

Inserting Videos into Videos

Supplementary Material

Donghoon Lee^{1,2} Tomas Pfister² Ming-Hsuan Yang^{2,3}

¹Electrical and Computer Engineering and ASRI, Seoul National University

²Google Cloud AI

³Electrical Engineering and Computer Science, University of California at Merced

In this supplementary material, we describe additional experimental results.

1. Video Files

As the problem on inserting videos into video is new in the field, there are no existing methods that achieve this task. In addition to this document, we provide 28 videos for comparisons against two strong baseline methods that require either expert manipulation or specific segmentation algorithm.

Video 1 - Video 22 (link). These videos focus on the DukeMTMC database. Sample frames from videos are shown in Figure 1 to Figure 6. For each figure, at the upper left corner of the footage, we display a frame from video *A* that contains the target object marked in a red box. Inserted objects into video *B* using the proposed algorithm are presented at the upper right corner. Rendering results of a video editing software, Adobe Premier Pro CC, is located at the bottom left corner as the first strong baseline method. We use blending mode of the software to automatically overlay two videos. The second strong baseline deployed at the bottom right corner is based on the state-of-the-art segmentation algorithm [1]. It often segments the target object incorrectly, *i.e.*, some parts are missing (Figure 1(a)) or backgrounds are included (Figure 1(b)). Experimental results show that the proposed algorithm synthesizes more realistic videos in most cases.

We discuss our two different failure cases shown in Figure 5 and Figure 6. If the image patch of the target object contains different objects or rare backgrounds, then the synthesized object is less realistic as shown in Figure 5. This issue can be alleviated by collecting more data. Occlusions caused by other pedestrians or objects in the scenes are another challenging case. If the object is occluded in video *A* as shown in Figure 6(a), then ideally the algorithm has to infer the occluded part and infill the missing part. In Figure 6(b), the object has to be inserted behind an existing object in video *B*. It is particularly challenging case since the algorithm has to decide whether the new object has to be inserted in front of the existing object or behind it. In addition, if the new object needs to be inserted behind the existing object, then it also has to determine which part should be visible. We note it requires scene parsing and understanding of 3D geometric to better infer how to seamlessly insert objects in videos, which will be our future work. It is also worth mentioning that our long-term goal is on video forensics (*i.e.*, to detect fake or tampered videos) although we focus on inserting videos into videos in this work.

Video 23 - Video 28 (link). In these videos, we present results of inserting a pedestrian in the DukeMTMC database into the TownCenter dataset. Results shows that objects are inserted realistically.

2. User Study

We perform a human subject study to evaluate the realism of synthesized videos. We conduct the experiments based on 22 test videos and 13 human workers. Each video contains 300 frames (5 seconds) while descriptions of each algorithm are replaced by method 1, method 2, and method 3 as shown in Figure 7. We ask workers to score each method from 1 to 5 (higher score for the better visual quality). Therefore, each worker actually needs to assess 66 different results. We provide two and three times slower videos with the original video to workers for more accurate evaluation. Table 1 shows the average score and percentage of cases that workers give the highest score to the method. We find that for 70% of the time the worker

Table 1: User study results on synthesized videos. Baseline 1 renders a video using a blending mode of the Adobe Premier CC Pro. Baseline 2 is based on a segmentation algorithm [1].

