

UNIVERSIDAD POLITECNICA DE VALENCIA

ESCUELA POLITECNICA SUPERIOR DE GANDIA

MASTER EN POSTPRODUCCION DIGITAL



UNIVERSIDAD
POLITECNICA
DE VALENCIA



ESCUELA POLITECNICA
SUPERIOR DE GANDIA

“Triangular matting con patrones temporales en pantallas retroreflectivas”

TRABAJO FINAL DE MASTER

Autor:

Juan Alberto Lafuente Hernández

Director:

Borja Vidal Rodríguez

Gandía, Septiembre de 2012

Agradecimientos

Primero que nada quiero agradecer a mi tutor **Borja Vidal Rodríguez** por su colaboración, dedicación, apoyo y sobre todo por la confianza que tuvo en mí en los momentos duros que uno sufre cuando dedica sus horas a programar y no obtiene ningún resultado.

También quiero agradecer al equipo técnico del plató de televisión por cedernos sus instalaciones y ayudarnos en todo lo necesario a lo largo de todo este tiempo y a los técnicos electrónicos por dedicarnos su tiempo, sus conocimientos y facilitarnos la investigación.

Y por último y no por eso menos importante dedicárselo con especial cariño a mi familia por estar siempre a mi lado apoyándome en todo y confiando en mí todos estos años durante la carrera y ahora en el Máster de Postproducción de Digital.

Muchas gracias

Resumen

La técnica propuesta permite evitar de forma sencilla las limitaciones de color en la filmación, basándose en la composición cromática de llaves de vídeo mediante una pantalla retroreflectiva y un anillo emisor de luz con diodos Led que están acoplados al mismo. Se aplica un patrón temporal asíncrono a la lona para generar el Chroma-Key y por procesamiento de imagen se consigue detectar el color del fondo sin restricciones en los colores presentes en la escena a grabar.

La técnica elimina por completo las limitaciones técnicas en la composición artística que existían en la etapa de procesado de un Chroma-Key. Es perfectamente compatible con todos los dispositivos que se emplean en los estudios de televisión y cine hasta la fecha para realizar la extracción, pero con una pequeña modificación electrónica de muy bajo coste en el anillo de diodos LED.

Palabras clave: Chroma-Key, Máscara, Patrón, Pantalla Retroreflectiva.

Abstract

A simple technique to avoid color limitations in filming based on Chroma key video composition using retroreflective screens and light-emitting diodes (LED) rings is proposed and demonstrated. An asynchronous temporal pattern is applied to the Chroma ring and through simple image processing; the background color can be substituted by any image or video without the usual restrictions on scene colors for still and moving objects in the foreground.

The technique removes technical constrains in stage composition allowing its design purely based on artistic grounds using off-the-self components widely used in TV and filming studios and low-cost electronics.

Keywords: Chroma-Key, Mask, Pattern, Retroreflective Screen.

CAPÍTULO 1. - INTRODUCCIÓN.....	Pág. 9
1.1 Motivación	Pág. 10
1.2 Objetivos	Pág. 11
CAPÍTULO 2. – REVISIÓN DEL ESTADO DEL ARTE Y PLANTEAMIENTO DEL PROBLEMA.....	Pág. 12
2.1 Descripción Chroma-Key	Pág. 12
2.1.1 Evolución Digital del Chroma-Key	
2.2 Setup Chroma-Key	Pág. 16
2.2.1 Pantalla Retroreflectiva	Pág. 18
2.2.2 Hardware/Software	Pág. 20
2.3 Limitaciones en el color	Pág. 23
2.4 Soluciones previas	
2.4.1 Cámara térmica	Pág. 24
2.4.1.1 Descripción	
2.4.1.2 Ventajas y limitaciones	
2.4.2 Substracción del fondo	Pág. 25
2.4.2.1 Descripción	
2.4.2.2 Ventajas e limitaciones	
2.4.3 Llave por profundidad	Pág. 27
2.4.3.1 Descripción	
2.4.3.2 Ventajas e limitaciones	
2.4.4 Ciclorama patrón	Pág. 30
2.4.4.1 Descripción	
2.4.4.2 Ventajas e limitaciones	

CAPÍTULO 3. - DESARROLLO DE LA TÉCNICA PARA EVITAR LIMITACIONES DE COLOR.....Pág. 34

3.1	Introducción	
3.2	Preparación e investigación de la técnica	Pág. 34
3.3	Desarrollo del algoritmo	Pág. 40
3.3.1	Descripción	
3.3.2	Resultados	
3.3.3	Ventajas e inconvenientes	
3.4	Diseño e Implementación GUI	Pág. 46
3.4.1	Descripción	
3.4.2	Implementación del código y del entorno gráfico	

CAPÍTULO 4.Pág. 49

4.	CONCLUSIONES	Pág. 49
5.	BIBLIOGRAFÍA	Pág. 51
6.	ANEXO	Pág. 54

- [Figura 1.](#) Técnica Chroma-Key Normal. Capturas de pantalla desde Matlab
- [Figura 2.](#) Sensor CCD. Google imágenes
- [Figura 3.](#) Patrón Bayer. www.caborian.com
- [Figura 4.](#) Conversión de la luz en datos mediante un sensor CCD. Google imágenes
- [Figura 5.](#) Set básico Chroma-Key. Google imágenes
- [Figura 6.](#) Pintura Chroma-Key. Google Imágenes
- [Figura 7.](#) Iluminación para un set Chroma-Key. www.fxphoto.webs.com
- [Figura 8.](#) Lona retroreflectiva. www.reflecmedia.com
- [Figura 9.](#) Composición interna de la lona. www.reflecmedia.com
- [Figura 10.](#) Anillo de Led's. www.reflecmedia.com
- [Figura 11.](#) Conjunto del Kit. www.reflecmedia.com
- [Figura 12.](#) De izquierda a derecha y arriba a bajo Ultimatte 11, Tricaster 8000, AW-HS5050, DVK-200, Safire
- [Figura 13.](#) Keylight 1.2. Captura de pantalla de la aplicación
- [Figura 14.](#) Primatte. Captura de pantalla de la aplicación
- [Figura 15.](#) Poliedro Primatte. www.wikipedia.com
- [Figura 16.](#) Anillos del algoritmo Primatte. www.wikipedia.com
- [Figura 17.](#) Ejemplo de las limitaciones de color. www.cassetteblog.com/2012/06/jessica-starr-la-chica-chroma/
- [Figura 18.](#) Chroma Térmico. www.nae-lab.org/project/thermo-key/
- [Figura 19.](#) Configuración cámaras. www.nae-lab.org/project/thermo-key/
- [Figura 20.](#) Máscaras para captura por sustracción [15]
- [Figura 21.](#) Configuración del sistema 3DV [16]
- [Figura 22.](#) Sistema 3DV [16]
- [Figura 23.](#) Diferencia de profundidad en 3DV [16]
- [Figura 24.](#) Máscara generada sin ajuste [16]
- [Figura 25.](#) Máscara ajustada [16]
- [Figura 26.](#) Máscara ajustada [16]
- [Figura 27.](#) Ciclorama Patrón [17]
- [Figura 28.](#) Interpolación de líneas [17]
- [Figura 29.](#) Detección de los colores fondo [17]

- [Figura 30.](#) Detección de la cuadrícula [17]
- [Figura 31.](#) Extracción del primer plano (interpolación) [17]
- [Figura 32.](#) Unión del Background y Foreground [17]
- [Figura 33.](#) Setup del estudio para la obtención de datos
- [Figura 34.](#) Diseño del circuito generador de patrón mediante Orcad9.2 Portable
- [Figura 35.](#) Señal recibida por el anillo, nivel captado por la cámara y evolución de la luz durante la conmutación
- [Figura 36.](#) Niveles de señal en componentes RGB captados por la cámara en varios frames
- [Figura 37.](#) Diferentes niveles de verde que puede generar el patrón
- [Figura 38.](#) Diagrama de bloques de los diferentes puntos del código
- [Figura 39.](#) Captura de pantalla del Foreground empleado para la composición
- [Figura 40.](#) Captura de pantalla del Background empleado para la composición
- [Figura 41.](#) Captura de pantalla de las Máscaras empleadas para la composición
- [Figura 42.](#) Captura de pantalla de los Histogramas RGB de la composición
- [Figura 43.](#) Captura de pantalla de la composición final. Pulsa en la figura para reproducir la composición final
- [Figura 44.](#) Captura de pantalla del entorno gráfico Guide
- [Figura 45.](#) Captura de pantalla del entorno gráfico Guide, Botonera de edición
- [Figura 46.](#) Captura de pantalla del entorno gráfico Guide, Ventana Inspector
- [Figura 47.](#) Captura de pantalla del entorno gráfico Guide Implementado.
- [Figura 48.](#) Medidor de Luminancia [22]
- [Figura 49.](#) Circuito eléctrico generador del patrón
- [Figura 50.](#) Multivibrador 555
- [Figura 51.](#) Simulación del funcionamiento del sistema. Si pulsa sobre la imagen se reproduce un archivo multimedia donde se puede visualizar el funcionamiento

CAPÍTULO 1.

INTRODUCCIÓN

En el mundo del cine y la televisión muchas son las técnicas que se emplean para crear contenidos alternativos a la simple representación de lo captado por la cámara y así fortalecer y dinamizar el material audiovisual como podría ser los efectos especiales en el Cine o la previsión meteorológica en la TV, pero para ello es necesario crear una máscara o recorte donde poder representarlos conjuntamente y uno de los métodos más empleados es el Chroma-Key. No sólo se emplea en el cine y televisión sino que también en prensa, educación, juegos o realidad aumentada.

El Chroma-Key se considera una técnica audiovisual la cual consiste en la sustitución de un fondo con un color uniforme y sólido (normalmente de color verde o azul) que sirva de recorte para añadir otro fondo, ya que el empleo de rotoscopia cuadro a cuadro es muy costoso en tiempo y dinero.

La definición que recomienda el diccionario del *British Kinematograph* sobre el Chroma-Key es [20]:

Es un proceso electrónico que combina las señales de salida de dos o más cámaras entre sí y/o con otras fuentes externas obteniendo como resultado una mezcla uniforme y visualmente indetectable.*

*Elementos como los escáneres de filmación, las trucas, las pinturas mate y los ordenadores.

En la figura1 podemos hacernos una idea en que consiste la técnica de una forma visual. En primer lugar se aprecia la imagen llamada foreground, seguido de la máscara de recorte generada y por último la combinación del foreground con el nuevo fondo o background. La figura muestra como todos los píxeles de alto contenido en verde de la imagen foreground se eliminan y son sustituidos por la imagen background en la composición.



Figura 1. Técnica Chroma-Key Normal. Capturas de pantalla desde Matlab

Esta técnica de composición se emplea desde hace años y tiene sus orígenes en las técnicas de “Travelling Matte” desarrolladas en los años 30. En la actualidad es una técnica muy empleada en diferentes ámbitos audiovisuales debido a la gran reducción en costes de producción que supone y también a la gran flexibilidad artística que ofrece. Gracias a la evolución tecnológica y a una mejora de rendimiento de los equipos a lo largo de todos estos años, se han conseguido resultados en los cuales no es visible el trabajo realizado debido a la buena depuración que se tiene sobre la técnica y por consiguiente se dice que es un trabajo poco apreciado [1].

1.1 MOTIVACIÓN

Como anteriormente hemos comentado el Chroma-Key es una técnica que en la actualidad se emplea bastante, está muy depurada y pocas son las variaciones que se le pueden ampliar para su mejora. Pero pensando un poco sobre la metodología empleada que se sigue para conseguir un óptimo resultado se llegó a la conclusión que sí que existía un campo por el cual sería interesante investigar.

Dicho campo está relacionado en las limitaciones que posee la técnica, entre éstas, una sería la difícil obtención de una iluminación uniforme que nos ayude a conseguir una máscara de recorte lo más ajustada posible y la otra la principal problemática que existe a la hora emplear vestuario del mismo color que dispone el Chroma-Key, limitando así las posibilidades creativas a la hora de componer. Por tanto la presente tesina de master aborda estos problemas.

1.2 OBJETIVOS

En este trabajo se pretende abordar el problema de las limitaciones en los colores presentes en una escena cuando se va a realizar una Chroma-Key. En concreto se usará un esquema basado en la utilización de un patrón definido de iluminación sobre una pantalla retroreflectiva, dicho patrón se obtiene de una forma sencilla a partir de un simple circuito eléctrico, para que un algoritmo asíncrono permita realizar la composición de video sin restricciones sobre los colores presentes en la escena. El algoritmo se programará en Matlab y se desarrollará una interfaz gráfica que implemente un Chroma-Key sobre secuencias de video capturadas en el Estudio de TV de la EPSG.

El presente documento está estructurado como se detalla a continuación: Como hemos visto en el Capítulo 1, hemos realizado una pequeña introducción sobre el Chroma-Key, hablado sobre la motivación que nos ha impulsado a realizar la tesina y también hemos descrito los objetivos a cumplir.

Seguidamente en el Capítulo 2 describiremos en mayor profundidad que es un Chroma-Key, cual ha sido su evolución a lo largo de estos años y también explicaremos como montar un set para Chroma-Key con las pautas más importantes a la hora de elegir una ubicación, tipo de materiales y equipamiento. Hablaremos de las limitaciones que tiene en la actualidad dicha técnica, mostrando algunas alternativas que han implementado otros investigadores como solución al problema.

El Capítulo 3 está destinado a explicar y desarrollar nuestra técnica que evita las limitaciones de color. Primero realizaremos una introducción donde hablamos a grandes rasgos de cual es la principal idea de la técnica. Seguidamente hablaremos de los materiales, configuración del set y explicación de los algoritmos necesarios para llevarla a cabo. También hablaremos de los diferentes problemas que pueden surgir con sus soluciones respectivas y por último explicaremos paso a paso el funcionamiento del programa diseñado mediante Matlab y realizaremos un entorno gráfico mediante GUI (Graphic User Interface), que está disponible en el propio Matlab.

En el Capítulo 4 comentaremos las conclusiones obtenidas a lo largo de toda la investigación y terminaremos con la bibliografía y anexos.

CAPÍTULO 2.

REVISIÓN DEL ESTADO DEL ARTE Y PLANTEAMIENTO DEL PROBLEMA

2.1 DESCRIPCIÓN CHROMA-KEY

Desde los albores de la cinematografía se ha querido extraer partes de una escena y ubicarlos en otra de una manera convincente. En sus principios eran procesos únicamente fotográficos con exposiciones múltiples en la misma película, técnicas completamente manuales como por ejemplo pintar directamente sobre la película con pintura para realizar el recorte (este proceso sería el precursor de la actual rotoscopia), pero eran procesos muy costosos e imposible de obtener resultados realistas.

Con estas limitaciones los fotógrafos de la época pensaron como podían resolver estas limitaciones con la química, ya que las películas tienen capas de emulsión que pueden ser aisladas químicamente para algún color, así de esta forma obtendrían el recorte del objeto en primer plano automáticamente empleando dicho color como background (finalmente sería el azul). En definitiva es un proceso muy parecido a lo que se emplea actualmente, se extrae el objeto "A" para llevarlo al fondo "B".

Como podemos observar el material con el que se registran las imágenes es la clave para obtener un buen Chroma-Key y decantarnos por un color u otro como llave. El color empleado tiene que tener una gama de color muy estrecha para facilitar su localización y evitar problemas con el resto de colores u objetos que se empleen en la composición.

Generalmente se ha empleado el color azul debido a la composición del celuloide que se emplea en las cámaras más antiguas, contienen mejores cristales en la capa azul que en la verde o roja, pero con la revolución tecnológica llegaron los sensores CCD/CMOS como el de la figura 2. Debido a la mayor sensibilidad del ojo a la componente verde, el sensor se diseña para sobreponderar esta componente, por ejemplo en la máscara de Bayer mostrada en la figura 4 el número de píxeles para el verde es el doble que para el rojo y azul.

Por consiguiente al emplear una llave de color verde, nos generará menor "ruido" en la imagen, al ser más sensible a dicho color la iluminación en la escena podrá ser menos restrictiva que con el azul [2].

