

Real-time video chroma keying: a parallel approach based on local texture and global colour distribution

Ling Yin, Wenyi Wang , Jiying Zhao

School of Electrical Engineering and Computer Science, University of Ottawa 800 King Edward Ave., Ottawa, Ontario K1N 6N5, Canada
 E-mail: wwang.wenyi@gmail.com

Abstract: This study presents an automatic, human perception based chroma-keying algorithm that extracts the objects of interest (i.e. foreground) from monochromatic background. Given an image to be chroma keyed, the global colour distribution and the local texture property are analysed in CIECAM02 colour appearance model. After the analysis, input image is automatically segmented into three parts: foreground, background, and uncertain regions. Afterwards, the background colour is propagated from known background to uncertain region by using interpolation functions; and the foreground colour is estimated based on global colour distribution and a linear cost criteria. The quantitative and perceptual comparisons on the matting results show that the proposed method can reliably remove the background region, correctly restore the intrinsic foreground colour, and accurately keep the fine details. In addition, the authors implement the proposed method on a heterogeneous parallel computing architecture which efficiently distributes the workload among different processors. The simulation results show that the foreground objects can be accurately extracted from high-definition and/or ultra-high-definition videos in real time.

1 Introduction

Compositing is a category of techniques that merge multiple images based on the information attached to each picture element (e.g. pixel, in digital imaging), which is known as the ‘key’ [1]. The chroma key is the most commonly used key in photography process, film making, television production, and even augmented reality because of its simplicity and efficiency. The underlying principals are alpha blending and alpha matting. The ‘alpha’ was first introduced in 1970s by Catmull [2] to notate the image opacity. In 1984, Porter and Duff [3] discussed the arithmetic for compositing digital images using the alpha channel and the basis of alpha blending for computer graphics. In alpha blending, the image representation \mathbf{I} of the compositing result can be calculated by mixing foreground colour \mathbf{F} and background colour \mathbf{B} with a normalised foreground opacity factor α in a linear combination manner

$$\mathbf{I} = \alpha\mathbf{F} + (1 - \alpha)\mathbf{B} \quad (1)$$

However, this formula only partially models the colour mixing situation in the real world, where the seemingly translucent phenomenon is actually caused by blurring. On such occasions, the colour mixing model is additive and hence (1) applies. On other occasions, coloured translucent objects act as colour filters that filter out partial spectrum. This subtractive colour mixing model is considered beyond the scope of this paper. Today, most of the chroma keying and alpha matting algorithms utilise only (1) for the matting problem.

Since the image/video to be chroma keyed is often pictured in front of monochromatic background, both \mathbf{I} and \mathbf{B} in (1) should be apparent in an ideal chroma-key scene. However, \mathbf{I} is often the only known variable in practice because of the uneven lighting and other contaminations on the background in real-world productions. Benefited from monochromatic background, most industrial chroma-keying algorithms solve this equation by mapping alpha values to colour variances empirically and correcting foreground colours based on automatic analysis or manual input. Advanced algorithms such as Primate by Photron Limited may feature up to 128 different colour ranges in linear colour space [4]. Some of them also provide automaticity and real-time processing capability on proprietary hardware. In [5], the

researchers proposed a quadmap method for chroma keying to generate high-quality alpha matte. This method, however, cannot process in real time. On the other hand, alpha matting for natural images fascinates researchers for decades because of its capability of extracting foreground objects from arbitrary complex background. Recent alpha matting algorithms for natural images use (1) as a soft constraint and focus on improving the matte quality through adopting more sophisticated analysis and optimisations. The major drawback of applying existing alpha matting algorithms to chroma keying is that alpha matting algorithms often require manual inputs such as trimap or scribbles to indicate foreground and background samples. Some previous researches [6] indicated that such manual inputs can be done automatically through rough segmentation based on chromatic differences, however, they generally did not distinguish the seemingly translucent foreground and lost fine details such as hairs. In addition, their generality is achieved at the cost of high computational complexity, making them unsuitable for real-time video processing. Some other techniques require additional information such as multiple exposure [7–9] and invisible light properties [10–12]. Some of them may generate better results; they are usually not practicable in real-world applications.

In existing approaches, colour is often analysed in rearranged red-green-blue (RGB) colour spaces that attempt to be more intuitive and perceptually relevant, such as hue-saturation-lightness (HSV) and YC_bC_r. In this paper, we propose a novel framework to automatically process chroma keying [13, 14]. The chroma consistencies of different advanced colour spaces are compared and CIECAM02 colour appearance model is chosen and used for chroma keying for the first time. We introduced a perceptual analysis on global colour distribution and local texture to generate a trimap. With the help of the trimap and the proposed high-order interpolation function, we can smoothly and reliably estimate the pixel-wise background colour under the uncertain regions. Finally, accurate alpha matte and reliable foreground colours can be estimated from the information on major colour composition and restored background. Due to the non-recursive and pixel-independent nature of the proposed method, GPU-based parallel computing is applied to significantly improve the computational efficiency and thus real-time processing can be achieved.

ISSN 1751-9659

Received on 24th July 2015

Revised on 2nd February 2016

Accepted on 21st February 2016

doi: 10.1049/iet-ipr.2015.0450

www.ietdl.org

The rest of this paper is organised as follows. In Section 2, we draw perceptual comparison among different colour spaces and propose to use CIECAM02 colour appearance model for chroma keying. Section 3 introduces the proposed chroma-keying system, including parallel colour histogram analysis, clean background plate restoration, foreground colour prediction, and alpha matte estimation. In Section 4, experimental results are presented and compared with respect to calculation efficiency, cross-platform robustness, quantitative quality, and perceptual quality. Finally, conclusion is drawn in Section 5.

2 Perceptually uniform colour space

Before we start to solve the foreground colour and alpha matte, it is necessary to choose an appropriate colour space from which our colour estimation can benefit. Considering the final goal of chroma keying is to provide perceptually good compositing, a colour space which is consistent with human perception is inherently desired. The colour model is concerned in two aspects: (i) the colour space is based on luminance and chrominance components that are perceptually relevant to human vision; (ii) the colour space is perceptually linear so that the same distances represent approximately the same visual difference in human colour sensation.

