MOTION-CONSISTENT VIDEO INPAINTING

Thuc Trinh Le[*‡], *Andrés Almansa*[†], *Yann Gousseau*[*], *Simon Masnou*[‡]

[*] LTCI, Télécom ParisTech, Université Paris-Saclay, 75013 Paris, France
[†] MAP5, CNRS & Université Paris Descartes, 75006 Paris, France
[‡] Univ Lyon, Univ Claude Bernard Lyon 1 & CNRS, Institut Camille Jordan, 69622 Villeurbanne, France

ABSTRACT

We propose a fast and automatic inpainting technique for high-definition videos which works under many challenging conditions such as a moving camera, a dynamic background or a long lasting occlusion. Built upon the previous work by Newson *et al.* [1] which optimizes a global patch-based function, our method makes a significant improvement, especially in motion preservation, by incorporating the optical flow in several stages of the algorithm. Moreover, code parallelization and a modification in the process of patches pairwise matching yield a significant reduction of computation time. Experimental results on both classical and challenging datasets show that our algorithm outperforms other state-of-the-art approaches.

Index Terms— Video inpainting, video restoration, patches, optical flow, video editing.

1. INTRODUCTION AND PRIOR WORKS

Video inpainting aims to fill in a missing region (an occlusion) in a video using the rest of that video to produce a "plausible" result. Video inpainting has numerous applications, ranging from restoring error concealment [2] or removing undesired objects [3] to restoring scratches or damages in vintage films [4]. While image inpainting has attracted much attention over the last two decades [5], video inpainting remains an underdeveloped and challenging field due to the difficulty of dealing with complex motions, the high sensitivity of our visual system to temporal inconsistencies, and the computational complexity. Most recent methods found in the literature addresses these issues using either object-based or patch-based approaches.

Under object-based approach, a preprocessing step is required to split the video into background and foreground objects, followed by an independent reconstruction of each part and the merging of the results at the end of the algorithm. Examples which fall under this category are the layered mosaic technique of Jia *et al.* [6], the homography-based algorithm using graph cut by Granados *et al.* [3] and the posture mapping scheme by Ling *et al.* [7]. Typically, object-based methods can provide reasonable results for reconstructing a specific object, e.g. a human. However, these techniques require some strict conditions such as periodic motion [6] or user assistance [3]. Furthermore, as the foreground and background completion procedures are performed independently, blending one with the other may cause artifacts.

In patch-based methods, patches from source regions are used to fill in the occlusion in a greedy or global fashion. Greedy methods inpaint incrementally the occlusion pixel by pixel, the reconstruction order is defined by a priority term. For example, Patwardhan *et al.* [8] extend to 3D space the well-known image inpainting technique in [9], whereas Daisy *et al.* [10] employ a tensor voting term to calculate the priority and focus on a geometry-guided blending technique to reduce space-time artifacts. In general, these greedy methods are very sensitive and, most importantly, do not have the capacity to reconstruct motion over a large occlusion and in a coherent way.

To ensure the global consistency, a global approach is needed. A natural strategy is to minimize a global patch-based function. In their seminal contribution [11], Wexler *et al.* pioneer the optimization of a global energy based on 3D spatio-temporal patches to preserve temporal coherency. Subsequent contributions propose various improvements. In [1], a significant step forward is made by using the 3D PatchMatch to strengthen the coherence and speed up the patch matching. In [12], Granados *et al.* focus on identifying a shift map in 3D space using graph-cut. Recently, Huang *et al.* [13] modify Wexler's energy by adding an optical flow term to enforce the temporal coherency. These methods not only provide very impressive results but they are also compatible with many scenarios. However, they have several drawbacks such as huge computation time [12], inability to deal with motion within large occlusion [1] or unpleasant artifacts [13].