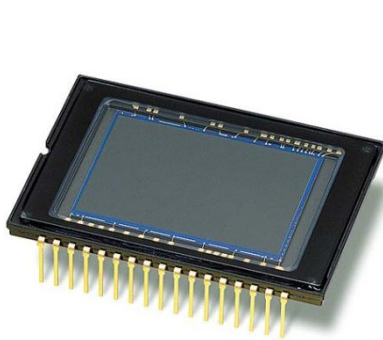


Figura 2. Sensor CCD. Google imágenes

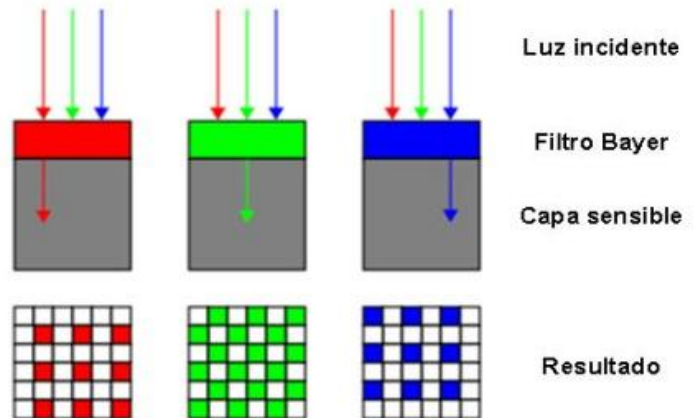


Figura 3. Patrón Bayer. www.caborian.com

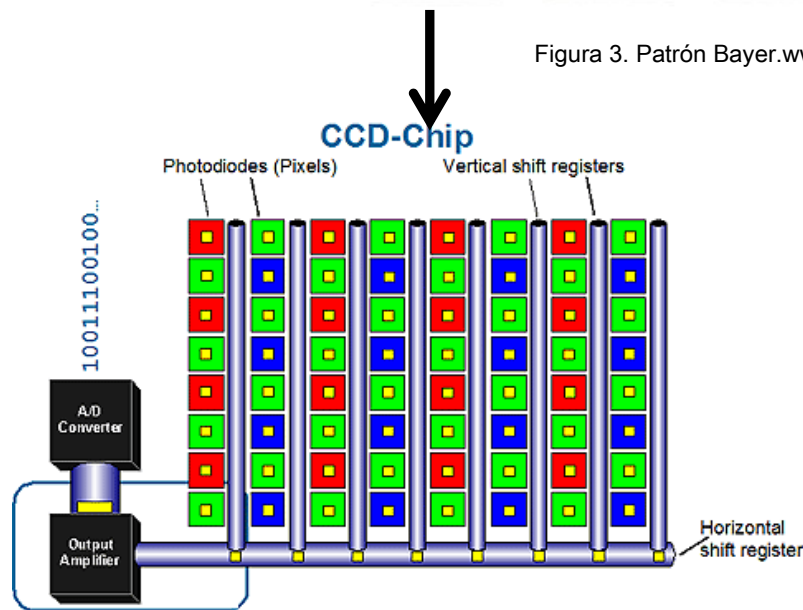


Figura 4. Conversión de la luz en datos mediante un sensor CCD. Google imágenes

Una vez explicado es fácil de entender porque el color más empleado actualmente en los Chroma-Key es el verde, pero siempre dependerá la elección de uno u otro color dependiendo de gustos y la composición artística. ¿Qué pasa cuando queremos emplear en la composición el mismo color que se está empleando en la llave?, a no ser que sea intencionadamente como por ejemplo hacer desaparecer un objeto o persona tendremos un grave problema que explicaremos e intentaremos solucionar posteriormente.

Otro punto importante como hemos mencionado anteriormente es la iluminación, ya que una mala o pobre iluminación puede acarrear problemas como sombras indeseadas, brillos en los objetos o en el ciclorama, coloración del ciclorama provocando que cambie de color en distintas zonas y con la consiguiente complejidad a la hora de generar la máscara.

2.1.1 Evolución Digital del Chroma-Key

Desde las primeras técnicas de Chroma-Key que consistían en crear una tira de película en blanco y negro donde el transparente era lo que se quería reemplazar y opaco en lo que se quería preservar de la imagen, hasta lo que se emplea en la actualidad ha pasado mucho y todo es gracias a la llegada de era digital, donde se consiguió que este proceso fuera más fácil, rápido y preciso.

Los primeros sistemas de Chroma-Key fueron desarrollados por Petro Vlahos fundador de la Corporación Ultimatte en 1976 [21]. Fue el primero en desarrollar el proceso de diferencia de color en pantalla azul digitalmente y consistía en un hardware que realizaba la tarea en tiempo real aunque no será hasta 1995 cuando realice plugins para software de terceros (AdvantEdge, UltimatteRT) y facilite así aún más la entrada del Chroma-Key digital en el mundo audiovisual.

Se emplearon ecuaciones muy básicas para el funcionamiento de Ultimatte, siendo la primera compañía que aplicaba algoritmos para dispositivos de vídeo en tiempo real.

$$\text{Si } (R < G) \quad E = B - G$$

$$\text{Si } (R > G) \quad E = R - B$$

Esta fórmula seguirá mejorándose con el paso de los años hasta cerca de principios de los años 90 para ser cada vez más precisa y rápida aumentando su complejidad matemática y ayudándose de equipos cada vez más potentes que fueran capaces de procesarlo.

En 1992 a 1994 surge otra alternativa llamada Primatte de Yaz Mishima desarrollado por IMAGICA Corp (Tokio).

En este caso no sólo emplea la información de color para realizar el recorte sino que también entra ahora en el algoritmo la variable de luminancia muy importante para diferenciar entre un color más oscuro o más claro, de esta manera Primatte es la mejor opción para fondos que no estén bien iluminados.

En 1996 sería cuando la empresa Primatte vendió su primera copia de la aplicación para otras empresas y estas al ver que era una aplicación rápida y muy eficaz se implantó rápidamente en el mercado de la mano de empresas como por ejemplo Apple, Red Giant, Combustion, Avid, Monkey, Nuke y Shake.

En 1997 aparece Keylight desarrollado por CFC, fue primeramente diseñado como un plugin para Cineon aunque se convertiría en un estándar para Shake de la mano de Apple y finalmente llegará a los demás software como por ejemplo After Effects. Emplea la diferencia de color en sus algoritmos por lo que funciona mejor con colores primarios. La principal baza que tiene Keylight es la eliminación de derrames en el proceso de extracción, lo que lo hace uno de los plugins más finos a la hora de trabajar sobre vidrio o cabello.

En 1998 nace Modular Keyer Flame de la mano de Pettigrew, este tenía como principal objetivo ser rápido a la hora de obtener una solución precisa, dispone de su propio motor de corrección de color y está especialmente diseñado para Chromas de bajo nivel.

En 2000 con la llegada de las miniDV se convierte en un problema porque estas trabajan con una compresión en el color 4:2:0 o 4:1:1 y por tanto los equipos que había en el momento no eran capaces de funcionar con esas compresiones, pero rápidamente da lugar a la llegada de DeltaMate, IndyMatte entre otros pero que no llegarán a ningún lugar hasta 2002 donde Syverson crea una versión de IndyMatte que altera la señal de las miniDV antes de procesarlas, permitiendo así trabajar con ellas.

En 2003 Syverson dijo que el proceso debería tener dos pasos y no uno como se hacía hasta ahora, en un primer paso obtener un recorte normal y en un segundo paso obtener el mismo recorte con más detalle, obteniéndolo de la luminancia para posteriormente solaparlos y obtener un detalle en los bordes superior al de sus antecesores, a partir de esa idea nace DVmatte Pro 1.5

En 2004 como viene siendo costumbre en años anteriores se busca un sistema cada vez más rápido y sobre todo simple, del que nace Master Keyer. Permite realizar el ajuste de color cogiéndolo directamente desde un pixel de la imagen, consiguiendo así un mejor ajuste de los parámetros de la llave.

Como se puede observar desde los principios del Chroma constantemente se ha intentado evolucionar y perfeccionar la técnica, intentando conseguir que la siguiente fuera más rápida, más estable, más sencilla, con más ajustes para que al usuario le sea más fácil conseguir lo que desea que con la anterior versión u otro software de otra compañía. Hasta el punto que en la actualidad no sólo se emplean las llaves para realizar la técnica sino que se acompañan de sistemas de rastreo de movimiento a los objetos en 3D para conseguir una sensación de realidad aún mayor en cada llave, porque no sólo tendremos la imagen en primer plano sino que logramos profundidad en el objeto. Obteniendo como resultado imágenes que en la mayoría de ocasiones es imposible diferenciar entre una imagen editada de una que no lo está [12].

2.2 SETUP CHROMA-KEY

En este apartado vamos a explicar de una forma algo más detallada un montaje práctico y que materiales son necesarios. Primero que nada tenemos que encontrar una ubicación donde disponer todo el material con que vamos a trabajar y una vez la tenemos localizada debemos pensar que necesidades tenemos para nuestra composición, para así decantarnos por una opción u otra.

Si lo que queremos es un pequeño estudio nos conformaremos con unas lonas que pueden ser portátiles, estas se sujetan con unos trípodes como la de la figura 5, pero si por lo contrario necesitamos más espacio lo más aconsejable es pintar las paredes y suelo que se vayan a emplear con unas pinturas especializadas para dicho fin como la de figura 6, ya que tendremos más garantías en el color, evitaremos brillos y obtendremos una mejor cobertura sobre la superficie.

Es aconsejable realizar pruebas con la ayuda de un medidor de luminancia o croma (figura 48) [22] y con la iluminación que dispondrá el estudio, así nos aseguraremos que los niveles de color se mantienen, consiguiendo así un Chroma-Key perfectamente ajustado a los niveles establecidos.



Figura 48. Medidor de Luminancia [22]



Figura 5. Set básico Chroma-Key. Google imágenes



Figura 6. Pintura Chroma-Key. Google Imágenes

La iluminación es una de las partes más importantes y en algunas ocasiones es la más abandonada o en la que menos se tiene en cuenta, sobre todo en los Chroma-Keys de bajo presupuesto.

Para obtener una buena iluminación necesitaremos en los focos una temperatura de color igual o superior a 5600° Kelvin (luz día) para evitar colorear el ciclorama como hemos comentado anteriormente. Estos dispondrán si es necesario de unos difusores (por ejemplo papel de cebolla más económico o difusores especializados que se ubican en la parte frontal de los focos) para así favorecer la uniformidad y dispersión máxima de la iluminación evitando que se marquen las sombras. La cantidad de focos va de un mínimo de tres a cuatro focos hasta un máximo dependiendo del presupuesto y necesidades [3], [13], [14].

La disposición para cuatro focos sería la siguiente. Dos directamente enfocados al ciclorama por delante y otros dos en dirección al objeto o personaje para evitar que este produzca sombras, en algunos casos se emplean reflectores para reforzar algunas zonas más oscuras sin la necesidad de otro foco.

Puede que se produzca alguna sombra indeseada o spill sobre el personaje, la solución para las sombras sería reubicar el foco que la provoca y para el spill o separar más al personaje u objeto del ciclorama o utilizar otros focos como contras que contrarresten esa coloración, para ello dispondremos de unos filtros en los focos de color magenta que es el opuesto directamente al color verde o color ámbar para foreground azul, en la figura 7 podemos hacernos una idea de cómo sería la configuración [4].

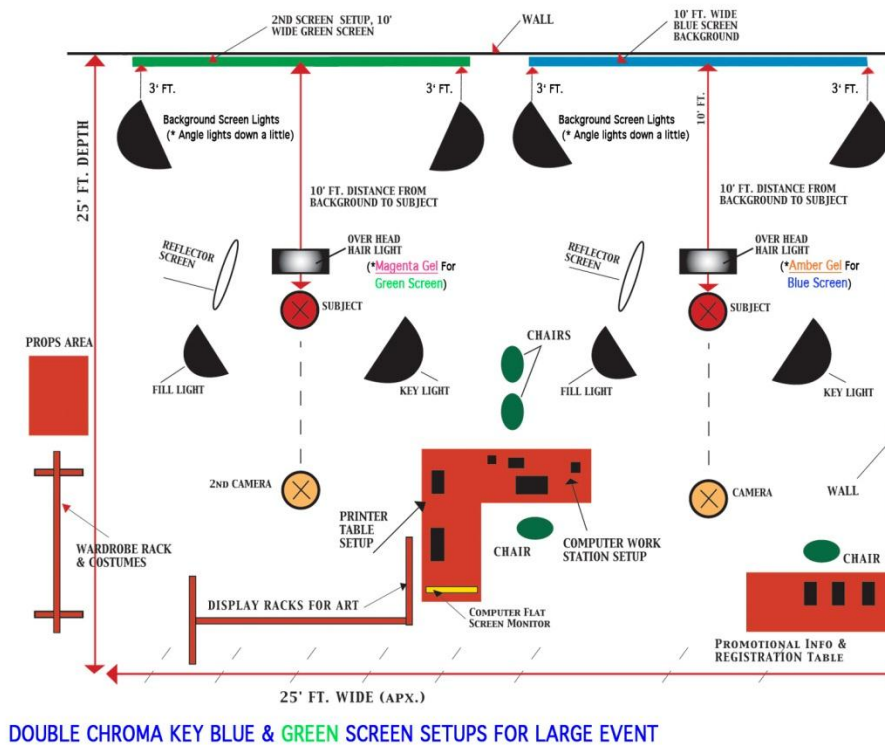


Figura 7. Iluminación para un set Chroma-Key. www.fxphoto.webs.com

Muchas son las dificultades que te puedes encontrar a la hora de realizar un Chroma-Key, sin olvidarnos también a la hora de realizar la captura con la cámara de emplear el mayor rango dinámico que se disponga, evitar la compresión de la señal y emplear la máxima resolución que esta permita para así reducir posibles problemas que puedan surgir a la hora de realizar el procesado, en pocas palabras trabajar con la mayor calidad visual que dispongamos con nuestros equipos.

Si no se tiene un control sobre cada uno de estos puntos a la hora de realizar el procesado de la imagen puede complicarse y hacerse muy tedioso.

2.2.1 Pantalla Retroreflectiva

Una empresa llamada Chromatte diseñó en 1990 una buena alternativa que simplifica en gran medida los problemas de iluminación y rapidez de montaje de un set [5].

Consiste en una lona compuesta por millones de cristales que se encargan de reflejar la luz que proviene de un anillo compuesto por Led's de color verde o azul (dependiendo de la necesidad) con potencias de entre 47 y 158 candelas que va ubicado en la cabeza de la cámara, en la figura 8 tenemos una imagen de la lona y en la figura 9 la composición interna de los cristales.



Figura 8. Lona retroreflectiva. www.reflecmedia.com

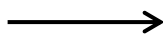


Figura 9. Composición interna de la lona. www.reflecmedia.com

El anillo va conectado a un controlador que se encarga de suministrar la energía necesaria para cada situación. En las figuras 10 y 11 podemos observar el anillo de Led's y los distintos componentes que componen el kit.



Figura 10. Anillo de Led's. www.reflecmedia.com



Figura 11. Conjunto del Kit. www.reflecmedia.com

Con este sistema se consigue reducir en gran medida el problema de iluminación del fondo y los posibles problemas de sombras e iluminación que pudiera ocasionar los movimientos de cámara, ya que del anillo es de donde proviene la luz que necesita la lona (ciclorama) para su perfecta iluminación.

A partir de esta interesante alternativa y de su disponibilidad en la escuela, nace la idea para nuestra tesina, del cual iremos hablando y explicando a lo largo de la misma.

2.2.2 Hardware/Software

Una vez tenemos la señal captada sólo quedará procesarla para su posterior edición, para ello disponemos de equipos Hardware y Software. Existe un gran variedad a la hora de elegir, pero la primera opción que viene a la cabeza es Ultimatte, una compañía con muchos años en el sector y dispone tanto de Hardware dedicado como plugins para programas de terceros.

En el caso del hardware existen varias opciones como el Ultimatte 11 SD/HD 4:2:2 como gama alta o el Ultimatte 500 gama más baja, emplean algoritmos matte para aumentar su precisión de separado del objeto, incorporan corrección de mattes, permiten realizar múltiples Chroma-Keys a la vez, 2 salidas de monitor y entradas/salidas en bucle entre otras características.

Otras alternativas hardware serían Tricaster 8000, Panasonic AW-HS50, DataVideo DVK-200 o la Safire HD 2 de Cristal Vision, todas permiten fuentes de alta definición, manipulación en directo, inserción de gráficos y platós virtuales, compatibles con los estándar SDI, entre otras características. En la figura 12 podemos ver los diferentes dispositivos [6].



Figura 12. De izquierda a derecha y arriba a bajo Ultimatte 11, Tricaster 8000, AW-HS5050, DVK-200, Safire HD 2

Luego están las alternativas software como podría ser After Effects con los Plugins Keylight 1.2, Primatte Keyer o Advant/Edge de la compañía Ultimatte, estos dos últimos son de los más potentes y empleados en el mundo del Cine y la Televisión para la realización de Chroma-Keys aunque no es el único software con esa capacidad porque también se podría realizar con Final Cut Pro Ultimatte/rt, Abode Premiere, Avid Media Composer, Nuke entre otros.

En dichos software dispondremos de múltiples configuraciones para ajustar lo máximo posible el recorte de la máscara y así obtener un buen Chroma-Key. Algunos de los parámetros a modificar serían **Screen Color** el cual define el color de la llave, **Screen Gain** se encarga de ajustar la cantidad de saturación del color del background para casos poco saturados, **Screen Balance** se encarga de ajustar el umbral de los diferentes tonos del background, de normal suele dejarse al 50% , **Spill** ajusta el derrame o borde contaminado por el fondo sobre el objeto, **Alpha** ajusta las transparencias y muchas otras configuraciones más que varían dependiendo del Plugin o software empleado según sea más o menos complejo. En las figuras 13 y 14 podemos ver como son las interface de Keylight y Primatte para After Effects.

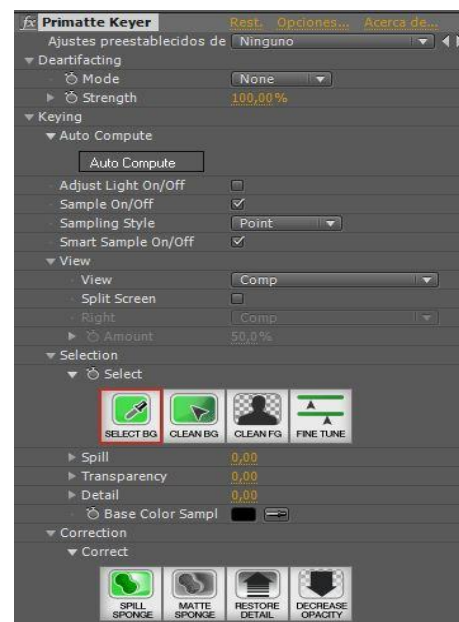
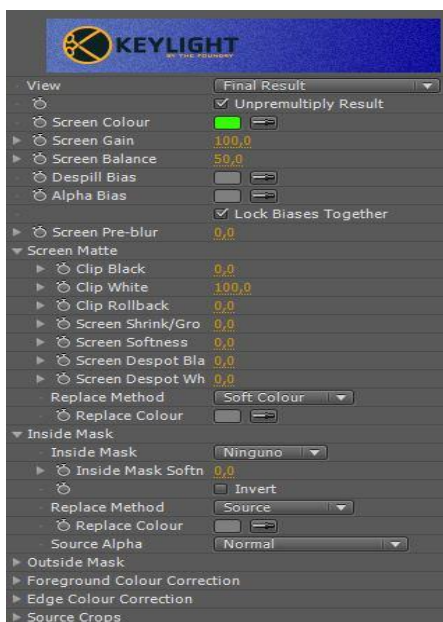


Figura 13. Keylight 1.2. Captura de pantalla de la aplicación

Figura 14. Primatte. Captura de pantalla de la aplicación

Una vez tenemos configurado los diferentes parámetros el programa realizará el cambio por el otro fondo automáticamente a falta de ajustes mínimos para conseguir un recorte más ajustado sobre el objeto. Esto es posible gracias a los algoritmos complejos que emplean.

