Efficient Mask Expansion for Green-Screen Keying using Color Distributions

Alexandre Derouet-Jourdan
OLM Digital Inc.

Marc Salvati
OLM Digital Inc.

Xiaoxiong Xing
OLM Digital Inc.

Takuro Nishikawa
OLM Digital Inc.

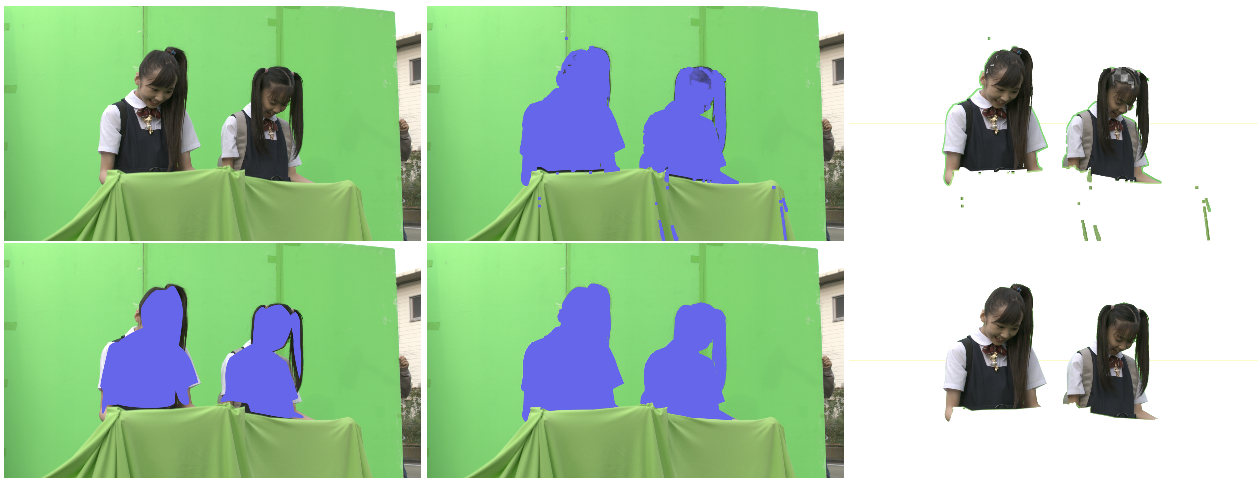

Figure 1: Mask expansion for a 4K frame. On the left, the input frame (top) and an input mask (bottom). In the center, the expanded masks, with Primatte (top), with our method (bottom), obtained with a similar working time. On the right, the respective expanded masks used in keying. The background color is pushed up to highlight the issues with Primatte's expanded mask. Frame is from the production of Secret×Heroine Phantomirage! ©T,O/PM,TX.

ABSTRACT

Masks are heavily used in image and video processing, particularly in the context of green screen keying. Designing good masks is a difficult task that involves painting over small details of images. Usually, only a rough mask is created. We propose an algorithm to expand such a mask, using color similarity. Our approach is fast, even on 4K images, and compares favorably with standard tools used in keying.

CCS CONCEPTS

• **Computing methodologies → Image processing**;

KEYWORDS

green screen keying, alpha matting, masking, color distributions

ACM Reference Format:

Alexandre Derouet-Jourdan, Marc Salvati, Xiaoxiong Xing, and Takuro Nishikawa. 2019. Efficient Mask Expansion for Green-Screen Keying using Color Distributions. In *Proceedings of SIGGRAPH '19 Posters.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3306214.3338595

SIGGRAPH '19 Posters, July 28 - August 01, 2019, Los Angeles, CA, USA

ACM ISBN 978-1-4503-6314-3/19/07.
https://doi.org/10.1145/3306214.3338595

1 INTRODUCTION

Masks are a crucial tool in production. They are heavily used in image and video processing to control algorithms and merge their results. In green screen keying, the compositing artist might provide a so called in-mask which contains pixels that must be preserved. They must not be considered as green screen or mixed with the green screen. In alpha matting, it is traditional to provide a trimask in input, that is a mask indicating known pixels from either background or foreground and unknown pixels. The problem is then to compute for each unknown pixel an alpha blending of a color attributed to background and one attributed to foreground.

As much as they are important and used in production, designing good masks is a tedious task. The design of a good mask requires the drawing of fine details on the input frame. It becomes infeasible when we consider a video, where drawing by hand detailed mask is far too demanding to be reasonable to tackle.

In the alpha matting algorithm presented in [Gastal and Oliveira 2010], a first step is to expand the mask by pixel similarity. For each pixel outside the mask, the algorithm looks in a given radius for a pixel belonging to the mask with a color difference below a given threshold. The justification behind this approach is that the expansion of the mask is faster than performing the alpha matting for all the unknown pixels.

Although this step greatly improves the computation time and the quality of the results in the original paper, we found out that it was scaling poorly for large pictures that we typically use in production. A higher resolution image requires a larger search radius, which means more pixels comparisons as we may intersect more pixels from the mask. In the case of green screen keying, we would like to draw by hand a small mask that is expanded further than what it was designed for in the original paper.