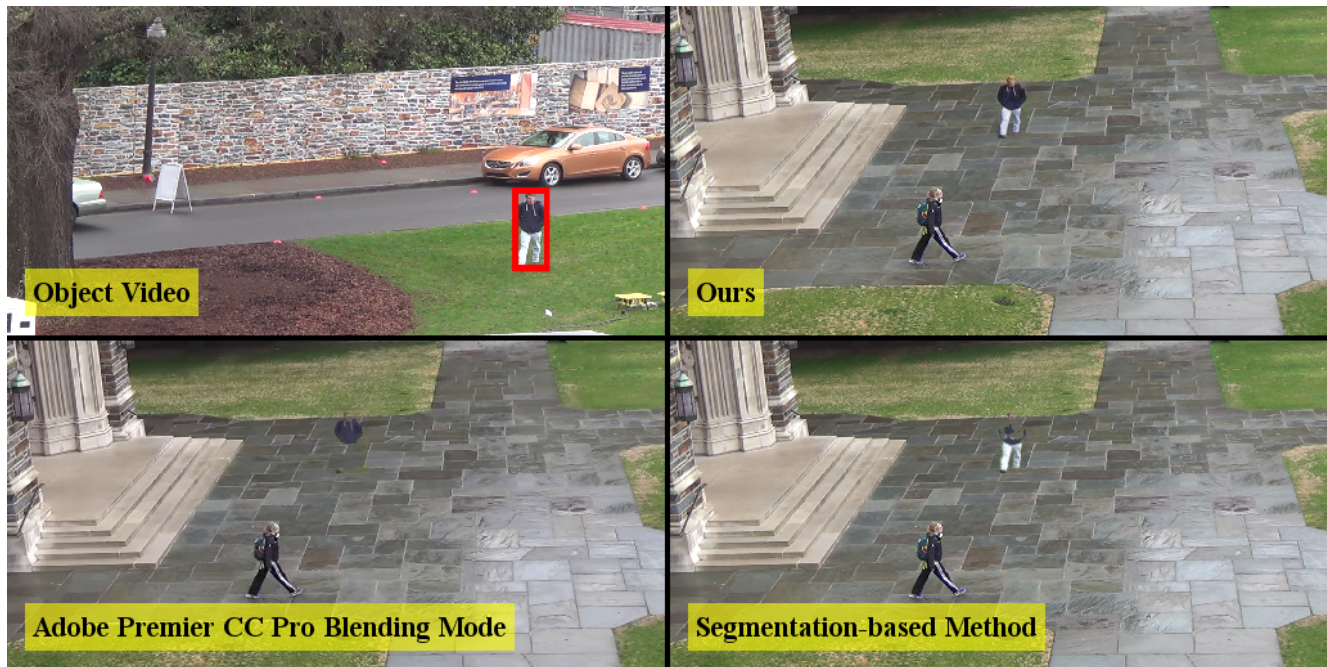
Method	Baseline 1	Baseline 2	Ours
Avg. Score	2.35	2.27	3.67
Preference	17.3%	13.7%	70.0%

preferred our approach than baseline methods. In addition, the proposed algorithm achieves significantly higher average scores.

3. More Implementation Details

Data preparation. The DukeMTMC dataset provides region of interest (ROI) to track pedestrians. We use bounding boxes of pedestrians in the ROI as training and test data. For r_A and r_B , we pick a random location and a size around the ROI. Then, we move r_A by following a movement of a random pedestrian in video A . We also scale the trajectory of the target object when it is inserted to video A based on the height ratio between r_B and u_A . It is based on our assumption that the length of each step is approximately proportional to the height of a person. For the TownCenter dataset, we use bounding boxes that are not cross the boundary of the image. As the dataset does not provide ROI, we randomly sample a location to insert an object around the center of the image.

Network training. While training, we use a parameter λ to control the importance between the real and fake pairs. It is multiplied with loss terms that are related to the fake pair. Empirically we find that $\lambda = 0.1$ makes the training process stable. To make the training more stable, we inject noise to previous frames when generating the current frame as discussed in the paper. Without the noise injection, the network blindly uses the information in the previous frame. It may result in propagating wrong pixel values over time as shown in Figure 8. To address this issue, we add $0.01 \times z$ at each pixel where z is sampled from a normal Gaussian distribution.

