

# Flow and Color Inpainting for Video Completion

Michael Strobel, Julia Diebold, Daniel Cremers

Technical University of Munich, Germany  
{m.strobel,julia.diebold,cremers}@tum.de

**Abstract.** We propose a framework for temporally consistent video completion. To this end we generalize the exemplar-based inpainting method of Criminisi *et al.* [7] to video inpainting. Specifically we address two important issues: Firstly, we propose a color and optical flow inpainting to ensure temporal consistency of inpainting even for complex motion of foreground and background. Secondly, rather than requiring the user to hand-label the inpainting region in every single image, we propose a flow-based propagation of user scribbles from the first to subsequent video frames which drastically reduces the user input. Experimental comparisons to state-of-the-art video completion methods demonstrate the benefits of the proposed approach.

**Keywords:** video completion, video inpainting, disocclusion, temporal consistency, segmentation, optical flow

## 1 Introduction

Videos of natural scenes often include disturbing artifacts like undesired walking people or occluding objects. In the past ten years, the technique of replacing disruptive parts with visually pleasing content grew to an active research area in the field of image processing. The technique is known as video inpainting and has its origin in image inpainting. While image inpainting has been researched very active in the past years the problem of video inpainting has received much less attention. Due to the additional temporal dimension in videos, new technical challenges arise and make calculations much more complex and time consuming. At the same time, video completion has a much larger range of applications, including professional post-productions or restoration of damaged film.

In this work, we focus on two central challenges in video completion, namely temporal consistency and efficient mask-definition.

### 1.1 Related Work

The literature on image inpainting can be roughly grouped into two complementary approaches, namely inpainting via partial differential equations (PDEs) and exemplar-based inpainting. PDE-based inpainting was first proposed by Masnou and Morel [13, 12] and popularized under the name of inpainting by Bertalmio *et al.* [4, 3]. The key idea is to fill the inpainting region by propagating isolines of

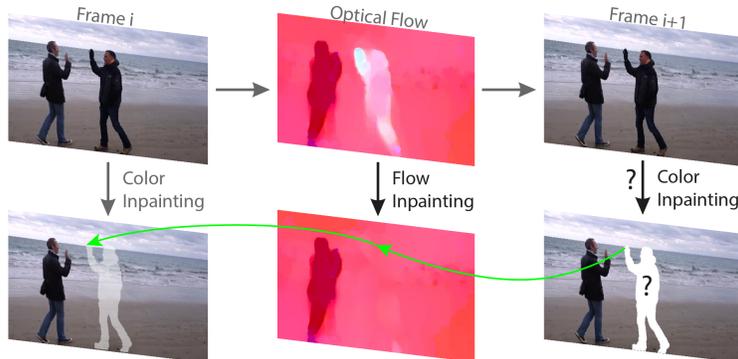


Fig. 1: **Sketched approach.** We propose an efficient algorithm for semi-automatic video inpainting. In particular, we impose temporal consistency of the inpainting not by a tedious sampling of space-time patches but rather by a strategy of flow- and color inpainting. We inpaint the optical flow and subsequently modify the distance function in an exemplar-based image inpainting such that consistency with corresponding patches in previous frames is imposed.

constant color from the surrounding region. These techniques provide pleasing results for filling small regions, for example to remove undesired text or scratches from images. For larger regions, however, the propagation of similar colors creates undesired smoothing effects. To account for this shortcoming, texture synthesis techniques were promoted, most importantly exemplar-based techniques [9, 1, 8] which can fill substantially larger inpainting regions by copy-pasting colors from the surrounding areas based on patch-based similarity. Criminisi *et al.* [6, 7] presented an approach which combines the two methods to one efficient image inpainting algorithm. The algorithm works at the image patch level and fills unknown regions effectively by extending texture synthesis with an isophote guided ordering. This automatic priority-based ordering significantly improves the quality of the completion algorithm by preserving crucial image structures.

Patwardhan *et al.* [17, 18] and Werlberger [24] extended and adapted Criminisi *et al.*'s [7] method for video inpainting. The approach of Patwardhan *et al.* is using a 5D patch search and takes motion into account. Their approach leads to satisfying results as long camera movement matches some special cases. We are not restricted to specific camera motion.