Four perceptually uniform colour models that are commonly used or have proven advantages [15, 16] were chosen as candidate colour models for our perceptual analysis – CIE 1976 (L^*, a^*, b^*), IPT, CIECAM02, and CIECAM02-UCS. In order to compare these colour models, the perceptual linearity is analysed with regard to hue constancy and saturation constancy on background regions. It is intuitively apparent that the chromaticity of the background

region is highly uniform to human sensation, whereas the hue and saturation of the background colour should be within a very small vicinity in a colour space with good perceptual linearity. We refer to this property as background hue and saturation constancy and compare it among aforementioned colour spaces as follows.

For CIE 1976 (L^*, a^*, b^*) colour space, the hue is taken from its cylindrical representation – CIE 1976 (L^*, C_{ab}^*, h_{ab}). Although the saturation property is not officially defined in CIE 1976 (L^*, a^*, b^*), we take the suggestion from Fairchild in [17] to approximate the perceived saturation by dividing its chroma (C_{ab}^*) by its lightness (L^*). Likewise for IPT colour space, the hue is also taken from its cylindrical representation, and the saturation is approximated by dividing its polar radial distance on the chrominance plane ($\sqrt{P^2 + T^2}$) by its luminance (I). As the CIECAM02 colour appearance model and its derived CIECAM02-UCS colour space share the same hue angle and hue composition, the two related properties are evaluated once while their saturations are evaluated independently. In total, 1302 green screen images and 617 blue screen images (screen part only) from photos and video frames were used for evaluation. The hue are quantised with a resolution of 0.0025 (equivalent to 0.9° angle); and the saturation is quantised with a resolution of 0.01. Each test image was evaluated independently in hue and saturation with each colour model by measuring the proportion of background pixels that can be covered within a certain derivation range around their peak values. The homogeneous properties are statistically analysed among all test images. The results are intuitively shown with box-whisker plots in Fig. 1. We picked up the hue composition (H), saturation (s), and lightness (J) in CIECAM02 colour appearance model as the perceptually uniform colour space, and CIE 1931 XYZ colour space as the linear blending colour space in our proposed framework.

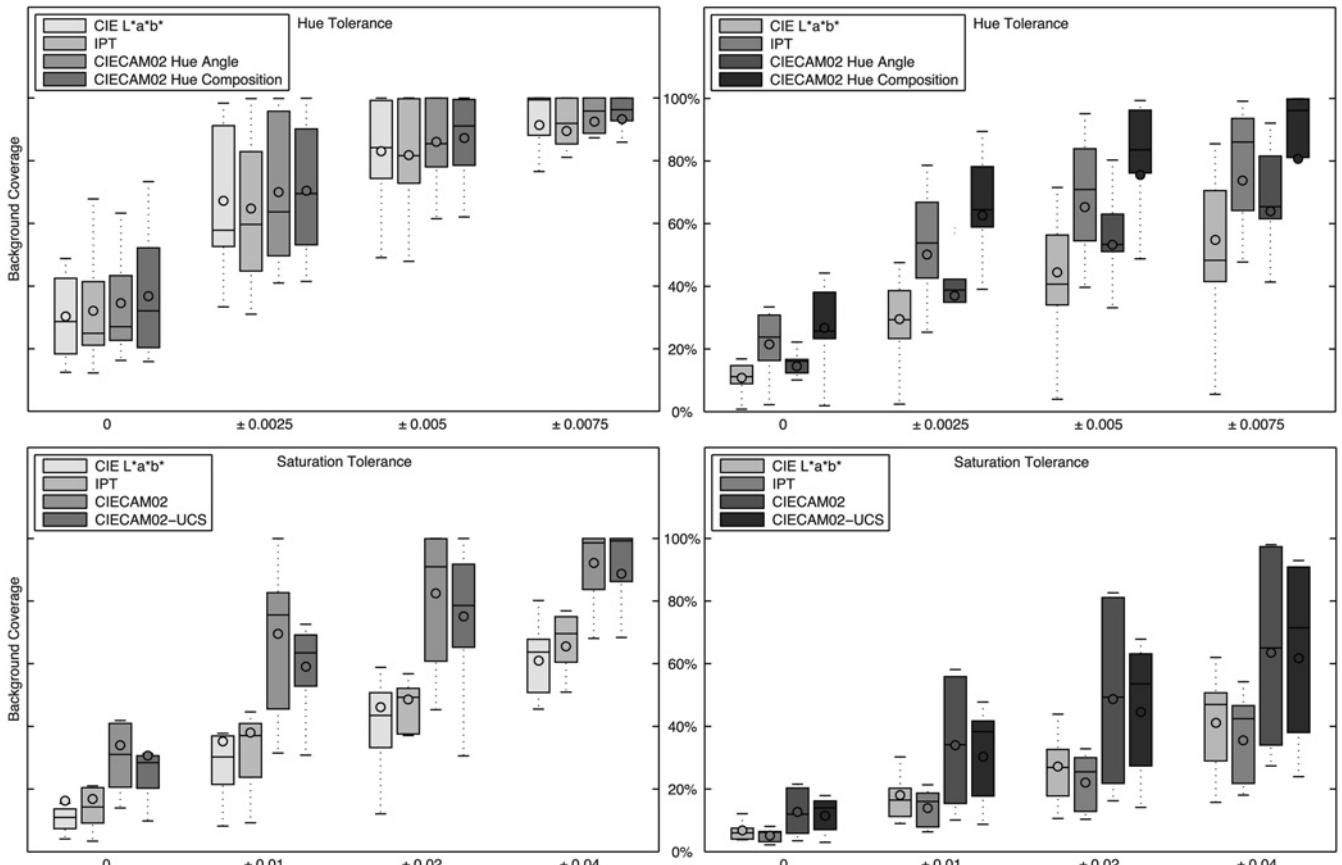


Fig. 1 Vertical axis shows the intra-image percentage of detected background in the entire background according to the background colour tolerance range shown on the horizontal axis, the box-whisker plots show statistic results across all test images against different background colour tolerance range. On each box, the bar is the median, the circle is the mean, the lower and higher edges of the box are the 25th and 75th percentiles, respectively. The shorter the box is, the more tested images have the similar background coverage. The higher the box is located, the more proportion of background is covered by the same derivation range of hue or saturation

