สัญญาเลขที่ R2560C057

รายงานฉบับสมบูรณ์

ชื่อโครงการ การแยกฉากหลังโดยการพิจารณาค่าสีด้วยหน่วยประมวลผลกราฟิก

คณะผู้วิจัย	สังกัด
1. ดร.ธนะธร พ่อค้า	ภาควิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร

สนับสนุนโดย

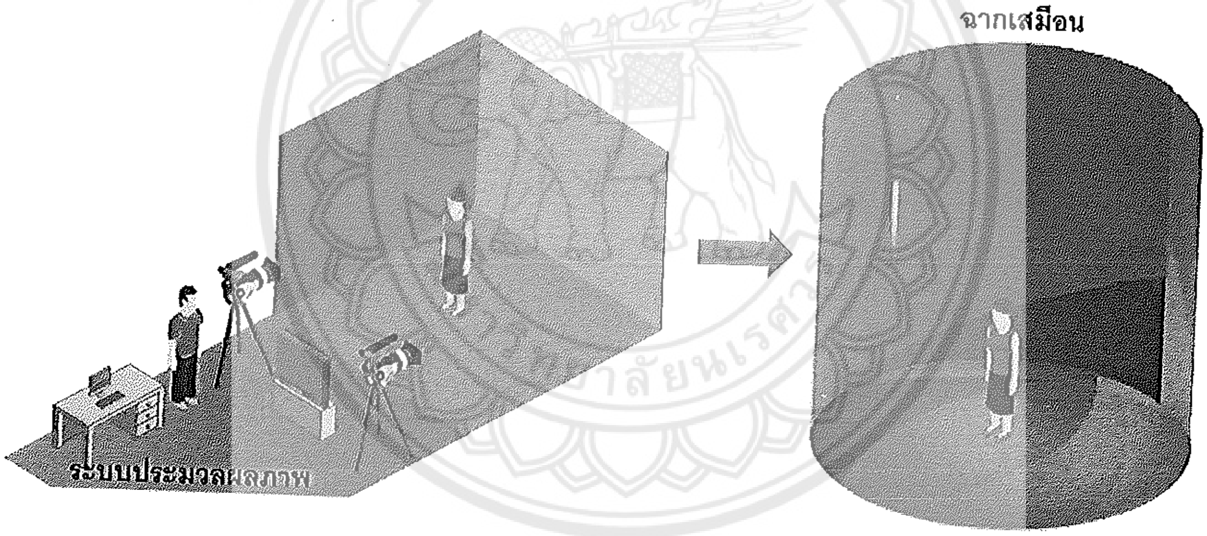
งบประมาณรายได้มหาวิทยาลัยนเรศวร

ปีงบประมาณ 2560

Executive Summary

งานแพร่ภาพออกอากาศเป็นสิ่งที่ได้รับความสนใจอย่างมากในยุคของทีวีดิจิตอลทั้งจากผู้ผลิตและผู้รับชม นำไปสู่การแข่งขันทางด้านเนื้อหาและเทคนิคต่างๆ ที่ใช้ในการออกอากาศ ระบบสตูดิโอเสมือนจริง (Virtual Studio) จึงเป็นระบบหนึ่งที่ถูกนำมาผสมผสานในการผลิตเนื้อหาที่ช่วยในด้านการนำเสนอให้ผู้ชมสามารถเห็นภาพเสมือนจริงประกอบการอธิบาย ทำให้เข้าถึงเนื้อหาได้ง่ายขึ้น รวมถึงการเพิ่มความน่าสนใจด้วยการใช้เอฟเฟคต่างๆ ที่ไม่สามารถทำในระบบสตูดิโอจริงได้

องค์ประกอบพื้นฐานของระบบสตูดิโอเสมือนจริงแสดงในรูปที่ 1 ประกอบไปด้วยกล้องที่ใช้ในการถ่ายทำฉากเขียวที่จะถูกตัดออก ฉากเสมือนและระบบประมวลผลภาพที่จะทำการแยกฉากเขียวแล้วนำคนเข้าไปประกอบกับภาพฉากเสมือน โดยหลักการแล้วระบบสตูดิโอเสมือนจริงสามารถแยกการประมวลผลเป็น 4 ส่วนหลัก เริ่มต้นที่การรับภาพจากกล้องเพื่อนำมาแยกคนออกจากฉากเขียว หลังจากนั้นภาพฉากเสมือนจริงจะถูกสังเคราะห์ตามตำแหน่งกล้องที่ถูกคำนวณมา ภาพผลลัพธ์จะได้จากการนำภาพคนมารวมกับภาพฉากเสมือน



รูปที่ 1 องค์ประกอบพื้นฐานของระบบสตูดิโอเสมือนจริง

จากการทำงานของระบบสตูดิโอเสมือนจริงจะเห็นได้ว่าการประมวลผลหลักที่สำคัญอย่างหนึ่งคือการแยกฉากเขียว ซึ่งการแยกฉากเขียวไม่เพียงแต่นำไปใช้กับระบบสตูดิโอเสมือนจริง การตัดฉากเขียวยังสามารถนำไปใช้กับงานซ้อนภาพทั่วไปได้ยกตัวอย่างเช่น การรายงานสภาพอากาศ หรือการเปิดรายการสั้นๆ ที่มีการใช้มุมกล้องเพียงมุมเดียว ด้วยประโยชน์ใช้สอยของระบบการแยกฉากเขียวจึงนำมาสู่งานวิจัยนี้ซึ่งมุ่งหวังผลในการผลิตระบบการแยกฉากเขียวที่มีประสิทธิภาพทัดเทียมหรือดีกว่าระบบที่มีจำหน่ายอยู่ในตลาด

สืบเนื่องจากระบบสตูดิโอในปัจจุบันที่เน้นการใช้อุปกรณ์เดิมที่มีอยู่เช่น กล้องวิดีโอ ไฟสตูดิโอ พื้นที่ที่จำกัด เพื่อลดต้นทุนในการผลิตรายการ ทำให้เกิดข้อจำกัดมากมายในการใช้ฉากเขียวประกอบการถ่ายทำกล่าวคือ กล้องวิดีโอมีความละเอียดต่ำหรือมีการรบกวนของภาพ ไฟสตูดิโอไม่เพียงพอหรือปรับความสว่างได้ไม่ละเอียด พื้นที่จำกัดทำให้เกิดการสะท้อนของแสงสีเขียว รวมไปถึงการใช้เทคนิคต่างๆ ในการให้ลวดลายกับฉากเขียวเพื่อเพิ่มความสามารถของระบบสตูดิโอเสมือนจริง ด้วยข้อจำกัดต่างๆ เหล่านี้ทำให้ปัญหาการแยกฉากเขียวไม่ใช่เพียงแค่เลือกสีเขียวสีเดียวแล้วตัดออก แต่ยังต้องมีความสามารถในการเลือกสีเขียวที่จะตัดออกได้อย่างเหมาะสมและมีความทนทานต่อการถูกรบกวนของภาพจากกล้องวิดีโอ

นอกจากนี้ความต้องการสำคัญอย่างหนึ่งสำหรับการออกอากาศคือการทำรายการสด สำหรับความละเอียดมาตรฐานมีขนาดของภาพ 720 x 576 จุดที่ 50 ภาพต่อวินาทีแบบสอดประสาน (interlace) และความละเอียดสูงแบบเต็มมีขนาดของภาพ 1920 x 1080 จุดที่ 50 ภาพต่อวินาทีแบบสอดประสาน ซึ่งการคำนวณแบบลำดับทีละจุดของภาพไม่สามารถทำงานได้ทันเวลาแม้จะเป็นวิธีที่พื้นฐานที่สุดก็ตาม

จากปัญหาที่กล่าวมาทั้งหมด งานวิจัยนี้จึงได้มุ่งเน้นในการพัฒนาขั้นตอนวิธีที่จะใช้ในการแยกคนออกจากฉากหลัง โดยฉากหลังที่สนใจคือฉากหลังที่มีสีเขียวเป็นหลักเนื่องจากมีการใช้งานอย่างแพร่หลาย โดยขั้นตอนวิธีที่พัฒนาจะมีความสามารถในการแยกฉากสีเขียวได้หลายโทน มีความทนทานต่อสภาพแสงและมีความทนทานต่อการถูกรบกวนของภาพจากกล้องวิดีโอ และมีแนวโน้มที่จะมีประสิทธิภาพเพียงพอในการใช้งานแบบทันทีได้ ซึ่งประสิทธิภาพจะเกิดจากการออกแบบขั้นตอนวิธีที่สามารถแก้ปัญหาแบบขนานได้ รวมถึงการใช้หน่วยประมวลผลกราฟิกที่มีความทันสมัยและรองรับกับขั้นตอนวิธีการแก้ปัญหาที่จะนำเสนอ

ผลที่ได้จากงานวิจัยนี้คือขั้นตอนวิธีการแก้ปัญหาที่เกิดขึ้นในกระบวนการแยกฉากเขียวทั้งกระบวนการแยกฉากเขียวและกระบวนการปรับปรุงการแยกฉากเขียว ในส่วนของการแยกฉากเขียวงานวิจัยนี้ได้นำเสนอวิธีการปรับชุดค่ากำหนดแบบละเอียด ในขณะที่วิธีที่มีอยู่ในปัจจุบันใช้การปรับชุดค่ากำหนดเพียงชุดเดียวกับทุกจุดสีในภาพซึ่งอาจจะมีคุณสมบัติในภาพที่แตกต่างกัน จึงอาจไม่เหมาะสมในบางกรณี ดังนั้นเพื่อที่จะให้มีความยืดหยุ่นในการปรับชุดค่ากำหนดเฉพาะกลุ่มที่มีคุณสมบัติในภาพคล้ายๆ กันงานวิจัยนี้จึงนำเสนอวิธีที่เพิ่มความสามารถให้ผู้ใช้สามารถปรับชุดค่ากำหนดเฉพาะกลุ่มได้ โดยการจัดกลุ่มของจุดสีจะใช้ค่าการส่องสว่างมาใช้ในการจัดกลุ่ม สำหรับการปรับปรุงการแยกฉากเขียวในงานวิจัยนี้ได้สนใจกระบวนการย่อยคือ การทำให้ผลการแยกฉากเขียวมีความราบรื่น การแก้การเปื้อนของสีเขียว และการรวมภาพเข้ากับพื้นหลังใหม่ที่ต้องการรวม เหตุที่ผลการแยกฉากเขียวต้องทำให้ราบรื่นเนื่องจากผลที่ได้จากกระบวนการแยกฉากเขียวนั้นมักจะให้ผลลัพธ์ที่มีรอยหยักตามเส้นขอบ อีกทั้งยังมีการเปื้อนของสีเขียวที่สะท้อนมาจากฉากหลัง หากนำผลที่ได้มาทำการรวมภาพโดยตรงจะส่งผลให้ผลลัพธ์ที่ได้เกิดความไม่สมจริงขึ้น ดังนั้นการทำให้ผลการแยกฉากเขียวมีความราบรื่นจึงเป็นสิ่งที่จำเป็นต้องทำ ซึ่งวิธีที่ใช้ในปัจจุบันใช้หลักการของตัวกรองแบบเกาส์เซียนเนื่องจากมีความซับซ้อนต่ำ แต่ผลที่ได้มักจะส่งผลให้เกิดความราบรื่นที่มากเกินไป ในส่วนของการแก้การเปื้อนของสีเขียวนั้น วิธีแก้ปัญหาที่นิยมคือการปรับค่าตามช่องสีซึ่งทำให้เกิดผลข้างเคียงคือความสว่างโดยรวมของภาพลดลงและเกิดการเพี้ยนของสี เพื่อที่จะลดปัญหาดังกล่าว

งานวิจัยนี้จึงได้เสนอวิธีการที่ใช้การกรองหลากหลายวิธีตามความเหมาะสม โดยตัวกรองแบบนำทางถูกนำมาใช้ในการทำให้การแยกฉากเขียวมีความราบรื่นและการแก้การเปื้อนของสีเขียวด้วยเหตุที่ตัวกรองแบบนำทางนั้นสามารถทำให้เกิดความราบรื่นและยังรักษาสภาพความเป็นเส้นขอบของภาพไว้ได้ ตัวกรองแบบเกาส์เซียนถูกนำมาใช้ภายหลังในขั้นตอนการรวมภาพเพื่อกำจัดสิ่งที่ไม่ต้องการตามขอบของการรวมภาพออก

บทคัดย่อ

งานวิจัยนี้สนใจกระบวนการในการแก้ปัญหาที่เกิดขึ้นในกระบวนการแยกฉากเขียวทั้งกระบวนการแยกฉากเขียวและกระบวนการปรับปรุงการแยกฉากเขียว ในส่วนของการแยกฉากเขียวงานวิจัยนี้ได้นำเสนอวิธีการปรับชุดค่ากำหนดแบบละเอียด ในขณะที่วิธีที่มีอยู่ในปัจจุบันใช้การปรับชุดค่ากำหนดเพียงชุดเดียวกับทุกจุดสีในภาพซึ่งอาจจะมีคุณสมบัติในภาพที่แตกต่างกัน จึงอาจไม่เหมาะสมในบางกรณี ดังนั้นเพื่อที่จะให้มีความยืดหยุ่นในการปรับชุดค่ากำหนดเฉพาะกลุ่มที่มีคุณสมบัติในภาพคล้ายๆ กันงานวิจัยนี้จึงนำเสนอวิธีที่เพิ่มความสามารถให้ผู้ใช้สามารถปรับชุดค่ากำหนดเฉพาะกลุ่มได้ โดยการจัดกลุ่มของจุดสีจะใช้ค่าการส่องสว่างมาใช้ในการจัดกลุ่มสำหรับการปรับปรุงการแยกฉากเขียวในงานวิจัยนี้ได้สนใจกระบวนการย่อยคือ การทำให้ผลการแยกฉากเขียวมีความราบรื่น การแก้การเปื้อนของสีเขียว และการรวมภาพเข้ากับพื้นหลังใหม่ที่ต้องการรวม เหตุที่ผลการแยกฉากเขียวต้องทำให้ราบรื่นเนื่องจากผลที่ได้จากกระบวนการแยกฉากเขียวนั้นมักจะให้ผลลัพธ์ที่มีรอยหยักตามเส้นขอบอีกทั้งยังมีการเปื้อนของสีเขียวที่สะท้อนมาจากฉากหลัง หากนำผลที่ได้มาทำการรวมภาพโดยตรงจะส่งผลให้ผลลัพธ์ที่ได้เกิดความไม่สมจริงขึ้น ดังนั้นการทำให้ผลการแยกฉากเขียวมีความราบรื่นจึงเป็นสิ่งที่จำเป็นต้องทำ ซึ่งวิธีที่ใช้ในปัจจุบันใช้หลักการของตัวกรองแบบเกาส์เซียนเนื่องจากมีความซับซ้อนต่ำ แต่ผลที่ได้มักจะส่งผลให้เกิดความราบรื่นที่มากเกินไป ในส่วนของการแก้การเปื้อนของสีเขียวนั้น วิธีแก้ปัญหาที่นิยมคือการปรับค่าตามช่องสีซึ่งทำให้เกิดผลข้างเคียงคือความสว่างโดยรวมของภาพลดลงและเกิดการเพี้ยนของสี เพื่อที่จะลดปัญหาดังกล่าวงานวิจัยนี้จึงได้เสนอวิธีการที่ใช้การกรองหลากหลายวิธีตามความเหมาะสม โดยตัวกรองแบบนำทางถูกนำมาใช้ในการทำให้การแยกฉากเขียวมีความราบรื่นและการแก้การเปื้อนของสีเขียวด้วยเหตุที่ตัวกรองแบบนำทางนั้นสามารถทำให้เกิดความราบรื่นและยังรักษาสภาพความเป็นเส้นขอบของภาพไว้ได้ ตัวกรองแบบเกาส์เซียนถูกนำมาใช้ภายหลังในขั้นตอนการรวมภาพเพื่อกำจัดสิ่งที่ไม่ต้องการตามขอบของการรวมภาพออก

คำสำคัญ: การแยกฉากเขียว, ความต่างของสี, การทำให้การแยกฉากเขียวมีความราบรื่น, การแก้การเปื้อน, การผสมที่เส้นขอบ, ตัวกรองแบบนำทาง, ตัวกรองแบบเกาส์เซียน

Abstract

This work considers all procedures required in the process of a green screen matting which are green screen matte extraction and green screen matte refinement. For the extraction process, this work addresses the problem of fine tuning thresholds of a color difference method for green screen matting. The conventional color difference method uses two thresholds to control a result matte. However, the thresholds are applied to all pixels in an image which sometimes present different structures in the image. In order to provide a flexibility to control each group of pixels that have the same property in an image locally in the property space, this work proposes a method that allows a user to adjust the thresholds associated to each group of pixels that share similar luminance. For the refinement process, this work considers alpha matte smoothing process, green spill suppression process and compositing process. An extracted matte usually has ridges on the edge and exceeding green spilling on the subject. Furthermore, the artifact on the edge of the original green screen image always exposes in the compositing procedure. These lead to the necessity of applying the green screen matte refinement. For simplicity, a popular conventional approach used for alpha matte smoothing is applying the Gaussian filter on the extracted matte but it may produce over-smoothing edge. The existing spill suppression approaches are to clamp the excess green based on the values of the red channel and the blue channel. This causes a hue shift and a brightness drop problem in a final composite. Unspill operation is the other existing approach proposed to relief the brightness drop problem but the hue shift problem remains. In order to resolve these limitations, this work proposes the uses of multiple filter approaches for all procedures. Guided filter is the main filter applied in this work for alpha matte smoothing and suppressing green spill because of its capability of smoothing an image while the details of the image are preserved according to a guidance image. Gaussian filter is taken into account for blending the edge of the matte and a new background because the artifact on the edge of the input image is sometimes needed to be removed.

