3-7-2019

High-speed Video from Asynchronous Camera Array

Si Lu
Portland State University, lusi@pdx.edu

Citation Details

High-speed Video from Asynchronous Camera Array

Si Lu

Portland State University

lusi@pdx.edu

Abstract

This paper presents a method for capturing high-speed video using an asynchronous camera array. Our method sequentially fires each sensor in a camera array with a small time offset and assembles captured frames into a high-speed video according to the time stamps. The resulting video, however, suffers from parallax jittering caused by the viewpoint difference among sensors in the camera array. To address this problem, we develop a dedicated novel view synthesis algorithm that transforms the video frames as if they were captured by a single reference sensor. Specifically, for any frame from a non-reference sensor, we find the two temporally neighboring frames captured by the reference sensor. Using these three frames, we render a new frame with the same time stamp as the non-reference frame but from the viewpoint of the reference sensor. Specifically, we segment these frames into super-pixels and then apply local content-preserving warping to warp them to form the new frame. We employ a multi-label Markov Random Field method to blend these warped frames. Our experiments show that our method can produce high-quality and high-speed video of a wide variety of scenes with large parallax, scene dynamics, and camera motion and outperforms several baseline and state-of-the-art approaches.

1. Introduction

Camera arrays have been studied for decades and a variety of camera arrays have been developed in both academic labs and companies. Consumer-level camera arrays are now available. These camera arrays innovate the way of photography and videography, making many tasks easy, such as high-dynamic imaging and refocusing after the fact.

This paper explores camera arrays for high-speed videography by sequentially firing each sensor in a camera array with a small time offset. In this way, a high-speed video can be captured by assembling the recorded frames according to their captured time. A camera array with n lenses, each capturing an m-fps video, can record an mn-fps video. Compared to single-lens high-speed cam-

eras [6, 16, 18, 32], this asynchronous camera array offers a number of advantages. First, a camera array can be made of a number of cheap normal-speed imaging sensors. Second, while the camera array method provides an economic solution for high-speed video capturing, it can be flexibly exploited to integrate multiple high-end high-frame rate cameras to capture videos with even higher frame rates. Third, a camera array can better meet the demand for high data throughput from high-speed imaging than a single-sensor camera as the processing of individual imaging sensors, such as compression, can be highly parallel. Finally, using a single-sensor camera to capture high-speed video limits the exposure time, leading to noisy images. A camera array can increase the exposure time by overlapping the explore duration between consecutive sensors.

As the imaging sensors in a camera array have small spatial baselines, the images from individual sensors must be transformed as if they were imaged from a single reference lens. Early attempts addressed this problem by treating the scene as a plane or assuming the scene is far away from the camera [14, 36, 37]. In this way, images from individual lenses can be transformed and aligned using a global 2D projective transformation (i.e. homography). This method cannot work well in many practical scenarios where the scene exhibits large depth variations. The resulting high-speed video typically suffers from parallax jittering. Alternatively, spatially-varying warping algorithms can be employed to warp these frames. These warping algorithms are more flexible than homography and are able to distribute distortions to visually less salient regions than the others while following a sparse set of motion displacements. These warping algorithms have been shown robust against moderate parallax in a range of applications such as image stitching [22, 38] and video stabilization [24]. However, as these algorithms warp an image as a whole, they will produce undesirable distortions in local regions when parallax is significant, as shown in Figure 1 (c).

This paper presents a novel view synthesis method that employs local spatially-varying warping and multi-label graph cuts to transform source frames as if they were captured from a common reference lens. Specifically, given

Top row: zoomed interpolation results by different apporaches

Bottom row: trajectories of a static point in interpolated videos

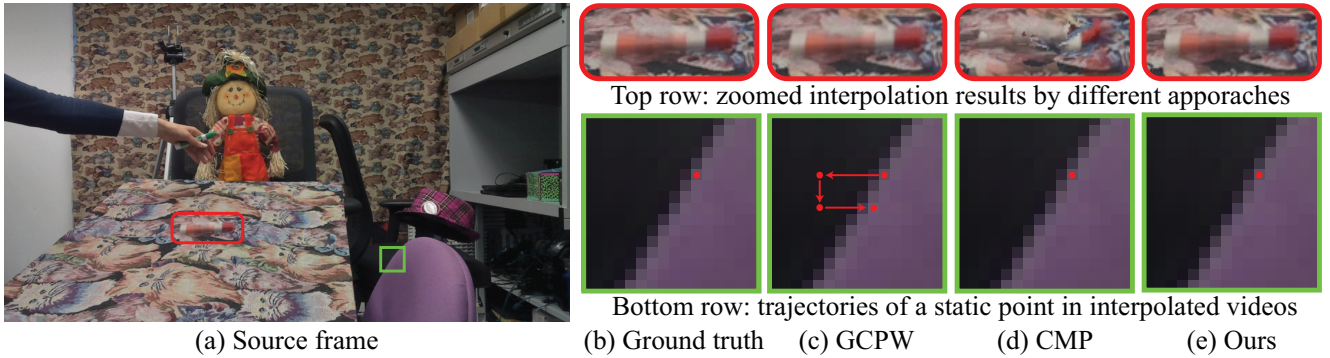| (a) Source frame | (b) Ground truth | (c) GCPW | (d) CMP | (e) Ours |

Figure 1: Frame synthesis for high-speed video. (a): a source frame. (b): ground truth of the interpolated content (top) and the trajectory (bottom) of a static pixel. Global content-preserving warp (GCPW) [24] suffers from parallax jittering in local regions as shown in (c) bottom. A state-of-the-art optical flow-based method (CMP) [17] cannot handle blurry object as shown in (d) top. Our method produces visually plausible results as shown in (e).

any source frame and its two temporally neighboring frames captured by the reference lens, our method partitions them into super pixels. Our method then estimates dense optical flow among any of these three frames to establish correspondence between the super pixels. As dense optical flow estimation is prone to errors, not all the super pixels can be matched across these frames. To address these problems, our method merges the unmatched super pixels with the neighboring matched super pixels. Based on super pixel correspondences, our method employs a local spatially-varying warping algorithm to warp all the super pixels in the three frames to the reference locations according to its time stamp as if they were viewed by the reference camera at that moment. Linearly blending these warped super pixels from three input frames will produce ghosting artifacts. Instead, our method formulates super pixel blending as a multi-label Markov Random Field problem that properly chooses the right blending schemes for pixels to achieve visually pleasing blending results while avoiding ghosting artifacts.