The idea of using graph cuts for video inpainting was recently introduced by Granados *et al.* [11]. They propose a semi-automatic algorithm which optimizes the spatio-temporal shift map. This algorithm presents impressive results however, the approach only has very limited practicability as the runtime takes between 11 and 90 hours for 200 frames.

Newson *et al.* [14, 15] provided an important speed-up by extending the PatchMatch algorithm [2] to the spatio-temporal domain thereby drastically accelerating the search for approximate nearest neighbors. Nevertheless, the runtime for high-resolution videos is about 6 hours for 82 frames.

## 1.2 Contributions

We propose a method for video completion which resolves several important challenges:

- + We propose a method to interactively determine the inpainting region over multiple frames. Rather than hand-labeling the inpainting region in every single frame, we perform a flow-based propagation of user-scribbles (from the first frame to subsequent frames), followed by an automatic foreground-background segmentation.
- + We introduce temporal consistency not by sampling spatio-temporal patches, but rather by a combination of color- and flow-based inpainting. The key idea is to perform an inpainting of the optical flow for the inpainting region and subsequently perform an exemplar-based image inpainting with a constraint on temporal consistency along the inpainted optical flow trajectories - see Figure 1. As a consequence, the proposed video completion method can handle arbitrary foreground and background motion in a single approach and with substantially reduced computation time.
- + The inpainting is computed without any pre- or post-processing steps. An efficient GPU-based implementation provides pleasing video completion results with minimal user input at drastically improved runtimes compared to state-of-the-art methods.

## 2 Interactive Mask-Definition

In [3, 11, 25, 26] manual labeling of the inpainting region in all frames of the videos is needed. This makes video editing an extremely tedious and somewhat unpleasant process. We present a simple tool for *interactive mask-definition* with minimal user input. The requirements for such a tool include: (i) an intuitive user interface (ii) a robust mask definition and (iii) a real-time capable algorithm.

The method of Nieuwenhuis and Cremers [16] provides a user-guided image segmentation algorithm that generates accurate results even on images with difficult color and lighting conditions. The user input is given by user scribbles drawn on the input image. The algorithm analyzes the spatial variation of the color distributions given by the scribbles. Thanks to their parallel implementation, computation times of around one second per frame can be obtained.

Based on this user input, we (i) automatically relocate the scribbles throughout the video sequence via optical flow and (ii) frame-wise apply the image segmentation method according to Nieuwenhuis and Cremers [16].

### 2.1 Scribble Relocation via Optical Flow

To transport scribbles over time we use the optical flow method of Brox *et al.* [5] which computes the displacement vector field  $(u, v)$  by minimizing an energy functional of the form:

$$E(u, v) = E_{Data} + \alpha E_{Smooth} \quad (1)$$

with some regularization parameter  $\alpha > 0$ . The data term,  $E_{Data}$ , measures the global deviations from the grey value and gradient constancy assumption. The smoothness term,  $E_{Smooth}$ , is given by the discontinuity-preserving total variation.

Figure 2 b) shows the optical flow between two frames of the image sequence by Newson *et al.* [15]. We use this flow to transport the scribbles from frame to frame (Figure 2 a,c). Green scribbles are placed on the region to be inpainted and yellow ones on the search space for the inpainting algorithm. Optionally, red scribbles can be used to mark unrelated image parts in order to shrink the search space. Depending on the user scribbles, a two- or three-region segmentation according to Nieuwenhuis and Cremers [16] is computed.

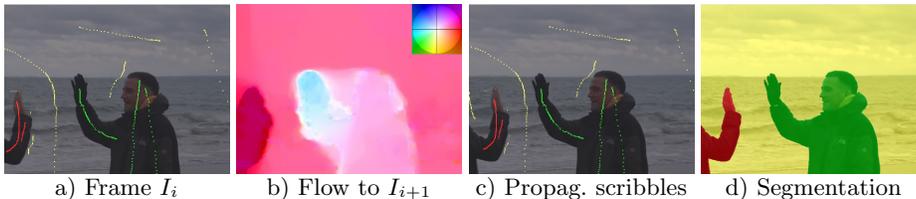


Fig. 2: **Automatic segmentation by scribble propagation via optical flow.** Scribbles are placed on the first frame and propagated to the next frames by optical flow. Segmentation is computed based on the transported scribbles.

