

Implementación de Interfaz Gráfica para Comparación Visual de Métodos de Segmentación y Procesamiento de Video, Usando Matlab

Xavier Ernesto San Andrés Lascano ⁽¹⁾, Vicente Julián Franco Pombo ⁽²⁾, Patricia Chávez Burbano ⁽³⁾

Facultad de Ingeniería en Electricidad y Computación (FIEC)

Escuela Superior Politécnica del Litoral (ESPOL)

Campus Gustavo Galindo, Km 30.5 vía Perimetral

Apartado 09-01-5863. Guayaquil-Ecuador

x.sanandres@gmail.com ⁽¹⁾, joseju96@hotmail.com ⁽²⁾, pchavez@fiec.espol.edu.ec ⁽³⁾

Resumen

El desarrollo de aplicaciones en el campo del Procesamiento Digital de Señales ha crecido muchísimo en los últimos años gracias al avance tecnológico de hoy. Por nuestras computadoras circulan a diario miles de imágenes y videos digitales que nos llegan de Internet a través de interfaces como el correo electrónico, facebook o el chat. Por esta razón, se considera muy importante que los tres métodos de segmentación de video e imágenes, desarrollados en este documento, sean revisados y examinados con cuidado por el lector, con el fin de poder entender su funcionamiento para luego incorporarlos en algunas de las ramas que usan con mayor frecuencia estas técnicas de procesamiento, como son: la Medicina, laboratorios científicos, empresas de seguridad, arte y entretenimiento.

En este documento se expone los problemas frecuentes que se pueden presentar durante las pruebas de los métodos y las soluciones para cada uno de ellos. Los resultados obtenidos por cada método de segmentación implementado son analizados, y se presentan los resultados de la encuesta de percepción visual de la aplicación realizada a 50 personas.

Palabras Claves: *Procesamiento Digital de Señales, método de segmentación, encuesta de percepción visual.*

Abstract

The development of applications in the field of Digital Signal Processing has grown very much in recent years thanks to technological advances today. On our computers circulate daily thousands of digital images and videos that come from Internet through interfaces such as email, facebook or chat. For this reason, it is very important that the three methods of segmentation of video and images, developed in this document, are reviewed and carefully examined by the reader, to understand their function and then incorporate them into some of the branches that use more often these processing techniques, such as: Medicine, science labs, security companies, arts and entertainment.

This document exposes the common problems that may occur during tests of the methods and the solutions for each one of them. The results for each implemented segmentation method are analyzed, and presents the results of the visual perception survey of the application made to 50 people.

Keywords: *Digital Signal Processing, segmentation method, visual perception survey.*

1. Introducción

El Procesamiento Digital de Imágenes es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información. Entre estas técnicas están el filtrado, la ecualización, la transformación y la segmentación digital.

La segmentación digital de un video o imagen se logra cuando dividimos o separamos éste en las partes u objetos que lo conforman. En este documento se verán tres de los métodos de segmentación más utilizados en la actualidad, su funcionamiento, problemas que pueden surgir segmentando ciertos videos con su respectiva solución, y finalmente el resultado de la encuesta de percepción visual, en la cual las personas pudieron comparar estos métodos y dar su opinión acerca de la utilidad, rendimiento de procesamiento, calidad del video segmentado (con respecto al original) de cada uno de ellos.

La implementación de la interface gráfica se la realizó con el programa Matlab versión R2010a, ya que esta versión maneja mucho mejor las características del video, tanto en la reproducción de los cuadros que comprenden el video como en las funciones para construcciones de datos. En la interface se pueden reproducir los cuadros originales del video y los cuadros segmentados por cada método; además, se puede trabajar con videos previamente grabados o haciendo uso de una cámara *web* conectada al computador.

2. Segmentación de Imágenes y Videos

La diferencia principal es que la imagen es estática, es decir que su forma inicial no cambia, mientras que el video es dinámico, al ser el resultado de una secuencia de imágenes. Esto, al momento de realizar la segmentación tiene su impacto, ya que en una imagen estática las formas, áreas, texturas e intensidad de colores no varían; en cambio en un video se pueden producir cambios de un cuadro a otro, al estar todas las figuras y objetos en movimiento, cambiando su forma y posición. Ver figura 1



Figura 1. Cuadros consecutivos de un video

Se puede además observar en la figura 1 que un cuadro está borroso o desenfocado (efecto denominado *Motion Blur*), debido al movimiento a gran velocidad de los objetos (en este caso de la cámara web), dificultando la detección de figuras y objetos en la imagen.