3 Chroma keying optimised for heterogeneous parallelism

Observations show that most viewers judge visual quality on harmony, vividness, and details in experiential terms, rather than accuracy and possibility in physics terms. As our ideology of chroma keying is perception oriented, we decided to build a system simulating human sensation and cognition on chroma-keying problems. Considering the complex processing nature of such sophisticated system, we carefully designed an architecture that distributes different kind of computational workloads to different type of processors in an asynchronous manner. It aims to maximise the efficiency of our algorithms and achieve real-time processing capability for high-definition (HD) and even ultra-high-definition (UHD) videos. This heterogeneous parallel computing architecture is briefly illustrated in Fig. 2.

While we use hardware and driver-level optimisation to ensure transmission efficiency between memories controlled by different processors, different stages of our chroma-keying algorithm are allocated on different processes to provide the extensibility to hardware configurations with multiple GPUs and/or CPUs.

Generally speaking, the calculation tasks are assigned to different CPUs/GPUs with the consideration of two requirements: parallelisation and synchronisation. The parallelised GPU computing involves the tasks that can be done independently for each pixels, such as histogram analysis, local gradient estimation, image initial segmentation, background colour propagation, foreground colour prediction, and alpha value estimation. Besides the parallel computing, synchronisation is the other problem to be concerned in our system design. A calculation is referred to as synchronous if it is done for each video frame independently. On the contrary, a calculation is referred to as asynchronous if it is done only once across multiple frames, and the calculation results are shared among these frames. Specifically, GPU1 in Fig. 2 takes responsibility of synchronous tasks, such as initial image segmentation, background propagation, foreground colour prediction, and alpha value estimation. At the same time, GPU0 in Fig. 2 takes responsibility of asynchronous tasks, such as histogram analysis because we assume that the global colour distributions of adjacent frames do not drastically change.

3.1 Automatic trimap generation based on global colour distribution and local texture gradient

Trimap is a segmentation map that categorises an image into three different types of regions – absolute foreground (T_F), absolute background background (T_B), and uncertain regions (T_U). Our automatic trimap generation uses histogram-based global colour distribution analysis to understand major colour composition and local texture gradient analysis to extract trivial stimuli like hairs.

3.1.1 Global colour distribution: Three consecutive stages of histogram analysis are, respectively, deployed on hue, saturation, and luminance channels to estimate the global colour distribution, as shown in Fig. 3. In Fig. 3a, The hue histograms are used to locate the dominant hues which represent the major colour tones in the image. The dominant hues are defined by the local maxima H_{pki} ($i \in \mathbb{N}$) in full hue composition range. After that, saturation histograms are drawn at each of the dominant hues, H_{pki} , as shown by the curves in Fig. 3b. The width of each saturation curve represents the saturation range of each dominant hue, H_{pki} . With dominant hues and their associated saturation range, the major colour components can be determined by the trapezoid clusters on the chroma panel as shown in Fig. 3b. Within each trapezoid cluster, the lightness histograms are further used to determine the lightness range of each colour component as shown in Fig. 3c. Note that, the numbers in Fig. 3c represent the upper/lower bounds of the lightness range for each major colour component. Considering the application constraints in a chroma-keying setting, the general background colour tone is expected to be known. In this case, the background dominant hue H_B can be easily picked out from H_{pki} . With H_B and its associated saturation range, we can find out the initial T_B . Since the colours of seemly translucent objects in T_U can be regarded as the transitional colours from background to foreground, the hue value in T_U should be close to the background dominant hue H_B . In this case, an angular distance threshold H_t is used to define the colours in the initial T_U . Pixels with hue values within the range of $[H_B - H_t, H_B + H_t]$ are considered to belong to T_U while the remaining regions are defined as T_F .