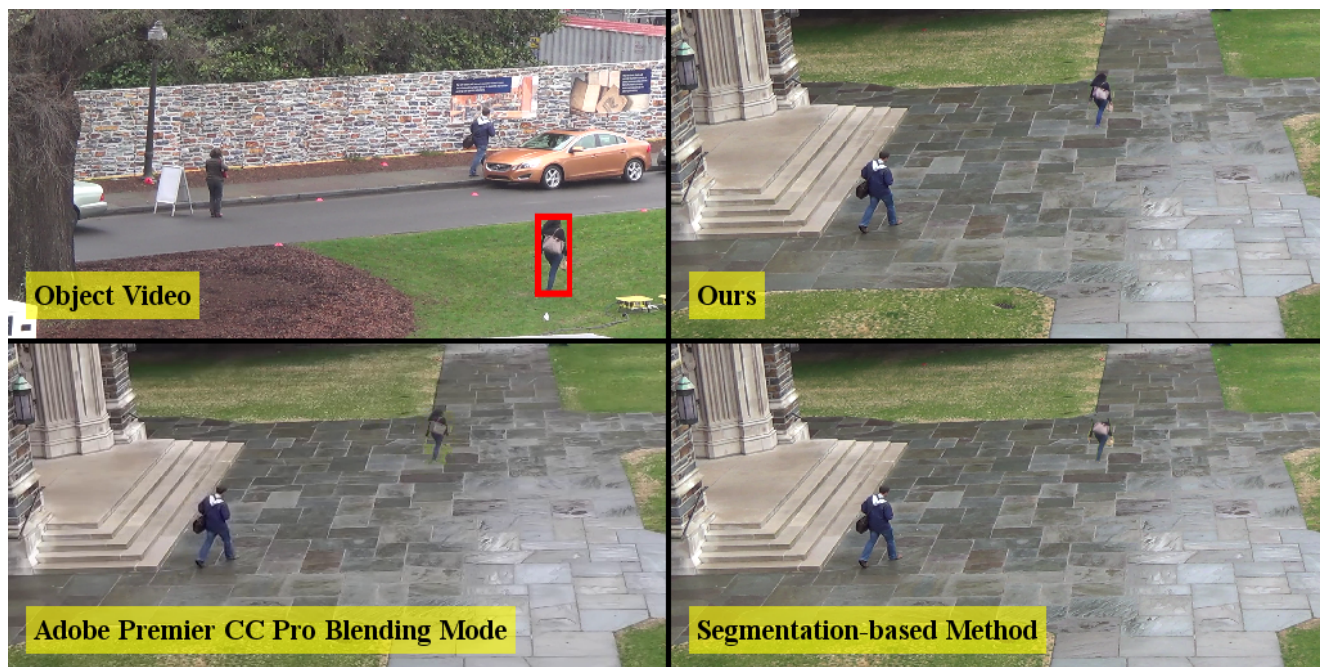


(a)



(b)

Figure 1: Performance evaluation with baseline methods. Upper left: a frame from a video of a target object. Upper right: results of the proposed algorithm. Bottom left: automated blending results using a video editing software. Bottom right: results based on a segmentation algorithm [1].



(a)



(b)

Figure 2: Performance evaluation with baseline methods.

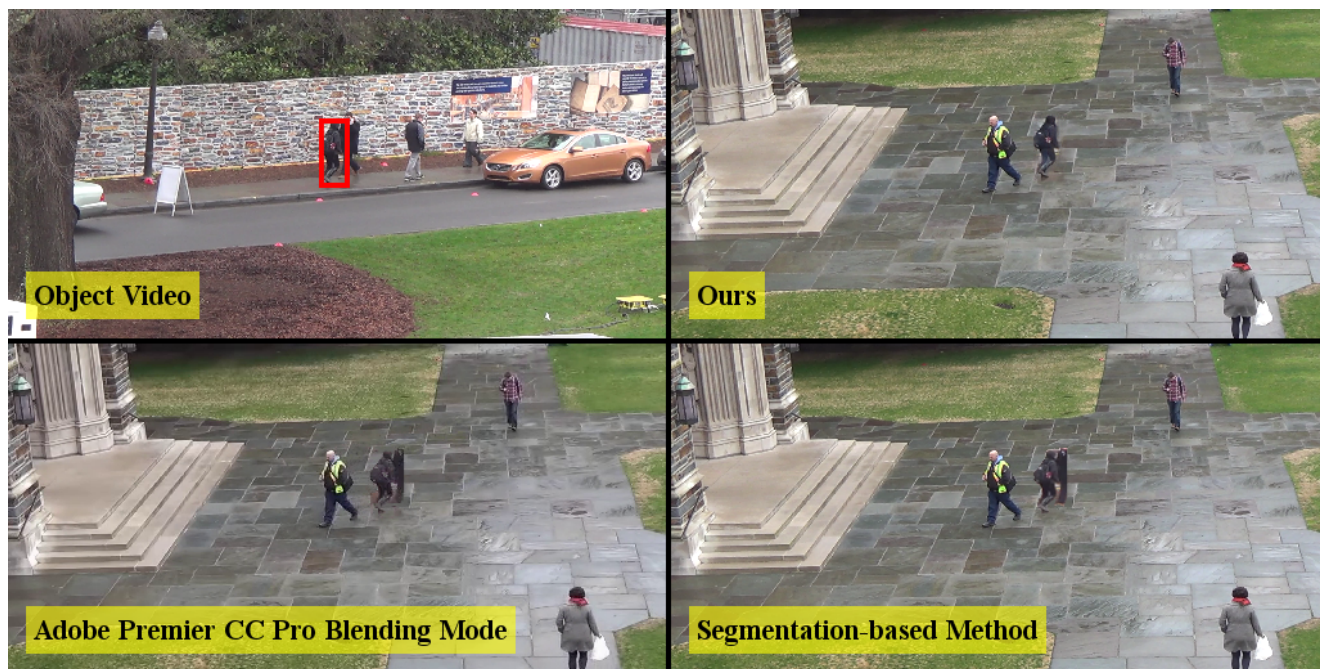


(a)



(b)

Figure 3: Performance evaluation with baseline methods.



