Vision, Modeling, and Visualization (2010)

Coherent Background Video Inpainting through Kalman Smoothing along Trajectories

A. Bugeau[1], P. Gargallo[1], O. D'Hondt[1], A. Hervieu[1], N. Papadakis[1] and V. Caselles[2]

[1]Image processing Group, Barcelona Media & Barcelona, Spain
[2]Dept. de Tecnologies de la Informació i les Comunicacions, Universitat Pompeu Fabra & Barcelona, Spain

Abstract

Video inpainting consists in recovering the missing or corrupted parts of an image sequence so that the reconstructed sequence looks natural. For each frame, the reconstruction has to be spatially coherent with the rest of the image and temporally with respect to the reconstructions of adjacent frames. Most of existing methods only focus on inpainting foreground objects moving with a periodic motion and consider that the background is almost static. In this paper we address the problem of background inpainting and propose a method that handles dynamic background (illumination changes, moving camera, dynamic textures...). The algorithm starts by applying an image inpainting technique to each frame of the sequence and then temporally smoothes these reconstructions through Kalman smoothing along the estimated trajectories of the unknown points. The computation of the trajectories relies on the estimation of forward and backward dense optical flow fields. Several experiments and comparisons demonstrate the performance of the proposed approach.

Categories and Subject Descriptors (according to ACM CCS): I.4.3 [Image Processing and Computer Vision]: Enhancement —Smoothing I.4.4 [Image Processing and Computer Vision]: Restoration—Kalman filtering

1. Introduction

Image inpainting consists in recovering the missing or corrupted parts of an image so that the reconstructed image looks natural. In the same way, video inpainting aims at completing the corrupted areas of a video. For each frame, the reconstruction has to be spatially coherent with the rest of the image and temporally with respect to the reconstructions of adjacent frames. There are many possible applications to the inpainting problem: movie post-production, product replacement, video stabilization, image restoration...

For still images, an extensive panel of approaches has been proposed. These methods are based on texture synthesis [EL99, CPT04], geometric diffusion [MM98, BSCB00, Tsc06, BM07], or on a combination of these two approaches [DSC03, ACS09, ALM08, BBCSar]. The extension of these methods to video is at its early stage of development but different strategies have already been proposed to ensure some temporal consistency between the successive reconstructions. We briefly review here a selected panel of methods from the exhaustive literature.

1.1. Related work

A straightforward extension of image inpainting methods to video inpainting consists in treating each frame independently. Nevertheless this technique fails as it does not take into account the high temporal correlation between successive frames that exists in video sequences.

When dealing with video inpainting, the patch-based texture synthesis approaches are the most efficient to produce more realistic results, since they allow to reconstruct textures whereas the geometrical methods lead to smoothed inpaintings [BBS01]. Such methods are inspired by the texture synthesis from non-parametric sampling [EL99]. The texture is synthesized by copying patches from the rest of the image after comparing the spatial neighborhood of the current pixel with all the patches lying within the known texture. Its natural extension to video inpainting has been proposed in [WSI04]. The mask of the video is filled by using spatio-temporal patches sampled from the whole known part of the video. The problem is posed as a global optimization scheme which makes it very computationally expensive.

DOI: 10.2312/PE/VMV/VMV10/123-130

Furthermore it assumes no illumination changes, a non deformable background and a static camera. Moreover, the objects have to move with a periodic motion and their size must not change significantly. A closely related area of research is dynamic texture synthesis. A dynamic texture is a sequence of images characterized by temporal stationarity. Examples of dynamic textures are videos representing flowing water, flames, moving grass... Non-parametric approaches extract different part of from the original video and fuse them together to obtain a new video. In [SSSE00], complete video frames are synthesized by assuming that the missing frame already appears entirely elsewhere in the input video sequence. Instead of copying entire frames, other methods use spatio-temporal patches [WL00, KSE*03] and then simply extend static texture synthesis method to dynamic textures.

Based on the seminal work of [KSE*03] for inpainting dynamic textures, the priority queue of spatio-temporal patches to inpaint videos has been enhanced in [KBBN05]. Spatio-temporal patches have also been used in [CFJ05] in which patch-based probability models (called epitomes) are learnt by compiling together a large number of spatio-temporal patches from the input video. The results obtained by using these epitomes are nevertheless over-smoothed.

More recent works on video inpainting separate the background from the foreground objects and inpaint these two parts independently. In [PSB07], background and foreground mosaics are created using optical flow. Foreground objects and stationary background are then both inpainted through a priority-based texture synthesis process. This method implies that the objects move in a repetitive way and that their size do not change significantly. The background is reconstructed by computing mosaics and is therefore assumed to be static while the camera motion has to be parallel to the plane of image projection. Similarly, in [ZXS05], a method based on motion layer estimation followed by motion compensation and texture synthesis has been proposed.

