2D-to-3D Conversion

Nicole Brosch, Tanja Schausberger, Margrit Gelautz; Institute of Software Technology and Interactive Systems, Vienna University of Technology; Vienna, Austria

Abstract

We propose a semi-automatic 2D-to-3D conversion algorithm that is embedded in an efficient optimization framework, i.e., cost volume filtering, which assigns pixels to depth values initialized by user-given scribbles. The proposed algorithm is capable of capturing depth changes of objects that move towards or farther away from the camera. We achieve this by determining a rough depth order between objects in each frame, according to the motion observed in the video, and incorporate this depth order into the depth interpolation process. In contrast to previous publications, our algorithm focuses on avoiding conflicts between the generated depth maps and monocular depth cues that are present in the video, i.e., motion-caused occlusions, and thus takes a step towards the generation of perceptually coherent depth maps. We demonstrate the capabilities of our proposed algorithm on synthetic and recorded video data and by comparison with depth ground truth. Experimental evaluations show that we obtain temporally and perceptually coherent 2D-to-3D conversions in which temporal and spatial edges coincide with edges in the corresponding input video. We achieve competitive 2D-to-3D conversion results. Our proposed depth interpolation can clearly improve the conversion results for videos that contain objects which exhibit motion in depth, compared to commonly performed naive depth interpolation techniques.

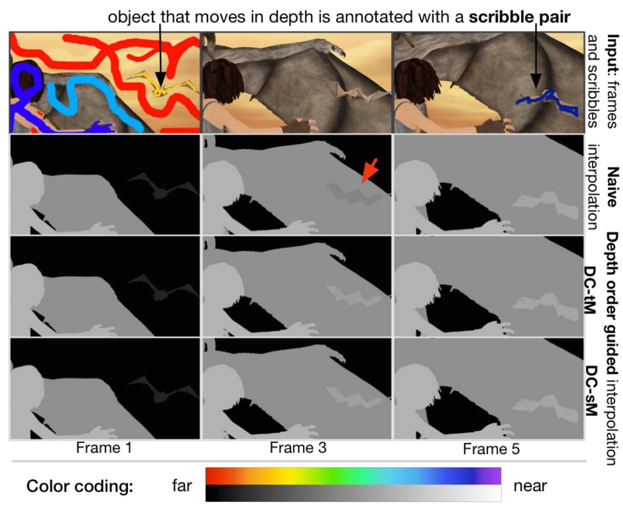

Figure 1. *Temporal depth change models. 1st row: Input video with user-provided scribbles. The small dragon is annotated with a yellow scribble in the first and a blue scribble in the last frame to indicate a temporal depth change. Scribble hues encode depth. 2nd-4th row: Corresponding 2D-to-3D conversions results that were obtained by using different temporal depth interpolation techniques, i.e., a naive linear depth interpolation (2nd row) and our proposed interpolation (3rd and 4th row) that takes motion-caused occlusions into account. The former is perceptually incoherent (red arrow).*

Introduction

Semi-automatic 2D-to-3D conversions can cost-efficiently convert existing monoscopic (2D) videos to stereoscopic (3D) videos (e.g., [1, 2, 3, 4, 5, 6]). The key idea of such algorithms is to propagate sparse user-given depth information in key frames to the remaining pixels in a video. The resulting depth video can provide the basis for applications such as the geometry-consistent generation of novel views or mixed reality scenarios. Contrary to stereo generated 3D content (e.g., [15]) or measured depth, semi-automatic 2D-to-3D conversion does not require special hardware (e.g., a stereo camera or a time-of-flight sensor) and the need for 3D content has not to be known before capturing a video.

This paper presents a semi-automatic 2D-to-3D conversion algorithm that generates depth maps for videos based on comfortable scribble input. Ideally, the generated depth maps are (1) spatio-temporally coherent and (2) result in a plausible 3D impression. The proposed 2D-to-3D conversion algorithm is embedded in an efficient optimization framework, i.e., cost volume filtering (CVF) [7], that has shown to satisfy condition (1) in the application of video object-segmentation [8]. Our 2D-to-3D conversion algorithm is based on the interactive video object-segmentation in [8]. In some similarity to [8], we use CVF to assign pixels to depth values that were initialized by multiple user-given scribbles (see Figure 1). Scribble pairs that are located in the first and the last frame of a video are used to indicate a depth change. In this context, the main contribution of our algorithm is the introduction of depth change models for these scribble pairs. The goal of these models is not only to capture temporal depth changes over time, but also to generate perceptually coherent depth maps. In other words, when capturing the depth changes of objects that are moving towards or farther away from the camera, our algorithm takes care that the depth values assigned to these moving objects harmonize with those of nearby objects. More precisely, the interpolation of the temporally changing depth values is performed in accordance with observed occlusions that were caused by nearby objects. We first determine a rough depth order in each frame according to motion calculated for the given video. Subsequently, this depth order is incorporated into the depth interpolation process to generate temporally and perceptually coherent depth maps. Thus, by addressing the problem of perceptual coherence in the context of motion-caused occlusions, our 2D-to-3D conversion approach takes a step towards the generation of perceptually coherent depth maps.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.5.SDA-433

While the importance of enabling temporal depth changes was already stressed by a few existing semi-automatic conversion approaches (e.g., [1, 2, 3, 4, 6]), we are not aware of any semi-automatic 2D-to-3D conversion approach that as well considers the problem of generating perceptually coherent depth maps. Relatedly, Liao et al.'s [5] 2D-to-3D conversion approach combines user-given depth information with automatically estimated depth information. This approach ensures perceptual coherence in reference to a single, previously extracted moving foreground object, but in principle could be extended to consider multiple moving objects similar to our approach. Moreover, they propagate depth values with a global optimization that incorporates inequality constraints between selected pairs of neighboring pixels. These constraints restrict the depth values of all pixels that are adjacent to the foreground object to be lower (that is, further in the background) than the depth value of the foreground object. Their propagation scheme further analyzes motion information to detect expanding or shrinking objects and infers a depth change depending on the object size. Contrary to [5], we aim for depth maps that are perceptually coherent with respect to multiple moving objects. We achieve this by inferring a rough depth order [12, 13] between multiple objects which is subsequently combined with the user-given depth values.

Experimental results and evaluations with ground truth data demonstrate that we obtain temporally coherent 2D-to-3D conversions in which depth edges coincide with edges in the corresponding input video. Concerning the depth change models, our temporal depth interpolation can clearly improve the conversion results for videos that contain objects which exhibit motion in depth, compared to commonly performed naive depth interpolations. We further show that our proposed algorithm is competitive when being compared to related 2D-to-3D conversion approaches.