## 2.2 Segmentation according to Nieuwenhuis and Cremers

Let  $I : \mathcal{I} \rightarrow \mathbb{R}^d$  denote the input frame defined on the domain  $\mathcal{I} \subset \mathbb{R}^2$ . The task of segmenting the image plane into a set of  $n$  pairwise disjoint regions  $\mathcal{I}_i$ :  $\mathcal{I} = \bigcup_{i=1}^n \mathcal{I}_i$ ,  $\mathcal{I}_i \cap \mathcal{I}_j = \emptyset \quad \forall i \neq j$  can be solved by computing a labeling  $u : \mathcal{I} \rightarrow \{1, \dots, n\}$ , indicating which of the  $n$  regions each pixel belongs to:  $\mathcal{I}_i = \{x \mid u(x) = i\}$ . The segmentation time for a video sequence can be speed-up by initializing the indicator function  $u$  with the resulting segmentation of the previous frame.

We compute a segmentation of each video frame by minimizing the following energy [16]:

$$E(\mathcal{I}_1, \dots, \mathcal{I}_n) = \frac{\lambda}{2} \sum_{i=1}^n \text{Per}_g(\mathcal{I}_i) + \lambda \sum_{i=1}^n \int_{\mathcal{I}_i} f_i(x) dx,$$

where  $f_i(x) = -\log \hat{\mathcal{P}}(I(x), x \mid u(x) = i)$ .  $\text{Per}_g(\mathcal{I}_i)$  denotes the perimeter of each set  $\mathcal{I}_i$ ,  $\lambda$  is a weighting parameter. The expression  $\hat{\mathcal{P}}(I(x), x \mid u(x) = i)$  denotes the joint probability for observing a color value  $I$  at location  $x$  given that  $x$  is part of region  $\mathcal{I}_i$ . It can be estimated from the user scribbles. For further details of the segmentation algorithm we refer to [16].

To summarize, our inpainting method brings along a tool which allows the user to quickly define the respective regions on the first video frame, and all the remaining calculations are working automatically. In contrast, state-of-the-art methods require the user to manually draw an exact mask on each single video frame [3, 11, 25, 26] or work with an inflexible bounding box [20].

### 3 Flow and Color Inpainting for Video Completion

The major challenge in video inpainting is the temporal dimension: The inpainted regions have to be consistent with the color and structure around the hole, and additionally temporal continuity has to be preserved. When applying image inpainting methods frame by frame, the inpainted videos show artifacts, like ghost shadows or flickering [20]. Several investigations have been done in the past years towards a temporally coherent video completion. State-of-the-art methods, however, have some drawbacks: several pre- and post-processing steps are required [14, 20], only specific camera motions can be handled [11, 14, 18, 26] and the calculations are extremely time consuming [10, 11, 14, 26].

We propose a novel approach inspired by the exemplar-based image inpainting by Criminisi *et al.* [7] overcoming these problems. We apply inpainting to the optical flow and define a *refined distance function* ensuring temporal consistency in video inpainting. No additional pre- or post-processing steps are required.

#### 3.1 Inpainted Flow for Temporal Coherence

In a temporally consistent video sequence, the inpainted region follows the flow of its surrounding region. Figure 3 a) shows a person who should be removed from the video sequence. The desired patches clearly should not follow the hand of the person, but the flow of the sea. To find the best matching patches, Criminisi *et al.* [7] consider the colors around the hole. We additionally claim a similarity to the patch which naturally flows into this position. This flow is obtained by *inpainting the original flow* - see Figure 3 d).

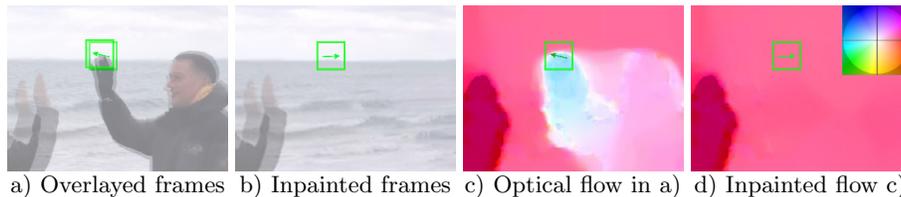


Fig. 3: **Inpainted flow ensures temporal consistency.** In order to ensure temporal consistency, we propose to inpaint the optical flow and additionally request the found patch to be similar to its origin. The inpainted flow d) should be approximately the flow of the inpainted video sequence.