All previous frameworks present the same drawbacks as patch-based approaches for still images: they assume that there is redundant information and that the appropriate patches are available in the video. Moreover, the dimension of the search space becomes very high when processing a long video. The search space can be reduced using object tracking [JHM05]. In [SLCF06], the authors reduce the search space from 3D to 2D by slicing the volume along the motion manifold of the moving object. The foreground and background layers are here separated and objects in the foreground volume are rectified to compensate the perspective projection. To accelerate the foreground reconstruction, dynamic programing has been proposed in [VCZ09].

In the general case of inpainting a (potentially) moving object in a (potentially) moving scene, another solution consists in inpainting the optical flow. This motion inpainting can, for example, be done with a maximization a posteriori through a multi-resolution variational approach [LN04]. In [BKGR09], the motion inpainting is done through total variation anisotropic diffusion in order to reconstruct the corrupted regions of a dense optical flow. Spatio-temporal patches of local motion can also be used to reconstruct the flow [SMKT06]. This method is limited to small motions and is sensitive to noise. Moreover, the final color propagation scheme produces blurred results. With a similar idea, [Zha04] first reconstructs dense optical flow fields that are further used to copy the colors from previous frames.

1.2. Contributions

In this paper, we want to relax all the previous assumptions on static background and camera, and illumination or size conservations. Our objective is thus to replace any object in the video by the unknown background, so that we do not consider interaction of objects. Therefore this paper addresses the background inpainting problem. We propose to tackle this problem from the filtering point of view, by combining an optical flow reconstruction with an independent inpainting of each frames within a Kalman smoothing process. To this end, we first independently inpaint each frame of the video with any classical technique dedicated to still images. Next, we smooth these inpaintings along the whole point trajectories defined thanks to a backward and forward motion inpainting. Hence, we want to take advantage of the whole information of the inpainted sequence in order to reconstruct the textures and structures that can only be partially observed from the original images. The global process is summed up in Algorithm 1. It can handle illumination changes, dynamic and deformable backgrounds, moving cameras and erroneous image inpaintings. Nevertheless, it relies on the assumption that no foreground objects interact with the hole, except if those objects are the ones to be removed. Also our method considers that we have a good (though not necessarily perfect) inpainting of the first and last frames of the sequence.

Algorithm 1 Video inpainting

Given a sequence of images with their masks to inpaint

1. Independent image inpainting at each frame
2. Estimation of points trajectories, through motion estimation and reconstruction inside the masks (section 3)
3. Kalman smoothing of the observed colors along each trajectory (sections 2.2 and 4)
4. Reconstruction of the colors in the masks (section 2.2.2)

1.3. Overview of the paper

The paper is organized as follows. Section 2 reminds the principle of Kalman smoothing and explains how we apply it to video inpainting in section 3. Next, the inpainting of motion and the extraction of trajectories are described. Illumination changes and textures handling is explained in section 4 and some experiments are finally shown in section 5.

2. Kalman smoothing and its application to video inpainting

In this section, we explain the scheme of our algorithm for dynamic background completion.

2.1. A reminder on Kalman smoothing

The goal of Kalman filtering is to track a state $x_t \in \mathbb{R}^N$ knowing some observations $z_t \in \mathbb{R}^M$ of x_t at each time instant t. The observations may belong to a different space than the state, but can be related with the linear operator $H : \mathbb{R}^N \mapsto \mathbb{R}^M$, called observation model. The dynamic of the state is defined through the linear operator F, called state transition model. We consider the linear system:

$$\begin{cases} x_{t+1} &= Fx_t + \mu_t \\ z_t &= Hx_t + \nu_t, \end{cases} \tag{1}$$

where μ_t represents the noise of the dynamics, while ν_t models the measurement noise. These noises are considered to be Gaussian with covariance matrices Q_t and R_t respectively. Such systems of equations are generally initialized with a condition x_0 up to an initial noise ε_0 of covariance B_0.

2.1.1. Kalman Filtering

The aim of filtering methods is to estimate the state at each time t from its past measures: $x_{t:z_0,\cdots z_t}$. The best estimator of x_t knowing all the previous data is given by the conditional expected value $\hat{x}_t = \mathbb{E}[x_t|z_{0:t}]$ and its covariance $B_t = \mathbb{E}[(\hat{x}_t - x_t)(\hat{x}_t - x_t)^T]$. These two first moments can be computed with the standard Kalman filter [Kal60], as long as the dimension M is small enough. The Kalman filter is divided in two steps:

- The prediction step:

$$\hat{x}_{t|t-1} = F\hat{x}_{t-1},$$
$$B_{t|t-1} = Q_t + F\, B_{t-1}\, F^T.$$

- The correction step:

$$\hat{x}_t = \hat{x}_{t|t-1} + K_t(z_t - H\hat{x}_{t|t-1}),$$
$$B_t = B_{t|t-1} - K_t H B_{t|t-1},$$

where $K_t = B_{t|t-1}\, H^T (R_t + H\, B_{t|t-1}\, H^T)^{-1}$ is the Kalman gain matrix. The parameters of the Kalman filter are the covariance matrices B_0, R_t and Q_t.