The rest of the paper is organized as follows. The next section describes the proposed semi-automatic 2D-to-3D conversion algorithm. Then, we perform a systematic evaluation that compares different versions our algorithm, including versions that apply our depth order guided interpolation and versions that apply a naive interpolation of depth values over time. We further evaluate our conversion results (with reference data) and compare it to related 2D-to-3D conversion algorithms [1, 3, 4]. Finally, we conclude our discussion.

Algorithm Description

Figure 2 gives an overview of the proposed semi-automatic 2D-to-3D conversion algorithm's main components. Following, we first discuss the basic conversion algorithm, which includes the generation of a cost volume from user-provided scribbles, cost volume filtering and the derivation of depth maps from the filtered cost volume (i.e., MS, STC, CON, cost volume filtering and depth assignment in Figure 2). Subsequently, we present our main contribution, i.e., depth change models and their performed depth order guided interpolation (i.e., DC in Figure 2). Their purpose is not only to obtain temporally coherent, but also perceptually coherent depth maps that smoothly capture motions in depth.

Basic 2D-to-3D conversion algorithm

In this section, we briefly review the used optimization framework [7] in context of a related application, i.e., interactive video object-segmentation [8]. We further discuss its usage in our application of semi-automatic 2D-to-3D conversion, which typically requires *multiple scribble labels* (i.e., depth values that are large in the foreground and low in the background). The increased number of labels (compared to mere foreground-background segmentation) might come with a larger ambiguity between their color models, which is addressed by an additional *spatio-temporal closeness constraint* and a *3D connectivity constraint*. We investigate an extension of the CVF framework [7] by performing *motion guided filtering* to improve the temporal coherence in our conversion results. We set the 2D-to-3D conversion algorithm further apart from video object-segmentation algorithms by introducing spatially smooth depth changes within objects with a simple *depth blending* approach.

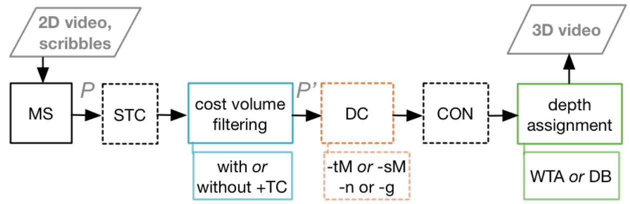

Figure 2. *Overview of semi-automatic 2D-to-3D conversion algorithm. Generation of cost volume P from multiple scribbles (MS). Spatio-temporal closeness constraint (STC). Cost volume filtering without or with motion guided filtering (+TC) to obtain P'. Depth change models (DC) that correct naive interpolations (-n) by guided interpolations (-g). They can alternatively be applied with respect to time (-tM) or object size and motion (-sM). 3D connectivity constraint (CON). Final depth assignment using a winner-takes-all (WTA) or a depth blending (DB) scheme. Dashed components can be disabled.*

Brief description of interactive segmentation via CVF [7, 8]. The underlying interactive video object-segmentation algorithm [8] partitions a video into foreground F and background B pixels. After a user has drawn scribbles in frames to indicate that the marked pixels belong to F or B, a fast optimization based on spatio-temporal CVF [7] is triggered. Based on the user-provided scribble input a foreground H_f and a background color model H_b (i.e., color histograms) are built from the marked pixels. H_f and H_b are then used in the probability computation. The CVF-based optimization first generates a cost volume $P(x,y,t)$, which contains the probabilities $p_i \in [0,1]$ that a pixel $i = (x,y,t)$ belongs to F (or B, i.e., $1-p_i$). Subsequently, a smoothness assumption propagates probabilities to neighboring pixels that are similar in terms of color. This is implemented efficiently by smoothing P spatio-temporally by using an edge-preserving filtering technique [11]. Finally, each pixel is assigned to either F or B using a winner-takes-all approach (WTA), i.e., according to its maximal probability in the filtered cost volume P'.

Scribble matching and CVF with multiple scribbles (MS). Similarly to the object-segmentation algorithm [8], we use a scribble-based user interface, which, however, supports multiple scribble labels (i.e., multiple depth values as opposed to only F and B in [8]). Here, each user-provided scribble S_l indicates a single depth value D_l. D_l is large for scribbles that are in the foreground and lower for scribbles that are in the background. Since we aim to capture

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.5.SDA-433

changes in depth, users can indicate them by performing appropriate scribble annotations in the first and last frame (e.g., Figure 1, annotation of small dragon). Based on the idea that scribbles which are contained in the same spatio-temporal segment belong to the same object, we use a motion segmentation (i.e., from [9] without depth) and color comparisons to group and match scribbles. First, we group those scribbles within the same frame and segment that indicate the same depth value. Subsequently, we match scribbles between the first and last frame that are located in the same spatio-temporal segment. The resulting scribble pairs can indicate different depth values of one and the same object at different points in time. Grouped scribbles with the same depth value (e.g., D_l) and matched scribble pairs that might have a different depth value in the first (e.g., $D_{l,first}$) and in the last frame (e.g., $D_{l,last}$) are from now on considered and processed as a single scribble S_l, except when explicitly mentioned otherwise.

The extension of [8] to multiple labels is straightforward. Analogue to [7, 8], we generate a cost volume $P(x,y,t,l)$, which contains the probabilities $p_{i,l}$ for a pixel i to belong to each scribble S_l. $p_{i,l}$ is computed by comparing i's color to each scribble's color model and a color model that is generated from all remaining scribbles. In Figure 2, P is the result of MS and has to be processed further to obtain a depth map. The smoothness assumption is incorporated analogously to [7, 8], by smoothing [11] each cost volume slice, i.e., each $P(x,y,t,.)$ for a fixed l. Finally, each pixel can be assigned to the depth of the scribble with the maximal probability in the filtered cost volume P' (i.e., WTA). The current result equals an interactive multi-label segmentation $\mathscr{R}$ in which each segment R_l is additionally assigned the depth D_l of its scribble S_l.

Spatio-temporal closeness constraint (STC). With scribble-based annotations users typically want to indicate local assignments rather than global ones. This may be complicated by color ambiguities between objects at different depth values. Thus, we further constrain a scribble's influence on each pixel in the video by their spatio-temporal closeness. A confidence weight $p_{close,l} \in]0,1]$ is computed from a distance transform [10] of each scribble S_l. It is applied to the cost volume entries from MS before filtering the probabilities (i.e., to $p_{i,l}$). While the computation of $p_{close,l}$ is straightforward for a single frame that contains user-provided scribbles (e.g., first frame), remaining frames require the additional step of scribble tracking. This is simply done by following pre-determined optical flow (OF) vectors at the pixels that are marked by a scribble throughout the video. The closeness weights $p_{close,l}$ are then computed based on the tracked scribbles S_l. The result of STC is an updated P that can be smoothed and used to obtain a different $\mathscr{R}$ and depth map than when only using MS (Figure 2).

