A PatchMatch-based Approach for Matte Propagation in Videos

Marcos H. Backes[1], **Manuel M. de Oliveira Neto**[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

Abstract. *This thesis presents a temporally-coherent matte-propagation method for videos based on PatchMatch and edge-aware filtering. Given an input video and trimaps for a few frames, including the first and last, our approach generates alpha mattes for all frames of the video sequence. We also present a user scribble-based interface for video matting that takes advantage of the efficiency of our method to interactively refine the matte results. We perform quantitative comparisons against the state-of-the-art sparse-input video matting techniques and show that our method produces significantly better results according to three different metrics. We also perform qualitative comparisons against the state-of-the-art dense-input video matting techniques and show that ours produces similar quality results while requiring less than 7% of their user input.*

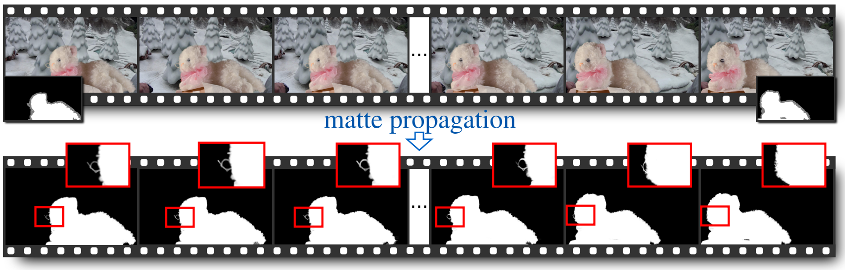

Figure 1. Given an input video sequence (top) and user-defined trimaps for the first and last frames, our method is able to efficiently compute and propagate temporally coherent alpha mattes (bottom) for the video.

1. Introduction

Natural image matting refers to the process of accurately extracting foreground objects from natural images. In such a context, the color I_p of any given pixel p is described as a linear combination of a foreground color F_p and a background color B_p, according to some opacity value α_p. Computing such *alpha mattings* is, therefore, an ill-posed problem. Thus, matting techniques require additional information, often presented in the form of *trimaps* or scribbles specifying three sets of pixels belonging, respectively, to foreground, to background, and to unknown regions.

Although image matting is a well studied problem and recent works can produce high-quality results [Levin et al. 2008, Gastal and Oliveira 2010, Xu et al. 2017], video matting still presents several challenges. As in most video applications, there are difficulties associated with fast motions, lighting changes, occlusion and disocclusion, and processing of large amounts of data. In addition, video-matting algorithms have two special requirements: they are expected to operate under sparse user input, and achieve temporal coherence. Video matting processes large amounts of data and most existing techniques use one trimap per frame. To reduce the user burden, some techniques generate the required trimaps [Wang et al. 2005], or use a sparse set of trimaps [Li et al. 2013, Zou et al. 2019]. Although these methods can reduce the amount of user-provided input, they are not fast enough for interactive use. The ability to interactively compute and refine mattes considerably reduces time and improves the quality of video matting results.

1.1. Contributions

We present **an efficient temporally-coherent matte-propagation method for videos**. *Our technique uses a sparse set of trimaps, requiring a relatively small amount of user input, and propagates the computed mattes to the entire video sequence*. Unlike previous approaches that can only handle a few frames at a time, ours processes an entire video sequence at once, naturally enforcing temporal coherence. We exploit the parallelism of modern GPUs and the use of linear-time edge-aware filters [Gastal and Oliveira 2011] to process high-resolution videos (*e.g.*, full HD or higher) in just a few milliseconds per frame, allowing for *interactive editing and propagation of the computed mattes on-the-fly*. Such interactivity improves productivity and the quality of the extracted mattes. Figure 1 illustrates the use of our technique to extract and propagate mattes for an entire video sequence. A paper describing our method and its associated contributions, entitled *A PatchMatch-based Approach for Matte Propagation in Videos* [Backes and Oliveira 2019] was published in the **Computer Graphics Forum** journal, one of the most prestigious in the field. A copy of the paper can be retrieved by clicking here. We encourage the reader to access the on-line Suplemental Materials, which provide lots of video matting examples as well as quantitative and qualitative comparisons against state-of-the-art techniques.