#### 3.2 Flow Inpainting

For the inpainting of the optical flow we extended the Telea-Inpainting [21] to optical flow. Telea-Inpainting is a fast PDE based approach and hence particularly suited to fill missing parts in optical flow images. Let  $\Omega$  denote the hole in the optical flow  $F$  which has to be replaced,  $\delta\Omega$  the contour of the hole and  $\Omega^c$  the search region (complement of  $\Omega$ ). Telea-Inpainting approximates the value

of a pixel  $p$  on the boarder of the fill-front  $\delta\Omega$  by a first order Taylor expansion combined with a normalized weighting function  $w(p, q)$  for  $q \in B_\epsilon(p)$  and  $\epsilon > 0$ :

$$\hat{F}(p) = \frac{\sum_{q \in B_\epsilon(p) \cap \Omega^c} w(p, q) [F(p) - \nabla F(q)(p - q)]}{\sum_{q \in B_\epsilon(p) \cap \Omega^c} w(p, q)}.$$

The pixel values are propagated into the fill region along the isophotes by solving the eikonal equation:  $|\nabla T| = 1$  on  $\Omega$ ,  $T = 0$  on  $\delta\Omega$  using the Tsitsiklis algorithm [19, 22]. The solution  $T$  of the eikonal equation describes the distance map of the pixels inside  $\Omega$  to its boundary  $\delta\Omega$ .

### 3.3 Exemplar-Based Inpainting

For the general inpainting, we focused on the exemplar-based inpainting method for region filling and object removal by Criminisi *et al.* [7]. This well known *best-first algorithm* uses texture synthesis and successfully propagates continuities of structures along isophotes to the inpainting region.

**Computation of the filling priorities** Let  $\Omega$  denote the hole to be replaced and  $\delta\Omega$  the contour of the hole. For each pixel  $p$  along the contour  $\delta\Omega$ , a filling priority  $P(p)$  is computed.  $P(p)$  is defined as the product [7]:

$$P(p) = ((1 - \omega) C(p) + \omega) D(p). \quad (2)$$

$\omega \in \mathbb{R}$  is a weighting factor.  $C(p) := \frac{\sum_{q \in \Psi_p \cap (\Omega - \Omega)} C(q)}{|\Psi_p|}$  is called the *confidence term* and  $D(p) := \frac{|\nabla I_p^\perp \cdot n_p|}{\alpha}$  the *data term*.  $|\Psi_p|$  denotes the area of the patch  $\Psi_p$ ,  $\alpha$  is a normalization factor and  $n_p$  is a unit vector orthogonal to  $\delta\Omega$  in the point  $p$ .

The confidence term  $C(p)$  measures the amount of reliable information surrounding the pixel  $p$ . The intention is to fill first those patches which have more of their pixels already filled. Wang *et al.* [23] introduced the weighting factor  $\omega$  to control the strong descent of  $C(p)$  which accumulates along with the filling. The data term  $D(p)$  is a function of the strength of isophotes hitting the contour of the hole. This factor is of fundamental importance because it encourages linear structures to be synthesized first. The pixel  $\hat{p}$  with the highest priority:  $\hat{p} = \arg \max_{p \in \delta\Omega} P(p)$  defines the center of the target patch  $\Psi_{\hat{p}}$  which will be inpainted.

**Search for the best matching patch** In the next step, the patch  $\Psi_{\hat{q}}$  which best matches the target patch  $\Psi_{\hat{p}}$  is searched within the source region  $\Phi$ . Formally [7]:

$$\Psi_{\hat{q}} = \arg \min_{\Psi_q \in \Phi} d(\Psi_{\hat{p}}, \Psi_q), \quad (3)$$

where the distance  $d(\cdot, \cdot)$  is defined as the sum of squared differences (SSD) of the already filled pixels in the two patches.

This distance, however, is only designed for image inpainting. For the problem of video inpainting the additional temporal dimension is not considered. We present a refined distance function, modeled explicitly to maintain temporal consistency along the video frames. The detailed definition follows in the next Section 3.4.