This paper contributes a method that explores the increasingly available camera array to produce high-speed video. The key enabling algorithm is a high-quality novel view synthesis algorithm that transforms video frames captured by spatially-distributed lenses as if they were captured by a common lens to avoid parallax jittering. This novel view synthesis algorithm integrates local spatially-varying warping and multi-label MRF optimization to produce a plausible novel view from multiple frames while avoiding ghosting artifacts and handling parallax. Our experiments also show that our method can produce high-quality and high-speed video of a wide variety of scenes with scene dynamics, parallax, and camera motion.

2. Related Work

This work falls into the area of frame interpolation [4] and novel view synthesis [20, 21]. A complete overview of this area is out of scope of this work. We discuss the work that directly related to this paper.

A typically video frame interpolation method estimates dense correspondence using optical flow between two consecutive frames and follows the optical flow to interpolate one or multiple frames in between them [4]. This method, however, can fail due to the difficulty of optical-flow estimation. While traditional optical flow methods [3, 5, 7, 9, 35] do not work well at object boundaries or in textureless regions, several edge-aware approaches [23, 33, 34] based on edge and feature mapping have been proposed. While these methods achieve better interpolation results at object boundaries, they can not handle large motion. Later, optimization-based approaches [8, 11] are developed according to different rules to deal with large motion and can generate appealing optical flow results. However, flow errors can still occur and lead to noticeable visual artifacts when using flow-based frame interpolation due to occlusion. Meyer *et al*. developed phase-based interpolation methods that requires no flow computation and modifies the phase difference to produce intermediate frames [28, 27]. These phase approaches produce impressive results; however, it is unclear how to employ these approaches to incorporate the extra frame in our problem for better interpolation. Niklaus *et al*. learned adaptive CNN [30] and content-aware CNN [29] to predict intermediate frames and achieves state-of-the-art performance. However, those method can not handle scenes with fast moving objects or large motion. Jiang *et al*. proposed Super slomo [19], a frame work that uses a U-Net architecture to pre-compute bi-directional optical flows and fuse them to generate intermediate bi-directional optical flows at the target time stamps. Then for any time stamp, they used another U-Net to properly fuse two warped frames from both forward and backward input frames. Our method differs from those frame interpolation methods in that we have extra frames that are captured at the same time but from different viewpoints as the frames to be interpolated. Therefore, we interpolate from these frames and we employ a multi-label graph cut algorithm to decide an optimal blending scheme to make optimal use of extra frames.

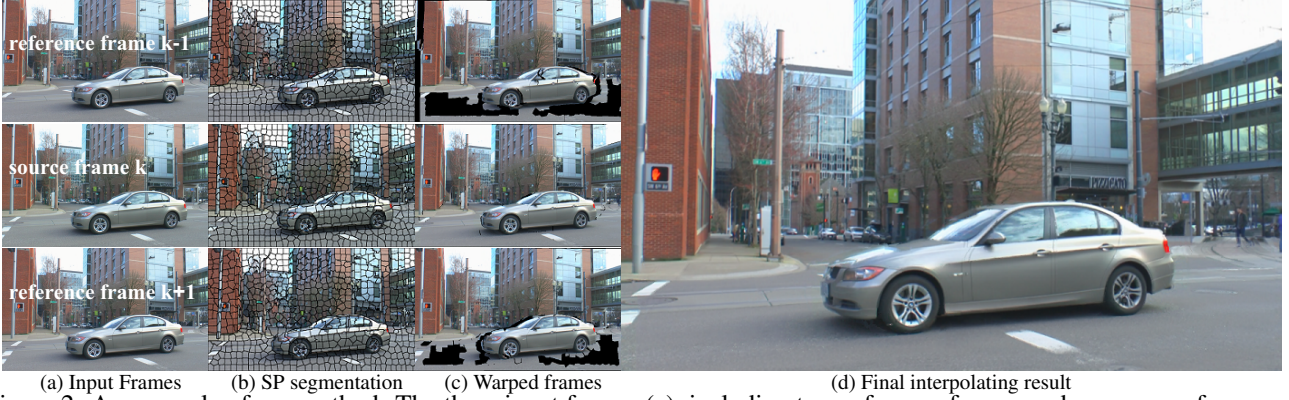| (a) Input Frames | (b) SP segmentation | (c) Warped frames | (d) Final interpolating result |

Figure 2: An example of our method. The three input frames (a), including two reference frames and one source frame, are over-segmented into superpixels (SP) (b), locally warped to the target position (c), and blended using our multi-label based optimization scheme (d).

Our problem can also be formulated as a video stabilization problem if we consider frames as captured by a regular camera moving along a zigzag path periodically. While a variety of video stabilization methods are now available [25], directly applying them to our problem is insufficient due to the highly patterned zigzag path, especially with large depth variation in the scene. As traditional homography stabilization approaches [26, 36] fail on those video scenes, Liu *et al.* [24] propose spatially varying warp to handle moderate depth variation. We tried to apply the content-preserving warping based approach to stabilize such a sequence. As reported in our experiment, the result still looks jittering in some local regions.