2.1.2. Kalman smoothing

For some applications, using only the observations from the past to compute the state at current time might not be sufficient. In order to reconstruct smooth trajectories along a whole time interval $[t_0;t_f]$, Kalman smoothing is more appropriate, since its allows computing the state at each time t from the whole set of measurements: $x_{t|z_0\cdots z_f}$, $\forall t \in [t_0;t_f]$. The Kalman smoothing is applied to the result of the Kalman filtering in order to obtain the estimation $\hat{x}_t^{t_0\cdots t_f}$. In practice, the process requires the definition of the matrix J_t (see [YSS04] for more details):

$$J_t = B_t F^T (B_{t+1|t})^{-1}.$$

The smoothed value is then obtained with:

$$\hat{x}_t^{t_0\cdots t_f} = \hat{x}_t + J_t(\hat{x}_{t+1}^{t_0\cdots t_f} - F\hat{x}_t),$$

by initializing $\hat{x}_{t_f}^{t_0\cdots t_f} = \hat{x}_{t_f}$. The posterior covariance $B_t^{t_0\cdots t_f}$ of the variable $\hat{x}_t^{t_0\cdots t_f}$ can also been estimated. Initializing $B_{t_f}^{t_0\cdots t_f} = B_{t_f}$, the estimation is given by:

$$B_t^{t_0\cdots t_f} = B_t + J_t(B_{t+1}^{t_0\cdots t_f} - B_{t+1|t})J_t^T.$$

2.2. Application to inpainting

By applying Kalman smoothing to video inpainting, we are willing to incorporate temporal consistency between successive independent inpaintings. In other words we want to temporally smooth the reconstruction of each frame using the motion information of each pixel. Hence, we assume that some inpainted images $Z(x,t)$ are available for all frames $t \in [t_0;t_f]$ on the image domain $x \in \bar{\Omega}$. These observations may be obtained with any classical method allowing to fill-in independently the masks Ω_t of the sequence (*e.g.* [CPT04, Tsc06]).

We also assume that the dynamic can result from a dense motion field reconstruction (see section 3). The introduction of these variables into the system (1) leads to the system:

$$\begin{cases} I_{t+1}(x) = I_t(x + w^b(x,t+1)) + \mu_t \\ z_{t+1}(x) = I_{t+1}(x) + \nu_{t+1}, \end{cases} \tag{2}$$

where x is a pixel of mask Ω_{t+1}, $I_t(x)$ its reconstructed color at time t and $w^b(t)$ is the backward dense optical flow field between times t and $t-1$. Applying the Kalman smoothing process to such a high dimensional system leads to the inversion of huge matrices and is therefore not feasible for large images. The solution we here propose consists in processing independently each point of the masks and smoothing the color along the trajectory of the point.

2.2.1. States, dynamics and observations

Let us describe in more details the state variable as well as the dynamics and observation equations.

2.2.1.1. State variable When dealing with the problem of inpainting filtering, the main trouble comes from the definition of the state variable. Indeed, as the mask size $|\Omega_t|$ can change at every frame, it is therefore impossible to define a discrete spatial variable representing the area to inpaint in time. Our claim is to define a pixel-based approach for filtering inpainted color values.

We then consider $p(t)$, the fixed 2D trajectory of a point in the video, obtained from a motion inpainting method (see section 3). Let $p(t_0^p)$ define the last position of the point before entering an inpainting mask (at time $t_0^p \geq t_0$) and $p(t_f^p)$ the position when leaving the inpainting masks (at time $t_f^p \leq t_f$). The goal is to filter the color value of the point p with respect to all the observations $Z(p(t),t)$ in the time range $[t_0^p, t_f^p]$. The state variable is $I(p(t),t)$, the color of the point p

that can change with time. For the sake of clarity, this state will be denoted as $I(p,t)$. The initial condition is given by $I(p,t_0^p) = Z(p,t_0^p) + \varepsilon_0$, the noise ε_0 being defined by the covariance B_0. Obviously, the initial condition will either be an original color (if $p(t_0^p) \notin \Omega_{t_0^p}$) or a color of the reconstruction of the first image (if $p(t_0^p) \in \Omega_{t_0^p}$ and $t_0^p = t_0$).

2.2.1.2. Dynamics To represent the dynamic of the color $I(p,t)$ of point p, we simply assume that the color is preserved through time, up to a noise μ_t, the model is then:

$$I(p,t+1) = I(p,t) + \mu_t,$$

where the dynamic noise μ_t is defined by the scalar covariance Q_t. The dynamic operator F is the identity matrix.

