# Video Matting Using Multi-frame Nonlocal Matting Laplacian

Inchang Choi, Minhaeng Lee, and Yu-Wing Tai

Korea Advanced Institute of Science and Technology (KAIST)

**Abstract.** We present an algorithm for extracting high quality temporally coherent alpha mattes of objects from a video. Our approach extends the conventional image matting approach, i.e. closed-form matting, to video by using multi-frame nonlocal matting Laplacian. Our multi-frame nonlocal matting Laplacian is defined over a nonlocal neighborhood in spatial temporal domain, and it solves the alpha mattes of several video frames all together simultaneously. To speed up computation and to reduce memory requirement for solving the multi-frame nonlocal matting Laplacian, we use the approximate nearest neighbor(ANN) to find the nonlocal neighborhood and the k-d tree implementation to divide the nonlocal matting Laplacian into several smaller linear systems. Finally, we adopt the nonlocal mean regularization to enhance temporal coherence of the estimated alpha mattes and to correct alpha matte errors at low contrast regions. We demonstrate the effectiveness of our approach on various examples with qualitative comparisons to the results from previous matting algorithms.

## 1 Introduction

Video matting is widely used in various video editing tasks [1], such as video object cut and paste. A major difference between image matting and video matting is that video matting requires *temporal coherence* across the alpha mattes extracted from each video frame. When applying conventional image matting algorithms [2–13] to video matting problem, noticeable flickering artifacts occur along matting boundaries due to the temporal inconsistency of extracted alpha mattes across consecutive video frames. In addition, it is cumbersome for users to draw trimap/scribbles for each video frame for image matting.

In conventional video matting approaches, such as [9, 14, 15], the goal was to minimize the amount of user input by propagating trimaps from small number of frames to the entire video sequence. Since the propagated trimap is temporally consistent, the extracted alpha mattes also exhibit a certain degree of temporal coherence, which reduces the flickering artifacts along matting boundaries. However, since alpha mattes are still estimated individually in a frame-by-frame manner, flickering artifacts still occur especially when the color distribution of foreground and background regions is similar (e.g. Figure 1). There are previous works that tried to maintain the temporal coherence of video mattes by using optical flow to warp the alpha matte from previous frame to current frame and

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part VI, LNCS 7577, pp. 540–553, 2012.
© Springer-Verlag Berlin Heidelberg 2012

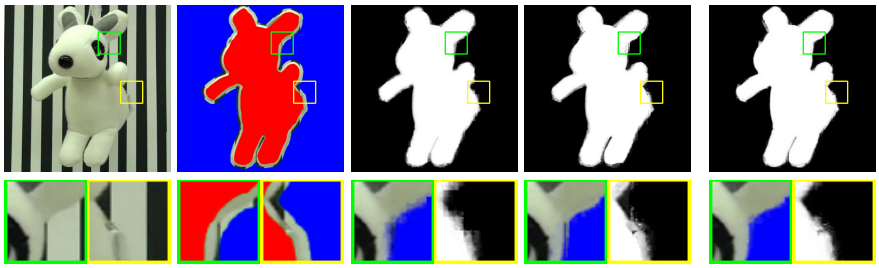


**Fig. 1.** In this example, the alpha matte for each individual frame can be easily distracted by the background with similar color. By including temporal coherence constraint, our results are less affected by the black and white strips of background. (a) Input frame. (b) Trimap. Alpha mattes from (c) Closed-form Matting [5], (d) Nonlocal Matting [11], (e) Our method.

then by using it as soft constraint in matting [16]. In [17], Lee *et al.* tried to include temporal coherence of alpha mattes by processing three consecutive frames together using a $3 \times 3 \times 3$ local window. Their method, however, still processes the video frames in a local manner, e.g. previous frame/current frame/next frame. Thus, temporal coherence across a long video sequence might not be well preserved.

In this paper, we propose a video matting algorithm which is based on the multi-frame nonlocal matting Laplacian. Our work is inspired by the recent work of nonlocal image matting [11]. We define a 3D nonlocal neighborhood in video to propagate matting constraints across a long video sequence. The 3D nonlocal neighborhood encourages pixels with similar color and texture across consecutive video frames to have similar alpha values. Our multi-frame nonlocal matting Laplacian solves the alpha mattes of multiple consecutive frames together simultaneously. Hence, it can effectively protect the temporal coherence of alpha mattes during optimization without smoothing out tiny structures of alpha mattes. To tackle the extensive memory requirement of the multi-frame nonlocal matting Laplacian and to improve efficiency of our algorithm, we use k-d tree to divide video frames into many small video blocks so that larger number of video frames can be encoded together in the multi-frame nonlocal matting Laplacian. After computing the alpha mattes for each video frame, we apply the non-local mean regularization [18] to further enhance the spatial and temporal coherence of alpha mattes across the whole video sequence.

In short summary, our technical contributions are in twofold: first, we propose the multi-frame nonlocal matting Laplacian for video matting; second, we introduce the nonlocal mean regularization in video matting to enhance the overall temporal coherence after individual estimation of alpha matte. We tested our video matting algorithm on various examples. Our experimental results show that our approach produces video mattes with better temporal coherence than video mattes from previous methods.