Keywords: Green screen matte, Color difference, Alpha matte smoothing, Spill suppression, Edge blending, Guided filter, Gaussian filter.

GPU - Based Alpha Matting Using Color Information

Thanathorn Phoka

September 14, 2017

Contents

Chapter 1

Green Screen Matte Extraction

1.1 Introduction

Green screen matting is broadly used in many applications, e.g., video editing and compositing, filmmaking or TV broadcasting. Especially, TV broadcasting has been in our consideration especially when digital technology has been involved. The digital technology allows computer science people to research and develop devices used in broadcasting. Virtual studio is a successful example of using the digital technologies. The two main components of a virtual studio are a 3D render engine and a green screen matting module. The 3D render engine produces a background image according to a virtual camera's view. The matting module obtains a video feed from a camera that subjects are filmed in a green screen. Every single frame from the video feed is then computed to extract the subjects from the background screen. Once the virtual studio has obtained both the rendered background image and the image of the extracted subject, the compositing is then taken place to produce an output image frame. The green screen matting module is important not only for the virtual studio but also a classical video compositing using an image or video feed as a background, e.g., a weather forecast programme.

However, the quality of a result image or a video feed depends on capability of green screen matting. According to various studios used in recording, there exist studios which cannot afford new high technology devices, such as high definition video cameras producing high quality video feed with low video noise or sufficient numbers of dimmable light sources. These limitations affect the quality of green screen matting. From preliminary experiment (not presented in this work) and observation, lighting condition is the most important factor affecting the quality of a result green screen matte. A shadow of a subject will be casted onto a green screen when the subject is closed to the insufficiently lit green screen. In another situation when a subject is not well lit, the subject can be spilled by green color. Overbright lighting can also induce a relatively white spot on a green screen.

In order to suppress the consequences of the lighting problems and obtain a higher quality of green screen matte, this work considers the problem of fine tuning thresholds of a color difference method for green screen matting which is the conventional matting method used in the broadcasting industry since the color difference method is robust for inconsistent lighting. Further, the method is not computationally intensive, it can be computed in real time.

In this work, the color difference method which uses two thresholds to control a green screen matte extraction is extended. One threshold is for determining the boundary of subjects or foreground and the other one is for determining the boundary of green screen color or background. These two thresholds are applied to all pixels in an image which might not be suitable for pixels in different image structures or uneven lighting conditions. For example, low luminance pixels might be a group of pixels presenting hair color which should be determined as a soft edge structure so that the green screen matte of the pixels with low luminance should be extracted by thresholds associated to their group. In another case, when a subject is wearing white clothes which the corresponding pixels are high luminance pixels, the pixels should be determined as a hard edge structure. However, white clothes are easily to be spilled by green color so that a result matte contains transparency. In order to get rid of the transparency in a hard edge structure, we should provide and adjust the thresholds for the high luminance pixels. Not only the problem with a subject, this work also considers the problem of an uneven lit green screen that different green screen pixels with different lighting conditions should be handled by their own thresholds.

The proposed method uses a luminance histogram produced by counting how many pixels in an image are at each brightness varying from 0 - 1. Peaks or in other words local maxima are extracted to be used to initialize thresholds for each maximal luminance. Once the thresholds for the maxima have been obtained, the thresholds for each step of luminance are interpolated using Catmull-Rom splines [1]. The thresholds associated with different luminances are then computed for green screen matting for each pixel in the image according to its brightness. The method also allows a user to fine tune thresholds by adjusting the provided control points or adding more control points of the splines for a better green screen matte.

1.2 Related Work

Matting can be divided into two categories which are alpha matting (or natural image matting) and green screen matting. A difference of these two categories is prior knowledge about foreground and background. Alpha matting uses a trimap to define foreground, background and unknown regions. The unknown region will be calculated for alpha value from the known foreground and background regions. Green screen matting uses a main color as a background color while the others are considered as foreground.

Alpha matting refers to the problem of decomposing an image or a video frame into foreground and background which an observed image can be modeled as a convex combination

$$C(p) = \alpha(p)F(p) + (1 - \alpha(p))B(p), \tag{1.1}$$

where $C(p)$ is the color of a pixel p, $F(p)$ is the foreground layer, $B(p)$ is the background layer and $\alpha(p)$ is opacity or alpha matte. From the equation, we can see that $F(p)$, $B(p)$ and $\alpha(p)$ are unknown values which makes matting be a highly ill-posed problem. Hence, the problem is constrained by additional information such as trimap or scribbles defining the foreground, background and unknown regions.

Existing approaches can be divided into two categories, propagation-based approaches and color sampling based approaches. Propagation-based approaches interpolate the unknown alpha values from the known regions assuming correlation between the neighboring pixels and use their affinities in the propagation [2, 3, 4]. Color sampling-based approaches collect a set of known foreground and background samples to estimate alpha values of unknown pixels from their best sample pair [5, 6, 7, 8, 9, 10]. In [11], the two approaches are combined by casting the matting problem as an energy minimization problem in which the color sampling procedure forms the data term and the alpha propagation component formes the smoothness term. In [12], propagation and sampling are adaptively combined according to smoothness of regions to estimate alpha matte.

The green screen matting problem was analyzed in [13]. A simple solution to the matting problem does exist when constraints are taken into account. In the general case, a unique solution can be obtained if the foreground subject is captured against two backgrounds that every pixel is different. However, this approach is applicable only for images that non moving subjects can be captured twice on two different backgrounds.

In general, two classical approaches, color difference method and chromakey method are bases of state-of-the-art devices, the Ultimatte® and the Primatte®, used in video editing [14]. The first approach calculates for each pixel the difference between the green color channel value $C_G(p)$ and the maximum value of the remaining channels $C_R(p), C_B(p)$. The difference value is then used to determine the alpha value of the pixel as shown in the following relation.

$$\alpha(p) \propto \max(C_R(p), C_B(p)) - C_G(p) \tag{1.2}$$

The second approach uses the distance between a reference color of a screen $C_{Ref}(p)$ and the color of a pixel $C(p)$ to calculate the alpha value as shown in the following relation.

$$\alpha(p) \propto \|C_{Ref}(p) - C(p)\| \tag{1.3}$$

These two approaches have low complexity and can utilize parallelism of GPU (graphic processing unit) so that they can compute a matte in real time. This work will extend the color difference method which is a practical approach used in the current broadcasting industry.

1.3 Color Difference

Alpha value is used to describe green screen matte. An alpha value of a pixel determines opacity of the pixel in an alpha matte. A pixel with 0 alpha value is defined as a background pixel while a pixel with 1 alpha value is defined as a foreground pixel. A pixel with alpha value in the range $(0, 1)$ is defined as a transparent pixel.

In order to calculate the alpha value $\alpha(p)$ of a pixel p against a green screen, lets define color difference $D(p)$ as follows

$$D(p) = C_G(p) - \max(C_R(p), C_B(p)) \tag{1.4}$$

$D(p)$ is then used to calculate $\alpha(p)$ by taking two parameters $gain_D$ and b_D into the following equation.

$$\alpha(p) = -gain_D D(p) + b_D \tag{1.5}$$

where $gain_D$ and b_D are used to control a color curve graph transforming $D(p)$ into $\alpha(p)$ as described in [14].

More intuitively, equation 1.5 can be represented in the following form

$$\alpha(p) = 1 - saturate((D(p) - t_{min})/(t_{max} - t_{min})) \tag{1.6}$$

where *saturate* is a function that bounds an input value into the range $[0, 1]$. $gain_D$ and b_D can be derived from t_{min} and t_{max} by

$$gain_D = \frac{1}{t_{max} - t_{min}} \tag{1.7}$$

$$b_D = \frac{t_{max}}{t_{max} - t_{min}}. \tag{1.8}$$

This work focuses on adjusting t_{min} and t_{max} of equation 1.6 since they intuitively induce $\alpha(p)$ by

$$\alpha(p) = \begin{cases} 1 \text{ when } D(p) < t_{min} \\ 0 \text{ when } D(p) > t_{max} \\ 1 - \frac{(D(p) - t_{min})}{(t_{max} - t_{min})}, \text{ otherwise.} \end{cases} \tag{1.9}$$

From the above calculations, t_{min} and t_{max} are thresholds determining that p is a foreground pixel, a background pixel or a transparency pixel according to the value $D(p)$.

Once t_{min} and t_{max} are obtained from user adjustment, they are used as the global thresholds that applied to all pixels in an image. This might leave some problems, for example, shadow casting on a green screen, uneven lighting condition on a green screen or green color spilling. In order to suppress consequences of the problems, this work extends the color difference method by providing more controls for fine tuning matte result which will be described in the next section.

1.4 Extended Color Difference

From observations, the most problems occur because of lighting. In the case of uneven lighting on a green screen, t_{max} has to be decreased to cover all green screen pixels. This is undesirable since it makes soft edge matte become sharper. Over brightness, under brightness, shadow casting on a green screen lead to the same effect by decreasing color difference $D(p)$. This possibly forces t_{max} to be drastically decreased and sometimes leads t_{min} to be decreased as well. Green color spilling is also a problem from insufficient lighting on a subject. In this situation, a green screen matte leaves undesirable transparency on some pixels that should be opaque. In order to suppress this consequence, t_{min} has to be increased so that all spilled pixels are covered.

According to the above observations, adjusting thresholds related to luminance is reasonable. We can find uses of luminance in many applications and researches. For example, color correction which is the process of adjusting raw image to look more realistic. It divides adjustments into three luminance ranges which are shadows, midtones and highlights. Luminance also plays an important role in face recognition [15, 16] and face detection [17] in a form of grey scale image.

The overall processes of the method are presented in Fig. 1.1. An input color image consisting of subjects and a green screen is converted to the luminance image by converting a color pixel p to $L(p)$ using the following equation.

$$L(p) = 0.2126C_R(p) + 0.7152C_G(p) + 0.0722C_B(p) \tag{1.10}$$

The luminance image is then computed for a luminance histogram in the range $[0, 1]$ where the range is discretized into 256 bins. The histogram is constructed by counting how many pixels in the luminance image are at the brightness corresponding to each bin. Once the histogram has been obtained, it is smoothed using a Gaussian blur to avoid local maxima from histogram fluctuation. In the experiment,kernel size is set to 9 for a Gaussian blur. Local maxima are extracted by tracing the smoothed histogram for bins that contain peaks of the smoothed histogram. Let the local maxima be defined by $\{lm_1, ..., lm_n\}$ where n is the number of local maxima.

The input color image is also computed for a D image by calculating all pixels for color difference value using equation 1.4. For each local maximum, the D image is then used for 2 cluster k-means clustering which finds the means of two groups according to D values of pixels that their luminances are similar to the luminance of the local maximum. All couples of the means associated to all local maxima are denoted by $\{(c_{min,1}, c_{max,1}), ..., (c_{min,n}, c_{max,n})\}$ where $c_{min,i} \leq c_{max,i}$ for $i = 1, ..., n$.

Before initializing couples of thresholds for all local maxima, a user is needed to put some effort for adjusting a threshold t_{bin} to produce a good binary matte. A good binary matte means that most green screen pixels are classified as background or $\alpha(p) = 0$ and most subject pixels are classified as foreground or $\alpha(p) = 1$. The threshold t_{bin} is applied in the classification by

$$\alpha(p) = \begin{cases} 1 \text{ when } D(p) < t_{bin} \\ 0 \text{ when } D(p) >= t_{bin}. \end{cases} \tag{1.11}$$

The couple of thresholds for each local maximum lm_i is now ready to be initialized. A couple of thresholds $t_{min,i}$ and $t_{max,i}$ are bounded by t_{bin} that is $t_{min,i} = c_{min,i}$ and $t_{max,i} = c_{max,i}$ by default. In the case when $t_{max,i} < t_{bin}$, $t_{min,i} = t_{max,i}$ and $t_{max,i} = t_{bin}$ are performed. With the similar condition when $t_{min,i} > t_{bin}$, $t_{max,i} = t_{min,i}$ and $t_{min,i} = t_{bin}$. These bounding conditions are applied since if we let $t_{min,i} = c_{min,i}$ and $t_{max,i} = c_{max,i}$ in the case that $t_{max,i} < t_{bin}$, a pixel p that $D(p) < t_{bin}$ and $D(p) > t_{max}$ will be determined as a foreground pixel while it would rather be a transparent pixel or a background pixel according to t_{bin}. In the other case when $t_{min,i} > t_{bin}$, a pixel p that $D(p) > t_{bin}$ and $D(p) < t_{min}$ will be determined as a foreground pixel. These cause an undesirable matte result so that the bounding condition are reasonably performed.

However, the obtained $t_{min,i}$ and $t_{max,i}$ still cause a transparency problem as illustrated in Fig. 1.2. Let us consider back to the k-means clustering. An obtained mean is not suitable to be directly used as a threshold since when it represents a $t_{min,i}$ for foreground pixels, a pixel p that is a member of the same group of the mean and $D(p) > t_{min,i}$ will be defined as a transparency pixel but sometimes it is needed to be a foreground pixel or very low transparency pixel. The same problem also occurs for $t_{max,i}$. In order to avoid too many transparent pixels, the method squeezes $t_{min,i}$ and $t_{max,i}$ closer to t_{bin} by the following linear interpolations.

$$t_{min,i} = t_{bin} + gain_s(t_{min} - t_{bin}) \tag{1.12}$$

$$t_{max,i} = t_{bin} + gain_s(t_{max} - t_{bin}) \tag{1.13}$$

where $gain_s$ is a user defined parameter in the range $[0, 1]$. In the experiment, $gain_s = 0.33$ is assigned.