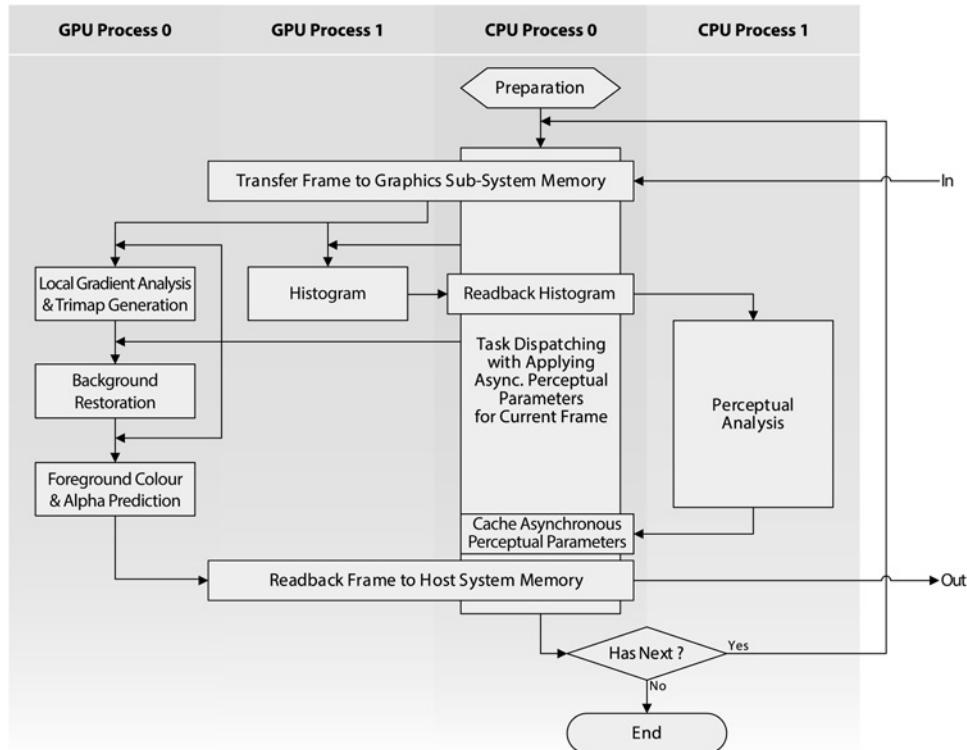


Fig. 2 Workflow of the proposed chroma-keying system

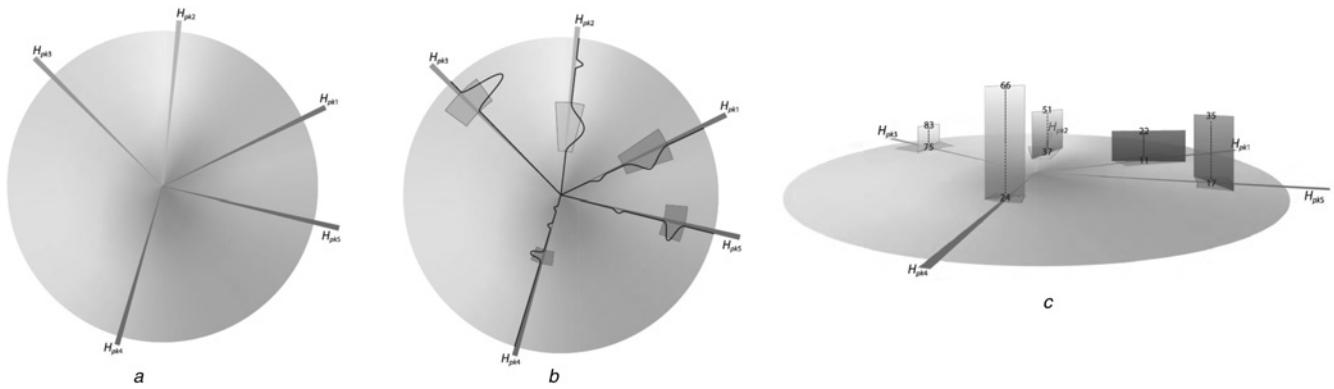


Fig. 3 Illustration for perceptual analysis in terms of hue, saturation, and lightness

- a Hue analysis
- b Saturation analysis
- c Lightness analysis

Our GPU-based implementation uses different conditions on performing three kinds of histograms, which are efficiently switched using vertex shader subroutines to avoid run-time conditional decisions on histogram procedure selection. For saturation and lightness histograms, multiple output arrays are used. Given a full HD video frame ($1920 \text{ px} \times 1080 \text{ px}$) with eight dominant hues, we test our parallel histogram analysing method across different GPU platforms, including GPUs from NVIDIA (GK20A and GK107), Intel (HD Graphics 4000), and Imagination Technologies (PowerVR G6430). In order to make the comparison, we also serially implemented our algorithm on a CPU (Intel Core i7) and recorded the running time. The comparison result can be found in Table 1. It can be observed that our histogram analysing can be fast enough for real-time processing on every GPU we have tested.

3.1.2 Local texture gradient: Detail perceiving is important in HD image and video processing because even tiny local inconsistency may attract visual interest and affect the quality of visual experience. On the other hand, noise is inevitable in image/video production and may complicate detail perceiving. To enhance the quality of trimap, we first conduct a separated bi-lateral filtering [18] for edge preserving and noise reduction as the pre-processing, and then analyse the local texture gradient using Sobel operator as shown in (2) and (3) to extract the horizontal and vertical details

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * L \quad (2)$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * L \quad (3)$$

Table 1 Time consumption of histogram analysing on different platforms (millisecond)

Processor	Core i7	G6430	GK20A	HD G4000	GK107
frequency	2.7 GHz	450 MHz	852 MHz	1.3 GHz	900 MHz
cores	4	64	192	16	384
FP32 Op/s	90 G	115 G	364 G	332vG	691 G
TMUs	–	8	8	8	32
Programming	C	GLSL ES	GLSL ES	GLSL	GLSL
hue	170.48	2.37	1.32	3.34	0.67
saturation	236.28	2.40	1.37	3.52	0.82
lightness	272.41	2.44	1.38	3.52	0.82
total	679.17	7.21	4.07	10.38	2.31

FP32 Op/s indicates how many 32-bit single precision floating point operation a processor can perform per second, TMU is the acronym of texture mapping unit which can be understood as the unit for pixel addressing.

where L is the lightness channel of the image under CIECAM02 representation.

Local gradient magnitude calculated from (4) is used to extract fine details. The overlapped regions of extracted details and initial T_B are categorised into the final T_U to ensure details are processed in the following stages. While the T_F is remain unchanged, the T_B usually becomes smaller

$$G_{\text{mag}} = \sqrt{G_x^2 + G_y^2} \quad (4)$$

Here we use the first frame of Godiva Medium image sequence from [19] as an example. In the trimaps shown in Fig. 4, white regions represent absolute T_F with α values equal to 1, black regions represent T_B with α value equal to 0, and grey regions represent T_U . The trimap can been viewed as a rough alpha matte with hard segmentations, and it also reduces the computational cost by eliminating unnecessary foreground colour prediction and α estimation on known regions. The comparison between trimaps shows that our automatically generated trimap has more details preserved in T_U and much cleaner T_B .

