Modeling Blurred Video with Layers
Supplemental material

Jonas Wulff, Michael J. Black
Max Planck Institute for Intelligent Systems, Tübingen, Germany
{jonas.wulff,black}@tue.mpg.de

July 6, 2014

Contents

1 Examples for frame extensions

As stated in the paper, the algorithm presented here is able to reconstruct portions of the background that become uncovered at some point in the video sequence. Below, we show examples for the reconstructed foreground and background, together with the unblurred image, for three example sequences. For the full results, please see the video.

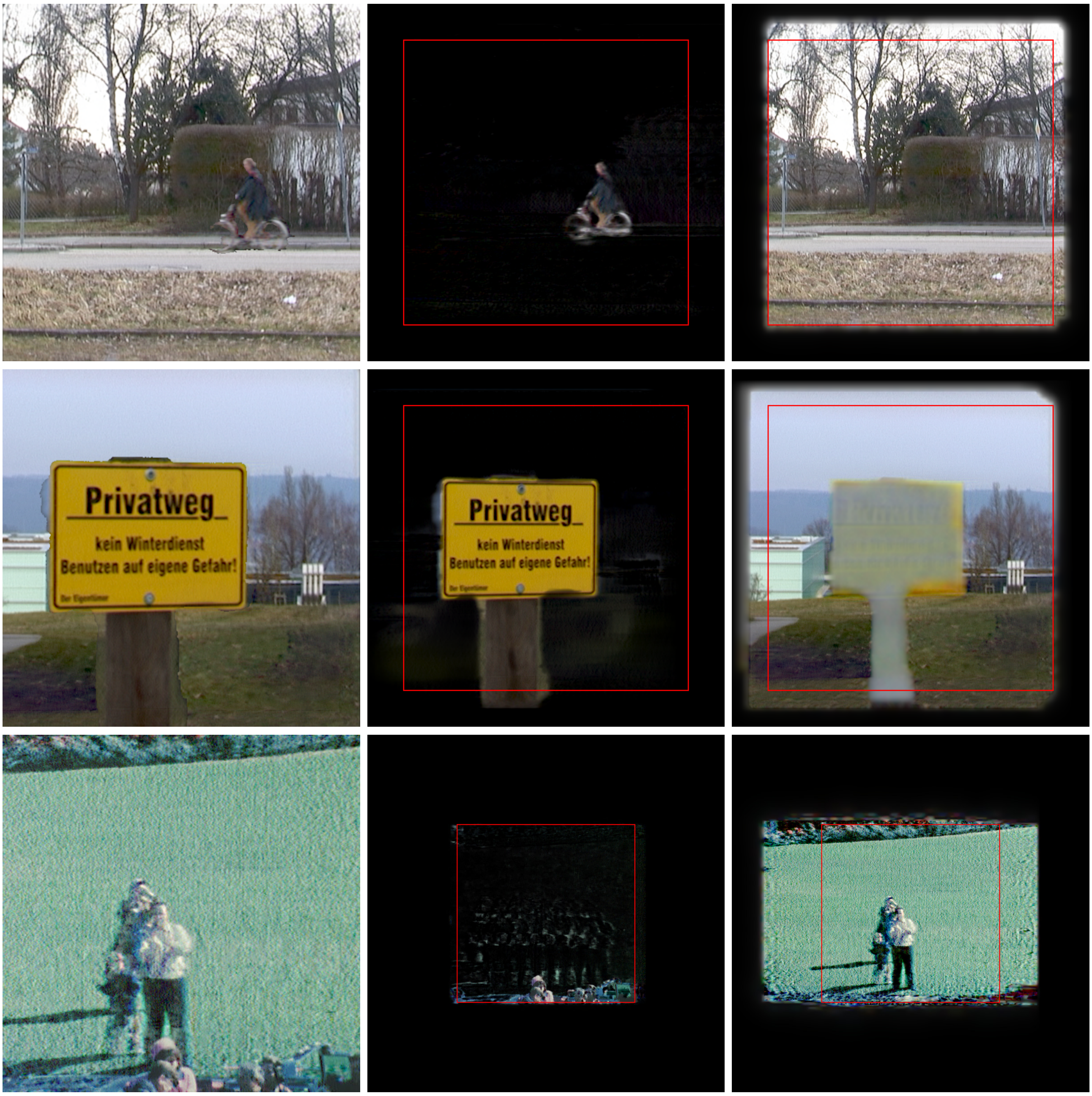

Figure 1: From top to bottom: Bike, Sign, Kennedy sequence. From left to right: deblurred frame, reconstructed foreground layer, reconstructed background layer. The red frame indicates the extent of the current image.