2.1. Estructura de imágenes RGB, escala de grises y binarias

Las imágenes RGB son un arreglo tridimensional compuesto por tres matrices: rojo, verde y azul. El color de cada pixel es determinado por la combinación de cada uno de los valores de las intensidades en cada plano. Ver figura 2

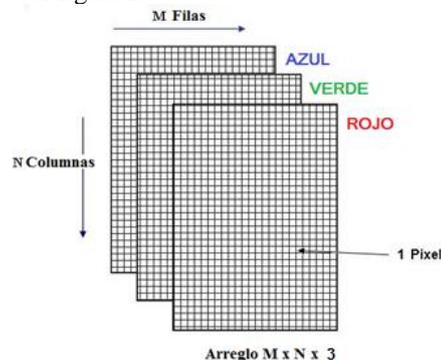


Figura 2. Arreglo RGB, $M \times N \times 3$

Las imágenes en escala de grises son arreglos unidimensionales. La escala de grises tiene un total de 256 intensidades, donde el 0 equivale al color negro y 255 equivale al blanco.

Las imágenes binarias sólo pueden tener dos colores: blanco o negro.

2.2. Métodos de Segmentación

Luego de investigar en Internet (en páginas como Mathworks.com), publicaciones y escritos de diferentes autores y universidades, y posteriormente su estudio individual, los métodos más comunes para realizar segmentación de imágenes en un video digital y que se implementarán en la interfaz gráfica son: segmentación por color, de bordes y por textura.

2.2.1. Segmentación por Color. Para comenzar a segmentar los colores de una imagen, podemos tener los siguientes casos: apagar colores (apagar un color y dejar encendido el resto), o encender colores (encender un color y apagar el resto). En este documento se trabajará con la segunda opción.

Se trabajará con los colores primarios RGB (Red, Green, Blue), ya que por cada color que uno desee ingresar a la segmentación, el tiempo de procesamiento del video se hará cada vez más pesado y la cantidad de cuadros que se generan por cada color se multiplican.

El algoritmo para encender cualquiera de los colores RGB se basa en tratar de detectar el color primario de cada plano, luego hallar el valor máximo entre los 3 valores de la celda, y si éste valor coincide en el plano del color que se desea segmentar, entonces no alteramos su valor y lo mantenemos. De igual manera se procede con el análisis de todas las celdas o pixeles que comprenden la imagen. Ver Figura 3

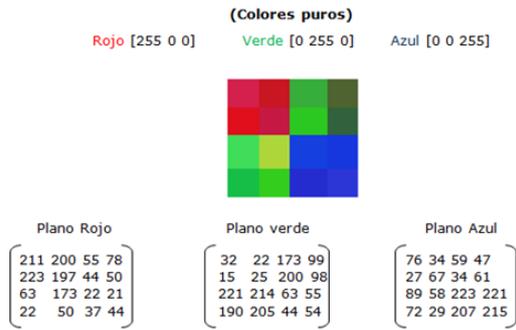


Figura 3. Arreglo RGB de una imagen con sus respectivos planos

Si el máximo valor de una celda pertenece a cualquiera de los otros dos planos diferente al que se está evaluando, entonces igualamos todos los valores de la celda a un valor L, la cual representa al Brillo de Luminancia, con el fin de normalizar el valor de la celda y que esta quede en escala de grises. Con este efecto logramos apagar todos los colores de una imagen RGB, a excepción del color del plano filtrado con la primera parte del algoritmo.

En la figura 4 se observa la segmentación del color azul y como quedan las matrices del arreglo RGB.

