

Estudio de herramientas de ecualización automática del color

Gustavo Brown Rodriguez
gbrown@fing.edu.uy

Curso: Algunos métodos de procesamiento de imágenes con aplicación a la postproducción de fotografía y cine

Abstract—En este artículo estudiaremos dos herramientas de ecualización automática del color y sus aplicaciones a distintos tipos de imágenes naturales. Por un lado, se aplicará el algoritmo ACE (Automatic Color Equalization) que está basado en modelos computacionales del sistema visual humano. Este algoritmo combina las nociones de *Gray World* y *White Patch* mediante un esquema de computación local/global que intenta asemejarse al funcionamiento del sistema visual humano. Por otro lado utilizaremos un mecanismo simple de *stretching* del histograma de color, ACS (*Automatic Color Stretching*), para utilizar todo el rango dinámico. Su principal ventaja sobre ACE es su baja complejidad, lo cual lo hace más idóneos para algunos casos donde ACE es inviable. Compararemos los resultados obtenidos de aplicar ACE y ACS sobre distintos tipos de imágenes (sub expuestas, sobre expuestas, subacuáticas) aplicando a su vez distintos parámetros de cada algoritmo.

Index Terms—ACE, Automatic Color Equalization, Retinex, Histogram stretching

I. INTRODUCCIÓN

UNA de las características del sistema visual humano es la cualidad de restar el iluminante de una escena. Esta característica, llamada constancia de color (*color constancy*), nos permite percibir una misma escena independientemente de los cambios de iluminación [1]. Si bien aún hoy no se entiende completamente el funcionamiento de sistema visual humano, se han desarrollado varios modelos computacionales que tratan de reproducirlo, al menos de forma parcial. La teoría *Retinex* [2] introducida por Land en la década de 1970, presenta una primera aproximación. En ella se asume que el sistema de visión humano procesa una escena en tres sistemas independientes; frecuencias bajas, medias o altas; que a su vez coinciden con los tres canales de color que trabajamos usualmente (rojo, verde y azul). A su vez, el modelo indica que la iluminación relativa de una región de la escena respecto de sus regiones adyacentes juega un papel preponderante en la percepción de la misma.

Recientemente, se ha propuesto un nuevo modelo de aproximación al sistema visual humano, ACE [3] (*Automatic Color Equalization*), que combina varios aspectos como son *White Patch* [2] y *Grey World* [4]. El primero normaliza los 3 canales (frecuencias bajas, medias y altas) maximizando los valores de acuerdo a una referencia de color blanco en la escena. El segundo nos hace percibir los objetos de una escena que reflejan la luminancia media de la escena como grises [5]. La principal desventaja de ACE es su costo computacional, que en su formulación original es de N^2 , con N siendo la cantidad

de pixels de la imagen. No obstante, trabajos posteriores en el área [6], [7] han permitido reducir el costo computacional sin introducir cambios significativos en los resultados.

Como contrapartida de ACE utilizaremos un mecanismo de corrección automática del color muy simple basado en ecualización de histogramas [8], al cual llamaremos ACS (*Automatic Color Stretching*). Este mecanismo toma las estadísticas de una imagen de entrada para cada uno de los tres canales (frecuencias bajas, medias y altas) y modifica su histograma para utilizar todo el rango dinámico de cada canal. Este algoritmo tiene un costo computacional reducido, de orden N y permite, con pequeñas modificaciones, aproximar en sus resultados el concepto de *Grey World*. En este artículo lo utilizaremos como una contrapartida de bajo costo computacional para compararlo con ACE en distintos tipos de imágenes naturales.

La aplicación de estos algoritmos automáticos permiten no solo mejorar la percepción visual de imágenes subexpuestas/sobrexpuestas, sino que incluso permiten reducir el fenómeno de *color cast* [9] presente en algunos tipos de imágenes como por ejemplo fotos subacuáticas o antiguas fotografías.

Este artículo se organiza de la siguiente manera. En la sección II se describe el algoritmo ACE y se comentan detalles de su implementación. En la sección III se describe el algoritmo ACS que utilizaremos para comparar resultados con ACE. La sección IV compara la aplicación de estos algoritmos a distintas imágenes y finalmente la sección V concluye brinda algunas perspectivas de la comparación.

II. ACE

El algoritmo ACE [3], [6], [10], *Automatic Color Equalization*, fue presentado por Rizzi et al. en 2003. Basándose en los principios postulados por Land [2] brindan un modelo computacional que aproxima la percepción visual del sistema visual humano. El algoritmo, se basa fuertemente en los principios *White Patch* y *Grey World* descritos en la introducción y permite mejorar el aspecto visual de una imagen natural sin intervención del usuario.