2 Derivatives and algorithm

2.1 Derivatives of regularization terms

2.1.1 Spatial smoothness prior

In the paper, the spatial smoothness prior was defined as (Eq. (9) in the paper)

$$E_{sparse}(Y, \alpha) = \sum_{x,y} \rho_c \left(\nabla_x Y(x,y)\right)^{\alpha} + \rho_c \left(\nabla_y Y(x,y)\right)^{\alpha}, \tag{1}$$

approximating the absolute $|\cdot|$ with the Charbonnier function $\rho_c(x) = \sqrt{x^2 + \epsilon^2}$ with $\epsilon = 10^{-3}$, and with

$$\psi_c(x) = \frac{\partial \rho_c(x)}{\partial d} = \frac{x}{\sqrt{x^2 + \epsilon^2}}. \tag{2}$$

The derivative is then given as

$$\begin{aligned} \left.\frac{\partial E_{sparse}(Y, \alpha)}{\partial Y}\right|_{x,y} =& \nabla_x^- \left[\alpha \rho_c \left(\nabla_x Y(x,y)\right)^{\alpha - 1} \psi_c \left(\nabla_x Y(x,y)\right)\right] \\ &+ \nabla_y^- \left[\alpha \rho_c \left(\nabla_y Y(x,y)\right)^{\alpha - 1} \psi_c \left(\nabla_y Y(x,y)\right)\right], \end{aligned} \tag{3}$$

Here, the gradient ∇ is approximated by a finite difference filter $[0, -1, 1]$, and the inverted gradient ∇^- is approximated by $[1, -1, 0]$.

As described in the paper, we use $\alpha_A = 0.8$ for the layer appearance, consistent with previous work on natural image statistics. For the segmentation mask, we use $\alpha_G = 1.0$.

2.1.2 Background preference

In the paper, the background preference prior is given as

$$E_{bg}(Y) = \sum_{x,y} Y(x,y)^2. \tag{4}$$

The derivative is given as

$$\left.\frac{\partial E_{bg}(Y)}{\partial Y}\right|_{x,y} = 2Y(x,y). \tag{5}$$

2.2 Derivatives with respect to A^1 and A^1

In this section, we return to the vector notation, expressing the input image I_t, the estimated image $\hat{I}_t$, and the layer data A^0, A^1, and G^1 as column vectors $\mathbf{i}_t$, $\hat{\mathbf{i}}_t$, $\mathbf{a}^0$, $\mathbf{a}^1$, and $\mathbf{g}^1$, respectively. Additionally, we define $\mathbf{\Psi}_t = \psi\left(\mathbf{i}_t - \hat{\mathbf{i}}_t\right)$ as a row vector, containing the element-wise derivatives of the point-wise error measure $\rho(\mathbf{x})$ with respect to the individual elements:

$$\psi(\mathbf{x}) = \mathbf{w}^\top \frac{\partial \rho(\mathbf{x})}{\partial \mathbf{x}}, \tag{6}$$

with $\mathbf{w}$ being the mask to extend the image (cf. Eq. (6) in the paper).

The derivatives of the objective function ((10) in the paper) with respect to the appearances are given as:

$$\begin{aligned}\frac{\partial E}{\partial \mathbf{a}^0} &= \frac{\partial E_D}{\partial \mathbf{a}^0} + \frac{\partial E_{Reg}}{\partial \mathbf{a}^0} \\ &= -\sum_t \left[\mathbf{\Psi}_t \odot \left(\mathbf{1}^\top - \left(\mathbf{H}^1_{t,t-1}\mathbf{T}^1_t\mathbf{g}^1\right)^\top\right)\right]\mathbf{H}^0_{t,t-1}\mathbf{T}^0_{t,t-1} \\ &\quad + \lambda_{sparse,A}\frac{\partial E_{sparse}\left(\mathbf{a}^0, \alpha_A\right)}{\partial \mathbf{a}^0} \end{aligned} \tag{7}$$

$$\begin{aligned}\frac{\partial E}{\partial \mathbf{a}^1} &= \frac{\partial E_D}{\partial \mathbf{a}^1} + \frac{\partial E_{Reg}}{\partial \mathbf{a}^1} \\ &= -\sum_t \mathbf{\Psi}_t\mathbf{H}^1_{t,t-1}\mathbf{T}^1_t + \lambda_{sparse,A}\frac{\partial E_{sparse}\left(\mathbf{a}^0, \alpha_A\right)}{\partial \mathbf{a}^0} \end{aligned} \tag{8}$$

