Stereo Conversion with Disparity-Aware Warping, Compositing and Inpainting

Lukas Mehl[1,*] Andrés Bruhn[1] Markus Gross[2,3] Christopher Schroers[3]

[1]Institute for Visualization and Interactive Systems, University of Stuttgart
[2]Computer Graphics Lab, Department of Computer Science, ETH Zurich
[3]DisneyResearch|Studios, Switzerland

lukas.mehl@vis.uni-stuttgart.de

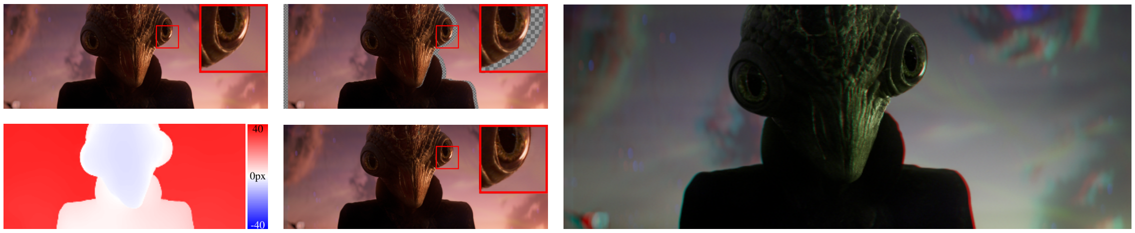


Figure 1. Stereo Conversion. Given a monocular image and a disparity estimate (left), our method performs disparity-aware warping, compositing and inpainting (center) to produce stereo (right, visualized anaglyph).

Abstract

Despite of exciting advances in image-based rendering and novel view synthesis, it is still challenging to achieve high-resolution results that can reach production-level quality when applying such methods to the task of stereo conversion. At the same time, only very few dedicated stereo conversion approaches exist, which also fall short in terms of the required quality. Hence, in this paper, we present a novel method for high-resolution 2D-to-3D conversion. It is fully differentiable in all of its stages and performs disparity-informed warping, consistent foreground-background compositing, and background-aware inpainting. To enable temporal consistency in the resulting video, we propose a strategy to integrate information from additional video frames. Extensive ablation studies validate our design choices, leading to a fully automatic model that outperforms existing approaches by a large margin (49-70% LPIPS error reduction). Finally, inspired from current practices in manual stereo conversion, we introduce optional interactive tools into our model, which allow to steer the conversion process and make it significantly more applicable for 3D film production.

1. Introduction

Live-action feature films are typically not filmed in stereo apart from a few notable exceptions. Still, a significant amount of high profile productions are available in stereo. This is possible through a post production process referred to as *stereo conversion*. Although there is some automation in parts of the conversion process, it still heavily relies on manual work, which makes converting even a single feature film extremely expensive. A fully automatic conversion pipeline could hence significantly reduce costs while enabling studios to make large amounts of legacy content available to new audiences in stereo.

In this work, we focus on stereoscopic movies, *i.e.* two-stream videos that are presented to each eye separately. From the way the objects in both streams are displaced to each other, *i.e.* via the pixelwise *disparity*, the human visual system perceives depth. In contrast to common stereoscopic datasets, 3D movie disparities can have positive and negative values describing objects behind and in front of the screen plane, respectively. While shooting 3D movies occasionally employs a converging camera setup, resulting in non-horizontal displacements, stereo conversion post production resorts to an orthoparallel setup with horizontal disparities. To avoid negative-only disparities, the latter introduces an additional horizontal shift to one of the views.

Although several stereo conversion approaches exist, they either do not work on high resolutions [3,36], only consider negative disparities [7,16,34,39], or produce blurry results [36,40]. Further, several methods [3,16,40] work with disparities of the *target* instead of the *input* frame, thus not taking advantage of the latest progress in single image depth prediction, making them fall behind in the perceived depth of their predictions. Last but not least, recent dynamic neural radiance fields [17–19] provide interesting results. However so far they have only been used at lower resolutions and

[*] Work was performed during a research stay at ETH Zurich

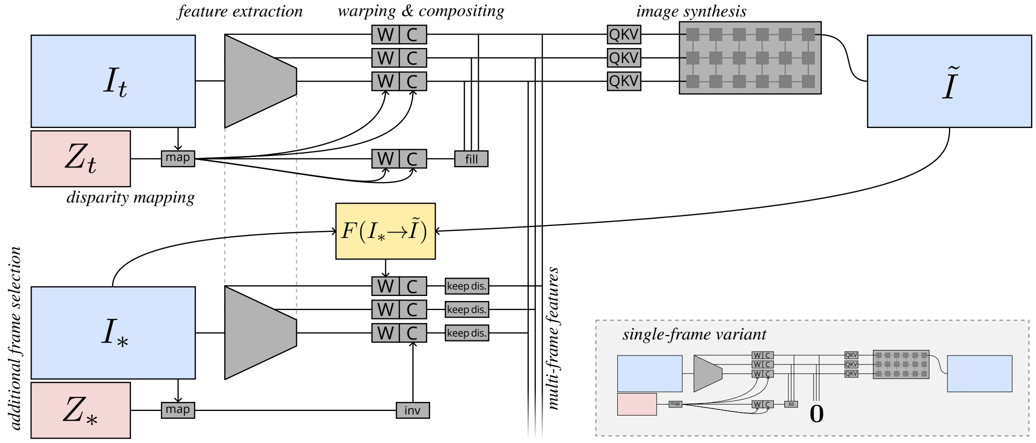


Figure 2. Overview of our approach.

exhibit long reconstruction times. Hence neither method so far reaches the bar for production-quality stereo conversion.

