Modeling Blurred Video with Layers

Jonas Wulff and Michael Julian Black

Max Planck Institute for Intelligent Systems, Tübingen, Germany
{jonas.wulff,black}@tue.mpg.de

Abstract. Videos contain complex spatially-varying motion blur due to the combination of object motion, camera motion, and depth variation with finite shutter speeds. Existing methods to estimate optical flow, deblur the images, and segment the scene fail in such cases. In particular, boundaries between differently moving objects cause problems, because here the blurred images are a combination of the blurred appearances of multiple surfaces. We address this with a novel layered model of scenes in motion. From a motion-blurred video sequence, we jointly estimate the layer segmentation and each layer's appearance and motion. Since the blur is a function of the layer motion and segmentation, it is completely determined by our generative model. Given a video, we formulate the optimization problem as minimizing the pixel error between the blurred frames and images synthesized from the model, and solve it using gradient descent. We demonstrate our approach on synthetic and real sequences.

Keywords: Optical Flow, Layers, Object Boundaries, Motion Blur.

1 Introduction

Common goals in the analysis of video include the estimation of optical flow, the localization of motion boundaries, and the segmentation of images into regions corresponding to objects. These are all hard problems that become even harder when the frames contain motion blur. However, when a dynamic scene is captured by a camera with finite shutter speed, this will always be the case. In such a setting, traditional assumptions of brightness constancy are violated, particularly at motion boundaries where the pixel values combine information from multiple surfaces blurring into each other.

To address these problems, we propose a novel layered model of images that explicitly models the motion of the layers, the blur induced by this motion, and the un-blurred appearance of each layer (Fig. 1). A key observation is that both the motion blur and the displacement of a surface are results of the same process in the world: the motion of the surface relative to the camera. Hence, the motion blur of a surface is completely determined by the motion of that surface – *estimating the optical flow gives us the blur kernel.* Unlike previous work we formulate this as a generative model and jointly solve for all unknowns. This produces an accurate layer segmentation and precise motion boundaries from a motion-blurred image sequence (Fig. 1(c) and (e)). To this end, layers

D. Fleet et al. (Eds.): ECCV 2014, Part VI, LNCS 8694, pp. 236–252, 2014.

provide a natural framework because they directly model surface interaction at occlusion boundaries. Here we focus on parametric motion within layers and use a two-layer model, as is common in recent approaches [36]. While being limited in terms of motion complexity, we nevertheless find that this model is able to analyze real scenes with different foreground and background motion.

Much of the work on motion blur focuses on the problem of *deblurring*, particularly in single images where the blur is caused by camera shake [13]. These approaches either assume homogeneous blur across the whole image or restrict the blur kernels to those caused by common camera motion paths. On the other hand, existing work on deblurring in the case of object motion is restricted to the case of static backgrounds [2]. We do not address single-image deblurring, but focus on optical flow estimation in sequences with motion blur and motion of both the foreground and the background. In particular, we deal with spatially-varying blur kernels that are determined by the layer motions.

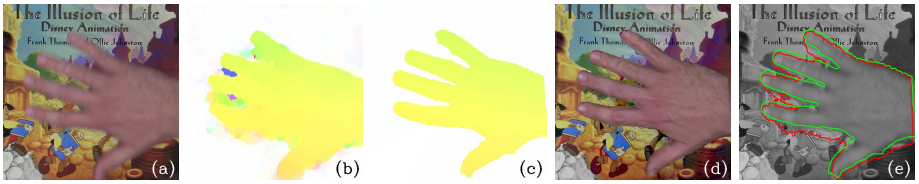


Fig. 1. When computing optical flow from motion blurred video (a), existing methods [44] fail at object boundaries ((b) and (e), red). Our method is able to accurately estimate optical flow (c), deblurred frames (d), and object boundaries ((e), green).

Figure 2 illustrates how a scene is generated as a composition of layers, each of which is *individually* warped and blurred by its motion. This compositing from layers that are independently blurred captures what happens at boundaries while simplifying optimization compared with previous work. We explicitly model the blur as a function of the estimated motion, resulting in an elegant formulation of the problem. Thus, given the estimated motion, the blur within each layer is known, and the latent, sharp, appearance of each layer can be reconstructed. As a result we can reconstruct accurate motion at the boundaries, as well as deblurred estimates for both layers.

To summarize, the main contribution of this work is to estimate optical flow in video sequences in the presence of multiple motions and motion blur. We treat motion blur in a layered framework, allowing us to simultaneously infer the sharp layer segmentation, the object motion, the corresponding motion blur, and the latent (deblurred) object appearances. Our formulation is the first fully generative model of blurred video sequences using a layered framework. We demonstrate the effectiveness of the approach using synthetic and real sequences containing multiple motions. In addition to improving optical flow accuracy, we can deblur sequences that previous methods cannot handle, and show accurate estimation of layer boundaries despite heavy motion blur. We show how it is

robust to noise by modeling a degraded sequence of the assassination of John F. Kennedy.

2 Previous Work

Motion Estimation with Motion Blur. Accurate optical flow estimation in the presence of motion blur has been rarely studied so far. One way to integrate motion blur into the optical flow computation is to include an additional regularization term for the motion, minimizing the difference between a latent, sharp image and the observed, blurry one [27]. Alternatively, the motion blur can be integrated directly into the data term, modulating the brightness constancy constraint [30]. This approach has also been used in sparse tracking [20], where the motion is used to modify the expected appearance change of features. However, these methods usually assume a smooth flow field and, unlike our layered model, fail at object boundaries. Additional previous work has focused on computing motion from a *single blurred image*, either using the image directly [6], or by extracting a matte [12,35]. From this matte, spatially varying motion information can be extracted. While somewhat related to our layered formulation, these approaches usually require user interaction to generate a good matte. In contrast we present a fully automatic approach, taking multiple frames into account.