Novel view synthesis methods for camera arrays are also related to our work [25, 39]. They estimate 3D scenes from captured images, warp and blend them to create novel views. Our method is most related to Chaurasia *et al.* [10]. This approach over-segments the input images into superpixels and synthesizes depth for challenging regions with poor depth estimation using similar neighbouring superpixels and warp each super-pixel individually. In this paper, we propose a frame interpolation approach to transform frames captured by an asynchronous camera array into a high-speed video. However, our input frames from different viewpoints are not taken at the same time, making it difficult to estimate depth for moving objects. Instead, we use optical flow as warping guidance and propose a validation process to eliminate bad warping guidance. We also propose a super-pixel merging scheme to propagate high quality warping guidance to nearby regions. More importantly, instead of blending all the warped frames as weighted average, our method formulates the subset selection of warped pixels for blending as a multi-labeling problem and employs a Markov Random Field method to optimize the selection to produce a visually plausible novel view.

3. Methodology Over View

Our method has two main steps: *optical flow guided local warp* and *graph cuts-based multi-label rendering*, as shown in Figure 2. Given a set of alternatively captured video frames by n lenses in a camera array, we consider the camera with the latest firing order at each shooting iteration as reference and other cameras as sources. We aim to transform the sources as if they were captured by the reference lens. We transform source frames one by one independently, therefore, the n-lens camera array problem can be simplified as a sequence of two-lens camera array ones. Without loss of generality, this section focuses on a two-lens camera array.

After we assemble the frames captured by an asynchronous two-lens camera array, we obtain a frame sequence $V = I_1^r, I_2^s, I_3^r, I_4^s, \cdots, I_{k-1}^r, I_k^s, I_{k+1}^r, \cdots$. Given two consecutive frames captured by the reference lens, I_{k-1}^r and I_{k+1}^r and a source frame I_k^s between these reference frames, our goal is to generate a synthesized frame as if it was captured by the reference lens at time stamp k. We first compute a set of dense pixel correspondences using SparseFlow [2], including the forward and backward optical flow between the source frames $(F_{k,k-2}^s, F_{k,k+2}^s)$, between the reference frames $(F_{k+1,k-1}^r, F_{k+1,k+3}^r)$, from the source frame to its two temporal neighbouring reference frames $(F_{k,k+1}^{sr}, F_{k,k-1}^{sr})$, and the ones from the two reference frames back to the source frame $(F_{k-1,k}^{rs}, F_{k+1,k}^{rs})$. We then over-segment [1] the three input frames into superpixels according to both pixel intensities and the estimated flow magnitudes. Note that our approach is independent of the choice of optical flow and segmentation approaches. In addition, since we use image-based rendering guided by estimated optical flows, additional geometry information between source and reference cameras is not needed.

Optical Flow Guided Local Warp. Given the three input frames with estimated optical flow, we aim to warp them to a target temporal position for final rendering. Given a pixel with its estimated optical flows as well as the time stamp in-

formation, we compute its corresponding positions in other views. Thus, each pixel with good optical flow could be considered as a feature point across multiple views and could be used to guide the frame warping. However, as optical flow often contains errors, especially in occluded/dis-occluded and blurred regions, we validate each pixel's flow using a simple but effective intensity matching approach to generate an optical flow weight map W for each input frame. Only pixels with high quality optical flow (large weight in W) are selected to guide the warp. For super-pixels with few good pixel correspondences (optical flow), a merging process is applied to merge such super-pixels to their neighbours with good pixel correspondences. This allows neighbouring super-pixels to guide the warp of such *bad* super-pixels.

A global content-preserving warp [24] can then be used to warp each input frame. However, this method can produce undesirable distortions when parallax is significant. We then follow the approach from Chaurasia *et al.* [10] to warp each superpixel individually to allow plausible warping results.

Rendering. A simple averaging approach could be used to generate the final rendering result. However, this might introduce undesirable visual artifacts, such as blurring and ghosting artifacts. While the three input frames are warped to the same temporal and spatial position, they still contain errors, such as intensity discontinuities caused by blurry objects and warping of occluded/dis-occluded regions. In addition, holes can exist due to the superpixel wise warp procedure. Thus, for each rendering pixel, the selection of its three warped sources should be carefully considered. Noticing that there are $2^3 = 8$ combinations of selection for each rendering pixel, we consider the selections of all rendering pixels as a labeling problem. We consider the whole rendering frame as an un-directed graph and use a graph cuts based multi-label energy minimization technique [15] with properly designed data term and smoothness term to solve the labeling problem.

Given the optimized labels for each rendering pixel, we weighted average the corresponding selected warped pixels according to optical flow validated weights W. Finally, we use Poisson Blending to fill the rest holes in the rendering result. In the next subsections, we will describe our *optical flow guided local warp* and *graph cuts-based multi-label rendering* in details.