Contributions In this paper, we propose a method for stereo conversion from monocular videos, which produces high-resolution results with high visual quality —even allowing to reach production-level quality for shots with limited complexity. In this context, we make the following contributions: (i) We present a fully automatic stereo conversion approach that is trained end-to-end on movie data. (ii) To enable temporal consistency, we further introduce a strategy to integrate information from multiple frames. (iii) Experiments not only show that our model outperforms existing approaches visually and quantitatively by a large margin, but also ablate the employed depth model and design choice of the architecture, loss function and multi-frame strategy. (iv) Further, we demonstrate that our approach is applicable to 3D movie production by introducing optional extensions to our model to interactively control depth perception and inpainting area.

2. Related work

Let us now discuss research on generating novel views from monocular video. Directly related is stereo conversion literature, but we also cover works from the broader novel view synthesis field that could be applied to our use case.

Stereo conversion There are several approaches that directly consider stereo conversion. While some are specifically designed for the 3D film setting with positive-negative disparities [3, 36, 40], others only work on negative disparities [7,16,34,39] in the context of automotive data [7,16,39] or training data creation [34]. Many of these approaches use backward warping [3, 7, 16, 40], which creates the ill-posed problem of estimating right-view registered disparity from left views in [3, 16, 40]. Only one method generates the second view through forward warping [34] but makes use of a simple non-differentiable warping while also not reasoning about correct inpainting. Other approaches circumvent warping by averaging integer-shifted copies of the input image [36,39]. Further, [3,16,34,36,39,40] only consider left-to-right prediction, only [7] jointly perform both left-to-right and right-to-left prediction.

In contrast to previous work, we propose a stereo conversion model for high-resolution image data based on a disparity estimate of the *input frame*. We support negative and positive disparities and all generation strategies (left-to-right, right-to-left or both from center). Also, unlike previous approaches, we perform differentiable disparity-aware forward warping of feature pyramids and design an image synthesis step that performs disparity-aware inpainting, leveraging information from multiple frames.

Novel view synthesis When considering the broader novel view synthesis literature, several methods could be applied that were not originally designed for stereo conversion.

A recent topic with rapid developments are neural radiance fields [22] (NeRFs). While originally considered only for static scenes, recent dynamic NeRFs also consider dynamic environments [5, 17–19], *i.e.* scenes with independent object motion. Since NeRFs are developed for general view generation, they can be adapted to stereo conversion by rendering from a horizontally displaced camera with additional shift. While dynamic NeRFs yield good results in scene reconstruction including complex reflection scenarios, they are still subject to several shortcomings. First, so far they have not been evaluated on high resolutions, as required for 3D movie production. Further, even for scenes of a few seconds, reconstruction times are in the order of multiple days [17–19]. Lastly, their performance is still unclear for complex motion scenes, *e.g.* with several moving objects or with dynamic objects covering large image parts, and for (near-)static scenes without motion parallax.

In a similar direction, multiplane images [11, 30, 31, 33]

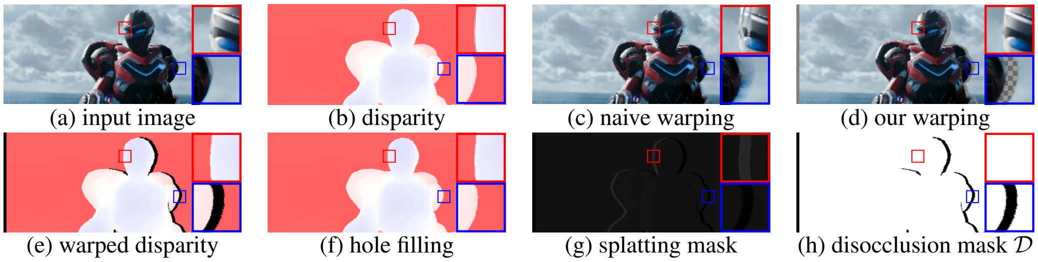

(a) input image
(b) disparity
(c) naive warping
(d) our warping
(e) warped disparity
(f) hole filling
(g) splatting mask
(h) disocclusion mask $\mathcal{D}$

Figure 3. Disparity-aware warping and compositing. Compared to naive warping, ours handles occlusions (red) and disocclusions (blue). We also perform hole filling of the warped disparity and extract splatting and disocclusion masks.

can also be used for rendering novel views. However, such strategies only allow very small camera displacements and are rarely evaluated for temporal consistency [37].

3. Stereo Video Conversion

Our method takes a frame sequence and converts it to a leftwards or rightwards displaced novel view for stereo vision. We assume given depth estimates for every frame. These can *e.g.* be obtained by a method for single image depth estimation or consistent video depth estimation. We show an overview of our method in Fig. 2. In the following, we first introduce the single-frame version of our model (shown in the upper part of Fig. 2) with the four steps of *disparity mapping*, *feature extraction*, *warping & compositing* and *image synthesis*. In this single-frame variant, we omit the multi-frame features (see Fig. 2, bottom right). Then, we describe our multi-frame approach, which is a two-stage process. It consists of first executing the single-frame model, and then, with a single-frame prediction as input, executing the model again with multi-frame features. To this end, we describe the *additional frame selection* and *multi-frame features* (see Fig. 2, bottom left).