2. Related Works

Most video-matting solutions handle the individual video frames independently, requiring a trimap per frame, and often compromising temporal coherence. Even when high-quality image-matting techniques are used, the resulting videos tend to exhibit temporal jittering and inconsistencies across frames [Erofeev et al. 2015]. Although some recent video-matting techniques [Karacan et al. 2017, Cao et al. 2019] are able to find interframe pixel relationships to produce temporally-coherent mattes, such techniques typically only handle up to five frames at a time.

Sparse-input video matting techniques either use a frame-by-frame propagation strategy [Bai et al. 2009, Li et al. 2013, Zou et al. 2019], or process the video as a whole [Wang et al. 2005]. The first group can only propagate the matte one way. Thus, whenever an error occurs it is propagated forward, resulting in temporal inconsistencies. Although, theoretically, by using the entire sequence the second group should be able to overcome this issue, in practice the presented solutions are temporally unstable [Wang et al. 2005] or do not scale to current video resolutions.

3. Matte Propagation

Our matte propagation technique for videos has three major steps: (i) Computing both forward and backward optical-flows along the temporal dimension with PatchMatch (Section 3.1); (ii) Using the computed optical-flows to propagate alpha values, as well as foreground and background colors from keyframes to unconstrained ones using a temporal version of the domain transform's recursive filter (Section 3.2); and (iii) Refining the computed alpha values to obtain locally-smooth mattes (Section 3.3).

3.1. Computing Forward and Backward Optical-Flow

We use PatchMatch [Barnes et al. 2009] to find correspondences between pairs of pixels across adjacent frames. Given a pair of RGB images A and B, for every overlapping square patch of side p in A, PatchMatch looks for its nearest neighbor in B under a distance metric d (originally L_2 distance). Given I^t and I^{t+1}, respectively the current and next video frames, we set $A = I^t$ and $B = I^{t+1}$ to compute the forward optical-flow, and $A = I^{t+1}$ and $B = I^t$ to obtain the backward optical-flow. The use of the edge-preserving matching cost function described by Bao et al. [Bao et al. 2014] produces more accurate matching around object borders when compared to traditional optical-flow approaches. According to our experience, it produces better results for our application than all tested alternatives. One should note, however, that the optical-flow obtained with PatchMatch has no subpixel accuracy, and that the use of more precise flow around the edges of the foreground objects should lead to even more accurate matte propagation.

3.2. Propagation

The alpha values, foreground and background colors for the keyframes (frames with provided trimaps) are obtained with the use of some matting technique (*e.g.*, [Gastal and Oliveira 2010], [Levin et al. 2008], [Xu et al. 2017]). This makes matte propagation orthogonal to the choice of the matte computation algorithm applied to the keyframes. Since every matting technique has its own strengths and weaknesses, the user can select the one that works best for the type of video at hand. The forward and backward optical-flows computed with PatchMatch guide the propagation of alpha values, and of foreground and background colors throughout the unconstrained frames between pairs of keyframes. We use the domain transform recursive filter to propagate these values in linear time with respect to the number of pixels in the video [Gastal and Oliveira 2011,Lang et al. 2012].

3.3. Refining the Propagated Matte

The propagated alpha values might be noisy. To refine the matte, we use the scheme presented by Gastal and Oliveira [Gastal and Oliveira 2010] for optimizing the alpha channel based on the matting Laplacian L [Levin et al. 2008]. Once a refined alpha matte has been obtained for each frame, we also refine the corresponding foreground and background colors, by minimizing the chromatic error for each pixel [Levin et al. 2008].