En el caso del algoritmo de Primatte emplea un espacio 3D para generar poliedros formados por cada componente RGB. Básicamente lo que hace Primatte es separar los píxeles de la imagen en cuatro áreas diferentes, compuestas cada una por 128 superficies, las cuales pueden distorsionarse para ajustarse a cada color. Con la ayuda de la figura 15 nos podemos hacer una idea de cómo sería.

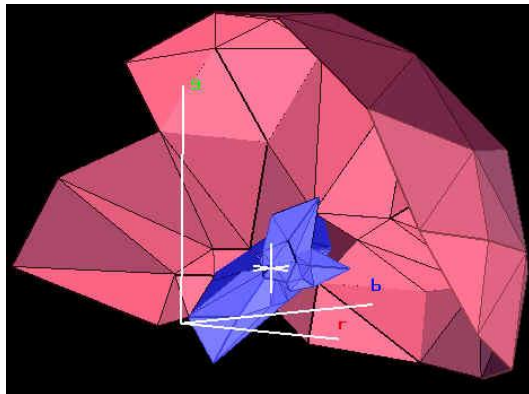


Figura 15. Poliedro Primatte. www.wikipedia.com

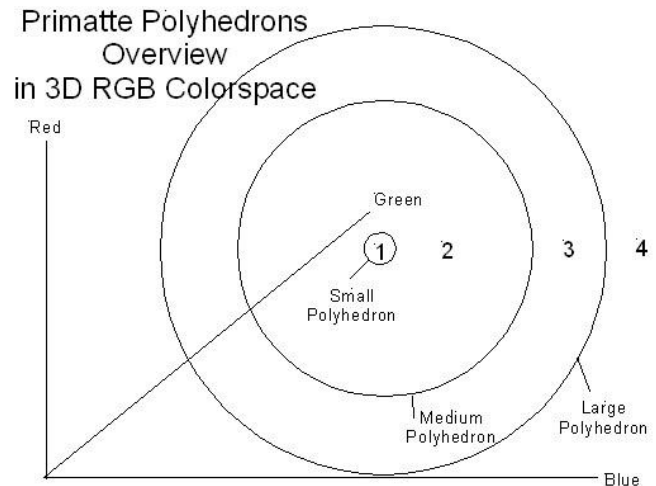


Figura 16. Anillos del algoritmo Primatte. www.wikipedia.com

Como puede verse en la figura 16 en la primera área hay un pequeño poliedro que contiene todos los píxeles de la imagen (completa), en la segunda área se ubican los píxeles de transición entre el fondo y el primer plano (define los bordes), en la tercera área nos encontramos con los poliedros medios-grandes que se encargan de la supresión del fondo sin spill y por último en la cuarta área la más grande en la cual se ubican los píxeles que no cambian de la imagen.

Con esta forma se tiene cada parte de la imagen completamente separada en áreas para su mayor optimización a la hora de trabajar con ella.

A diferencia de Primatte, Ultimatte emplea una mezcla de aditivos para combinar el primer plano con el fondo, pudiendo así obtener detalles muy sutiles, tales como son las transparencias de un cristal o el humo. Para ello emplean un Chroma-Key lineal, éste consiste en detectar el color llave e ir reduciéndolo de nivel hasta conseguir la combinación de primer plano y el fondo. Los algoritmos que emplean no los desvelan pero aseguran que son los únicos que consiguen reproducir objetos transparentes y sombras proyectadas mientras a su vez eliminan los posibles derrames que se puedan producir sobre el sujeto [7].

2.2.3 Limitaciones de Color

Todos sabemos a estas alturas que es un Chroma-Key y en que consiste, por tanto ahora podremos entender mejor cuando hablamos de las limitaciones que existen en el empleo del mismo color del vestuario que el de la llave. Si empleamos ese mismo color nos encontraremos con composiciones como la de la figura 17 en la cual podemos identificar que cuando se emplea el mismo color que en la llave el Chroma-Key no se realiza correctamente porque elimina zonas del foreground (en este caso el vestido de la presentadora) que en realidad no queremos que las elimine.



Figura 17. Ejemplo de las limitaciones de color. www.cassetteblog.com/2012/06/jessica-starr-la-chica-chroma/

Esto sucede porque el sistema lo identifica como fondo por ser de la misma tonalidad generando una máscara de recorte que no tiene en cuenta esa condición, por ello muchos han sido los que han intentado diseñar una alternativa con la que poder eliminar ese problema, a continuación podremos ver algunos de esos estudios.

2.4 SOLUCIONES PREVIAS

Como se ha podido observar las evoluciones y mejoras de la técnica año tras año son una realidad y que se sigue investigando para su mejora también es otra, por eso vamos a exponer algunos artículos sobre técnicas algo diferentes a lo visto hasta ahora ya que se basan en prescindir de un fondo de color uniforme y fijo como el que se emplea en la actualidad en la mayoría de Chroma-Keys para así conseguir eliminar las limitaciones de color. Pero en definitiva tienen el mismo fin, conseguir la combinación de un objeto A con un fondo B.

2.4.1 Cámara Térmica

2.4.1.1 Descripción

Consiste en un sistema compuesto dos cámaras, una RGB normal y otra sensible a la luz infrarroja. La cámara RGB se encarga de captar la imagen estándar y la infrarroja de captar el calor corporal del personaje con la que poder realizar la máscara para conseguir el remplazo por otro fondo, en la figura 18 podemos verlo.

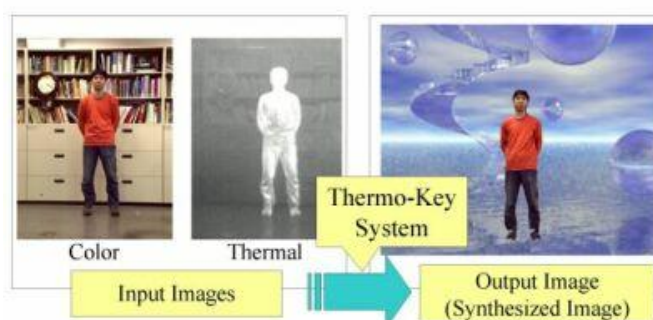


Figura 18. Chroma Térmico. www.nae-lab.org/project/thermo-key/

Para ello necesitan que las dos cámaras estén apuntando al mismo lugar exactamente, ello se consigue empleando un cristal entre ambas, este deja pasar la luz visible pero la infrarroja la refleja permitiendo que las dos capten la misma imagen, en la figura 19 nos podemos hacer una idea de cómo sería la configuración de las cámaras [8].

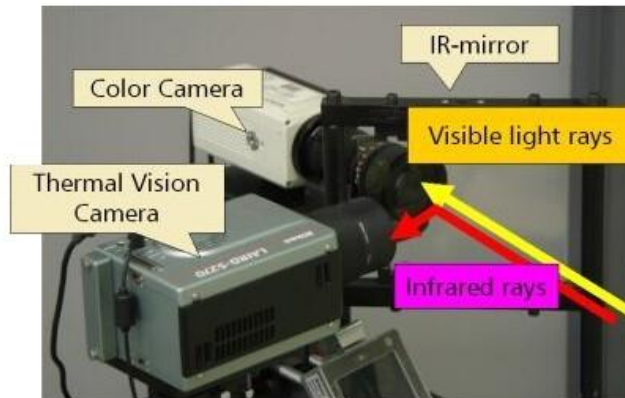


Figura 19. Configuración cámaras. www.nae-lab.org/project/thermo-key/

2.4.1.2 Ventajas y Limitaciones

Como podemos observar los resultados son muy buenos a la hora de obtener la máscara pero tiene un problema, se necesita un objeto que emita calor y se limita únicamente a seres vivos de sangre caliente que emitan calor y otro problema que puede surgir es con la iluminación del set, si se emplean focos de alta potencia puede llegar a emitir tanta calor que el sistema obtenga datos erróneos y no realice el remplazo con el fondo correctamente, aunque tendría fácil solución empleando iluminación fluorescente o LED ya que estas emiten muy poca calor. Además se requiere un equipamiento caro y específico (del que no se tiene habitualmente en un estudio de televisión).

2.4.2 Substracción de Fondo

2.4.2.1 Descripción

Consiste en la sustracción del fondo empleando técnicas de segmentado en los objetos de interés que tenga la escena, eso quiere decir que una vez tenemos la imagen capturada mediante procesado se detecta el objeto que nos interesa y se le sigue dejando el resto de la imagen como fondo a remplazar.

Para su implementación se siguen tres pasos, detección en primer plano, mantenimiento del objeto en movimiento y post procesado.

La detección se realiza de una forma muy simple, se capta la imagen sin objetos de interés (fondo) y luego se compara con la imagen que si tiene objetos de interés y mediante la resta y umbralización se obtienen los objetos (como el fondo no cambia y los objetos de interés sí, la detección de los mismos no se hace muy complicada).

$$|It - Bt| > \tau$$

$$Bt + 1 = \alpha It + (1 - \alpha)Bt$$

“It” es el objeto en el tiempo, “Bt” el fondo en el tiempo, “ τ ” es el umbral y “ α ” reduce el derrame.

Para mantener localizados únicamente los objetos de interés, localizar nuevos objetos y evitar lecturas erróneas, el sistema está sujeto a varios algoritmos que se encargan de ello y constantemente van actualizando el fondo.

$$Bt + 1 = It$$

Hay muchos más algoritmos para detectar el objeto y diferenciarlo del fondo sin introducir errores por cambios de luz, brillos y mantener el objeto aunque esté estático en el fondo. En la figura 20 podemos ver un ejemplo [9] [15].

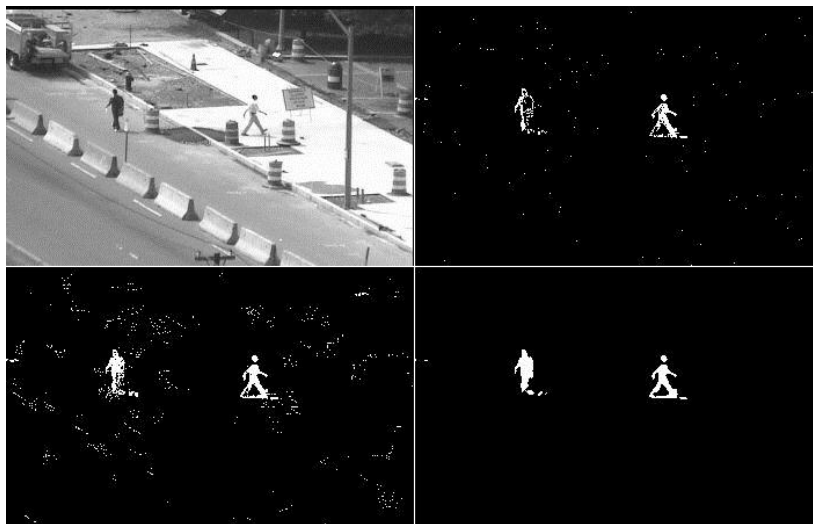


Figura 20. Máscaras para captura por sustracción [15]

2.4.2.2 Ventajas y Limitaciones

Nos encontramos ante otra alternativa que no emplea un fondo con color específico sino que mediante algoritmos de diferencia entre frames detecta el objeto y lo sigue. Es una idea muy buena pero debido a la gran cantidad de procesado que necesita cada imagen para compensar las posibles mediciones erróneas el tiempo de procesado es demasiado y el recorte de los objetos no es muy bueno por tanto no sería una buena opción para emplearlo como una alternativa.

Las posibles soluciones sería emplearlo en un ambiente más controlado como podría ser un estudio para evitar problemas con lecturas erróneas y así reducir los algoritmos y hacer más fluido el procesado.

2.4.3 Llave por Profundidad

2.4.3.1 Descripción

Nos encontramos frente a una alternativa muy interesante, consiste en emplear una cámara que es capaz de captar RGB y a su vez calcular las distancias a las que se encuentran los píxeles captados respecto al sensor en tiempo real, gracias a un sistema especial que incorpora.

El sistema genera un “muro de luz” con las dimensiones necesarias del campo de visión del sensor de la cámara, este “muro” es generado por unos láseres cuadrados de corta duración (intermitencias de 1nseg) que emiten luz no visual para evitar interferencias con el vídeo, estos están incorporados en la parte frontal como se puede ver en la figura 21.

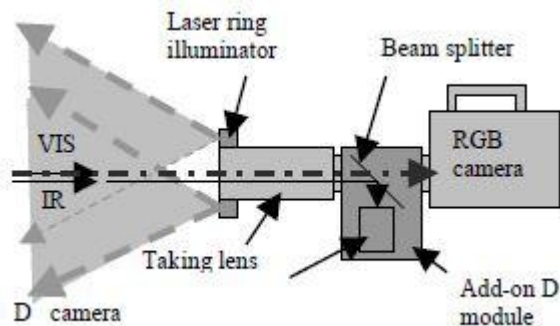


Figura 21. Configuración del sistema 3DV [16]

Conforme viaja la luz de los láseres se topa contra los objetos y esta retorna hacia la cámara con los datos necesarios de profundidad para cada píxel. Partiendo como condición que la profundidad es la inversa proporcional a la luz reflejada se podrán generar las diferentes máscaras a diferentes distancias.

Como puede verse en la figura 21 la luz de retorno a la cámara es tanto la visual como la no visual, evitando que el sensor CCD capte luz infrarroja de los láseres se ubica un espejo justo antes del sensor para desviarla y así dirigirla hacia el sistema que se encarga de calcular las profundidades de los píxeles.

La distancia a la que se proyecta el “muro” es ajustable para así tener un mayor control sobre los objetos que se quieran incluir o no. En la figura 22 puede verse un ejemplo claro donde el segundo y tercer personaje no entran dentro de la proyección del muro.



Figura 22. Sistema 3DV [16]

Como hemos comentado es capaz de reconocer diferentes distancias y generar diferentes máscaras como puede verse en la figura 23, donde se diferencia perfectamente el primer personaje (en primer plano) que tiene una máscara definida en blanco y la segunda (en segundo plano) en un tono gris.



Figura 23. Diferencia de profundidad en 3DV [16]

Cuando se generan las máscaras por primera vez (figura 24) puede que sean más pequeñas de lo que deberían en los bordes, esto es así porque la cantidad de luz que regresa de los objetos con ángulos es menor.



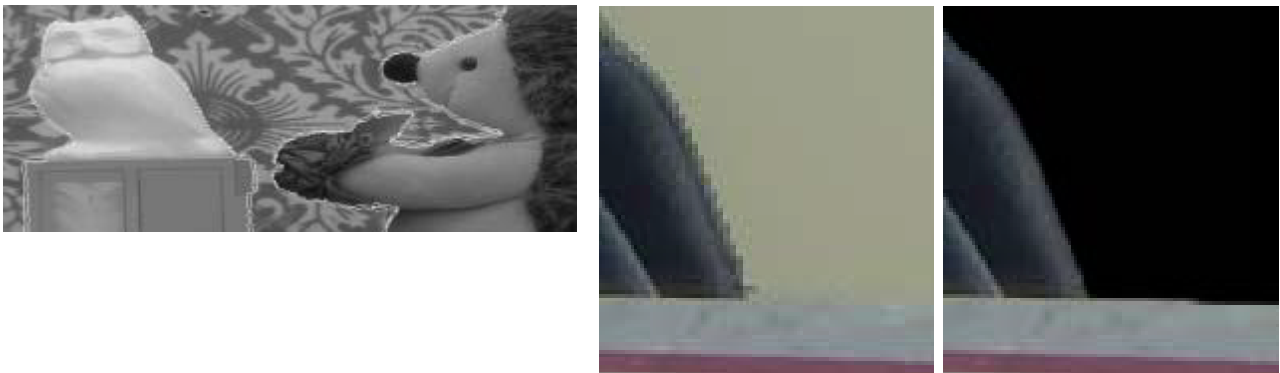
Figura 24. Máscara generada sin ajuste [16]

Para solucionar este problema se emplea el calculo de la distancia de color con los píxeles vecinos, donde D_i es la distancia entre p y q (espacio de color RGB), la raíz se emplea para reducir ruido de vídeo y los pesos entre P y sus vecinos se normalizan para que la suma sea siempre 8.

$$W_i = \frac{1}{\sqrt[4]{D_i}}$$

$$\alpha_p = \sum W_i \alpha_q$$

Se repetirán hasta que la distribución máscara es similar a las transiciones de color, de normal no se superan las 3 repeticiones y así obtener la máscara mejor recortada como las de las figuras 25 y 26 [16].