Motion guided filtering (+TC). As in [7, 8], our smoothness assumption is implemented with an edge-preserving filtering technique [11]. It smoothes the cost volume P (that was generated in MS and updated in STC, Figure 2) to propagate probabilities to neighboring pixels that are similar in terms of color. This smoothing is performed within a spatio-temporal filter window of fixed size (i.e., r_s and r_t) and, thus, accounts for motion between frames only implicitly. While this local approximation of the smoothness assumption is sufficient for most dynamic scenes, it is less robust for scenes that contain fast moving objects (e.g., Figure 5, *b)*, *red arrows*, motion up to approximately 100 pixels). Specifically, if the movement of an object exceeds the size of the filter window, the filtering of the object is performed independently for each frame. We address this issue by incorporating motion in the filtering process by allowing the filter window to adjust its spatial position between frames according to the motion in a video. To this end, we modify the sliding box filter that is used in the applied filtering technique [11]. In motion guided filtering it is applied by following motion vectors from frame to frame [16]. Instead of computing the temporal average across constant spatial pixel positions in different frames, the temporal average is built from corresponding pixels in different frames that are connected by motion vectors (e.g., Figure 3). Motion guided filtering can be applied instead of the original filtering operation that is used in the CVF-framework (Figure 2).

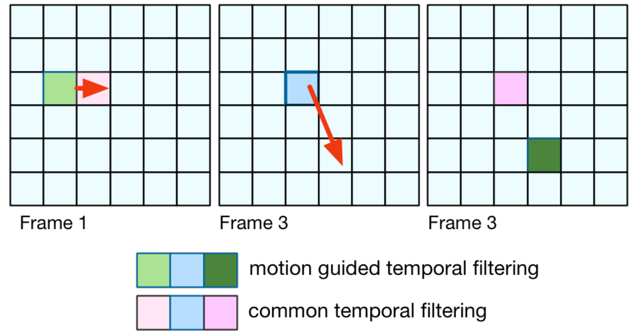

Figure 3. *Motion guided filtering. The temporal box filter in [11] deviates from its motion guided version. While the former filters straight through a video, the latter filters along motion vectors (red arrows).*

3D connectivity constraint (CON). We enforce that depth assignments and their corresponding scribbles are connected in 3D to avoid unwanted changes in areas not connected to the local user input. This connectivity constraint operates on both P' and its current WTA depth map. Essentially, it reduces filtered probabilities $p'_{i,l}$ in P' if they result in a WTA depth map in which i's depth (e.g., D_l from S_l) is not connected to its corresponding scribble (e.g., S_l). In this context, we consider a pixel i connected to a particular scribble (e.g., S_l), if the frame contains a *connectivity-path* [15] that connects all pixels, including i, with the same depth as this scribble (e.g., D_l from S_l) and the scribble (e.g., S_l or tracked pixels from S_l). Such a connectivity-path consists of pixels that either (i) are assigned the same depth (e.g., D_l from S_l) or (ii) are assigned to a larger depth (e.g., background D_l from S_l is occluded by foreground objects with larger D_l). We implement the 3D connectivity constraint by consecutively applying it on each scribble S_l in descending order of scribble depth values, i.e., starting in the foreground. We identify pixels that violate the constraint using a flood fill algorithm that is initialized at pixels that are cov-

DOI: 10.2352/ISSN.2470-1173.2016.5.SDA-433

ered by S_l (or its tracked pixels). We then reduce the probability $p'_{i,l}$ of the identified pixels i by a small factor. The result of CON is an updated P', which can be processed further (Figure 2). When re-computing the WTA depth map from the updated probabilities in P', pixels are assigned to a different depth than before.

Depth blending (DB). Since objects might be rounded or slanted, they can exhibit multiple depth values that blend into each other. We support this case by (optionally) substituting the WTA scheme with a simple depth blending scheme (Figure 2). Specifically, the final depth for a pixel i is a weighted average that is determined from the depth values of all scribbles according to their probabilities $p'_{i,l}$. In this context, depth values of scribbles with high probabilities can significantly influence the final depth of a pixel, while depth values of scribbles with low probabilities hardly contribute to its final depth. Their influence on the final depth can be further restricted by considering only the depth values with the n highest probabilities for each pixel.

Depth guided interpolation algorithm

In order to capture motion in depth in the 2D-to-3D conversion results, the depth change of scribble pairs that are located in the first and the last frame (e.g., Figure 1, *yellow scribble* and *dark blue scribble*) has to be defined. Specifically, for each scribble pair a *depth change model* (DC) has to specify an interpolation between its depth given in the first and its depth given the last frame of the video. Naively, such models could interpolate these depth values linearly. While this solution might work in some cases, it has a major drawback – the resulting depth map might be perceptually incoherent (e.g., Figure 1, *second row*). We address this issue by performing the temporal depth interpolation in accordance with observed occlusions that were caused by nearby moving objects. The underlying basic idea is that, if an object A (e.g., small dragon in Figure 1) moves in front of another object B (e.g., large dragon's wing in Figure 1) in frame t – i.e., if A occludes B in frame t – we can conclude that in t A has a larger depth than B. Thus, when interpolating A's depth over time it should not fall below B's depth in frame t, i.e., should be *restricted* by B's depth in frame t. We implement this idea by, first, determining a *rough depth order* in each frame according to the motion observed in the given video. Subsequently, this depth order is used to define *depth restrictions*. Finally, we perform a *depth order guided interpolation* according to these restrictions. These steps rely on the filtered cost volume, i.e., P', and intermediate results that can be derived from it (Figure 2).