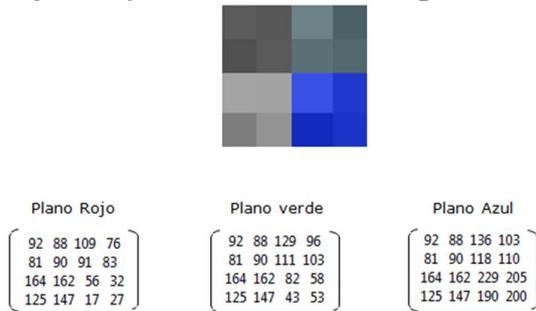


Figura 4. Encendido color azul y apagado de los otros colores

La última fase de la segmentación por color luego de efectuado el realzado o encendido, es la extracción final de la región de color, es decir aislar en cada cuadro del video las regiones que contengan el color deseado. Esto se logra binarizando la imagen que viene como producto del realzado. Ver Figura 5



Figura 5. Color aislado por completo en imagen

2.2.2. Segmentación por Bordes. Esta segmentación se basa en la detección de los bordes que pueden existir en una imagen, con el fin de reconocer objetos o regiones. Por borde se puede definir que son las zonas de una imagen en donde se advierte un notable cambio en los valores de intensidad o niveles en la

escala de grises. Se han desarrollado en Matlab algoritmos capaces de lograr estas detecciones de bordes, entre las más usados están: Sobel, Prewitt y Canny.

Tanto el algoritmo de Sobel como el de Prewitt detectan estas discontinuidades (variaciones o cambios bruscos) en la escala de grises pasando una máscara a través de la imagen; estas máscaras son matrices de convolución de 3x3, las cuales pueden detectar líneas en distintas orientaciones (horizontales, verticales o diagonales) y bordes de la imagen. Estas máscaras varían según el algoritmo que se utilice.

El algoritmo de Canny combina un operador diferencial con un filtro gaussiano, el cual suaviza la imagen (reduce el ruido, detalles y texturas que no interesan); en esta imagen se determinan los píxeles donde se produce máxima variación, se suprimen los píxeles que no son máximos locales y se realiza un proceso de doble umbralización, tanto para determinar los píxeles del borde como para eliminar falsos bordes o bordes dobles.

En la figura 6 se puede observar los resultados de una detección de bordes mediante los métodos de Sobel, Prewitt y Canny, así como la imagen original.

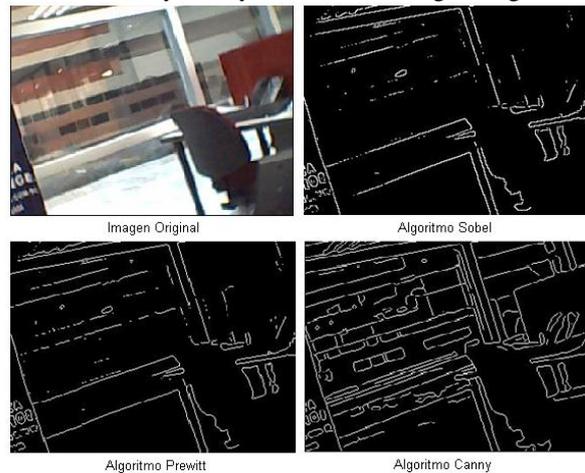


Figura 6. Detección de Bordes usando diferentes algoritmos

Tomando en cuenta que el algoritmo de Canny detectó la mayor cantidad de bordes, que emplea un filtro que reduce considerablemente el ruido de la imagen original y que desecha los falsos bordes, se implementará la segmentación con este algoritmo.

La función *edge* es la encargada de invocar a los diferentes algoritmos desarrollados para la detección de bordes, donde la imagen que debe de cargarse debe estar en escala de grises. Si se quiere cargar una imagen RGB debe ser transformada previamente a escala de grises usando la función *rgb2gray*.

Se puede visualizar los bordes de una región o determinado sector de la imagen haciendo uso de máscaras binarias manuales, definidas directamente sobre la imagen original en el primer cuadro del video que se esté procesando. Ver figura 7



Figura 7. Aplicación manual de Máscara sobre Imagen

2.2.3. Segmentación por Texturas. Segmentación que se refiere a la capacidad de poder distinguir las diversas regiones de una imagen por su contenido de textura (áspero, sedoso, huecos, rugoso, luminoso, etc), determinando las variaciones en los valores de brillo o niveles de gris en la imagen.

El programa Matlab incluye varios filtros para realizar la detección y filtrado de texturas, los cuales son: *STDFILT*, *RANGEFILT* y *ENTROPYFILT*.