2.3 Derivative with respect to G^1

In order to optimize the discrete-valued segmentation mask $\mathbf{g}^1$, we approximate it as $\mathbf{g}^1 \approx \mathbf{u}\left(\tilde{\mathbf{g}}^1\right)$, with the element-wise heavy-side function $\mathbf{u}(\tilde{\mathbf{g}}^1) = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{\tilde{\mathbf{g}}^1 - 0.5}{\sigma_u}\right)$. The optimization is then carried out with respect to $\tilde{\mathbf{g}}^1$. Using the same notation as above, the derivative of the objective function with respect to $\tilde{\mathbf{g}}^1$ is given as

$$\begin{aligned}\frac{\partial E}{\partial \tilde{\mathbf{g}}^1} &= \left(\frac{\partial E_D}{\partial \mathbf{g}^1} + \frac{\partial E_{Reg}}{\partial \mathbf{g}^1}\right)\frac{\partial \mathbf{u}}{\partial \tilde{\mathbf{g}}^1} \\ &= \left[\sum_t \left[\mathbf{\Psi}_t \odot \left(\left(\mathbf{H}^0_{t,t-1}\mathbf{T}^0_t\mathbf{a}^0\right)^\top\right)\right]\mathbf{H}^1_{t,t-1}\mathbf{T}^1_{t,t-1}\right. \\ &\quad \left. +\lambda_{sparse,G}\frac{\partial E_{sparse}\left(\mathbf{g}^1, \alpha_G\right)}{\partial \mathbf{g}^1} + \lambda_{bg}\frac{\partial E_{bg}\left(\mathbf{g}^1\right)}{\partial \mathbf{g}^1}\right]\frac{\partial \mathbf{u}}{\partial \tilde{\mathbf{g}}^1}. \end{aligned} \tag{9}$$

Note that, since $\mathbf{u}(\tilde{\mathbf{g}}^1)$ operates element-wise, only the diagonal entries of $\frac{\partial \mathbf{u}}{\partial \tilde{\mathbf{g}}^1} \in \mathbb{R}^{mn \times mn}$ are non-zero, and contain the element-wise derivatives of $\mathbf{u}(\tilde{\mathbf{g}}^1)$.

2.4 Derivatives with respect to Θ

With respect to Θ, we observe that $\hat{I}_t$ is only a function of θ_t and θ_{t-1}. Usually, $\theta_t \in \mathbb{R}^C$, with $C = 2$ for the translational case. For clarity, we here give the derivatives with respect to a single components

$\theta_{t,(c)}$ of $\theta_t = \left[\theta_{t,(1)} \cdots \theta_{t,(C)}\right]$.

$$\begin{aligned} \frac{\partial E}{\partial \theta^0_{t,(c)}} \;=\;& \delta\left(t>0\right)\left\{ -\boldsymbol{\Psi}_t \left[\left(\mathbf{1} - \mathbf{H}^1_{t,t-1}\mathbf{T}^1_t\mathbf{g}^1\right) \odot \left(\frac{\partial \mathbf{H}^0_{t,t-1}}{\partial \theta^0_{t,(c)}} \mathbf{T}^0_t \mathbf{a}^0 + \mathbf{H}^0_{t,t-1} \frac{\partial \mathbf{T}^0_t}{\partial \theta^0_{t,(c)}} \mathbf{a}^0 \right) \right] \right\} \\ &+ \delta\left(t<T\right)\left\{ -\boldsymbol{\Psi}_{t+1} \left[\left(\mathbf{1} - \mathbf{H}^1_{t+1,t}\mathbf{T}^1_{t+1}\mathbf{g}^1\right) \odot \left(\frac{\partial \mathbf{H}^0_{t+1,t}}{\partial \theta^0_{t,(c)}} \mathbf{T}^0_{t+1} \mathbf{a}^0 \right) \right] \right\} \end{aligned} \tag{10}$$