2.2.1.3. Observations The first simple idea is to consider as observation the color values $Z(p,t)$ for $t \in [t_0^p, t_f^p]$. However, as detailed in section 4, such filtering smoothes the observed value and is not able to deal with bad observations. Therefore we here propose to use patches $Z_s(p,t)$ of size $M = s \times s$ (or $M = 3 \times s \times s$ for color images) taken from the image $Z(t)$. They are centered on the closest pixel to the position $p(t)$, instead on $p(t)$ directly in order to avoid the smoothing that would result from a bilinear interpolation of the observations. We use the different pixels of a patch as if they were different observations of the same state. The derivation of the Kalman filtering and smoothing equations for multiple observations such that

$$\begin{cases} x_{t+1} &= F x_t + \mu_t \\ z_t^i &= H x_t + \nu_t^i, \ \forall i = 1 \cdots M \end{cases} \tag{3}$$

can be obtained similarly as for one observation following for example [YSS04]. It leads to defining the following pseudo-observation

$$\tilde{z}_t = \left(\sum_{j=1}^{M} (R_t^j)^{-1} \right)^{-1} \sum_{i=1}^{M} (R_t^i)^{-1} z_t^i, \tag{4}$$

associated to its pseudo-covariance matrix:

$$\tilde{R}_t = \left(\sum_{i=1}^{M} (R_t^i)^{-1} \right)^{-1}, \tag{5}$$

and to a gain defined as $\tilde{K}_t = B_{t|t-1} H^T \left[\tilde{R} + H B_{t|t-1} H' \right]^{-1}$. Therefore, it is similar to computing a weighted mean of all the observations, the weight being dependent on the confidence of each observation. The patches are finally used in the observation equation as:

$$\tilde{Z}(p,t) = I(p,t) + \nu_t, \tag{6}$$

with

$$\tilde{Z}(p,t) = \left(\sum_{j=1}^{M} (R_t^j)^{-1} \right)^{-1} \sum_{i=1}^{M} (R_t^i)^{-1} Z_s(p+i,t). \tag{7}$$

and ν_t computed such that equation (5) is verified. Details on the computation of the noises will be given in section 4. The Markov chain for applying Kalman filtering on one point is summed up on figure 1.

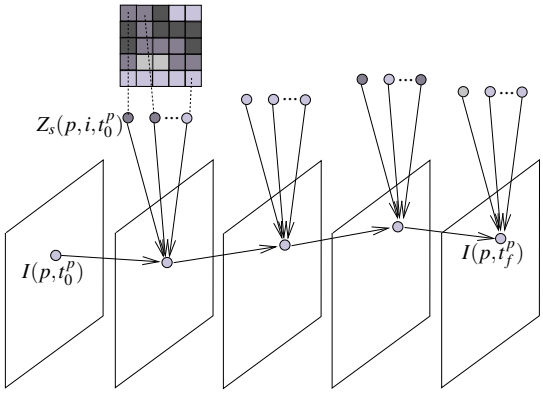


Figure 1: *Markov chain representing the filtering process for one point p. The observations are here* 5×5 *patches, each pixel of the patch being taken as a different observation.*

2.2.2. Images reconstruction

Once the colors have been smoothed along trajectories, we need to transfer them into the masks Ω_t. The color of a pixel $x \in \Omega_t$ is given by the median value of the colors $I(p,t)$ of all the points p crossing this pixel. We consider that a trajectory crosses a pixel if it passes through the 8-neighborhood of this pixel. We know that there will be at least one from the definition of trajectories (see subsection 3).

3. Extracting point trajectories

In order to extract the trajectories of the points, we have to inpaint the motion into the holes Ω_t. The dense optical flow field is first computed using a convexified multi-label approach [PBGC10]. Then its reconstruction within the mask is obtained by applying the texture synthesis method from [CPT04] on motion patches. In practice the mask is enlarged before doing the inpainting so that the possibly erroneous flow vectors at the boundary are also inpainted.

We respectively denote by $w^b(t)$ and $w^f(t)$ the backward (between t and $t-1$) and forward (between t and $t+1$) dense motion fields $w(x,t) = [u(x,t), v(x,t)]$. Using these fields for the whole sequence we can now define trajectories. Let $p(t_0)$ be a point of the mask Ω_{t_0}, its position at next frame is: $p(t_1) = p(t_0) + w^f(p,t_0)$. Doing so recursively, we can extract the whole trajectory of the point in the video. The trajectory ends when the point leaves the masks (*i.e.* when $p(t) \notin \Omega_t$) or when the last frame of the sequence is reached. In practice, a bilinear interpolation is used to compute $w(p,t)$ (in case the point $p(t)$ does not belong to the grid of pixels).