3.1. Disparity Mapping

Our method uses a disparity estimate to perform the warping, compose foreground over background, and guide the inpainting process. To obtain this disparity, we rely on a preceding depth estimation, *e.g.* with a single-image or video depth estimation method. Given a depth estimate Z, it can be mapped to disparity d as

$$d = a \cdot \frac{1}{Z} - b \,, \tag{1}$$

with a steering the perceived deepness of the scene, and b controlling the positioning of the scene relative to the screen plane, *i.e.* selecting zero disparity. If Z is given in metric depth, the technical choice for a would be the stereo camera baseline distance times its focal length. However this might be infeasible, for non-metric depth [27], unknown focal length, or if the disparity range is adjusted for creative or perceptual reasons [13, 15]. At the same time, selecting b is always a creative choice in 3D movie production [20].

For these reasons, we propose two strategies to select these parameters: First, we describe a simple automatic method, which we use throughout our experiments to compare against other methods with automatic disparity mapping [36]. Second, we introduce interactive strategies in Sec. 6. For automatic selection, we first normalize $\frac{1}{Z}$ in Eq. (1) to the range $[0;1]$, making a the distance between minimum and maximum disparity, the *depth bracket* [20]. Then, we propose to leverage the *shot scale* of the given video to select reasonable values for a and b for each sequence. To this end, we use a method to perform shot scale classification [29] based on the input image, select fixed values a and b per class and compute the disparity with Eq. (1).

3.2. Feature Extraction

While many previous conversion approaches directly warp images [3,7,16,34,36], it has been shown that working in feature space, *i.e.* warping feature representations and decoding them afterwards, is advantageous [1,25,35]. Hence, we make use of a *feature pyramid* encoder [25] to extract features at full, 1/2 and 1/4 resolution, with 32, 64 and 96 channels respectively. We also append the original image to the highest pyramid level.

3.3. Warping & Compositing

With disparity and extracted features, we perform disparity-aware warping and compositing to transition features towards the novel view. In contrast to methods resorting to backward warping [3, 7, 16, 40], our approach uses *differentiable forward warping* with disparity as horizontal displacement vectors, enabling us to handle compositing and inpainting in occlusions and disocclusions, respectively.

For disocclusions, forward warping leaves unfilled areas, which are inpainted in the subsequent image synthesis step. For occlusions, *i.e.* multiple pixels mapped to the same location, we use *disparity-aware compositing* to position foreground over background pixels. For the latter, we employ differentiable exponential warping [25] that uses per-pixel weights of the warping input controlling the compositing weighting in occlusions. As weights, we again use the disparity, scaled with a single learned parameter α. Since

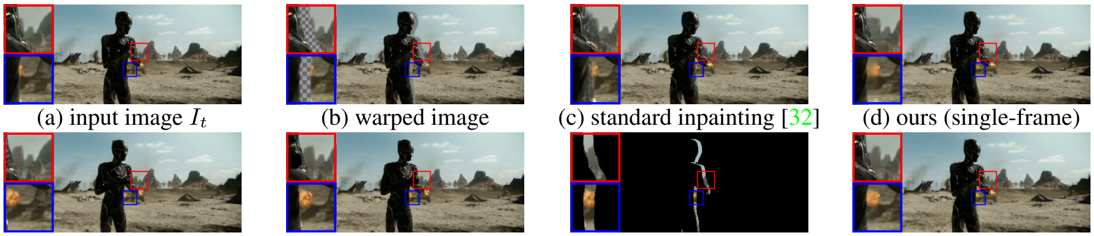

(a) input image I_t
(b) warped image
(c) standard inpainting [32]
(d) ours (single-frame)
(e) additional frame I_*
(f) warped additional frame
(g) only keep disocclusions
(h) ours (multi-frame)

Figure 4. Image synthesis. After the input (a) is warped (b), standard inpainting (c) wrongly extends the foreground, while ours uses local background for inpainting (d). Wrongly extended disoccluded objects (mountains) can be resolved by an additional frame (e). Warped with inverted weights (f) with only disocclusion areas kept (g) it guides multi-frame prediction (h).

the original exponential warping [25] is subject to numerical instabilities [23], we use a stable implementation [1].

An overview of this step is given in Fig. 2. After feature extraction, we warp each pyramid level separately, with warping and compositing steered by disparity, accordingly adapted with bilinear interpolation. We visualize results in the first row of Fig. 3. When warping the input (a) using its disparity (b) with simple nearest-neighbor forward warping (c), occlusions and disocclusions are not handled appropriately. In contrast, our warping strategy (d) deals with occlusions correctly, leaving disocclusions unfilled.

In an additional step, we warp the disparity *with itself* (see Fig. 2), to obtain disparity aligned to the *target frame*. Again, occlusions are handled by the warp and disocclusion holes remain. Since we want to use the warped disparity as guidance for the inpainting step, we propose a simple approach to fill these holes with neighboring *background* disparity pixels. To this end, we first note that these pixels lie on the right-hand[1] side of the disocclusions. Thus, we fill the holes through differentiable single-sided dilation with a structuring element that has height 1 and its origin located at its left-most[1] pixel, leading to dilation that only fills leftwards[1]. We empirically choose the width of the structuring element as 7 and apply dilation multiple times until all holes are closed. In Fig. 3, we show the warped disparity with disocclusion holes (e) and the result of the hole filling (f). With this, we obtain a dense estimate of the target frame disparity, which can be used as guidance in the image synthesis. Finally, we extract two additional masks from warping: The splatting mask (g) determining for every pixel the number of pixels that are splatted onto it and the binary disocclusion mask $\mathcal{D}$ (h) obtained by thresholding the splatting mask.

3.4. Image Synthesis

The final and essential step of our method is the *image synthesis* module compiling the final image. Its core goal is to perform *inpainting* in *disocclusions*. To this end, it is crucial that the inpainting uses only the *local background* instead of generating *arbitrary* realistic content as done by standard inpainting methods, see Fig. 4 (top row). Given the input (a), we perform warping (b), leaving disocclusion areas unfilled. When using general-purpose inpainting [32] (c), these areas are plausibly filled. However, the foreground is extended, resulting in misaligned object edges in the synthesized output, leading ultimately to a wrongly perceived depth. In contrast, the desired inpainting should only rely on information from the local background area (d).