## 2 Related Work

There is a great deal of works targeting image/video matting problems. In this section, we review the most relevant works to ours. We refer readers to [9] and [19] for a more extensive survey on image/video matting.

In image matting problem, conventional approaches typically start with a user supplied trimap/scribbles. The provided trimap indicates definite foreground regions, definite background regions, and the unknown regions where the alpha values need to be estimated. Representative works included in [2–13]. The basic idea of these approaches is to use the definite foreground and background regions as hard constraints to infer the alpha values within the unknown regions. It is assumed that the colors within local regions are smooth [2], that the color gradients are smooth [3], or that the local color distribution satisfies a linear model [5]. As demonstrated in many previous works, better sampling strategy of foreground and background colors enables to estimate better alpha mattes [7, 12, 20, 21]. Such sampling strategy is not limited to a local manner. For instance, Lee and Wu [11] introduced the usage of nonlocal neighbors and modified the matting Laplacian formulation of [5]. Their approach demonstrates a higher quality of alpha mattes compared to the results from [5] but with fewer user input scribbles. This property is important in video matting since it is too labor extensive to ask users to draw trimap/scribbles in each video frame across the whole video. An approach to minimize user inputs while maintaining the original matting quality is necessary for video matting.

Early works in video matting such as [14, 22, 23] target to reduce user inputs by propagating trimaps from limited video frames [14] or by estimating the trimaps automatically through altering the video captures [22, 23]. However these approaches still estimate the alpha mattes individually for each video frame. Although accurate and temporally coherent trimap can alleviate the problem of temporal inconsistency of the estimated video mattes, there are still cases where temporal coherence is not well preserved. There are limited number of works that try to preserve temporal coherence of alpha mattes implicitly at the optimization level. In [16], Eisemann *et al.* used the alpha matte from previous frame as soft constraint to guide the alpha matte at current frame. Lee *et al.* [17] processed 3 consecutive frames together to estimate the alpha mattes of video through color distribution analysis in a $3 \times 3 \times 3$ local window. These approaches show that temporal coherence can be preserved better if temporal coherence can be modeled explicitly in the optimization process.

Comparing our works with previous works, especially with [16, 17], our approach considers a larger nonlocal neighborhood in spatial temporal domain to preserve the temporal coherence better. The uniqueness of our approach is the multi-frame nonlocal matting Laplacian that explicitly models temporal coherence in the matte optimization level. Further, by using nonlocal mean regularization as post-processing, we can effectively correct errors and remove artifacts that arose from our k-d tree segmentation based computation.

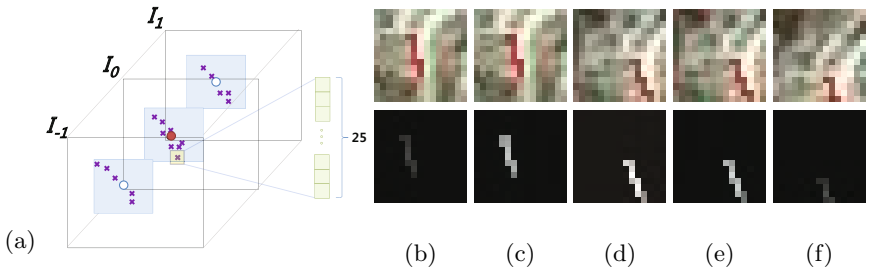


**Fig. 2.** Illustration of 3D Nonlocal Neighbor. (a) The purple crosses in spatial temporal volume of video are the 3D nonlocal neighbors of the red-colored pixel. (b)-(f) show the 3D nonlocal neighbor (red pixels) of the blue pixel in (d) in five consecutive frames. The 3D nonlocal neighbors track the hair structures according to color and texture similarity.

## 3 Algorithm

In this section, we describe our algorithm for video matting. We will first define the *3D nonlocal neighborhood*. Then, we will define the *multi-frame nonlocal matting Laplacian*. We will also describe our *k-d tree segmentation* to solve the memory requirement problem of multi-frame nonlocal matting Laplacian. Finally, we will describe the *nonlocal mean regularization* that refines the initial alpha mattes resulted from solving the multi-frame nonlocal matting Laplacian.

### 3.1 3D Nonlocal Neighbor

We define the 3D nonlocal neighborhood of a pixel by measuring the textural similarity of pixels in a spatial temporal video volume. The 3D nonlocal neighborhood extends the simple spatial window, i.e. $3 \times 3$ window, in closed-form matting [5] to formulate our multi-frame nonlocal matting Laplacian in video.

In order to find the 3D nonlocal neighbors of a pixel efficiently, we use the *Approximate Nearest Neighbor*(ANN) [24] searching algorithm. For each pixel in a video frame, we define a feature vector, which is a collection of pixel intensities around the $5 \times 5$ local window. Within each block of k-d tree implementation(Section 3.3), we search 30 nonlocal nearest neighbors from the ANN data structure. In order to guarantee sufficient nonlocal neighbors within the same video frame, we force the nonlocal neighborhood to have at least 10 samples coming from the same video frame within the same block. For each pixel, we have also searched 10 nonlocal neighborhood across neighboring blocks of k-d tree. The nonlocal neighbors within and across different blocks of k-d tree will be handled differently and will be detailed in Section 3.3. Figure 2 shows an example of 3D nonlocal neighborhood defined over a spatial temporal video volume.