3.2 Solving the background colours

With the generated trimap, pixel-wise background colour is estimated in this section. The textureless background is one of the largest advantages in chroma keying over general image matting. Besides the occluded regions by the foreground, there are shades and foreground reflections that may contaminate the background uniformity. However, with prior knowledge on the background properties, observers tend to assume smooth transition for background colour. Therefore, we propose a smoothness enforced interpolation function $f(t)$ to recover the background colours in unknown region. In order to make the interpolated values to change smoothly from known boundary value, we enforce on $f(t)$ the conditions specified by (5) [22]. The second-order differential is used to enhance the smoothness enforcement. By solving the conditions in (5), we can get the interpolation function as shown in (6)

$$\begin{cases} f(0) = 0, \\ f(1) = 1, \\ f'(0) = f'(1) = f''(0) = f''(1) = 0. \end{cases} \quad (5)$$

$$f(t) = 6t^5 - 15t^4 + 10t^3 \quad (6)$$

To interpolate, boundary values are desirable. In this paper, an actinomorphic sampling strategy is used to collect background samples as interpolation boundary. Given a pixel $p \in T_U$ as shown by the grey dot in Fig. 5, a series of straight searching lines are

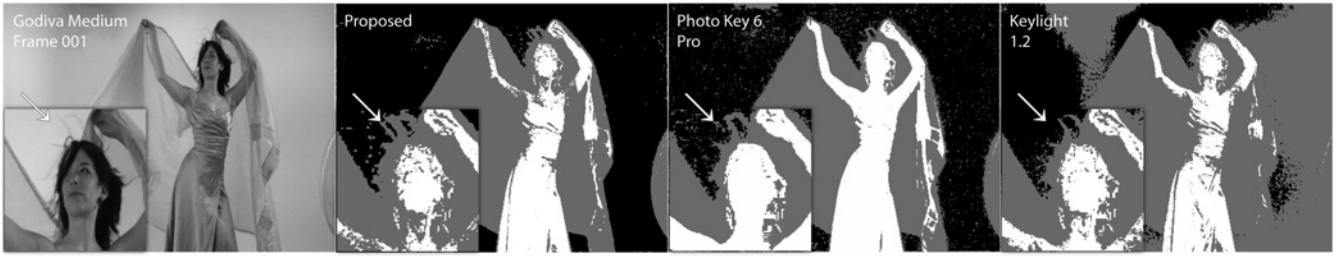


Fig. 4 Original video frame (the first frame of Godiva medium sequence from [3]), the automatically generated trimap from proposed method, the trimap from Photo Key 6 Pro [20], and the trimap from Keylight 1.2 [21] (bottom right). Arrows indicate the position of the hardly visible hair

drawn through this point. Taking the third straight line for example, two background points with colours $B_{P_i^a}$ and $B_{P_i^b}$ at locations P_i^a and P_i^b are supposed to be collected. These two points are the nearest ones to unknown pixel p in the two opposite directions of the third straight line. Since the searching range is limited, sometimes no background pixel can be found. In our example, we cannot find a background pixel for P_i^b . In this case, we assign P_i^b to be ∞ , and $B_{P_i^b}$ to be the global average background colour \bar{B} . Given one pair of background samples, an interpolation background colour B_p^i for pixel p can be calculated by (7). Combining all possible B_p^i together by (8), the final background colour for pixel p can be estimated. By doing this to every pixel in unknown region, reliable background colour estimation can be achieved

$$B_p^i = (B_{P_i^b} - B_{P_i^a})f\left(\frac{p - P_i^a}{P_i^b - P_i^a}\right) + B_{P_i^a} \quad (7)$$

where p is the spatial location of unknown pixel, f is the interpolation function we derived above, i represents the i th searching line, P_i^a and P_i^b are spatial locations of background samples, $B_{P_i^a}$ and $B_{P_i^b}$ are background sample colours

$$B_p = \frac{\sum_i w_i * B_p^i}{\sum_i w_i} \quad (8)$$

where w_i is defined as follows

$$w_i = e^{-(p - P_i^a)} + e^{-(p - P_i^b)} \quad (9)$$



Fig. 5 Illustration of actinomorphic sampling. The dashed lines indicate searching paths, filled circles indicate gathered background samples. Since the searching range is limited, rays are terminated with a red 'X' if the background samples are out of searching range

3.3 Solving the alpha matte and foreground colours

Unlike traditional sampling-based approaches which use limited number of samples from T_F and T_B , we use the colour information gathered from global colour distribution to form an abstract foreground colour set. The foreground colour of unknown pixel can be estimated based on this colour set as the following proposed method.

As shown in Fig. 6, the possible foreground colour chrominance are the colours in the trapezoid regions, which are also called the major colour compositions of the image in Section 3.1. Given an observed pixel $p_{(x,y)} \in T_U$, its foreground colour range can be determined by the recovered background colour $\hat{B}_{(x,y)}$ and the observed colour $O_{(x,y)}$ as shown by the dashed lines in Fig. 6. Ideally, the dashed line passes through the trapezoid regions (i.e. major colour components), and the colours on overlapped part between dashed line and trapezoid regions are chosen to be foreground chrominance candidates. Sometimes, the dashed line cannot pass through any of the trapezoid regions. In this case, colours in nearest trapezoid region are chosen as foreground chrominance candidates. Similarly, we can also determine the foreground lightness candidates depending on the lightness range obtained from colour distribution in Section 3.1. With foreground chrominance and lightness, we can regenerate the foreground colour $\hat{F}_{(x,y)}$ and further estimate the corresponding alpha value as shown in (10). In order to pick out the optimal $(\alpha_{(x,y)}, F_{(x,y)})$ pair, a cost function is defined by (11). The optimal $(\alpha_{(x,y)}, F_{(x,y)})$ pair