**Copy and refresh** When the search for the best matching patch  $\Psi_{\hat{q}}$  is completed, the target region  $\Psi_{\hat{p}} \cap \Omega$  is inpainted by copying the pixels from  $\Psi_{\hat{q}}$  to the target patch  $\Psi_{\hat{p}}$ . Besides, the boundary of the target region is updated.

The above steps are done iteratively until the target region is fully inpainted.

### 3.4 Flow Preserving Distance Function

The main difficulty of generalizing classical exemplar-based inpainting to videos is maintaining temporal consistency. Therefore, we modify the distance function (3) by Criminisi *et al.* [7]. The key idea of our approach is that scenes do not change vastly and changesets can be determined by optical flow. So we assume to already have a well inpainted frame and for further frames to inpaint we demand similarity to this reference frame. The connection between the reference frame and the current inpainting point is obtained via the inpainted optical flow  $\hat{f}$  of the original scene (compare Section 3.2).

The corresponding distance function reads as follows:

$$\hat{d}(\Psi_{\hat{p}}, \Psi_q) := d(\Psi_{\hat{p}}, \Psi_q) + \frac{\beta}{|\Psi_{\hat{p}} \cap \Phi|} d(\Psi_{\hat{f}^{-1}(\hat{p})}, \Psi_q). \quad (4)$$

The first term ensures local consistency, as proposed by Criminisi *et al.* The second one enforces similarity to a previous inpainted frame and hence temporal consistency.  $\Psi_{\hat{f}^{-1}(\hat{p})}$ , using inverse optical flow, points back to the already inpainted image and ensures temporal consistency.

This distance function enables us to reduce complexity of the patch match since we do not have to choose a set of 3D patches. Our algorithm can greedily choose the best patch for the current hole to fill yet can select from all frames to exploit time redundancy. An illustration is shown in Figure 1.

### 3.5 Overview of the Algorithm

**Interactive Mask Definition.** Let  $\mathcal{I}[k]$  denote the  $k$ 'th video frame. The user is asked to roughly scribble (see Section 2) the desired regions in the first frame  $\mathcal{I}[0]$ . These scribbles are propagated via optical flow (Figure 2 b) throughout the video. Depending on the user scribbles a two-region segmentation in object  $\Omega$  (green) and search space  $\Phi$  (yellow) or a three-region segmentation with additional region  $\Phi_r$  (red) for neglecting parts is computed:  $\mathcal{I} = \Omega \dot{\cup} \Phi (\dot{\cup} \Phi_r)$ .

This processing gives an accurate mask in an easy and quick manner. State-of-the-art methods do not tackle how to obtain an accurate mask definition.

**Video Completion by Flow and Color Inpainting.** In the proposed image inpainting algorithm one can choose the number of frames to be inpainted at the same time. This allows to exploit redundancy in the video sequence.

Using the inpainted optical flow  $\hat{f}$  of the original video sequence we fill the target region  $\Omega$  step by step according to Criminisi *et al.* using our new distance function (4). Our distance function ensures, that the chosen patch is both locally consistent and similar to its origin in a previous inpainted frame. This leads to a temporal consistent inpainted video sequence without any flickering.

## 4 Experiments and Results

In the following we will show results on various datasets and compare our results to state-of-the-art approaches for video inpainting. The evaluations show that we can handle different object and camera motions.

Depending on the video size we choose a patchsize between  $8 \times 8$  and  $12 \times 12$  and inpaint 3 to 8 frames at the same time to exploit time redundancy. We choose  $\beta$  around 1.1 to weight local and temporal consistency.

In Figure 5 we compare two adjacent frames with and without our proposed consistency term. Without the flow consistency term the results have large deviations from one frame to the next one. In the final video such deviations are observed as disruptive flickering. In contrast, the video sequence inpainted with our proposed term shows smooth transitions between the frames. We obtain great results for complex scenes with detailed structures and different types of camera motions at substantially reduced runtime. Figures 4 and 6 compare our results to the results of Patwardhan *et al.* [18] and Newson *et al.* [15]. Table 1 compares the runtime of our method with the state-of-the-art methods [11, 14, 15, 18, 26].

Table 1: **Runtimes.** Although our approach includes an interactive mask-definition we outperform state-of-the-art methods up to a factor of five.