4. Optical Flow Guided Local Warp

Our input is a set of three video frames, including two neighbouring reference frames (I_{k-1}^r, I_{k+1}^r) captured by reference camera at time $t_r = 0, 1$, respectively, and one source frame (I_k^s) captured by the source camera at time $t \in (0, 1)$. Our goal is to warp all the three input frames to the same temporal position as if they were imaged from the

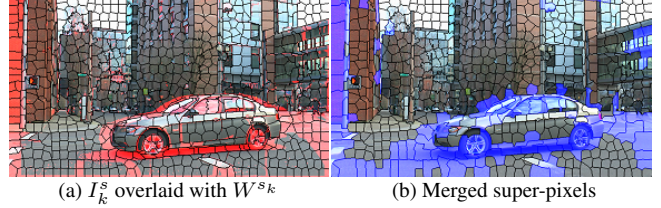(a) I_k^s overlaid with W^{s_k} (b) Merged super-pixels

Figure 3: (a): Super-pixels near frame boundary or in occluded/dis-occluded regions often have optical flow with poor qualities (marked in red). (b): Super-pixels with low quality optical flow are merged to nearby super-pixels to form merged super-pixels (marked in blue) with enough pixels having good optical flow guidance.

reference camera at a time spot in-between I_{k-1}^r and I_{k+1}^r with a temporal interpolating parameter $t \in (0, 1)$.

4.1. Optical Flow Validation

Optical flow estimation results often contain errors due to the existence of blurred moving objects, occluded/dis-occluded regions and parallax effects. As shown in Figure 3 (a), these optical flow estimation, with poor accuracy, need to be excluded for latter warp guidance. We thus propose an intensity patch matching approach to effectively validate optical flow estimation for each pixel in the input frames. Generally, given a pixel $p_{i_1,j_1}^{m_1}$ in one of the three input frames, we first search for its corresponding pixels $p_{i_2,j_2}^{m_2}, p_{i_3,j_3}^{m_3}$ in the other two frames according to the estimated optical flow. To validate the optical flow estimation for pixel $p_{i_1,j_1}^{m_1}$, we compare the corresponding patches centered at these three pixels as follows,

$$d_c = max(||P_{i_1,j_1}^{m_1} - P_{i_2,j_2}^{m_2}||^2, ||P_{i_1,j_1}^{m_1} - P_{i_3,j_3}^{m_3}||^2) \quad (1)$$

where $P_{i_n,j_n}^{m_n}$ is the patch centered at $p_{i_n,j_n}^{m_n}$ with $n = 1, 2, 3$ and $m_n \in \{r_{k-1}, r_{k+1}, s_k\}$ indicating the three input frames, respectively. The optical flow validation weight is then computed as $W_{i,j}^m = e^{-(d_c^2/2\sigma_m^2)}$, where σ_m is a pre-selected parameter. This allows us to assign high weight to a pixel if and only if it is similar to both corresponding pixels in the other two frames. To further exclude outliers, we follow the approach from Baker *et al.* [4] to add a forward/backward optical flow check.

4.2. Superpixel Merging

Our assumption is that neighbouring super-pixels are more likely to share similar motions. We thus merge bad super-pixels to their neighbours with good flow estimation and essentially let neighbouring super-pixels to guide the warp. Here, we define *good* superpixel as a superpixel that has more than h (with a default value 100) pixels with optical flow validated weights larger than a threshold th_{sp} (with a default value 0.96). Specifically, for each bad superpixel S_b, we start from a queue Q containing S_b and search via

expanding. At each step, we look at neighbouring super-pixels of all super-pixels in Q and en-queue either a good neighbouring superpixel with good flow estimations or a bad neighbouring superpixel with the smallest motion difference to S_b if no good neighbouring super-pixels exists. We repeat this step until at least one good superpixel is added to Q. All searched super-pixels in Q are then combined to form a merged superpixel.

Note that our superpixel merging is only applied to source frames. This means that we try to warp all pixels in the source frames as scenes in these frames are captured at the interpolating time. For the reference frames, we simply do not warp bad pixels as they might conflict with their correspondences in the warped source frames.

4.3. Local Content-Preserving Warp

We now have the pre-validated optical flow and modified superpixel segmentation for all three input frames. We aim to warp each superpixel S in the input frame to a target position $\hat{S}$ in the warped frame. Specifically, for each superpixel in the input frame, we construct an axis-aligned bounding box and divide it into regular mesh grids with its vertices represented as V. The warping of a superpixel can then be formulated as a mesh warping problem, where the unknowns are the corresponding grid vertices $\hat{V}$ in the warped frame. This mesh warping problem can then be solved as an optimization problem with a data term $E_{p,m}$ that encourages pixels to be re-projected to its potential locations for each feature point P_m and a smoothness term or a similarity term $E_s(\tilde{V}_T)$ for vertices V that aims to preserve local image structures. Please refer to [24] for the derivation of those two terms. We then compute the final energy term as

$$E = \alpha \sum_m E_{p,m} + \sum_T E_s(\tilde{V}_T) \qquad (2)$$

where α is the weight for the data term with a default value 0.5 for features in homogeneous regions and 1 for features at edge points. The minimization of E is solved by constructing a linear system and solving it using standard sparse linear solver. The final warping result is rendered using texture mapping according to the output mesh.

5. Labeling-based Frame Rendering

The three input frames are now warped to the same temporal position. However, directly blend them together might introduce visible visual artifacts because warping holes and mis-matches still exist. For each rendering pixel in the final result, a subset selection of its three warped pixels (denoted as $p_k^s, p_{k-1}^r, p_{k+1}^r$) needs to be made. As there are $2^3 = 8$ combinations of selections (as shown is Table 1) for each rendering pixel, we formulate the decision making of all rendering pixels as a labeling problem, where each pixel is to be assigned one of the 8 labels, where each label $L = (l_k^s, l_{k-1}^r, l_{k+1}^r)$ contains three binary numbers indicating the selection of each warped pixel. Note that for pixels

Table 1: Eight labels for each rendering pixel

label No.	1	2	3	4
notation	$(0,0,0)$	$(1,0,0)$	$(0,1,0)$	$(0,0,1)$
selection	none	p_k^s	p_{k-1}^r	p_{k+1}^r
label No.	5	6	7	8
notation	$(0,1,1)$	$(1,0,1)$	$(1,1,0)$	$(1,1,1)$
selection	p_{k-1}^r, p_{k+1}^r	p_k^s, p_{k+1}^r	p_k^s, p_{k-1}^r	all

assigned with label 1 with $L = (0,0,0)$, leading to holes in the final rendering results, we use Poisson image inpainting [31] with zero gradient to infill them.