- **STDFILT:** Calcula la desviación estándar de una imagen. Este filtro define la vecindad 3x3 alrededor del pixel evaluado y calcula la desviación de dicha vecindad para determinar el valor del pixel en la imagen resultante.
- **RANGEFILT:** Calcula el rango local de una imagen. Este filtro identifica vecindades 3x3 y diferentes formas y tamaños alrededor del pixel evaluado, determinando los valores de los pixeles en la imagen resultante.
- **ENTROPYFILT:** Calcula la entropía local de una imagen. Este filtro define la vecindad 9x9 alrededor del pixel evaluado, calcula la entropía de dicha vecindad y asigna ese valor al pixel de la imagen resultante.

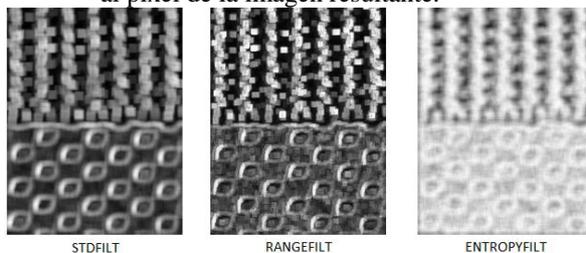


Figura 8. Imagen filtrada por texturas

En la figura 8 se puede observar que el filtro *STDFILT* pudo identificar con mayor precisión los cambios de texturas en las vecindades de cada pixel de la imagen original, dando como resultado una imagen filtrada más suavizada (menor cantidad de ruido).

La aplicación del filtro debe realizarse en una imagen en escala de grises, pudiendo realizar dos acciones que son básicas para poder segmentar por texturas: lograr marcar los límites de las diversas regiones de texturas de la imagen, y poder segmentar una región determinada a través de una máscara lógica. Ver figura 9

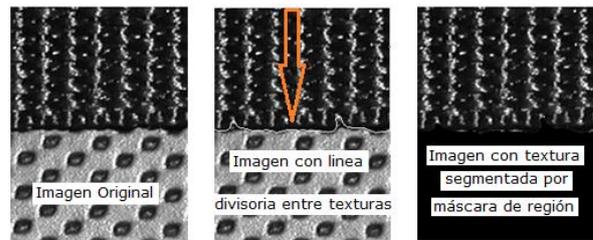


Figura 9. Aplicación de filtro en Imagen

3. Implementación de Interfaz Gráfica

La interfaz consolida los tres métodos de segmentación de video digital en una sola vista. Consta de 4 pantallas de salida donde podemos reproducir el video original y ver el resultado de cada una de las segmentaciones realizadas.

Se puede hacer uso de cámaras *web*, en donde la interfaz puede detectar todas las cámaras conectadas al computador y permitir seleccionar una en particular. También se puede trabajar con videos previamente grabados, y el formato de video con el que se podrá trabajar será AVI. Ver figura 10



Figura 10. Módulos de Control de Cámara Web y Video

Posee dos cuadros de texto, en los cuales se puede saber de cuantos cuadros está constituido el video que se quiere segmentar y el número de cuadros extraídos del mismo. Los cuadros extraídos del video original estarán en formato JPG y serán guardados de forma secuencial en la carpeta *C:\Original_Frames*. De igual manera, los cuadros segmentados por cualquiera de los métodos serán almacenados con el mismo formato y en una carpeta independiente.

En la pantalla visual asignada para la segmentación de color, se pueden mostrar tanto los cuadros encendidos (o realizados) como los cuadros extraídos complementemente el color que se desee segmentar (rojo, verde o azul). En la pantalla de la segmentación de bordes, se pueden mostrar tanto la detección de bordes de los cuadros que componen el video como la segmentación de los bordes en un área específica de dichos cuadros, en donde esta área puede ser ingresada manualmente. En la pantalla de la segmentación de texturas, se puede ingresar manualmente la magnitud de las áreas que se quieren segmentar en los cuadros y la intensidad en escala de

grises que pueden tener dichas áreas. Los cuadros segmentados de cada método pueden ser grabados a video en formato AVI. Ver figura 11



Figura 11. Pantalla Segmentación por Textura

La interfaz consta además de tres ecualizadores, uno para cada método de segmentación, con los cuales podemos mejorar y ajustar la calidad de los videos originales, de tal manera que los algoritmos de segmentación sean más efectivos en sus resultados.