Once sets of thresholds $\{t_{min,1}, ..., t_{min,n}\}$ and $\{t_{max,1}, ..., t_{max,n}\}$ have been obtained, they are used as control points for two Catmull-Rom splines, one is for interpolating $\{t_{min,b}$ and the other one is for interpolating $\{t_{max,b}$ where b is a level of luminance and $b \in [0, 1]$ such that the range $[0, 1]$ is discretized into 256 levels. At this point, the method provides two user defined parameters which are t_{bin} and $gain_s$. Moreover, the method allows a user to fine tune more thresholds by adjusting or adding more control points $\{r_{min,1}, ..., r_{min,k}\}$ and $\{r_{max,1}, ..., r_{max,l}\}$ where k and l are the numbers of control points of the two splines. The splines then turn to be controlled by user defined control points and can be used to refine a matte result until a desired one is acquired.

1.5 Experimental Result

In the preliminary experiments, all input images are full definition with resolution of 1920×1080. All experiments are run on a desktop computer with 3.4 GHz CPU and 16 GB ram. The video card is NVIDIA Geforce GTX 680 with 2 GB ram. An illustration input image is shown in Fig. 1.3(a). The threshold t_{bin} is adjusted for a binary matte resulted in Fig. 1.3(b). The matte produced from the set of initial thresholds according to the threshold t_{bin} is shown in Fig. 1.3(c) and Fig. 1.3(d) shows its composition with a blue background. Fig. 1.3(e) and (f) present fine tuned results. The sets of thresholds $\{r_{min,1}, ..., r_{min,k}\}$ and $\{r_{max,1}, ..., r_{max,l}\}$ used for fine tuning are as appeared in the bottommost window containing two splines.

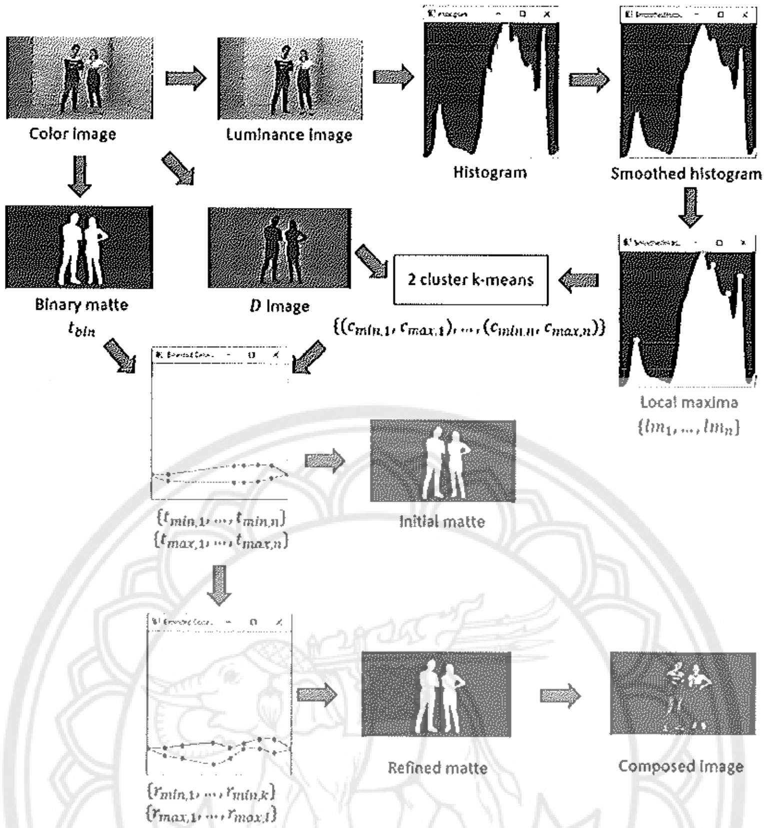


Figure 1.1: The processes of the proposed method and illustrations

This work compares the proposed method with the conventional color difference method, the results of which are shown in Fig. 1.3(g) and (h). The thresholds of the color difference are adjusted to result a matte that mainly covers foreground detail. As we can see from the results of the color difference method, there exist pixels that should be classified as background but they appear as foreground because they are casted by shadow. The method handles this consequence quite well as we can see from Fig. 1.3(c), some undesirable foreground pixels become either background pixels or transparency pixels. However, there still exist some transparency pixels that should be foreground pixels because of green color spilling on white cloths. A result can be improved by adding more control points and adjust them until a good matte in Fig. 1.3(e) is obtained.

Not only green color spilling, but fine tuning can also handle with trans-

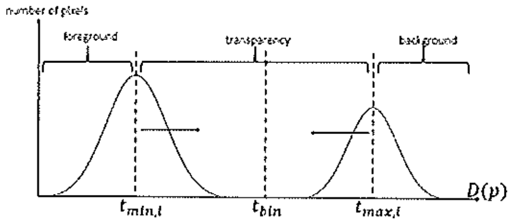


Figure 1.2: Transparency problem

parency on a matte. Fig. 1.4 presents the results focusing on the hair of the lady. Fig. 1.4(a), (b) and (c) are from initialized control points, fine tuned control points and the color difference method, respectively. Even just the result from the initialized control points is better than from the color difference method.

Running time of the color difference method, the extended color difference method and control point initialization are also measured. They take 0.138 ms, 0.385 ms and 139.89 ms on average for the input image in Fig. 1.3(a). In general, running time of a control point initialization is varied according to statistic of luminances appearing on an input image but it does not exceed 1 second so that the threshold initialization process works with the extended color difference method smoothly in real time.

1.6 Conclusion

This work has proposed an extension of the color difference method for green screen matting problem. The method allows a user to adjust the thresholds associated to each group of pixels sharing similar luminance. Pixels with similar luminance are clustered into the same bin of a histogram. The histogram is then smoothed and local maxima extracted from the smoothed histogram will be used to initialize the thresholds associated to the groups corresponding to the local maxima. The method applies 2 cluster k-means to find two means of similar color difference value pixels that share similar luminance for initializing such thresholds. The preliminary results show that the method works better even just apply initialized thresholds. The method also provides fine tuning for a better matte result by putting a little user's effort.

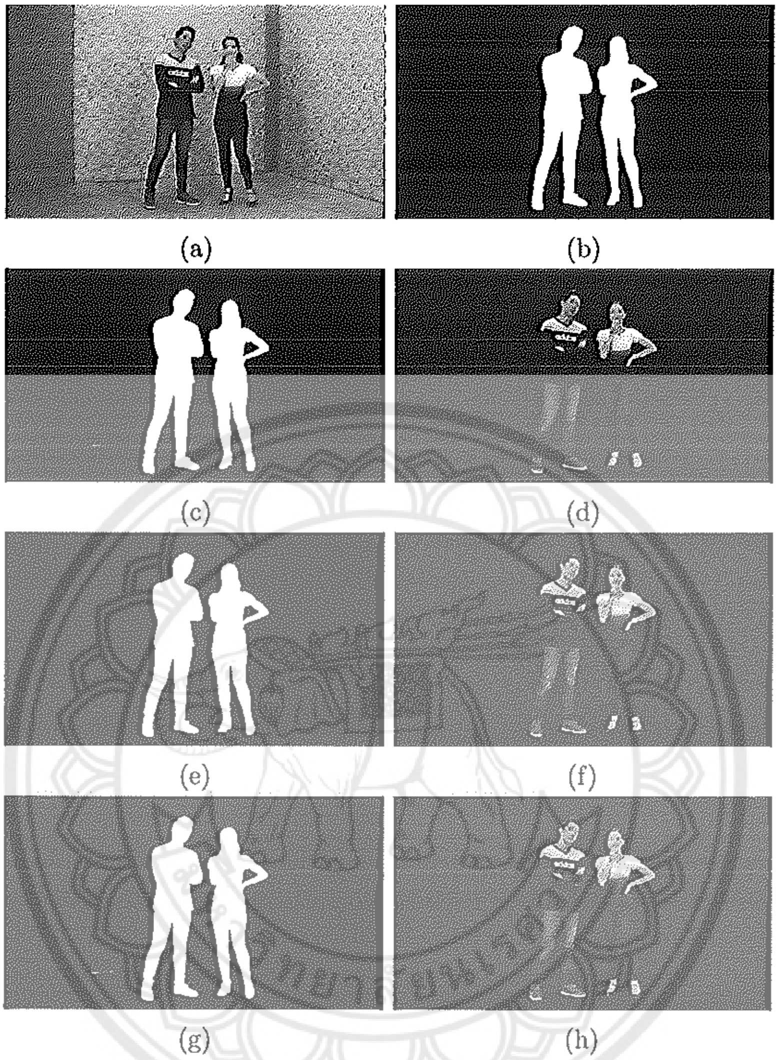


Figure 1.3: The compared results

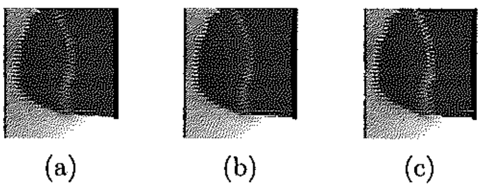


Figure 1.4: The results zoomed in on hair

Chapter 2

Green Screen Matte Refinement

2.1 Introduction

Digital compositing is the digital process that adjusts and combines multiple images to produce a final image. An important procedure in digital compositing is filming a subject with a green screen plate and extracting the subject from the screen to be used in a compositing. The process of extracting the subject from the green screen is referred to as green screen matting or matting while the resulting extraction is referred to as a green screen matte or matte consisting of the foreground, background and alpha matte. Once a matte is acquired, it is then combined with a new background to produce a final composition. However, directly composting the matte may cause a nonnatural composition.

For an input example in Figure 2.1(A), the final composition with the proposed refinement is shown in Figure 2.1(B). Let us look in more details. The zoom-in compositions without any refinement, with smoothing, with spill suppression and with edge blending are shown in Figure 2.1(C - F), respectively. We can see from Figure 2.1(C) that the unrefined composition has ridges at the edge and looks unnatural because of green spill. Therefore, the alpha matte smoothing is usually needed to remove ridges on the edge of the extracted matte.

A main defect of the input image is from improper lighting. It may cause problems both on the green screen and the subject. A matte suffers from improper lighting especially when a subject is underlit since scattering green light will shine on the subject as appeared on the left side of the face in Figure 2.1(A). More clearly, the composition without spill suppression in

Figure 2.1(D) expresses a low quality composition with green glowing on the edge of the subject and green spill on the face. This leads to a necessity of applying spill suppression on the subject.

The conventional spill suppressions remove exceeding green or compensate by increasing blue and red values. The approaches globally adjust red, green and blue values that may cause brightness drop and hue shift artifacts as stated in [14]. The image in Figure 2.2(A) presents a test plate consisting of three strips which present skin tone, yellow and light blue. A modified test plate by filling green gradients is in Figure 2.2(B). The results from clipping exceeding green by red value and blue value are presented in Figure 2.2(C - D). They are severely affected by brightness drop and hue shift and still leave relative green on the skin tone strip in Figure 2.2(C) and the light blue strip in Figure 2.2(D). The unspill approach which concurrently decreases green and increases blue and red values can avoid the brightness drop but the hue shift still maintains as shown in Figure 2.2(E).

The main contribution of this work is to propose the uses of multiple filters for matte refinement especially in the spill suppression procedure that avoids both brightness drop and hue shift artifacts by applying an iterative guided filter for diffusing proper differences between the green value and the blue and red values. A result from the proposed approach indicating a strong point compared to the existing approaches is presented in Figure 2.2(F) that is almost exactly the same as the original test plate. However, Figure 2.1(E) presents a result of the proposed spill suppression, the result looks more natural but still leaves light gray glowing on the edge of the subject. This leads to another contribution of this work to propose a heuristic for edge blending. Brightness of a new background is used as a constraint for the edge blending. Gaussian filter is then applied to generate weights for the edge blending. A result from the edge blending is shown in Figure 2.1(F). It decreases the glow artifact and produces a high quality final composition.

2.2 Related Work

Three main topics about matting, filtering and spilling suppression will be discussed in this section. Generally, matting is the process of extracting a subject from a natural background image to produce a matte in the form of alpha image, foreground and background color image. Before compositing the matte with a new background, the alpha matte smoothing process is usually needed to remove ridges on the edge of the matte. Filtering is then taken into account for this task. However, the green screen may leave some artifacts on the subject by spilling green color on them. The spill suppression

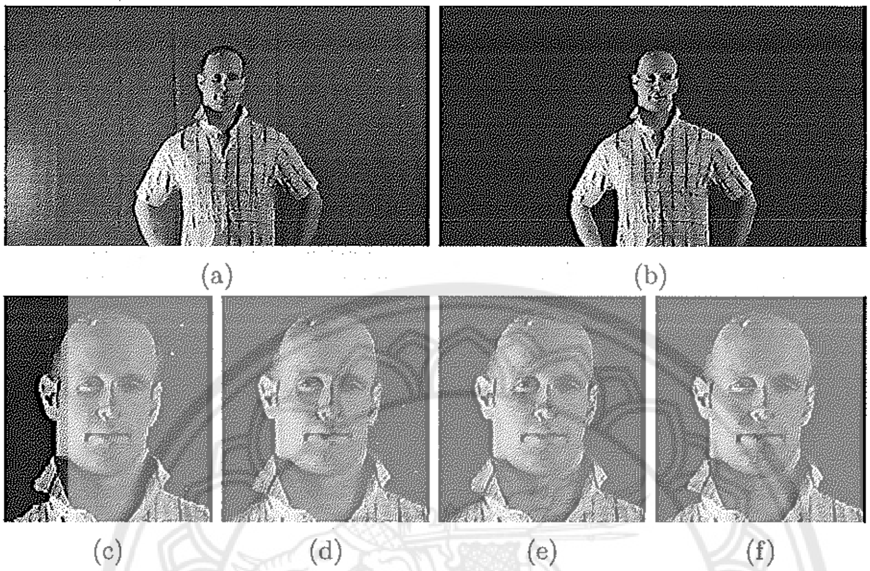


Figure 2.1: Refinement procedures

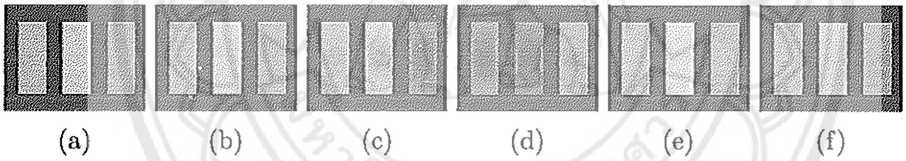


Figure 2.2: Spill suppression artifacts

is applied to remove the unwanted green from the foreground subject.

2.2.1 Matting

Given an image, matting is the process that extracts a foreground subject from a background. It can be modeled as a convex combination of foreground and background color as

$$C_i = \alpha_i F_i + (1 - \alpha_i) B_i, \tag{2.1}$$

where C_i is the input color, F_i is the foreground color, B_i is the background color and α_i is the opacity or blending of a pixel i. All these values of all pixels in an image is called a matte and let the opacity be called an alpha matte. From the equation, F_i, B_i and α_i have 7 unknown values but there are only 3 linear constraints. This makes matting be an ill-posed problem.

Natural image matting needs constraints to compute a result matte. Additional information such as trimap or scribbles determining the foreground, background and unknown regions has to be defined beforehand. There are two main approaches solving the natural image matting. Propagation-based approaches [2, 3, 4] and color sampling-based approaches [5, 6, 7, 8, 10].

In this work, the background color is assumed to be a green color and an extracted matte is given. The analysis of the green screen matting problem was presented in [13]. In practice, color difference and chroma-key are the fundamentals of commercial softwares for the green screen matting used in digital compositing [14]. Recently, an extension of the color difference was proposed to allow a user to fine tune color difference thresholds based on luminances [18].