Motion Deblurring. The treatment of motion blur has focused mostly on removing blur from single images captured in low-light scenarios. The scene is usually assumed to be static, and the blur is caused by camera shake. The blur process is modeled either as a complex but spatially-invariant blur kernel [7,13] or as a spatially-varying kernel generated by possible camera motions in 3D [43]. Single-image deblurring is beyond our scope, as are methods relying on special hardware for multiple exposures [34].

In the case of multiple frames, an alternative strategy first estimates motion from the blurred sequence using sparse feature tracking [17], shapes [3], or optical flow [23,46]. The estimated motion is then used to synthesize the blur kernel and deblur the input images non-blindly. These methods fail in the case of strong blur where accurate tracking becomes infeasible [27]. Our approach solves this by explicitly modeling the blur as a function of the motion being estimated. Like us, Li et al. [25] formulate a joint energy function between latent images and motion parameters and optimize it using gradient descent. However, they assume a single motion and cannot handle independently moving objects.

Methods that take object motion into account usually assume purely translational motion [2,4,24] or require user input to generate a matte [11,26]. In these cases the background is generally assumed to be static, or even known, and only foreground motion is modeled. This eliminates the challenge of estimating overlapping motions at object boundaries, making the problem considerably easier. In contrast, we model motion of both the background and the foreground, and assume neither to be known a priori.

Model-free methods for multi-frame deblurring remove motion blur using deconvolution in a coarsely up-sampled spatio-temporal volume [38] or patch-based

synthesis methods [8]. While such methods give good deblurring results even in the presence of independently moving objects, they yield neither motion information nor scene segmentation.

Layered Motion Models. Layered models of optical flow describe a scene as a set of overlapping moving regions. This effectively separates motion estimation from segmentation, allows the use of simple models for the motion within a layer [41], and provides an explicit model of the occlusion process [36]. Layered motion models also facilitate temporal reasoning because the structure of most scenes changes slowly over time [36]. Thus, layers are useful for motion segmentation [10,21] and the computation of optical flow [19,36,39,42]. None of these approaches deal with motion blur.

One of the early motivations for layered models, however, was to address blur [40]. The idea was to first compute motion using a layered model and then, given the layer segmentation, model the blur process. Paramanand and Rajagopalan [28] follow this approach and use a layered scene representation to capture different blurs. However, they only consider a static scene with camera shake, model the motion as a similarity transform, and do not explicitly model layer interactions at the object boundaries.

Beyond motion estimation, several authors have used layered models to extract persistent image appearance ("sprites," mosaics, etc.) [15,18,21,31,32,45,47]. This estimation of appearance is a key part of our method but these previous approaches do not model motion blur. For example in [32] the super-resolved layer appearance is estimated using a hand-set blur kernel. This blur only represents lens blur and not spatially varying motion blur. Rav-Acha et al. [31] rely on feature tracking and point out that motion blur causes problems for their method. Chunhe et al. [9] use a layered formulation for single image deblurring. Their work considers non-overlapping layers, more akin to a segmentation, and adapt the spatial prior accordingly. Our notion of layers, in contrast, handles overlapping layers and occlusions.

Closest to ours is the work of Kumar et al. [29] in which the authors segment the video into layers, estimate the appearance of layers and model motion blur in the estimation of flow. Our method differs in several important ways. First, they estimate the appearance of a layer as the mean of the aligned image pixels within the layer. This process *does not model how the appearance is blurred by motion* and consequently cannot deblur the appearance. Second, they *model the blur of each layer independently.* This ignores the critical effect of blur at layer boundaries where the appearance of two elements of the scene are combined. This further means that information about motion blur is not properly incorporated into the segmentation of the layers. Third, their method for estimating flow relies on normalized cross correlation while our method is fully generative, modeling the full appearance of each image from the model. Fourth, they do not directly parameterize the blur kernel based on the motion. In contrast, our explicit parameterization of blur facilitates a simpler unified formulation and optimization scheme.

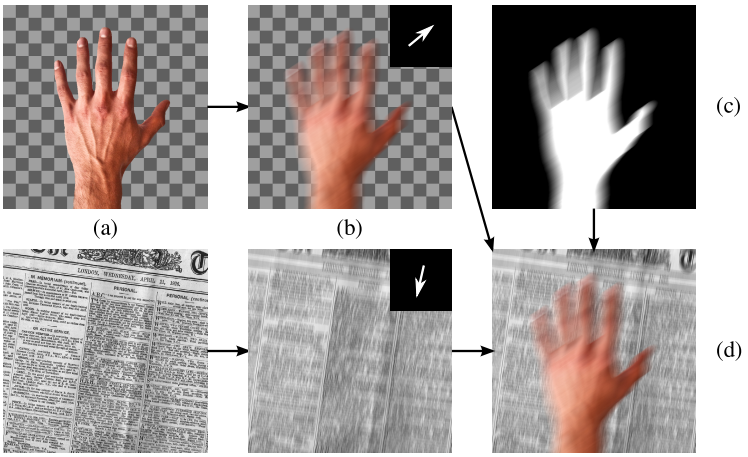


Fig. 2. Generative Model. The sharp layers (a) are blurred (b) using the motion indicated in the top right corners. Together with the blurred layer segmentation (c), an image (d) with complex spatially-varying motion blur is generated. The checkerboard-pattern indicates transparency. Image inspired by [41].