With such a process, not all the pixels of all the masks Ω_t are processed. Some new trajectories are therefore created for all pixel $x \in \Omega_t$, $t > t_0$ which have not been previously crossed by a trajectory. As mentioned before, we consider that a pixel has been crossed if a trajectory passes within

its 8-neighborhood. In order to improve the results for these new trajectories, we also compute the backward trajectory using the inpainted backward motion from image $I(t)$ to image $I(t-1)$. Here again, the trajectory is stopped when leaving a mask or when reaching the first frame of the video.

4. Dealing with textures and illumination changes

In this section, we will explain how the observation model H and the observation noise R_t are defined. If using directly the independently inpainted images, the results obtained with the Kalman smoothing are often too blurred. This is not surprising as the Kalman filtering consists in doing a weighted mean between the prediction and the observation. Therefore, imagine that an observed pixel is white while the prediction is black, the resulting color will be gray. In such a case before taking into account the observation, one should ensure that it is correct by comparing it to the first and last colors of the trajectory: if the first color is white and the last one is black, getting a gray pixel seems more coherent. We then give more importance to an observation that is close to the linear interpolation from $Z(p,t_0^p)$ to $Z(p,t_f^p)$. To do so let us define the value:

$$r_i(p,t) = \exp\left(\frac{-D^2}{\sigma^2}\right),$$

with,

$$D = \left\| Z(p,t) - \left[\frac{t_f^p - t}{t_f^p - t_0^p} Z(p,t_0^p) + \frac{t - t_0^p}{t_f^p - t_0^p} Z(p,t_f^p)\right]\right\|,$$

where $\|.\|$ defines euclidean norm (computed for the three channels for color images), σ is a parameter monitoring the deviation to the expected color and set by hand ($\sigma = 5$). The covariance of the observation noise is now defined as:

$$R_i(p,t,c,c') = \begin{cases} \rho_t \; r_i(p,t), & \text{if } c = c' \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

ρ_t being a parameter giving more or less importance to the observations with respect to the dynamic, and c refering to the color channel. One can verify that if the observation is far from the expected color, then r_i and R are big, which leads to not trusting the observation. Abrupt changes of colors (such as an impulse function) are then discarded.

However, it may happen that such an observation far from the linear interpolation should be taken into account. For example one could think of a rectangular function, in which case it is better to consider the observations (see figure 2). Equation (8) must then not be used if an observation is close to its temporal neighbors but far from the mean value of the observations along the trajectory. Let m_T and σ_T be the mean and standard deviation of the observation computed on the whole trajectory of the point, and m_t and σ_t the mean and standard deviation computed on a temporal window centered at time t. The covariance now reads:

$$R(p,t) = \begin{cases} R_i(p,t) \text{ if } \|Z(p,t) - m_T\| > 2\sigma_T, \|Z(p,t) - m_t\| < 2\sigma_t \\ \rho \qquad \text{otherwise,} \end{cases} \tag{9}$$

which allows robustness to illumination changes and bad observations. In practice, the size of the temporal window is 5.

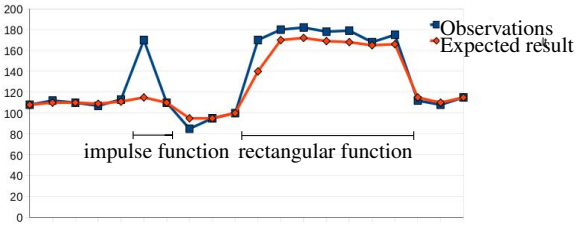


Figure 2: *The first step is an isolated change of color which should not be taken into account in the smoothing process. The second one is as a rectangular function for which the resulting color should be close to the observations.*

Finally, as mentioned in section 2, we consider patches of observations to reduce the blur in the results. The pseudo observation and pseudo-covariance are computed combining equations (7) (5) and (9). These patches being treated as multiple observations. Remark that instead of using patches, one could also use different 2D inpainting (texture synthesis [CPT04] or diffusion [Tsc06]) for still images in order to get different kind of information on textures and structures.

5. Experiments

In this section, after explaining parameters settings, we will describe the experiments on four sequences.

5.1. Setting the parameters

The parameters of the Kalman smoothing process are the covariance matrix Q_t, the observation influence ρ_t and the covariance B_0 of the initial condition. The covariance matrices Q_t and B_0 are diagonal matrices (3×3 for color sequences), such that $Q_t = q_t \mathrm{Id}$ and $B_0 = \rho_0 \mathrm{Id}$. Therefore, there are two parameters to set for each trajectory and at each time: q_t and ρ_t. In all our experiments, as we have no knowledge on the sequences, we set $\rho_t = 1$ and $q_t = 1$, $\forall t \in]t_0^p; t_f^p[$ and $\forall p$. That way, we do not favour neither the predictions nor the observations. However, in order to define the value of the noise for the first and last times of a trajectory, we distinguish the following cases. If the trajectory starts and ends when it leaves the mask, we can be very confident on the observations for these two times. This is done by setting $\rho_{t_f^p} = 0$ and $\rho_{t_0^p} = 0$. If the trajectories starts at the first frame or end at the last frame, we either set $\rho_t = 0$ or $\rho_t = 1$ depending on whether or not we expect the algorithm to modify the reconstructions of the first and last frames.