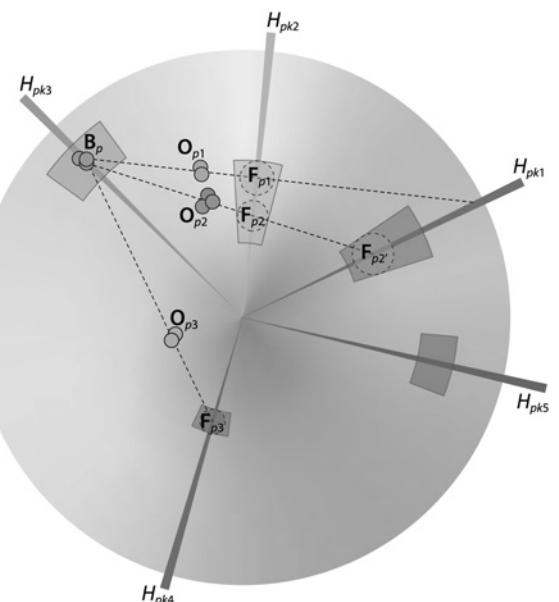


Fig. 6 Forecast chrominance prediction on hue-saturation chrominance plane

Table 2 Platforms used for the proposed implementations

Platform	CPU arch.	GPU model	Video acquisition	Avg. proc. time per frame	
				1920 × 1080	3084 × 2160
I	Apple iPad ME279C/A	ARMv8	G6430	on-device camera	30 ms
II	NVIDIA Dev. Board PM375	ARMv7	GK20A	external	14 ms
III	Personal Computer	x86-64	GK107	external	9 ms
				29 ms	46 ms

For GPU specifications, please refer to Table 1.

Table 3 Algorithms used to be compared

Algorithm	Host software	Default user input
proposed	Standalone	automatic
Shared Matting	Standalone [6]	trimap
Photo Key 6 Pro	Standalone (FXHome)	automatic
Modular Keyer	Smoke (Autodesk) [23]	colour picker
Ultra Key	Premiere Pro 8.0 (Adobe) [24]	colour picker
Keylight 1.2	After Effects 11.0 (Adobe) [25]	colour picker
Primate	NukeX 8.0 (The Foundry) [26]	automatic
Ultimate	NukeX 8.0 (The Foundry) [26]	colour picker

is the one minimising cost function as shown in (12)

$$\hat{\alpha}_{(x,y)} = \frac{(O_{(x,y)} - \hat{B}_{(x,y)})(\hat{F}_{(x,y)} - \hat{B}_{(x,y)})}{\|\hat{F}_{(x,y)} - \hat{B}_{(x,y)}\|^2} \quad (10)$$

$$d_p = \|O_{(x,y)} - (\hat{\alpha}_{(x,y)}^{(H,i)} \hat{F}_{(x,y)}^{(H,i)} + (1 - \hat{\alpha}_{(x,y)}^{(H,i)}) \hat{B}_{(x,y)})\| \quad (11)$$

$$(\alpha_{(x,y)}, F_{(x,y)}) = \underset{\hat{\alpha}_{(x,y)}^{(H,i)}, \hat{F}_{(x,y)}^{(H,i)}}{\operatorname{argmin}} d_{(x,y)}(\hat{\alpha}_{(x,y)}^{(H,i)}, \hat{F}_{(x,y)}^{(H,i)}) \quad (12)$$

where H represents a major colour component, i represents the index of colour candidate in one major colour component, and (x, y) represents the spatial location.

4 Experimental results

4.1 Processing speed

We implemented the prototype of our proposed chroma-keying system on three different platforms as listed in Table 2. Despite the trivial difference in programming language, all of the implementations share the same algorithm as presented in Section 3. According to our tests, the proposed method can process HD videos in real time at the speed of 30, 14, and 9 ms/frame on platform I/II/III. If the input video is UHD with the resolution of 3084 × 2160, our method can still work in real time on platforms II and III at the speed of 46 and 29 ms/frame.

While we had trouble to get the entire pipeline running on iPad with UHD resolution, we still get a compelling result on mediocre commercial-of-the-shelf GPU that can process an input with such resolution at 30 frame/s.

4.2 Evaluation and comparisons on chroma-key matting

In this section, our proposed chroma-key matting method is evaluated and compared with other eight state-of-the-art algorithms listed in Table 3. These methods include both academic researches in recently published papers and industry-established chroma keyers in commercial software. All evaluations are performed on platform III (personal computer) and the commercial software is running with their highest quality rendering setting. In the second column of Table 3, ‘Standalone’ indicates that the software is developed by the researchers while others indicate the name of the commercial software and their associated company. In the third column of Table 3, methods marked with ‘Automatic’ input do not need users to provide extra side information; methods with input ‘colour Picker’ need users to select one or multiple points on the image to specify the background colour; methods with input ‘Trimap’ need users to provide the segmentation map as we mentioned in Section 3.1. For algorithms that need a trimap as user input, we provide the