2.2.2 Filters

Filtering is applied for smoothing an alpha matte. Box filter and Gaussian filter are commonly used because of their simplicity but they may decay the edge detail of the alpha matte. Bilateral filter [19] was proposed as an edge preserved filter. However, an alpha matte does not present the detail of an image. Joint bilateral filter [20] computes weights from another guidance image that is the color input image in this work. The main drawback of the joint bilateral filter is that it may suffer from gradient reversal artifacts when neighboring pixels of a smoothed pixel are too different. Nonlocal filter [21] considers similarity of square patches to compute weights. A use of the nonlocal filter for an alpha matte smoothing was proposed in [22]. The filter can smooth a matte while preserving edge detail but the complexity of the algorithm is $O(Nr^2m^2)$ where m is the radius of a square patch used in weight computation.

Guided filter [23] is another edge preserved filtering with $O(N)$ complexity. Assuming a local linear model between the guidance image I and the filtering output image q, the guided filter smooths the input image p while preserving edge information transferred from the guidance image I:

$$q_i = a_k I_i + b_k, \forall i \in \omega_k, \tag{2.2}$$

where k is the center of a window ω_k, the radius of which is defined by r, (a_k, b_k) are constant linear coefficients in each ω_k. Finding the solution of the coefficients (a_k, b_k) can be casted into a minimization problem solved by the ridge regression and the solution is given by:

$$a_k = \frac{\frac{1}{|\omega|}\Sigma_{i\in\omega_k} I_i p_i - \mu_k \bar{p}_k}{\sigma_k^2 + \epsilon}, \tag{2.3}$$

$$b_k = \bar{p}_k - a_k \mu_k, \tag{2.4}$$

where μ_k and σ^2 are the mean and variance of I in ω_k, $\bar{p}_k$ is the mean of p in ω_k and ϵ is a regularization parameter.

A pixel i gets involved in all windows containing it therefore q_i is computed from the averages $\bar{a}_i = \frac{1}{|\omega|}\Sigma_{k\in\omega_i} a_k$ and $\bar{b}_i = \frac{1}{|\omega|}\Sigma_{k\in\omega_i} b_k$ as:

$$q_i = \bar{a}_i I_i + \bar{b}_i. \tag{2.5}$$

There are two parameters that can be adjusted, r and ϵ. As presented in [23], r and ϵ are corresponding to the variances of spatial and range values used in the bilateral filter, respectively. This work applies the guided filter for alpha matte smoothing and spill suppression which will be described in Section 2.3.

2.2.3 Spill suppression

Spill suppression is applied to handle with exceeding green from a matte based on some constraints. Some simple but practical constraints are described in [14] which are limiting the green value by the blue or red value as follows:

$$C_{i,G} = min(\beta C_{i,R} + \gamma C_{i,B}, C_{i,G}), \tag{2.6}$$

where β and γ are user-defined parameters. The green value is trimmed by the proportion of the red value when $\beta > 0$ and $\gamma = 0$ or of the blue value vice versa. The parameters can be freely adjusted by a user not restricted to the previous conditions to produce the best appearance. Applying this constraint is referred to as despill operation. However, the despill operation has two main artifacts which are brightness drop and hue shift. The brightness drop comes from decreasing the most influential color channel on brightness, the green value. The hue shift is from adjusting the green channel only. Modifying one channel in the RGB space clearly affects hue change.

The other practical spill suppression described in [14] is called unspill operation. The unspill operation implicitly provides brightness adjusting by allowing a user to decrease the green value and increase the red and blue values concurrently as follows:

$$C_{i,c} = C_{i,c} + g_c(C_{i,G} - \frac{C_{i,R} + C_{i,B}}{2}), c \in \{R, G, B\}, \tag{2.7}$$

where g_c are user-defined gains for each channel, g_R and g_B are normally positive while g_G is negative. Even though the unspill operation reliefs the brightness drop but it still suffers from the hue shift since all pixels share the same gain adjustments.

2.3 Matt Refinement

This section will describe all the procedures necessary to refine an obtained matte for a good final composition in order.

2.3.1 Alpha matte smoothing

Alpha matte smoothing is the first procedure applied to the extracted matte to refine the edge of the matte. As stated in [23], the guided filter can be used for guided feathering which is the alpha matte smoothing in this work. The guided filter is directly applied by assigning I to be the input color image, p is an alpha matte obtained from a matte extraction procedure and q is the smoothed alpha matte. The intuition of this process is to smooth the alpha matte while preserving edge detail appearing in the input color image to avoid oversmoothing.

2.3.2 Spill suppression

This work remodels the spill suppression by considering the differences between the green value and the remaining channel values. Once such differences are obtained, the new values of the red and blue channels are from the following:

$$C_{i,c} = C_{i,G} + D_{i,c}, c \in \{R, B\}, \tag{2.8}$$

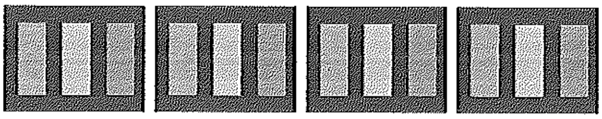
where $D_{i,R} = C_{i,R} - C_{i,G}$ is the difference between the red and the green values and $D_{i,B} = C_{i,B} - C_{i,G}$ is the difference between the blue and the green values. Let such differences be called spill differences. The next stage is to find spill differences of each pixel that retain the appearance of the original input image, remove green spill but induce less artifact than the existing approaches in both the hue shift and the brightness drop.

Assuming that pixels spilled by green is the minority, the guided filter is again used in this stage not only for smoothing but also for propagating spill differences. In theory, expanding r is not sufficient to propagate the spill differences and may smooth out edge detail so that an iterative guided filter is then applied instead. The guided filter in Equation 2.5 is reformulated as follows:

$$D_{i,c,t} = \bar{a}_{i,c,t} I_{i,t} + \bar{b}_{i,c,t}, \tag{2.9}$$

where the subscript t is the iteration of applying the guided filter. The color image I_t is updated for the next iteration by:

$$I_{i,c,t+1} = \max(I_{i,G,t} + D_{i,c,t}, I_{i,c,t}), \tag{2.10}$$



(a) (b) (c) (d)

Figure 2.3: An evolution of applying the iterative guided filter

where I_0 is the composition of the smoothed matte from the alpha matte smoothing and a new background and $D_{i,c,0} = I_{i,c,0} - I_{i,G,0}$. The max function is applied to obstruct relative green diffusion. The value of $D_{i,c,t}$ that induces lower values of red or blue channels has to be blocked from the diffusion. The advantage of performing spill suppression on the composition is that it concurrently decreases the effect from the green background and increases the effect from the new background. The intuition of the iterative guided filter is that for each iteration the spill differences are smoothed according to the edge detail transferred from the composition. The composition is then updated by the smoothed spill differences and used in the next iteration. With proper adjustments of r, ϵ and t, the iterative guided filter can produce high quality spill suppressed results presented in Figure 2.3. The test plate is then again used to demonstrate an evolution of applying the iterative guided filter. The values of r and ϵ are assigned to be 20 and 0.05 while t is varied from $\{1, 5, 10, 15\}$ and the associated results are expressed in Figure 2.3(A - D), respectively. As we can see from Figure 2.3(A), the original guided filter cannot handle with this situation but the iterative guided filter produces better results through iterations.

2.3.3 Edge blending

The proposed spill suppression works well in regular situations but not under some conditions such as edge enhancement from an image capturing device or ringing artifact from an image compression. These artifacts will leave glowing effect on the edge of a composition when the brightness of the original green background and of the new background is extremely different. The other artifact is from the spill suppression model in Equation 2.8 that $D_{i,c,t}$ is often greater than $D_{i,c,0}$ in the regions that are spilled by green. This makes the overall foreground of the spill suppressed composition look a little brighter.

In order to alleviate the artifacts, this work proposes a heuristic for edge blending based on the brightness of the original foreground and the new background. The proposed heuristic not only blends the edge but also constrains the brightness of a spill suppressed foreground back to of the original one. The first step is to blur the alpha matte obtained from the alpha matte smoothing procedure from the border to the inner side using the Gaussian filter and let the blurred alpha matte be denoted by α_b. Secondly, the input color image, new background image are converted from the RGB space to the YUV space and let S and T denote the Y channels of the input image and new background image, respectively. The brightness U of the output from this process is casted into the following convex combination:

$$U_i = \alpha_{b,i} S_i + (1 - \alpha_{b,i}) T_i. \tag{2.11}$$

However, green spill naturally induces the brightness drop so that the brighter foreground of the spill suppressed composition is sometimes preferred. In this situation, S can be replaced by the Y channel of the spill suppressed composition. Once U has been obtained, it is merged with the UV channels of the spill suppressed composition. The result is then recomposited with the new background to produce the final composition.

2.4 Experimental Result

For all experiments, the images in Figure 2.1(A) and 2.4(A) are from the internet where their resolution are 640×360 and 1440×1080, respectively. The machine used in the experiments consists of 3.4 GHz CPU and 16 GB ram. The proposed procedures are implemented in C++ language with OpenCV library. The results are compared among the proposed procedures. As stated in Section 2.1 that all proposed procedures are needed to produce a high quality composition, Figure 2.4(C-F) show the zoom-in versions of the compositions without refinement, with guided filter applying to the alpha matte, with spill suppression and with edge blending, respectively. The results look better for every single procedure applied until the last step that produces the final composition expressed in Figure 2.4(B).

The spill suppression approach is evaluated by varying the values of the number of iterations t and the radius r while the regularization parameter ϵ is treated as a constant with the value 0.05 to avoid over propagation. The zoom-in results from adjusting these parameters are presented in Figure 2.4(G). The compositions look more satisfying by properly removing excess green when t and r are increased. However, the running time is increased according to the number iterations performing the guided filter as expressed in Table 2.1 - 2.3.

The running times in Table 2.1 - 2.3 measured in the spill suppression procedure show that adjusting r does not affect the running time but increasing t does. The running time goes up linearly according to t. This complies with the complexity analysis and experiment in [23] that the guided filter runs in $O(N)$ time complexity. An image with resolution 1920×1080 is also considered in the running time measurement and the comparisons reflect that the number of pixels also influences the running time. The running times of the alpha matte smoothing for the 640×360, 1440×1080 and 1920×1080 inputs are $0.05, 0.36$ and 0.48 and the edge blending procedure takes $0.04, 0.11$ and 0.13, respectively which indicate the linear relations between the procedures and the number of pixels.

Furthermore, the spill suppression approach proposed in this work aims to resolve the hue shift and brightness drop artifacts. As we can see from Figure 2.4(A), the three color blobs are intentionally added into the image, skin tone, yellow and light blue colors. The final composition expresses that the proposes approach can avoid both artifacts while removing the unwanted green.

Table 2.1: Running times in seconds

t \ r	640 × 360			
	5	10	15	20
1	0.059	0.059	0.059	0.065
2	0.12	0.125	0.129	0.14
3	0.183	0.185	0.197	0.206
4	0.236	0.269	0.273	0.282
5	0.311	0.325	0.344	0.354

Table 2.2: Running times in seconds

t \ r	1440 × 1080			
	5	10	15	20
1	0.775	0.775	0.793	0.797
2	1.514	1.577	1.606	1.622
3	2.307	2.377	2.436	2.42
4	3.116	3.209	3.301	3.22
5	3.948	4.013	4.075	4.008

Table 2.3: Running times in seconds

t \ r	1920 × 1080			
	5	10	15	20
1	0.932	0.917	0.94	0.931
2	1.962	1.933	1.918	1.892
3	3.026	3.021	2.926	2.866
4	4.161	4.097	3.917	3.848
5	5.322	5.2	4.943	4.872

2.5 Conclusions

This work has proposed the procedures of a green screen matte refinement based on the guided filter and the Gaussian filter. The guided filter is the main filter used in the alpha matte smoothing and the proposed iterative version is applied in the spill suppression procedure. The edge blending step applies the Gaussian filter to produce parameters for the convex combination in the Y channel of the YUV space of the foreground image and a new background image. The experimental results show that the proposed approaches can produce high quality compositions but the main drawback is the running time that linearly depends on the number of iterations applying the guided filter. However, the proposed approaches can be further implemented and optimized on a GPU because of their parallelism which is an extension of this work.

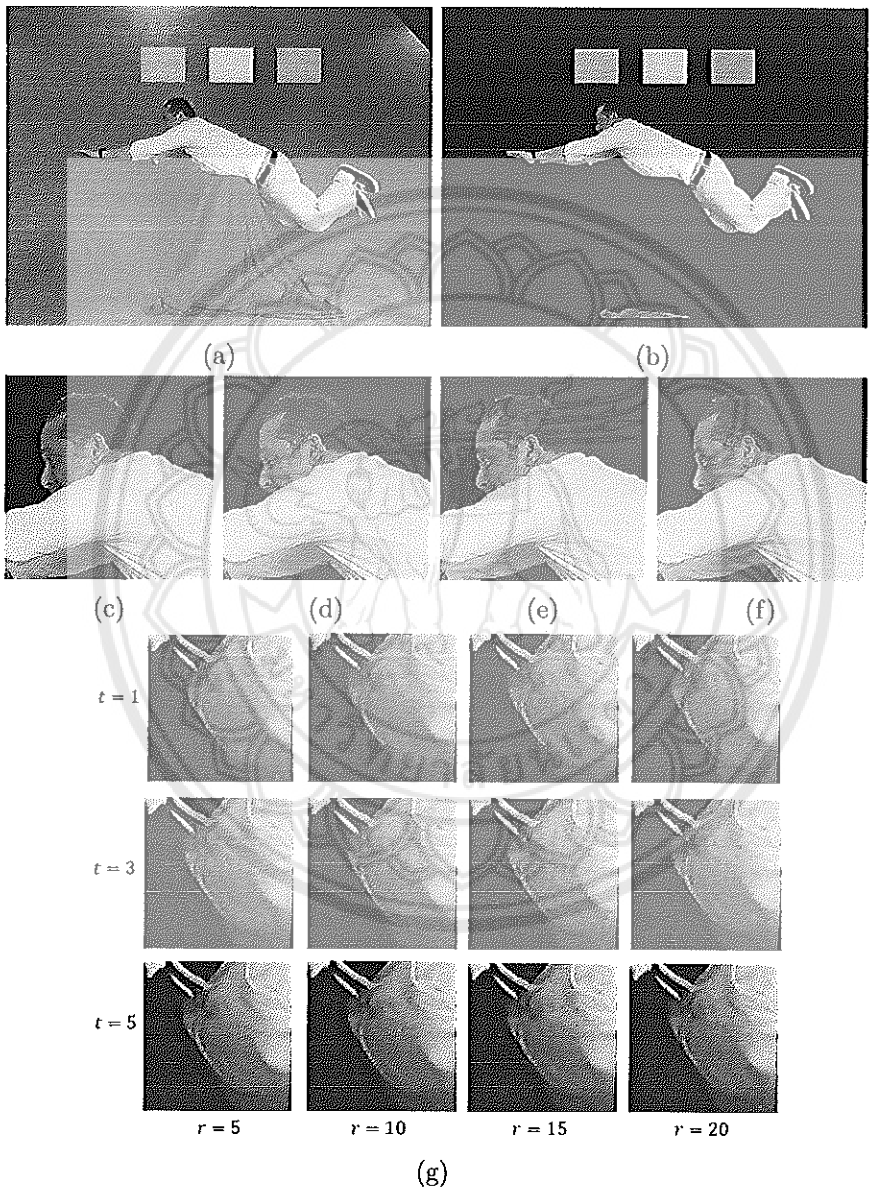


Figure 2.4: Experimental results

Bibliography

[1] Edwin Catmull and Raphael Rom. A class of local interpolating splines. In ROBERT E. BARNHILL and RICHARD F. RIESENFELD, editors, *Computer Aided Geometric Design*, pages 317–326. Academic Press, 1974.