4. Operación y Pruebas

4.1. Operación y Pruebas del Método de Segmentación por Color

Para realizar la segmentación por color, se debe tener previamente extraídos los cuadros del video que se desea segmentar en la carpeta *C:\Original_Frames*. Se presiona *Iniciar* para comenzar con el proceso del encendido de los colores primarios RGB de los cuadros originales, y al presionar *Mostrar* podemos ver los cuadros encendidos del color de interés. Ver figura 12



Figura 12. Resultado encendido color rojo

Luego del encendido de colores se procede a la segmentación total del color deseado. El botón *Cálculo Región* sirve para obtener regiones de interés

del video (en este caso los colores), y el botón *Video Región* permite observar la secuencia de cuadros con el color elegido totalmente extraído de cada uno de ellos. Ver figura 13

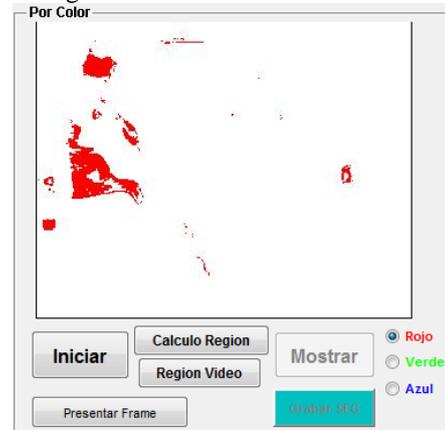


Figura 13. Segmentación completa color rojo

4.2. Operación y Pruebas del Método de Segmentación por Bordes

Al igual que la segmentación por color, esta necesita que los cuadros del video original estén ubicados en la carpeta *C:\Original_Frames*. Se presiona *Iniciar* para comenzar con el proceso de detección de bordes de los cuadros del video original, y luego *Mostrar* para ver dichos cuadros.

Para segmentar una determinada región en el resultado de los cuadros con detección de borde, se debe presionar *Máscara*, con el cual nos permite ingresar de manera manual una máscara de región en el primer cuadro original del video, la cual servirá como patrón de segmentación para todos los demás cuadros del video. Ver figura 14



Figura 14. Proceso Segmentación Bordes

4.3. Operación y Pruebas del Método de Segmentación por Texturas

Al igual que los otros métodos, esta segmentación necesita que los cuadros del video original estén ubicados en la carpeta *C:\Original_Frames*. Cuenta con dos campos de entrada para realizar ajustes sobre el resultado obtenido: el *Área Objeto*, el cual identifica las regiones con determinada cantidad de píxeles, y la *Intensidad*, que denota el nivel en escala de gris que tendrán dichas áreas segmentadas en cada cuadro del video. Luego de ser introducidos los valores de estos dos campos, se presiona *Textura* para

comenzar con el proceso de segmentación. Al término del mismo se mostrará el resultado de la segmentación de los cuadros, con lo que se podrá ajustar los valores de los campos si así se lo requiriere. Ver figura 15

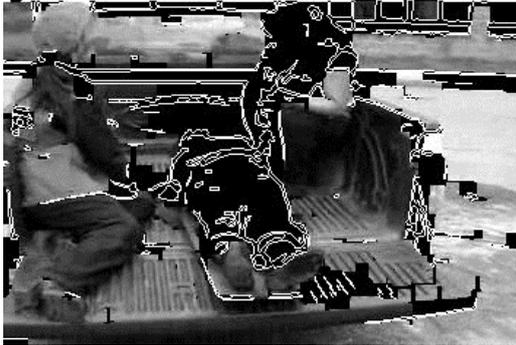


Figura 15. Segmentación por Texturas, con Área Objeto=10 e Intensidad=0.25

5. Presentación y Análisis de Resultados

5.1. Análisis Visual Segmentación por Color

En ciertos casos en los cuales las tonalidades sean parecidas (color turquesa, anaranjado, concho e vino, fucsia, etc) pueden surgir problemas, ya que al ser estas tonalidades una combinación de colores, puede que el algoritmo no los segmente. Así mismo, cuando los colores están concentrados en áreas muy reducidas, éstos pueden no ser identificados por el algoritmo. Para evitar este tipo de problemas se puede hacer uso del Ecuilizador de Color, cuya función es realzar el contraste de cada uno de los planos que conforman la imagen RGB, para después concatenar dichos planos. Ver figura 16