$$\begin{aligned} \frac{\partial E}{\partial \theta^1_{t,(c)}} \;=\;& \delta\left(t>0\right)\left\{ -\boldsymbol{\Psi}_t \left[\mathbf{H}^0_{t,t-1}\mathbf{T}^0_t\mathbf{a}^0 \odot \left(\frac{\partial \mathbf{H}^1_{t,t-1}}{\partial \theta^1_{t,(c)}} \mathbf{T}^1_t \mathbf{g}^1 + \mathbf{H}^1_{t,t-1} \frac{\partial \mathbf{T}^1_t}{\partial \theta^1_{t,(c)}} \mathbf{g}^1 \right) \right. \right. \\ &\qquad \left. \left. + \frac{\partial \mathbf{H}^1_{t,t-1}}{\partial \theta^1_{t,(c)}} \mathbf{T}^1_t \mathbf{a}^1 + \mathbf{H}^1_{t,t-1} \frac{\partial \mathbf{T}^1_t}{\partial \theta^1_{t,(c)}} \mathbf{a}^1 \right] \right\} \\ &+ \delta\left(t<T\right)\left\{ -\boldsymbol{\Psi}_{t+1} \left(\mathbf{H}^0_{t+1,t}\mathbf{T}^0_{t+1}\mathbf{a}^0 \odot \frac{\partial \mathbf{H}^1_{t+1,t}}{\partial \theta^1_{t,(c)}} \mathbf{T}^1_{t+1} \mathbf{g}^1 + \frac{\partial \mathbf{H}^1_{t+1,t}}{\partial \theta^1_{t,(c)}} \mathbf{T}^1_{t+1} \mathbf{a}^1 \right) \right\} \end{aligned} \tag{11}$$

with $\delta(x) = 1$ if the argument x is true, and $\delta(x) = 0$ otherwise.

Note that it is possible to explicitly construct the $\mathbf{H}$ and $\mathbf{T}$ matrices, since they are usually sparse. However, we found that in practice a finite difference approach achieves comparable performance, and has clear speed benefits. Therefore, we use finite differences in the optimization of θ, and separately optimize over the rotational and translational parameters of the affine transformation.

3 Optimization algorithm

This section describes the alternating, gradient-descent based optimization algorithm in pseudocode.

In the following, `truncatedOptimize[`V`](`$\mathcal{P}$`)` denotes a truncated, *ie.* prematurely terminated optimization of variable V, given the parameters $\mathcal{P}$. Here, we use a basic gradient descent optimization with line search, and terminate after 3 steps. `Energy` denotes the objective function.

As in the paper, unbracketed superscripts indicate the layer. Additionally, bracketed superscripts indicate the iteration number, bracketed subscripts the pyramid level. $\mathcal{A}$, $\mathcal{G}$, Θ indicate the set of appearances, segmentation maps, and parameters, over all layers, respectively, and Θ^l denotes the set of parameters of layer l over time. $\tilde{\mathcal{G}}$ denotes the set of relaxed, continuous layer segmentation masks, with $G = u(\tilde{G})$, as described in the paper.