Rough depth order. We implement this idea for each frame by, first, collecting pairwise depth order cues for segments that belong to a scribble pair. In this context, we use segments that are derived by interpreting the current 2D-to-3D conversion result as an interactive multi-label segmentation, i.e., $\mathscr{R}$ (e.g., Figure 4, *b)*). Occlusions between pixels of these segments are detected by checking the consistency of the forward OF and the backward OF [14]. As suggested by [13], in case of conflicting pairwise occlusion information for a segment, only the more frequent depth order cue is used. For each frame, these depth cues are recorded in a *cycle-free directed acyclic graph* (DAG) [12, 13] (e.g., Figure 4, *c)*). The

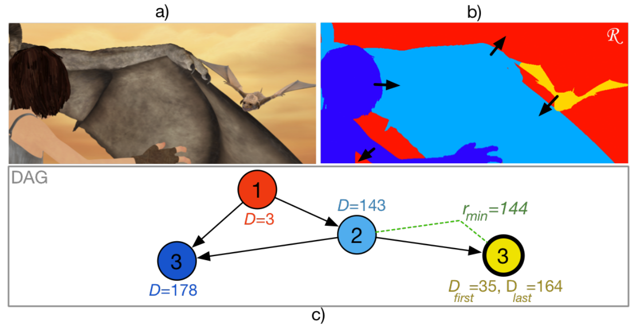

Figure 4. *Depth restriction example. a) Intermediate input frame from example in Figure 1. b) Corresponding multi-label segmentation $\mathscr{R}$: Segment colors correspond to scribble colors in the first frame in Figure 1. Black arrows visualize depth order cues, i.e., segment at the arrow's shaft occludes segment at its pointy end. c) DAG: Node colors correspond to segment colors. Directed edges between nodes constitute a occlusion relation, i.e., parent node is occluded by child node. The numbers in the nodes are their depth level λ and D their user-given depths. The minimum restriction $r_{min} = 144$ of the yellow node can be determined based on the light blue node.*

nodes in this graph correspond to segments $R_l \in \mathscr{R}$, while direct edges between the nodes record an occlusion relation between them. The hierarchy level (or *depth level*) λ of the nodes captures the global relative depth order of the segments within a frame [12, 13]. In this graph nodes with a large λ are in front of nodes with a low λ.

Depth restrictions. Given this DAG, its depth levels λ and the user-provided depth annotations, we define depth restrictions that guide the temporal depth interpolation. These depth restrictions are a range $[r_{min}(R_l,t), r_{max}(R_l,t)]$ that defines a minimal r_{min} and maximal r_{max} allowed depth value that can be taken on by a specific scribble pair S_l in a specific frame t. As stated above, we assume that only scribble pairs indicate a change in depth over time. Scribbles without annotations in both the first and the last frame are associated with their fixed user-assigned depth values within the entire video. These fixed depth values can be exploited when deriving *depth restrictions* for the scribble pairs as following:

$$r_{min}(R_l,t) = \max_{R_k \in parent(R_l,t)} (D_k + |\lambda_{l,t} - \lambda_{k,t}|), \qquad (1)$$

$$r_{max}(R_l,t) = \min_{R_k \in child(R_l,t)} (D_k - |\lambda_{l,t} - \lambda_{k,t}|). \qquad (2)$$

Here, R_l and R_k are two segments in the currently observed frame t. The segment R_k is invoked by a scribble S_k which has a user-assigned fixed depth D_k. S_l is a scribble pair. In Eq. (1), the minimum restriction for R_l is determined by selecting the maximum depth of all its parent nodes, i.e., segments that are behind R_l (e.g., Figure 4, *d)*, *yellow node* derives r_{min} from *light blue node*). Since we know that R_l has a larger depth than R_k, D_k is additionally increased by the difference in depth levels of the current and the found node, i.e., $|\lambda_{l,t} - \lambda_{k,t}|$. The maximum restriction $r_{max}(R_l,t)$ for R_l in t is determined analogously. To support scenes that contain scribble pairs that occlude each other, we traverse the DAG in ascending order of depth level when determining $r_{min}(R_l,t)$ and $r_{max}(R_l,t)$. This means, the restrictions of parent nodes (background) are determined before the re-

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.5.SDA-433

strictions of their child nodes (foreground). In the case of segments R_l and R_k that correspond to scribble pairs and occlude each other, $r_{max}(R_l,t)$ and $r_{min}(R_k,t)$ are at first calculated momentarily and are updated as soon as the depth assignment for R_l is fixed (after depth interpolation). Hence, the proposed algorithm as well supports videos with annotations that solely consist of scribble pairs.

Depth order guided interpolation. The final depth of each scribble pair can be determined by interpolating their user-given depth values over time while taking into account the restrictions defined above. The starting point of our depth order guided interpolation is a naive, linear interpolation (e.g., Figure 1, *second row*, Figure 2, *-n*) that is performed with respect to time (i.e., *DC-tM*) or with respect to segment size and motion (i.e., *DC-sM*). In particular, the latter interpolation considers irregular moving objects by changing the depth in conjunction with the object size in different frames. Furthermore, a depth change is only performed if changes in the segment size (i.e., height in pixel) and vertical movement [1] are observed. To begin, the naive interpolation is performed between two fixed data points, i.e., the user-given depth values in the first and the last frame, and might produce perceptual incoherencies. Then, we perform a recursive depth verification and adjustment step according to the depth restrictions for each frame in order to remove these incoherencies (e.g., Figure 1, depth order guided interpolations, Figure 2, *-g*). We compare the current depth of each scribble pair (e.g., S_l) with the upper and lower bounds that are provided by its depth restrictions (e.g., $[r_{min}(R_l,t), r_{max}(R_l,t)]$). If the current depth violates these restrictions, it is adjusted to the closest depth within the allowed depth range. This adjustment adds another depth data point to the interpolation and triggers an accordant update (i.e., re-computation) of the depth values of preceding and following frames. These re-computated depth values are again recursively verified and, if necessary, adjusted until only valid depth values are used. Thus, after this procedure the depth change model of each scribble pair specifies a depth that is consistent with the previously extracted depth restrictions in each frame.

Experimental Results

We perform our experiments on video data from [18] that was provided with ground truth (GT) depth maps and motion information. We also perform experiments using video data that was provided with disparity maps (including [19]). These reference solutions contain (GT) disparities, which are inverse proportional to depth. In order to unify these two types of depth information and our specified format for input depth (i.e., low values in the background, large values in the foreground), given depth values are inverted. To enable the comparison of our 2D-to-3D conversion result with these reference solutions, we perform the conversion based on depth values at user-provided scribble positions (i.e., mean depth of all pixels marked by a scribble). Each pair of conversion result and corresponding reference solution is normalized by their joint maximal depth. Subsequently, our obtained conversion results are compared with the reference solutions, where we list the mean squared error (MSE) averaged over all frames of a video. In this manner, we perform a systematic evaluation that compares different versions of our algorithm. These evaluations also include the investigation of the algorithms sensitivity to the quality of the used motion information. Furthermore, we compare our obtained results with related 2D-to-3D conversion algorithms, including our implementation of Guttmann et al.'s algorithm [1], which belongs to the first works that explore semi-automatic 2D-to-3D conversion from user-provided scribbles. The comparison additionally considers more recent 2D-to-3D conversion algorithms, i.e., Phan et al.'s [3] algorithm and Ivancsics et al.'s [4] implementation of [2]. The implementation of Phan et al.'s [3] algorithm was provided by the authors and is applied volumetrically on the entire video. In some similarity to our proposed algorithm, these algorithms incorporate segmentation information into the conversion process. Our 2D-to-3D conversion algorithm is evaluated with the following constant cost volume filter parameters to obtain depth maps for monoscopic videos: $r_s = 11$, $r_t = 2$, $\varepsilon = 0.0016$. All shown results that use DB are obtained with $n = 2$.