Based on this motivation, we design our image synthesis module. To realize inpainting guided by local foreground-background information, we utilize the warped disparity [12] after performing the previously introduced hole filling. We thus concatenate it to all feature pyramid levels (*cf*. Sec. 3.2), resized accordingly. Additionally, we concatenate splatting and disocclusion masks, as well as multi-frame features, which we will introduce in the following sections; for the single-frame setting we leave them zero-initialized. Then, our image synthesis module consists of two steps. In a first step, we employ *local multi-scale self-attention*, to let the multi-resolution features interact locally with each other in a pair-wise manner. To this end, self-attention is performed on every feature pyramid level, including also the highest resolution. To handle the large memory requirements of full self-attention, we restrict the attention to a local neighborhood, since through our multi-scale approach the receptive field is still reasonably large. This way, self-attention can be applied to large input resolutions without any tokenization or downsampling. We make use of a recent efficient implementation [8] and apply single-headed attention with neighborhood size 13. In the second step, we generate the final image by feeding the attention-filtered multi-scale features into a 3×6 GridNet [4], replacing transposed convolutions with bilinear upsampling [24].

3.5. Additional Frame Selection

So far, we presented stereo conversion based on a single input frame. However, when considering disocclusion areas, revealing content that is visible in a *different* time step of the input video, inpainting only from a single input frame is not sufficient. To this end, we select additional frames I_*.

[1] For *rightwards* stereo conversion, *leftwards* is vice-versa.

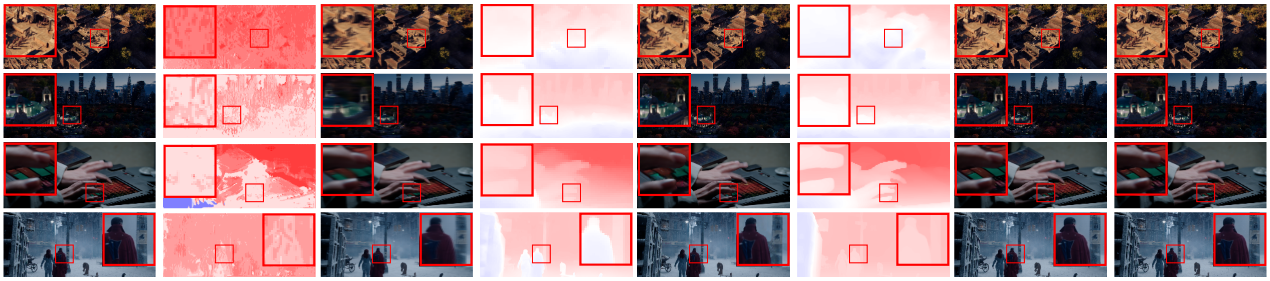


Figure 5. Further results and comparison to state of the art. Our approach shows significantly sharper results of higher visual quality.

In order to manage computational effort, we refrain from using a large set of additional frames, but choose a compromise of long-term and short-term information propagation. For long-term information, we select the first and last frame of the input frame sequence, to cover cases where these depict a significantly large amount of the background regions needed for inpainting. For short-term information, we additionally choose the frames at a half-second distance (12 frames) to the input frame, if existent. For each of these up to four candidate frames, we perform a *similarity* check to confirm that the LPIPS distance between input and candidate frame is < 0.7, removing the candidate frame otherwise. Intuitively, this check rejects frames that are very dissimilar to the current one, *e.g*. in sequences with strong changes. With the resulting set of additional frames at hand, we continue with the extraction of multi-frame features.

3.6. Multi-Frame Features

By considering additional frames from the input sequence for stereo conversion, we aim at more stable results and to integrate valuable information in disoccluded areas. To this end, the main idea is to *register* information from additional frames *to the target frame* to exploit it during image synthesis. This motivates our two-stage multi-frame model: Based on a single input frame, we first generate an initial target frame prediction, which is used in the second step to align additional frames to it. For every additional frame I_* with associated depth Z_* we proceed as shown in Fig. 2 (bottom left): First, we compute the optical flow $F(I_* \to \tilde{I})$ using [9]. Afterwards, we apply the same feature extractor (*cf*. Sec. 3.2) to I_* and then employ the warping and compositing as before, however with two differences: First, we use optical flow to determine target positions, to align features to the target frame. Second, we *invert* the disparity weights for compositing to intentionally composite local background over foreground. While unintuitive at first, this ensures that foreground objects do not disturb background content that is required for disocclusion inpainting. During optical flow warping, a second disocclusion mask $\mathcal{D}_{\text{OF}}$ is obtained, marking invalid locations after warping. Then, we use $\mathcal{D}$ from the preceding stage (see *e.g*. Fig. 3 (h)) to zero-out all non-disocclusion areas. By keeping only contents warped into the disocclusion areas, we prevent disturbances in the image synthesis from other parts of the image, which might be significantly different from the input image.

This is visualized in Fig. 4, where after single-frame prediction (d) from the input image (a), background regions are stretched into the foreground character (see *e.g*. mountains) and an additional frame (e) is needed, which shows key background areas unoccluded. After warping I_*, compositing *background over foreground* (f), background areas are aligned to the target frame (red/blue square in (d) vs. (f)), while the essential regions remain unoccluded. Only features in disocclusion areas are kept (g) leading to the multi-frame prediction (h) resolving the single-frame issues.