3 A Layered Representation of Motion-Blurred Video

Notation. A superscript denotes a layer l, $0 \leq l \leq L-1$, where larger l's are closer to the observer. Here we use a simplified model with $L = 2$ layers. A subscript denotes the image frame at time t, $1 \leq t \leq T$, for a sequence with T frames. $I_t \in \mathbb{R}^{m \times n \times 3}$ is an observed color image, $A^l \in \mathbb{R}^{m \times n \times 3}$ is the unblurred color "appearance" of layer l. $G^l \in \mathbb{R}^{m \times n \times 3}$ is the segmentation mask for l; for $l = 0$ (i.e. the background) the mask is assumed to be uniformly one. While G^l does not necessarily have to be binary, here we only consider opaque layers. Note that while G^l does have three color channels, we enforce all three to be the same for a given pixel. A^l and G^l are assumed to be constant across the sequence. For longer sequences, this limitation could either be relaxed, or the sequence could be split into smaller sub-sequences, each exhibiting approximately constant layer shape and layer appearance. For readability, we use the column-vectorized forms of I_t, A^l, G^l; i.e. $\mathbf{i}_t \in \mathbb{R}^{3mn \times 1}$, $\mathbf{a}^l \in \mathbb{R}^{3mn \times 1}$, and $\mathbf{g}^l \in \mathbb{R}^{3mn \times 1}$, respectively.

θ_t^l are the transformation (i.e. motion) parameters for layer l at frame t. The complexity of θ_l^t depends on the motion model, and can range from a single (u, v) value pair in case of purely translational motion to displacement values for every pixel in the case of dense optical flow. Here we focus on affine motion, which is the most common for layered flow. It provides a good middle ground by capturing the most common frame-to-frame transformations, still being a linear transformation. While more complicated motion models such as a full homography are possible to include in our model, perspective effects from frame to frame in a video are often negligible, making affine motion a good approximation.

To cope with object areas leaving the visible frame, $m \times n$ is set to be larger than the observation data by padding in all directions. With this simple approach, we are able to reconstruct objects leaving the frame; i.e. panoramas.

A Single Layer with Motion Blur. To fix ideas, first consider the simplest case of motion blur, in which a single layer undergoes an arbitrary transformation during the period of open shutter. A pixel in the blurred image can then be approximated as a linear combination of pixels of the unblurred layer [33]. Thus, we can model the blur as a blur matrix $\mathbf{H}(\theta_t, \theta_{t-1}, s) \in \mathbb{R}^{3mn \times 3mn}$, and write the blurred image as

$$\mathbf{i}_{singlelayer} = \mathbf{H}(\theta_t, \theta_{t-1}, s)\,\mathbf{a}. \tag{1}$$

$\mathbf{H}$ depends on the affine parameters of the current and the previous frame, as well as the shutter speed s. In this work, we assume the shutter speed to be known and constant; this is a reasonable assumption for digital video cameras and even archival film footage as we show in the experiments. $\mathbf{H}$ is set to influence different color channels equally.[1]

Two Layers without Motion Blur. Without motion blur, an image that is composed of two different layers with appearances $\mathbf{a}^0$ and $\mathbf{a}^1$, can be written as

$$\mathbf{i}_{noblur} = \left(\mathbf{1} - \mathbf{g}^1\right) \odot \mathbf{a}^0 + \mathbf{a}^1. \tag{2}$$

Here, $\odot$ denotes element-wise multiplication. Similar to [2], we enforce $\mathbf{a}^1$ to be zero everywhere where its segmentation is zero.

Two Layers with Foreground Motion and Blur. For readability, we abbreviate $\mathbf{H}^l_{t,t-1} = \mathbf{H}\left(\theta^l_t, \theta^l_{t-1}, s\right)$. Assuming a static background, we get

$$\mathbf{i}_{fg-blur} = \left(\mathbf{1} - \mathbf{H}^1_{t,t-1}\mathbf{g}^1\right) \odot \mathbf{a}^0 + \mathbf{H}^1_{t,t-1}\mathbf{a}^1. \tag{3}$$

Now considering multiple points t in time, Eq. (3) becomes

$$\mathbf{i}_{fg-blur,t} = \left(\mathbf{1} - \mathbf{H}^1_{t,t-1}\mathbf{T}^1_t\mathbf{g}^1\right) \odot \mathbf{a}^0 + \mathbf{H}^1_{t,t-1}\mathbf{T}^1_t\mathbf{a}^1. \tag{4}$$

In addition to the blur matrices $\mathbf{H}^l_{t,t-1}$, we use the transformation matrices $\mathbf{T}^l_t \in \mathbb{R}^{3mn \times 3mn}$ to transform a vectorized image according to the transformation parameters θ^l_t.

Since any blur and transformation can be expressed as linear (re-)combination of pixels in the image, this formulation is very flexible, allowing complex motion models such as homographies or dense optical flow [33].

A Two-Layer Model. While algorithms dealing with spatially-varying blur commonly assume a static background [2,4], this assumption is often invalid in practice; eg. in the presence of camera motion in addition to object motion. Incorporating background motion and its corresponding blur into Eq. (4) yields an estimated image

$$\hat{\mathbf{i}}_t = \left(\mathbf{1} - \mathbf{H}^1_{t,t-1}\mathbf{T}^1_t\mathbf{g}^1\right) \odot \mathbf{H}^0_{t,t-1}\mathbf{T}^0_t\mathbf{a}^0 + \mathbf{H}^1_{t,t-1}\mathbf{T}^1_t\mathbf{a}^1. \tag{5}$$

[1] To perform super resolution, A would be larger than I and we would multiply by another matrix to decimate the blurred A in generating I.