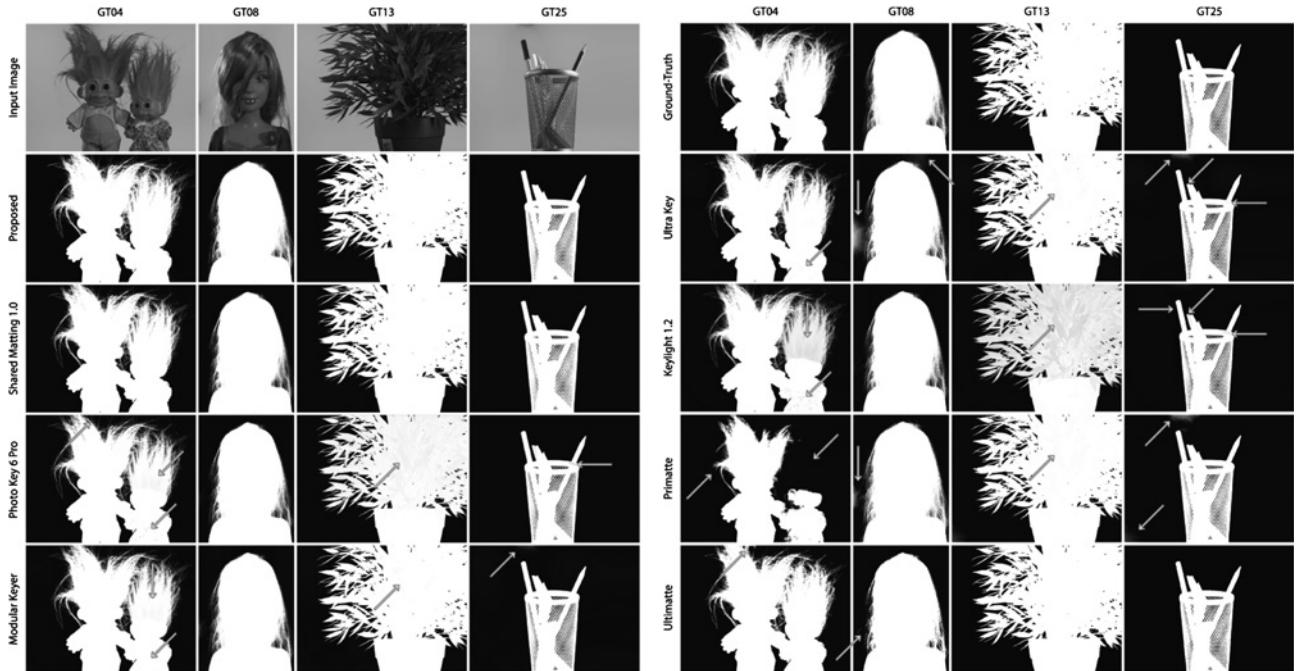


Fig. 7 Input images and chroma-keying results from different algorithms. The foreground objects are labelled as GT04, GT08, GT13, and GT25 from left to right

Table 4 SAD ($\times 10^3$) and MSE ($\times 10^{-3}$) of generated alpha mattes

	Algorithm	GT04		GT08		GT13		GT25	
		SAD	MSE	SAD	MSE	SAD	MSE	SAD	MSE
green background	proposed	10.70	1.55	5.93	0.63	2.38	0.22	2.79	0.49
	Shared Matting	11.05	1.98	9.21	1.63	4.31	0.66	4.69	1.33
	Photo Key 6 Pro	13.64	2.12	9.91	1.61	36.26	7.19	2.50	0.33
	Modular Keyer	17.81	2.37	21.82	3.22	19.32	2.34	31.84	3.70
	Ultra Key	22.07	4.49	24.38	8.56	16.83	2.13	23.63	3.66
	Keylight 1.2	20.94	3.56	10.37	1.62	61.12	13.30	6.59	0.74
	Primate	22.13	6.59	18.57	6.23	12.49	1.74	8.65	3.26
	Ultimatte	25.81	10.26	19.85	7.11	7.64	2.63	3.78	1.17
blue background	proposed	10.69	1.30	9.61	1.74	4.54	0.44	2.30	0.36
	Shared Matting	12.62	2.51	10.07	2.02	6.01	1.29	5.08	1.75
	Photo Key 6 Pro	15.48	3.29	13.85	4.03	6.07	1.58	2.94	0.88
	Modular Keyer	26.62	3.47	20.81	5.26	12.98	1.19	25.95	2.87
	Ultra Key	12.11	1.78	12.65	2.91	7.65	1.38	4.67	0.86
	Keylight 1.2	9.14	1.05	13.75	2.17	7.16	1.16	2.59	0.58
	Primate	12.30	2.06	11.89	3.29	4.44	0.73	2.68	0.75
	Ultimatte	15.37	3.04	13.50	3.58	4.58	0.84	2.58	0.76

automatically generated trimap map in our proposed method. It can be observed that only a few methods (i.e. ‘Photo Key 6 Pro’, ‘Primate’, and our proposed matting method) can automatically solve chroma-key matting problem without extra user input.

4.3 Quantitative evaluation

In this section, a quantitative evaluation is conducted on composited images, which are generated by composing 27 different foreground objects [27] onto real-world blue or green background.

Our proposed chroma-keying system is compared with other eight matting methods. Without losing generality, four foreground objects labelled as GT04, GT08, GT13, and GT25 are chosen to illustrate the algorithms’ capabilities. These foreground objects, which can be found in Fig. 7, address certain real-world challenges such as fine details, overlapped foreground and background colour distributions, ambient diffusions, and uneven background lighting. The sum of absolute difference (SAD) and mean squared error (MSE) are used to evaluate the difference between the generated alpha mattes and corresponding ground truth alpha mattes. The bold quality values are emphasised to represent the best matting result for each testing image. The numeric results in Table 4 show our overall advantages.

4.4 Perceptual quality comparison

The perceptual quality of the chroma-keying results from evaluated algorithms are also compared. The input images used for quantitative evaluation in Section 4.3 are presented together with their ground truth alpha mattes in the first row of Fig. 7. While the name of each algorithm is indicated at the beginning of each row, the arrows indicate obvious local flaws in alpha mattes such as: (i) dirty background in the results of ‘Modular Keyer’, ‘Ultra Key’, and ‘Primate’ for GT25; (ii) loss of fine details in the results of ‘Photo Key’, ‘Primate’, and ‘Ultimatte’ for GT04; (iii) erroneous foreground opacity in the results of ‘Photo Key’, ‘Ultra Key’, ‘Keylight’, and ‘Primate’ for GT04, GT13, and GT25.

The perceptual comparison proves the superiority of our proposed method in visual qualities, such as clean background plate, perceptually reasonable foreground, and accurate boundary details. On the contrary, other methods hardly generate high-quality matte with all those visual requirements satisfied, instead they usually compromise one in order to keep others.

In addition, three chroma-keying scenes, which were taken in field, are also used in the perceptual comparison. The scenes in the first row of Fig. 8 is set up by ourself, which contains challenging characteristics for chroma keying, such as uneven background,



Fig. 8 Evaluation frames (first column) and corresponding red-yellow checker compositing (second) results of proposed chroma-keying system, and magnified local comparisons on the alpha mattes between proposed and other seven state-of-the-art algorithms: A) Shared Matting, B) Photo Key 6 Pro, C) Modular Keyer, D) Ultra Key., E) Keylight 1.2, F) Primate, and G) Ultimatte

large transparent areas on the bottle and the cup, bright speckles on the camera body and the mug. The scenes in the second and third rows of Fig. 8 are from [19]. Large transparent and reflective regions can be, respectively, found in these two scenes. Compositing results show that our proposed method can predict reliable foreground colours even if the original scene contains large translucency or refracted areas. It can be also observed that our proposed method performs well in preserving fine details such as hairs floating in the air. Meanwhile, the alpha estimation is robust to the colour variation in the foreground object as well.