El algoritmo se divide en dos etapas. Una primer etapa, llamada *Ajuste cromático/espacial*, trabaja en los canales R, G y B (frecuencias baja, media y alta) independientemente. Toma como entrada la imagen del canal asociado y aplica un algoritmo local/global a cada pixel para producir una imagen intermedia R_c . En esta etapa, para cada pixel de la imagen se computa el pixel asociado en la imagen R_c de acuerdo a la siguiente ecuación:



Figure 1. Ejemplo de imagen procesada con ACE

$$R_c(p) = \frac{\sum_{j \in \text{imagen}, j \neq p} \frac{r(I_c(p) - I_c(j))}{d(p,j)}}{\sum_{j \in \text{imagen}, j \neq p} \frac{r_{max}}{d(p,j)}}$$

donde p indica cada pixel de la imagen, $I_c(i)$ indica el pixel i de la imagen de entrada, $d(\cdot)$ es una función de distancia que maneja el peso en la contribución local vs. global de los pixels de la imagen. Por su parte, la función $r(\cdot)$ actúa como una función de ajuste del contraste. El divisor en la ecuación actúa como un factor de normalización especialmente útil en los bordes de la imagen. A efectos del estudio y comparación de este artículo, se utilizaron las siguientes funciones de distancia y de ajuste de contraste, donde $r(\cdot)$ ha sido normalizada al rango $[0, 1]$, y por tanto r_{max} vale 1:

$$d(\cdot) = \begin{cases} euclidea & \sqrt{dx^2 + dy^2} \\ Manhattan & |dx| + |dy| \\ máximo & \max(dx, dy) \end{cases}$$

$$r(l) = \begin{cases} lineal & \frac{l}{255} \\ slope(S) & \begin{cases} -1 & \frac{l}{255} < -1 \\ \frac{l*S}{255} & -1 \leq \frac{l}{255} \leq 1 \\ 1 & \frac{l}{255} > 1 \end{cases} \\ signum & \frac{l}{|l|} \end{cases}$$

La segunda etapa del algoritmo, llamada *Mapeo del rango dinámico*, aplica el principio corrección global *Grey World* y

White Patch utilizando los valores máximo y medio de la imagen intermedia R_c como estimaciones del color blanco y gris respectivamente. Luego, se aplica un escalado lineal de la imagen para utilizar todo el rango dinámico (256 valores para cada canal) siguiendo la siguiente ecuación:

$$R_o(p) = \text{round}(127,5 + sc \cdot R_c(p))$$

donde sc es la pendiente del segmento $[(m_c, 0), (M_c, 255)]$, con m_c y M_c el valor mínimo y máximo de R_c .

La figura 1 muestra el resultado de aplicar ACE sobre una imagen subexpuesta y la salida del algoritmo utilizando distintas medidas de distancia. En las imágenes procesadas se puede ver mayor detalle en las zonas donde previamente estaba mucho más oscuro, y no se nota diferencia entre los distintos tipos de medidas de distancia aplicadas. En la figura 2 se muestra el resultado de aplicar ACE utilizando distintas funciones de ajuste de contraste. Como puede observarse, la aplicación de una función lineal deja la imagen poco nítida. Por otro lado, el otro extremo (*signum*) brinda el mayor contraste pero la imagen no se ve natural. En la práctica los mejores resultados se han encontrado utilizando una función con pendiente de valor 5.

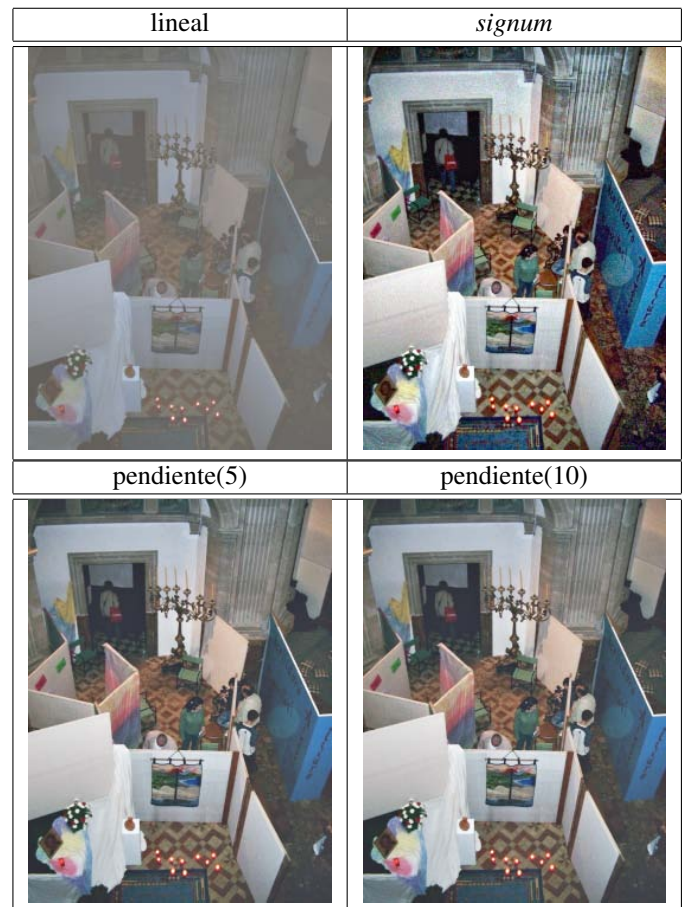


Figure 2. Ejemplo de imagen procesada con ACE

Desde el punto de vista computacional, puede verse que el cálculo de R_c implica que por cada pixel de la imagen de entrada I_c se compute la proporción de peso sobre dicho pixel de acuerdo a todos los demás pixels de la imagen. Esto quiere decir que esta etapa tiene un orden $O(N^2)$ con N