Likelihood. The image $\hat{\mathbf{i}}_t$ generated by our model should match the observed image $\mathbf{i}_t$. Hence to estimate the model parameters we minimize

$$E_D\left(\mathcal{I},\mathcal{G},\mathcal{A},\mathbf{\Theta}\right)=\sum_t \mathbf{w}^\top \rho\left(\mathbf{i}_t-\hat{\mathbf{i}}_t\right) \tag{6}$$

summed over all pixels, where $\rho(x)=\sqrt{x^2+\varepsilon^2}$ is a robust Charbonnier function [5]. Here, $\mathbf{w}\in\mathbb{R}^{3mn\times 1}$ contains zeros in the outside padded area, and ones in the inside, restricting the summation to the visible part of the image. The input is the image sequence $\mathcal{I}=\{\mathbf{i}_1,\cdots,\mathbf{i}_T\}$. The estimated parameters are the set of segmentations (excluding the background), $\mathcal{G}=\left\{\mathbf{g}^1\right\}$, the set of appearances, $\mathcal{A}=\left\{\mathbf{a}^0,\mathbf{a}^1\right\}$, and the set of transformation parameters, $\mathbf{\Theta}=\left\{\theta_0^0,\theta_0^1,\cdots,\theta_T^0,\theta_T^1\right\}$. For each layer, we obtain $T+1$ values for θ, since each frame $\mathbf{i}_t$, including the first, depends on θ_t and θ_{t-1}.

Regularization. To make Eq. (6) better behaved in weakly structured regions, we impose a number of regularization terms on both the appearance maps $A^{\{0,1\}}$ and the segmentation G^1.

Spatial Smoothness. We regularize both the appearance images $A^{\{0,1\}}$ and the segmentation maps G. Like standard deblurring methods we model the fact that the spatial derivatives of natural images exhibit a heavy-tail distribution [24]. This can be captured using a sparse prior:

$$E_{sparse}(Y,\alpha)=\sum_{x,y}|\nabla_x Y(x,y)|^\alpha+|\nabla_y Y(x,y)|^\alpha. \tag{7}$$

Consistent with natural image statistics, we use $\alpha_A=0.8$ for the appearance maps.

For a binary segmentation mask G, we use the L1 total variation prior, given by Eq. (7) by setting $\alpha_G=1$. This prior prefers smooth contours, and has been successfully used in similar tasks before [2].

We approximate the non-differentiable absolute $|\cdot|$ in Eq. (7) with a Charbonnier function [5] with $\epsilon=10^{-3}$.

Background Preference. We assume the background to generally cover more pixels than the foreground layer. Thus, we impose a slight penalty for pixels that are assigned to the foreground.

$$E_{bg}(Y)=\sum_{x,y}Y(x,y)^2. \tag{8}$$

Objective. The final objective function is

$$E\left(\mathcal{I},\mathcal{G},\mathcal{A},\mathbf{\Theta}\right)=E_D\left(\mathcal{I},\mathcal{G},\mathcal{A},\mathbf{\Theta}\right)+E_{Reg}\left(\mathcal{G},\mathcal{A}\right), \tag{9}$$

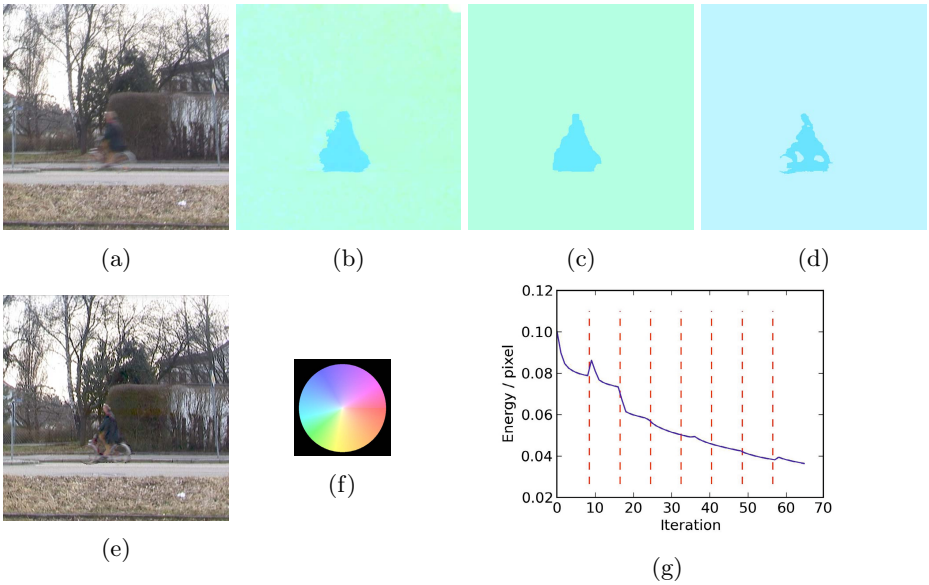


Fig. 3. Illustration of different steps in the algorithm. (a) One of the 5 input frames in the sequence. (b) Initial flow, computed using [44]. (c) Motion initialization from optical flow. (d) Final motion estimate. (e) Deblurred image produced by the generative model. (f) Color key used for optical flow. (g) Normalized energy vs. number of iterations. Red lines show transitions between the pyramid levels. See text for details.

with

$$\begin{aligned} & E_{Reg}\left(\mathcal{G}, \mathcal{A}\right) = \\ & + \lambda_{sparse,A}\left(E_{sparse}\left(A^0, \alpha_A\right) + E_{sparse}\left(A^1, \alpha_A\right)\right) \\ & + \lambda_{sparse,G} E_{sparse}\left(G^1, \alpha_G\right) \\ & + \lambda_{bg} E_{bg}\left(G^1\right) \ . \end{aligned} \tag{10}$$

We set the parameters to $\lambda_{bg} = 0.05$, $\lambda_{sparse,A} = 0.001$, $\lambda_{sparse,G} = 0.05$. For the effects of the individual regularizations, please see the **Sup. Mat.**

4 Optimization

We assume that the shutter speed is known and that there are only two moving layers. Our algorithm computes the latent (unblurred) appearance of both the background and the foreground, the motion parameters of both, and the segmentation mask for the foreground. Figure 3 shows an input image, and how the solution changes after each step described here. Note that it is usually possible to reconstruct the parts of the background that are visible in at least one frame of the sequence.