In order to fix these issues, we propose a fast video inpainting technique which builds upon [1] with three major improvements. The first and most significant advancement is the heavy use of optical flow in several stages of the algorithm. Optical flow has already been used in some previous works in video inpainting. For example, Strobel *et al.* [14] inpaint the optical flow field first and use the result to guide

This work was supported by the French Research Agency under Grant ANR-14-CE27-001 (MIRIAM)

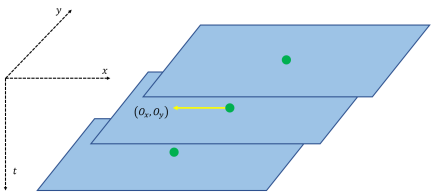


Fig. 1. Our patches are parallelepipeds with $x - y$ skew controlled by the optical flow field.

the nearest neighbor search. Huang *et al.* [13] rely on optical flow path to find the best patches and reconstruct pixel intensities. As their method uses only 2D patches, optical flow is the only part contributing to the preservation of temporal coherency, therefore the inaccurate synthesis of optical flow will generate artifacts. Our method, on the other hand, maintains the temporal consistency by using both 3D spatio-temporal patches and an optical flow term. This term is incorporated in several stages: it is inserted in the patch distance, controls the patch shape, supports the nearest neighbor search, and serves as a guide in coarse initialization. All these stages enable us to ensure the temporal coherency and the reconstruction of objects with complex motions occluded for long time periods. The second contribution is a significant reduction in computation cost achieved by parallelizing the algorithm and modifying the patch matching process. The final improvement is the integration of a confidence map and a separation map in the pixel reconstruction step to reduce artifacts. We evaluate our method under various conditions and compare it with some other state-of-the-art approaches using their public datasets. The results can be found in a dedicated website.

2. PROPOSED METHOD

2.1. Overview

Our method is based on a non-local patch-based energy, in the spirit of [11, 1]. The energy is minimized thanks to an iterative procedure embedded in a coarse-to-fine pyramidal scheme. Our algorithm involves two core steps: the computation of a nearest neighbor field in the occlusion which approximates the patches best pairwise matches, and a reconstruction step using this field to determine the values of all occluded pixels.

Within this framework, it is essential to address many problems such as the coherent preservation of motion, the searching strategy to find the appropriate patches, the computational complexity and the border artifacts. These problems are addressed in our method by modifying patch shapes and patch metric, integrating a novel optical flow-driven initialization scheme, parallelizing the algorithm, speeding up the nearest neighbor search and enhancing the reconstruction step. These techniques will be presented in the following sections.

2.2. Energy

To handle the instability caused by camera movements, a stabilization pre-processing is performed using the method in [15]. After stabilization, we minimize a Wexler-like energy $E(u, \phi)$ to find the inpainted sequence u and the corresponding patch correspondence (or shift map) ϕ. Denoting $W_p(u)$ the patch centered at a pixel p in the given occlusion domain $\mathcal{H}$, the shift $\phi(p)$ at p is defined as the spatial offset $q-p$ where q is a minimizer in $\mathcal{H}^c$ of the patch distance $d(W_p^u, W_q^u)$ (see below for the definition of d). The energy E associated with an image u (known outside a given occlusion domain $\mathcal{H}$) and a shift map ϕ is defined as

$$E\left(u,\phi\right) = \sum_{p\in\mathcal{H}} d^2\left(W_p^u, W_{p+\phi(p)}^u\right)$$

Minimizing this energy ensures that each patch W_p^u centered around a pixel p in the occlusion domain $\mathcal{H}$ is as close as possible to its nearest neighbor $W_{p+\phi(p)}^u$ outside the occlusion (in the sense that $p + \phi(p) \notin \mathcal{H}$). We use a metric d between patches defined by:

$$d^2\left(W_p^u, W_q^u\right) = \frac{1}{|N_p|}\sum_{\substack{r\in N_p \\ r-p+q\notin\mathcal{H}}} [\alpha\left(\|u(r) - u(r-p+q)\|_2^2\right) +$$

$$\beta\left(\|T(r) - T(r-p+q)\|_2^2\right) + \gamma\left(\|O(r) - O(r-p+q)\|_2^2\right)].$$