siendo el tamaño total de la imagen. La segunda etapa del algoritmo implica recorrer la imagen intermedia para realizar el escalado, por lo que esta etapa tiene un orden $O(N)$. Para la implementación de este estudio se permitió definir un parámetro del algoritmo que indica si se desea utilizar toda la imagen para el cálculo de la contribución del peso de cada pixel de la imagen en el cálculo de R_c o si, por el contrario, se desea utilizar un subconjunto de la imagen (centrado en el píxel a procesar).

Los autores del algoritmo indican que los resultados del mismo son muy prometedores. El algoritmo permite obtener una corrección en la constancia de color, imágenes que utilizan un mayor rango dinámico, y que además eliminan el color cast presente en algunas imágenes como puede verse por ejemplo en las figuras 7 y 8. En la sección IV se verán los resultados obtenidos con la implementación de este algoritmo utilizando los parámetros aquí mencionados.

III. ACS

El algoritmo ACS, *Automatic Color Stretching*, es un algoritmo de baja complejidad que toma como entrada una imagen y genera como salida una imagen modificada para utilizar todo el rango dinámico en cada uno de los canales R, G, B (frecuencias bajas, medias y altas). El algoritmo funciona en dos etapas. En la primera etapa, mostrada en el algoritmo 1, se computa el histograma para cada canal. Un parámetro p indica la porción de las estadísticas de los bordes (histograma acumulado) a ignorar. Es decir que se ignora una pequeña porción del histograma (los valores más pequeños y los más grandes) dado que se asume que ellos suelen aparecer por causa de errores en el sistema de captura de las cámaras fotográficas.

Algorithm 1 Cálculo del histograma y extremos

Entrada: Imagen I , proporción p de descarte (%/1000)
Salida: Histograma H , extremos m_i , M_i y media M_m

```
for(i = 0; i < N; i++) H[I[i]]++;
for(cum=0, mi=0; cum < N/p; cum+=H[mi]) mi++;
for(cum=0, Mi=255; cum < N/p; cum+=H[Mi]) Mi--;
for(cum=0, Mm=0; cum < N/2; cum+=H[Mm]) Mm++;
```

En una segunda etapa se toma el rango restante (mínimo, máximo) y se aplica un escalado lineal para utilizar todo el rango dinámico del canal. El algoritmo permite la aplicación de algunas variantes. Por un lado se puede utilizar un pequeño kernel para tener una mayor resolución en cada canal (en vez de 256 niveles, se pasa a tener por ejemplo 512 niveles). Una segunda variante es utilizar un escalado lineal pero a intervalos $[(m_i, 0), (M_m, 127, 5)]$ Y $[(M_m, 127, 5), (M_i, 255)]$, donde m_i es el valor mínimo encontrado en la imagen, M_m es el valor donde el histograma acumulado suma la cantidad de pixels equivalente a la mitad del total de la imagen, y M_i es el valor máximo encontrado en la imagen. De esta manera se utiliza el total del rango dinámico del canal pero escalando desde el nivel negro al *gris* (medio) de la imagen y aplicando

otro escalado desde el *gris* (medio) al blanco, como puede observarse en la siguiente ecuación:

$$R_o(p) = \begin{cases} \frac{p-m_i}{2 \cdot (M_m-m_i)} & p \leq M_m \\ 1 - \frac{M_i-p}{2 \cdot (M_i-M_m)} & p > M_m \end{cases}$$

La figura 3 muestra el resultado de procesar una imagen con este algoritmo, variando los parámetros de procesamiento. Las figuras 7 y 8 muestran la habilidad de remover el *color cast* de las escenas subacuática.

Desde el punto de vista computacional, se puede ver fácilmente que las dos etapas del algoritmo deben recorrer una vez la imagen, por lo que cada etapa tiene un orden $O(N)$, donde N es la cantidad total de pixels que componen la imagen.



Figure 3. Ejemplo de imagen procesada con ACS

IV. COMPARACIÓN Y RESULTADOS

A los efectos de este artículo se buscó un conjunto de imágenes naturales de diversas características, tomadas con una cámara fotográfica digital de uso general. Entre otras, se eligieron fotos tomadas al aire libre de día, fotos con zonas sobreexpuestas (donde aparece el sol), y fotos donde una región aparece bajo sombra (mitad iluminada por el sol y mitad bajo sombra de un techo). Luego se eligieron algunas fotos iluminadas con luz artificial. Estas suelen mostrar un *color cast* típico de la fuente de luz artificial que incide en la escena. Por último se testearon los algoritmos con fotos

tomadas bajo el mar, las cuales presentan un claro tono verde-amarillento propio de las condiciones en las cuales fueron tomadas (mediana transparencia del agua de mar) y el hecho de que éstas también fueron tomadas con una cámara acuática digital de uso general. Los tamaños de imagen utilizados varían entre pequeñas (~480x360) a medianas (~1024x768).