Initialization. Since the objective function (10) is non-convex, a good initialization of $\mathcal{A}$, $\mathcal{G}$, and Θ is important. While standard optical flow estimates from a blurred sequence are not generally accurate enough to actually perform deblurring [27], they nevertheless provide a useful initial estimate of the motion. Thus, to initialize, we first compute dense optical flow using an off-the-shelf optical flow algorithm, MDP-Flow [44] (Fig. 3(b)). Note that the choice of initial optical flow algorithm is not critical, as long as it produces reasonable results. We tested a number of different optical flow algorithms, but did not observe significant differences in the end result. The reason for this is that the precise spatial configuration of the initial optical flow is less important than the extraction of the dominant motions, which all methods were capable of.

From the initial dense flow field, $L = 2$ dominant motions are robustly estimated,[2] yielding the initial motion parameters for each frame. Additionally, this gives a per-frame pixel assignment estimate (i.e. foreground or background), similar to the approach used in [41] (Fig. 3(c)). Using the estimated motion parameters, the pixel assignments are aligned across all frames and added up. The result can be interpreted as an unnormalized foreground probability. We combine this with a spatial consistency term, and optimize via graph cuts [22], resulting in a single assignment estimate for the whole sequence.

Given this segmentation and the estimated motion for each layer, we separate both layers by masking, and use a simple non-blind deblurring method [14] on each layer to obtain initial deblurred estimates. While this initial deblurring causes ringing artifacts and is not strictly necessary, it speeds up convergence by providing a reasonable initialization.

Coordinate Descent. Starting with our initial estimate, we use an iterative, alternating optimization method. We optimize one variable at a time using gradient descent, but terminate the optimization after 3 iterations to avoid reaching local optima, and switch to the next variable. One optimization cycle over all variables makes up a single iteration. We iterate for at most 50 iterations, or until the relative change in energy falls below 1 percent.

For optimization purposes, we relax the binary-valued $\mathbf{g}^1$ and use an element-wise shifted heavy-side function $u(x) = 0.5 + \frac{1}{\pi} \arctan\left(\frac{x-0.5}{\sigma_u}\right)$. We approximate $\mathbf{g}^1 \approx u\left(\tilde{\mathbf{g}}^1\right)$, and optimize the continuous-valued $\tilde{\mathbf{g}}^1$ instead. Empirically, we set $\sigma_u = 0.05$.

To deal with large motions, we use a standard multi-scale approach with a Gaussian pyramid with P levels. We found $P = 7$ and a scaling factor of 1.5 to work well. The initialization is done at the highest resolution, and the estimated starting values are rescaled to the highest pyramid level. The complete optimization schedule is then performed at each pyramid level, and the results form the input of the next level. See **Sup. Mat.** for pseudocode. Figure 3(g) shows the energy per pixel vs. number of iterations. The red dashed lines show the transition points between pyramid levels. While the gradient descent scheme

[2] For translational motion, we use the median; for affine, RANSAC.

we use is not guaranteed to reach the global optimum, we nevertheless observe a well-behaved falloff.

We have experimented with different optimizers for the single variables, but found this simple gradient descent algorithm to work best. For the detailed derivatives of the objective function, please see **Sup. Mat.**

After the optimization has converged, we obtain the final mask by smoothing and thresholding $\tilde{\mathbf{g}}^1$, and multiply the binary mask element-wise with A^1 to compute the final foreground appearance estimate. Figure 3(d) shows the final estimated flow. Note that the segmentation is significantly improved, with even the shape of the person and rims of the bicycle being evident. A full composite with blur removed is shown in Fig. 3(e).

Using unoptimized Python code, our algorithm takes less than 40min per frame with a resolution of $m = n = 640$ pixels.

5 Evaluation

We evaluate the algorithm in terms of motion accuracy and deblurring performance. Furthermore, we compare the results of our algorithm with and without the blur model. We use both synthetic and real sequences, containing translational and affine motion. The length of the sequences varies from 5 to 10 frames. Two of our 8 test sequences contain static backgrounds (eg. in Fig. 1, or the middle column of Fig. 4), while the remaining 6 contain moving foreground and background. Three of the test sequences contain significant affine motions including scale change and rotation. Here we present an overview of the results; a complete list is given in the **Sup. Mat.**

Motion Estimation Accuracy. We compare our motion estimation accuracy with different methods from the optical flow literature: Sun et al. [36] use a layered optical flow model; Portz et al. [30] incorporate motion blur into an algorithm for optical flow computation, but do not take layers into account; Xu and Jia [44] are representative of a non-layered, but accurate optical flow method. The implementations were either obtained from the author's websites, or provided upon request. In all cases, we used the default parameters. The only exception was [30], for which we were unable to compute reasonable results using the included initialization optical flow method. Therefore, we use the same initialization [44] as for our method.