Comparison of different algorithm versions. Table 1 provides quantitative evaluation results for our 2D-to-3D conversion algorithm, in which different components are en- or disabled. More precisely, we compare following versions of our algorithm: *MS* is only based on color and does not capture temporal depth changes. *STC* additionally applies the spatio-temporal closeness constraint. *CON* builds upon STC and also applies the 3D connectivity constraint. *DC-tM* and *-sM* further capture depth changes performing the interpolation with respect to time and with respect to segment size and motion, respectively. The depth change is captured with a naive (*-n*) and our depth order guided (*-g*) interpolation. Finally, we investigate versions of our algorithm that apply motion guided filtering (*+TC*), as opposed to common filtering with [11]. Since the evaluation results when using WTA behave analogously to those when using DB, Table 1 only lists the former. The evaluation is performed on videos from [18] that were provided with GT depth maps. Table 1 shows that both, the spatial closeness constraint and the 3D connectivity constraint have a positive impact on the obtained 2D-to-3D conversion results. Specifically, when comparing the errors of MS with those of STC and the errors of STC with those of CON, the additional constraint decreases the errors for nearly all tested videos. As shown in Figure 5, *d)*, these components reduce unwanted depth assignments that are located spatially far away from their corresponding scribble input. When comparing the evaluation results of versions that are using common filtering and *+TC*, the latter on average reduces the measured errors by approximately 16 percent. Large improvements are observed for videos with fast motion, e.g. *Ambush2* in Figure 5, *a)* and *b)* (up to approximately 100 pixels between frames). This indicates that our motion guided filtering is important for videos with fast moving objects, while for videos with small motion common filtering is sufficient. Table 1 further demonstrates that our depth interpolation can significantly improve the conversion results for videos that contain objects which exhibit motion in depth (e.g., *Shaman3* and *Sleeping1* contain a camera-zoom), compared to versions that do not capture temporal depth changes.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.5.SDA-433

Table 1. Comparison of different versions of the proposed 2D-to-3D conversion algorithm (WTA). The table lists the MSE$\times 100$ of the depth values averaged over all frames when applying our algorithm with GT OF (*top*) and estimated OF [17] (*bottom*).

$MSE \times 100$	**GT OF**							
WTA	*Alley1*	*Ambush2*	*Ambush5*	*Ambush7*	*Shaman2*	*Shaman3*	*Sleeping1*	*Temple3*
MS	0.06	1.19	1.45	0.99	0.79	1.91	3.65	0.28
MS +TC	0.06	0.27	1.44	0.99	0.79	1.91	3.66	0.15
STC	0.05	1.16	0.65	0.53	0.57	1.90	3.62	0.27
STC +TC	0.05	0.24	0.65	0.53	0.56	1.90	3.62	0.21
CON	0.04	1.16	0.63	0.54	0.42	1.96	3.61	0.27
CON +TC	0.04	0.24	0.65	0.54	0.41	1.96	3.61	0.21
DC-tM -n	0.04	1.16	0.64	0.49	0.39	0.18	0.57	0.27
DC-tM -n +TC	0.04	0.23	0.63	0.49	0.39	0.18	0.58	0.21
DC-sM -n	0.04	1.16	0.64	0.55	0.42	0.23	0.64	0.27
DC-sM -n +TC	0.04	0.23	0.63	0.55	0.42	0.23	0.65	0.21
DC-tM -g	0.04	1.16	0.64	0.47	0.41	0.41	0.48	0.27
DC-tM -g +TC	0.04	0.23	0.64	0.48	0.41	0.40	0.48	0.21
DC-sM -g	0.04	1.16	0.64	0.47	0.42	0.51	0.56	0.27
DC-sM -g +TC	0.04	0.23	0.64	0.48	0.42	0.49	0.57	0.21
$MSE \times 100$	**estimated OF** [17]							
WTA	*Alley1*	*Ambush2*	*Ambush5*	*Ambush7*	*Shaman2*	*Shaman3*	*Sleeping1*	*Temple3*
MS	0.07	1.22	1.46	0.96	0.78	1.91	3.65	0.28
MS +TC	0.07	0.31	1.47	0.96	0.79	1.91	3.66	0.15
STC	0.05	1.30	0.69	0.50	0.55	1.90	3.61	0.27
STC +TC	0.03	0.29	0.68	0.50	0.55	1.90	3.61	0.14
CON	0.04	1.30	0.69	0.46	0.40	1.92	3.61	0.28
CON +TC	0.04	0.29	0.68	0.46	0.39	1.92	3.61	0.15
DC-tM -n	0.05	1.23	1.43	0.49	0.40	0.30	0.57	0.27
DC-tM -n +TC	0.05	0.27	1.44	0.51	0.40	0.30	0.58	0.18
DC-sM -n	0.04	1.23	1.43	0.49	0.40	0.41	0.65	0.26
DC-sM -n +TC	0.04	0.27	1.44	0.51	0.40	0.41	0.67	0.18
DC-tM -g	0.05	1.23	0.67	0.47	0.39	0.30	0.56	0.28
DC-tM -g +TC	0.05	0.28	0.66	0.47	0.38	0.34	0.56	0.15
DC-sM -g	0.05	1.23	0.66	0.47	0.40	0.40	0.60	0.28
DC-sM -g +TC	0.05	0.28	0.66	0.47	0.39	0.42	0.61	0.15

Comparison of interpolation approaches. In Table 1, we also compare equivalent versions of our algorithm that perform the temporal interpolation with (*-g*) and without (*-n*) taking depth order cues into account. When applying our algorithm with estimated OF [17] our depth order guided interpolation quantitatively improves the results on average by 19 percent. In Table 1 the major quantitative improvement can be observed for *Ambush5*, which contained perceptual conflicts between large foreground objects (i.e., large depth differences). For test videos in which such conflicts occurred for smaller objects (e.g., *Temple3*) or between background objects, the observed qualitative improvements are smaller. In case of GT OF the obtained error rates of the naive interpolation are practically not affected by our guided interpolation. Figure 6 further shows corresponding examples of our 2D-to-3D conversion results in case of a naive and motion guided interpolation. The shown examples demonstrate that our depth guided interpolation visually improves the results concerning their perceptual coherence. The error rates of DC-tM and the error rates of DC-sM in Table 1 are similar. In this context, it is worth noting that the tested videos do not contain objects with irregularly movements, e.g., objects that stop for a few frames, which are the focus of DC-sM.