Para cada imagen de entrada se aplicó el algoritmo ACE con las tres funciones de distancia presentadas en la sección II (euclídea, Manhattan y máxima), y con tres tipos de funciones de ajuste de contraste r (*signum*, lineal y con pendiente de valor 5).

Para el caso del algoritmo ACS se utilizaron 3 valores para el porcentaje de descarte del histograma (0%, 1% y 3%) y se procesó la imagen con y sin la utilización del kernel. A su vez, las imágenes fueron procesadas con y sin la aplicación del escalado a tramos.

En las figuras presentadas en el artículo en general se mostrará las imágenes procesadas con estos parámetros que han dado los mejores resultados para cada uno de los dos algoritmos.

A. Fotos con luz natural

El procesamiento aplicado a fotos tomadas al aire libre (figuras 4 y 9) genera, para el caso de ACE, imágenes donde se pueden ver detalles escondidos de la foto original, si bien en algunos casos la imagen queda un poco “lavada”. Utilizando la función de ajuste de contraste *signum* se puede obtener una imagen de alto contraste. Con ACS las fotos ganan contraste pero algunos colores se ven diferentes (por ejemplo el cielo).

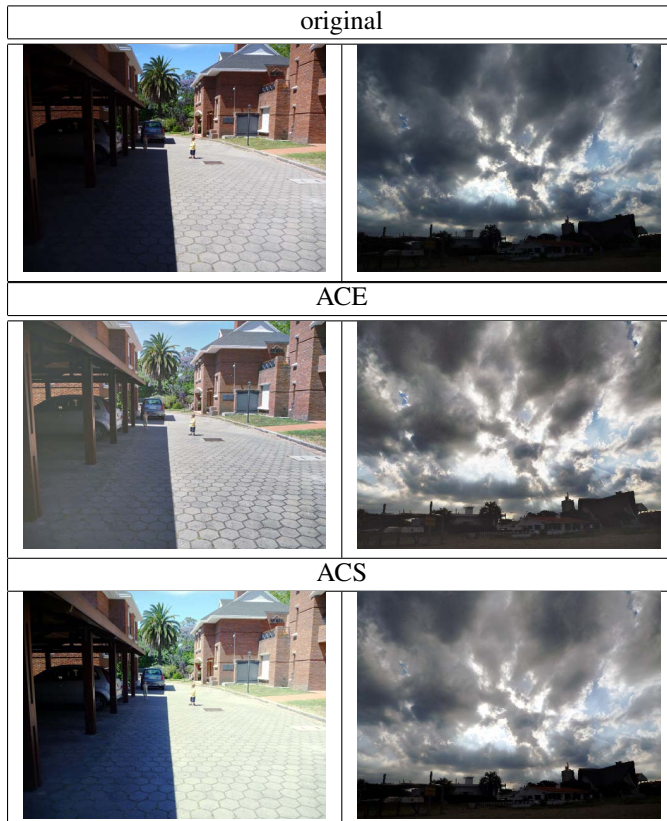


Figure 4. Fotos con luz natural

B. Fotos con luz artificial

Las fotos tomadas en interiores suelen tener poca luz, especialmente en rincones de la escena. A su vez presentan cierto *color cast*, cuya tonalidad depende de la fuente artificial de luz que incide en la escena. En las figuras 1, 3 y 5 se puede ver el resultado de aplicar los algoritmos a este tipo de imágenes. La figura 10 muestra la reducción de la coloración de la escena por la fuente de luz.



Figure 5. Fotos con luz artificial

C. Fotos subacuáticas

Las subacuáticas, especialmente las que fueron tomadas con una cámara fotográfica no profesional presentan un tono verde-amarillento propio de las condiciones en las cuales fueron tomadas, y en general aparecen un poco difusas. La aplicación de los algoritmos de equalización logran remover en gran medida el tono de las mismas y, a su vez, generan como resultado imágenes mucho más nítidas. La figura 11 muestran ejemplos aplicados con ambos algoritmos.

D. Fotos donde los algoritmos fallan

No todas las fotos producen los resultados esperados. En particular, las fotos con alto rango dinámico donde, por ejemplo, una zona aparece fuertemente iluminada por el sol y otras zonas están bajo sombras suelen no siempre generar buenos resultados, tanto con ACE como con ACS. Por ejemplo, la foto de la figura 6 puede verse como la imagen procesada queda “lavada” en el caso de ACE y con efectos de blocking en el caso de ACS por etapas o directamente no se ve diferencia en ACS sin etapas.