To quantitatively compare the results, we use two metrics on the synthetic sequences. First, we compare the full dense flow fields of all methods using the average endpoint error ("Error frame" in Table 1). Second, we fit two parametric motions to all flow fields, as described in Sec. 4, and compare those ("Error fitted" in Table 1). Figure 5 shows an example, and Table 1 shows the average errors over all sequences. Flow accuracy with our method improves significantly over the other techniques. As can be seen in Fig. 5, our method is able to extract even very fine details, however, in the absence of texture in the sky, the segmentation of foreground and background is ambiguous.

Fig. 4. Example results. The top row shows a frame of the input sequence, the middle row the estimated motion and segmentation, the bottom row the output of our generative image model (i.e. a deblurred image).

To investigate the effect of our explicit motion blur model, we compare the motion estimation accuracy for our method with and without the motion blur modeling enabled. Disabling the motion blur is equivalent to setting the shutter speed to be infinitely short. Without modeling blur, the accuracy of our method is comparable to the other methods. The results in Table 1 clearly show the critical role that the blur model plays in motion estimation accuracy.

Note that compared to other optical flow methods, our method computes good segmentations, as implicitly shown in the optical flow maps. More explicitly, consider Fig. 1(e). Here, the object boundaries extracted from the standard optical flow (as described in Sec. 4) are shown in red, while the layer boundaries extracted from our method are shown in green. While the standard flow method [44] fails to capture the fine details between the fingers due to the motion blur, our method is capable of recovering those details. A similar effect can be observed in Fig. 3(d), and Fig. 5(e).

(a) Input frames (b) Xu and Jia [44] (c) Portz et al. [30] (d) Sun et al. [36] (e) Ours

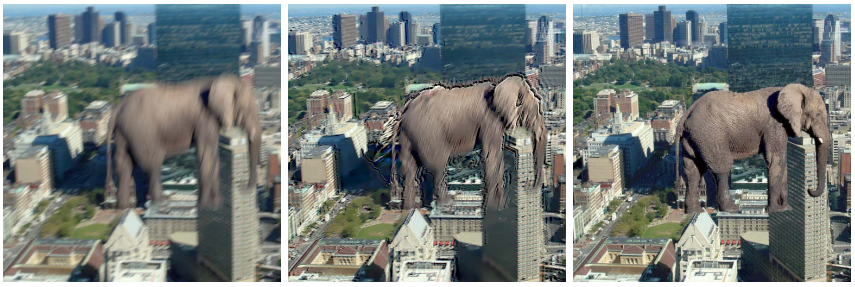
Fig. 5. Conventional optical flow estimation methods fail at object boundaries in the presence of motion blur, since the motion in those areas is a combination of two motion-blurred image regions. (e) shows our result.

Table 1. Optical flow accuracies (average endpoint error)

	Error frame	Error fitted
MDP-Flow [44]	3.58	3.28
Blurflow [30]	4.59	4.92
Layerflow [36]	3.27	4.15
Ours-noblur	3.06	3.42
Ours	**0.73**	**1.25**

Deblurring Accuracy. We compare the results of our approach with those of [7] for deblurring. Figure 6(a) shows an input from a scene in which both layers only undergo translational motion. Within each layer, the blur kernel is therefore spatially invariant. Note that [7] is designed for spatially invariant blur caused by camera shake. To apply it to these images with multiple motions, we use the approach described in Sec. 4 to obtain a rough estimate of background and foreground segmentations. Both segments are then separately deblurred, and the result is composed again. Figure 6 shows a visual comparison of deblurring results. On average, this modified version of [7] achieves a PSNR of 18.34 dB, while our generative model has a PSNR of 29.31 dB. Since [7] was not intended to be used in a spatially-discontinuous setting, this is not an entirely fair comparison. However, by the same token it is not our goal to present a perfect deblurring method. Rather, we show that our generative model of layered motions effectively deblurs the layers resulting in better deblurred images.

Historical Challenge. Figure 7 demonstrates the method for an extremely challenging archival video with large blur, film grain, and extreme noise – the famous Zapruder film of the John F. Kennedy assassination. Using 7 frames, our method removes the motion blur in the scene, both in the rightward driving car and in the background. The non-rigid motions of the people in the scene (e.g. the child's legs) produce some artifacts and require a more flexible motion model. However, even without this our method performs surprisingly well.



(a) One input frame (b) [7], segmented (c) Our result

Fig. 6. Deblurring example (one frame from sequence). Note that there is different foreground and background blur. Conventional deblurring methods suffer from discontinuously varying motion blur at object boundaries.

(a) One original frame and magnification (b) Deblurred frame, magnification, and computed motion and layer segmentation

Fig. 7. Kennedy Assassination. Our method is robust to severely degraded input images. *Zapruder Film* ©*1967 (Renewed 1995) The Sixth Floor Museum At Dealey Plaza.*

6 Assumptions, Limitations, and Future Work

To demonstrate our approach, we assume parametric models of the layer motion, which currently restricts our method to scenes with suitable motion. Our model formulation however is general and just as valid for smoothly varying flow within a layer. Future work will address this more general formulation (e.g. [36,42]). The model also assumes a two-layer scene. While such a model is frequently used in comparable work [28,37], it is again somewhat restrictive. Future work will extend our current model to more layers, and allow more varied motion within a given layer. For a preliminary experiment on the effects of gradually increasing perspective motion, please see the **Sup. Mat.**

Our model assumes constant layer appearance over the sequence. Figure 8 shows a case in which reflections in the windshield of the car cause a change in appearance over time. This leads to ringing artifacts in the segmentation and foreground layer estimation. Future work should allow gradual changes in the appearance, or split the sequences into smaller subsets, for which the appearance constancy assumption approximately holds.