The other parameters of the whole process are the ones for the image inpainting algorithm and for the optical flow estimation. For each experiment, we precise which method and parameters were used but note that similar results could be obtained with other choices.

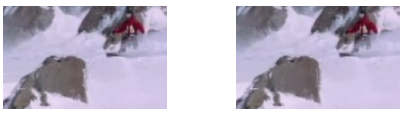
5.2. Results

The first result (figure 3) presents a sequence of a skier in which the objective is to remove the watermarking logo. The observations were obtained with [Tsc06] and the following parameters: $p1 = 0.001, p2 = 100, \sigma = 4, dt = 50$ and 100 iterations. For this result, we corrected the inpainting of the first ($t = 1$) and last ($t = 35$) frames by hand and set $\rho_{t_0^p} = 0$ and $\rho_{t_f^p} = 0$. We were able to correctly reconstruct the snow and the rocks and to discard the errors within the observations. By simply predicting the first reconstruction with optical flow (by setting $q_t = 0$ all along), good results can also be obtained. Nevertheless, adding the observations enables to correct the errors at the boundaries (figure 4).

The constant mask used for all the frames of the sequence

Original images Observations Filtered results

Figure 3: *Results on the ski sequence for frames 1, 5, 20, 35.*



Prediction (i.e. q=0) Smoothing (with q=r)

Figure 4: *Comparison between the smoothing result and the prediction on one frame ($t = 22$) of the ski sequence.*

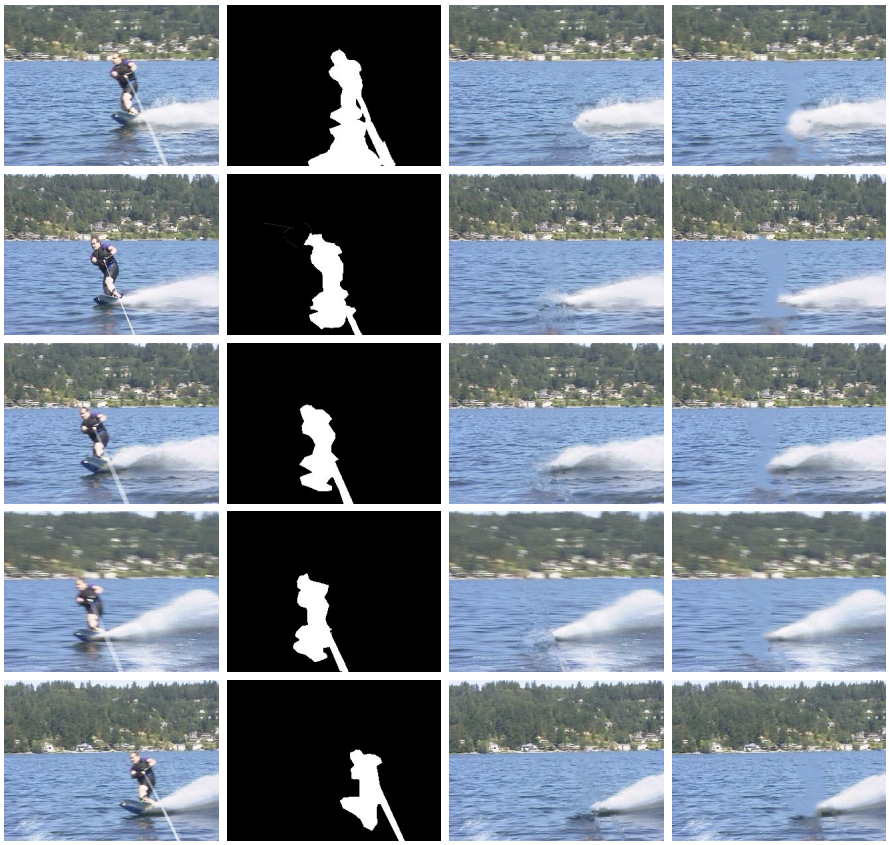
Figures 5 and 6 demonstrate how our method handles the reconstruction of dynamic textures. For these two sequences, we set $\rho_{t_0^p} = 0$ and $\rho_{t_f^p} = 0$, and obtained the observations using the algorithm from [CPT04] with 9×9 patches. On figure 5, it is interesting to remark how our process adds temporal consistency compared to independent inpainting (obviously the temporal consistency is better visible by watching the videos associated to these results). In particular, the white dandelion highlighted with the red circle is correctly reconstructed in each frame. As can be noticed on figure 6, our method may produce blur within the reconstructed texture despite the use of the textures handler (section 4). This is due both to the color reconstruction scheme and definition of the observation noise (section 4). Some more intensive work should therefore be dedicated to this problem.