Figuras 25 y 26. Máscara ajustada [16]

2.4.3.2 Ventajas y Limitaciones

Nos encontramos frente a una técnica diferente a las demás ya que nos da la posibilidad de generar máscaras a diferentes distancias sin emplear ningún tipo de llave de color, por tanto se elimina por completo la limitación de los colores en los objetos u personajes, ampliando las alternativas de creatividad en las composiciones. También hay que mencionar que es muy eficiente siendo capaz de realizar a tiempo real las diferentes máscaras debido a la optimización que se dispone en los algoritmos, obteniendo unos resultados óptimos como pueden verse en las figuras anteriores.

Aunque la eficiencia se ve truncada cuando las imágenes son algo borrosas y el proceso es menos eficiente debido a que la transición entre áreas es más gradual y por tanto hay que aplicar más procesado. Y comentan que se podría mejorar el rendimiento empleando métodos Bayesianos.

2.4.4 Ciclorama Patrón

2.4.4.1 Descripción

Esta última alternativa está más relacionada con la idea de la tesina y la emplearemos para introducir algunas de las ideas que nosotros emplearemos.

Consiste en el empleo de un patrón definido como fondo que en este caso consiste en tablero de ajedrez pero con los colores verde y azul. Se emplea igual que los otros cicloramas pero a diferencia del resto la iluminación y la perfecta uniformidad del mismo no será un problema debido a que el sistema no busca un color definido como en los otros cicloramas sino que el sistema busca un patrón definido que en este caso es un tablero de ajedrez [17].

Por tanto con una iluminación básica o simplemente la del propio set será lo suficiente para identificar el patrón. En la figura 27 puede verse el ejemplo del patrón y ver como realiza el remplazo del fondo sin ningún problema, aun llevando el personaje en su vestuario los colores dedicados para las llaves en un Chroma-Key.

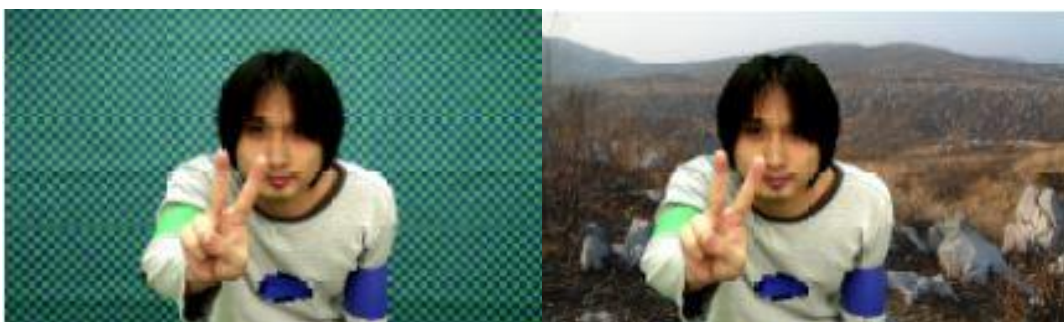


Figura 27. Ciclorama Patrón [17]

La obtención de la máscara respecto al patrón se obtiene mediante interpolación de los puntos del objeto, de esta forma conseguimos un buen recorte y definición del objeto u personaje. En la figura 28 podemos ver de una forma simple como sería.

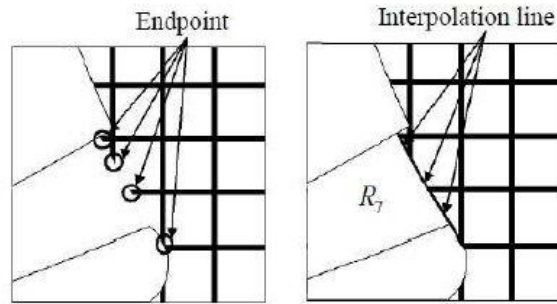


Figura 28. Interpolación de líneas [17]

El método consta de 4 pasos:

1. Detección del fondo empleando espacio de trabajo HSL para una mejor detección de los colores (figura 29).

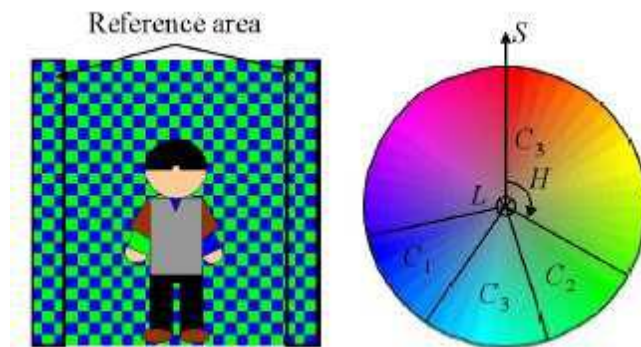


Figura 29. Detección de los colores fondo [17]

2. Obtención de la cuadrícula del fondo a partir de la intersección de los cuadros de diferente color, se empezará a generar la cuadrícula por el borde de la imagen para evitar problemas a la hora de crearlo (figura 30).

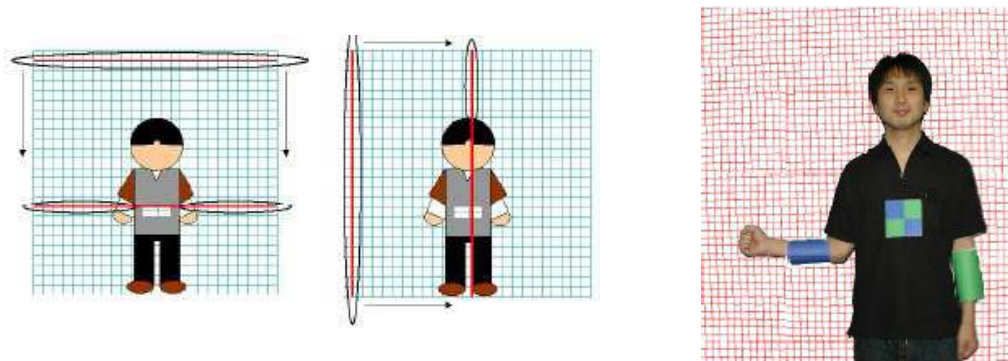


Figura 30. Detección de la cuadrícula [17]

- Extracción del primer plano, con las siguientes fórmulas aseguran conseguir extraer el primer plano sin problemas (figura 31).



$$E_{spline}(s_i) = \sum_{i=1}^n (w_{sp1} |s_i - s_{i-1}|^2 + w_{sp2} |s_{i+1} - 2s_i + s_{i-1}|^2),$$

$$E_{image}(s_i) = - \sum_{i=1}^n (w_{image} |\nabla I(s_i)|),$$

$$E_{area}(s_i) = \sum_{i=1}^n w_{area} \{u_i(v_{i+1} - v_i)$$

$$|\nabla I(s_i)| = \begin{cases} |I(u_i + 1, v_i) - I(u_i, v_i)| + |I(u_i, v_i + 1) - I(u_i, v_i)| & \text{if } (u, v) \notin R_3, R_4 \\ |I(u_i + 1, v_i) - I(u_i, v_i)| & \text{if } (u, v) \in R_3 \\ |I(u_i, v_i + 1) - I(u_i, v_i)| & \text{if } (u, v) \in R_4 \end{cases}$$

Figura 31. Extracción del primer plano (interpolación) [17]

- Composición de las imágenes, se empleará un canal alfa para obtener un degradado entre el recorte y el fondo y así conseguir una mejor unión entre background y foreground (figura 32).

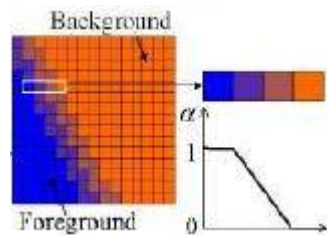


Figura 32. Unión del Background y Foreground [17]

2.4.4.2 Ventajas y Limitaciones

Como hemos podido ver es una alternativa muy original y relacionada con nuestro método como veremos posteriormente en el empleo de un patrón definido con el que generar la máscara. Con esta técnica consiguen eliminar por completo los problemas de iluminación en el set y también la eliminación de la problemática de emplear el mismo color en los objetos que en el fondo.

Pero tiene una pega debido a la gran cantidad de procesado para obtener la cuadrícula y la interpolación de todas las líneas para el recorte del objeto se hace un proceso algo lento y en imágenes en movimiento como podría ser un vídeo podría llegar a ser un proceso arduamente lento y sin obtener los buenos resultados que en imagen estática debido a que la cuadrícula estaría constantemente variando y ralentizaría aun más el proceso.

Con esta revisión se ha podido ir viendo cómo se encuentra la investigación en el tema del Chroma-Key y así enlazarlo con nuestra propuesta que intenta coger algunas ideas de estos métodos y mejorarlos en sus limitaciones. Como podría ser el empleo de un patrón definido o utilizar dispositivos como podría ser las pantallas retroreflectivas. Aunque existen más alternativas de composición sin restricciones de color como las de [18], [19] que suponen demostraciones más difíciles de implementar de forma práctica en un estudio de televisión.

CAPÍTULO 3.

DESARROLLO DE LA TÉCNICA PARA EVITAR LIMITACIONES DE COLOR

3.1 INTRODUCCIÓN

A lo largo de todo el trabajo hemos ido comentando en que consiste, cuales son las características y limitaciones de un Chroma-Key. También que es una técnica que se emplea en numerosas situaciones y que está en constante evolución como hemos podido ver en el punto anterior con la intención de mejorar el método o simplemente implementar uno completamente diferente pero con el mismo fin.

En la presente tesina se aborda una de las limitaciones presentes en los sistemas Chroma-Key: Las limitaciones en el vestuario, maquillaje o atrezzo y los problemas de obtener una iluminación correcta de una forma diferente pero con la misma intención, remplazar un fondo por otro en video.

El sistema consiste en generar mediante el anillo de la empresa Reflecmedia Chromatte que vimos en el punto 2.2.1, un patrón asíncrono que alterna entre dos intensidades (encendido 1, apagado 0) y que lo proyecte sobre la lona retroreflectiva para que la cámara lo capte y mediante procesado de la señal sea capaz de detectarlo y sustituirlo por otro material audiovisual.

3.2 PREPARACIÓN E INVESTIGACIÓN DE LA TÉCNICA

El sistema que proporciona Chromatte no incluye dicha función sino que simplemente dispone de un potenciómetro donde dar más o menos potencia a los LED's por lo que tendremos que fabricar un dispositivo conectado en serie con el anillo que sea capaz de controlarlo y generar dicho patrón.

Este dispositivo tiene un reducido coste, es de fácil construcción y simplifica la configuración a diferencia de las alternativas anteriormente vistas.

La idea de emplear las pantallas retroreflectivas es debido a sus características, ya que nos ofrece una señal con un nivel bajo de ruido, genera un color uniforme alrededor de toda la lona, es un sistema de fácil instalación y tiene un buen rendimiento. La uniformidad del color se consigue gracias a la gran cantidad de cristales microesféricos recubiertos de material reflectante que hay repartidos por todo el material de forma aleatoria en su orientación. En la figura 33 podemos observar como sería la configuración del set, que básicamente es igual que la de cualquier Chroma-Key con pantallas retroreflectivas pero añadiendo el circuito de control del anillo que genera el patrón.

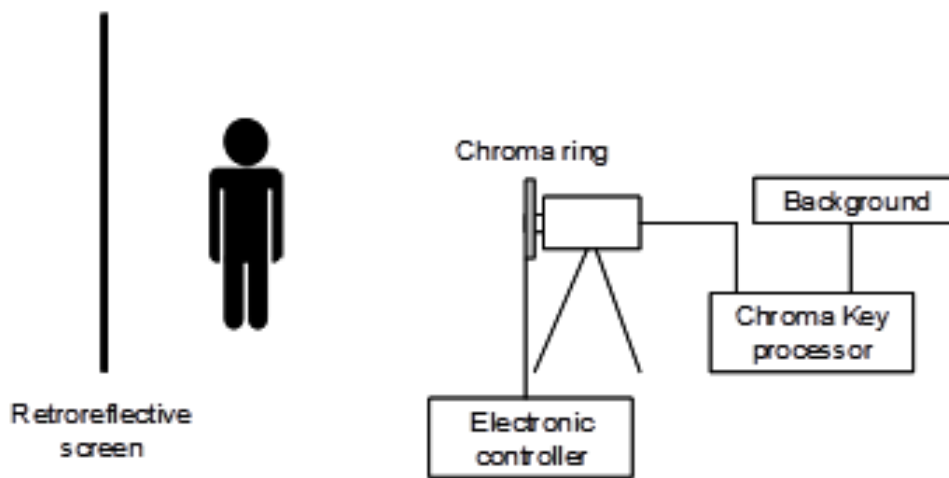


Figura 33. Setup del estudio para la obtención de datos

Las siguientes ecuaciones muestran las condiciones para realizar la Chroma-Key. Se trata de añadir condiciones adicionales para evitar la eliminación no deseada de partes de la imagen que tienen colores similares a los del fondo.

$$C(t) = \alpha(t)C_f(t) + (1 - \alpha(t))C_b(t) \quad (1)$$

$C(t)$ es el color compuesto en cada píxel que se calcula mediante la combinación lineal de C_f (foreground) y colores del fondo C_b y $\alpha(t)$ es el valor clave alfa

$\alpha(t)$ indica las variaciones del color del fondo, marcando los cambios de nivel de una imagen a otra en función del tiempo, $\alpha_{conventional}$ es el alfa matte tradicional

$$\alpha(t) = \alpha_{conventional}(t) \wedge [A_1(t) \vee A_2(t)] \quad (2)$$

$$A_1(t) = \begin{cases} |G_f(t) - G_f(t-1)| \geq Th_G \wedge |R_f(t) - R_f(t-1)| \leq Th_R, \\ \forall t \mid \frac{dG(t)}{dt} > 0 \\ |G_f(t) - G_f(t-1)| \leq Th_G \wedge |R_f(t) - R_f(t-1)| \leq Th_R, \\ \forall t \mid \frac{dG(t)}{dt} < 0 \end{cases} \quad (3)$$

$$A_2(t) = \begin{cases} |G_f(t+1) - G_f(t)| \geq Th_G \wedge |R_f(t+1) - R_f(t)| \leq Th_R, \\ \forall t \mid \frac{dG(t)}{dt} > 0 \\ |G_f(t+1) - G_f(t)| \leq Th_G \wedge |R_f(t+1) - R_f(t)| \leq Th_R, \\ \forall t \mid \frac{dG(t)}{dt} < 0 \end{cases} \quad (4)$$

$A_1(t)$ y $A_2(t)$ son el algoritmo principal, compara las variaciones de verde (G_f) del frame anterior (3) o posterior (4) respecto al frame que estamos, obteniendo el recorte sin limitación de color. Donde Th es valor de incremento/disminución del umbral y T es el período del patrón.

El aumento o disminución del color se utiliza para identificar los píxeles del fondo, ya que sólo la pantalla reacciona a los cambios de estado de los diodos LED's. Se aplica una condición adicional para pequeños cambios de la componente roja (R_f) y así asegurar que sólo el color clave es el que se reemplaza (3)-(4).

El tipo de patrón esta definido por el control electrónico porque por parte de los diodos LED's no hay problema porque son capaces de conmutar de estado en ns. Otro de los problemas que si que podría surgir es la velocidad de refresco de los LED's porque puede ser algo molesto para los actores, sobre todo cuando las intermitencias tienen un índice de modulación alto. Pero sin embargo para frecuencias superiores al umbral crítico del ojo humano pasará desapercibido.

Hemos realizado una serie de experimentos para probar la viabilidad de la técnica, dichas experiencias se han llevado a cabo en el plató de TV de la escuela con la configuración del set anteriormente mostrada en la figura 33. Se empleó una pantalla retroreflectiva conjuntamente con un anillo estándar de LED's verdes. Para la captura de vídeo empleamos una cámara de alta definición, una Canon XL H1 [9] y para el procesado una unidad NewTek Tricaster [5] con una resolución de proyecto 1920x1080i.

El controlador electrónico como podemos ver en la figura 34, es un sistema muy sencillo, económico y completamente asíncrono con el sistema, consta de un circuito impreso 555 [10] configurado en modo aestado, varias resistencias, condensadores y un interruptor con el que configurar el circuito a diferentes frecuencias de oscilación. En el Anexo 1 podemos ver una imagen del circuito y en el Anexo 2 una imagen de la disposición de las patillas del circuito impreso 555.