In this expression, N_p indicates a spatio-temporal neighborhood of p. It is a parallelepiped whose shape is controlled by the optical flow vector as indicated in Figure 1. This adaptive shape, different from a classical rectangle cuboid, enables us to reduce the number of patches which contain both background and foreground data. Following [1, 16], our distance incorporates texture features $T = (|\frac{\partial u}{\partial x}|, |\frac{\partial u}{\partial y}|)$. In addition, to enhance temporal coherency, we use motion features $O = (|O_x|, |O_y|)$, which is composed of the modulus of the optical flow vector coordinates. Values of weights α, β, γ must be set according to the data (automatic setting is the purpose of ongoing work).

Our energy E is high dimensional and highly non-convex, but as observed in [17] a good local minimum can be obtained by alternate minimization *w.r.t* u and ϕ, coupled with a good initialization and a coarse-to-fine multiscale scheme. Texture and motion features in the similarity metric are key to guiding the algorithm towards a good local minimum from the coarsest scale. The general structure of our algorithm is as follows:

- Build multiscale pyramids for color u, occlusion domain $\mathcal{H}$, texture features T and motion features O.
- Initialization at coarsest scale (see section 2.3).
- From coarsest to finest scale do:
  - Iterate until convergence:
    - $\min_\phi$ (nearest neighbor search, section 2.4).
    - $\min_u$ (pixels reconstruction, section 2.5).
    - features reconstruction.
  - If not finest scale: Upsample ϕ, u, and features.

2.3. Coarse initialization

Due to the non-convexity of the functionals which are typically used in global patch-based methods, having a reliable initialization is crucial for the local minimization. Nevertheless, this step is often left unspecified in the literature, with the exception of [1] where a greedy inpainting technique using onion peel priority is proposed at coarsest scale. Such method can produce a good initialization for small occlusions. However, it tends to wipe out moving objects in long lasting occlusions. To solve this issue, we propose to use the optical flow in the priority term, which somehow extends to space-time the 2D inpainting approach of Criminisi *et al.* [9]. More precisely, the priority term at pixel i is defined as $Pr_i = C_i.D_i$, where D_i is the average of the optical flow magnitude in the patch centered at i, and $C_i \propto \exp\{-d^2(i, \mathcal{H}_{\text{coarse}})\}$ measures how close the pixel i is to the border of the original occlusion $\mathcal{H}_{\text{coarse}}$ at coarsest scale).

The coarse initialization is then obtained as follows. Starting from $\mathcal{H}' = \mathcal{H}_{\text{coarse}}$, we repeat the following procedure until there remain only "background" pixels, defined as all pixels i such that $D_i \leq S$, where S is an adaptive threshold obtained by Otsu's method.

- Let $\mathcal{B}' = \mathcal{H}' \setminus (\mathcal{H}' \ominus B(0,1))$, i.e. $\mathcal{B}'$ is the one-pixel wide outer boundary of $\mathcal{H}'$. Calculate Pr_i for $i \in \mathcal{B}'$.
- Select patch P_i with highest priority term Pr_i, and define the region to inpaint $R_i = P_i \cap \mathcal{B}'$.
- Inpaint R_i and get new occlusion region $\mathcal{H}' \rightarrow \mathcal{H}' \setminus R_i$.

Thereafter, the rest of the occlusion (i.e. background pixels) is inpainted following onion peel order.