(a) One input frame (b) Motion estimation (c) Deblurring result

Fig. 8. Failure case. A reflection in the windshield causes a changing appearance of the foreground layer, leading to an incorrect segmentation.

The limitations mentioned above should be considered in the light of our claims. We do not claim that we have developed either a full motion estimation or a full deblurring system. Instead we propose a new layered generative model of blurred video and show its feasibility for motion estimation and deblurring. Recent work on layered flow has addressed the estimation of the number of layers, their depth order, and the use of static image cues to improve layer segmentation [36]. Our framework is consistent with these techniques and could be incorporated into them. Future work should include optimizing the code and learning the regularization terms from training data, using more layers, and allowing the layer appearances, $\mathcal{A}$, and segmentation, $\mathcal{G}$, to vary over time. Additionally, future work will include improving the optimization with respect to $\mathcal{A}$ by using a dedicated, state-of-the-art deblurring method, instead of the coordinate descent method employed here.

Here we have considered a classical shutter, but low-cost sensors today, using CMOS technology, have rolling shutters. Modeling such shutters is possible in our framework but remains future work (cf. [16]). Beyond motion blur, images contain further artifacts such as focal blur, discretization, and camera noise. These have been modeled before and could be incorporated into our model, enabling joint motion estimation, denoising, and super-resolution (cf. [1]).

7 Conclusion

We have developed a principled formulation of motion blur in layers, have shown how it can be used to jointly estimate parametric motion, deblurred appearance, and scene segmentation, and have demonstrated the effectiveness of this approach using synthetic and real video sequences with appropriate motion. The results point to the value of incorporating a model of motion blur into the formulation of optical flow. A key insight is that, given the optical flow and the shutter speed, the blur is completely determined. This simplifies the modeling and estimation problem. Additionally, we argue that the scene structure, resulting in

occlusion, has to be modeled in order to capture the complex interplay between different depth layers, especially if motion blur causes different depth layers to smear into each other. We have shown that the layered model captures the blur at boundaries between surfaces and, by modeling the blur process, we achieve better motion estimation results, layer segmentation, and layer deblurring.

Acknowledgements. We thank D. Sun for discussions on optical flow, T. Adelson for insights into motion blur and layers, and R. Zavada for the JFK video.

References

1. Baker, S., Kanade, T.: Super-resolution optical flow. Tech. Rep. CMU-RI-TR-99-36, Carnegie Mellon University, The Robotics Institute (1999)
2. Bar, L., Berkels, B., Rumpf, M., Sapiro, G.: A variational framework for simultaneous motion estimation and restoration of motion-blurred video. In: IEEE Int. Conf. on Computer Vision (ICCV), pp. 1–8 (2007)
3. Bascle, B., Blake, A., Zisserman, A.: Motion deblurring and super-resolution from an image sequence. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065, pp. 573–582. Springer, Heidelberg (1996), http://dl.acm.org/citation.cfm?id=645310.649034
4. Chakrabarti, A., Zickler, T., Freeman, W.: Analyzing spatially-varying blur. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 2512–2519 (June 2010)
5. Charbonnier, P., Blanc-Feraud, L., Aubert, G., Barlaud, M.: Two deterministic half-quadratic regularization algorithms for computed imaging. In: IEEE Int. Conf. Image Proc. (ICIP), vol. 2, pp. 168–172 (1994)
6. Chen, W.G., Nandhakumar, N., Martin, W.: Image motion estimation from motion smear-a new computational model. IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) 18(4), 412–425 (1996)
7. Cho, S., Lee, S.: Fast motion deblurring. In: ACM Trans. Graphics (TOG) – Proc. SIGGRAPH Asia, pp. 145:1–145:8. ACM, New York (2009), http://doi.acm.org/10.1145/1661412.1618491
8. Cho, S., Wang, J., Lee, S.: Video deblurring for hand-held cameras using patch-based synthesis. ACM Trans. Graph. 31(4), 64:1–64:9 (2012), http://doi.acm.org/10.1145/2185520.2185560
9. Chunhe, S., Hai, Z., Wei, J.: Motion deblurring from a single image using multi-layer statistics priors. In: 2011 IEEE International Conference on Consumer Electronics (ICCE), pp. 481–482 (January 2011)
10. Cremers, D., Soatto, S.: Variational space-time motion segmentation. In: Triggs, B., Zisserman, A. (eds.) Int. Conf. on Computer Vision (ICCV), Nice, vol. 2, pp. 886–892 (October 2003)
11. Dai, S., Wu, Y.: Removing partial blur in a single image. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 2544–2551 (2009)
12. Dai, S., Wu, Y.: Motion from blur. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (June 2008)
13. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. ACM Trans. Graphics (TOG) – Proc. SIGGRAPH 25(3), 787–794 (2006), http://doi.acm.org/10.1145/1141911.1141956