Figura 16. Imagen Original y ecualizada por color

El resultado del encendido de color con la imagen ecualizada fue más eficiente debido a que el ecualizador hace una mejor redistribución de los 256 niveles de grises por toda la escala y evita que estos se concentren en un solo rango que ocasiona que la imagen aparezca muy oscura o brillante.

5.2. Análisis Visual Segmentación por Bordes

En algunos casos la detección de bordes, dependiendo de la calidad del video, presenta mucho ruido y un exceso de bordes en la imagen resultante, lo que dificulta mucho la identificación o apreciación de la imagen central. Para evitar esto, se puede hacer uso del Ecuilizador Binario, el cual binariza las imágenes, para luego proceder con la detección de bordes. El resultado es una imagen con menos detalle pero con el suficiente como para identificar formas y figuras. Ver figura 17

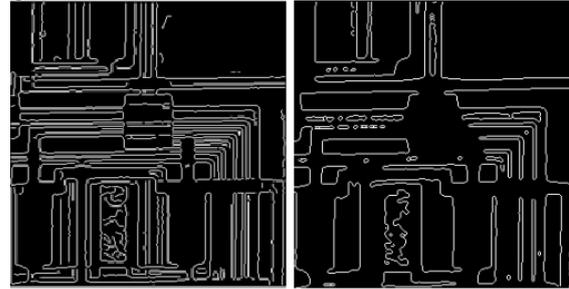


Imagen de un Circuito detectados los Bordes, y haciendo uso del Ecuilizador

Figura 17. Imagen usando detección de Bordes y Ecuilizador Binario

5.3. Análisis Visual Segmentación por Textura

Cuando se está realizando la segmentación por textura sobre un video, hay ocasiones en que las divisiones entre texturas no están bien marcadas o completas. Haciendo ajustes a los campos de entrada (*Intensidad* y *Área Objeto*) podemos mejorar parcialmente estas demarcaciones. Cuando esto suceda podemos hacer uso del Ecuilizador de Contraste, el cual optimiza la calidad de la imagen ya que distribuye mucho mejor las tonalidades en escala de grises, pudiendo así diferenciar las distintas texturas en cada cuadro, inclusive en áreas pequeñas. Ver figura 18



Figura 18. Imagen Original y ecualizada, segmentada por Texturas

5.4. Análisis de Rendimiento de los Métodos de Segmentación

Se realizó una prueba para medir el tiempo que tomaba efectuar el procesamiento de cada cuadro del video durante cada método de segmentación. La

medición se la tomó para 10, 50, 100 y 200 cuadros. Después realizamos las mediciones en otros equipos con distinto procesador y memoria, con el fin de comparar resultados. Los equipos utilizados se pueden observar en la figura 19.

MODELO	HP 8440p	HP 4420s	HP 530
CARACTERÍSTICA			
MEMORIA RAM(GB)	4	3	2
PROCESADOR	Core i5 - 2.4 Ghz	Core i3 - 2.4 Ghz	Core2duo-1.67 Ghz
DISCO DURO(GB)	500	250	100

Figura 19. Equipos utilizados para análisis de rendimiento

En el primer equipo, tanto el método de bordes como el de textura trabajaron a una velocidad aproximada de 10 cuadros/segundo, mientras que la de color 1 cuadro/segundo. Ver figura 20

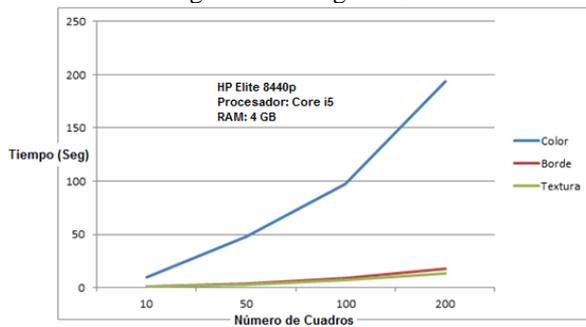


Figura 20. Velocidad de procesamiento en HP 8440p

En el segundo equipo, el método de bordes y textura trabajaron a 10 y 12 cuadros/segundo respectivamente, y el de color a 0.8 cuadros/segundo. En el tercer equipo se registró una velocidad para el método de bordes y textura de 4 y 5 cuadros/segundo respectivamente, y de color 0.3 cuadros/segundo.