Evaluation of sensitivity to motion information. The errors of our algorithm in Table 1 were computed when using GT OF from [18] (Table 1, *top*) and by using estimated OF from [17] (Table 1, *bottom*). When comparing these errors, on average the quality of our results is hardly affected by the change of used motion information. When examining the results in detail, we already observe that the error rates for MS differ slightly. This is caused by the scribble matching and grouping results, which differ when considering estimated OF from those when considering GT OF. Consequently, the 2D-to-3D conversions are performed based on different color models which also affect subsequent versions of the proposed algorithm. Erroneously matched scribble pairs can also lead to additional perceptual conflicts in the conversion results that were generated using a naive interpolation technique. In fact, in this evaluation of our algorithm's sensitivity to the quality of the used motion information, the different matching and grouping emerged as a main reason for the observed de- and increases of error rates in Table 1 (i.e., Table 1, *bottom* versus *top*). Nonetheless, analogously to the conversion results that were obtained with GT OF, we observe improvements when additionally enabling TC, STC, CON and DC-sM or -tM.

Comparison to related work. In Table 2, we compare our proposed algorithm to related 2D-to-3D conversion algorithms [1, 3, 4] using the same user input. This evaluation is performed on a dataset that contains five recorded videos with stereo generated reference solutions [2, 4] (i.e., *City*, *Parade*, *Palace*, *Stairs* and *Football*), long videos with up to

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.5.SDA-433

Table 2. Comparison to related semi-automatic 2D-to-3D conversion algorithms [1, 3, 4] and reference solutions. For our algorithm with guided interpolation, we list the versions with the best results for each video.

$MSE \times 100$	Ours, **WTA**	Ours, **DB**	**WTA**	**DB**	**[4]**	**[1]**	**[3]**
City	DC -tM+TC	DC -sM	1.08	1.06	0.47	1.24	1.08
Parade	STC	STC	0.74	0.63	0.28	0.99	0.85
Palace	DC -tM+TC	CON+TC	1.16	1.31	1.20	1.56	1.14
Stairs	DC -tM+TC	DC -tM+TC	0.86	0.96	0.51	0.72	0.60
Football	STC	STC	0.51	0.52	0.40	0.57	0.64
Child	CON+TC	CON+TC	0.57	0.55	0.58	1.09	1.13
Head	DC -tM+TC	DC -tM+TC	0.49	0.44	0.65	4.68	1.45
Interview	CON+TC	CON+TC	0.80	1.10	0.56	12.76	15.57
Tsuk50	DC -tM+TC	DC -tM+TC	0.15	0.17	0.15	2.61	1.92
Tsuk380	DC -tM+TC	CON+TC	0.44	0.54	0.21	2.22	0.69
Tsuk1	DC -tM+TC	DC -tM+TC	0.10	0.09	0.15	2.22	0.79

101 frames that are provided with disparity and depth GT (i.e., *Child Head*, *Interview*) and three computer-generated videos with GT disparity maps from the new Tsukuba dataset [19] (i.e., *Tsuk1*, *Tsuk50*, *Tsuk380*). Table 2 lists the measured errors for each tested algorithm on this dataset. For our proposed algorithm we list the best version for each label-assignment scheme, i.e., WTA and DB. The shown evaluation results indicate that our algorithm achieves competitive 2D-to-3D conversion results. It outperforms the previous work of Guttmann et al. [1] on ten test videos and work of Phan et al. [3] on nine videos. For five of the test videos, we also achieve better results than Ivancsics et al. [4]. These five videos, i.e., *Head*, *Palace*, *Child*, *Tsuk1* and *Tsuk50*, contain motion in depth (e.g., due to camera movements). Figure 8 exemplary shows the results for *Tsuk1* and Figure 7 for *Palace*. It can be seen (e.g., Figure 7, *e)* and *f)*, Figure 8, *f)* and Table 2) that our algorithm produces plausible conversions that also capture the change in depth in a video. This is not the case for all tested algorithms, e.g., Phan et al.'s [3] algorithm (Figure 8, *c)*) that does not address the problem of temporal depth changes due to object motion or of perceptual coherence. However, it is fair noting that further developments of Phan et al.'s [3] algorithm in [6] address temporal depth changes when converting 2D videos to 3D. Ivancsics et al. [4] (Figure 8, *d)*) capture temporal depth changes, however, in the shown example the results contain artifacts (Figure 8, *d)*, *blue arrows*). These artifacts are caused by a temporal disparity interpolation that is performed within multiple small segments with different temporal extent. Our 2D-to-3D conversions contain hard disparity edges near object boundaries (e.g., Figure 7, *e)* and *f)*), which is challenging for related algorithms (e.g., Figure 7, *d)*, *yellow arrow*). We observe limitations of our algorithm for videos that contain (close-by) objects with similar colors or scribble annotations that result in overlapping color models (e.g., Figure 7, *e)* and *f)*, *red arrows*). Concerning these limitations, SPC and CON improved our results. In fact, in Table 2 MS is not listed, i.e., it was never the version that exhibited the smallest MSE.

Conclusion

The semi-automatic 2D-to-3D conversion algorithm that was presented in this paper has taken a step towards the generation of perceptually coherent depth maps. With a video object-segmentation algorithm as foundation we were able to use spatio-temporal segmentation information to capture hard edges in depth maps and perform smooth depth interpolations over time. These depth interpolations were performed in accordance with motion-caused occlusions. Evaluations with reference solutions demonstrated that our proposed algorithm generates plausible depth maps that capture the depth change of dynamic objects in a video. Enabling different components of our algorithm, e.g., additionally using a motion guided filtering instead of a common filtering, decreased the error rates of our results by approximately 16 percent. Further evaluations of our algorithm's sensitivity to motion information revealed that our scribble matching and grouping is often influenced by the motion information used. This suggests that it would be beneficial to support manual adjustments of the scribble matching and grouping results during processing. In presence of perceptual conflicts in conversion results that were generated with a naive depth interpolation technique, our proposed depth interpolation has demonstrated its ability to improve the conversion results. In comparison to related semi-automatic 2D-to-3D conversion approaches, our algorithm generates competitive results on a set of recorded and computer-generated videos.

Acknowledgments

The authors would like to thank R. Phan for sharing the implementation of his 2D-to-3D conversion algorithm. Part of this work has been funded by the Austrian Research Promotion Agency (FFG) under project Hyperion3D (no. 840222).