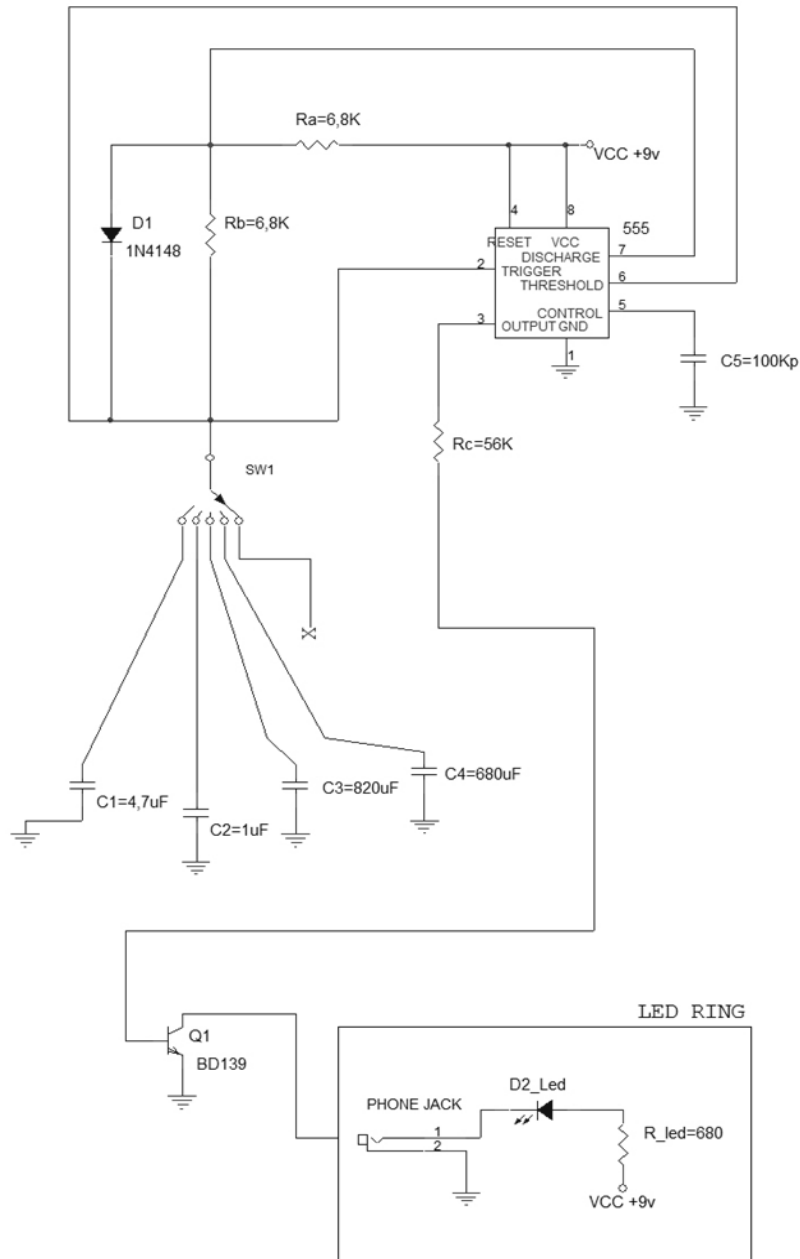


Figura 34. Diseño del circuito generador de patrón mediante Orcad9.2 Portable

A la entrada del mismo recibe una señal continua del propio controlador del anillo y a la salida obtendremos una señal cuadrada con la que alimentamos el anillo y nos proporcionará el patrón deseado. En la figura 35 podemos ver la gráfica de la señal que recibe el anillo (a), la señal que capta la cámara (b) y la muestra de luz en el transcurso de encendido/apagado (c).

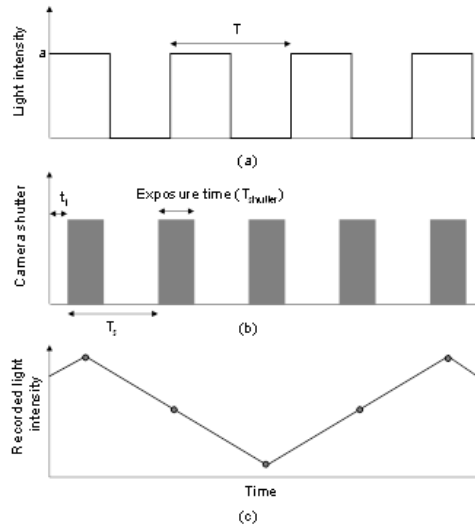


Figura 35. Señal recibida por el anillo, nivel captado por la cámara y evolución de la luz durante la conmutación

Con el interruptor podemos obtener señales cuadradas de $T=44\text{ms}$ (22Hz); $T=9,5\text{ms}$ (106Hz); $T=7,7\text{ ms}$ (130Hz) y $T=6,4\text{ms}$ (156Hz), pero si combinamos los 4 interruptores podremos obtener diferentes señales cuadradas proporcionales a los interruptores activados. En el Anexo 3 podemos ver un ejemplo de simulación del controlador electrónico en funcionamiento.

Cuando realizamos las pruebas toda frecuencia por debajo de los 50Hz resultaba molesta para la persona que se coloca delante del anillo debido a un excesivo parpadeo.

$$y[n] = \int_{t_1+T_s}^{t_1+nT_s+T_{shutter}} \sum_{n=-\infty}^{+\infty} a \prod(t - kT) dt \quad (5)$$

Con la ecuación (5) obtenemos la intensidad de luz en el fotograma n , donde t_1 es el tiempo de desplazamiento entre la luz y la velocidad de muestreo de la cámara, T_s es el periodo de muestreo de la cámara (inversa a la velocidad de cuadro), $T_{shutter}$ es el tiempo exposición de la cámara (inversa velocidad obturación), a es la intensidad de la luz, \prod es la forma de onda cuadrada y T periodo de la onda de luz.

Suponiendo que el circuito está generando una señal cuadrada al 50% de ciclo de trabajo y $T_{shutter} < T/4$ se podría expresar a partir de la ecuación (5) de la siguiente forma:

Donde $x=t_1+(n-1) T_s \bmod T$

(6)

$$y[n] = \begin{cases} 0, & x < \frac{T}{4} - T_{shutter} \\ a \left(x - \frac{T}{4} + T_{shutter} \right), & \frac{T}{4} - T_{shutter} < x < \frac{T}{4} \\ aT_{shutter}, & \frac{T}{4} < x < \frac{3T}{4} - T_{shutter} \\ -ax + \frac{3T}{4}a, & \frac{3T}{4} - T_{shutter} < x < \frac{3T}{4} \\ 0, & x > \frac{3T}{4} \end{cases}$$

En la figura 36 se puede apreciar la evolución del verde, rojo y azul en dos píxeles para una frecuencia de cuadro de 98Hz, donde podemos ver una señal triangular debido al submuestreo cuadrado por la cámara (con velocidad de obturación 1/210). También podemos observar pequeños cambios en la componente azul y por eso como hemos comentado anteriormente en las funciones la componente roja se comprueba para asegurar que el cambio de color se debe a la pauta temporal y no a movimientos de la escena.

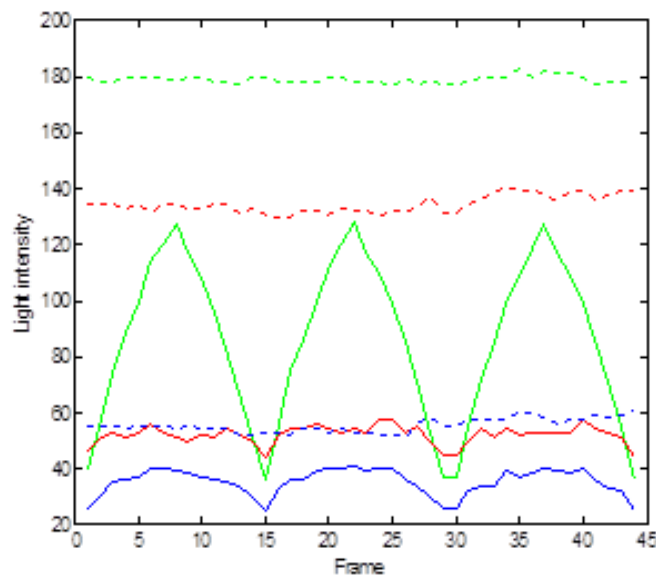


Figura 36. Niveles de señal en componentes RGB captados por la cámara en varios frames

Una vez tenemos todas las condiciones se pasa a obtener un código de procesado en Matlab para así poder obtener la máscara.

En la figura 37 podemos observar los diferentes niveles de verde que nos podríamos encontrar dentro de un archivo de video, los cuales el programa es capaz de localizar y reemplazar. Para comprobar que con las condiciones de las ecuaciones anteriores se permiten extraer el fondo sin limitaciones de color, se ha añadido una cartulina con varios tonos de verde lo más próximos posible al color del fondo para confirmar que queda completamente excluido de la composición (no depende del color sino del patrón).



Figura 37. Diferentes niveles de verde que puede generar el patrón

3.3 DESARROLLO DEL ALGORITMO

3.3.1 Descripción

A continuación vamos a explicar paso a paso cada punto del código Matlab que hemos empleado para realizar la técnica y así si existe alguna duda pueda quedar mejor explicada.

Primero que nada para identificar cada una de las partes que componen el sistema hasta conseguir el resultado final vamos a mostrar el siguiente diagrama de bloques (figura 38) que tiene como función explicar a groso modo los puntos principales del funcionamiento del código. El código completo lo podemos encontrar en el Anexo 4.

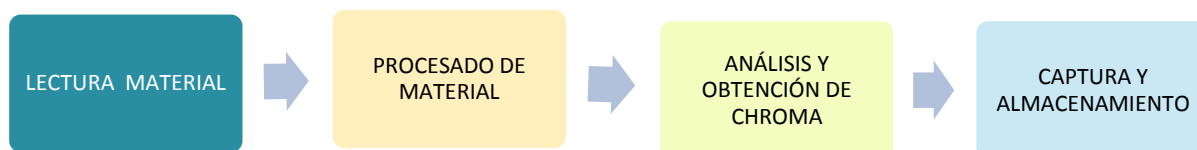


Figura 38. Diagrama de bloques de los diferentes puntos del código

En el apartado de “Lectura Material” nos vamos a encontrar con los diferentes comandos que hemos empleado para cargar los vídeos. Para ello hemos empleado la función “**mmreader**” la cual convierte el vídeo en un objeto del cual podemos obtener información del mismo como su resolución o número de frames y para poder extraer dicha información emplearemos la función “**get**”. Seguidamente inicializaremos las variables de los bucles que recorrerán cada frame del vídeo y crearemos la matriz donde se almacenará el vídeo resultante con la función “**moviein**”. En las figuras 39 y 40 podemos ver capturas del material desde la aplicación Matlab.



Figura 39. Captura de pantalla del Foreground empleado para la composición

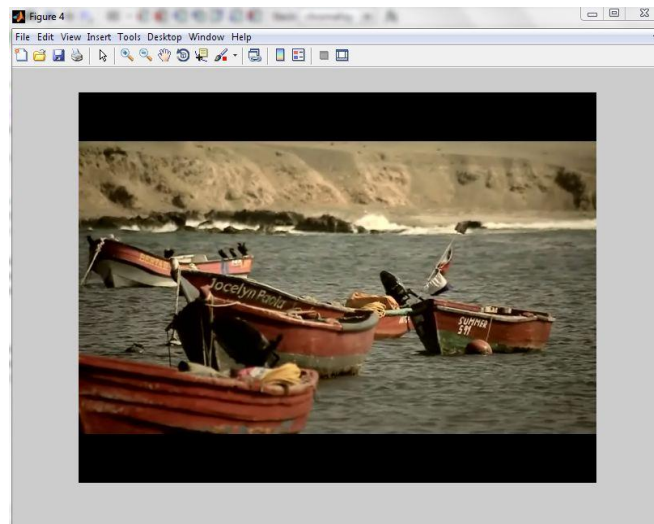


Figura 40. Captura de pantalla del Background empleado para la composición

Seguimos con el apartado de “Procesado de Material” en dicho punto nos encontramos con la inicialización de los bucles uno para el almacenamiento de los frames procesados y obtención del vídeo final, en el cual emplearemos un “for” que irá desde 1 hasta el número de frames del mismo y otro será el bucle encargado de obtener el material necesario en cada pasada para su posterior análisis, para este emplearemos un bucle “while” que irá desde el frame $k=2$ hasta el número de frames del vídeo (se puede limitar restándole frames).

Se empieza desde el frame 2 porque para realizar el análisis necesitamos de tres frames en cada pasada donde poder comparar las diferencias de verde (patrón) del frame anterior y posterior. Para la obtención de los frames empleamos la función “**read**” que nos devolverá el frame que nosotros le digamos y sumaremos 3 a la variable k para que en la siguiente pasada del bucle obtengamos los 3 frames siguientes con los que poder seguir comparando. En este mismo punto con la misma función “**read**” obtenemos los frames del vídeo background y los reescalaremos si es necesario con la función “**imresize**”, por último daremos valores a las constantes de ajuste para la obtención de las máscaras.

El apartado de “Análisis y obtención de Chroma-Key” es el encargado de crear las máscaras de recorte a partir de condiciones que compararán los diferentes frames con las diferentes componentes de un frame, que en nuestro caso son R, G ó B (Rojo, Verde o Azul). En el caso de la máscara1 nos encontraremos con la condición de realizar un simple Chroma-Key, en la máscara2 que es una combinación con la máscara3 nos mostrará el ajuste de color por diferencia de verde, la cual se encarga de eliminar de la (máscara1) todos los niveles de verde que no tengan variación, dejando así una máscara que sólo eliminará el fondo que varía de tonalidad de verde.

En la máscara3 se introduce un nivel de verde definido (podría estar definido por le usuario) y se le aplica posteriormente una función de reajuste llamada “**dilate_mult**” que su función consiste en rellenar los posibles píxeles que puedan quedar sin seleccionar del ajuste de verde para conseguir así un Chroma-Key más homogéneo. Una vez tenemos generada la máscara final se almacena en dos variables $ImFu$ y $ImFm$ donde se almacenará la mascara normal y la invertida para luego poder realizar la composición matricialmente. Durante el análisis se muestra los histogramas de cada frame diferenciados por componentes, donde se podrá observar que los niveles de rojo y azul son siempre iguales mientras que el verde va variando y así poder ver como el patrón en el color verde va variando en cada frame y para ello hemos empleado la función “**imhist**”.

Una vez es procesado el material y analizado obtenemos las máscaras que empleamos en la composición, como se puede observar en la figura 41 obtenemos tres máscaras la primera es la máscara de un Chroma-Key normal donde se puede ver que la zona verde sobre el cuerpo se elimina por ser verde, en la segunda al aplicar los algoritmos de las ecuaciones descritas anteriormente permite obtener una tercera máscara como la suma de las dos anteriores donde se aprecia que únicamente elimina el fondo (2)-(4).

En la figura 42 podemos observar los histogramas de los niveles de Rojo, Verde y Azul del foreground que empleamos en la técnica. Su principal función es poder observar las variaciones de verde que va sufriendo el fondo debido al patrón generado por el circuito eléctrico y así entender mejor que gracias a esas variaciones podemos generar una máscara que no dependa del verde sino de las variaciones del mismo.

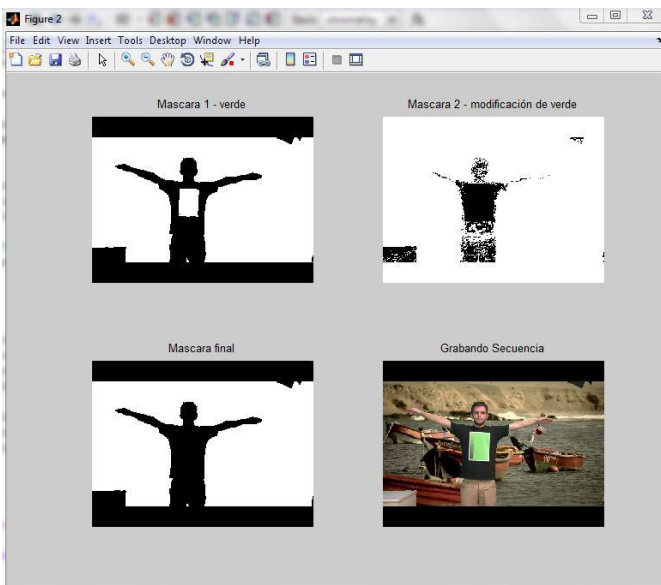


Figura 41. Captura de pantalla de las Máscaras empleadas para la composición

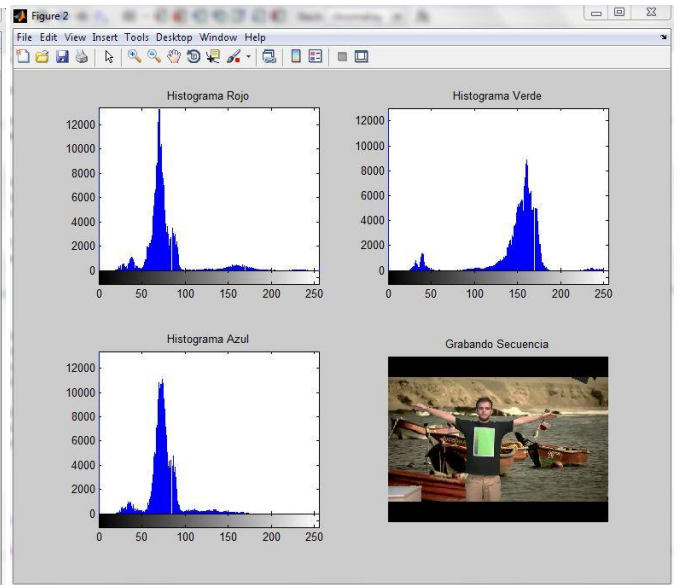


Figura 42. Captura de pantalla de los Histogramas RGB de la composición

Por último llegamos al punto de “Captura y Almacenamiento” que como bien dice es el punto del programa donde se captura los frames de la composición finalizada con **“getframe”** que su función es coger el ultimo frame generado y almacenarlo en una variable que posteriormente servirá para que pueda ser reproducido por la función **“movie”**. Por último empleamos la función **“movie2avi”** para convertir la secuencia en un archivo .AVI y lo almacenaremos con la función **“save”** y así se podrá reproducir desde cualquier programa externo a Matlab.

Mostraremos por pantalla los histogramas y las máscaras de la última imagen para poder observar los niveles de RGB resultantes y como son las máscaras generadas por las condiciones creadas en el punto de “Análisis y Obtención de Chroma-Key”. Por último la composición quedará y será capturada como se puede ver en la figura 43 (es posible reproducir la secuencia al pulsar sobre al figura en su versión electrónica).



Figura 43. Captura de pantalla de la composición final. Pulsa en la figura para reproducir la composición final

3.2.1 Resultados

Después de todo el proceso de investigación que hemos realizado a lo largo de esta tesina podemos afirmar que nos encontramos frente una técnica completamente factible para ser implementada y utilizada en composiciones audiovisuales de una forma simple y de bajo coste, con la que eliminar por completo la problemática que existe hoy en día en el vestuario del mismo color que la llave empleada en el Chroma-Key.