[2] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):228–242, 2008.

[3] P. G. Lee and Y. Wu. Nonlocal matting. In *2011 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2193–2200. IEEE, 2011.

[4] Qifeng Chen, Dingzeyu Li, and Chi-Keung Tang. Knn matting. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 869–876. IEEE, 2012.

[5] Eduardo S. L. Gastal and Manuel M. Oliveira. Shared sampling for real-time alpha matting. *Computer Graphics Forum*, 29(2):575–584, May 2010. Proceedings of Eurographics.

[6] Kaiming He, C. Rhemann, C. Rother, Xiaoou Tang, and Jian Sun. A global sampling method for alpha matting. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2049–2056, June 2011.

[7] Deepu Rajan. Weighted color and texture sample selection for image matting. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 718–725, Washington, DC, USA, 2012. IEEE Computer Society.

[8] Ehsan Shahrian, Deepu Rajan, Brian Price, and Scott Cohen. Improving image matting using comprehensive sampling sets. In *Proceedings*

of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '13, pages 636–643, Washington, DC, USA, 2013. IEEE Computer Society.

[9] Jubin Johnson, Deepu Rajan, and Hisham Cholakkal. Sparse codes as alpha matte. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[10] Wenyi Wang and Jiying Zhao. Robust image chroma-keying: A quadmap approach based on global sampling and local affinity. *Broadcasting, IEEE Transactions on*, 61(3):356–366, Sept 2015.

[11] J. Wang and MF Cohen. Optimized color sampling for robust matting. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*, pages 1–8, 2007.

[12] Meiguang Jin, Byoung-Kwang Kim, and Woo-Jin Song. Adaptive propagation-based color-sampling for alpha matting. *Circuits and Systems for Video Technology, IEEE Transactions on*, 24(7):1101–1110, July 2014.

[13] Alvy Ray Smith and James F. Blinn. Blue screen matting. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 259–268, New York, NY, USA, 1996. ACM.

[14] S. Wright. *Digital Compositing for Film and Video*. Focal Press visual effects and animation series. Elsevier/Focal Press, 2010.

[15] White P Kemp R, Pike G and Musselman A. Perception and recognition of normal and negative faces: the role of shape from shading and pigmentation cues. *Perception*, 25:37–52, 1997.

[16] Bruce V and Young A. *In the Eye of the Beholder. The Science of Face Perception*. Oxford: Oxford University Press, 1998.

[17] Paul Viola and Michael J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, May 2004.

[18] T. Phoka and A. Sudsang. Fine tuning for green screen matting. In *2017 9th International Conference on Knowledge and Smart Technology (KST)*, pages 317–322, February 2017.

[19] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846, Jan 1998.

[20] Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph.*, 23(3):664–672, August 2004.

[21] A. Buades, B. Coll, and J. M. Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65 vol. 2, June 2005.

[22] T. Phoka and A. Sudsang. Gpu-based nonlocal smoothing for alpha matting. In *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–6, July 2016.

[23] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1397–1409, 2013.

Appendices

ICIC Express Letters
Part B: Applications
Volume 8, Number 11(tentative), November 2017

ICIC International ©2017 ISSN 2185-2766
pp. 1–ICICIC2017-041

GREEN SCREEN MATTE REFINEMENT WITH MULTIPLE FILTERS

THANATHORN PHOKA
Department of Computer Science and Information Technology
Faculty of Science
Naresuan University
Phitsanulok 65000, Thailand
thanathornp@nu.ac.th

Received May 2017; accepted August 2017

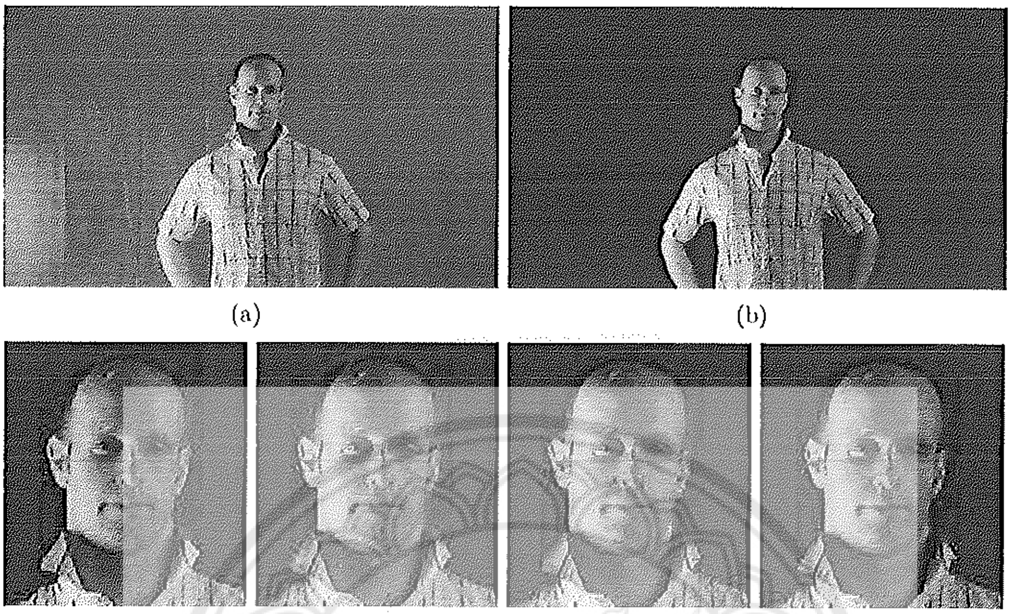
ABSTRACT. *This paper considers all procedures required in the process of a green screen matte refinement which are alpha matte smoothing, green spill suppression and compositing. An extracted matte usually has ridges on the edge and exceeding green spilling on the subject. Furthermore, the artifact on the edge of the original green screen image always exposes in the compositing procedure. These lead to the necessity of applying the green screen matte refinement. For simplicity, a popular conventional approach used for alpha matte smoothing is applying the Gaussian filter on the extracted matte but it may produce over-smoothing edge. The existing spill suppression approaches are to clamp the excess green based on the values of the red channel and the blue channel. This causes a hue shift and a brightness drop problem in a final composite. Unspill operation is the other existing approach proposed to relief the brightness drop problem but the hue shift problem remains. In order to resolve these limitations, this paper proposes the uses of multiple filter approaches for all procedures. Guided filter is the main filter applied in this paper for alpha matte smoothing and suppressing green spill because of its capability of smoothing an image while the details of the image are preserved according to a guidance image. Gaussian filter is taken into account for blending the edge of the matte and a new background because the artifact on the edge of the input image is sometimes needed to be removed. The experimental results will show that the proposed approach can produce high quality compositions.*

Keywords: Green screen matte, Alpha matte smoothing, Spill suppression, Edge blending, Guided filter, Gaussian filter

1. Introduction. Digital compositing is the digital process that adjusts and combines multiple images to produce a final image. An important procedure in digital compositing is filming a subject with a green screen plate and extracting the subject from the screen to be used in a compositing. The process of extracting the subject from the green screen is referred to as green screen matting or matting while the resulting extraction is referred to as a green screen matte or matte consisting of the foreground, background and alpha matte. Once a matte is acquired, it is then combined with a new background to produce a final composition. However, directly composting the matte may cause a nonnatural composition.

For an input example in Figure 1(a), the final composition with the proposed refinement is shown in Figure 1(b). Let us look in more details. The zoom-in compositions without any refinement, with smoothing, with spill suppression and with edge blending are shown in Figures 1(c)-1(f), respectively. We can see from Figure 1(c) that the unrefined composition has ridges at the edge and looks unnatural because of green spill. Therefore, the alpha matte smoothing is usually needed to remove ridges on the edge of the extracted matte.

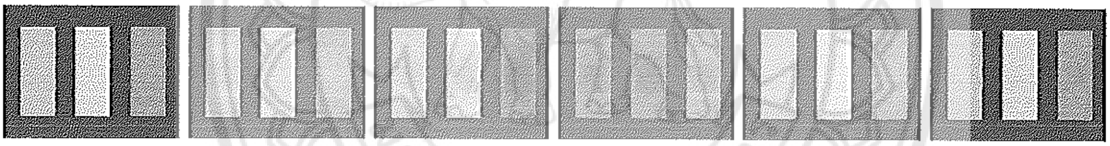
A main defect of the input image is from improper lighting. It may cause problems both on the green screen and the subject. A matte suffers from improper lighting especially



(a) (b)

(c) (d) (e) (f)

FIGURE 1. Refinement procedures



(a) (b) (c) (d) (e) (f)

FIGURE 2. Spill suppression artifacts

when a subject is underlit since scattering green light will shine on the subject as appearing on the left side of the face in Figure 1(a). More clearly, the composition without spill suppression in Figure 1(d) expresses a low quality composition with green glowing on the edge of the subject and green spill on the face. This leads to a necessity of applying spill suppression on the subject.

The conventional spill suppressions remove exceeding green or compensate by increasing blue and red values. The approaches globally adjust red, green and blue values that may cause brightness drop and hue shift artifacts as stated in [1]. The image in Figure 2(a) presents a test plate consisting of three strips which present skin tone, yellow and light blue. A modified test plate by filling green gradients is in Figure 2(b). The results from clipping exceeding green by red value and blue value are presented in Figures 2(c) and 2(d). They are severely affected by brightness drop and hue shift and still leave relative green on the skin tone strip in Figure 2(c) and the light blue strip in Figure 2(d). The unspill approach which concurrently decreases green and increases blue and red values can avoid the brightness drop but the hue shift still maintains as shown in Figure 2(e).

The main contribution of this paper is to propose the uses of multiple filters for matte refinement especially in the spill suppression procedure that avoids both brightness drop and hue shift artifacts by applying an iterative guided filter for diffusing proper differences between the green value and the blue and red values. A result from the proposed approach indicating a strong point compared to the existing approaches is presented in Figure 2(f) that is almost exactly the same as the original test plate. However, Figure 1(e) presents a result of the proposed spill suppression, and the result looks more natural but still leaves light gray glowing on the edge of the subject. This leads to another contribution of this paper to propose a heuristic for edge blending. Brightness of a new background is used as a constraint for the edge blending. Gaussian filter is then applied to generating weights for the edge blending. A result from the edge blending is shown in Figure 1(f). It decreases the glow artifact and produces a high quality final composition.

The organization of this paper is as follows. Related work and detail on matting, filters and spill suppression are described in Section 2. The proposed matte refinement procedures are explained in Section 3. The experimental results are presented in Section 4. The conclusions and an extension of this work are discussed in Section 5.

2. Related Work. Three main topics about matting, filtering and spilling suppression will be discussed in this section. Generally, matting is the process of extracting a subject from a natural background image to produce a matte in the form of alpha image, foreground and background color image. Before compositing the matte with a new background, the alpha matte smoothing process is usually needed to remove ridges on the edge of the matte. Filtering is then taken into account for this task. However, the green screen may leave some artifacts on the subject by spilling green color on them. The spill suppression is applied to removing the unwanted green from the foreground subject.

2.1. Matting. Given an image, matting is the process that extracts a foreground subject from a background. It can be modeled as a convex combination of foreground and background color as

$$C_i = \alpha_i F_i + (1 - \alpha_i) B_i, \tag{1}$$

where C_i is the input color, F_i is the foreground color, B_i is the background color and α_i is the opacity or blending of a pixel i. All these values of all pixels in an image are called a matte and let the opacity be called an alpha matte. From the equation, F_i, B_i and α_i have 7 unknown values but there are only 3 linear constraints. This makes matting be an ill-posed problem.

Natural image matting needs constraints to compute a result matte. Additional information such as trimap or scribbles determining the foreground, background and unknown regions has to be defined beforehand. There are two main approaches solving the natural image matting: propagation-based approaches [2, 3, 4] and color sampling-based approaches [5, 6, 7, 8, 9].

In this paper, the background color is assumed to be a green color and an extracted matte is given. The analysis of the green screen matting problem was presented in [10]. In practice, color difference and chroma-key are the fundamentals of commercial softwares for the green screen matting used in digital compositing [1]. Recently, an extension of the color difference was proposed to allow a user to fine tune color difference thresholds based on luminances [11].

2.2. Filters. Filtering is applied for smoothing an alpha matte. Box filter and Gaussian filter are commonly used because of their simplicity but they may decay the edge detail of the alpha matte. Bilateral filter [12] was proposed as an edge preserved filter. However, an alpha matte does not present the detail of an image. Joint bilateral filter [13] computes weights from another guidance image that is the color input image in this paper. The main drawback of the joint bilateral filter is that it may suffer from gradient reversal artifacts when neighboring pixels of a smoothed pixel are too different. Nonlocal filter [14] considers similarity of square patches to compute weights. A use of the nonlocal filter for an alpha matte smoothing was proposed in [15]. The filter can smooth a matte while preserving edge detail but the complexity of the algorithm is $O(Nr^2m^2)$ where m is the radius of a square patch used in weight computation.

Guided filter [16] is another edge preserved filtering with $O(N)$ complexity. Assuming a local linear model between the guidance image I and the filtering output image q, the guided filter smooths the input image p while preserving edge information transferred from the guidance image I:

$$q_i = a_k I_i + b_k, \quad \forall i \in \omega_k, \tag{2}$$

where k is the center of a window ω_k, the radius of which is defined by r, and (a_k, b_k) are constant linear coefficients in each ω_k. Finding the solution of the coefficients (a_k, b_k) can be cast into a minimization problem solved by the ridge regression and the solution is given by:

$$a_k = \frac{\frac{1}{|\omega|}\sum_{i\in\omega_k} I_i p_i - \mu_k \bar{p}_k}{\sigma_k^2 + \epsilon}, \tag{3}$$

$$b_k = \bar{p}_k - a_k \mu_k, \tag{4}$$

where μ_k and σ^2 are the mean and variance of I in ω_k, $\bar{p}_k$ is the mean of p in ω_k and ϵ is a regularization parameter.

A pixel i gets involved in all windows containing it; therefore, q_i is computed from the averages $\bar{a}_i = \frac{1}{|\omega|}\sum_{k\in\omega_i} a_k$ and $\bar{b}_i = \frac{1}{|\omega|}\sum_{k\in\omega_i} b_k$ as:

$$q_i = \bar{a}_i I_i + \bar{b}_i. \tag{5}$$

There are two parameters that can be adjusted, r and ϵ. As presented in [16], r and ϵ are corresponding to the variances of spatial and range values used in the bilateral filter, respectively. This paper applies the guided filter for alpha matte smoothing and spill suppression which will be described in Section 3.

2.3. Spill suppression. Spill suppression is applied to handling with exceeding green from a matte based on some constraints. Some simple but practical constraints are described in [1] which are limiting the green value by the blue or red value as follows:

$$C_{i,G} = \min\left(\beta C_{i,R} + \gamma C_{i,B}, C_{i,G}\right), \tag{6}$$

where β and γ are user-defined parameters. The green value is trimmed by the proportion of the red value when $\beta > 0$ and $\gamma = 0$ or of the blue value vice versa. The parameters can be freely adjusted by a user not restricted to the previous conditions to produce the best appearance. Applying this constraint is referred to as despill operation. However, the despill operation has two main artifacts which are brightness drop and hue shift. The brightness drop comes from decreasing the most influential color channel on brightness, the green value. The hue shift is from adjusting the green channel only. Modifying one channel in the RGB space clearly affects hue change.