2.4. Nearest neighbor patch search

Since its introduction by Barnes *et al.* [18], PatchMatch has become a classical tool for approximate nearest neighbor search in patch spaces, especially in the context of image and video inpainting, not only for computational speed but also for spatial consistency. The core part of the algorithm includes a propagation step to spread out good matches and a random search step to jump out of the local optima. These two steps are repeated in several iterations. In our video inpainting context, the spatio-temporal extension of PatchMatch by Newson *et al.* [1] is adopted with two important modifications to improve its efficiency:

- The first modification is a speedup of PatchMatch by parallelization, following the jump flooding technique of Barnes *et al.* [18]. To save even more computational cost, we use a sparse grid during the random search step. For PatchMatch searching in video, it is not necessary to perform the random search step for every occluded pixel; instead, we can apply this step only for pixels on a sparse grid, without losing the efficiency. The final improvement is to use foreground/background patch clustering to reduce the search space. Combining these factors enables our PatchMatch algorithm to run 5-7 times faster than the traditional one with the same accuracy.
- The second modification is to guide the propagation step using the optical flow. From the assumption that adjacent patches are more likely to have similar nearest neighbor offsets, PatchMatch achieves a good preservation of the spatial coherency. For temporal coherency, this assumption is only valid if the background is static or in periodic motion. Otherwise, it may not hold true. To enforce the temporal consistency, instead of propagating offsets to a fixed temporal neighbor, we propagate it following the optical flow direction. To be more formal, in the propagation step, the temporal neighbor for the patch centered at pixel (x, y, t), $P_{(x,y,t)}$, is $P_{(x+O_x, y+O_y, t+1)}$ rather than $P_{(x,y,t+1)}$ where O_x and O_y are the optical flow components in the x and y directions.

2.5. Pixel reconstruction

In this step, all pixels in the occlusion $\mathcal{H}$ are reconstructed using the following weighted average:

$$u(p) = \frac{\sum_{q \in N_p} s_q^p u\left(p + \phi(q)\right)}{\sum_{q \in N_p} s_q^p},$$

where the weight s_q^p is defined as:

$$s_q^p = \exp\left(-\frac{d^2\left(W_q^u, W_{q+\phi(q)}^u\right)}{2\sigma_p^2}\right) \psi_q \varphi_q^p.$$

In this expression, the first term is the original weight of [11] based on patches similarity. We combine it with two other factors ψ_q and φ_q^p. The first one is a confidence map inspired by Fedorov *et al.* [19], and given by

$$\psi_q = \begin{cases} (1 - C_0) \exp\left(-\frac{d(q, \mathcal{H}^c)}{\sigma^2}\right) + C_0 & \text{if } q \in \mathcal{H} \\ 1 & \text{otherwise} \end{cases}$$

where $d(q, \mathcal{H}^c)$ is the distance from pixel q to the occlusion border, and C_0, σ^2 are tuning parameters. This map is used to guide the information from the border towards the center and enables us to eliminate some border artifacts. The second term φ_q^p relies on a distinction between foreground and background pixels, obtained by thresholding the modulus of the optical flow. It is set to 1 if p and q are of same type (background or foreground), otherwise it is set to 0. Therefore, when we reconstruct background pixels, we use only background patches and similarly for foreground pixels. This is a simple way to avoid the common but undesirable effect of blending between background and foreground in the final result.

3. EXPERIMENTAL RESULTS

Our algorithm is implemented in Matlab with the core parts (nearest neighbor serch and pixel reconstruction) in C++. For the optical flow computation, Liu's method [20] is used.

Our method is evaluated under a wide variety of conditions, including moving objects occluded by a fixed or moving domain, static or moving camera, dynamic background, large occlusions, etc. To prove the effectivity of our method, we compare its performances with other state-of-the-art algorithms [1, 13] using their publicly available datasets. Results can be found at `http://perso.enst.fr/~gousseau/vid_inp_motion/`.

3.1. Comparison with Huang *et al.* [13]

In this experiment, we remove undesired objects in videos recorded with hand-held camera. The dataset used is the same as in Huang *et al.* [13], obtained from a recent benchmark dataset in object segmentation [21]. It constitutes a very challenging dataset due to the dynamic scenes, the complex camera movements, the motion blur effects and the large occlusions. The occlusion mask is constructed by dilating the ground truth using a 15x15 structuring element.