Al introducir esta opción en los Chroma-Key la creatividad artística no se verá truncada como pasa ahora, por tanto creemos que es una buena alternativa a emplear ya que en la mayoría de sets disponen de los materiales necesarios para poder realizar la técnica, aunque conseguirlo no sería un desembolso muy elevado para la empresa interesada a diferencia de las otras alternativas que hemos estudiado, que son más complejas de implementar y más costosas económicamente hablando.

3.2.2 Ventajas e Inconvenientes

La ventaja que tiene esta técnica es la eliminación de la problemática de emplear el mismo color en el vestuario que en la llave del Chroma-Key, aunque hay otras alternativas que también solucionan el problema pero ninguna tan simple de implementar, configurar y adaptar al set como la nuestra. No supondría ningún gran desembolso a la empresa que esté interesada ya que el circuito generador de patrón se puede conseguir por unos pocos euros.

Otra principal baza que tiene esta alternativa es que es completamente asíncrona con el set de Chroma-Key, simplemente lo añades a tu set y con la aplicación pertinente consigues eliminar las limitaciones de color. Por lo que se evita el coste de un circuito de sincronización y se simplifica el trabajo en el estudio al no necesitar cables de sincronía.

Por el contrario hay que mencionar algunos inconvenientes que dispone, al no estar sincronizada con los otros equipos en algunas ocasiones podrían producirse algunos pequeños errores en la composiciones como podría ser que algunas pequeñas zonas estén parcialmente remplazadas, aunque con algún pequeño ajuste en el código estaría solucionado.

Otra desventaja sería que debido al procesado que se necesita, la obtención de la composición se ralentiza más que cuando se emplea un Chroma-Key normal. Este tiempo extra que suma la aplicación puede ser reducido optimizando más el código y eliminando información intermedia que el programa ofrece como podría ser muestras por pantalla de las máscaras o histogramas.

3.3 DISEÑO e IMPLEMENTACIÓN GUI

Una vez hemos conseguido generar una composición óptima con las dos fuentes (foreground y background) es necesario crear una aplicación para así unificar de alguna manera toda la información y poder visualizar de una forma más dinámica todo el proceso y ayudar a entender mejor la técnica. La aplicación consta de una interfaz gráfica con varias ventanas donde poder ver como se genera la composición, la máscara de recorte, los histogramas y la carga del material (Foreground, Background).

3.3.1 Descripción

Para desarrollar dicha interfaz gráfica nos hemos decantado por la aplicación interna que dispone Matlab, se llamada GUI (Graphic User Interface), tiene una implementación fácil y combina perfectamente con el código ya implementado. La aplicación genera dos archivos uno .m y otro .fig en el primero se guarda el código y en la segunda todo el entorno gráfico diseñado desde la aplicación. En la figura 44 podemos ver como es la aplicación.

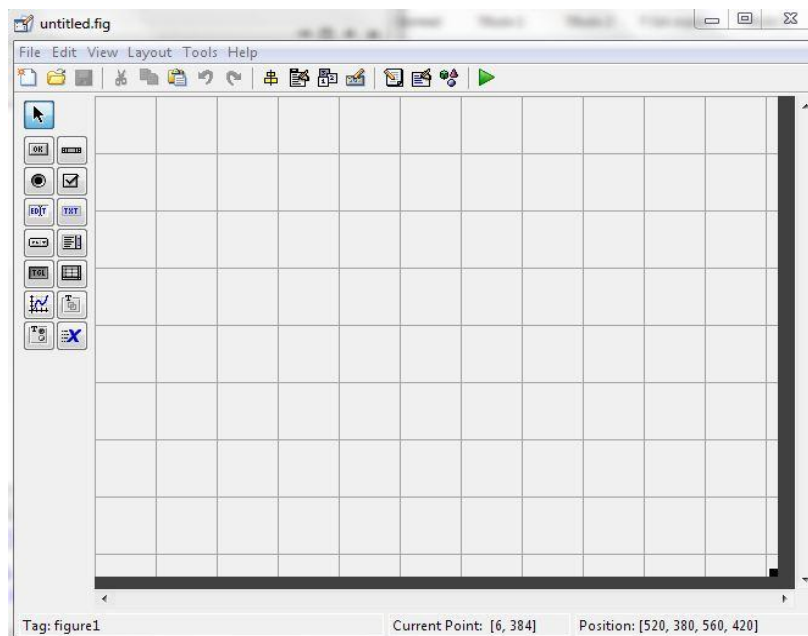


Figura 44. Captura de pantalla del entorno gráfico Guide

Como puede verse en la figura 44 el programa dispone de una zona editable (cuadrícula) que puedes cambiar de tamaño para generar una interface más o menos grande.

También podemos observar los diferentes botones con los que poder diseñar el entorno gráfico a nuestro gusto desde la inserción de texto editable o fijo, gráficas con las que mostrar imágenes, botones, deslizadores entre otros. En la figura 45 podemos verlos de cerca.



Figura 45. Captura de pantalla del entorno gráfico Guide, Botonera de edición

Cada uno de estos dispone de una configuración donde cambiarlo de tamaño, color, nombre entre otras muchas más configuraciones. Esto se puede configurar con la ventana inspector que se activa pulsando dos veces sobre cualquier ventana, pulsador, texto editable o deslizador. En la figura 46 podemos ver la ventana inspector.

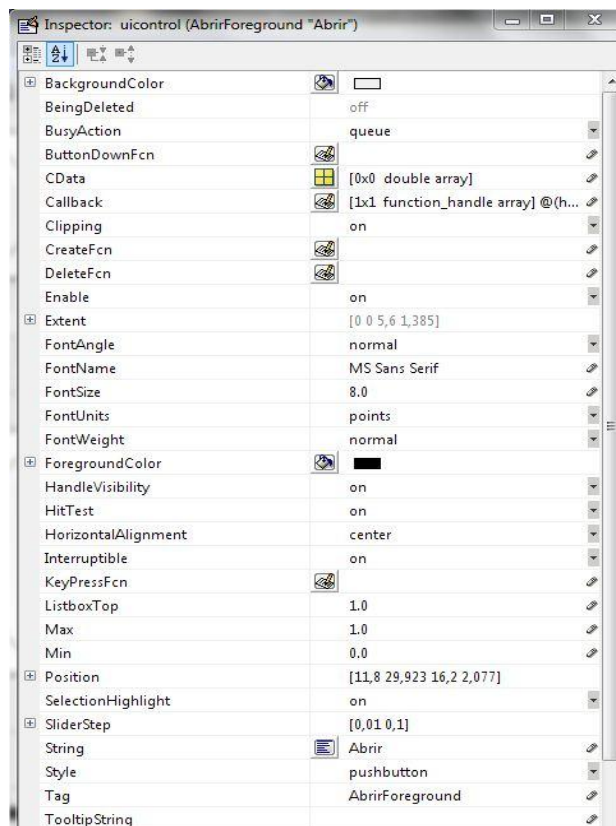


Figura 46. Captura de pantalla del entorno gráfico Guide, Ventana Inspector.

3.3.2 Implementación del código y del entorno gráfico

El código que hemos empleado es el mismo que el del Anexo 4 pero adaptado para que la aplicación GUIDE sea capaz de interpretar toda la información necesaria para su posterior procesado. Para ello nos hemos valido de los comandos **“uigetfile”** con el que poder abrir archivos desde una carpeta, **“axes (handles.axes)”** indica donde se tiene que mostrar la información en la ventana definida del entorno gráfico y los comandos **“handles.Nomarchivo=Nombarchivo”** y **“guidata(hObject,handles)”** con los que almacenar variables de una función para emplearlas en otras funciones.

El entorno gráfico que hemos implementado es muy básico y simple como se puede ver en la figura 47, está destinado a mostrar la información de una manera más dinámica que como se puede mostrar por Matlab. Intentando conseguir mayor interacción con el usuario y que así pueda llegar a entender mejor la técnica.

En el entorno gráfico se puede ver la ventana “Fuentes” donde nos encontraremos con los videos “Foreground” y “Background” previamente cargados desde el botón “Carga Foreground o Cargar Background”. En la ventana “Histogramas” podemos observar las gráficas de cada componente de color (Rojo, Verde, Azul) con los que poder visualizar los cambios de verde a lo largo de la reproducción del “Foreground”. En la ventana “Máscara” como bien indica su nombre podremos obtener la máscara obtenida a partir del patrón. En la ventana “Composición Final” encontraremos la unión del “Foreground” y el “Background” para formar el Chroma-Key mediante el botón “Generar Secuencia”. Cuando termina el proceso se genera un archivo de vídeo como resultado de la composición, que podrá ser reproducido desde el botón “Reproducir Secuencia”. El código implementado de la interface gráfica, se encuentra en el Anexo 6.

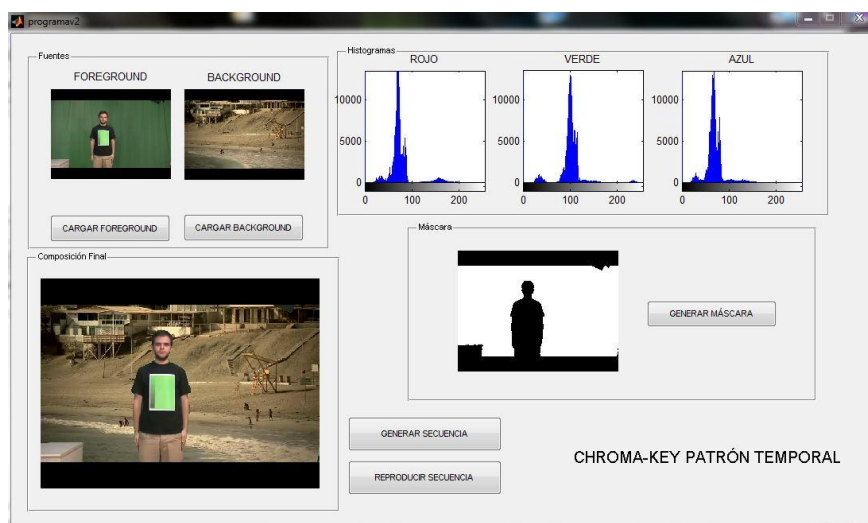


Figura 47. Captura de pantalla del entorno gráfico Guide Implementado.

4. EVALUACIÓN Y CONCLUSIONES

Con la elaboración de esta tesina se ha demostrado que tenemos una nueva técnica con la que evitar las limitaciones en los colores que se pueden presentar en una escena de Chroma-Key. Dicha técnica la hemos implementado en con la interfaz gráfica de Matlab 2008b, el material de vídeo con el que hemos trabajado ha sido grabado en un estudio de televisión, aplicándole el algoritmo obtenemos una imagen compuesta sin presentar problemas con la presencia en la escena del color usado como llame en el Chroma-Key.

Después de la investigación podemos decir que es perfectamente factible su implementación debido a las mejoras que este introduce a la técnica del Chroma-Key como punto positivo, pero como punto negativo queremos decir que introduce algo de retardo en el procesado del Chroma-Key pero que podría ser solventado optimizando el código empleado en Matlab y también mejorar los algoritmos introduciendo más variables para así conseguir un mejor recorte sobre el objeto en primer plano y obtener un Chroma-Key lo más perfecto posible.

Nos encontramos ante una técnica muy simple de implementar a un Chroma-Key evitando las restricciones de color e iluminación. La técnica está basada únicamente en la electrónica que compone el controlador electrónico porque el resto es material que actualmente se está utilizando en la industria del mundo audiovisual.

De esta forma también intentamos mostrar que nos encontramos frente a una alternativa viable, económica (no más de 10€ en componentes electrónicos) y muy simple de implantar en las composiciones que actualmente se están haciendo pero ganando un punto en la creatividad debido a la eliminación de la limitación del color y a la menor problemática en la iluminación, demostrando que es posible evitar esas restricciones sin configuraciones excesivamente complejas.

Queremos mencionar en última instancia que toda la información que hemos mostrado en esta Tesina de Master en lo que respecta a nuestra técnica ha sido aceptada para su implementación como artículo en SMPTE MOTION IMAGING JOURNAL y se encuentra a la espera de publicación. En el Anexo 6 se encuentra el texto enviado.

5. BIBLIOGRAFÍA

- [1] http://en.wikipedia.org/wiki/Chroma_key
<http://www.taringa.net/posts/info/913402/La-magia-del-Chroma-Key-FX.html>
<http://www.edicionmania.com/index.php?name=News&file=article&sid=60&theme=Printer>
<http://www.edicionmania.com/index.php?name=News&file=article&sid=60>
(5/8/2012)
- [2] <http://www.caborian.com/2009/11/page/2/>
http://www.loresdelsith.net/3po/rep/c_blue.htm
http://es.wikipedia.org/wiki/Charge-coupled_device
http://en.wikipedia.org/wiki/Bayer_filter
(8/8/2012)
- [3] http://www.digitalanarchy.com/demos/chroma_studio.html
http://www.digitalproducer.com/2003/12_dec/tutorials/12_22/cw_ckeypat_tips.htm
(10/8/2012)
- [4] <http://fxphoto.webs.com/chromakeylightingtips.htm> (12/8/2012)
- [5] <http://www.reflecmedia.com/education/products/chromatte/index.htm>
<http://vimeo.com/28390219> (11/8/2012)
- [6] <http://www.chromakey-forum.com/showthread.php?39-Hardware-Realtime-Chromakeyers>
<http://www.jigsaw24.com/product-details/g089asa/datavideo-dvk-200-chroma-keyer>
http://www.ultimatte.com/UltimatteMain/11_Main.html
<http://www.newtek.com/products/tricaster-8000.html>
http://pro-av.panasonic.net/en/sales_o/switcher_mixer/aw-hs50/index.html
<http://www.crystalvision.tv/products/safirehd2.html>
(11/8/2012)

- [7] http://en.wikipedia.org/wiki/Primatte_chroma_key_technology#The_Algorithm
http://www.ultimatte.com/UltimatteMain/Ultimatte_vs_ChromaKey.html
(13/8/2012)
- [8] <http://nae-lab.org/project/thermo-key/>
(15/8/2012)
- [9] <http://www-staff.it.uts.edu.au/~massimo/BackgroundSubtractionReview-Piccardi.pdf>
(15/8/2012)
- [10] http://www.canon.es/For_Home/Product_finder/Camcorders//professional/XL_H1S/
(16/8/2012)
- [11] http://es.wikipedia.org/wiki/Circuito_integrado_555 (17/8/2012)
- [12] http://www.fxguide.com/featured/art_of_keying/ (17/8/2012)
- [13] R. Brinkmann, "The Art and Science of Digital Composing", Ed. Elsevier, 2008
- [14] R.C. Kennedy, F.J. Gaskins, "Electronic Composites in Modern Television", Ed. Proceedings of the IRE, vol. 46, no. 11, pp. 1798-1807, Nov. 1958.
- [15] Alan M. McIvor, "Background Subtraction Techniques", Reveal Ltd PO Box 128-221, Remuera, Auckland, New Zealand, 2000.
<http://www-staff.it.uts.edu.au/~massimo/BackgroundSubtractionReview-Piccardi.pdf>
- [16] Ronen Gvili, Amir Kaplan, Eyal Ofek and Giora Yahav, "Depth keying", Proc. SPIE 5006, 564 (2003)
<http://research.microsoft.com/en-us/um/people/eyalofek/depth%20key/depthkey.pdf>
- [17] H. Agata, A. Yamashita, T. Kaneko, "Chroma Key using a CheckerPattern Background", IEICE Trans. Inf & Syst, vol. E90-D, no. 1, January 2007
- [18] P. Vlahos, "Comprehensive Electronic Compositing System". US Patent 4,100,569, July 1978.

[19] Yoshio Monjo, "Chroma key system for discriminating between a subject and a background of the subject based on a changing background color". US Patent 6,490,006 B1, Dec 2002.

[20] <https://www.bksts.com/secure/about.asp>

[21] <http://patft.uspto.gov/netacgi/nph-Parser?Sect2=PTO1&Sect2=HITOFF&p=1&u=%2Fnetahhtml%2FPTO%2Fsearchbool.html&r=1&f=G&l=50&d=PALL&RefSrch=yes&Query=PN%2F5032901>

[22] <http://sensing.konicaminolta.com.mx/products/t-10-t-10m-illuminance-meter/>

Software Empleado:

Matlab R2008b (Implementación del código)

Orcad 9.2 Portable (Diseño del circuito generador de patrón)

Adobe Photoshop CS4 (Edición de imágenes)

Adobe Premiere CS4 (Edición y recorte de los vídeos almacenados por Tricaster)

Adobe After Effects CS4 (Recreación del sistema generador de patrón)

Adobe Acrobat 9 (Reproducción de archivos PDF)

Format Factory 2.6 (multimedia file converter)

Microsoft Word 2010 (Redactar TFM)

6. ANEXOS

ANEXO 1 - Imagen del circuito eléctrico que genera el patrón.

La siguiente figura se muestra el circuito electrónico encargado de realizar el control de la iluminación del anillo de LED's que posteriormente será proyectado en la lona retroreflectiva.



Figura 49. Circuito eléctrico generador del patrón

ANEXO 2 - 555 multivibrador aestable

En la siguiente figura mostramos el esquema eléctrico del integrado 555, para conocer cual es su interconexión en el circuito eléctrico.