The constant mask used for all the frames of the sequence

Original images Observations Filtered results

Figure 5: *Result on the grass sequence for frames 1, 6, 16, 31.*

The last result, presented on figure 7, is on a highly dynamic sequence in which we aim at removing the wakeboarder. In particular, the sequence presents motion blur and a dynamic and deformable background. The observations were obtained with the algorithm from [CPT04] with 11×11 patches. Contrary to previous experiments, we here do not completely trust the reconstructions of the first and last frames and then set $\rho_{t_0^p} = 1$ and $\rho_{t_f^p} = 1$. The result obtained is encouraging as the method is able to correctly reconstruct both the trees and the water, and to extend the wave inside the mask. To prove the validity of our method, we compared the results with the ones obtained using the technique from [WSI04], with 100 iterations, 3 scales and $5 \times 5 \times 3$ patches. This approach produces highly blurred results, mainly because pixels are synthesized by a weighted



Original images Masks Filtered results Results of [WSI04]

Figure 7: *Result on the wake-boarder sequence for frames 1, 10, 15, 20, 30.*

average of the best candidates in the video. Note that we tested several patches size, all leading to blurred results.

The computational time for the complete video process depends on the method used to get the observations and on the optical flow computation. Indeed steps (iii) and (iv) of Algorithm 1, that only concern Kalman smoothing are very fast and could probably be processed in real time with an optimized implementation.

6. Conclusion

In this paper we have proposed a simple framework for inpainting the background in video sequences. The technique is based on Kalman smoothing along points trajectories using independent image inpaintings as observations. The trajectories are the results of dense motion estimation and inpainting with a non-parametric approach. Results of the proposed process are very promising and open to several future works. First, a more extensive study should be done on textures handling to avoid having blurred reconstruction. Furthermore, as mentioned at the end of section 4, we are planning to test the method using jointly observations obtained from different image inpainting methods. Obviously, the whole process could be added to the method that mainly aims at inpainting foreground moving objects. Finally, its extension to stereo video inpainting could be also considered.

Acknowledgment

A. Bugeau, P. Gargallo, O. D'Hondt, A. Hervieu and N. Papadakis hold a grant from the Torres Quevedo Program of the Spanish Science and Innovation Ministry, cofinanced by the European Social Fund. A. Hervieu, P. Gargallo and V. Caselles acknowledge partial support by IP project "2020 3D Media: Spacial Sound and Vision", Financed by EC. V. Caselles also acknowledges partial support by MICINN project, reference MTM2009-08171, by GRC reference 2009 SGR 773 and by "ICREA Academy" prize for excellence in research funded both by the Generalitat de Catalunya. This work was partially funded by Mediapro through the Spanish project CENIT-2007-1012 i3media and by the Centre for the Industrial & Technological Development within the Ingenio 2010 initiative.

References

[ACS09] ARIAS P., CASELLES V., SAPIRO G.: A variational framework for non-local image inpainting. In *In Proceedings of*

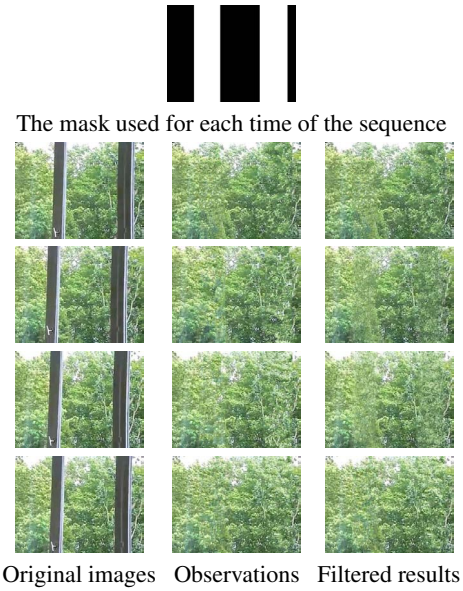


Figure 6: *Result on the tree sequence for frames 1, 11, 31, 41.*

the Energy Minimization Methods in Computer Vision and Pattern Recognition (2009), pp. 345–358. 1

[ALM08] AUJOL J.-F., LADJAL S., MASNOU S.: Exemplar-based inpainting from a variational point of view. 2008. 1

[BBCSar] BUGEAU A., BERTALMIO M., CASELLES V., SAPIRO G.: A comprehensive framework for image inpainting. *IEEE Transactions on Image Processing* (to appear). 1