References

[1] M. Guttmann, L. Wolf and D. Cohen-Or, Semi-automatic stereo extraction from video footage, Proc. ICCV, pg. 136–142. (2009).

[2] N. Brosch, C. Rhemann and M. Gelautz, Segmentation-based depth propagation in videos, Proc. ÖAGM/AAPR, pg. 1–8. (2011).

[3] R. Phan, R. Rzeszutek and D. Androutsos, Semi-automatic 2D to 3D image conversion using scale-space random walks and a graph cuts based depth prior, Proc. ICIP, pg. 865–868. (2011).

[4] M. Ivancsics, N. Brosch and M. Gelautz, Efficient depth propagation in videos with GPU-acceleration, Proc. VCIP, pg. 1–4. (2014).

[5] M. Liao, J. Gao, R. Yang and M. Gong, Video stereolization: Combining motion analysis with user interaction, Trans. Visualization and Computer Graphics, 18, 7, 1079–1088 (2012).

[6] R. Phan and D. Androutsos, Robust semi-automatic depth map generation in unconstrained images and video sequences for 2D to stereoscopic 3D conversion, Trans. Multimedia, 16, 1, 122–136 (2014).

[7] C. Rhemann, A. Hosni, M. Bleyer, C. Rother and M. Gelautz, Fast cost volume filtering for visual correspondence and beyond, Proc. CVPR, pg. 3017–3024. (2011).

[8] N. Brosch, A. Hosni, C. Rhemann and M. Gelautz, Spatio-temporally coherent interactive video object segmentation via efficient filtering, Proc. DAGM, pg. 418–427. (2012).

[9] S. Ghuffar, N. Brosch, N. Pfeifer and M. Gelautz, Motion estimation and segmentation in depth and intensity videos, J. of Integrated Computer-Aided Engineering, 31, 3, 203–218 (2014).

[10] G. Borgefors, Distance transformations in digital images, Comput. Vision Graph. Image, Process., 34, 4, 344–371 (1986).

[11] K. He, J. Sun and X. Tang, Guided image filtering, Proc. ECCV, pg. 1–4. (2010).
[12] G. Palou and P. Salembier, Depth order estimation for video frames using motion occlusions, Proc. IET Computer Vision, pg. 152–160. (2013).
[13] E. Turetken and A. Alatan, Temporally consistent layer depth ordering via pixel voting for pseudo 3D representation, Proc. 3DTV Conference, pg. 1–4. (2009).
[14] N. Sundaram, T. Brox and K. Keutzer, Dense point trajectories by GPU-accelerated large displacement optical flow, Proc. ECCV, pg. 438–451. (2010).
[15] M. Bleyer, C. Rother, P. Kohli, D. Scharstein and S. Sinha, Object Stereo – Joint stereo matching and object Segmentation, Proc. CVPR, pg. 3081–3088. (2011).
[16] M. Lang, O. Wang, T. Aydin, A. Smolic and M. Gross, Practical temporal consistency for image-based graphics applications, Trans. Graph., 31, 4, 1–8 (2012).
[17] C. Liu, Beyond pixels: exploring new representations and applications for motion analysis, Doctoral Thesis, Electrical Engineering and Computer Science at the Massachusetts Institute of Technology. (2009).
[18] D. Butler, J. Wulff, G. B. Stanley and M. J. Black, A naturalistic open source movie for optical flow evaluation, Proc. ECCV, pg. 611–625. (2012).
[19] S. Martull, M. P. Martorell and K. Fukui, Realistic CG stereo image dataset with ground truth disparity maps, Proc. ICPR, pg. 1038–1042. (2012).

Author Biography

Nicole Brosch received DI (2009) and Mag.rer.soc.oec. degrees (2010) from and is a PhD-student at Vienna University of Technology. Her research interest include 2D-to-3D conversion.

Tanja Schausberger received her DI degree from Vienna University of Technology (2015).

Margrit Gelautz received her PhD (Telematics) from Graz University of Technology (1997). She is an associate professor at Vienna University of Technology with a focus on image and video analysis and synthesis, with special interest in 3D-film applications.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.5.SDA-433

Figure 5. *Visual comparison of different versions of the proposed 2D-to-3D conversion algorithm (WTA) for Ambush2 (a) and b)) and Alley1 (c) and d)). a) and c) Input video with scribbles in the first and the last frame, including corresponding depth GT. Blue arrows point to scribbles that cannot be tracked through the entire video: a) head is occluded by axe, c) arm leaves the scene. For these scribbles the spatio-temporal closeness constraint and the 3D connectivity constraint cannot be applied. b) and d) Our 2D-to-3D conversion results that were obtained with GT OF and with estimated (est.). OF.: Foreground bright, background dark. For better visualization the contrast of the depth maps was enhanced. For est. OF and GT OF: b) Red arrows highlight areas that are improved (green arrows) by motion guided filtering (+TC). d) Red arrows highlight areas that are improved (green arrows) by STC or CON. Original video from [18].*