Algorithm 1 Optimization

```
Require: $\mathcal{I}, \mathcal{A}^{(init)}, \tilde{\mathcal{G}}^{(init)}, \Theta^{(init)}$
  $\mathcal{A}^{(0)}_{(0)} \leftarrow$ ScaleToPyramidLevel $\left(\mathcal{A}^{(init)}, 0\right)$
  $\mathcal{G}^{(0)}_{(0)} \leftarrow$ ScaleToPyramidLevel $\left(\mathcal{G}^{(init)}, 0\right)$
  $\tilde{\mathcal{G}}^{(0)}_{(0)} \leftarrow \mathcal{G}^{(0)}_{(0)}$
  $\Theta^{(0)}_{(0)} \leftarrow$ ScaleToPyramidLevel $\left(\Theta^{(init)}, 0\right)$
  $p \leftarrow 0$
  while $p < P$ do
    $\mathcal{I}_{(p)} \leftarrow$ ScaleToPyramidLevel $(\mathcal{I}, p)$
    $E^{(0)}_{(p)} \leftarrow$ Energy $\left(A^{0,(0)}_{(p)}, A^{1,(0)}_{(p)}, G^{1,(0)}_{(p)}, \Theta^{0,(0)}_{(p)}, \Theta^{1,(0)}_{(p)}\right)$
    $i \leftarrow 0$
    repeat
      $A^{0,(i+1)}_{(p)} \leftarrow$ truncatedOptimize $\left[A^0_p\right] \left(A^{0,(i)}_{(p)}, A^{1,(i)}_{(p)}, G^{1,(i)}_{(p)}, \Theta^{0,(i)}_{(p)}, \Theta^{1,(i)}_{(p)}\right)$
      $\Theta^{0,(i+1)}_{(p)} \leftarrow$ truncatedOptimize $\left[\Theta^0_p\right] \left(A^{0,(i+1)}_{(p)}, A^{1,(i)}_{(p)}, G^{1,(i)}_{(p)}, \Theta^{0,(i)}_{(p)}, \Theta^{1,(i)}_{(p)}\right)$
      $\tilde{G}^{1,(i+1)}_{(p)} \leftarrow$ truncatedOptimize $\left[\tilde{G}^1_p\right] \left(A^{0,(i+1)}_{(p)}, A^{1,(i)}_{(p)}, \tilde{G}^{1,(i)}_{(p)}, \Theta^{0,(i+1)}_{(p)}, \Theta^{1,(i)}_{(p)}\right)$
      $G^{1,(i+1)}_{(p)} \leftarrow u\left(\tilde{G}^{1,(i+1)}_{(p)}\right)$
      $A^{1,(i+1)}_{(p)} \leftarrow$ truncatedOptimize $\left[A^1_p\right] \left(A^{0,(i+1)}_{(p)}, A^{1,(i)}_{(p)}, G^{1,(i+1)}_{(p)}, \Theta^{0,(i+1)}_{(p)}, \Theta^{1,(i)}_{(p)}\right)$
      $\Theta^{1,(i+1)}_{(p)} \leftarrow$ truncatedOptimize $\left[\Theta^1_p\right] \left(A^{0,(i+1)}_{(p)}, A^{1,(i+1)}_{(p)}, G^{1,(i+1)}_{(p)}, \Theta^{0,(i+1)}_{(p)}, \Theta^{1,(i)}_{(p)}\right)$
      $E^{(i+1)}_{(p)} \leftarrow$ Energy $\left(A^{0,(i+1)}_{(p)}, A^{1,(i+1)}_{(p)}, G^{1,(i+1)}_{(p)}, \Theta^{0,(i+1)}_{(p)}, \Theta^{1,(i+1)}_{(p)}\right)$
      $i \leftarrow i + 1$
    until $i = \text{maxiter}$ or $\frac{E^{(i+1)}_{(p)} - E^{(i)}_{(p)}}{E^{(i)}_{(p)}} < \delta$
    $\mathcal{A}^{(0)}_{(p+1)} \leftarrow$ ScaleToPyramidLevel $\left(\mathcal{A}^{(i)}_{(p)}, p + 1\right)$
    $\tilde{\mathcal{G}}^{(0)}_{(p+1)} \leftarrow$ ScaleToPyramidLevel $\left(\tilde{\mathcal{G}}^{(i)}_{(p)}, p + 1\right)$
    $\mathcal{G}^{(0)}_{(p+1)} \leftarrow u\left(\tilde{\mathcal{G}}^{(0)}_{(p+1)}\right)$
    $\Theta^{(0)}_{(p+1)} \leftarrow$ ScaleToPyramidLevel $\left(\Theta^{(i)}_{(p)}, p + 1\right)$
    $p \leftarrow p + 1$
  end while
  $\mathcal{A}^{(opt)} \leftarrow \mathcal{A}^{(0)}_{(P)}$
  $\mathcal{G}^{(opt)} \leftarrow \mathcal{G}^{(0)}_{(P)}$
  $\Theta^{(opt)} \leftarrow \Theta^{(0)}_{(P)}$
  return $\mathcal{A}^{(opt)}, \mathcal{G}^{(opt)}, \Theta^{(opt)}$
```

4 Parameter sensitivity analysis

For all sequences, we set the default parameter values to $\lambda_{bg} = 0.05$, $\lambda_{sparse,G} = 0.05$, and $\lambda_{sparse,A} = 0.001$. To analyze the sensitivity to variations of a parameter, we run our algorithm with different values within a range around the default parameter values. All other parameters are kept constant.

Here we show a synthetic and a real sequence, illustrating the robustness of the results to changes of the individual weights. For λ_{bg} and $\lambda_{sparse,G}$ we show the segmented motion field, since changing these parameters primarily affects the segmentation. For $\lambda_{sparse,A}$ we show the deblurred images.

In the following, the parameter always varies within a row, while the sequence varies within a column. Here we show results for $\lambda_{bg} \in \{0.01, 0.05, 0.1\}$, $\lambda_{sparse,G} \in \{0.01, 0.05, 0.1\}$, and $\lambda_{sparse,A} \in \{0, 0.001, 0.1\}$.

As shown, the results are robust to changes of the parameters within reasonable ranges. Small differences can be seen in the very fine details, such as the front wheel of the bicycle. However, they have very little effect on the results overall.