We consider the final rendered frame as an un-directed graph in which each rendering pixel is represented as a node and each pair of spatially neighbouring pixels are connected by an un-directed edge. This labeling problem can then be effectively solved using a graph cuts based multi-label energy minimization technique from Fulkerson *et al.* [15]. Given the optimized labels for each pixel, we weighted average the corresponding selected subset of three warped pixels using optical flow validated weights W.

Labeling Data Term. Following a statistic rule that more similar samples lead to better reconstruction results, we define the labeling data term for all rendering pixels (p) as

$$E_{data}^L = \sum_p (E_{ind}^{L,p} + \beta_L \cdot E_{sim}^{L,p}) \cdot Z_{L,p}^{-1} \qquad (3)$$

where $Z_{L,p}$ is a normalizing factor that encourages more selected samples and is defined as

$$Z_{L,p} = (l_k^s + l_{k-1}^r + l_{k+1}^r + \epsilon)_p^{\alpha_L} \qquad (4)$$

where ϵ is a small constant ($\epsilon \ll 1$). $E_{ind}^{L,p}$ gives credits to each individual selected pixels and is defined as

$$E_{ind}^{L,p} = (K_k^s \cdot l_k^s + K_{k-1}^r \cdot l_{k-1}^r + K_{k+1}^r \cdot l_{k-1}^r)_p \qquad (5)$$

where $K_k^s = 1$ and $K_{k-1}^r = K_{k+1}^r = 1.5$. They are parameters that control the weights for pixels from source frames and reference frames. We set a smaller K_k^s value than the other two to prefer single selection from the source frames to reference frames. This is because scenes warped from the source frame is captured at the same time as the interpolating time stamp.

$E_{sim}^{L,p}$ is a similarity measuring term that penalizes large intensity difference between two pixels if both of them are selected. We thus define $E_{sim}^{L,p}$ as follows.

$$E_{sim}^{L,p} = l_k^s l_{k-1}^r d(p_k^s, p_{k-1}^r) + l_k^s \cdot l_{k+1}^r d(p_k^s, p_{k+1}^r) + l_{k-1}^r \cdot l_{k+1}^r d(p_{k-1}^r, p_{k+1}^r) \qquad (6)$$

where $d(a,b)$ is the $l2$ norm difference of two pixels a and b. This term encourages that only similar pixels are preferred to be added to the final selection. α_L and β_L are two controlling parameters with default values of 3 and 8.

Note that for a pixel p, it could be possible that not all 8 labels are valid due to the existence of warping holes.
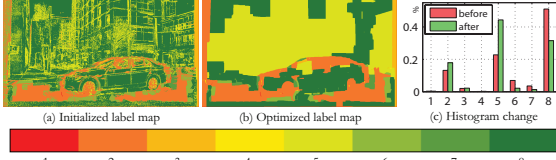
Figure 4: Graph cuts-based labeling. (a): Initialized label map. (b): The final optimized label map by our method. (c): Label histogram comparison before and after optimization.

For example, if no pixel is warped to some location in the warped source frame, then pixel p_k^s can not be selected at this location as it does not exist. Thus, labels $(1, 0, 0), (1, 0, 1), (1, 1, 0)$ and $(1, 1, 1)$ are all invalid labels. For these invalid labels, we directly assign a large labeling data term to avoid invalid label selection $E_d^{L,p} = \infty$.

Labeling Smoothness Term. Neighboring pixels are more likely to have the same labels. We thus define the smoothness term as the $l2$ norm of label differences.

$$E_{smooth}^{L,p,q} = ||L_p - L_q||^2 \qquad (7)$$

where L_p and L_q are two labels for a pair of connected rendering pixels. The final smoothness term for all neighboring pixels are then defined as follows.

$$E_{smooth}^L = \sum_{p,q} E_{smooth}^{L,p,q} \qquad (8)$$

The final energy function is then defined as

$$E^L = E_{data}^L + \gamma_L \cdot E_{smooth}^L \qquad (9)$$

where γ_L is a controlling parameter with default value 2. After all labels are obtained via optimization, each of the final rendering pixels can be computed as a weighted average of all selected warped pixels.

We first initialize the label map for each pixel by selecting the label that minimizes the data term at current position, as shown in Figure 4 (a). We then use graph cuts multi-label optimization [15] to get the final label map.