In this abstract, we propose a new approach to expand masks, based on color distributions [Aksoy et al. 2016]. Using color distributions removes the need to compare to specific pixels from the mask. Each unknown pixel in a given radius around the mask is compared to the color distributions only. Because we do not compare to specific pixels from the mask, we can use a distance transform to find candidates from the unknown pixels, which improves the search part of the mask expansion. Using a limited number of color distributions, typically under 10, the number of color comparisons is also reduced compared to the naive approach. Both improvements reduce drastically the computation time and make the mask expansion usable in practical applications (4K frames) such as production green screen keying.

2 ALGORITHM

As an input, we consider a frame in the RGB color space with a binary mask with value 0 for pixels outside of the mask and 1 for pixels inside the mask (Figure 1, first column). The algorithm visits the unknown pixels that are potential candidates to be included in the mask (subsection 2.1) and determine the distance to the pixel color similarity with the colors in the original mask (subsection 2.2). If the similarity is below a given threshold, then the pixel is added to the mask.

2.1 Distance to mask

Before comparing the color of an unknown pixel to the pixels inside the mask, we check the distance to the mask. To perform this query efficiently, we first construct a distance transform using [Borgefors 1986]. To prevent the mixing of pixels from different regions, we visit the unknown pixels in a breadth first approach. We add the direct unknown neighbors to the pixel in the mask to a queue. Then, when the first pixel from the queue is decided to be in the mask, we add its unknown neighbors to the queue.

2.2 Color similarity

Ideally, we would compare the unknown pixels' colors to every pixel from the mask. Unfortunately, when the mask is somewhat large, it is too heavy to compute efficiently.

To reduce the number of comparisons, we represent the pixels' colors in the mask with a limited number of color distributions [Aksoy et al. 2016], the number of which being left as a parameter to the user. To obtain N color distributions, we use the k-means algorithm to separate the colors into N sets. Then we compute the means and covariance matrices of each set.

In [Aksoy et al. 2016], the distance between a pixel's color and a color distribution is given as the squared Mahalanobis distance:

$$d_M(c, \mu, \Sigma) = (c - \mu)^T \Sigma^{-1} (c - \mu),$$

where c is the pixel's color, μ the mean of the distribution and Σ its covariance matrix. This distance involves the inverse of the covariance matrix of the distribution, which means that the maximum distance to a distribution can be huge. It makes this distance cumbersome to use and hard to predict for compositing artists. To tackle this issue, we use the Euclidean distance in RGB space, but we apply a scaling factor based on the covariance matrix:

$$d(c, \mu, \Sigma) = \|\exp(-\Sigma) \cdot (c - \mu)\|^2 .$$

When the covariance matrix is close to zero, this distance behaves like the Euclidean distance in the color space. As the covariance matrix grows, the space around μ will shrink and more pixels will fall under the threshold.

3 RESULTS

3.1 Time improvement over the naive expansion

The naive mask expansion approach consists in looking in a given radius to find a pixel that closely matches the unknown pixel. Because the frames we consider are large, it requires a large search radius, which makes this approach computationally costly. We compared the expansion of the same mask on a 4K video frame, using both our method and the naive one. Although the two expanded masks are of similar quality, our method took 2 seconds of computation, whereas the naive approach took above 10 minutes.

3.2 Quality comparison to a commercial keying tool

We compare our mask expansion with the commercial Nuke plugin Primatte (see Figure 1). It is an efficient keyer used in movie and tv production. Among many possibilities, it provides an in-mask from the keying input. Primatte being efficient, a possible use for these masks is to plug them into another keying node as a good first approximation of the masks to refine them. Compared to our mask expansion technique, it is more prone to noise, adding specks of false positive in the mask, which requires extra steps to clean up. It also sometimes leaves holes in the mask which need to be then filled manually. Overall, our approach, being tailored for mask expansion, allows to create mask of good quality with fewer interactions than Primatte, which remains a tool with much broader capabilities.

4 CONCLUSION

We presented in this abstract an efficient and powerful algorithm to expand masks based on color distributions. In this task, our mask expander compares favorably to Primatte, a standard Nuke plugin for keying. It can be used on large images and is thus fit for a production context.

REFERENCES

Yağiz Aksoy, Tunç Ozan Aydin, Marc Pollefeys, and Aljoša Smolić. 2016. Interactive High-Quality Green-Screen Keying via Color Unmixing. *ACM Transactions on Graphics* 35, 5 (Aug. 2016), 1–12. https://doi.org/10.1145/2907940

Gunilla Borgefors. 1986. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing* 34, 3 (June 1986), 344–371. https://doi.org/10.1016/S0734-189X(86)80047-0

Eduardo S. L. Gastal and Manuel M. Oliveira. 2010. Shared Sampling for Real-Time Alpha Matting. *Computer Graphics Forum* 29, 2 (May 2010), 575–584. https://doi.org/10.1111/j.1467-8659.2009.01627.x