The other practical spill suppression described in [1] is called unspill operation. The unspill operation implicitly provides brightness adjusting by allowing a user to decrease the green value and increase the red and blue values concurrently as follows:

$$C_{i,c} = C_{i,c} + g_c\left(C_{i,G} - \frac{C_{i,R} + C_{i,B}}{2}\right), \quad c \in \{R, G, B\}, \tag{7}$$

where g_c are user-defined gains for each channel, g_R and g_B are normally positive while g_G is negative. Even though the unspill operation relieves the brightness drop but it still suffers from the hue shift since all pixels share the same gain adjustments.

3. Matte Refinement. This section will describe all the procedures necessary to refine an obtained matte for a good final composition in order.

3.1. **Alpha matte smoothing.** Alpha matte smoothing is the first procedure applied to the extracted matte to refine the edge of the matte. As stated in [16], the guided filter can be used for guided feathering which is the alpha matte smoothing in this paper. The guided filter is directly applied by assigning I to be the input color image, p is an alpha matte obtained from a matte extraction procedure and q is the smoothed alpha matte. The intuition of this process is to smooth the alpha matte while preserving edge detail appearing in the input color image to avoid oversmoothing.

3.2. **Spill suppression.** This paper remodels the spill suppression by considering the differences between the green value and the remaining channel values. Once such differences are obtained, the new values of the red and blue channels are from the following:

$$C_{i,c} = C_{i,G} + D_{i,c}, \quad c \in \{R, B\}, \tag{8}$$

where $D_{i,R} = C_{i,R} - C_{i,G}$ is the difference between the red and the green values and $D_{i,B} = C_{i,B} - C_{i,G}$ is the difference between the blue and the green values. Let such differences be called spill differences. The next stage is to find spill differences of each pixel that retain the appearance of the original input image, remove green spill but induce less artifact than the existing approaches in both the hue shift and the brightness drop.

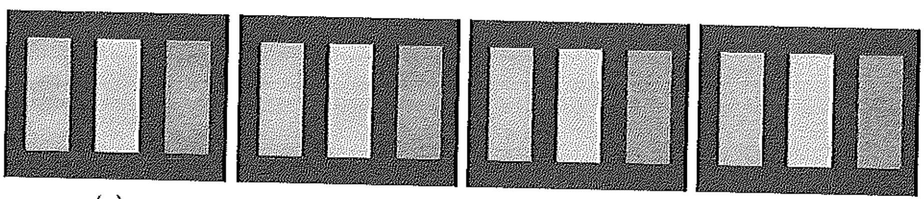
Assuming that pixels spilled by green is the minority, the guided filter is again used in this stage not only for smoothing but also for propagating spill differences. In theory, expanding r is not sufficient to propagate the spill differences and may smooth out edge detail so that an iterative guided filter is then applied instead. The guided filter in Equation (5) is reformulated as follows:

$$D_{i,c,t} = \bar{a}_{i,c,t} I_{i,t} + \bar{b}_{i,c,t}, \tag{9}$$

where the subscript t is the iteration of applying the guided filter. The color image I_t is updated for the next iteration by:

$$I_{i,c,t+1} = \max\left(I_{i,G,t} + D_{i,c,t}, I_{i,c,t}\right), \tag{10}$$

where I_0 is the composition of the smoothed matte from the alpha matte smoothing and a new background and $D_{i,c,0} = I_{i,c,0} - I_{i,G,0}$. The max function is applied to obstructing relative green diffusion. The value of $D_{i,c,t}$ that induces lower values of red or blue channels has to be blocked from the diffusion. The advantage of performing spill suppression on the composition is that it concurrently decreases the effect from the green background and increases the effect from the new background. The intuition of the iterative guided filter is that for each iteration the spill differences are smoothed according to the edge detail transferred from the composition. The composition is then updated by the smoothed spill differences and used in the next iteration. With proper adjustments of r, ϵ and t, the iterative guided filter can produce high quality spill suppressed results presented in Figure 3. The test plate is then again used to demonstrate an evolution of applying the iterative guided filter. The values of r and ϵ are assigned to be 20 and 0.05 while t is varied from $\{1, 5, 10, 15\}$ and the associated results are expressed in Figures 3(a)-3(d),



(a) (b) (c) (d)

FIGURE 3. An evolution of applying the iterative guided filter

respectively. As we can see from Figure 3(a). the original guided filter cannot handle with this situation but the iterative guided filter produces better results through iterations.

3.3. **Edge blending.** The proposed spill suppression works well in regular situations but not under some conditions such as edge enhancement from an image capturing device or ringing artifact from an image compression. These artifacts will leave glowing effect on the edge of a composition when the brightness of the original green background and of the new background is extremely different. The other artifact is from the spill suppression model in Equation (8) that $D_{i,c,t}$ is often greater than $D_{i,c,0}$ in the regions that are spilled by green. This makes the overall foreground of the spill suppressed composition look a little brighter.

In order to alleviate the artifacts, this paper proposes a heuristic for edge blending based on the brightness of the original foreground and the new background. The proposed heuristic not only blends the edge but also constrains the brightness of a spill suppressed foreground back to the original one. The first step is to blur the alpha matte obtained from the alpha matte smoothing procedure from the border to the inner side using the Gaussian filter and let the blurred alpha matte be denoted by α_b. Secondly, the input color image, new background image are converted from the RGB space to the YUV space and let S and T denote the Y channels of the input image and new background image, respectively. The brightness U of the output from this process is cast into the following convex combination:

$$U_i = \alpha_{b,i} S_i + (1 - \alpha_{b,i}) T_i. \tag{11}$$

However, green spill naturally induces the brightness drop so that the brighter foreground of the spill suppressed composition is sometimes preferred. In this situation, S can be replaced by the Y channel of the spill suppressed composition. Once U has been obtained, it is merged with the UV channels of the spill suppressed composition. The result is then recomposited with the new background to produce the final composition.

4. **Experimental Results.** For all experiments, the images in Figures 1(a) and 4(a) are from the Internet where their resolutions are 640×360 and 1440×1080, respectively. The machine used in the experiments consists of 3.4 GHz CPU and 16 GB ram. The proposed procedures are implemented in C++ language with OpenCV library. The results are compared among the proposed procedures. As stated in Section 1 that all proposed procedures are needed to produce a high quality composition, Figures 4(c)-4(f) show the zoom-in versions of the compositions without refinement, with guided filter applying to the alpha matte, with spill suppression and with edge blending, respectively. The results look better for every single procedure applied until the last step that produces the final composition expressed in Figure 4(b).

The spill suppression approach is evaluated by varying the values of the number of iterations t and the radius r while the regularization parameter ϵ is treated as a constant with the value 0.05 to avoid over propagation. The zoom-in results from adjusting these parameters are presented in Figure 4(g). The compositions look more satisfying by properly removing excess green when t and r are increased. However, the running time is increased according to the number of iterations performing the guided filter as expressed in Table 1.

The running times in Table 1 measured in the spill suppression procedure show that adjusting r does not affect the running time but increasing t does. The running time goes up linearly according to t. This complies with the complexity analysis and experiment in [16] that the guided filter runs in $O(N)$ time complexity. An image with resolution 1920×1080 is also considered in the running time measurement and the comparisons reflect that the number of pixels also influences the running time. The running times of the alpha matte smoothing for the 640×360, 1440×1080 and 1920×1080 inputs are

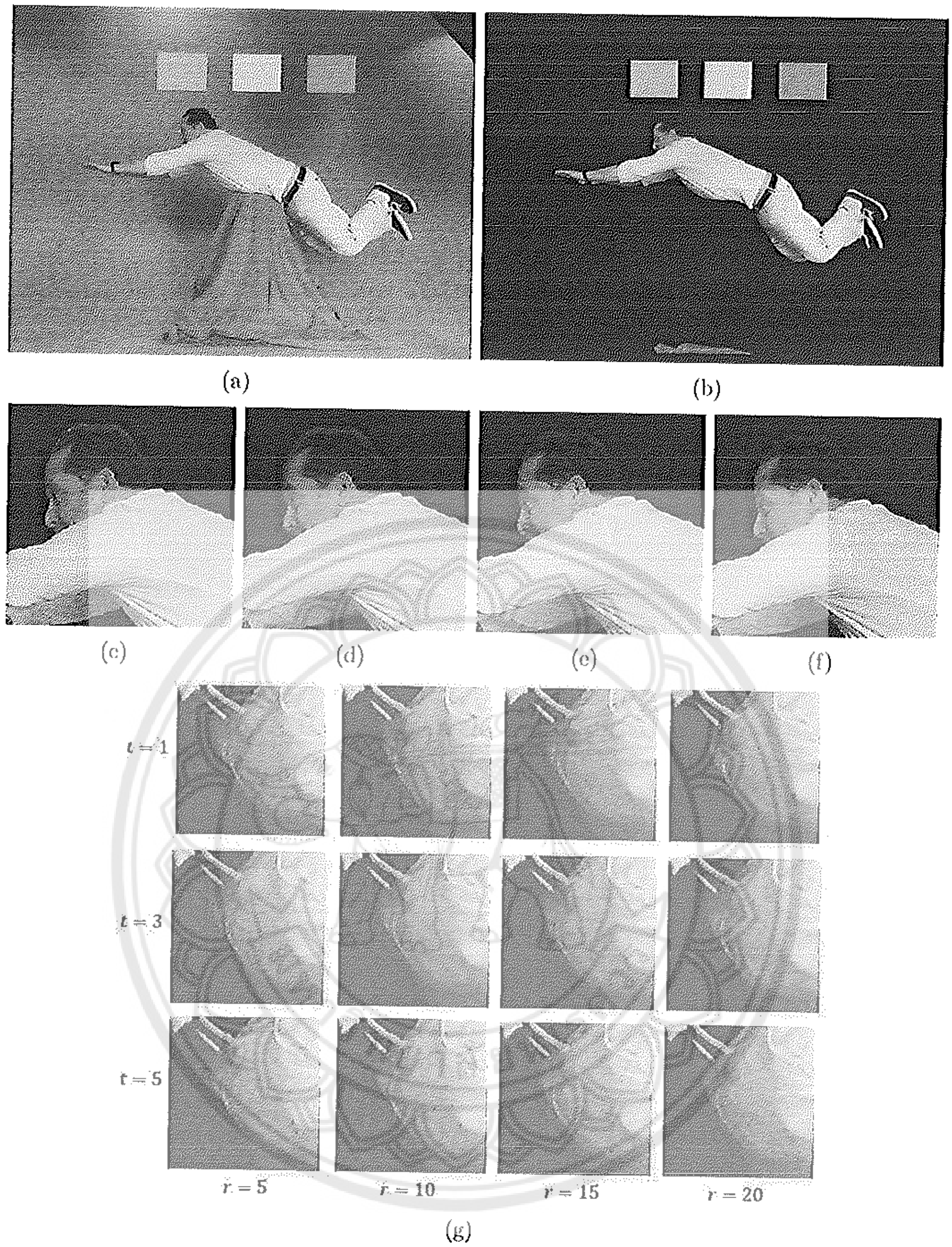


FIGURE 4. Experimental results

0.05, 0.36 and 0.48 and the edge blending procedure takes 0.04, 0.11 and 0.13, respectively which indicate the linear relations between the procedures and the number of pixels.

Furthermore, the spill suppression approach proposed in this paper aims to resolve the hue shift and brightness drop artifacts. As we can see from Figure 4(a), the three color blobs are intentionally added into the image, skin tone, yellow and light blue colors. The final composition expresses that the proposed approach can avoid both artifacts while removing the unwanted green.

5. Conclusions. This paper has proposed the procedures of a green screen matte refinement based on the guided filter and the Gaussian filter. The guided filter is the main

TABLE 1. Running times in seconds

t \ r	640 × 360				1440 × 1080				1920 × 1080			
	5	10	15	20	5	10	15	20	5	10	15	20
1	0.059	0.059	0.059	0.065	0.775	0.775	0.793	0.797	0.932	0.917	0.94	0.931
2	0.12	0.125	0.129	0.14	1.514	1.577	1.606	1.622	1.962	1.933	1.918	1.892
3	0.183	0.185	0.197	0.206	2.307	2.377	2.436	2.42	3.026	3.021	2.926	2.866
4	0.236	0.269	0.273	0.282	3.116	3.209	3.301	3.22	4.161	4.097	3.917	3.848
5	0.311	0.325	0.344	0.354	3.948	4.013	4.075	4.008	5.322	5.2	4.943	4.872

filter used in the alpha matte smoothing and the proposed iterative version is applied in the spill suppression procedure. The edge blending step applies the Gaussian filter to producing parameters for the convex combination in the Y channel of the YUV space of the foreground image and a new background image. The experimental results show that the proposed approaches can produce high quality compositions but the main drawback is the running time that linearly depends on the number of iterations applying the guided filter. However, the proposed approaches can be further implemented and optimized on a GPU because of their parallelism which is an extension of this paper.

Acknowledgment. This work is partially supported by research grant R2560C057 from Naresuan University which is greatly appreciated.

REFERENCES

[1] S. Wright, *Digital Compositing for Film and Video*, Focal Press Visual Effects and Animation Series, Elsevier/Focal Press, 2010.

[2] A. Levin, D. Lischinski and Y. Weiss, A closed-form solution to natural image matting, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.30, no.2, pp.228-242, 2008.

[3] P. G. Lee and Y. Wu, Nonlocal matting, *IEEE Conference on Computer Vision and Pattern Recognition*, pp.2193-2200, 2011.

[4] Q. Chen, D. Li and C.-K. Tang, KNN matting, *IEEE Conference on Computer Vision and Pattern Recognition*, pp.869-876, 2012.

[5] E. S. L. Gastal and M. M. Oliveira, Shared sampling for real-time alpha matting, *Computer Graphics Forum*, vol.29, no.2, pp.575-584, 2010.

[6] K. He, C. Rhemann, C. Rother, X. Tang and J. Sun, A global sampling method for alpha matting, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.2049-2056, 2011.

[7] D. Rajan, Weighted color and texture sample selection for image matting, *Proc. of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Washington, DC, USA, pp.718-725, 2012.

[8] E. Shahrian, D. Rajan, B. Price and S. Cohen, Improving image matting using comprehensive sampling sets, *Proc. of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, pp.636-643, 2013.

[9] W. Wang and J. Zhao, Robust image chroma-keying: A quadmap approach based on global sampling and local affinity, *IEEE Trans. Broadcasting*, vol.61, no.3, pp.356-366, 2015.

[10] A. R. Smith and J. F. Blinn, Blue screen matting, *Proc. of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, pp.259-268, 1996.

[11] T. Phoka and A. Sudsang, Fine tuning for green screen matting, *The 9th International Conference on Knowledge and Smart Technology (KST)*, pp.317-322, 2017.

[12] C. Tomasi and R. Manduchi, Bilateral filtering for gray and color images, *The 6th International Conference on Computer Vision*, pp.839-846, 1998.

[13] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe and K. Toyama, Digital photography with flash and no-flash image pairs, *ACM Trans. Graph.*, vol.23, no.3, pp.664-672, 2004.

[14] A. Buades, B. Coll and J. M. Morel, A non-local algorithm for image denoising, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol.2, pp.60-65, 2005.

[15] T. Phoka and A. Sudsang, GPU-based nonlocal smoothing for alpha matting, *The 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp.1-6, 2016.

[16] K. He, J. Sun and X. Tang, Guided image filtering, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.35, pp.1397-1409, 2013.