4.1 Background preference weight λ_{bg}

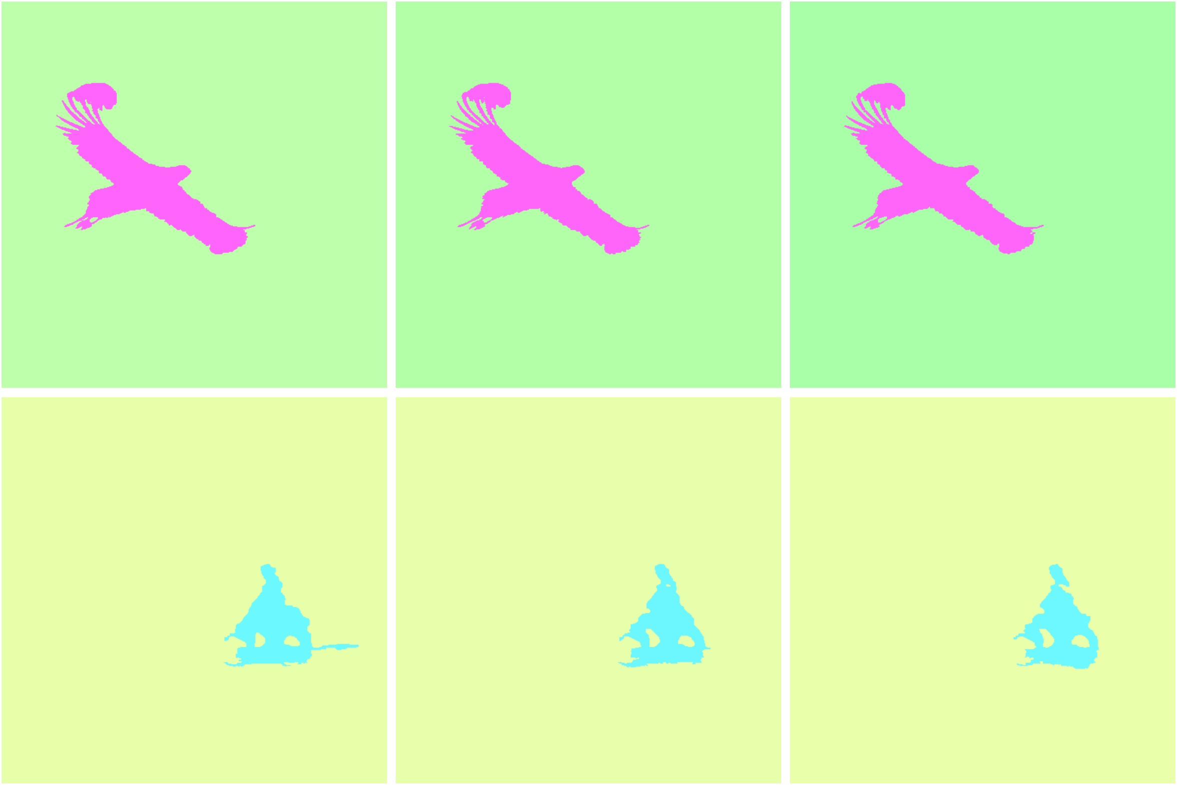

Figure 2: Left column: $\lambda_{bg} = 0.01$. Middle column: $\lambda_{bg} = 0.05$. Right column: $\lambda_{bg} = 0.1$. We find the results to be robust to small changes of λ_{bg}. Differences are visible in small details, primarily around the bicycle (bottom row). For all sequences, we chose $\lambda_{bg} = 0.05$.