[BBS01] BERTALMIO M., BERTOZZI A., SAPIRO G.: Navier-stokes, fluid dynamics, and image and video inpainting. In *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2001). 1

[BKGR09] BERKELS B., KONDERMANN C., GARBE C., RUMPF M.: Reconstructing optical flow fields by motion inpainting. In *In Proceedings of the Energy Minimization Methods in Computer Vision and Pattern Recognition* (2009). 2

[BM07] BORNEMANN F., MÄRZ T.: Fast image inpainting based on coherence transport. *Journal of Mathematical Imaging and Vision 28*, 3 (2007), 259–278. 1

[BSCB00] BERTALMIO M., SAPIRO G., CASELLES V., BALLESTER C.: Image inpainting. In *SIGGRAPH: ACM Special Interest Group on Computer Graphics and Interactive Techniques* (2000). 1

[CFJ05] CHEUNG V., FREY B., JOJIC N.: Video epitomes. In *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2005). 2

[CPT04] CRIMINISI A., PÉREZ P., TOYAMA K.: Region filling and object removal by exemplar-based inpainting. *IEEE Transactions on Image Processing 13*, 9 (2004), 1200–1212. 1, 3, 4, 5, 6

[DSC03] DEMANET L., SONG B., CHAN T.: *Image Inpainting by Correspondence Maps: a Deterministic Approach*. Tech. Rep. 03-04, UCLA CAM R, August 2003. 1

[EL99] EFROS A., LEUNG T.: Texture synthesis by non-parametric sampling. In *In Proceedings of the International Conference on Computer Vision* (1999). 1

[JHM05] JIA Y. T., HU S. M., MARTIN R. R.: Video completion using tracking and fragment merging. 601–610. 2

[Kal60] KALMAN R.: A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering* (1960), 35–45. 3

[KBBN05] KUMAR S., BISWAS M., BELONGIE S., NGUYEN T.: Spatio-temporal texture synthesis and image inpainting for video applications. In *In Proceedings of the International Conference on Image Processing* (2005). 2

[KSE*03] KWATRA V., SCHÖDL A., ESSA I. A., TURK G., BOBICK A. F.: Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph. 22*, 3 (2003), 277–286. 2

[LN04] LAUZE F., NIELSEN M.: A variational algorithm for motion compensated inpainting. In *In Proceedings of the British Machine Vision Conference* (2004). 2

[MM98] MASNOU S., MOREL J.: Level-lines based disocclusion. In *In Proceedings of the International Conference on Image Processing* (1998). 1

[PBGC10] PAPADAKIS N., BAEZA A., GARGALLO P., CASELLES V.: Polyconvexification of the multi-label optical flow problem. In *In Proceedings of the International Conference on Image Processing* (2010). 4

[PSB07] PATWARDHAN K. A., SAPIRO G., BERTALMÍO M.: Video inpainting under constrained camera motion. *IEEE Transactions on Image Processing 16*, 2 (2007), 545–563. 2

[SLCF06] SHEN Y., LU F., CAO X., FOROOSH H.: Video completion for perspective camera under constrained motion. In *In Proceedings of the International Conference on Pattern Recognition* (2006). 2

[SMKT06] SHIRATORI T., MATSUSHITA Y., KANG S. B., TANG X.: Video completion by motion field transfer. In *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2006). 2

[SSSE00] SCHÖDL A., SZELISKI R., SALESIN D. H., ESSA I.: Video textures. In *SIGGRAPH: ACM Special Interest Group on Computer Graphics and Interactive Techniques* (2000), pp. 489–498. 2

[Tsc06] TSCHUMPERLÉ D.: Fast anisotropic smoothing of multi-valued images using curvature-preserving pde's. *International Journal of Computer Vision 68*, 1 (2006), 65–82. 1, 3, 5, 6

[VCZ09] VENKATESH M. V., CHEUNG S.-C. S., ZHAO J.: Efficient object-based video inpainting. *Pattern Recognition Letters 30*, 2 (2009), 168–179. 2

[WL00] WEI L., LEVOY M.: Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH: ACM Special Interest Group on Computer Graphics and Interactive Techniques* (2000). 2

[WSI04] WEXLER Y., SHECHTMAN E., IRANI M.: Space-time video completion. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on* (2004), vol. 1. 1, 6, 7

[YSS04] YU B. M., SHENOY K. V., SAHANI M.: *Derivation of Kalman filtering and smoothing equations*. Tech. rep., Technical report, Stanford University, 2004. 3, 4

[Zha04] ZHAO W.-Y.: Motion-based spatial-temporal image repairing. In *In Proceedings of the International Conference on Image Processing* (2004). 2

[ZXS05] ZHANG Y., XIAO J., SHAH M.: Motion layer based object removal in videos. In *IEEE Workshop on Applications of Computer* (2005). 2