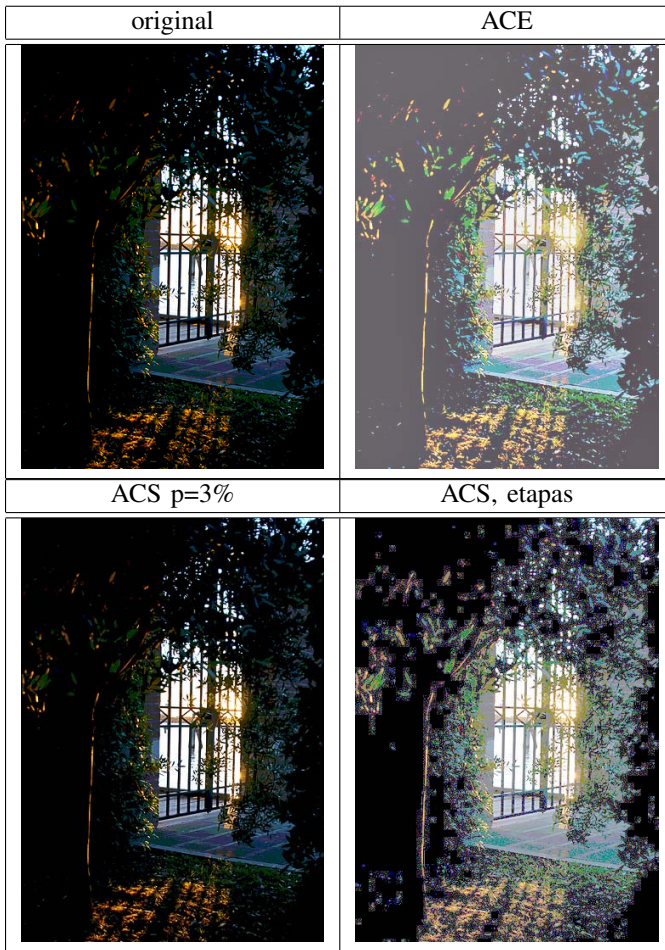


Figure 6. Ejemplo de imagen fallido

E. Análisis de complejidad

Como se discutió en la sección II, el algoritmo ACE tiene, en su formulación original, una complejidad temporal de orden $O(N^2)$ con N siendo el tamaño total de la imagen. Dicha complejidad se hace notar en las imágenes de tamaño regular (por ejemplo mayores a 640x480).

A efectos de este estudio se implementó un parámetro opcional al algoritmo que indica el tamaño máximo del subconjunto de pixels con los cuales trabajar para cada pixel, con lo cual su complejidad pasa a ser de orden $O(K.N)$, con K una constante que a efectos prácticos aún sigue haciendo que el algoritmo sea lento. En los casos de prueba se tomo para las imágenes mayores a 640x480 un conjunto máximo de pixels de 500x500. Como puede verse en la figura 12, aún trabajando con subconjuntos de este tamaño se puede notar la degradación en la calidad de las imágenes producidas. Otro tipo de optimizaciones a ACE [6], [7] reducen la complejidad notablemente pero no fueron utilizadas para este estudio.

El algoritmo ACS tiene en cambio una complejidad temporal de orden $O(N)$, lo cual permite aplicar la equalización en imágenes de gran tamaño en un tiempo reducido.

La tabla 1 muestra una comparación del tiempo de ejecución para el procesamiento de algunas imágenes de prueba utilizando los dos algoritmos en una computadora Core 2 duo de 3GHz. Como los dos algoritmos trabajan en los tres canales R,

G, B de forma independiente se podría paralelizar el cómputo para cada canal por separado.

Por su parte, la complejidad de espacio de ambos algoritmos es de $O(N)$: se necesita tener en memoria la imagen de entrada y de salida (de hecho, no es necesario mantener en memoria la imagen de salida completa). Para el caso del algoritmo ACS se debe mantener también el histograma de la imagen de entrada, que puede implementarse básicamente con un arreglo de tamaño predefinido.

Table I
COMPARACIÓN DEL TIEMPO DE EJECUCIÓN DE ACUERDO AL ALGORITMO Y AL TAMAÑO DE IMAGEN

Tamaño imagen	ACE	ACE 100x100	ACE 500x500	ACS
256x341	13m	8m	13m	1s
480x360	50m	10m	35m	1s
1024x768	17h15m	15m	4h7m	2s

V. CONCLUSIONES

La utilización de algoritmos de equalización automática de color permiten mejorar la calidad de imagen de un conjunto interesante de imágenes naturales tomadas en condiciones desfavorables. Las imágenes resultantes permiten distinguir con mayor detalle fotografías que previamente se veían oscuras. A su vez, permiten remover con éxito el color cast producto de la iluminación incidente en la escena, incluso para el caso de escenas subacuáticas.

Para el caso de ACE, se definen dos tipos de parámetros: una medida de distancia y una función de ajuste de contraste. En la práctica no se observa diferencia de acuerdo a la medida de distancia tomada de entre las relevadas (euclídea, *Manhattan*, máxima). En cambio, la función de ajuste de contraste sí tiene efecto visual observable. Las función de ajuste lineal produce, generalmente, imágenes con bajo contraste “lavadas”. En el otro extremo, si se utiliza la función *signum* se obtienen imágenes con el mayor contraste donde se distingue el mayor detalle pero visualmente irreales. Según las pruebas realizadas, los mejores resultados se obtuvieron al utilizar una función de contraste con pendiente de valor 5.