As the source and reference frames compliment each other, the optimization allows our approach to make good use of them. The warped source frames capture what is really going on at the current time stamp. They thus have better quality in occluded/dis-occluded or blurred regions. As shown in Figure 4 (b), in most occluded/dis-occluded regions, our labeling optimization selects pixels only from the source frames (indicating the selection of label 2 in Table 1). However, as the source frames are imaged from a slightly different view point to the reference frames, they often suffer from parallax jittering effects. Thus, the warped reference frames have generally better qualities in such regions. From Figure 4, it can be seen that in most regions in the background, the combination of pixels from the two reference frames are preferred (indicating the selection of label 5 in Table 1). The fusion of the two types of frames thus makes our approach robust against various types of scenes.
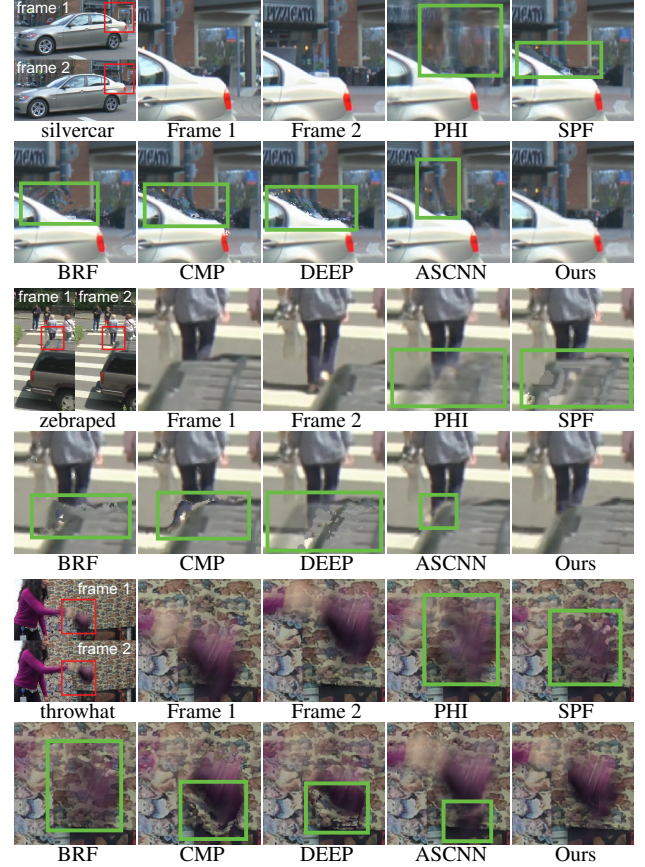

Figure 5: Comparison to single-lens interpolation methods.

Statistically, it can be seen from Figure 4 (c) that our optimization replaces part of label 8 (selection of all three pixels) with label 5. Overall, our optimization does not change the labels' distribution significantly while preserving more neighbouring smoothness.

The warping of all input frames can create holes due to the existence of dis-occluded regions. We fill these holes using Poisson image inpainting [31]. We follow the approach from Chaurasia et al. [10] to assign zero gradients to these pixels for inpainting.

6. Results

We evaluate our approach using videos from RMIT3DV dataset [12], Choubassi et al. [13], adtv.at and videos captured by our own cameras as well as simulated videos generated using Maya 2016. Choubassi's dataset consisting of videos captured using 2 by 2 camera arrays, the Maya-simulated videos are captured using a virtual 3 by 3 camera array and all other videos are captured using 2 by 1 camera arrays. These videos contain a wide variety of scenes, including indoor and outdoor scenes with various levels of motion. There are also challenging scenes with parallax, large camera motion and blurred moving objects.
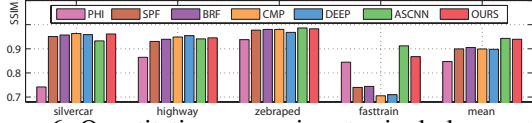
Figure 6: Quantitative comparison to single-lens methods.

Comparison with single-lens frame interpolation. We compare our method to six recent single-lens frame interpolation methods. We compare to Phase-Based method (PHI) [28] and Adaptive Separable Convolution (AS-CNN) [30] using the authors' implementations. We also compare our method to optical flow-based methods, including SparseFlow (SPF) [2], BroxFlow (BRF) [7], CMPFlow (CMP) [17] and DeepFlow (DEEP) [34]. For all optical flow algorithms, we use the codes provided by the authors. To interpolate the in-between images given the estimated flow fields, we use the code provided by the author of the Middlebury interpolation benchmark [4].

In Figure 5 we visually compare our interpolation results on scenes with large motion or blurred moving objects. PHI introduces additional blur to moving contents as high frequency contents cannot be represented by phase estimation. Optical flow-based methods produce distortions at occluded/dis-occluded regions. Optical flow-based methods also fail to interpolate blurred moving objects. They tend to blend the foreground and background as they ignore the blurred moving objects and consider them as static. CMP and DEEP produce better optical flow estimations. However, they introduce serious artifacts at moving boundaries. In contrast, our approach can generate visual plausible results by invalidating the guidance of incorrect flow estimations in the local warp and let nearby superpixels to help with the warp. ASCNN can generate good results for scenes with small or moderate motion and occlusion. However, for scenes with large motion (as shown in *slivercar* in Figure 5) or occlusion (as shown in *throwhat* in Figure 5), ASCNN introduces noticeable visual artifacts while our approach can still generate plausible interpolated frames for those challenging scenes. This observation can also be confirmed by results shown in Figure 8.

We quantitatively test our method on 4 videos (as shown in Figure 5 and Figure 9) with ground truth, which are obtained using the leave-some-out method. Specifically, we interpolate intermediate frames and compare them to the original ones. We report the perceptually motivated structural similarity (SSIM) in Figure 6. We also report the Mean Square Error (MSE) in the supplementary video due to space limit. In general, our approach has comparable quantitative performance to ASCNN and outperforms
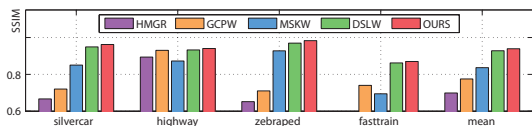

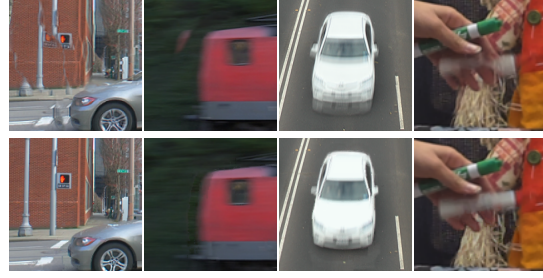Figure 7: Quantitative comparison to warp-based methods.