4.2 Segmentation sparseness prior weight $\lambda_{sparse,G}$

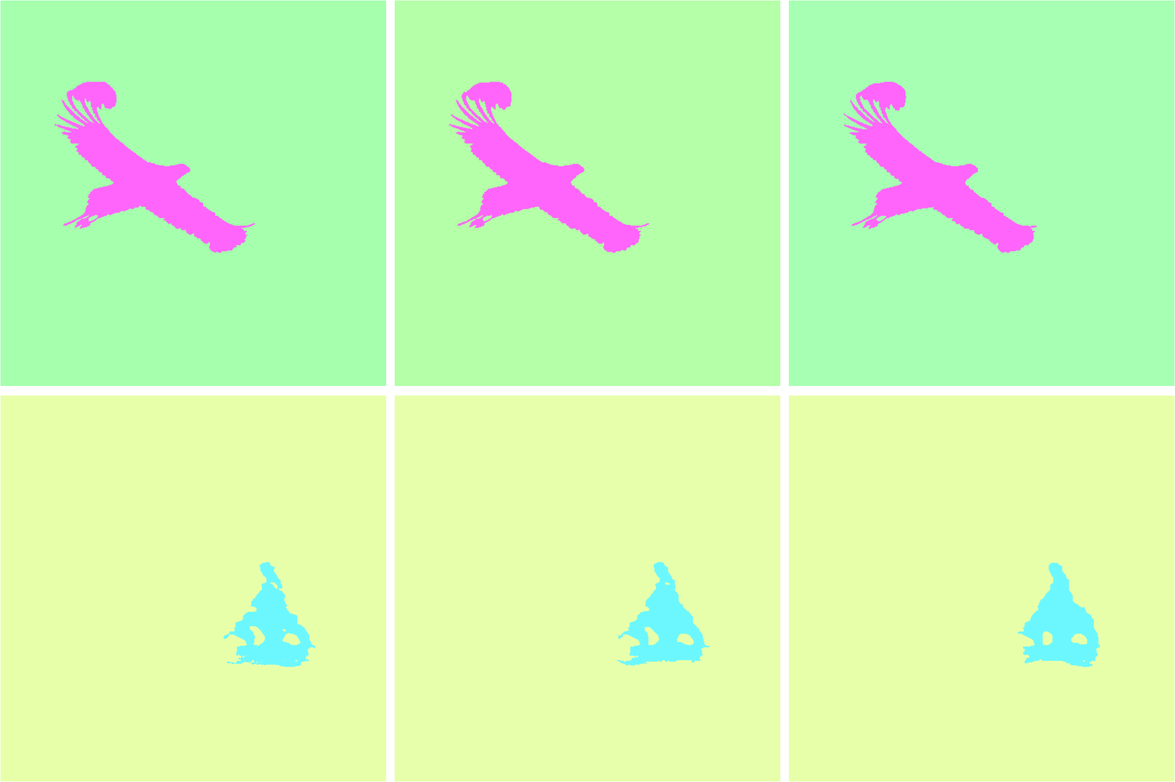

Figure 3: Left column: $\lambda_{sparse,G} = 0.01$. Middle column: $\lambda_{sparse,G} = 0.05$. Right column: $\lambda_{sparse,G} = 0.1$. The results are stable with respect to the choice of $\lambda_{sparse,G}$, as long as it does not become too large (right column, bottom). For all sequences, we chose $\lambda_{sparse,G} = 0.05$.