Before piping the additional features into the image synthesis, we perform two additional checks to determine if the features are actually valuable. First, the *photometric check* ensures that lighting and content changes between I_* and the target frame are not too large. To this end, we compute the L_1 distance between the optical-flow-warped additional image and the single-frame target prediction $\tilde{I}$, not counting optical flow disocclusion areas $\mathcal{D}_{\text{OF}}$ and omit I_*, if $L_1 > 4$. Secondly, we perform a *filling check* to verify that I_* can significantly fill the disocclusion areas. We remove the additional frame, if the fill amount $|\neg\mathcal{D}_{\text{OF}} \cap \mathcal{D}|/|\mathcal{D}|$ is below 30%. Finally, we generate the final multi-frame feature pyramid. Initializing it as zero, we accumulate the features from all additional frames, which are masked by their optical-flow and disparity disocclusion masks $\neg\mathcal{D}_{\text{OF}} \cap \mathcal{D}$.

4. Training and Implementation Details

Data Generation We create a large dataset from 21 recent 3D movies with horizontal disparities. Since movie data is highly correlated, especially within shots, we subsample to select meaningful frames as follows: We separate every movie into shots using an edit decision list, excluding studio intros and credits. Then, for every shot, we select the time-central frame, and from there, all non-black frames in

Table 1. Comparison to state-of-the-art.

Method	LPIPS ↓	PSNR ↑	d_{MAE} ↓
Deep3D	0.2351	28.078	12.31
3DPhoto	0.1391	28.006	8.25
ours (single-frame)	0.0715	28.227	**8.22**
ours (multi-frame)	0.0714	**28.231**	**8.22**
ours (w/ Deep3D's depth)	0.1261	25.933	12.31
ours (w/ 3DPhoto's depth)	**0.0706**	28.147	8.25

1 second steps. From the 21 movies, we select 17 for training (250992 frames in total) and 4 for testing, from which we randomly select 250 frames each (1000 test frames in total). Afterwards, we compute *reference disparity*, *i.e.* the disparity from the left to the right frame and vice-versa. To compute these, we follow [27] and employ a recent optical flow method [9, 10], keeping only horizontal displacements.

Loss function As loss, we use the L1 distance between prediction $\tilde{I}$ and ground truth second frame I. Here, we make use of the disocclusion mask $\mathcal{D}$ to separately weight areas with known information and disocclusions, *i.e.* areas where information has to be extrapolated. We choose a larger weight for known areas, since we want a precise reconstruction of the other image in those areas. The L1 loss reads

$$\mathcal{L}_{L1} = \sum_{x\in\mathcal{D}} \left|\tilde{I}(x) - I(x)\right| + \beta \sum_{x\notin\mathcal{D}} \left|\tilde{I}(x) - I(x)\right| . \qquad (2)$$

Additionally, we employ a perceptual LPIPS loss [38] $\mathcal{L}_{\text{LPIPS}}$ to ensure visually pleasing results. Our total loss then reads $\mathcal{L} = \mathcal{L}_{L1} + \gamma \cdot \mathcal{L}_{\text{LPIPS}}$, where we empirically choose $\beta = 10$ and $\gamma = 10$.

Training We train our model end-to-end using the left-right pairs with associated reference disparity. We use the reference disparity as a direct input to our network, circumventing depth estimation and disparity mapping, since this gives a precise alignment of predicted frame and ground truth frame. We train for 200K steps using a batch size of 16 and the Adamax optimizer [14] with learning rate 1e-3, decayed by 0.8 every 10K steps. We train our model in CIELAB space as it better models human color perception. During training, we remove the two-stage execution of our model, as it comprises time-consuming optical flow computations, but still train jointly for single-frame and multi-frame prediction: Randomly in half of the training steps, we use zero-initialized multi-frame features to train single-frame prediction, otherwise we supply multi-frame features. For the latter case, we apply the pipeline as described above, but omit optical flow warping. Since this would result in unaligned features not too useful to the model, with a random probability of 50%, we use the *ground truth frame* as I_*. Intuitively, this lets the model learn that there might be reliable information in the multi-frame features, helpful to fill disocclusions. We provide further details in the supplement.

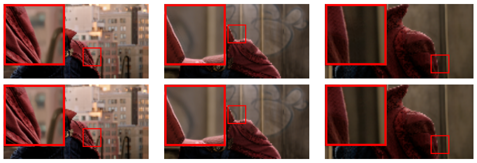


Figure 6. Single-frame (top) vs. multi-frame (bottom).

5. Experiments

In the following, we compare our model to the state of the art, ablate design choices and assess dynamic NeRFs for stereo conversion. If not stated otherwise, throughout our experiments, we use MiDaS [26, 27] for depth estimation.

Comparison to state of the art There are not many approaches available that can be compared directly to ours. We thus select two methods: First, we compare to the only stereo conversion method with public source code, Deep3D [36]. As it only predicts left-to-right, we remove all right-to-left samples from our test split before evaluating. Second, we select the recent 3DPhoto [30] that also focuses on background-aware inpainting. While their approach renders arbitrary close-by camera poses, there is no direct way to use it for stereo conversion with positive and negative disparities. We thus use their method by rendering from a specific camera pose that is displaced in an orthoparallel way and using an additional shift of the image plane. With these additions, their approach can be used to generate results for precise disparity ranges, similar to our method. To select these ranges we use our automatic selection (*cf*. Sec. 3.1) generating results spatially aligned to ours.