Para el algoritmo ACS, se definen tres parámetros: el porcentaje del histograma cumulativo a ignorar a efectos del cómputo de los extremos, el uso de un pequeño kernel para aumentar el rango dinámico del canal y la posibilidad de utilizar un escalado por etapas. La utilización del parámetro que define el porcentaje de descarte tiene alto impacto en el resultado del algoritmo, especialmente en casos donde la imagen de entrada ya cuenta con un rango dinámico grande (de hecho, si el canal utiliza todo el rango dinámico en ese caso por defecto no habría *stretching*). La utilización del pequeño kernel en general da como resultado imágenes un poco difusas. Se pudo observar que la utilización del escalado en dos etapas es, en general, la que genera mejores resultados. Por un lado posibilita la eliminación del *color cast* que ocurre cuando uno de los canales tiene su histograma desplazado respecto de los otros dos canales. Por el otro, aun en casos donde no se presenta este tipo de desviaciones igualmente se pudo observar mejores resultados aplicando el escalado en dos etapas, junto con un porcentaje de descarte del 1%.

Analizando comparativamente ACE con ACS, en la mayoría de los casos se observa un mejor resultado visual con ACE. Ambos algoritmos fallan con algunos tipos de imágenes, especialmente cuando una zona de la escena está muy iluminada pero a su vez cuenta con partes donde la fuente de iluminación esta ocluida. El costo computacional de ACS es mucho más simple que el de ACE lo cual permite trabajar fácilmente con imágenes de gran tamaño. ACE, en cambio, puede ser aplicado sin problemas en imágenes de pequeño a mediano tamaño. Su complejidad temporal puede volverlo prohibitivo de utilizar para imágenes de tamaño mediano a grande. Para estos casos se podría utilizar algunas de las optimizaciones vistas en la sección anterior.

REFERENCES

- [1] E. H. Land, J. J. McCann, *Lightness and retinex theory*, Journal of the Optical Society of America, vol 61, pp. 1-11, 1971.
- [2] E. H. Land, *The retinex theory of color vision*, Scientific American, vol 237, nro 6, pp. 108-128, 1977.
- [3] A. Rizzi, C. Gatta, D. Marini, *A New Algorithm for Unsupervised Global and Local Color Correction*, Pattern Recognition Letters, Vol 24 (11), pp. 1663-1677, Jul 2003.
- [4] G. Buchsbaum, *A spatial processor model for object colour perception*, Journal of The Franklin Institute-engineering and Applied Mathematics, vol 310, pp. 1-26, 1980.
- [5] A. Rizzi, C. Gatta, D. Marini, *Color correction between gray world and white patch*, Proc. of the Electronic Imaging, 2002.
- [6] A. Artusi, C. Gatta, D. Marini, W. Purgathofer, A. Rizzi, *Speed-up Technique for a Local Automatic Colour Equalization Model*, Computer Graphics Forum, vol 25(1), pp. 5-14, 2006.
- [7] M. Bertalmío, V. Caselles, E. Provenzi, A. Rizzi, *Perceptual Color Correction Through Variational Techniques*, IEEE Transactions on Image Processing, vol 16, 2007.
- [8] R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, Prentice Hall, 2nd ed, isbn 0-20-118075-8, 2002.
- [9] K. Barnard, V. Cardei, B. Funt, *A Comparison of Computational Color Constancy Algorithms*, IEEE trans. On Image Processing, Vol 11(9), 2002.
- [10] A. Rizzi, C. Gatta, D. Marini, *From Retinex to Automatic Color Equalization: issues in developing a new algorithm for unsupervised color equalization*, Journal of Electronic Imaging 13(1), pp. 75-84, 2004.