After we define the 3D nonlocal neighbors of a pixel, we encode the textural and color similarity between each pair of nonlocal neighbors into a kernel function $k_{3D}(i, j)$:

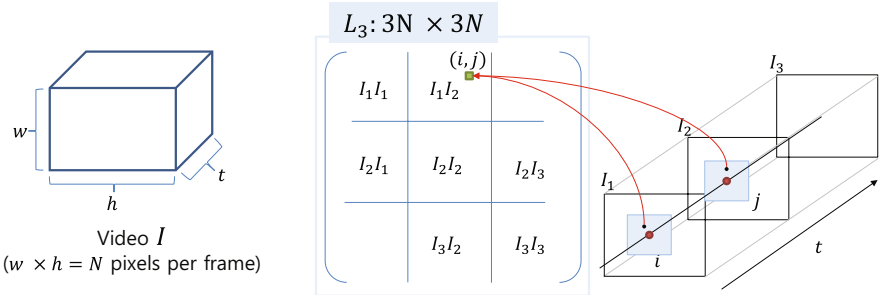


**Fig. 3.** Multi-Frame Nonlocal Matting Laplacian. For illustration, we show a 3-frame matting Laplacian $L_3$. The $3N \times 3N$ matting Laplacian encodes correlation across consecutive frames. For a pair of nonlocal neighbor, e.g. $i$-th pixel in $I_1$ and $j$-th pixel in $I_2$, across different frames, their similarity weighting is encoded at the $i$-th row and $j$-th column in the block $I_1I_2$.

$$k_{3D}(i,j) = k_t(i,j) \times k_s(i,j), \tag{1}$$

$$k_t(i,j) = exp(-\frac{dist_t(i,j)^2}{2\sigma^2}), \tag{2}$$

$$k_s(i,j) = exp(-\frac{1}{{h_1}^2}\|I(S(i)) - I(S(j))\|_g^2), \tag{3}$$

where $dist_t(i,j)$ is the temporal distance between pixel $i$ and pixel $j$, $S(i)$ is the spatial patch around of pixel $i$, and $\|\cdot\|_g$ is a Gaussian function with standard deviation equal to the radius of spatial patch $S$. The kernel function will be used to define the multi-frame nonlocal matting Laplacian to set weight on its matrix element as illustrated in Figure 3. Comparing our kernel function with the kernel function from [11], ours has additional term to give more weight to nonlocal neighbor with smaller temporal distance. The 3D nonlocal neighbors can effectively propagate alpha matte constraints within the multi-frame nonlocal matting Laplacian weighted by temporal distance and textural and color similarity. In our implementation, we set $\sigma^2 = 2$, $h_1^2 = 0.01$, and the window radius of $S(i)$ is 3 in Equation (1).

### 3.2 Multi-frame Nonlocal Matting Laplacian

The multi-frame nonlocal matting Laplacian is a matting Laplacian containing the affinities between pixels in a video volume (Figure 3). This big and sparse matting Laplacian matrix consists of several sub-matrices, where entries in each sub-matrix represent relationship between pixels in spatial temporal volume. We fill the multi-frame nonlocal matting Laplacian by using the 3D nonlocal neighbors defined in the previous subsection. Since the 3D nonlocal neighbors are defined across several video frames, it allows us to propagate matting constraints along temporal domain to preserve temporal coherence. The multi-frame nonlocal matting laplacian is defined as:

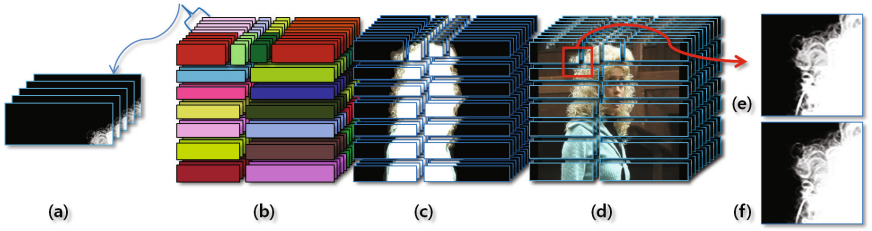


**Fig. 4.** K-d Tree Based Volume Segmentation. (a) Alpha mattes of a segment. (b) Visualization of segments. Segments with the same color belong to the same nonlocal matting Laplacian (c) Segmented trimaps. (d) Segmented frames. (e) Seam caused by individual estimation of alpha mattes. (f) Seams removed by a global step.

$$\alpha^T \mathbf{L_F} \alpha = \sum_{f=1}^{F} \sum_{i=1}^{N} \alpha(N_{3D}(i))^T \bar{G}_i^T \bar{G}_i \alpha(N_{3D}(i)), \tag{4}$$

where $F$ is the number of participating frames, $N$ is the number of pixels, $N_{3D}(i)$ is the 3D nonlocal neighbors of $i$, and $\alpha(N_{3D}(i))$ is a vector that represents alpha values of pixels in $N_{3D}(i)$. Our multi-frame nonlocal matting Laplacian is denoted as $\mathbf{L_F}$. $\bar{G}_i$ encodes pixel affinities among 3D nonlocal neighbors. We refer readers to [11] and our appendix for its derivation and details.