4.3 Appearance sparseness prior weight $\lambda_{sparse,A}$

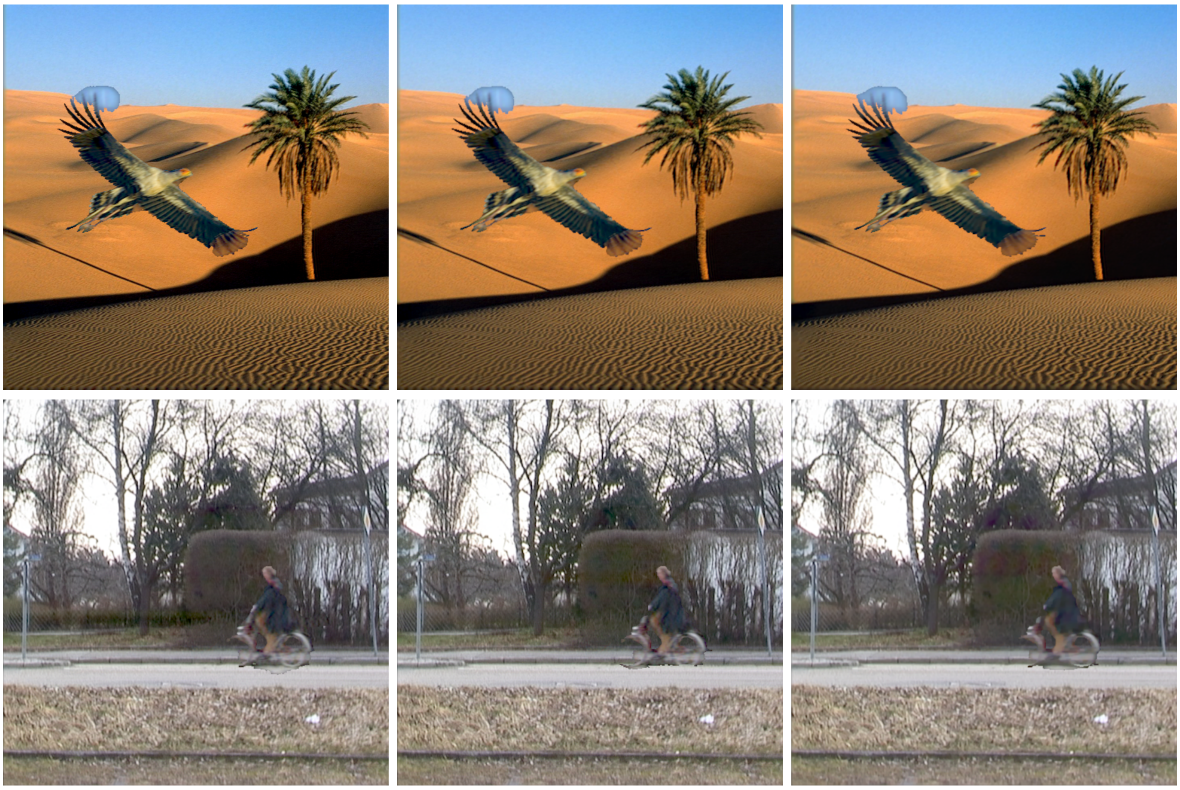

Figure 4: Left column: $\lambda_{sparse,A} = 0$. Middle column: $\lambda_{sparse,A} = 0.001$. Right column: $\lambda_{sparse,A} = 0.1$. Again, we observe a high robustness with respect to the choice of $\lambda_{sparse,A}$. The differences are only visible in small details, like the front wheel of the bicycle. For all sequences, we chose $\lambda_{sparse,A} = 0.001$.

5 Effects of perspective transformations

To test the effects perspective transformations, we used a simple affine sequence with a rotating background and a zooming foreground, and added an out-of-plane rotation of gradually increasing magnitude, from 0 to 30 degrees over a sequence length of 8 frames. The focal length is kept short in order to create perspective foreshortening.

The effect are as expected: With increasing out-of-plane rotation, the flow error generally increases, while the PSNR decreases. However, note that the overall impact is fairly low. When going from 0 to 30 degrees out-of-plane rotation, the average endpoint error increases from 1.43 pixel to 1.84 pixel; the PSNR decreases from 23.7 to 22.6.

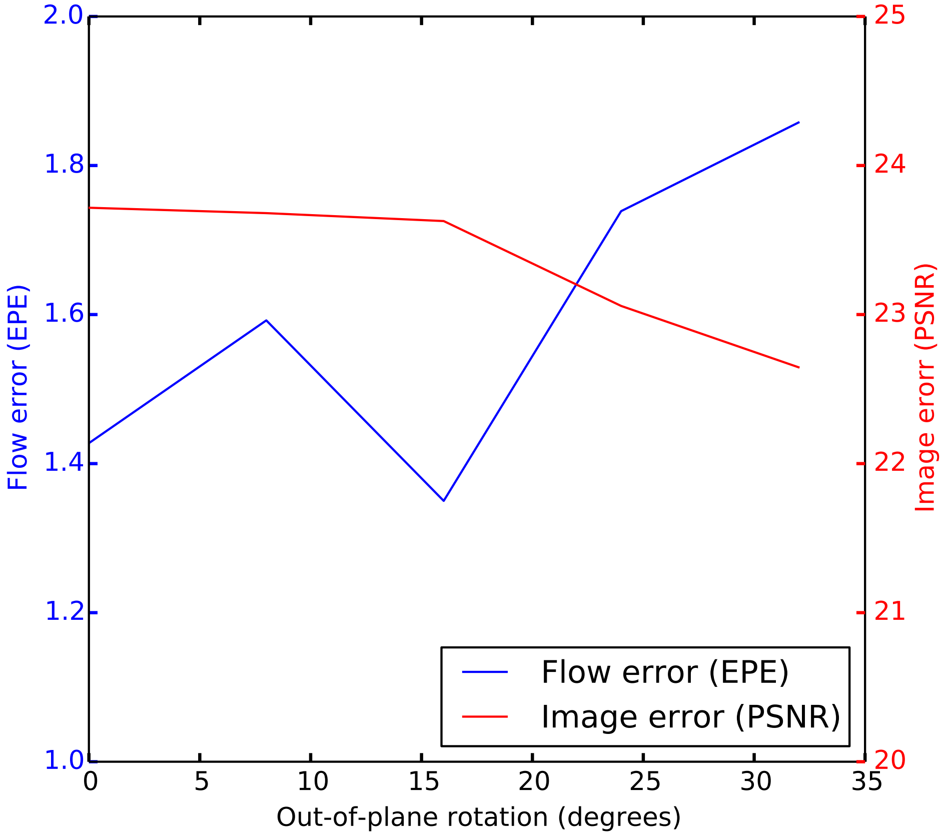

Figure 5: Effects of increasing out-of-plane rotation. While the quality is impacted, the overall degradation is farily low.