(a)



(b)

Figure 4: Performance evaluation with baseline methods.



(a)



(b)

Figure 5: Performance evaluation with baseline methods. The results present failure cases of our algorithm.



(a)




(b)

Figure 6: Performance evaluation with baseline methods. The results show two different occlusion cases.


Video Synthesis User Study

We aim to insert a person in the red box to other scene videos. We present results of 3 different methods. Please evaluate each of them using a scale of 1 to 5. Score 5 represents the best quality. Thank you!


1




Object Video



Method 1



Method 2



Method 3

	1	2	3	4	5
Method 1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Method 2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Method 3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 7: An example of our user study layout. In the middle, a user can play the video.

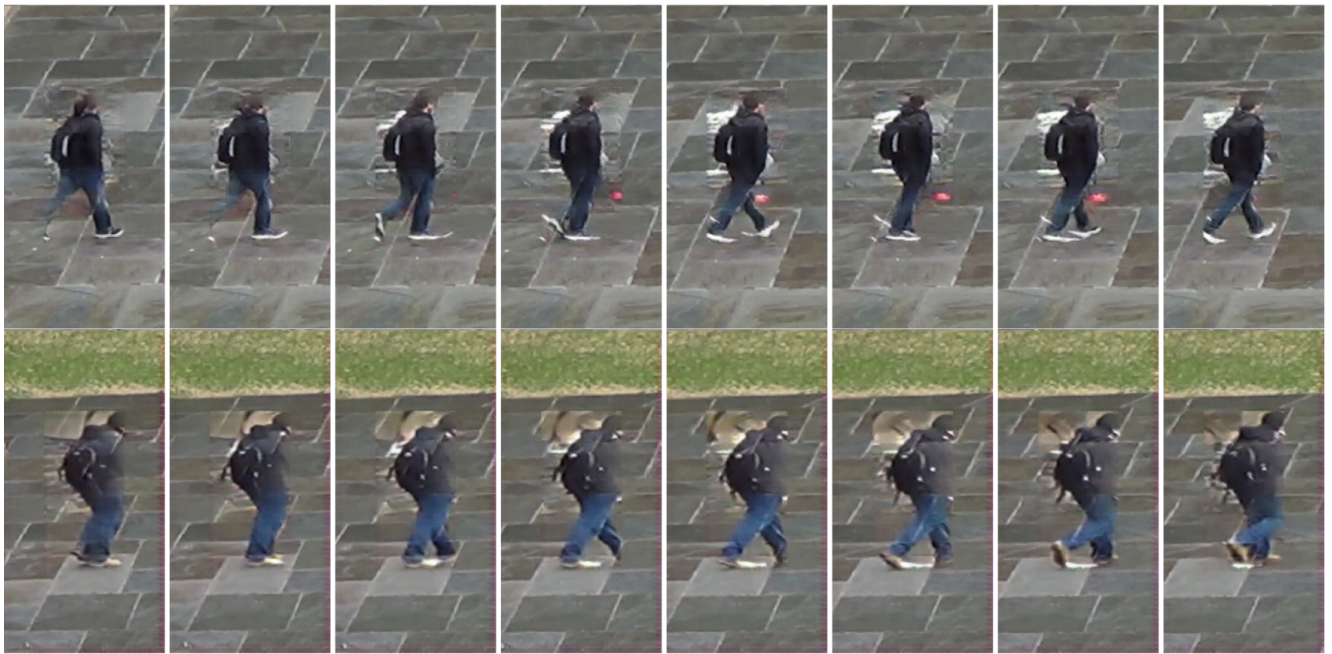


Figure 8: Results without noise injection to previous frames while rendering the current frame. It becomes easy for network to rely on previous frames and propagate wrong pixel values over time.

References

- [1] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision*, 2018. [1](#), [2](#), [3](#)