Fine Tuning for Green Screen Matting

Thanathorn Phoka
Department of Computer Science and Information Technology, Faculty of Science, Naresuan University, 65000, Thailand
Email: thanathornp@nu.ac.th

Warayu Jariyawattanarat and Attawith Sudsang
Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, 10330, Thailand
Email: warayu.ja@student.chula.ac.th and attawith@cp.eng.chula.ac.th

***Abstract*—This paper addresses the problem of fine tuning thresholds of a color difference method for green screen matting. The conventional color difference method uses two thresholds to control a result matte. However, the thresholds are applied to all pixels in an image which sometimes present structures in the image. In order to provide a flexibility t o c ontrol e ach group of pixels that have the same property in an image locally in the property space, this paper proposes a method that allows a user to adjust the thresholds associated to each group. In this paper, we use the luminance of a pixel to group the pixels in an image. Pixels with similar luminance are clustered into the same bin of a histogram. Local maxima extracted from the histogram will be used to initialize the thresholds associated to the groups corresponding to the local maxima. The experimental results will present that our proposed method take a little more effort adjusting thresholds to obtain a higher quality matte result.**

I. Introduction

Green screen matting is broadly used in many applications, e.g., video editing and compositing, filmmaking or TV broadcasting. Especially, TV broadcasting has been in our consideration especially when digital technology has been involved. The digital technology allows computer science people to research and develop devices used in broadcasting. Virtual studio is a successful example of using the digital technologies. The two main components of a virtual studio are a 3D render engine and a green screen matting module. The 3D render engine produces a background image according to a virtual camera's view. The matting module obtains a video feed from a camera that subjects are filmed in a green screen. Every single frame from the video feed is then computed to extract the subjects from the background screen. Once the virtual studio has obtained both the rendered background image and the image of the extracted subject, the compositing is then taken place to produce an output image frame. The green screen matting module is important not only for the virtual studio but also a classical video compositing using an image or video feed as a background, e.g., a weather forecast programme.

However, the quality of a result image or a video feed depends on capability of green screen matting. According to various studios used in recording, there exist studios which cannot afford new high technology devices, such as high definition video cameras producing high quality video feed with low video noise or sufficient numbers of dimmable light sources. These limitations affect the quality of green screen matting. From our preliminary experiment (not presented in this paper), we have observed that lighting condition is the most important factor affecting the quality of a result green screen matte. A shadow of a subject will be casted onto a green screen when the subject is closed to the insufficiently lit green screen. In another situation when a subject is not well lit, the subject can be spilled by green color. Overbright lighting can also induce a relatively white spot on a green screen.

In order to suppress the consequences of the lighting problems and obtain a higher quality of green screen matte, we consider the problem of fine tuning thresholds of a color difference method for green screen matting which is the conventional matting method used in the broadcasting industry since the color difference method is robust for inconsistent lighting. Further, the method is not computationally intensive, it can be computed in real time.

In this paper, we extend the color difference method which uses two thresholds to control a green screen matte extraction. One threshold is for determining the boundary of subjects or foreground and the other one is for determining the boundary of green screen color or background. These two thresholds are applied to all pixels in an image which might not be suitable for pixels in different image structures or uneven lighting conditions. For example, low luminance pixels might be a group of pixels presenting hair color which should be determined as a soft edge structure so that the green screen matte of the pixels with low luminance should be extracted by thresholds associated to their group. In another case, when a subject is wearing white clothes which the corresponding pixels are high luminance pixels, the pixels should be determined as a hard edge structure. However, white clothes are easily to be spilled by green color so that a result matte contains transparency. In order to get rid of the transparency in a hard edge structure, we should provide and adjust the thresholds for the high luminance pixels. Not only the problem with a subject, we also consider the problem of an uneven lit green screen that different green screen pixels with different lighting conditions should be handled by their own thresholds.

Our proposed method uses a luminance histogram produced

This work was partially supported by research grant R2560C057 from Naresuan University and research grant B2-1-0002/57 from Office of the National Broadcasting and Telecommunications Commission, both are greatly appreciated.

978-1-4673-9077-4/17/$31.00 ©2017 IEEE

by counting how many pixels in an image are at each brightness varying from 0 - 1. Peaks or in other words local maxima are extracted to be used to initialize thresholds for each maximal luminance. Once the thresholds for the maxima have been obtained, the thresholds for each step of luminance are interpolated using Catmull-Rom splines [1]. The thresholds associated with different luminances are then computed for green screen matting for each pixel in the image according to its brightness. Our method also allows a user to fine tune thresholds by adjusting the provided control points or adding more control points of the splines for a better green screen matte.

The organization of this paper is as follows. Research on alpha matting and green screen matting are discussed in section II. The detail of the color difference method is presented in section III. Our method based on the color difference method is proposed in section IV. Experimental results are discussed in V and we conclude our approach and future work in section VI.

II. Related Work

Matting can be divided into two categories which are alpha matting (or natural image matting) and green screen matting. A difference of these two categories is prior knowledge about foreground and background. Alpha matting uses a trimap to define foreground, background and unknown regions. The unknown region will be calculated for alpha value from the known foreground and background regions. Green screen matting uses a main color as a background color while the others are considered as foreground.

Alpha matting refers to the problem of decomposing an image or a video frame into foreground and background which an observed image can be modeled as a convex combination

$$C(p) = \alpha(p)F(p) + (1 - \alpha(p))B(p), \quad (1)$$

where $C(p)$ is the color of a pixel p, $F(p)$ is the foreground layer, $B(p)$ is the background layer and $\alpha(p)$ is opacity or alpha matte. From the equation, we can see that $F(p)$, $B(p)$ and $\alpha(p)$ are unknown values which makes matting be a highly ill-posed problem. Hence, the problem is constrained by additional information such as trimap or scribbles defining the foreground, background and unknown regions.

Existing approaches can be divided into two categories, propagation-based approaches and color sampling based approaches. Propagation-based approaches interpolate the unknown alpha values from the known regions assuming correlation between the neighboring pixels and use their affinities in the propagation [2]–[4]. Color sampling-based approaches collect a set of known foreground and background samples to estimate alpha values of unknown pixels from their best sample pair [5]–[10]. In [11], the two approaches are combined by casting the matting problem as an energy minimization problem in which the color sampling procedure forms the data term and the alpha propagation component formes the smoothness term. In [12], propagation and sampling are adaptively combined according to smoothness of regions to estimate alpha matte.

The green screen matting problem was analyzed in [13]. A simple solution to the matting problem does exist when constraints are taken into account. In the general case, a unique solution can be obtained if the foreground subject is captured against two backgrounds that every pixel is different. However, this approach is applicable only for images that non moving subjects can be captured twice on two different backgrounds.

In general, two classical approaches, color difference method and chroma-key method are bases of state-of-the-art devices, the Ultimatte® and the Primatte®, used in video editing [14]. The first approach calculates for each pixel the difference between the green color channel value $C_G(p)$ and the maximum value of the remaining channels $C_R(p), C_B(p)$. The difference value is then used to determine the alpha value of the pixel as shown in the following relation.

$$\alpha(p) \propto \max(C_R(p), C_B(p)) - C_G(p) \quad (2)$$

The second approach uses the distance between a reference color of a screen $C_{Ref}(p)$ and the color of a pixel $C(p)$ to calculate the alpha value as shown in the following relation.

$$\alpha(p) \propto \|C_{Ref}(p) - C(p)\| \quad (3)$$

These two approaches have low complexity and can utilize parallelism of GPU (graphic processing unit) so that they can compute a matte in real time. In this paper, we will extend the color difference method which is a practical approach used in the current broadcasting industry.

III. Color Difference

Alpha value is used to describe green screen matte. An alpha value of a pixel determines opacity of the pixel in an alpha matte. A pixel with 0 alpha value is defined as a background pixel while a pixel with 1 alpha value is defined as a foreground pixel. A pixel with alpha value in the range $(0, 1)$ is defined as a transparent pixel.

In order to calculate the alpha value $\alpha(p)$ of a pixel p against a green screen, we define color difference $D(p)$ as follows

$$D(p) = C_G(p) - \max(C_R(p), C_B(p)) \quad (4)$$

$D(p)$ is then used to calculate $\alpha(p)$ by taking two parameters $gain_D$ and b_D into the following equation.

$$\alpha(p) = -gain_D D(p) + b_D \quad (5)$$

where $gain_D$ and b_D are used to control a color curve graph transforming $D(p)$ into $\alpha(p)$ as described in [14].

More intuitively, equation 5 can be represented in the following form

$$\alpha(p) = 1 - saturate((D(p) - t_{min})/(t_{max} - t_{min})) \quad (6)$$

where $saturate$ is a function that bounds an input value into the range $[0, 1]$. $gain_D$ and b_D can be derived from t_{min} and t_{max} by

$$gain_D = \frac{1}{t_{max} - t_{min}} \quad (7)$$

$$b_D = \frac{t_{max}}{t_{max} - t_{min}}. \quad (8)$$

In this work, we focus on adjusting t_{min} and t_{max} of equation 6 since they intuitively induce $\alpha(p)$ by

$$\alpha(p) = \begin{cases} 1 \text{ when } D(p) < t_{min} \\ 0 \text{ when } D(p) > t_{max} \\ 1 - \frac{(D(p) - t_{min})}{(t_{max} - t_{min})}, \text{ otherwise.} \end{cases} \quad (9)$$

From the above calculations, t_{min} and t_{max} are thresholds determining that p is a foreground pixel, a background pixel or a transparency pixel according to the value $D(p)$.

Once t_{min} and t_{max} are obtained from user adjustment, they are used as the global thresholds that applied to all pixels in an image. This might leave some problems, for example, shadow casting on a green screen, uneven lighting condition on a green screen or green color spilling. In order to suppress consequences of the problems, we extend the color difference method by providing more controls for fine tuning matte result which will be described in the next section.

IV. Extended Color Difference

From our observations, we found that most problems occur because of lighting. In the case of uneven lighting on a green screen, t_{max} has to be decreased to cover all green screen pixels. This is undesirable since it makes soft edge matte become sharper. Over brightness, under brightness, shadow casting on a green screen lead to the same effect by decreasing color difference $D(p)$. This possibly forces t_{max} to be drastically decreased and sometimes leads t_{min} to be decreased as well. Green color spilling is also a problem from insufficient lighting on a subject. In this situation, a green screen matte leaves undesirable transparency on some pixels that should be opaque. In order to suppress this consequence, t_{min} has to be increased so that all spilled pixels are covered.

According to the above observations, adjusting thresholds related to luminance is reasonable. We can find uses of luminance in many applications and researches. For example, color correction which is the process of adjusting raw image to look more realistic. It divides adjustments into three luminance ranges which are shadows, midtones and highlights. Luminance also plays an important role in face recognition [15], [16] and face detection [17] in a form of grey scale image.

The overall processes of our method are presented in Fig. 1. An input color image consisting of subjects and a green screen is converted to the luminance image by converting a color pixel p to $L(p)$ using the following equation.

$$L(p) = 0.2126C_R(p) + 0.7152C_G(p) + 0.0722C_B(p) \quad (10)$$

The luminance image is then computed for a luminance histogram in the range $[0, 1]$ where the range is discretized into 256 bins. The histogram is constructed by counting how many pixels in the luminance image are at the brightness corresponding to each bin. Once the histogram has been obtained, it is smoothed using a Gaussian blur to avoid local maxima from histogram fluctuation. In our experiment, we use a kernel size equal to 9 for a Gaussian blur. Local maxima are extracted by tracing the smoothed histogram for bins that contain peaks of the smoothed histogram. Let the local maxima be defined by $\{lm_1, ..., lm_n\}$ where n is the number of local maxima.

The input color image is also computed for a D image by calculating all pixels for color difference value using equation 4. For each local maximum, the D image is then used for 2 cluster k-means clustering which finds the means of two groups according to D values of pixels that their luminances are similar to the luminance of the local maximum. All couples of the means associated to all local maxima are denoted by $\{(c_{min,1}, c_{max,1}), ..., (c_{min,n}, c_{max,n})\}$ where $c_{min,i} \le c_{max,i}$ for $i = 1, ..., n$.

Before initializing couples of thresholds for all local maxima, a user is needed to put some effort for adjusting a threshold t_{bin} to produce a good binary matte. A good binary matte means that most green screen pixels are classified as background or $\alpha(p) = 0$ and most subject pixels are classified as foreground or $\alpha(p) = 1$. The threshold t_{bin} is applied in the classification by

$$\alpha(p) = \begin{cases} 1 \text{ when } D(p) < t_{bin} \\ 0 \text{ when } D(p) >= t_{bin}. \end{cases} \quad (11)$$

The couple of thresholds for each local maximum lm_i is now ready to be initialized. A couple of thresholds $t_{min,i}$ and $t_{max,i}$ are bounded by t_{bin} that is $t_{min,i} = c_{min,i}$ and $t_{max,i} = c_{max,i}$ by default. In the case when $t_{max,i} < t_{bin}$, $t_{min,i} = t_{max,i}$ and $t_{max,i} = t_{bin}$ are performed. With the similar condition when $t_{min,i} > t_{bin}$, $t_{max,i} = t_{min,i}$ and $t_{min,i} = t_{bin}$. These bounding conditions are applied since if we let $t_{min,i} = c_{min,i}$ and $t_{max,i} = c_{max,i}$ in the case that $t_{max,i} < t_{bin}$, a pixel p that $D(p) < t_{bin}$ and $D(p) > t_{max}$ will be determined as a foreground pixel while it would rather be a transparent pixel or a background pixel according to t_{bin}. In the other case when $t_{min,i} > t_{bin}$, a pixel p that $D(p) > t_{bin}$ and $D(p) < t_{min}$ will be determined as a foreground pixel. These cause an undesirable matte result so that the bounding condition are reasonably performed.

However, the obtained $t_{min,i}$ and $t_{max,i}$ still cause a transparency problem as illustrated in Fig. 2. Let us consider back to the k-means clustering. An obtained mean is not suitable to be directly used as a threshold since when it represents a $t_{min,i}$ for foreground pixels, a pixel p that is a member of the same group of the mean and $D(p) > t_{min,i}$ will be defined as a transparency pixel but sometimes it is needed to be a foreground pixel or very low transparency pixel. The same problem also occurs for $t_{max,i}$. In order to avoid too many transparent pixels, our method squeezes $t_{min,i}$ and $t_{max,i}$ closer to t_{bin} by the following linear interpolations.

$$t_{min,i} = t_{bin} + gain_s(t_{min} - t_{bin}) \quad (12)$$

$$t_{max,i} = t_{bin} + gain_s(t_{max} - t_{bin}) \quad (13)$$

where $gain_s$ is a user defined parameter in the range $[0, 1]$. In our experiment, we assign $gain_s = 0.33$.

Once sets of thresholds $\{t_{min,1}, ..., t_{min,n}\}$ and $\{t_{max,1}, ..., t_{max,n}\}$ have been obtained, they are used as control points for two Catmull-Rom splines, one is for interpolating $\{t_{min,b}$ and the other one is for interpolating $\{t_{max,b}$ where b is a level of luminance and $b \in [0, 1]$ such that the range $[0, 1]$ is discretized into 256 levels. At this point, our method provides two user defined parameters which are t_{bin} and $gain_s$. Moreover, our method allows a user to fine tune more thresholds by adjusting or adding more control points $\{r_{min,1}, ..., r_{min,k}\}$ and $\{r_{max,1}, ..., r_{max,l}\}$ where k and l are the numbers of control points of the two splines. The splines then turn to be controlled by user defined control points and can be used to refine a matte result until a desired one is acquired.

V. Experimental Result

In our preliminary experiments, all input images are full definition with resolution of 1920×1080. All experiments are run on a desktop computer with 3.4 GHz CPU and 16 GB ram. The video card is NVIDIA Geforce GTX 680 with 4 GB ram. An illustration input image is shown in Fig. 3(a). We adjust the threshold t_{bin} for a binary matte resulted in Fig. 3(b). The matte produced from the set of initial thresholds according to the threshold t_{bin} is shown in Fig. 3(c) and Fig. 3(d) shows its composition with a blue background. Fig. 3(e) and (f) present fine tuned results. The sets of thresholds $\{r_{min,1}, ..., r_{min,k}\}$ and $\{r_{max,1}, ..., r_{max,l}\}$ used for fine tuning are as appeared in the bottommost window containing two splines.