Figure 8: Visual comparison between the ASCNN [30] (top row) and our approach (bottom row) on challenging scenes with large motion and blurry moving objects.
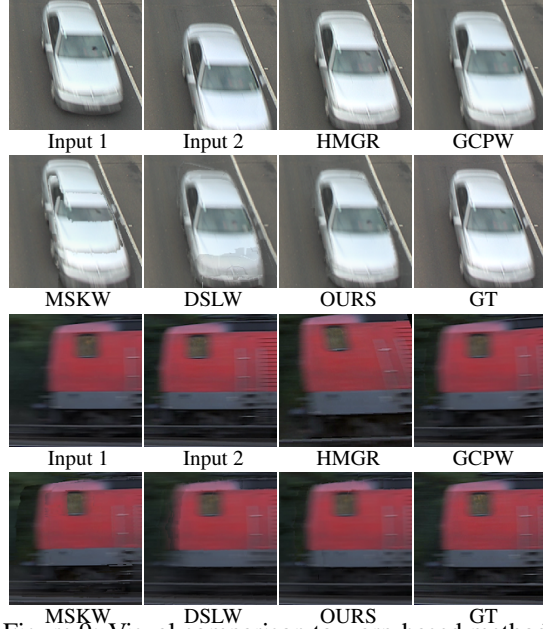

Figure 9: Visual comparison to warp-based methods.

other competing methods. As shown in Figure 5 and Figure 8, Our method performs significantly better in challenging places such as occluded regions, blurry and fast moving objects and regions with large parallax. While those regions often occupy a small portion of the scenes, leading to limited overall quantitative improvements, they have large impact on the visual qualities, please refer to our supplementary material for details.

Comparison with warp-based methods. We compare our method to four warp-based methods, include homography transformation (HMGR), global content-preserving warp (GCPW) [24], depth synthesis and local warps (DSLW) [10] and mask-based warps (MSKW) from Choubassi et al. [13]. We show the visual comparison in Figure 9. Comparing to DSLW and MSKW, our approach generates interpolating results with less visual artifacts (duplication/blur) as our approach effectively eliminates bad optical flow for warping guidance. While HMGR and GCPW tend to have less visual artifacts as they globally warp the source frames to the reference, they often suffer from parallax jittering, as can be seen in the second example
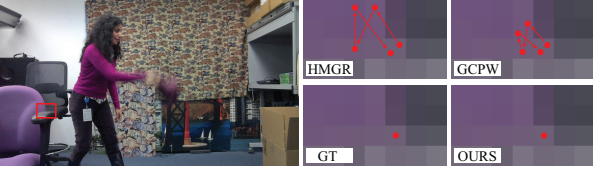
Figure 10: HMGR fails to align the static background features. GCPW performs better, but still suffers from moderate parallax jittering. Our result properly maintains temporal coherence according to the ground truth (GT).
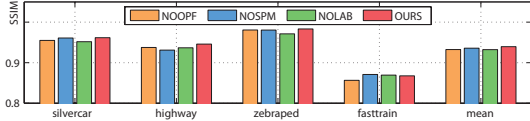

Figure 11: Quantitative leave-one-out component analysis.

(a) OPF$_{out}$ (b) SPM$_{out}$ (c) LAB$_{out}$ (d) Our result

Figure 12: Leave-one-out evaluation of our method.

in Figure 9. To further verify this we compare our method to those two approaches by tracking static feature trajectories across consecutive interpolated frames in Figure 10. It can be seen that the trajectory in our result better preserves temporal coherence and is closer to the ground truth trajectory (GT). The difference becomes more obvious in the supplementary video. The plot in Figure 7 shows that our approach quantitatively performs better than other warp-based interpolation methods. This observation can also be confirmed by the MSE comparison in our supplementary video.

Component analysis. As our main contribution is an optical flow validation and a superpixel merging for local content preserving warp as well as a labeling-based frame rendering to blend multiple warped frames, we analyze these components by conducting leave-one-out experiments on them. Specifically, to leave the optical flow validation out (OPF$_{out}$), we set $W = 1$ for all pixels. To leave out the superpixel merging (SPM$_{out}$), we simply skip it. To leave out labeling-based frame rendering, we replace it with a simple averaging scheme as used by Chaurasia *et al.* [10] (LAB$_{out}$). The plot in Figure 11 shows quantitative degradation when leaving out any of those components. For local content preserving warp, our optical flow validation effectively removes flow outliers and superpixel merging assigns reasonable good flow for bad superpixels from neighbouring regions. The following labeling-based frame rendering then makes the best use of all individually warped frames to attenuates errors by letting them complement each other. As shown in Figure 12, leaving out any component would introduce noticeable visual artifacts.

Implementation The proposed approach is implemented using C++ and MATLAB on a desktop with a 4-core In-

(a) Frame 1   (b) Frame 2   (c) Interpolated

Figure 13: Thin structures that can not full-fill a single superpixel is blurred by our method.

(a) 1 frame out   (b) 3 frames out   (c) Ground truth

Figure 14: Our method can handle relatively large motion (a), but can still fail when the motion becomes too large(b).

tel i7-4770 3.40GHz CPU. This unoptimized off-line implementation takes an average of 87.34 seconds to synthesize a frame of size 720×396.

Discussion and limitations. While our approach can generate plausible interpolated videos, it has some limitations. Although our approach can handle parallax for moderately small objects in different depth, it introduces some blurring when the foreground objects are too small to be covered by a single superpixel. As can be seen in Figure 13, the final interpolated antenna is blurred as the local warp is mainly guided by flow in the background regions in its corresponding superpixels. In addition, while our approach can deal with large motion, it can fail when the motion becomes too large. As shown in Figure 14 (a), our method is able to generate interpolated frame with reasonably good quality for scenes with a foreground motion of about -35 pixels and background motion of 70 pixels. However, when we double the motion by leaving more frames out to synthesize the input frames, noticeable visual artifacts occur in the interpolated result, as shown in Figure 14 (b).