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.5.SDA-433

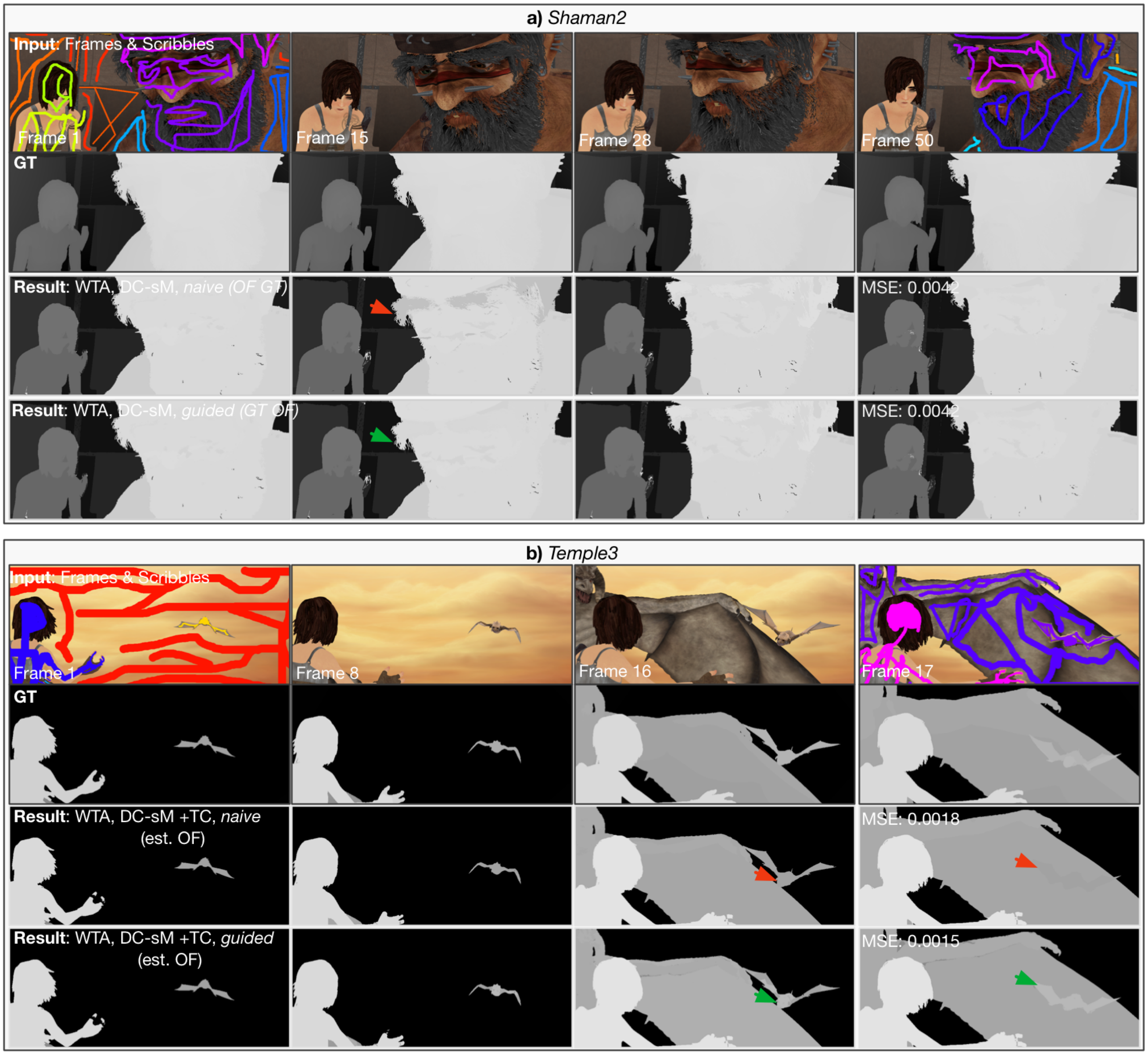

Figure 6. *Visual comparison of naive and depth order guided interpolation. Input video and depth GT with scribbles in the first and last frame (top). Obtained 2D-to-3D conversions for the shown example frames (bottom): Foreground bright, background dark. We show the results that were obtained when using a naive interpolation and our depth order guided interpolation for a) Shaman2 (with GT OF) and b) Temple3 (with estimated OF [17]). Perceptual incoherencies (red arrows) are corrected (green arrows) by our depth guided interpolation. Original video from [18].*

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.5.SDA-433

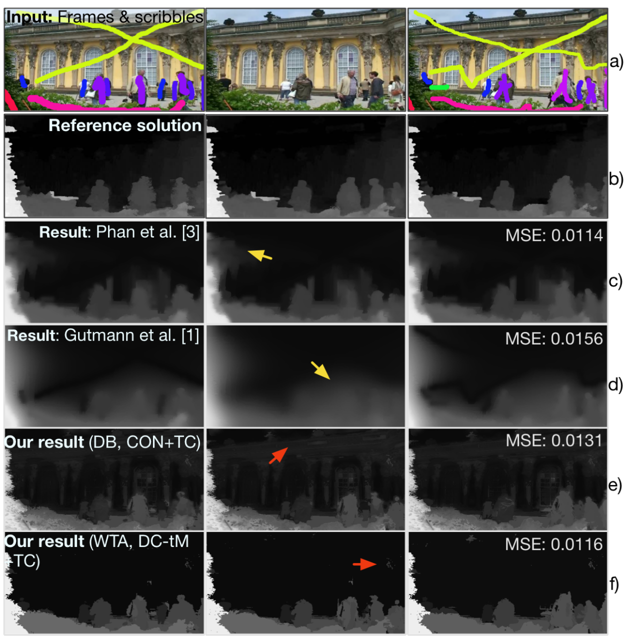

Figure 7. *Visual comparison to [1, 3]. a) Input frames and scribbles from Palace. b) Corresponding reference solution (estimated disparity maps): Foreground bright, background dark. Result obtained with c) Phan et al.'s [3] and d) our implementation of Guttmann et al.'s [1] algorithm: Over-smoothed disparity edges at object border (yellow arrows). e) and f) Results from our algorithm. Compared to c) and d), e) and f) provide hard disparity edges at object borders. Red arrows indicate errors in our results that are caused by overlapping color models (e.g., grey facade versus grey clothing).*

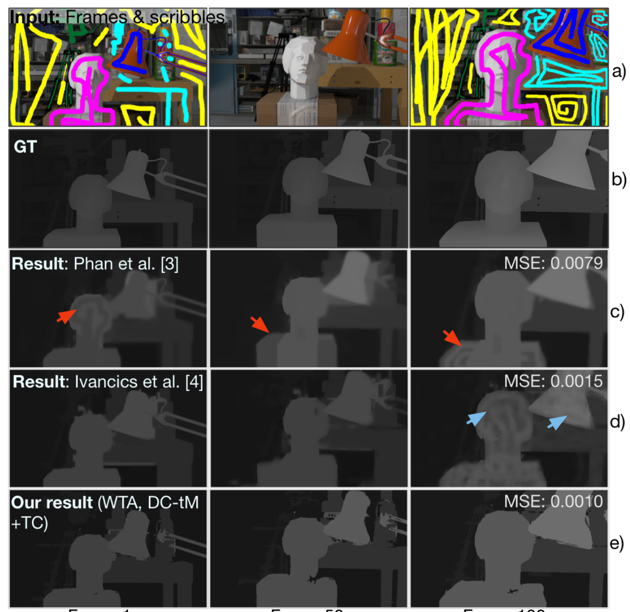

Figure 8. *Visual comparison to [3, 4]. a) Input frames and scribbles from Tsuk1. b) Corresponding GT disparity maps: Foreground bright, background dark. c) Result obtained with Phan et al.'s [3] algorithm and d) with Ivancics et al.'s [4] implementation of [2]. e) Our results. Contrary to c) and d), in e) the disparity change due to a camera zoom is captured smoothly. In c) temporal depth changes are not addressed, i.e., disparities from the first and last frame are propagated independently (red arrows). Result in d) contains artifacts due to a temporal disparity interpolation within multiple small segments with different temporal extent (blue arrows). Original video from [19].*