3.4. Discarding False Foreground Components

Occasionally, optical-flow mismatches may lead to incorrect classification of background pixels as foreground ones. To minimize the occurrence of such events, users can specify the maximum number of foreground components present in a sequence. In this case, for each frame we use a flood filling strategy to detect connected pixel regions with $\alpha > 0$ and keep at most a user-specified number of the largest ones. The remaining are treated as background pixels. Once false foreground components are detected, our technique propagates the corrected mates both forwards and backwards.

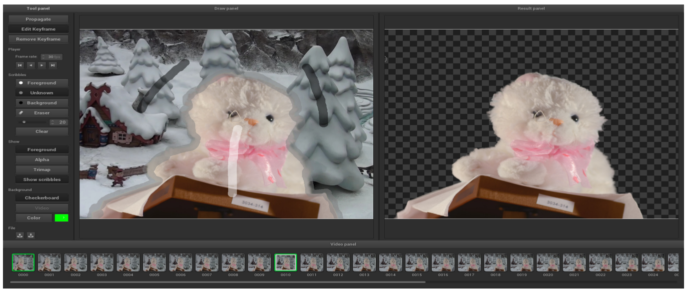

Figure 2. Our interactive video matting system interface. Scribbles on the keyframes indicate the foreground (white), background (black), and unknown (gray) regions. The extracted foreground object is instantly updated on the right window. Our system then propagates the extracted mattes for the unconstrained frames (frames without trimaps). Users can inspect the matte of any frame and interactively refine it with additional scribbles.

Table 1. Mean error metrics computed for the three video sequences using nine keyframes. AE - Adobe After Effects Rotobrush Tool, MAKNN - Motion-aware KNN matting, SLR - Sparse Low-Rank matting, OURS+CF - Ours with Closed-form Matting, and OURS+SM - Ours with Shared Matting.

Video		Alex			castle			Dmitriy			**Total**	
Metric	SSDA	dtSSD	MESSDdt	SSDA	dtSSD	MESSDdt	SSDA	dtSSD	MESSDdt	SSDA	dtSSD	MESSDdt
AE	147.83	90.76	16,076.97	214.65	85.84	15,692.79	144.66	104.97	11,853.67	178.71	91.53	14,732.34
MAKNN	121.80	85.49	12,419.27	150.93	84.90	12,540.72	140.65	102.90	19,001.01	140.12	89.20	14,090.91
SLR	147.66	93.18	11,965.11	200.01	92.76	19,274.85	258.17	154.34	38,768.78	200.46	108.08	22,279.33
OURS+CF	**33.18**	**38.58**	**1,166.11**	125.83	74.81	9,206.04	**43.59**	**55.51**	2,851.13	80.64	**60.28**	5,492.45
OURS+SM	41.44	48.48	1,705.46	**111.91**	**70.61**	**7,911.91**	45.51	57.38	**2,769.59**	**76.48**	61.24	**4,980.80**

4. Interactive Video Matting

We implemented an interactive video matting and editing interface using CUDA/C++, as illustrated by Figure 2. Please refer to the video illustrating the use of our system, in the supplementary material [1]. Considering a 1080p video and an NVIDIA GTX 1070 graphics card, the average running time for each step of our algorithm is 1.24 seconds per frame for computing the forward and backward optical flows using PatchMatch, 67 milliseconds per frame for matte propagation, and 720 milliseconds per frame for matte and color refinement. Note that the optical flow is computed at the beginning of an interactive session, and can be precomputed. Since the user only has to wait for the recursive filter to obtain some visual feedback, our technique provides instant feedback, as opposed to other state-of-art sparse-input video matting methods [Li et al. 2013, Zou et al. 2019].

5. Results

To evaluate our method, we perform both quantitative and qualitative evaluations against the state-of-the-art video matting techniques, on various types of videos. More specifically, we perform quantitative evaluations against two techniques that, like ours, do not require the specification of one trimap per frame [Li et al. 2013, Zou et al. 2019], as well as against Adobe After Effects Rotobrush Tool (AE).