Figure 2 (a) shows some representative frames of the result. From that figure, we can see that, similar with Huang *et al.* [13], the spatial structure (e.g the letters in the panel) is well-preserved with our approach. Figure 2(b) shows the result as an x-t slice of the video. It can be seen that our method has the ability to preserve temporal consistency due to the combination of 3D spatio-temporal patches and dense flow field. This is also achieved in [13]; however, because only 2D patches are used in [13], the quality of the output temporal coherency strongly depends on the accuracy of the optical flow computation. Such accuracy cannot be guaranteed in several complex sequences such as *mallard*, *drift-chicante* or *breakdance*. Furthermore, the incorrect synthesis of optical flow may lead to several displeasing artifacts. This is reported in [13] with the sequence *loulous*; meanwhile, our method provides a very plausible result with that sequence.

Another advantgae of our method is its speed. While it takes Huang *et al.* [13] approximately 3 hours to complete one video in this dataset using 2D patches, our method takes around 50 minutes.

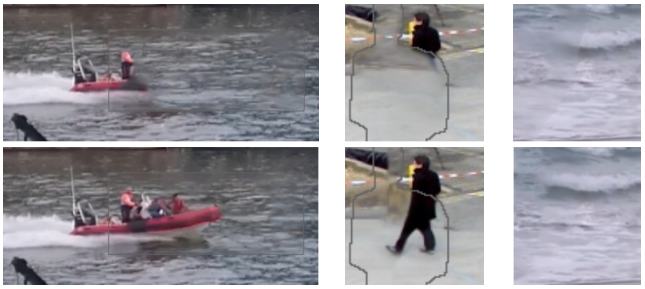
3.2. Comparison with Newson *et al.* [1]

This experiment evaluates our performance in the context of the reconstruction of moving objects. We consider several videos in which moving objects cross a fixed or a moving occlusion for a long period. Such objects can be partly or even completely occluded (sequence *jumping girl*) and the background can be either static or dynamic.

Representative results are illustrated in figure 3. It is clearly seen from this figure that our result outperforms that

(a) (b)

Fig. 2. (a) some representative result frames with sequences *tennis* (left) and *bmx-bump* (right). From top to bottom: our result with occlusion mask boundary in green, our result without occlusion mask, result of Huang *et al.* [13]. (b) x-t slice along the profile (green line) in the sequence *elephant*. From top to bottom: position of the slice, occlusion mask, our result, result of Huang *et al.* [13].



(a) (b) (c)

Fig. 3. Sample cropped frames of the results for some sequences: (a) *boat*, (b) *S2L1*, (c) *loulous*. Top: result of [1], bottom: our result.

of Newson *et al.* [1]. Figure 3 (a) and (b) show that Newson *et al.*'s method [1] cannot reconstruct the foreground object (e.g. the boat) within a long occlusion. Our method, on the other hand, has the ability to fully reconstruct the background and foreground in a consummate manner even though the object is completely occluded. Moreover, by integrating the confident map in the reconstruction step, our result has less artifacts in the border than the one in [1], as can be seen in figure 3 (c).

4. CONCLUSION

This paper presents a new video inpainting technique which shows great performance in terms of both output quality and computation time thanks to a thorough use of the optical flow, a modified patch-based energy which incorporates complex informations, a modified patch searching strategy using sparse grid and patch clustering, and finally a suitable code parallelization.

5. REFERENCES

[1] Alasdair Newson, Andrés Almansa, Matthieu Fradet, Yann Gousseau, and Patrick Pérez, "Video inpainting of complex scenes," *SIAM Journal on Imaging Sciences*, vol. 7, no. 4, pp. 1993–2019, 2014.