We show results in Tab. 1. It can be clearly seen that our method outperforms all other approaches by a large margin, especially in the human-perception like LPIPS error. These improvements hold for the single-frame method, which is directly comparable to the other approaches, but also for multi-frame, which will be discussed in the following sections. We also show image results in Fig. 5. Visually, our method generates significantly sharper results than the other approaches, which both exhibit visual artifacts. For Deep3D, a strong horizontal blur can be observed, which is probably caused by their weighted sum of displaced images. For 3DPhoto, too sharp object edges stemming from the mesh representation can be seen, as well as in some cases the limitations of their background inpainting. Both comparison methods, of which only Deep3D was targeted for videos, also show significant temporal stability issues for video data.

Influence of depth While our final model outperforms both other approaches significantly in the image metrics LPIPS and PSNR, we also see improvements in the disparity error d_{MAE}, comparing disparity prediction to reference disparity. Thus, to investigate the reason for improvement,

Table 2. Single-frame vs. multi-frame prediction.

Frame	checks				LPIPS ↓	PSNR ↑	
Selection	sim.	pho.	fill	#fr.	all	disocc.	not disoc.
single				1	0.0715	26.14	28.36
FL	✓	✓	✓	2.13	0.0715	26.11	28.36
±12	✓	✓	✓	2.04	0.0715	26.13	28.36
FL,±12	✓	✓	✓	3.17	**0.0714**	**26.15**	28.36
FL,±12	✗	✓	✓	3.17	**0.0714**	**26.15**	28.36
FL,±12	✓	✗	✓	3.61	0.0715	26.14	28.36
FL,±12	✓	✓	✗	4.01	0.0715	26.09	28.36
FL,±12	✗	✗	✗	4.65	0.0715	26.08	28.36

Table 3. Ablation of our architecture and loss function.

Method	LPIPS ↓	PSNR ↑
ours	**0.0126**	**41.041**
forward warping NN-interpolation	0.0441	34.835
disparity-aware warping & compositing	0.0308	36.803
ours w/o attention	0.0127	40.963
ours w/o disparity filling	0.0130	40.958
ours w/o disparity	0.0140	40.321
ours w/o disparity, warping masks	0.0141	40.310
$\mathcal{L}_{L1}$	0.0202	40.852
$\mathcal{L}_{L1} + \mathcal{L}_{\text{style}}$	0.0159	40.714
$\mathcal{L}_{L1} + \mathcal{L}_{\text{LPIPS}} + \mathcal{L}_{\text{style}}$	**0.0126**	41.032

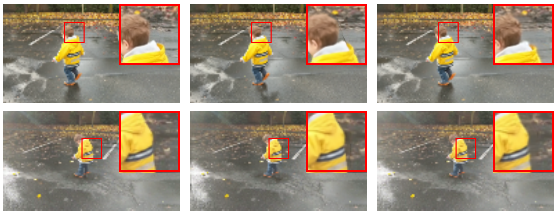


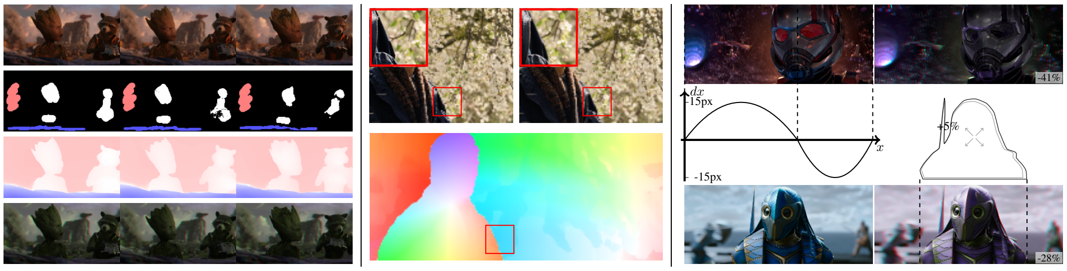
Figure 7. Comparison to recent dynamic NeRFs. *Left:* input images, *center:* results for Dynibar [18], *right:* our result.

we separate disparity estimation from image generation by also reporting results of our multi-frame model, when using disparity results of other approaches. First, it can be seen that when using the same disparity, our model still significantly outperforms Deep3D in the LPIPS metric. At the same time, PSNR, which strongly depends on spatial alignment, drops worse than Deep3D, which can be explained through their weighted average warping leading to blur to which PSNR is not too susceptible [38]. When using the 3DPhoto depth with our disparity mapping for our method, we get even better LPIPS values at a reduced PSNR. This is not surprising since their implementation uses a slightly different variant of the same depth model [21, 26, 27].

Multi-frame strategy We evaluate details of our multi-frame strategy in Tab. 2, reporting PSNR separately for disocclusions and the rest as well as the average number of frames (#fr.) that was considered for the prediction. First, we note that there are no large quantitative differences between single-frame and the tested multi-frame approaches. This is not surprising, since we only expect improvements in the disocclusion areas, and the PSNR metrics show that, in fact, only these areas change, while the rest is unaffected. We show a visual comparison between single- and multi-frame results in Fig. 6. While there are visible improvements in the images, they can be seen best in motion, since there the issue of background stretching into foreground is most obvious.

We also ablate aspects of our multi-frame strategy. First, we evaluate the frame selection (*cf*. Sec. 3.5). When only considering two additional frames, either selecting first and last (FL) or distance-12 frames (±12), results are not able to outperform the single-frame method, potentially due to the low resulting frame count, which after checks is around 2 (*i.e*. only 1 additional). Only with the combination of both these strategies, yielding 3.17 frames, a small improvement is reached. In a second experiment, we investigate the checks that remove frames due to low *similarity* (sim.), high *photometric* distance (pho.) or low *fill* amount. Here, disabling the similarity check yields the same results, likely since the photometric check later removes the same problematic frames. However, we still keep the check, since it saves computation time as it is used before optical flow computation and warping, resulting in a 4.4% speedup. We also investigate the influence of the other checks, which when disabled, yield a larger number of frames, but at a decreased performance. The worst results with highest frame count can be observed without any checks, which is clear since in this case all frames, with strong content or illumination changes or from unhelpful camera angles are used.