Table 1 summarizes the results of the quantitative evaluation. It shows the average per-frame error computed considering three video-matting error metrics (SSDA, dtSSD, and MESSDdt) [Erofeev et al. 2015] for each video sequence, using nine keyframes. The

[1] The supplementary material is available here.

two tested variants of our technique (Ours+SM) and (Ours+CF) performed significantly better than MAKNN, SLR, and AE in the three metrics for all tested videos. For the videos Alex and Dmitriy, our SSDA and MESSDdt results are one order of magnitude better than the other approaches. The last column (Total) shows the average per-frame error considering all frames in the three videos. Overall, the results of our technique were 45% more accurate (SSDA), 31% more temporally coherent (dtSSD), and 64% more temporally coherent considering motion estimation (MESSDdt).

We also perform qualitative comparisons against After Effects plus top four ranked techniques by the video matting benchmark [Erofeev et al. 2015]: *Deep Matting* (DM) [Xu et al. 2017], *Self-Adaptive Matting* (SAM) [Cao et al. 2019], *Learning Based Matting* (LB) [Zheng and Kambhamettu 2009], and *Sparse Sampling Matting* (SpSM) [Karacan et al. 2017]. These techniques require one trimap per frame, and our method produces similar results using less than 7% of their trimaps (Figure 3).

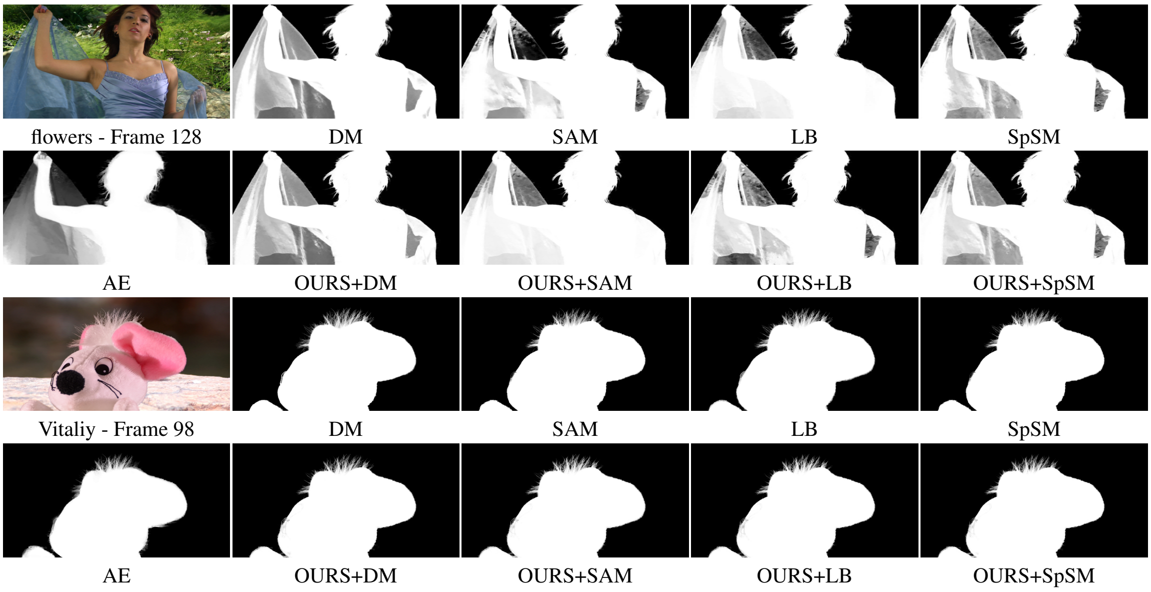

Figure 3. Comparison of results produced by techniques that require one trimap per frame against results produced by our method. AE - *After Effects* [Bai et al. 2009], DM - *Deep Matting* [Xu et al. 2017], SAM - *Self-Adaptive Matting* [Cao et al. 2019], LB - *Learning Based Matting* [Zheng and Kambhamettu 2009] and SpSM - *Sparse Sampling Matting* [Karacan et al. 2017]. OURS+DM, OURS+SAM, OURS+LB and OURS+SpSM stand for our method initialized by these respective matting methods every 15 frames.