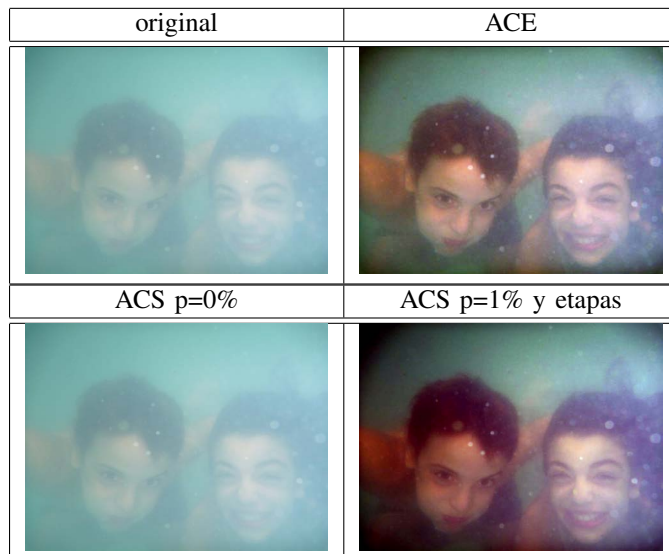


Figure 7. Mejora en detalle

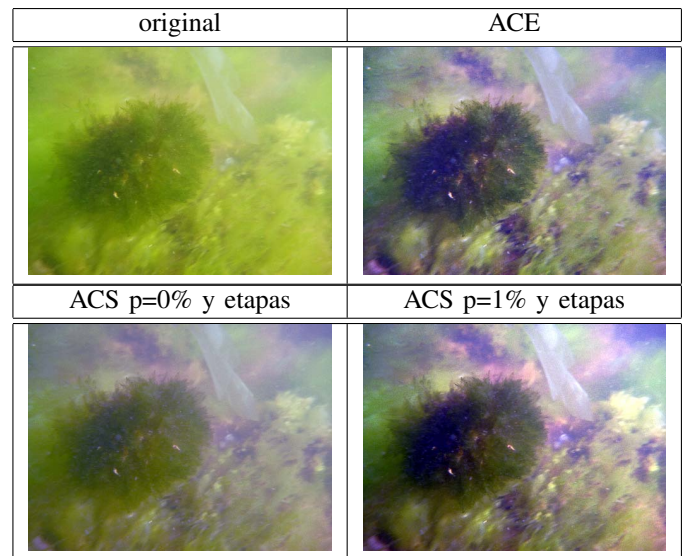


Figure 8. Reducción del color cast

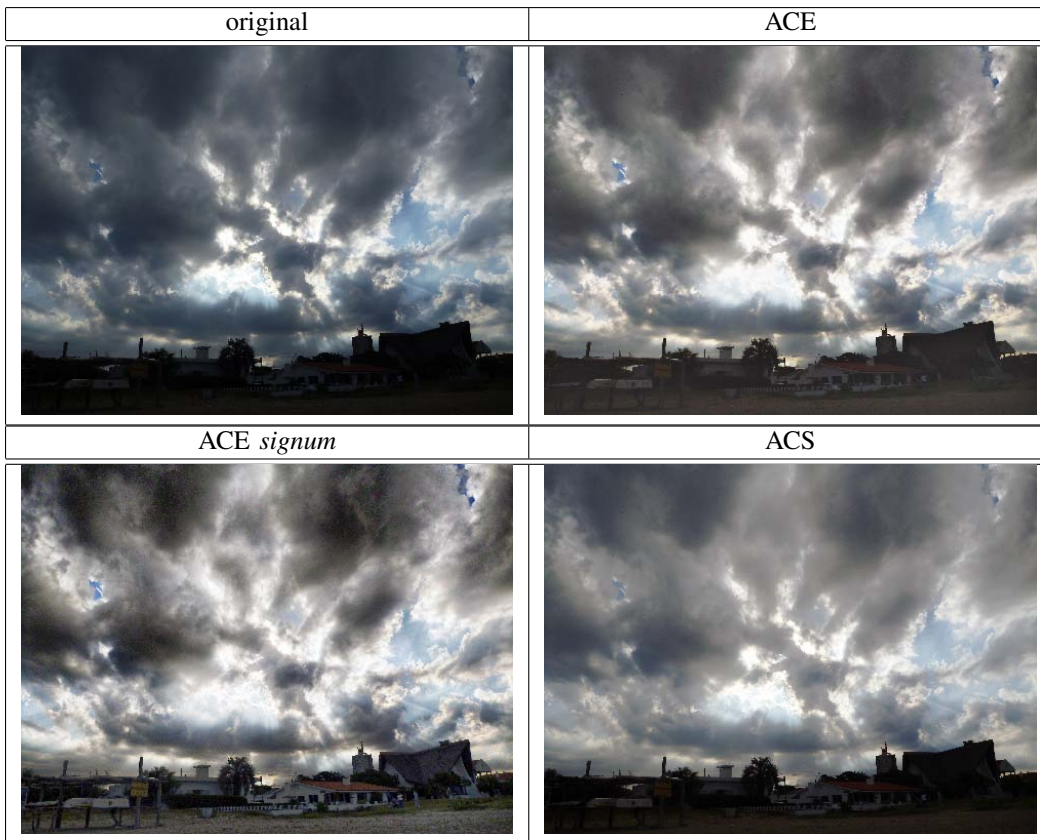


Figure 9. Fotos con luz natural

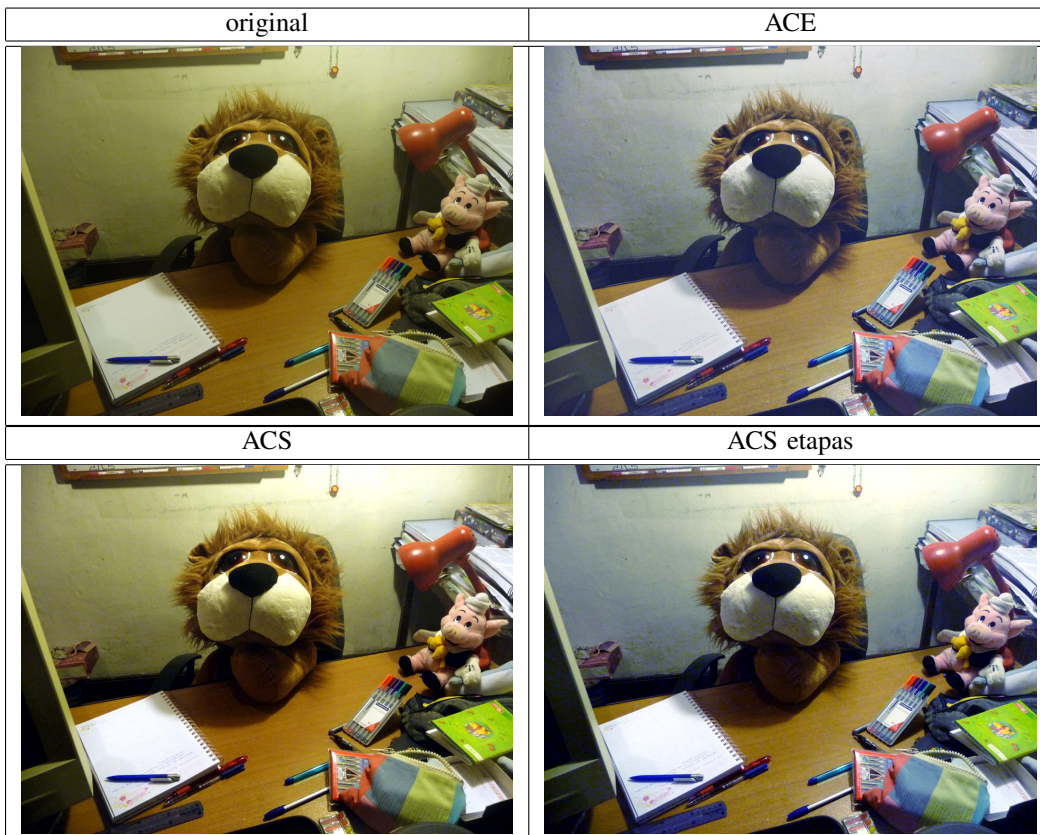