Ablation We ablate design choices of our model in Tab. 3, using the reference disparity as input. First, we compare our model to base variants. When using the disparity for nearest-neighbor interpolated forward warping, *i.e*. performing compositing and inpainting without disparity guidance, we obtain the worst results. Disparity-aware compositing improves results, but only when employing our full disparity-aware image synthesis, largest gains are reached. Second, we evaluate architecture and loss choices by retraining our single-frame model for 100K steps. When removing self-attention or disparity filling, performance slightly decreases. Removing disparity completely as an input to image synthesis leads to a significant quality drop, proving the importance of disparity guidance. We also show that also removing splatting and disocclusion warp masks decreases performance even more. To evaluate the choice of perceptual loss, we compare to removing perceptual loss, using the recent style loss [6, 28] or using LPIPS and style loss. Here, employing LPIPS brings the largest advantage, additional style loss does not improve further.



% disocclusion reduction				
	stretch			
zoom	0px	5px	10px	15px
0%	0	-18	-34	-44
1%	-11	-29	-41	-48
2%	-20	-35	-44	-49
3%	-27	-38	-45	-48
4%	-31	-39	-44	-47
5%	-32	-39	-43	-46

Figure 8. Optional interaction in our approach. *Left:* Sparse scribbles control the disparity mapping. *Center left*: Comparison of 2D (left) and 3D (right) movie version with optical flow (bottom). *Center right*: Implemented mitigation strategies; 15px background stretch, 5% foreground zoom. *Right:* Reduction of disocclusion areas by mitigation strategies.

Comparison to dynamic NeRFs We also compare our model to the recent DynIBaR [18], which we retrain on one of their low-resolution scenes. With slight changes to their rendering, we generate results with positive-negative disparities, see Fig. 7. It can be seen that their approach performs similar to ours, with only small artifacts visible. However, there are still drawbacks when considering dynamic NeRFs for stereo conversion such as time-consuming training and the requirement of camera poses. Also, so far they are not used for high resolutions, DynIBaR uses maximum resolutions of 768×432.

6. Interaction Points

Finally, we extend our model with optional interactive tools for disparity mapping and inpainting mitigation. We describe the implementation details in the supplement.

Controlling Disparity Mapping While we propose an approach with automatic disparity mapping, in 3D movie production this is creatively selected [13, 20]. We thus present interactive solutions. First, we support directly choosing a and b (*cf*. Sec. 3.1), in contrast to [36]. Secondly, we show mapping with user-provided scribbles on a single reference frame in Fig. 8 (left). Given the input video (top row), we propagate reference frame scribbles (second row, left) to the other frames to get disparity (third row) and results (last row). Figure 8 shows the user mapping with the characters on the screen plane (zero disparities, white), the foreground in front of it (blue), and the background behind it (red).

Mitigation Strategies As a second interaction point, we implement strategies that reduce the inpainting area. Fig. 8 (second column) documents two strategies occasionally used in stereo conversion by comparing the 2D (top left) with the 3D movie version (top right) using an optical flow visualization [2] (bottom): The foreground is slightly enlarged, and the left side of the background is stretched towards the right and vice-versa. Both strategies strongly reduce disocclusions which have to be manually inpainted, at nearly no visual difference. Based on this observation, we implement these two strategies as optional additions into our approach, reducing inpaint areas, where content has to be extrapolated. Since scene content is changed, these can only be applied when generating a novel left and right view. We show results in Fig. 8 (third column) reaching strong reductions of the disocclusion areas with barely noticeable visual changes using a small horizontal displacement dx in the background and foreground zooming. This is also confirmed by a quantitative study on 10 sequences from our test split in Fig. 8 (fourth column).

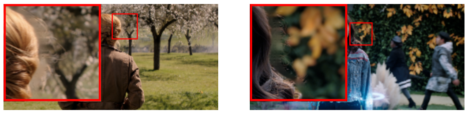


Figure 9. Limitations of our model for thin structures such as hair.

7. Limitations

While giving consistent results in many cases, our model sometimes fails for very thin structures such as hair that might not be correctly displaced, leading to visible artifacts, *cf*. Fig. 9, or if the underlying depth estimation is unstable.

8. Conclusions

We presented an automatic method for stereo conversion that efficiently operates at high resolution and generates results of high visual quality. This is enabled through our disparity-guided warping, compositing, and inpainting as well as integrating information from additional frames. Systematic experiments show the influence of our individual design choices regarding the architecture, losses, and our multi-frame strategy. By additionally integrating interactive extensions, our approach also seamlessly fits into practical stereo conversion workflows. Summing up, our approach advances the state of the art in stereo conversion showing significant advantages over prior work from both stereo conversion and novel view synthesis.

Acknowledgements. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 251654672 – TRR 161 (B04).