Similar to the previous works in [5, 11], the optimal alpha mattes can be obtained by solving:

$$(\mathbf{L_F} + \lambda D)\alpha = \lambda D \beta, \tag{5}$$

where $\alpha$ is a $FN \times 1$ vector whose entries are alpha values for participating frames, $D$ is a $FN \times FN$ diagonal matrix whose elements are 1 for constrained pixels and 0 for the unconstrained pixels, $\beta$ is a $FN \times 1$ vector which encodes constraints from user given trimap/scribbles, and $\lambda$ is a large constant that guarantees the solution is consistent with the constrained pixels. Note that the alpha values of $F$ frames will be solved together in this setting.

### 3.3 K-d Tree Segmentation

Solving Equation (5) allows us to process multiple frames together, but the size of the linear equation system is too large, and it requires a lot of memory and computation. In addition, this large linear system limits the number of frames that can be processed together. Hence, we introduce k-d tree segmentation, which partition video frames into many smaller blocks to reduce size of the linear system in Equation (5).

Our k-d tree segmentation is inspired by Fast Matting [6]. We partition the spatial temporal volume recursively to build the k-d tree. Within each block, we count the number of unknown pixels. If the number of unknown pixels is larger than a threshold, we continue our partitioning. In order to encode more

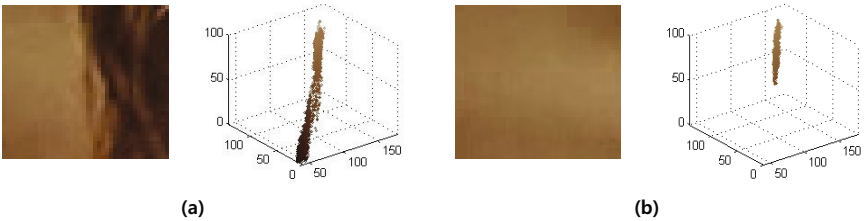


**Fig. 5.** (a) An image patch with high KL divergence. (b) An image patch with low KL divergence. High KL divergence means that the foreground and background colors are easily distinguishable which implies quality of estimated alpha mattes.

video frames, we fixed the number of frames in each block and partitioned the video blocks in spatial domain recursively as illustrated in Figure 4. The k-d tree manages the connectivity of each neighboring blocks.

Using the k-d tree segmentation, we can estimate alpha mattes of each video block individually. However, this method falls short in handling the nonlocal neighbors of a pixel across neighboring blocks and the boundary condition of alpha mattes between neighboring blocks. Here, we introduce a method to predict the goodness of the estimated alpha mattes within each block, and use this prediction to rank the order of video blocks. We solve the alpha mattes of video block according to the sorted order. The alpha mattes of solved video blocks are then used as soft boundary constraint to guide the alpha matte estimation of unsolved video block. The alpha values of nonlocal neighbors across different video blocks are also used as soft constraint to guide the alpha estimation within the center of video block. Hence, we can effectively propagate the alpha values across different video blocks without introducing much computation overhead. Also, since we start with salient video blocks where the estimated alpha mattes are more accurate, we can improve the accuracy of alpha mattes of video block with ambiguous color distribution.

To predict the quality of estimated alpha mattes within each video block, we evaluate the color distribution. We use *KL divergence* to measure color distance of unknown pixels to the foreground and background color distribution:

$$KL(S) = \sum_{i \in S} ||I(i) - C_0|| \log \frac{||I(i) - C_0||}{||I(i) - C_1||}, \tag{6}$$

where $S$ is unknown regions in a video block, and $C_0$ and $C_1$ are the two cluster centers representing the foreground and the background color distribution of a video block. Figure 5 shows an example of two image patches with high and low KL divergence. For a video block with distinguishable color distribution, we assume the estimated alpha mattes will be more accurate.

After we estimate the alpha mattes of all video blocks, we apply the global step [6] to further reduce seams across different video segment. The global step

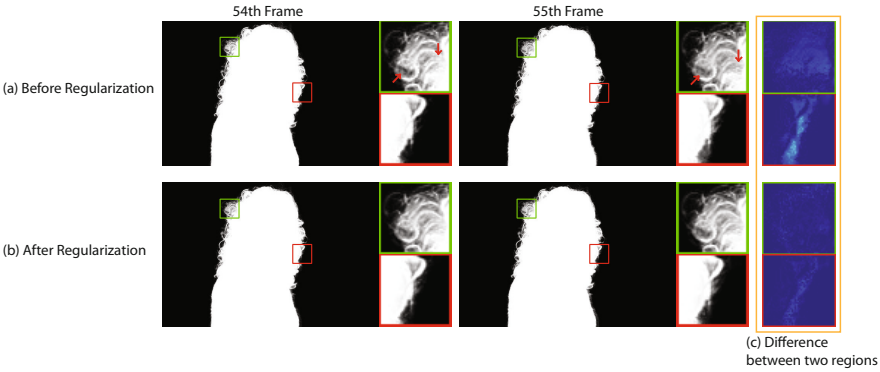


**Fig. 6.** Matte Regularization. From the left, each column represents the 54th frame and the 55th frame, respectively. (a) Before regularization. (b) After regularization. (c) Color map showing the intensity difference between two frames. A brighter pixel corresponds to a higher value. The 54th and 55th frames have exactly same scenes. Despite of it, you can notice different matting results for two frames in (c). When we integrate each frame into a video, this temporal inconsistency makes flickering effects.

perform Closed-form matting [5] on each frame individually with the estimated alpha mattes from previous processes as soft constraint as follow:

$$E = \alpha^T L\alpha + \lambda_1(\alpha - \alpha_{local})^T D(\alpha - \alpha_{local}) \tag{7}$$

where $L$ is the matting Laplacian of Closed-form matting [5], $\alpha_{local}$ is the estimated alpha mattes from the previous processes, and $D$ is a $N \times N$ identity matrix. We use the soft constraint as initial solution of the global step and run the conjugate gradient for 5 iterations to obtain our results.