	Beach Umbrella 	Jumping Girl 	Stairs 	Young Jaws 
	264 × 68 × 98	300 × 100 × 239	320 × 240 × 40	1280 × 720 × 82
Wexler <i>et al.</i> [26]	1h	-	-	-
Patwardhan <i>et al.</i> [18]	≈ 30 min	≈ 1h 15min	≈ 15 min	-
Granados <i>et al.</i> [11]	11 hours	-	-	-
Newson <i>et al.</i> [14]	21 min	62 min	-	-
Newson <i>et al.</i> [15]	24 min	40 min	-	5h 48 min
<b>proposed approach</b>	<b>4.6 min</b>	<b>8 min</b>	<b>5 min 20 sec</b>	<b>3h 20min</b>

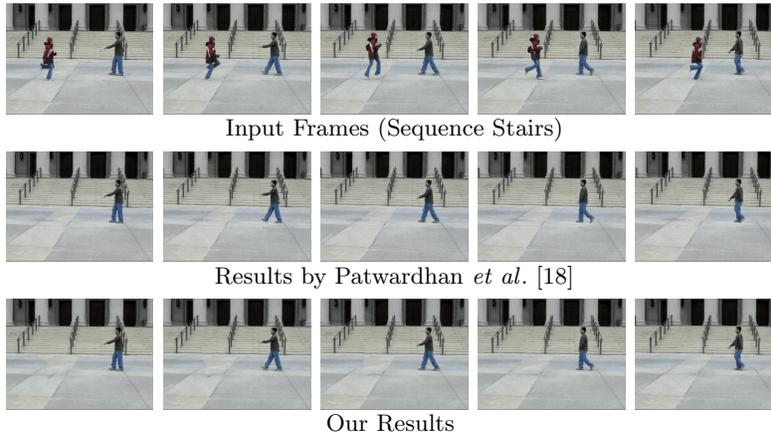


Fig. 4: Comparison to Motion SSD dataset with slight camera movement.

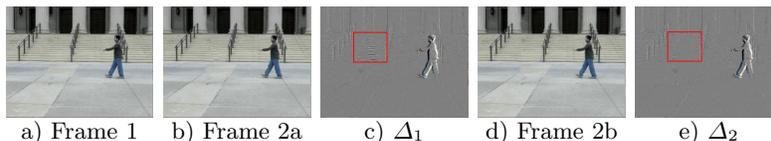


Fig. 5: **Transition comparison.**  $\Delta_1$  shows the transition between a) and b). The transition is computed without regularization and shows strong video flickering. In contrast, the transition  $\Delta_2$  with our approach between a) and d) is smooth and does not show disruptive flickering.

#### 4.1 Implementation & Runtime

Runtime is a big challenge to all video inpainting algorithms. Especially on high resolution videos a large amount of data has to be processed. Our parallel implementation takes around 2 to 150 seconds per frame, depending on the resolution of the input video on a NVIDIA GeForce GTX 560 Ti. This outruns state-of-the-art algorithms, requiring much more computing power (like Granados *et al.* [11] on a mainframe with 64 CPUs) and runtime (compare Table 1).

## 5 Conclusion

We propose an interactive video completion method which integrates two innovations: Firstly, we replace the tedious hand-labeling of inpainting regions in all video frames by a semi-automatic procedure which consists of a flow-based propagation of user scribbles from the first to subsequent frames followed by an automatic foreground-background segmentation. Secondly, we propose a novel solution for assuring temporal consistency of the inpainting. Rather than performing a computationally intense sampling of space-time patches, we perform an optical flow inpainting followed by a flow-constrained image inpainting. An efficient GPU implementation provides a semi-automatic video inpainting method which requires substantially less user input and provides competitive video inpainting results which is around five times faster than competing methods.



Fig. 6: **Our results compared to state-of-the-art methods.** Evaluations on the sequences *Fountains*, *Les Loulous* and *Young Jaws* by [15] show that we obtain the same precision of results, whereas our runtime is much faster. Furthermore, we are not restricted to a static mask and can easily remove different objects - see our results of the *Young Jaws* sequence.