Con estas pruebas se puede determinar que mientras mejor procesador y mayor memoria tenga el computador que se esté utilizando, más rápida será la velocidad de procesamiento, lo cual es de suma utilidad tener conocimiento principalmente si se quiere trabajar con videos mucho más grandes (videos compuestos por más de 500 cuadros).

5.5. Resultados Encuestas de Percepción Visual

La encuesta se realizó a 50 personas de diferente género y edad, donde pudieron escoger el método de segmentación preferido, de mayor utilidad, de mejores resultados, etc.

Los encuestados escogieron a la segmentación de color como el método más útil entre los tres (ya sea esta utilidad para su trabajo, entretenimiento, ayuda para la comunidad científica, etc.). Ver figura 21

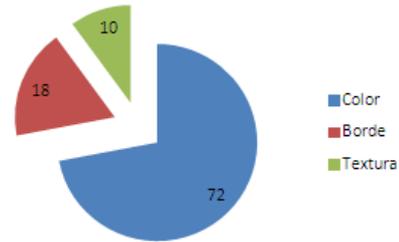


Figura 21. Utilidad de los Métodos de Segmentación

La segmentación por color fue el método de mayor acogida para fines científicos, profesionales y de entretenimiento; la de bordes tuvo acogida para fines comerciales y la de textura para fines profesionales, después de la de color.

Se preguntó a los encuestados que mejoraría de la interfaz, a lo cual respondieron agregar más métodos de segmentación, añadir más colores a la segmentación de color, rediseñar la interface (hacer más grandes las salidas visuales, menos botones), y otros dijeron que no necesitaba cambios. Ver figura 22

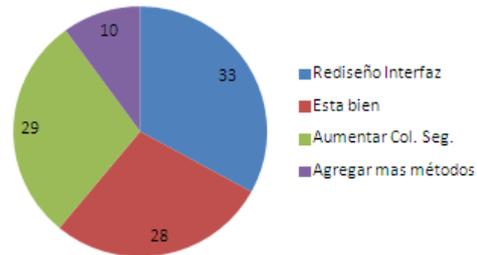


Figura 22. Recomendaciones de encuestados

Entre los ecualizadores de cada método, el de color fue el que tuvo mejores resultados con los encuestados, ya que los cambios entre los cuadros del video original y el ecualizado eran más notorios a simple vista y de mayor interés.

La gran mayoría de personas concluyó que los métodos de segmentación no demoran en segmentar los videos cuadro por cuadro. Asimismo, la gran mayoría opinó que la calidad del video segmentado (nitidez, pixeleo, intensidad, etc.) es igual al de los videos de entrada. Ver figura 23

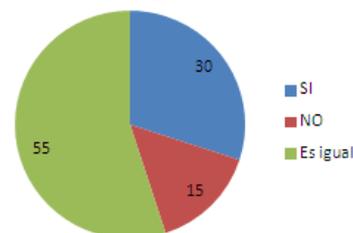


Figura 23. Calidad video de salida vs. entrada

También se preguntó al grupo que consideraban más útil, la segmentación por imágenes independientes o la segmentación de imágenes (cuadros) en secuencia

para modificar todo un video, a lo que un 60% opinó que preferían la segmentación del video completo, mientras que el 40% restante piensa que mejor es segmentar imagen por imagen.

6. Conclusiones y Recomendaciones

6.1. Conclusiones

El método de segmentación que mejor aplica para trabajar con video digital es el de segmentación por color, ya que el resultado que este método entrega es de mucho valor e impacto visual al permitir el apagado y encendido de colores tradicionales como rojo, verde y azul en áreas grandes y pequeñas. Además, este método presenta muchas posibilidades de usarlo en aplicaciones científicas, comerciales y de entretenimiento.