We compare our method with the conventional color difference method, the results of which are shown in Fig. 3(g) and (h). The thresholds of the color difference are adjusted to result a matte that mainly covers foreground detail. As we can see from the results of the color difference method, there exist pixels that should be classified as background but they appear as foreground because they are casted by shadow. Our method handles this consequence quite well as we can see from Fig. 3(c), some undesirable foreground pixels become either background pixels or transparency pixels. However, there still exist some transparency pixels that should be foreground pixels because of green color spilling on white cloths. We put some effort on fine tuning thresholds by adding more control points and adjust them until a good matte in Fig. 3(e) is obtained.

Not only green color spilling, but fine tuning can also handle with transparency on a matte. Fig. 4 presents the results focusing on the hair of the lady. Fig. 4(a), (b) and (c) are from initialized control points, fine tuned control points and the color difference method, respectively. Even just the result from the initialized control points is better than from the color difference method and with fine tuning, we acquire a little better result than the result from the initialized control points.

We also measure runtime of the color difference method, our extended color difference method and control point initialization. They take 0.138 ms, 0.385 ms and 139.89 ms on average for the input image in Fig. 3(a). In general, runtime of a control point initialization is varied according to statistic of luminances appearing on an input image but it does not exceed 1 second so that the threshold initialization process works with our extended color difference method smoothly in real time.

VI. Conclusion and Future Work

We have proposed an extension of the color difference method for green screen matting problem. Our method allows a user to adjust the thresholds associated to each group of pixels sharing similar luminance. Pixels with similar luminance are clustered into the same bin of a histogram. The histogram is then smoothed and local maxima extracted from the smoothed histogram will be used to initialize the thresholds associated to the groups corresponding to the local maxima. Our method applies 2 cluster k-means to find two means of similar color difference value pixels that share similar luminance for initializing such thresholds. The preliminary results show that our method works better even just apply initialized thresholds. Our method also provides fine tuning for a better matte result by putting a little user's effort.

Our future work will focus on extensive experiment by taking various images with different image structures into account. We will also consider the problem of video feed input which is more complicated than a still input image. Utilizing hue instead of luminance for initializing and adjusting thresholds is an interesting extension as well.

References

[1] E. Catmull and R. Rom, "A class of local interpolating splines," in *Computer Aided Geometric Design*, R. E. BARNHILL and R. F. RIESENFELD, Eds. Academic Press, 1974, pp. 317–326.

[2] A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 228–242, 2008.

[3] P. G. Lee and Y. Wu, "Nonlocal matting," in *2011 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2011, pp. 2193–2200.

[4] Q. Chen, D. Li, and C.-K. Tang, "Knn matting," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 869–876.

[5] E. S. L. Gastal and M. M. Oliveira, "Shared sampling for real-time alpha matting," *Computer Graphics Forum*, vol. 29, no. 2, pp. 575–584, May 2010, proceedings of Eurographics.

[6] K. He, C. Rhemann, C. Rother, X. Tang, and J. Sun, "A global sampling method for alpha matting," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, June 2011, pp. 2049–2056.

[7] D. Rajan, "Weighted color and texture sample selection for image matting," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ser. CVPR '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 718–725.

[8] E. Shahrian, D. Rajan, B. Price, and S. Cohen, "Improving image matting using comprehensive sampling sets," in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 636–643.

[9] J. Johnson, D. Rajan, and H. Cholakkal, "Sparse codes as alpha matte," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[10] W. Wang and J. Zhao, "Robust image chroma-keying: A quadmap approach based on global sampling and local affinity," *Broadcasting, IEEE Transactions on*, vol. 61, no. 3, pp. 356–366, Sept 2015.

[11] J. Wang and M. Cohen, "Optimized color sampling for robust matting," in *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*, 2007, pp. 1–8.

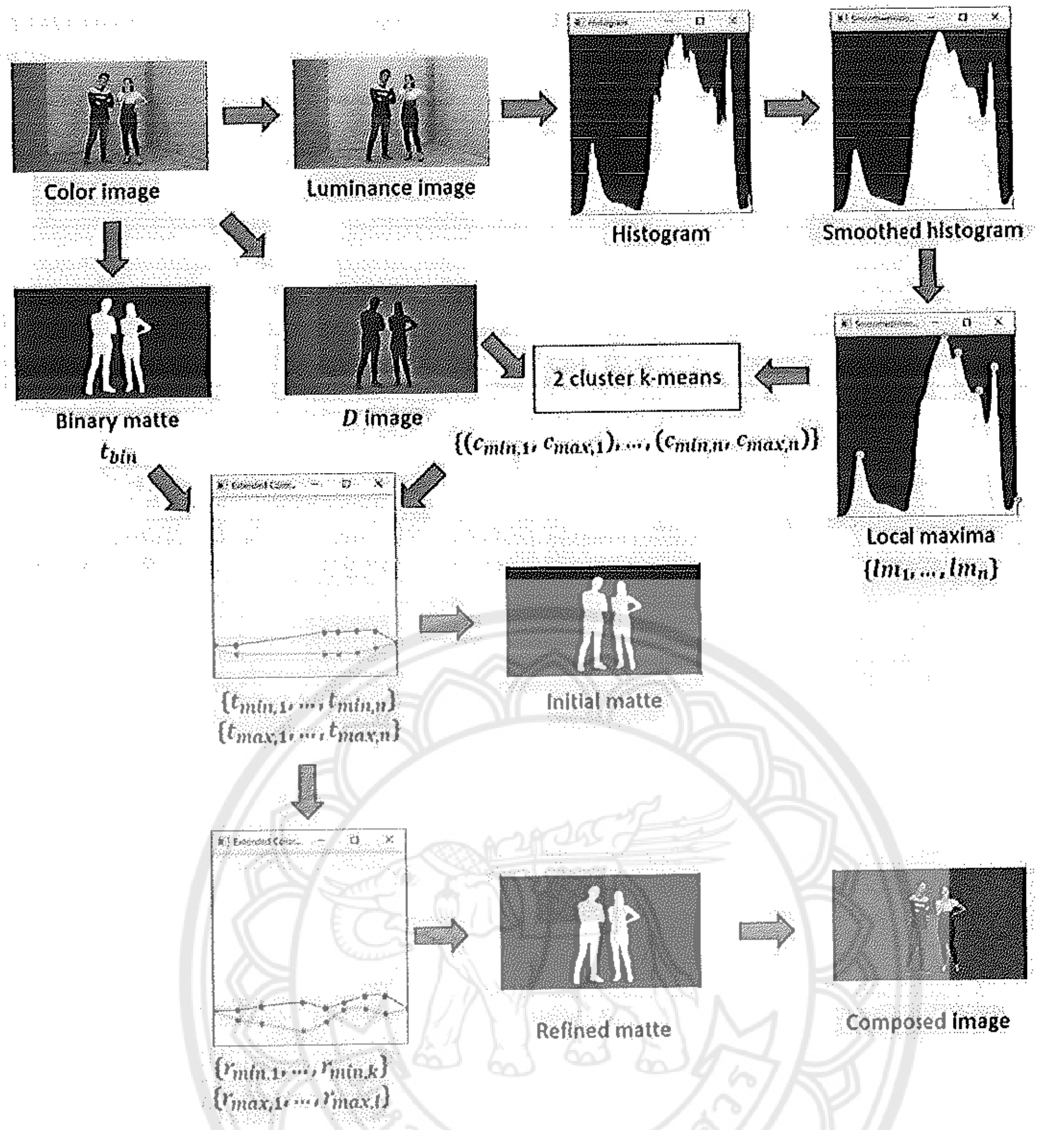


Fig. 1: The processes of our proposed method and illustrations

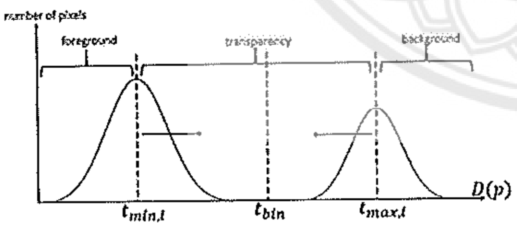


Fig. 2: Transparency problem

[12] M. Jin, B.-K. Kim, and W.-J. Song, "Adaptive propagation-based color-sampling for alpha matting," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 24, no. 7, pp. 1101–1110, July 2014.

[13] A. R. Smith and J. F. Blinn, "Blue screen matting," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '96. New York, NY, USA: ACM, 1996, pp. 259–268.

[14] S. Wright, *Digital Compositing for Film and Video*, ser. Focal Press visual effects and animation series. Elsevier/Focal Press, 2010.

[15] W. P. Kemp R, Pike G and M. A, "Perception and recognition of normal and negative faces: the role of shape from shading and pigmentation cues," *Perception*, vol. 25, pp. 37–52, 1997.

[16] B. V and Y. A, *In the Eye of the Beholder. The Science of Face Perception*. Oxford: Oxford University Press, 1998.

[17] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vision*, vol. 57, no. 2, pp. 137–154, May 2004.

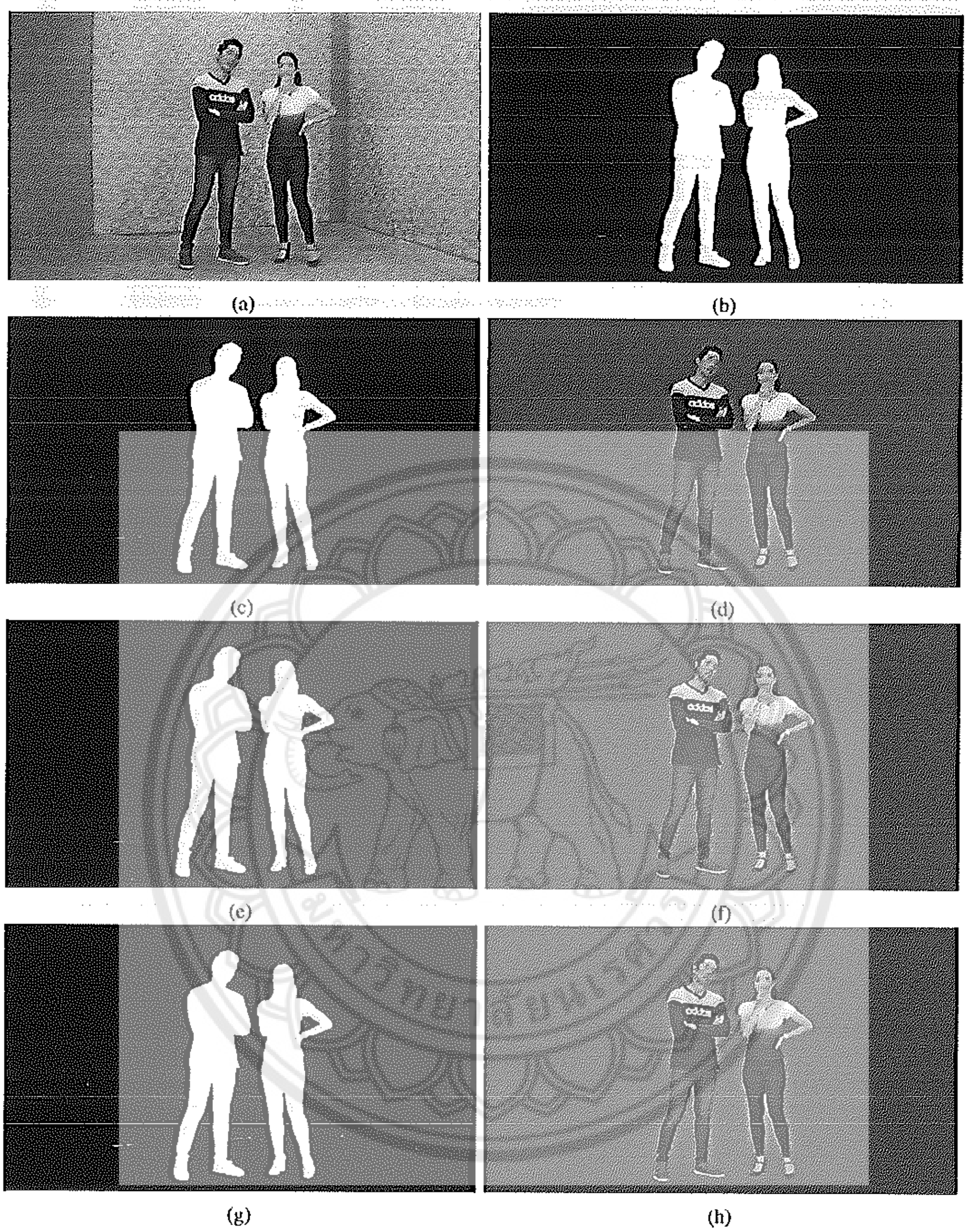


Fig. 3: The compared results

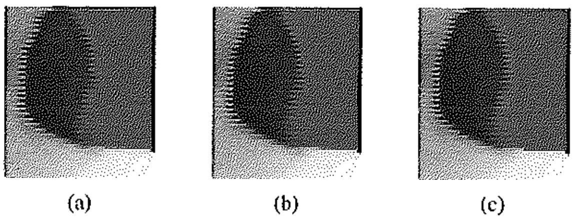


Fig. 4: The results zoomed in on hair

กิจกรรมที่วางแผนและได้ดำเนินการ

กิจกรรม	เดือนที่											
	1	2	3	4	5	6	7	8	9	10	11	12
1. ศึกษาและค้นคว้าทฤษฎีและผลงานตีพิมพ์ที่เกี่ยวข้อง												
2. สรุปแนวคิด ข้อดีข้อเสีย และหลักการทำงานของขั้นตอนวิธีที่เกี่ยวข้อง												
3. ดัดแปลงและปรับปรุงขั้นตอนวิธีที่เกี่ยวข้องที่สามารถนำมาประยุกต์ใช้กับหน่วยประมวลผลกราฟิก												
4. พัฒนาขั้นตอนวิธีการแยกฉากหลังสีเขียวที่มีความสามารถในการแยกฉากสีเขียวได้หลายโทน มีความทนทานต่อสภาพแสงและการถูกรบกวนของภาพจากกล้องวิดีโอ และมีแนวโน้มที่จะมีประสิทธิภาพเพียงพอในการใช้งานแบบทันทีได้												
5. ออกแบบการทดลอง ทำการทดลองและสรุปผล												
6. สรุปผลที่ได้และจัดทำบทความวิจัยเพื่อตีพิมพ์ผลงานลงในวารสารวิชาการระดับนานาชาติ												

ผลที่ได้รับตลอดโครงการ

1. งานวิจัยนี้จะก่อให้เกิดการประยุกต์ใช้ขั้นตอนวิธีการแยกฉากหลังบนหน่วยประมวลผลกราฟิกที่มีประสิทธิภาพดียิ่งขึ้น
2. งานวิจัยนี้จะก่อให้เกิดขั้นตอนวิธีการแยกฉากหลังสีเขียวที่มีความสามารถในการแยกฉากสีเขียวได้หลายโทน มีความทนทานต่อสภาพแสงและการถูกรบกวนของภาพจากกล้องวิดีโอ และมีแนวโน้มที่จะมีประสิทธิภาพเพียงพอในการใช้งานแบบทันทีได้
3. ทำให้ทราบถึงแนวทางการต่อยอดงานวิจัยทางด้านการแยกฉากหลังทั้งแบบที่เป็นสีเขียวและเป็นสีไม่เฉพาะ
4. พัฒนาศักยภาพนักวิจัยในโครงการและผลิตผลงานเพื่อตีพิมพ์ในวารสารระดับนานาชาติ