14. Fish, D.A., Brinicombe, A.M., Pike, E.R., Walker, J.G.: Blind deconvolution by means of the richardson–lucy algorithm. J. Opt. Soc. Am. A 12(1), 58–65 (1995), http://josaa.osa.org/abstract.cfm?URI=josaa-12-1-58
15. Frey, B., Jojic, N., Kannan, A.: Learning appearance and transparency manifolds of occluded objects in layers. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 45–52 (2003)
16. Grundmann, M., Kwatra, V., Castro, D., Essa, I.: Calibration-free rolling shutter removal. In: IEEE International Conference Computational Photography (ICCP), pp. 1–8 (2012)
17. He, X., Luo, T., Yuk, S.C., Chow, K., Wong, K., Chung, R.H.Y.: Motion estimation method for blurred videos and application of deblurring with spatially varying blur kernels. In: Int. Conf. on Computer Sciences and Convergence Information Technology (ICCIT), pp. 355–359 (2010)
18. Jackson, J., Yezzi, A., Soatto, S.: Dynamic shape and apperance modeling via moving and deforming layers. International Journal of Computer Vision (IJCV) 79(1), 71–84 (2008)
19. Jepson, A.D., Fleet, D.J., Black, M.J.: A layered motion representation with occlusion and compact spatial support. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part I. LNCS, vol. 2350, pp. 692–706. Springer, Heidelberg (2002), http://dx.doi.org/10.1007/3-540-47969-4_46
20. Jin, H., Favaro, P., Cipolla, R.: Visual tracking in the presence of motion blur. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 2, pp. 18–25 (2005)
21. Jojic, N., Frey, B.: Learning flexible sprites in video layers. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. I-199–I-206 (2001)
22. Kolmogorov, V., Zabin, R.: What energy functions can be minimized via graph cuts? IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI) 26(2), 147–159 (2004)
23. Lee, S.H., Moon, N.S., Lee, C.W.: Recovery of blurred video signals using iterative image restoration combined with motion estimation. In: Int. Conf. on Image Processing (ICIP), vol. 1, pp. 755–758 (1997)
24. Levin, A.: Blind motion deblurring using image statistics. In: Schölkopf, B., Platt, J., Hoffman, T. (eds.) Advances in Neural Information Processing Systems, vol. 19, pp. 841–848. MIT Press, Cambridge (2007)
25. Li, Y., Kang, S.B., Joshi, N., Seitz, S., Huttenlocher, D.: Generating sharp panoramas from motion-blurred videos. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 2424–2431 (2010)
26. Lin, H.T., Tai, Y.W., Brown, M.: Motion regularization for matting motion blurred objects. IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI) 33(11), 2329–2336 (2011)
27. Nir, T., Kimel, R., Bruckstein, A.: Variational approach for joint optic-flow computation and video restoration. Tech. Rep. CIS-2005-03, Technion Israel Institute of Technology (2005)
28. Paramanand, C., Rajagopalan, A.: Non-uniform motion deblurring for bilayer scenes. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 1115–1122 (2013)
29. Pawan Kumar, M., Torr, P., Zisserman, A.: Learning layered motion segmentations of video. International Journal of Computer Vision (IJCV) 76(3), 301–319 (2008), http://dx.doi.org/10.1007/s11263-007-0064-x

30. Portz, T., Zhang, L., Jiang, H.: Optical flow in the presence of spatially-varying motion blur. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 1752–1759 (2012)
31. Rav-Acha, A., Kohli, P., Rother, C., Fitzgibbon, A.: Unwrap mosaics: A new representation for video editing. ACM Transactions on Graphics (TOG) - Proc. SIGGRAPH 27(3), 17:1–17:11 (2008)
32. Schoenemann, T., Cremers, D.: A coding cost framework for super-resolution motion layer decomposition. IEEE Trans. Im. Proc. 23(3), 1097–1110 (2012)
33. Seitz, S., Baker, S.: Filter flow. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 143–150 (2009)
34. Sellent, A., Eisemann, M., Goldlucke, B., Cremers, D., Magnor, M.: Motion field estimation from alternate exposure images. IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI) 33(8), 1577–1589 (2011)
35. Shan, Q., Xiong, W., Jia, J.: Rotational motion deblurring of a rigid object from a single image. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2007)
36. Sun, D., Sudderth, E., Black, M.J.: Layered segmentation and optical flow estimation over time. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 1768–1775 (2012)
37. Sun, D., Wulff, J., Sudderth, E., Pfister, H., Black, M.: A fully-connected layered model of foreground and background flow. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Portland, OR, pp. 2451–2458 (June 2013)
38. Takeda, H., Milanfar, P.: Removing motion blur with space-time processing. IEEE Transactions on Image Processing 20(10), 2990–3000 (2011)
39. Torr, P.H.S., Szeliski, R., Anandan, P.: An integrated bayesian approach to layer extraction from image sequences. In: IEEE Int. Conf. on Computer Vision (ICCV), vol. 2, pp. 983–990 (1999)
40. Wang, J.Y.A., Adelson, E.H.: System for encoding image data into multiple layers representing regions of coherent motion and associated motion parameters. US Pat. 5557684 (1996)
41. Wang, J., Adelson, E.: Representing moving images with layers. IEEE Trans. Image Processing 3(5), 625–638 (1994)
42. Weiss, Y.: Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, pp. 520–526 (1997), http://dl.acm.org/citation.cfm?id=794189.794437
43. Whyte, O., Sivic, J., Zisserman, A., Ponce, J.: Non-uniform deblurring for shaken images. International Journal of Computer Vision (IJCV) 98(2), 168–186 (2012)
44. Xu, L., Jia, J., Matsushita, Y.: Motion detail preserving optical flow estimation. IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI) 34(9), 1744–1757 (2012)
45. Yalcin, H., Black, M.J., Fablet, R.: The dense estimation of motion and appearance in layers. In: IEEE Workshop on Image and Video Registration (June 2004)
46. Yamaguchi, T., Fukuda, H., Furukawa, R., Kawasaki, H., Sturm, P.: Video deblurring and super-resolution technique for multiple moving objects. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part IV. LNCS, vol. 6495, pp. 127–140. Springer, Heidelberg (2011), http://dl.acm.org/citation.cfm?id=1966111.1966123
47. Zhou, Y., Tao, H.: A background layer model for object tracking through occlusion. In: IEEE Int. Conf. on Computer Vision (CVPR), vol. 2, pp. 1079–1085 (2003)