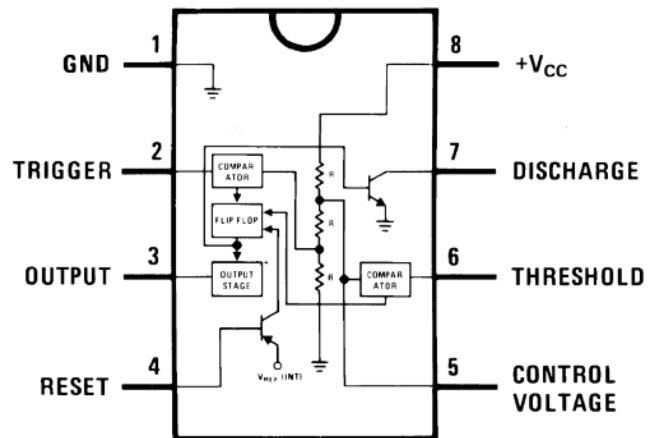


Figura 50. Multivibrador 555

ANEXO 3 - Simulación del funcionamiento del sistema generador de patrón,

En la siguiente figura podemos observar una simulación generada por After Effects cs4 del funcionamiento del anillo y el circuito eléctrico con el que se genera el patrón.

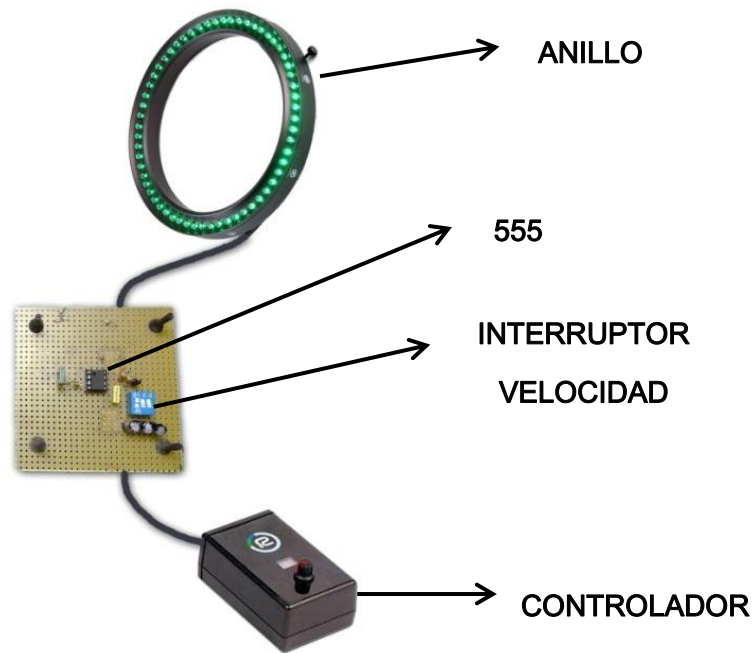


Figura 51. Simulación del funcionamiento del sistema. Si pulsa sobre la imagen se reproduce un archivo multimedia donde se puede visualizar el funcionamiento

Anexo 4 - Código completo del sistema implementado en Matlab.

```

function chromaKey
tic %cálculo tiempo
%LECTURA MATERIAL=====

background=mmreader('ATARDECER.avi'); %lee el video
foreground=mmreader('foreground6.avi'); %lee el video
numFrames=get(foreground,'numberOfFrames'); %calcula el número de frames
k=2; %inicialización frame foreground
F=moviein((numFrames-75)/3); %inicializar la matriz video
T=1245; %inicialización del frame del background

%PROCESADO DE MATERIAL=====

for j=1:((numFrames-75)/3)
while k<=numFrames-75 %quitamos frames para hacer el video más corto

ima=read(foreground,k-1); %guarda el frame anterior a "k"
imb=read(foreground,k); %guarda el frame "k"
imc=read(foreground,k+1); %guarda el frame siguiente a "k"
k=k+3; % suma 3 a la variable "k" para que las siguientes comparaciones
sean los siguientes 3 frame

imd=read(background,T); %carga en imd el frame marcado por T
if size(imd)~=size(ima)
imd=imresize(imd, [size(foreground,1) size(foreground,2)]);%Cambia el tamaño de background por el de foreground
end

T=T+10; %suma acumulada de frames para el fondo

a2=1.018; %constantes de ajuste
a22=0.92;

%ANÁLISIS Y OBTENCIÓN DE CHROMA-KEY=====

mascara1=imb(:,:,2)>imb(:,:,3)& imb(:,:,2)>imb(:,:,1)& imb(:,:,2)>65; % máscara normal

mascara2=(uint8(imc(:,:,2)))>=(uint8(ima(:,:,2))*a2)|(uint8(imb(:,:,2))*a2)<=(uint8(imc(:,:,2))&(uint8(imb(:,:,2))*a2
)<(uint8(ima(:,:,2))|(uint8(imb(:,:,2))<=(uint8(ima(:,:,2))*a22)|
(uint8(imb(:,:,2))*a22)>=(uint8(imc(:,:,2))|(uint8(ima(:,:,2))*a22)>(uint8(imc(:,:,2)))));
% cambio de verde con el tiempo

mascara3=imb(:,:,1)>115 & imb(:,:,2)>225 & imb(:,:,3)>115; %color definido por el usuario
mascara3=dilate_mult(~mascara3,13); %ajuste de relleno del color definido por le usuario
mascara2=mascara2.*mascara3;

mascara= mascara1&mascara2 ; %máscara final

imFu = cast(mascara, 'uint8'); %genera una tabla con los datos de máscara compatible con
ima,imb,imc
imFm = cast(~mascara, 'uint8');

```

```
imCk2(:,:,1) = imb(:,:,1).*imFm + imFu.*imd(:,:,1); %realiza el Chroma-key en cada componente de color
imCk2(:,:,2) = imb(:,:,2).*imFm + imFu.*imd(:,:,2);
imCk2(:,:,3) = imb(:,:,3).*imFm + imFu.*imd(:,:,3);
```

%CAPTURA Y ALMACENAMIENTO=====

```
figure(1),imshow(imCk2);
F(j)=getframe; %obtiene el frame y lo almacena
j=j+1; %suma 1 a la variable que aumenta un frame de la película

figure(2),
subplot(2,2,1);imhist(imb(:,:,1));title('Histograma Rojo') %Histograma separado por componente del foreground
subplot(2,2,2);imhist(imb(:,:,2));title('Histograma Verde')
subplot(2,2,3);imhist(imb(:,:,3));title('Histograma Azul')
subplot(2,2,4);imshow(imCk2);title('Grabando Secuencia')

end
close;
end
toc %fin cálculo tiempo
title('Reproducción Secuencia')
movie(F); %reproduce la secuencia
movie2avi(F, 'Secuencia'); %pasa la secuencia de movie a Avi llamando al archivo Secuencia
save Secuencia %almacena ese vídeo en el directorio

figure;
subplot(1,3,1);imhist(imCk2(:,:,1));title('ROJO')% histograma separado por componente de la imagen final
subplot(1,3,2);imhist(imCk2(:,:,2));title('VERDE')
subplot(1,3,3);imhist(imCk2(:,:,3));title('AZUL')

figure;
subplot(2,2,1);imshow(mascara1);title('Mascara 1 - verde') %se muestran las diferentes máscaras
subplot(2,2,2);imshow(mascara2);title('Mascara 2 - modificación de verde')
subplot(2,2,3);imshow(mascara);title('Mascara final')
```

Anexo 5 – Código adaptado para GUIDE

```

function varargout = programav2(varargin)
% PROGRAMAV2 M-file for programav2.fig
%   PROGRAMAV2, by itself, creates a new PROGRAMAV2 or raises the existing
%   singleton*.
%
%   H = PROGRAMAV2 returns the handle to a new PROGRAMAV2 or the handle to
%   the existing singleton*.
%
%   PROGRAMAV2('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in PROGRAMAV2.M with the given input arguments.
%
%   PROGRAMAV2('Property','Value',...) creates a new PROGRAMAV2 or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before programav2_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to programav2_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help programav2

% Last Modified by GUIDE v2.5 21-Aug-2012 17:01:55

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @programav2_OpeningFcn, ...
                  'gui_OutputFcn',  @programav2_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before programav2 is made visible.
function programav2_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to programav2 (see VARARGIN)

```

```

% Choose default command line output for programav2
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes programav2 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = programav2_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in CargarForeground.
function CargarForeground_Callback(hObject, eventdata, handles)
% hObject handle to CargarForeground (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

archiVideo=uigetfile('*.avi','Abrir');
if archiVideo==0
    return
end
VideoF=mmreader(archiVideo);
numFramesF=get(VideoF,'numberOfFrames');
handles.vidforeground=VideoF;
guidata(hObject,handles);

for K=1 : (numFramesF-75)
    foreground=read(VideoF,K);
    axes(handles.axes1)
    imshow(foreground);title('FOREGROUND')
    rx=foreground(:,:,1);
    gx=foreground(:,:,2);
    bx=foreground(:,:,3);
    axes(handles.axes3)
    imhist(rx);title('ROJO')
    axes(handles.axes4)
    imhist(gx);title('VERDE')
    axes(handles.axes5)
    imhist(bx);title('AZUL')
end

% --- Executes on button press in CargarBackground.
function CargarBackground_Callback(hObject, eventdata, handles)
% hObject handle to CargarBackground (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
archiVideo=uigetfile('*.avi','Abrir');
if archiVideo==0
    return
end

```

```

end
VideoB=mmreader(archiVideo);
handles.vidbackground=VideoB;
guidata(hObject,handles);
numFramesF=240;
P=1245;

for j=1 : ((numFramesF-75)/3)
    background=read(VideoB,P);
    axes(handles.axes2)
    imshow(background);title('BACKGROUND')
    P=P+7;
end

% --- Executes on button press in ObtenerMascara.
function ObtenerMascara_Callback(hObject, eventdata, handles)
% hObject      handle to ObtenerMascara (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

numFramesF=get(handles.vidforeground,'numberOfFrames');

k=2;
a2=1.018;
a22=0.92;
while k<= (numFramesF-75)
    ima=read(handles.vidforeground,k-1);
    imb=read(handles.vidforeground,k);
    imc=read(handles.vidforeground,k+1);
    k=k+3;

    mascara1=imb(:,:,2)>imb(:,:,3) & imb(:,:,2)>imb(:,:,1) & imb(:,:,2)>65;

mascara2=(uint8(imc(:,:,2)))>=(uint8(ima(:,:,2))*a2) | (uint8(imb(:,:,2))*a2)<=(uint8(imc(:,:,2))&(uint8(imb(:,:,2))*a2)<(uint8(ima(:,:,2))) | (uint8(imb(:,:,2)))<=(uint8(ima(:,:,2))*a22) |
(uint8(imb(:,:,2))*a22)>=(uint8(imc(:,:,2))) | (uint8(ima(:,:,2))*a22)>(uint8(imc(:,:,2)))) ;
    mascara3=imb(:,:,1)>115 & imb(:,:,2)>225 & imb(:,:,3)>115;
    mascara3=dilate_mult(~mascara3,13);
    mascara2=mascara2.*mascara3;

    mascara= mascara1&mascara2 ;

    axes(handles.axes6)
    imshow(mascara);
end

% --- Executes during object creation, after setting all properties.
function axes6_CreateFcn(hObject, eventdata, handles)
% hObject      handle to axes6 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes6

% --- Executes on button press in GenerarComposicion.

```

```

function GenerarComposicion_Callback(hObject, eventdata, handles)
% hObject    handle to GenerarComposicion (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

numFramesF=get(handles.vidforeground,'numberOfFrames');
k=2;
a2=1.018;
a22=0.92;
T=1245;
for j=1 : ((numFramesF-75)/3)
    while k<= (numFramesF-75)

        ima=read(handles.vidforeground,k-1);
        imb=read(handles.vidforeground,k);
        imc=read(handles.vidforeground,k+1);
        k=k+3;
        imd=read(handles.vidbackground,T);      %carga en la variable imd el frame
        marcado por T
        if size(imd)~=size(ima)
            imd=imresize(imd, [size(handles.vidforeground,1)
                size(handles.vidforeground,2)]);%Cambia el tamaño de background por el
            de foreground
        end
        T=T+10;

        mascara1=imb(:,:,2)>imb(:,:,3) & imb(:,:,2)>imb(:,:,1) & imb(:,:,2)>65;
        mascara2=(uint8(imc(:,:,2)))>=(uint8(ima(:,:,2))*a2) | (uint8(imb(:,:,2))*a2)<=
            (uint8(imc(:,:,2))) & (uint8(imb(:,:,2))*a2)<(uint8(ima(:,:,2))) | (uint8(imb(:,:,
            2)))<=(uint8(ima(:,:,2))*a22) |
            (uint8(imb(:,:,2))*a22)>=(uint8(imc(:,:,2))) | (uint8(ima(:,:,2))*a22)>(uint8(i
            mc(:,:,2))) ;
        mascara3=imb(:,:,1)>115 & imb(:,:,2)>225 & imb(:,:,3)>115;
        mascara3=dilate_mult(~mascara3,13);
        mascara2=mascara2.*mascara3;

        mascara= mascara1&mascara2 ;
        imFu = cast(mascara, 'uint8');
        imFm = cast(~mascara, 'uint8');

        imCk2(:,:,1) = imb(:,:,1).*imFm + imFu.*imd(:,:,1);
        imCk2(:,:,2) = imb(:,:,2).*imFm + imFu.*imd(:,:,2);
        imCk2(:,:,3) = imb(:,:,3).*imFm + imFu.*imd(:,:,3);
        axes(handles.axes7)
        imshow(imCk2);
        F(j)=getframe;          %obtiene el frame anterior y lo almacena
        j=j+1;
    end
end
movie2avi(F, 'Secuencia');
save Secuencia

% --- Executes on button press in ReproducirSecuencia.
function ReproducirSecuencia_Callback(hObject, eventdata, handles)
% hObject    handle to ReproducirSecuencia (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
imshow('Secuencia1.avi')

```

Anexo 6 - Paper aceptado en la revista SMPTE MOTION IMAGING JOURNAL

Chroma Key without Color Restrictions based on an Asynchronous Temporal Background Pattern on a

Borja Vidal, *Member, IEEE*, Juan Alberto Lafuente

Retroreflective Screen

Abstract— A simple technique to avoid color limitations in filming based on chroma key video composition using retroreflective screens and light-emitting diodes (LED) rings is proposed and demonstrated. An asynchronous temporal pattern is applied to the chroma ring and through simple image processing, the background color can be substituted by any image without the usual restrictions on scene colors for still and moving objects in the foreground. The technique removes technical constrains in stage composition allowing its design purely based on artistic grounds using off-the-self components widely used in TV and filming studios and low-cost electronics.

Index Terms—Video production, chroma-key, image composition, video segmentation, matting.

INTRODUCTION

CHROMA KEY video composition is a widespread technology in filming and video production used to combine a background image with video captures of real objects [1-5]. This technique offers good performance, cost savings by avoiding exterior filming and expensive scenery as well as versatility to integrate real and computer-generated shots while requiring no complex infrastructure. Due to these advantages chroma key is employed not only in the film industry but also in other areas such as press, education, gaming, and augmented reality ([4,6]).

Conventional chroma key is based on filming a real object in front of a monochromatic even surface known as cyclorama. This image provides an input into a chroma key processor which removes the cyclorama color and substitutes it by the desired background image (which can be a still image or moving video of real objects or a computer-generated image). Thus, the

chroma key processor combines two images in such a way that they can be keyed to a different background by replacing the color key of the real image with the background image. After chroma processing, the composite image is broadcast or recorded.

At first, any color can be used for the cyclorama provided it is not present in the foreground. Initially, blue was the main choice since it is complementary to human skin and thus the replacing of the chroma key color with the background image can be eased. When digital cameras expanded in the studios, green became more popular since it requires less light in the background and is less common in costumes. Thus, blue and green are the most common colors employed although other options, as magenta for example, have been used.

Although a part of everyday filming and television studios, chroma key present the problem that the background color, or a similar one, cannot be present in the foreground since areas of this color will be removed. This result in restrictions in the design of costumes, props, makeup, etc which limit artistic creativity.

To ease this problem, modifications to the conventional chroma key system have been presented. Among others these include a nonparametric sampling-based technique [7], the use of a two-tone checker pattern background [8], synchronized flash keying [9-10], and synchronized video projections and illumination for luma keying [12]. In [12] it is shown for still images that if the image is shot with two background colors (triangular matting), it is possible to extract a foreground containing any color. A technique to extend this approach to process moving objects was proposed in [13]. By adding a flatpanel monitor, a backprojection screen or a softbox to the setup, it is shown that a synchronized color sequence can be displayed in the background and, by doubling the frame rate and averaging two subsequent frames, color spill can be neutralized while the system can operate with any

foreground color.

In this Letter, work is reported to further simplify chroma keying systems without restrictions on foreground colors. Instead of a synchronized system [9-12], a technique to eliminate color restrictions based on an asynchronous background illumination alternating between two background intensities and a retroreflective screen is presented. An asynchronously electronic-controlled LED ring (chroma ring) is used to project a temporally alternating color pattern onto a retroreflective screen. In this approach color spill is inherently reduced due to the properties of the retroreflective material. Unlike previous work, this technique only introduces a simple electronic controller plus extra processing while the rest of the components and infrastructure are those already present in a chroma key system. Thus, it simplifies the setup in comparison to previous proposals and reduces cost while allows chroma keying with any color in the scene.

PRINCIPLE OF OPERATION

For chroma key segmentation a bright monochromatic surface is need. During the last years has become increasingly common to use a screen made of retroreflective fabrics which reflects the light of a chroma ring (typically a ring of LEDs mounted around the lens designed to illuminate retroreflective screens) to generate a good key signal with low noise. That approach offers a uniform surface color as well as a quick deployment. Additionally, thanks to its highly directive retroreflexion, color spill from the background is reduced in comparison to diffuse screens.