7. Conclusion

In this paper, we propose a warping-based method to generate high frame rate videos using an asynchronous low frame rate camera array in which the video frames are alternatively captured by each camera. We first over-segment the input frames into superpixels, we locally warp each individual superpixel from the source frames to the reference with the help of validated optical flow fields and modified superpixel maps in which superpixels with poor flow estimations are merged to nearby neighbours. By utilizing the fusion of both the current source frame and temporally neighbouring reference frames using a graph cuts-based optimization labeling, our approach can produce plausible high-speed videos with high qualities on a variety of scenes with different levels of motions.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 2012.

[2] A. Ayvaci, M. Raptis, and S. Soatto. Sparse occlusion detection with optical flow. *IJCV*, 2012.

[3] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 2004.

[4] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 2011.

[5] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer vision and image understanding*, 1996.

[6] G. Bloch and T. Sattelmayer. Effects of turbulence and secondary flows on subcooled flow boiling. *Heat and Mass Transfer*, 50(3):427–435, 2014.

[7] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004.

[8] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 2011.

[9] A. Bruhn and J. Weickert. Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In *ICCV*, 2005.

[10] G. Chaurasia, S. Duchene, O. Sorkine-Hornung, and G. Drettakis. Depth synthesis and local warps for plausible image-based navigation. *ACM Transactions on Graphics (TOG)*, 2013.

[11] Q. Chen and V. Koltun. Full flow: Optical flow estimation by global optimization over regular grids. In *CVPR*, 2016.

[12] E. Cheng, P. Burton, J. Burton, A. Joseski, and I. Burnett. Rmit3dv: Pre-announcement of a creative commons uncompressed hd 3d video database. In *Quality of Multimedia Experience (QoMEX), 2012 Fourth International Workshop on*, pages 212–217. IEEE, 2012.

[13] M. El Choubassi and O. Nestares. Stabilized high-speed video from camera arrays. *Electronic Imaging*, 2017(15):7–13, 2017.

[14] C. Etheredge. Goslow: Design and implementation of a scalable camera array for high-speed imaging. Master's thesis, University of Twente, 2016.

[15] B. Fulkerson, A. Vedaldi, S. Soatto, et al. Class segmentation and object localization with superpixel neighborhoods. In *ICCV*, 2009.

[16] U. Gülan, B. Lüthi, M. Holzner, A. Liberzon, and W. Kinzelbach. Experimental analysis of the lagrangian flow field in an ascending aorta by particle tracking velocimetry. In *5th European Conference of the International Federation for Medical and Biological Engineering*, pages 595–598. Springer, 2011.

[17] Y. Hu, R. Song, and Y. Li. Efficient coarse-to-fine patchmatch for large displacement optical flow. In *CVPR*, 2016.

[18] I. Ishii, T. Tatebe, Q. Gu, Y. Moriue, T. Takaki, and K. Tajima. 2000 fps real-time vision system with high-frame-rate video recording. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1536–1541. IEEE, 2010.

[19] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[20] S. B. Kang, Y. Li, X. Tong, and H.-Y. Shum. Image-based rendering. *Foundations and Trends in Computer Graphics and Vision*, 2(3):173–258, 2006.

[21] J. Kopf, F. Langguth, D. Scharstein, R. Szeliski, and M. Goesele. Image-based rendering in the gradient domain. *ACM Transactions on Graphics (TOG)*, 2013.

[22] W.-Y. Lin, S. Liu, Y. Matsushita, T.-T. Ng, and L.-F. Cheong. Smoothly varying affine stitching. In *IEEE CVPR*, pages 345–352, 2011.

[23] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 2011.

[24] F. Liu, M. Gleicher, H. Jin, and A. Agarwala. Content-preserving warps for 3d video stabilization. *ACM Transactions on Graphics*, 2009.

[25] S. Liu, P. Tan, L. Yuan, J. Sun, and B. Zeng. Meshflow: Minimum latency online video stabilization. In *ECCV*, 2016.

[26] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum. Full-frame video stabilization with motion inpainting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1150–1163, 2006.

[27] S. Meyer, A. Djelouah, B. McWilliams, A. Sorkine-Hornung, M. Gross, and C. Schroers. Phasenet for video frame interpolation. In *CVPR*, June 2018.

[28] S. Meyer, O. Wang, H. Zimmer, M. Grosse, and A. Sorkine-Hornung. Phase-based frame interpolation for video. In *CVPR*, 2015.

[29] S. Niklaus and F. Liu. Context-aware synthesis for video frame interpolation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[30] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive convolution. In *ICCV*, 2017.

[31] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM Transactions on Graphics (TOG)*, 2003.

[32] T. S. Perry. Sanstreak lowers the cost of high-speed photography [resources_startups]. *IEEE Spectrum*, 52(3):27–27, 2015.

[33] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015.

[34] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: Large displacement optical flow with deep matching. In *ICCV*, 2013.

[35] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic huber-l1 optical flow. In *BMVC*, 2009.

[36] B. Wilburn, N. Joshi, V. Vaish, M. Levoy, and M. Horowitz. High-speed videography using a dense camera array. In *CVPR*, 2004.

[37] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. In *ACM Transactions on Graphics (TOG)*, pages 765–776. ACM, 2005.

[38] F. Zhang and F. Liu. Parallax-tolerant image stitching. In *CVPR*, 2014.

[39] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. In *ACM Transactions on Graphics (TOG)*, 2004.