References

[1] Karlis Martins Briedis, Abdelaziz Djelouah, Mark Meyer, Ian McGonigal, Markus Gross, and Christopher Schroers. Neural frame interpolation for rendered content. *ACM Transactions on Graphics (TOG)*, 40(6), 2021. 3, 4

[2] Daniel J. Butler, Jonas Wulff, G. B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *Proc. European Conference on Computer Vision (ECCV)*, pages 611–625. Springer LNCS 7577, 2012. 8

[3] Bei Chen, Jiabin Yuan, and Xiuping Bao. Automatic 2d-to-3d video conversion using 3d densely connected convolutional networks. In *Proc. IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 361–367, 2019. 1, 2, 3

[4] Damien Fourure, Rémi Emonet, Elisa Fromont, Damien Muselet, Alain Trémeau, and Christian Wolf. Residual conv-deconv grid network for semantic segmentation. In *Proc. British Machine Vision Conference (BMVC)*. BMVA, 2017. 4

[5] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[6] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 7

[7] Clement Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 270–279, 2017. 1, 2, 3

[8] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page To appear., 2023. 4

[9] Azin Jahedi, Maximilian Luz, Lukas Mehl, Marc Rivinius, and Andrés Bruhn. High resolution multi-scale RAFT (Robust Vision Challenge 2022). In *arXiv preprint 2210.16900*, pages 1–3. arXiv, 2022. 5, 6

[10] Azin Jahedi, Lukas Mehl, Marc Rivinius, and Andrés Bruhn. Multi-scale RAFT: Combining hierarchical concepts for learning-based optical flow estimation. In *Proc. IEEE International Conference on Image Processing (ICIP)*, pages 1236–1240, 2022. 6

[11] Varun Jampani, Huiwen Chang, Kyle Sargent, Abhishek Kar, Richard Tucker, Michael Krainin, Dominik Kaeser, William T. Freeman, David Salesin, Brian Curless, and Ce Liu. Slide: Single image 3d photography with soft layering and depth-aware inpainting. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12518–12527, 2021. 2

[12] Vijayalakshmi Kanchana, Nagabhushan Somraj, Suraj Yadwad, and Rajiv Soundararajan. Revealing disocclusions in temporal view synthesis through infilling vector prediction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3541–3550, 2022. 4

[13] Takashi Kawai, Masahiro Hirahara, Yuya Tomiyama, Daiki Atsuta, and Jukka Häkkinen. Disparity analysis of 3d movies and emotional representations. In *Stereoscopic Displays and Applications XXIV*, volume 8648, pages 293–301. SPIE, 2013. 3, 8

[14] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *Proc. International Conference on Learning Representations (ICLR)*, 2015. 6

[15] Manuel Lang, Alexander Hornung, Oliver Wang, Steven Poulakos, Aljoscha Smolic, and Markus Gross. Nonlinear disparity mapping for stereoscopic 3d. *ACM Transactions on Graphics (TOG, Proc. SIGGRAPH)*, 29(4), 2010. 3

[16] Jiyoung Lee, Hyungjoo Jung, Youngjung Kim, and Kwanghoon Sohn. Automatic 2d-to-3d conversion using multi-scale deep neural network. In *Proc. IEEE International Conference on Image Processing (ICIP)*, pages 730–734, 2017. 1, 2, 3

[17] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6494–6504, 2021. 1, 2

[18] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page To appear., 2023. 1, 2, 7, 8

[19] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page To appear., 2023. 1, 2

[20] Bernard Mendiburu. *3D Movie Making: Stereoscopic Digital Cinema from Script to Screen*. Focal Press, 2009. 3, 8

[21] S. Mahdi H. Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yagiz Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9685–9694, 2021. 7

[22] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proc. European Conference on Computer Vision (ECCV)*, pages 405–4021. Springer LNCS 12346, 2020. 2

[23] Simon Niklaus, Ping Hu, and Jiawen Chen. Splatting-based synthesis for video frame interpolation. In *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 713–723, 2023. 4

[24] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1710, 2018. 4

[25] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5436–5445, 2020. 3, 4

[26] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12179–12188, 2021. 6, 7

[27] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(3):1623–1637, 2022. 3, 6, 7

[28] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. Film: Frame interpolation for large motion. In *European Conference on Computer Vision (ECCV)*, 2022. 7

[29] Mattia Savardi, Alberto Signoroni, Pierangelo Migliorati, and Sergio Benini. Shot scale analysis in movies by convolutional neural networks. In *IEEE International Conference on Image Processing (ICIP)*, pages 2620–2624, 2018. 3

[30] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 6

[31] Pratul P. Srinivasan, Richard Tucker, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[32] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, KublisherngACM Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proc. IEEE/CVF Winter Conference on Applications in Computer Vision (WACV)*, pages 2149–2159, 2022. 4

[33] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 551–560, 2020. 2

[34] Jamie Watson, Oisin Mac Aodha, Daniyar Turmukhambetov, Gabriel J. Brostow, and Michael Firman. Learning stereo from single images. In *Proc. European Conference on Computer Vision (ECCV)*, pages 722–740. Springer LNCS 12346, 2020. 1, 2, 3

[35] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[36] Junyuan Xie, Ross Girshick, and Ali Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *Proc. European Conference on Computer Vision (ECCV)*, pages 842–857, 2016. 1, 2, 3, 6, 8

[37] Wenpeng Xing and Jie Chen. Temporal-mpi: Enabling multi-plane images for dynamic scene modelling via temporal basis learning. In *Proc. European Conference on Computer Vision (ECCV)*, pages 323–338. Springer LNCS 13675, 2022. 3

[38] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 6, 7

[39] Yu Zhang, Dongqing Zou, Jimmy S. Ren, Zhe Jiang, and Xiaohao Chen. Structure-preserving stereoscopic view synthesis with multi-scale adversarial correlation matching. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5860–5869, 2019. 1, 2

[40] Zheyu Zhang and Ronggang Wang. Temporal3d: 2d-to-3d video conversion network with multi-frame fusion. In *Proc. International Conference on Advances in Computer Technology, Information Science and Communications (CTISC)*, pages 1–5, 2022. 1, 2, 3