[2] Mounira Ebdelli, Olivier Le Meur, and Christine Guillemot, "Video inpainting with short-term windows: application to object removal and error concealment," *IEEE Transactions on Image Processing*, vol. 24, no. 10, pp. 3034–3047, 2015.

[3] Miguel Granados, Kwang In Kim, James Tompkin, Jan Kautz, and Christian Theobalt, "Background inpainting for videos with dynamic objects and a free-moving camera," in *European Conference on Computer Vision*. Springer, 2012, pp. 682–695.

[4] Nick C Tang, Chiou-Ting Hsu, Chih-Wen Su, Timothy K Shih, and Hong-Yuan Mark Liao, "Video inpainting on digitized vintage films via maintaining spatiotemporal continuity," *IEEE Transactions on Multimedia*, vol. 13, no. 4, pp. 602–614, 2011.

[5] Marcelo Bertalmío, Vicent Caselles, Simon Masnou, and Guillermo Sapiro, "Inpainting," in *Computer Vision: A Reference Guide*, pp. 401–416. Springer US, Boston, MA, 2014.

[6] Jiaya Jia, Yu-Wing Tai, Tai-Pang Wu, and Chi-Keung Tang, "Video repairing under variable illumination using cyclic motions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 832–839, 2006.

[7] Chih-Hung Ling, Chia-Wen Lin, Chih-Wen Su, Hong-Yuan Mark Liao, and Yong-Sheng Chen, "Video object inpainting using posture mapping," in *2009 16th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2009, pp. 2785–2788.

[8] Kedar A Patwardhan, Guillermo Sapiro, and Marcelo Bertalmio, "Video inpainting of occluding and occluded objects," in *IEEE International Conference on Image Processing 2005*. IEEE, 2005, vol. 2, pp. II–69.

[9] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on image processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[10] Maxime Daisy, Pierre Buyssens, David Tschumperlé, and Olivier Lézoray, "Exemplar-based video completion with geometry-guided space-time patch blending," in *SIGGRAPH Asia 2015 Technical Briefs*. ACM, 2015, p. 3.

[11] Yonatan Wexler, Eli Shechtman, and Michal Irani, "Space-time completion of video," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 29, no. 3, pp. 463–476, 2007.

[12] Miguel Granados, James Tompkin, K Kim, Oliver Grau, Jan Kautz, and Christian Theobalt, "How not to be seen—object removal from videos of crowded scenes," in *Computer Graphics Forum*. Wiley Online Library, 2012, vol. 31, pp. 219–228.

[13] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf, "Temporally coherent completion of dynamic video," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 196, 2016.

[14] Michael Strobel, Julia Diebold, and Daniel Cremers, "Flow and color inpainting for video completion," in *German Conference on Pattern Recognition*. Springer, 2014, pp. 293–304.

[15] Jean-Marc Odobez and Patrick Bouthemy, "Robust multiresolution estimation of parametric motion models," *Journal of visual communication and image representation*, vol. 6, no. 4, pp. 348–365, 1995.

[16] Yunqiang Liu and Vicent Caselles, "Exemplar-based image inpainting using multiscale graph cuts.," *IEEE Transactions on Image Processing*, vol. 22, no. 5, pp. 1699–711, may 2013.

[17] P. Arias, V. Caselles, and G. Facciolo, "Analysis of a Variational Framework for Exemplar-Based Image Inpainting," *Multiscale Modeling & Simulation*, vol. 10, no. 2, pp. 473–514, jan 2012.

[18] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan Goldman, "Patchmatch: a randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics-TOG*, vol. 28, no. 3, pp. 24, 2009.

[19] Vadim Fedorov, Gabriele Facciolo, and Pablo Arias, "Variational framework for non-local inpainting," *Image Processing On Line*, vol. 5, pp. 362–386, 2015.

[20] Ce Liu, *Beyond pixels: exploring new representations and applications for motion analysis*, Ph.D. thesis, Citeseer, 2009.

[21] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Computer Vision and Pattern Recognition*, 2016.