### 3.4 Matte Regularization

After performing the global step, we obtain alpha mattes for each video frame. Although our multi-frame approach which solve alpha mattes of several consecutive frames simultaneously to produce temporally coherent mattes, there are still weak temporal inconsistency between alpha mattes. This inconsistency happened across alpha mattes in temporal domain when two consecutive frames belong to different video blocks in k-d tree segmentation. To further enhance temporal coherence, we post-processed the alpha mattes using nonlocal mean regularization. The nonlocal mean regularization [18] is commonly adopted in image denoising. We extended this process to video matting to filter out subtle temporally inconsistent of alpha values. We reuse the 3D nonlocal neighborhood defined in the previous subsection to enforce spatial and temporal coherence.

The nonlocal regularization of alpha mattes is defined as follow:

$$\alpha_i = \frac{\sum_{j\in\Omega} w(i,j) \times KL(S(j)) \times \alpha_j}{\sum_{j\in\Omega} w(i,j) \times KL(S(j))}, \tag{8}$$

where $w(i, j)$ is a weighting term in Equation (1) that measures colors and texture similarity between pixels, and $KL(S(i))$ is the KL divergence of color distribution of video block in Equation (6) which evaluates the accuracy of estimated alpha mattes in each video block. The inclusion of KL divergence allows us to give more weight to a pixel with a more accurate alpha value. After nonlocal mean regularization, we obtain our final temporally coherent video mattes. Figure 6 shows the effect of nonlocal mean regularization. The regularization smoothed the flickering effects of the estimated alpha mattes across video frames while the fine details of alpha mattes are preserved.

## 4 Results and Discussion

We show our experimental results in this section. We used video frames from a movie clip and the standard data set of [14]. In Figure 7 and Figure 8, we compared the results of our algorithm to those of the closed-form matting [5] and the nonlocal matting [11]. Since closed-form matting and nonlocal matting are methods for image matting problem, we also present comparisons to spectral video matting [16] in Figure 7-(B) and in Figure 8.

We performed our experiments on a machine with Intel Core i7 860 and 6.0 GB memory. The input video of our experiments has $720 \times 480$ resolution, and each frame has 38798 unknown pixels on average. In the experiment, it took 1.5 hours to process a 6-second video with 30 frames per second. Hence, it takes about 30 seconds for processing one frame. Our single-threaded C++ implementation was not well optimized. Therefore, we anticipate that a fully-optimized implementation or a gpu-based implementation would reduce the running time. We measured memory usage for estimating alpha mattes of k-d tree video segments. In the experiments, on average 417 MB was used for each video segments. This quantity includes space for sparse matrix data and additional space for solving linear equation systems. By virtue of k-d tree implementation, we could run the alpha matting within the memory budget.

Throughout the comparisons, our method shows better quality than closed-form matting, nonlocal matting, and spectral video matting in the experimental data set. In a relatively hard example where the boundary of the foreground object can be distracted by the background color, such as Figure 7-(A) and Figure 8, our method outperformed others. Because of the ambiguous foreground and background, the closed-form matting [5] failed to extract clean boundary of the object. Although the nonlocal matting [11] still results in noisy boundary, our results show clean and smooth object boundary extraction. In the case where the video frames have distinct colors for foreground and background, such as Figure 7-(B), our results are similar to those of [11], but better than the results from [5] and [16].

Together with the individual matting quality, our method also produced good results in term of temporal coherence. Methods of [5], [11] and [16] could not overcome their limitation as they still process each frame separately and individually. Even though the trimap is temporally consistent, their estimated alpha mattes failed to maintain some temporal coherence. In contrast, our results were