5 Conclusion

In this paper, an automatic high-performance chroma-key matting framework is proposed. Based on global colour distribution and local texture gradient, the generated trimaps are more effective because of the smaller unknown regions and richer details. The background inpainting technique significantly reduces the computational cost, and provides smoother and cleaner background plate in less time. The foreground colour is estimated in a faster and more robust way because the colour candidates are selected from image major colour composition in a statistic manner rather than geometry-based sampling.

The non-iterative and pixel-independent nature makes the proposed framework efficient in parallel computing environment. The proposed method is tested on three different GPU-based platforms. The qualitative and quantitative evaluations show that the proposed method is able to process HD and/or UHD chroma keying in real time with good composites and precise alpha mattes.

6 References

- 1 Kanade, T., Narayanan, P.J., Rander, P.W.: ‘Virtualized reality: concepts and early results’. Proc. IEEE Workshop on Representation of Visual Scenes, 1995, pp. 69–76
- 2 Catmull, E.: ‘A hidden-surface algorithm with anti-aliasing’, *Comput. Graph.*, 1978, **12**, (3), pp. 6–11
- 3 Porter, T., Duff, T.: ‘Compositing digital images’. Proc. ACM 11th Annual Conf. on Computer Graphics and Interactive Techniques, SIGGRAPH ‘84, New York, NY, USA, 1984, pp. 253–259
- 4 Photron Ltd.: ‘Primate whitepapers’. Available at <http://www.primatte.com/content.cfm?n=whitepapers>, accessed 2013
- 5 Wang, W., Zhao, J.: ‘Robust image chroma-keying: a quadmap approach based on global sampling and local affinity’, *IEEE Trans. Broadcast.*, 2015, **4**, (99), pp. 1–11
- 6 Gastal, E.S.L., Oliveira, M.M.: ‘Shared sampling for real-time alpha matting’, *Comput. Graph. Forum*, 2010, **29**, (2), pp. 575–584
- 7 Smith, A.R., Blinn, J.F.: ‘Blue screen matting’. Proc. ACM 23rd Annual Conf. on Computer Graphics and Interactive Techniques, SIGGRAPH ‘96, New York, NY, USA, 1996, pp. 259–268
- 8 Zongker, D.E., Werner, D.M., Curless, B., et al.: ‘Environment matting and compositing’. Proc. 26th Annual Conf. on Computer Graphics and Interactive Techniques, 1999, pp. 205–214
- 9 Sun, J., Li, Y., Kang, S.B., et al.: ‘Flash matting’, *ACM Trans. Graph.*, 2006, **25**, (3), pp. 772–778
- 10 Ben-Ezra, M.: ‘Segmentation with invisible keying signal’. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2000, pp. 32–37
- 11 McGuire, M., Matusik, W., Yerazunis, W.: ‘Practical, real-time studio matting using dual imagers’. Proc. 17th EuroGraphics Conf. on Rendering Techniques, 2006, pp. 235–244
- 12 Bentabeb, L., Zhang, H.: ‘Bayesian image matting using infrared and color cues’. Proc. Fifth Int. Conf. on Image and Signal Processing, 2012, pp. 1–8
- 13 Yin, L., Zhao, J.: ‘Real-time automatic chroma-key matting using perceptual analysis and prediction’. Proc. IEEE Canadian Conf. on Electrical and Computer Engineering, 2014, pp. 1–4
- 14 Yin, L., Zhao, J.: ‘Hybrid key: an automatic tool for real-time high quality chroma keying’. Workshop of IEEE Int. Conf. on Image Processing, Quebec City, QC, Canada, 2015
- 15 Li, C., Luo, M.R., Hunt, R.W.G., et al.: ‘The performance of CIECAM02’. Proc. 10th Color Imaging Conf.: Final Program and Proc., 2002, pp. 28–32
- 16 Xue, Y.: ‘Uniform color spaces based on CIECAM02 and IPT color difference equations’ (Rochester Institute of Technology, New York, NY, USA, 2008)
- 17 Fairchild, M.D.: ‘Color appearance models’ (Wiley Publishing, USA, 2013, 3rd edn.)
- 18 Pham, T.Q., van Vliet, L.J.: ‘Separable bilateral filtering for fast video preprocessing’. Proc. 2005 IEEE Int. Conf. on Multimedia and Expo, New York, NY, USA, 2005, pp. 1–4
- 19 Hollywood Camera Work LLC: ‘Green screen plates’. Available at <http://www.hollywoodcamerawork.us/greenscreensplates.html>, accessed 2013
- 20 FXHOME Ltd.: ‘Photokey 6’. Available at <http://fxhome.com/photokey-6>, accessed 2013
- 21 The Foundry Visionmongers Ltd.: ‘Keylight’. Available at <http://www.thefoundry.co.uk/products/plugins/keylight>, accessed 2013
- 22 Perlin, K.: ‘Improving noise’, *ACM Trans. Graph.*, 2002, **21**, (3), pp. 681–682
- 23 Autodesk Inc.: ‘Professional video effects software – smoke’. Available at <http://www.autodesk.com/products/smoke>, accessed 2013
- 24 Adobe Systems Inc.: ‘Adobe premiere pro’. Available at <http://www.adobe.com/products/premiere.html>, accessed 2013
- 25 Adobe Systems Inc.: ‘Adobe after effects’. Available at <http://www.adobe.com/products/aftereffects.html>, accessed 2013
- 26 The Foundry Visionmongers Ltd.: ‘Nuke’. Available at <http://www.thefoundry.co.uk/products/nuke-product-family/nuke>, accessed 2013
- 27 Rhemann, C., Rother, C., Wang, J., et al.: ‘A perceptually motivated online benchmark for image matting’. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2009, pp. 1826–1833