6. Conclusion

We presented an efficient temporally-coherent matte-propagation method for videos. Our technique uses a sparse set of trimaps, requiring a relatively small amount of user input. Our solution performs both forward and backward matte propagation, lending to better temporal coherence. It is also orthogonal to the choice of alpha matte technique applied to the keyframes, allowing us to select the one that works best for the type of video at hand. We demonstrated the effectiveness of our technique by performing quantitative and qualitative evaluations against the state-of-the-art methods for video matting. Compared to approaches that only require sparse-input, ours performs significantly better with respect to three error metrics. When compared to techniques that require one trimap per frame, ours produces similar-quality results with less than 7% of user input. Given its computational efficiency, our technique provides instant feedback, allowing the development of interactive video matting systems for accurate matte extraction and compositing.

Acknowledgments

This work was sponsored by CNPq-Brazil (fellowships and grants 131965/2018-2, 312975/2018-0, 423673/2016-5).

References

Backes, M. H. and Oliveira, M. M. (2019). A patchmatch-based approach for matte propagation in videos. *Computer Graphics Forum*, 38(7):651–662.

Bai, X., Wang, J., Simons, D., and Sapiro, G. (2009). Video SnapCut: Robust Video Object Cutout Using Localized Classifiers. *ACM TOG*, pages 70:1–70:11.

Bao, L., Yang, Q., and Jin, H. (2014). Fast Edge-Preserving PatchMatch for Large Displacement Optical Flow. *IEEE Transactions on Image Processing*, pages 4996–5006.

Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B. (2009). PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. *ACM TOG*, pages 24:1–24:11.

Cao, G., Li, J., Chen, X., and He, Z. (2019). Patch-based self-adaptive matting for high-resolution image and video. *Vis Comput*, pages 133–147.

Erofeev, M., Gitman, Y., Vatolin, D., Fedorov, A., and Wang, J. (2015). Perceptually Motivated Benchmark for Video Matting. In *BMVC*.

Gastal, E. S. L. and Oliveira, M. M. (2010). Shared Sampling for Real-Time Alpha Matting. *Computer Graphics Forum*, pages 575–584.

Gastal, E. S. L. and Oliveira, M. M. (2011). Domain Transform for Edge-aware Image and Video Processing. In *ACM SIGGRAPH 2011 Papers*, pages 69:1–69:12.

Karacan, L., Erdem, A., and Erdem, E. (2017). Alpha Matting With KL-Divergence-Based Sparse Sampling. *IEEE Transactions on Image Processing*, pages 4523–4536.

Lang, M., Wang, O., Aydin, T., Smolic, A., and Gross, M. (2012). Practical Temporal Consistency for Image-based Graphics Applications. *ACM TOG.*, pages 34:1–34:8.

Levin, A., Lischinski, D., and Weiss, Y. (2008). A Closed-Form Solution to Natural Image Matting. *IEEE TPAMI 2019*, pages 228–242.

Li, D., Chen, Q., and Tang, C. (2013). Motion-Aware KNN Laplacian for Video Matting. In *2013 IEEE International Conference on Computer Vision*, pages 3599–3606.

Wang, J., Bhat, P., Colburn, R. A., Agrawala, M., and Cohen, M. F. (2005). Interactive Video Cutout. In *ACM SIGGRAPH 2005 Papers*, pages 585–594.

Xu, N., Price, B., Cohen, S., and Huang, T. (2017). Deep Image Matting. *arXiv:1703.03872 [cs]*.

Zheng, Y. and Kambhamettu, C. (2009). Learning based digital matting. In *2009 IEEE 12th International Conference on Computer Vision*, pages 889–896.

Zou, D., Chen, X., Cao, G., and Wang, X. (2019). Unsupervised Video Matting via Sparse and Low-Rank Representation. *IEEE TPAMI 2019*, pages 1–1.