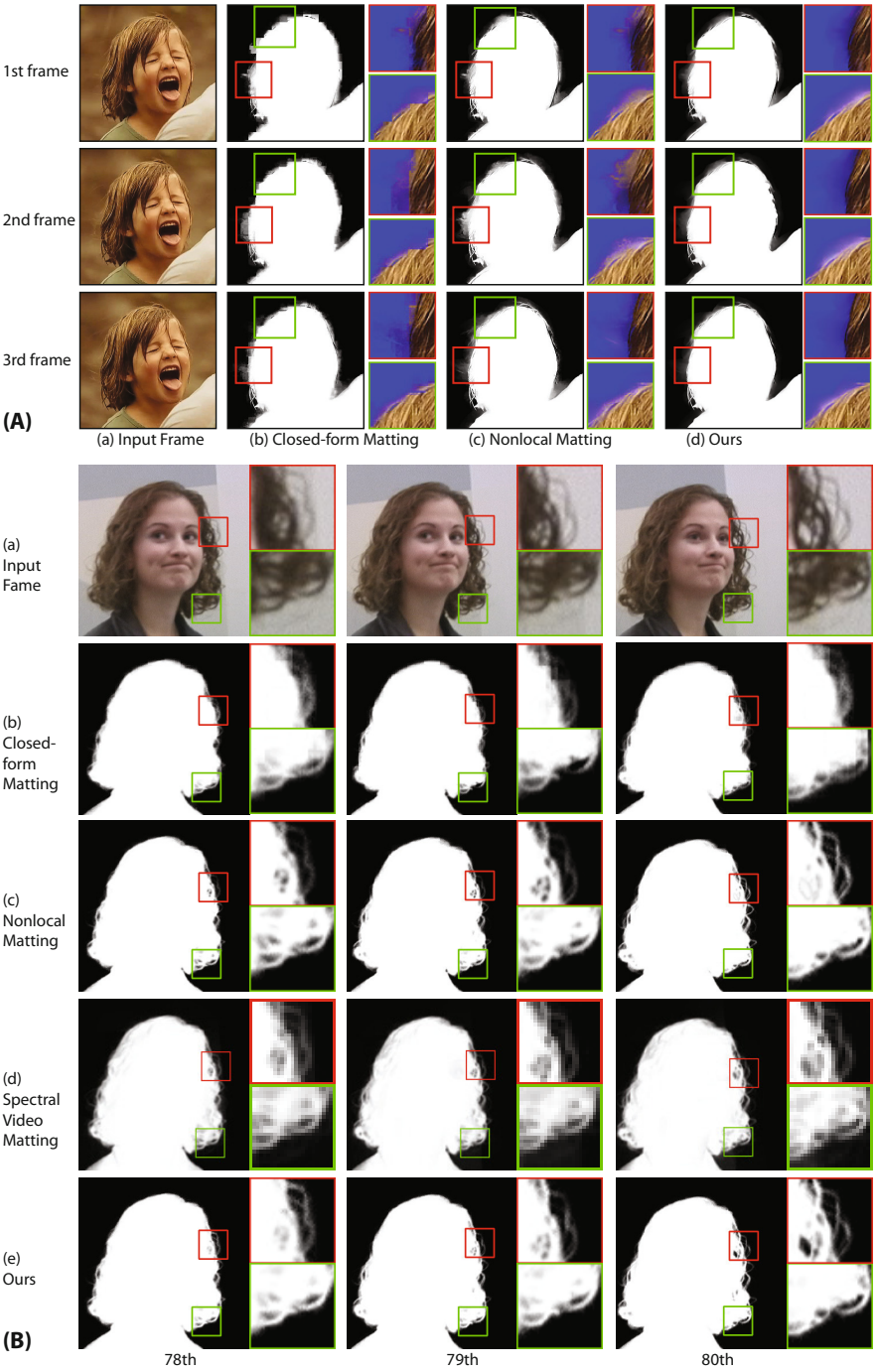


**Fig. 7.** Comparisons of our alpha matting results with results from other methods

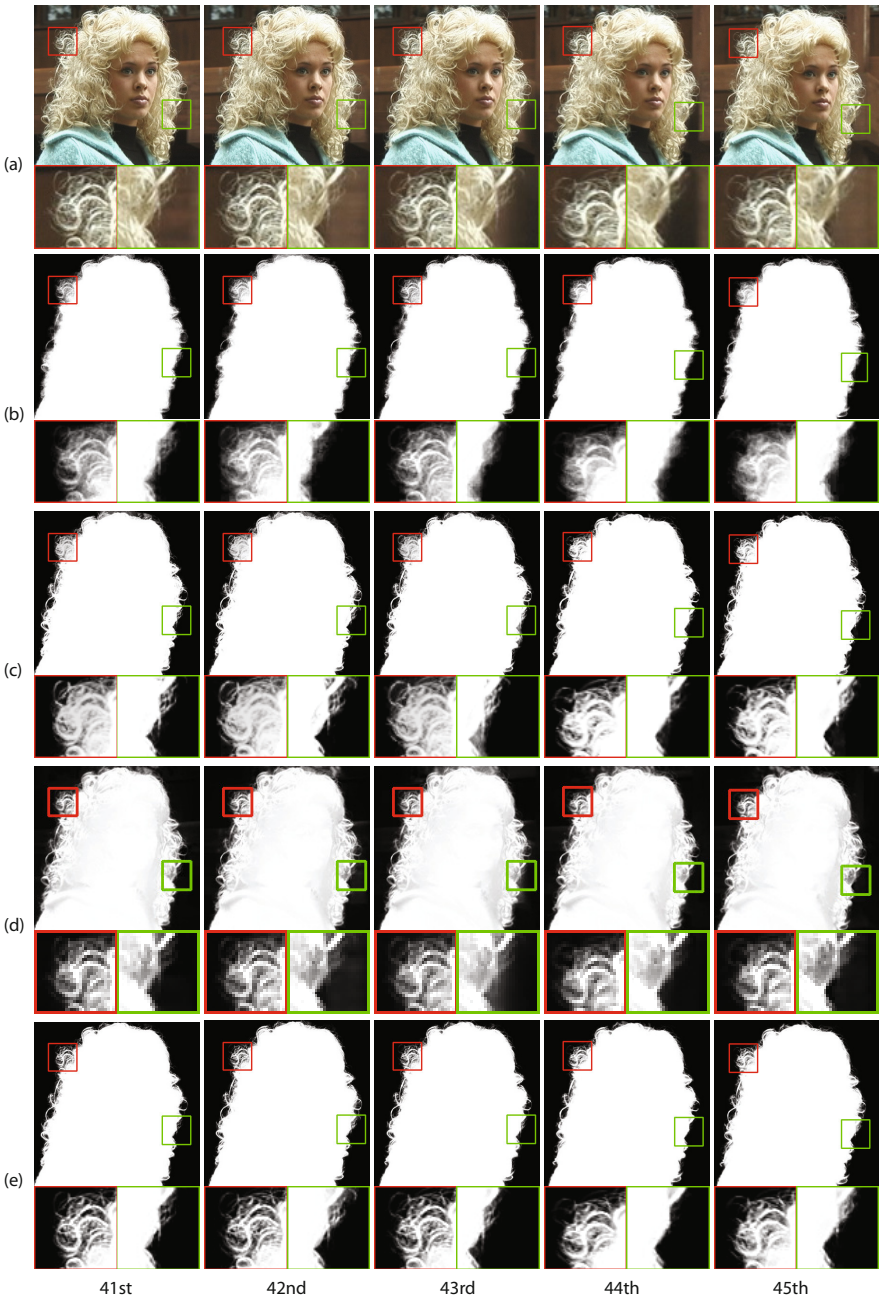


**Fig. 8.** Comparisons of alpha mattes. Each column represents results from different frames. (a) Inputs, (b) Results from Closed-form Matting [5], (c) Results from Nonlocal Matting [11], (d) Results from Spectral Video Matting [16], (e) Our results.

temporally coherent. We show the zoom-in areas of Figure 8 for better visualization and comparisons on the temporal coherence. In Figure 8, even though the zoom-in area display the same part for each frame, the alpha mattes' intensities of the closed-form matting [5] and the nonlocal matting [11] for the consecutive frames are not consistent. This causes flickering effects when we composite the object on a new video background. In comparison, our results in Figure 8-(e) exhibits not only clear and distinct hair textures in different frames, but also consistent alpha mattes over time. Our consistent alpha mattes, which are regarded as temporally coherent, produce natural and visually pleasing video without any flickering effect.

## 5 Summary and Conclusion

In this paper, we have presented a method for video matting, which enforces temporal coherence of alpha mattes explicitly using 3D nonlocal neighborhood and multi-frame nonlocal matting Laplacian. The 3D nonlocal neighborhood is effective in maintaining both the spatial and the temporal coherence of alpha mattes. To deal with heavy computations of video matting with multi-frame nonlocal matting Laplacian, we presented k-d tree implementation. It reduces computation by dividing the video volume into small video blocks. Each blocks are processed separately in local step and then combine together in global step to remove seams on the boundaries of