## References

1. Ashikhmin, M.: Synthesizing natural textures. In: Proceedings of the 2001 symposium on Interactive 3D graphics. pp. 217–226. ACM (2001)
2. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.* 28(3) (2009)
3. Bertalmio, M., Bertozzi, A.L., Sapiro, G.: Navier-stokes, fluid dynamics, and image and video inpainting. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 355–362 (2001)
4. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques. pp. 417–424. ACM Press/Addison-Wesley Publishing Co. (2000)
5. Brox, T., Bruhn, A., Papenbergh, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Europ. Conf. on Computer Vision, pp. 25–36. Springer (2004)
6. Criminisi, A., Perez, P., Toyama, K.: Object removal by exemplar-based inpainting. In: Int. Conf. on Computer Vision and Pattern Recognition. vol. 2, pp. 721–728 (June 2003)
7. Criminisi, A., Perez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. *Image Processing, IEEE Transactions on* 13(9), 1200–1212 (Sept 2004)
8. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. pp. 341–346. ACM (2001)
9. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999. vol. 2, pp. 1033–1038 (1999)
10. Granados, M., Kim, K.I., Tompkin, J., Kautz, J., Theobalt, C.: Background inpainting for videos with dynamic objects and a free-moving camera. In: Proceedings of the 12th European Conference on Computer Vision. pp. 682–695. ECCV'12, Springer-Verlag, Berlin, Heidelberg (2012)
11. Granados, M., Tompkin, J., Kim, K.I., Grau, O., Kautz, J., Theobalt, C.: How not to be seen - object removal from videos of crowded scenes. *Comput. Graph. Forum* 31(2), 219–228 (May 2012)
12. Masnou, S.: Disocclusion: a variational approach using level lines. *IEEE Trans. on Image Processing* 11(2), 68–76 (2002)
13. Masnou, S., Morel, J.M.: Level lines based disocclusion. In: Int. Conf. on Image Processing. vol. 3, pp. 259–263 (1998)
14. Newson, A., Almansa, A., Fradet, M., Gousseau, Y., Pérez, P.: Towards fast, generic video inpainting. In: Proceedings of the 10th European Conference on Visual Media Production. pp. 1–8. CVMP '13, ACM, New York, NY, USA (2013)
15. Newson, A., Almansa, A., Fradet, M., Gousseau, Y., Pérez, P.: Video inpainting of complex scenes (January 2014), <http://hal.archives-ouvertes.fr/hal-00937795>
16. Nieuwenhuis, C., Cremers, D.: Spatially varying color distributions for interactive multi-label segmentation. *IEEE Trans. on Patt. Anal. and Mach. Intell.* 35(5), 1234–1247 (2013)
17. Patwardhan, K., Sapiro, G., Bertalmio, M.: Video inpainting of occluding and occluded objects. In: IEEE International Conference on Image Processing. vol. 2, pp. 69–72 (Sept 2005)

18. Patwardhan, K.A., Sapiro, G., Bertalmo, M.: Video inpainting under constrained camera motion. *IEEE Transactions on Image Processing* 16(2), 545–553 (2007)
19. Sethian, J.A.: A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences* 93(4), 1591–1595 (1996)
20. Shih, T., Tang, N., Hwang, J.N.: Exemplar-based video inpainting without ghost shadow artifacts by maintaining temporal continuity. *IEEE Transactions on Circuits and Systems for Video Technology* 19(3), 347–360 (March 2009)
21. Telea, A.: An image inpainting technique based on the fast marching method. *Journal of graphics tools* 9(1), 23–34 (2004)
22. Tsitsiklis, J.N.: Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control* 40(9), 1528–1538 (1995)
23. Wang, J., Lu, K., Pan, D., He, N., kun Bao, B.: Robust object removal with an exemplar-based image inpainting approach. *Neurocomputing* 123, 150–155 (2014), contains Special issue articles: *Advances in Pattern Recognition Applications and Methods*
24. Werlberger, M.: *Convex Approaches for High Performance Video Processing*. Ph.D. thesis, Institute for Computer Graphics and Vision, Graz University of Technology, Graz, Austria (June 2012)
25. Wexler, Y., Shechtman, E., Irani, M.: Space-time video completion. In: *Int. Conf. on Computer Vision and Pattern Recognition*. vol. 1, pp. 120–127 (June 2004)
26. Wexler, Y., Shechtman, E., Irani, M.: Space-time completion of video. *IEEE Trans. on Patt. Anal. and Mach. Intell.* 29(3), 463–476 (March 2007)