Retroreflective materials are typically made of small glass beads with a reflective coating on half the surface or corner-cube prisms. For chroma key applications [4] a cloth coated with a large number of microscopic randomly-oriented half-silvered microsphere glass beads is usually employed. Random orientation of the glass beads offers a higher uniform reflectivity for any angle of incidence in comparison with corner-cube reflective materials.

In this Letter, it is shown that besides its easy deployment, low cost and good performance, a retroreflective screen can be used to avoid color restrictions in video segmentation based on chroma key by adding a simple electronic controller.

The technique is based on using a temporal pattern on the background in such a way that a processing algorithm can identify changing pixels with the expected pattern as background.

Figure 1 shows a diagram of the proposed setup which is basically a standard recording setup for chroma key filming based on retroreflective screens. A chroma ring is attached to the camera and the actors are shot against a retroreflective screen. To implement the temporal background pattern from this setup it is only needed to

add a simple control circuit to control the light intensity emitted by the chroma ring and include some additional processing in the Chroma Key unit.

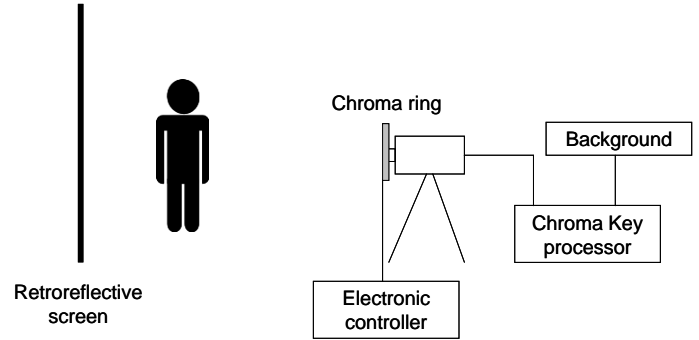


Fig. 1.- Schematic diagram of the proposed technique.

The composite color (C) at each pixel for frame t is computed from a linear combination of foreground (C_f) and background (C_b) colors as given by,

$$C(t) = \alpha(t)C_f(t) + (1 - \alpha(t))C_b(t) \quad (1)$$

where $\alpha(t)$ is the alpha key value or alpha matte which represents the transparency of each pixel. The color of each pixel can be represented as $C_i = [R_i; G_i; B_i]$.

Instead of the conventional alpha matte based on the level of a certain color from a single image, in the proposed technique this information is combined with a second alpha matte. In this new alpha matte pixels where the color of the constant background changes from one image to the next are marked as background. Taking green as the background color (key color), the final alpha matte can be expressed as:

$$\alpha(t) = \alpha_{conventional}(t) \wedge [A_1(t) \vee A_2(t)] \quad (2)$$

where $\alpha_{conventional}(t)$ is the traditional alpha matte algorithm, and

$$A_1(t) = \begin{cases} |G_f(t) - G_f(t-1)| \geq Th_G \wedge |R_f(t) - R_f(t-1)| \leq Th_R, \\ \forall t \mid \frac{dG(t)}{dt} > 0 \\ |G_f(t) - G_f(t-1)| \leq Th_G \wedge |R_f(t) - R_f(t-1)| \leq Th_R, \\ \forall t \mid \frac{dG(t)}{dt} < 0 \end{cases} \quad (3)$$

$$A_2(t) = \begin{cases} |G_f(t+1) - G_f(t)| \geq Th_G \wedge |R_f(t+1) - R_f(t)| \leq Th_R, \\ \forall t \left| \frac{dG(t)}{dt} \right| > 0 \\ |G_f(t+1) - G_f(t)| \leq Th_G \wedge |R_f(t+1) - R_f(t)| \leq Th_R, \\ \forall t \left| \frac{dG(t)}{dt} \right| < 0 \end{cases} \quad (4)$$

where Th_i with $i = R, G, B$ is the color increase/decrease threshold for green, red and blue, respectively, and T is the period of the temporal pattern. An increase or decrease of the key color is used to identify background pixels since only the retroreflective screen will follow the evolution of the chroma ring. An additional condition, based on small changes in the red components, is included to ensure that only the key color is changing and avoid identification errors due to movements in the scene.

The rate of the temporal pattern is given by the electronic controller. This rate could be limited by the switching speed of the chroma key ring but since they are made of standard LEDs which show switching times of the order of ns there is not restriction from this side. Another restriction comes from the fact that the LEDs rate has to be fast enough to avoid discomfort to actors from the blinking illumination, especially in situations where actors look directly at the camera with the attached chroma ring, since it can be really unpleasant especially when light is switched on/off with a high modulation index. However, rates above the critical flicker frequency of human vision strongly reduce the perception of the temporal pattern.

In the proposed technique, no synchronization is needed. Although a system where background illumination is synchronized to the camera is also possible, using no synchronization simplifies the setup and reduces cost, cable connections and deployment time.

EXPERIMENTAL RESULTS

To validate the feasibility of the proposed technique some experiments have been carried out in a TV studio using the setup of Fig. 1. A retroreflective screen by Reflectomedia is used jointly with a standard chroma key ring made of green LEDs. For video recording a HD camera (model Canon XL H1) and a NewTek Tricaster production unit have been employed to capture 1920x1080i video sequences. A very simple circuit (whose estimated cost is under 10 dollars) based on a 555 timer integrated circuit, shown in Figure 2, has been used to control light emission by the chroma ring. It provides a square signal to feed the chroma ring with a configurable frequency by means of a switch.

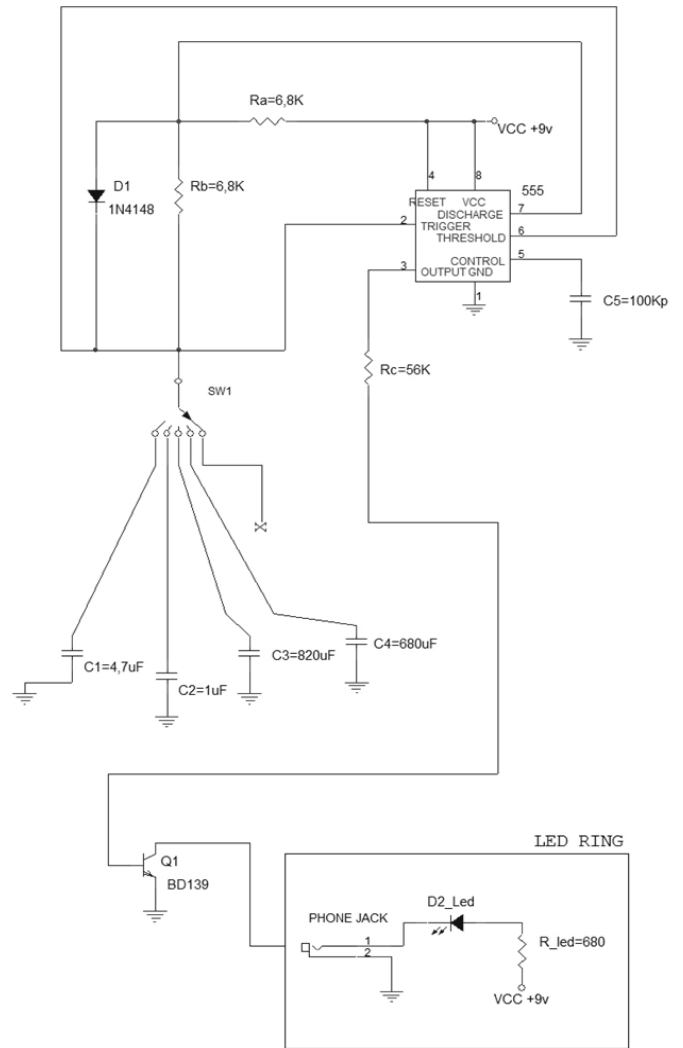


Fig. 2.- Asynchronous chroma ring illumination controller based on an astable-mode 555 circuit connected to the LED ring. A switch allows the selection of the square wave light: C1) T=44 ms (22 Hz); C2) T=9.5ms (106 Hz); C3) T=7.7 ms (130 Hz) and C4) T=6.4 ms (156 Hz) as well as combinations between them.

Several tests were carried out showing that using frequencies beyond 50 Hz no flicker was observed. On the other hand, recording was not recommended with lower frequencies due to considerable discomfort generated by the flickering.

The square illumination pattern generated with the setup of Fig. 2 is sampled by the camera at 25 frames per second as shown in Figure 3. Since the sampling frequency does not satisfy Nyquist theorem (what would require a very slow chroma ring frequency which will be very disturbing to people on the stage), the shape of the light pattern is not conserved by the camera and a certain background intensity pattern is generated. For frequencies close but not equal to the sampling frequency a triangular pattern can be observed as it is schematically depicted in Fig. 3.

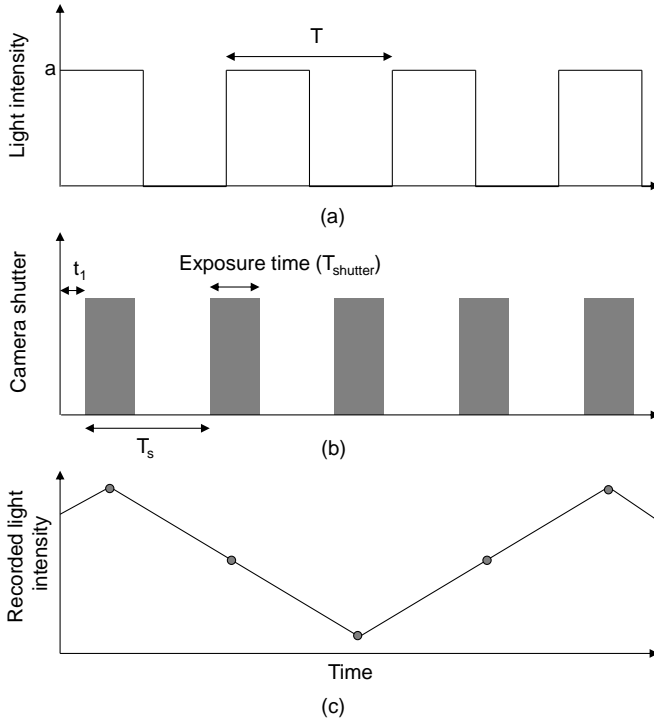


Fig. 3.- Background light subsampling by camera shutter: a) on-off lighting generated by the circuit of Fig. 2; b) sampling by the camera; c) light pattern captured by the camera.

Light intensity at frame n given by the gating and subsampling is,

$$y[n] = \int_{t_1 + T_s}^{t_1 + nT_s + T_{shutter}} \sum_{n=-\infty}^{+\infty} a \prod(t - kT) dt \quad (5)$$

where t_1 is the time offset between the background light and camera sampling rate, T_s is the camera sampling period (inverse of the frame rate), $T_{shutter}$ is the exposure time of the camera (inverse of the shutter speed), a is the light intensity, \prod is the square waveform, and T is the period of the square wave background light.

Assuming that the circuit of Fig. 2 is generating a square wave light with a 50% duty cycle and $T_{shutter} < T/4$, equation (5) can be expressed as,

$$y[n] = \begin{cases} 0, & x < \frac{T}{4} - T_{shutter} \\ a \left(x - \frac{T}{4} + T_{shutter} \right), & \frac{T}{4} - T_{shutter} < x < \frac{T}{4} \\ aT_{shutter}, & \frac{T}{4} < x < \frac{3T}{4} - T_{shutter} \\ -ax + \frac{3T}{4}a, & \frac{3T}{4} - T_{shutter} < x < \frac{3T}{4} \\ 0, & x > \frac{3T}{4} \end{cases}$$

where $x = t_1 + (n-1)T_s \bmod T$.

Figure 4 shows color intensity (for red, green and

blue) for two pixels corresponding to: (solid) the retroreflective screen (background); and (dashed) an object in the scene (foreground) when the frequency of the square illumination is 98 Hz.

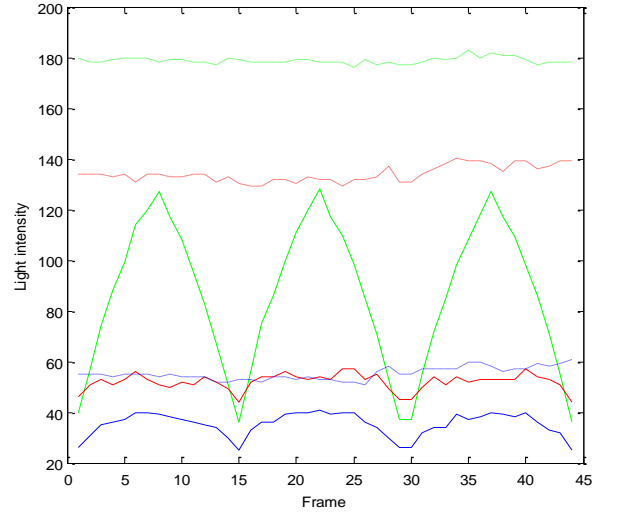


Fig. 4.- Color evolution for two pixels: background made by the retroreflective screen (solid) and an object in the foreground (dashed).

As can be seen from Figure 4 a temporal pattern (triangular signal) is present in green component of pixels of the background due to subsampling of the square illumination by the camera (in this case with a shutter time of 1/210). A small change with time of the blue component can also be seen. Due to this change only the red component is checked in (3)-(4) to ensure that color change is due to the temporal pattern given by the circuit of Fig. 2 and not due to movements in the scene.

Using the conditions given by (2)-(4), a processing code in MATLAB was developed to compute the alpha matte and implement the chroma key. The added computational cost due to the extra processing of the algorithm is around a 14% higher in computing time than a conventional chroma key system. For these experiments a frequency of 58 Hz and a shutter time of 1/120 were chosen.

Figure 5b shows an unprocessed frame captured using the proposed setup as well as a previous (5a) and a posterior (5c) frame used for processing. A cloth with a set of green tones as close as possible to the green tone of the background has been included to test the capability of chroma keying without color restrictions.



(a)



(b)



(c)

Fig. 5.- Original moving image sequence for composition: a) Previous image; b) image to be processed; c) Posterior image.

Figure 6 shows image composition using a conventional chroma key algorithm based on a single image. As it can be seen, foreground extraction fails and regions of the same color of the background are wrongly interpreted as background and they are transparent in the composed image.



Fig. 6.- Image composition based on conventional chroma key algorithm.



Fig. 7.- Alpha matte derived using (2)-(4).

Using the temporal background pattern, and taking advantage of the use of the retroreflective screen, the alpha matte can be derived independently of the foreground colors. The alpha matte can be obtained using (2)-(4) and it is shown in Figure 7.



Fig. 8.- Image composition based on the algorithm shown in (2)-(4).

The extracted foreground image using the alpha matte of Fig. 7 and a background image are combined using (1). The composed image is shown in Figure 8. In this figure is shown that foreground regions whose colors are the same as the background color are successfully extracted.

CONCLUSION

A simple technique to implement a chromakey which avoids restrictions on the colors used in the stage has been proposed. The technique is completely based on simple electronics and components of daily use in the television industry which result in straightforward compatibility with TV and filming studios with very low additional cost (less than \$10 in hardware) plus a reduced increase in computer time. It shows that color restrictions can be avoided without the need of complex electronics or setups.

REFERENCES

- [1] R.C. Kennedy, F.J. Gaskins, "Electronic Composites in Modern Television", *Proceedings of the IRE*, vol. 46, no. 11, pp. 1798-1807, Nov. 1958.
- [2] P. Vlahos, "Comprehensive Electronic Compositing System". US Patent 4,100,569, July 1978.
- [3] S. Shimoda, M. Hayashi, H. Kanatsugu, "New Chroma-key Imaging Technique with Hi-Vision Background", *IEEE Transactions in Broadcasting*, vol. 35, n. 4, p. 357-261, December 1989.
- [4] O. Grau, T. Pullen, G.A. Thomas, "A combined studio production system for 3D capturing of live action and immersive actor feedback", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 3, pp. 370-380, March 2004.
- [5] J. Wang, M.F. Cohen, "Image and Video Matting: A Survey", *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 2, 97-175, 2008.
- [6] S. Gibbs, C. Arapis, C. Breiteneder, V. Lalioti, S. Mostafawy, J. Speier, "Virtual Studios: An Overview", *IEEE Multimedia*, vol. 5, no. 1, pp. 18-35, Jan-Mar 1998.
- [7] Y. Mishima, "Soft edge chroma-key generation based upon hexoctahedral color space," U.S. Patent 5,355,174, 1993.
- [8] H. Agata, A. Yamashita, T. Kaneko, "Chroma Key using a Checker Pattern Background", *IEICE Trans. Inf & Syst*, vol. E90-D, no. 1, January 2007.
- [9] J. Sun, J. Sun, S.B. Kang, Z.B. Xu, X. Tang, H.Y. Shum, "Flash cut: foreground extraction with flash and no-flash image pairs", *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
- [10] T. Graham, R. Richards, "Flash-based keying", GB Patent, 2404108, 2005.
- [11] A. Grundhöfer, O. Bimber, "VirtualStudio2go: digital video composition for real environments", *Computer Graphics (SIGGRAPH08)*, pp. 1-8, 2008.
- [12] A. R. Smith and J. F. Blinn, "Blue screen matting," *Computer Graphics (SIGGRAPH96)*, pp. 259-268, 1996.
- [13] A. Grundhöfer, D. Kurz, S. Thiele, O. Bimber, "Color invariant chroma keying and color spill neutralization for dynamic scenes and cameras", *Visual Computer*, Vol. 26, no. 9, pp. 1167-1176, September 2010.
- [14] R.A. Moses, W.M. Hart, "The temporal responsiveness of vision", *Adler's Physiology of the Eye*, 1987