segments. We have also adopted the nonlocal mean regularization to further enhance temporal coherence of our video mattes. The nonlocal mean regularization considers the color distribution of local regions, and a larger weight is given to the mattes of regions with distinct foreground and background color distribution. Our experimental results demonstrated good results in term of spatial matting quality and also in terms of temporal coherence comparing to previous works.

**Acknowledgement.** This research is supported by MCST and KOCCA in CT Research & Development Program 2012(R2010050008-00000003), the MKE Korea and Microsoft Research Asia under IT/SW Creative research program supervised by the NIPA (NIPA-2011-C1810-1102-0014), and the Basic Science Research Program through NRF Korea funded by MEST (2012-0003359).

## Appendix A Derivation of Multi-frame Nonlocal Laplacian

This appendix shows the derivation of Equation (4). According to the derivation of [5] and [17], we can express alpha values as:

$$\alpha(N_{3D}(i)) = \left[ I(N_{3D}(i)) \; \mathbf{1} \right] \begin{bmatrix} \mathbf{a_i} \\ b_i \end{bmatrix},$$

where $N_{3D}(i)$ is the 3D nonlocal neighbors of $i$, $\alpha(N_{3D}(i))$ is a vector that represents alpha values of pixels in $N_{3D}(i)$, and $I(N_{3D}(i))$ is the color vector of

$N_{3D}(i)$ with its column size equal to 3 for RGB channels and its row size equal to the number of 3D nonlocal neighbors of pixel $i$.

By multiplying the diagonal nonlocal weighting matrix $Q$, we obtain the following:

$$Q\alpha(N_{3D}(i)) = Q\left[I(N_{3D}(i))\ \mathbf{1}\right]\begin{bmatrix}\mathbf{a_i}\\ b_i\end{bmatrix} \triangleq I_0(N_{3D}(i))\begin{bmatrix}\mathbf{a_i}\\ b_i\end{bmatrix},$$

where

$$\begin{aligned} Q &= diag(k_{3D}(i,\cdot)/d_i),\\ d_i &= \sum_{j\in N_{3D}(i)} k_{3D}(i,j). \end{aligned}$$