Figure 10. Eliminación del *color cast* presente en escenas bajo iluminación artificial



Figure 11. Eliminación del *color cast* y mejora de la nitidez en escenas subacuáticas



Figure 12. ACE con procesamiento con subconjuntos de pixels

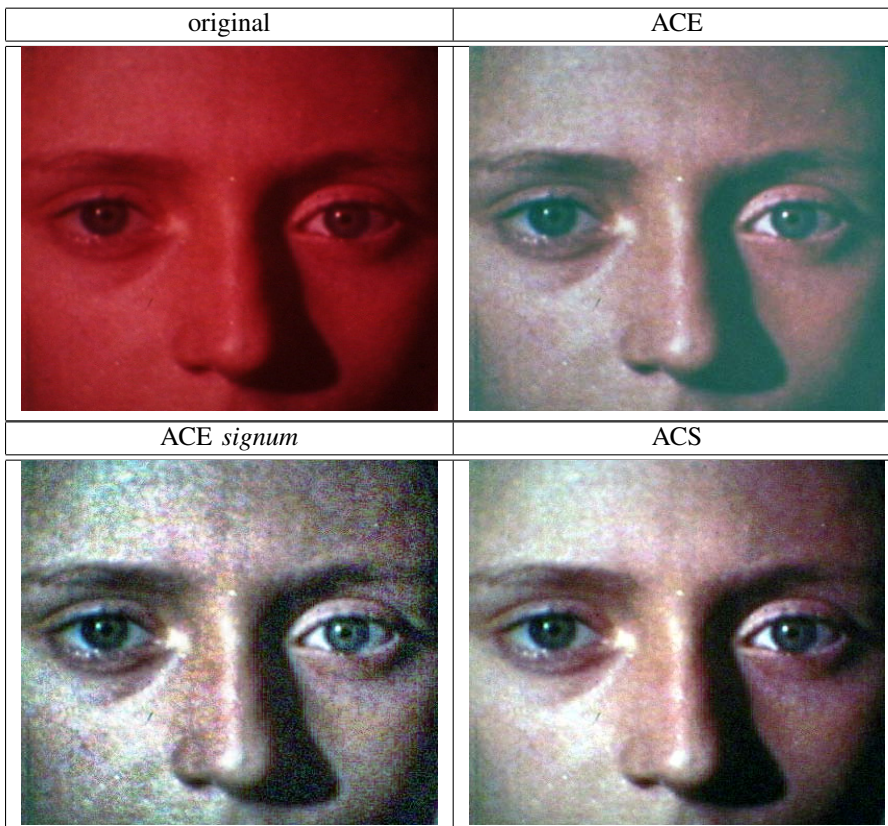


Figure 13. Corrección de *color cast*