El rendimiento de la herramienta de segmentación fue bastante estable en su funcionamiento, sobre todo cuando los videos no superaban los 600 cuadros. Esto se debe a que en videos largos o muy comprimidos (de miles de cuadros) la segmentación de las imágenes en Matlab involucra mucho procesamiento, agotando la memoria RAM de la computadora. Si se quisiera segmentar videos que tienen miles de cuadros hay que contar con mayor cantidad de memoria disponible en el sistema y contar con un buen procesador.

El uso de los ecualizadores en los métodos de segmentación fue de suma ayuda, especialmente en los videos en los cuales se tuvo problemas segmentando las áreas de interés, ya que estos ecualizadores pueden cambiar las amplitudes, la frecuencia, realzar colores, ajustar la distribución de intensidades de grises de la imagen que está siendo procesada para obtener una mejor visualización de la misma, pudiendo con esto segmentar las zonas que no pudieron ser detectadas en el video original.

La velocidad de segmentación de un video depende de algunos factores, como lo son el método de segmentación utilizado, la memoria RAM y el procesador con que cuenta el computador, la cantidad de cuadros que componen el video, la compresión del video, y cantidad de programas que se estén ejecutando en el computador. El método que mayores tiempos de respuesta tuvo fue el de color, seguido por el de bordes, y el de textura fue el más veloz; esto se debe a que hay mayor cantidad de procesamiento en el encendido y apagado de todos los píxeles de la imagen y luego segmentar las áreas de interés (color deseado).

6.2. Recomendaciones

Para desarrollar interfaces gráficas en Matlab en las cuales se quiera trabajar con videos digitales, hacerlo con la versión R2010a del producto o superior, ya que esta versión maneja mucho mejor las características

del video, tanto en la reproducción como en la toma de cuadros.

Hacer uso del cuadro de texto *Ajuste de Frames* cuando un video previamente grabado presente una diferencia entre los cuadros leídos y extraídos. Esta diferencia de cuadros (que suele ser de 1 o 2 cuadros) se presenta debido tanto a la edición como a la compresión del video que se desea evaluar.

Contar con un buen procesador en el computador, para que el rendimiento y velocidad de procesamiento de los cuadros del video sea mayor. Si se quiere trabajar con videos cortos (hasta 500 cuadros) es suficiente contar con una memoria RAM de 2GB, caso contrario se recomienda aumentar la capacidad a 3 o 4GB para realizar una lectura y extracción completa de los cuadros del video.

Aplicar el método de segmentación por textura a nivel de imagen individual, pudiendo de esta manera tener una mayor apreciación de las diferentes texturas que se encuentran en la imagen. En un video, al ser una sucesión de imágenes, no se podrán distinguir las texturas ya que en cada cuadro habrá diferentes regiones segmentadas.

7. Agradecimientos

A Dios por la confianza que deposita todos los días al brindarnos la fuerza, el valor y la sabiduría necesaria para culminar la carrera universitaria.

A nuestra Directora de Tesis MSc. Patricia Chávez, quién fue nuestra guía en cada momento y confió en nosotros para la realización de este proyecto.

A nuestros padres y hermanos por el apoyo incondicional a lo largo de la vida universitaria y a nuestra querida ESPOL que nos ha brindado todo.

8. Referencias

- [1] Báez, J.J., Segmentación de imágenes de color, <http://www.ejournal.unam.mx/rmf/no506/RMF50605.pdf>
- [2] Banterla, Flavio, Análisis de Texturas en Matlab, <http://www.ehu.es/ccwintco/uploads/d/d7/Texturas.pdf>
- [3] Valverde, Jorge, Detección de Bordes mediante el algoritmo de Canny, <http://www.seccperu.org/files/DetecciondeBordes-Canny.pdf>
- [4] Arfan, Jaffar, color video segmentation, <http://www.wseas.us/e-library/conferences/2009/istanbul/SIP-WAV/SIP-WAV-03.pdf>
- [5] Zaldivar, Daniel, Procesamiento Digital de imágenes, <http://es.scribd.com/doc/61579860/23371-Procesamiento-de-Imagenes-Con-Matlab>