In order to get a closed-form equation, we need to estimate $\left[\mathbf{a_i^*}\ b_i^*\right]^T$ that satisfies

$$\begin{bmatrix}\mathbf{a_i^*}\\ b_i^*\end{bmatrix} = \arg\min \left|\left| I_0(N_{3D}(i))\begin{bmatrix}\mathbf{a_i}\\ b_i\end{bmatrix} - Q\alpha(N_{3D}(i))\right|\right|.$$

After some mathematical rearrangement, we get $\left[\mathbf{a_i^*}\ b_i^*\right]^T$:

$$\begin{aligned}\begin{bmatrix}\mathbf{a_i^*}\\ b_i^*\end{bmatrix} &= (I_0^T(N_{3D}(i))I_0(N_{3D}(i)) + \epsilon\begin{bmatrix}\mathbf{I_d} & 0\\ 0 & 0\end{bmatrix})^{-1} I_0^T(N_{3D}(i))Q\alpha(N_{3D}(i))\\ &\triangleq I_0^{\dagger}\alpha(N_{3D}(i)).\end{aligned}$$

where $\epsilon$ is a small value that ensures $I_0^T(N_{3D}(i))I_0(N_{3D}(i))$ in Equation (9) has its pseudo-inverse.

Substituting $\left[\mathbf{a_i^*}\ b_i^*\right]^T$ in Equation (9), and multiplying $Q^{-1}$ in the both side of Equation (9) gives us

$$\begin{aligned}\alpha(N_{3D}(i)) &= Q^{-1}I_0(N_{3D}(i))I_0^{\dagger}\alpha(N_{3D}(i))\\ &\triangleq G_i\alpha(N_{3D}(i)).\end{aligned}$$

Finally, we can derive the quadratic cost function. Our multi-frame nonlocal matting Laplacian is denoted as $\mathbf{L_F}$, and $\bar{G}_i$ as $\mathbf{I} - G_i$. $\mathbf{I}$ is an identity matrix.

$$\alpha^T\mathbf{L_F}\alpha = \sum_{f=1}^{F}\sum_{i=1}^{N}\alpha(N_{3D}(i))^T\bar{G}_i^T\bar{G}_i\alpha(N_{3D}(i)).$$

## References

1. Litwinowicz, P.: Processing images and video for an impressionist effect. In: ACM SIGGRAPH (1997)
2. Chuang, Y.Y., Curless, B., Salesin, D.H., Szeliski, R.: A bayesian approach to digital matting. In: CVPR (2001)

3. Sun, J., Jia, J., Tang, C.K., Shum, H.Y.: Poisson matting. ACM Transactions on Graphics 23, 315–321 (2004)
4. Wang, J., Cohen, M.F.: An iterative optimization approach for unified image segmentation and matting. In: ICCV (2005)
5. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE Trans. on PAMI 30 (2008) 0162–8828
6. He, K., Sun, J., Tang, X.: Fast matting using large kernel matting laplacian matrices. In: CVPR, pp. 2165–2172 (2010)
7. Wang, J., Cohen, M.F.: Optimized Color Sampling for Robust Matting. In: CVPR (2007)
8. Wang, J., Agrawala, M., Cohen, M.: Soft scissors: An interactive tool for realtime high quality matting. ACM Transactions on Graphics 26 (2006)
9. Wang, J., Cohen, M.F.: Image and video matting: a survey. Found. Trends. Comput. Graph. Vis. 3 (2007)
10. Zheng, Y., Kambhamettu, C.: Learning based digital matting. In: ICCV (2009)
11. Lee, P., Wu, Y.: Nonlocal matting. In: CVPR (2011)
12. He, K., Rhemann, C., Rother, C., Tang, X., Sun, J.: A global sampling method for alpha matting. In: CVPR (2011)
13. Chen, Q., Li, D., Tang, C.K.: Knn matting. In: CVPR (2012)
14. Chuang, Y.Y., Agarwala, A., Curless, B., Salesin, D.H., Szeliski, R.: Video matting of complex scenes. ACM Transactions on Graphics 21 (2002)
15. Apostoloff, N., Fitzgibbon, A.: Bayesian video matting using learnt image priors. In: CVPR (2004)
16. Eisemann, M., Wolf, J., Magnor, M.: Spectral video matting. In: Proc. Vision, Modeling and Visualization, VMV (2009)
17. Lee, S.Y., Yoon, J.C., Lee, I.K.: Temporally coherent video matting. Graph. Models 72 (2010) 1524–0703
18. Buades, A., Coll, B.: A non-local algorithm for image denoising. In: CVPR (2005)
19. Rhemann, C., Rother, C., Wang, J., Gelautz, M., Kohli, P., Rott, P.: A perceptually motivated online benchmark for image matting. In: CVPR (2009)
20. Rhemann, C., Rother, C., Gelautz, M.: Improving color modeling for alpha matting. In: BMVC (2009)
21. Gastal, E.S.L., Oliveira, M.M.: Shared sampling for real-time alpha matting. In: Eurographics (2009)
22. Joshi, N., Matusik, W., Avidan, S.: Natural video matting using camera arrays. ACM Transactions on Graphics 25 (2006)
23. McGuire, M., Matusik, W., Pfister, H., Hughes, J.F., Durand, F.: Defocus video matting. ACM Transactions on Graphics 24 (2005)
24. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.: An optimal algorithm for approximate nearest neighbor searching. Journal of the ACM 45, 891–923 (1998)