



Reconocimiento y rastreo de imágenes en aplicaciones de Realidad Aumentada

Images recognition and tracking in Augmented Reality applications

CONTRERAS , Martha S. [1](#); BARRIOS , Erik M. [2](#) y RODRIGUEZ , Julio C. [3](#)

Recibido: 12/06/2019 • Aprobado: 21/09/2019 • Publicado 07/10/2019

Contenido

- [1. Introducción](#)
- [2. Implementación de las aplicaciones de RA](#)
- [3. Evaluación de los resultados](#)
- [4. Conclusiones](#)
- [Agradecimientos](#)
- [Referencias bibliográficas](#)

RESUMEN:

La Realidad Aumentada (RA) es una poderosa combinación de tecnologías que permite obtener una vista de un entorno del mundo real mezclada con objetos virtuales, en tiempo real. La RA mejora notoriamente la visualización de información y la interacción del usuario. El presente trabajo propone 4 aplicaciones de RA, dentro de un entorno académico, donde se describen las principales fases de desarrollo usadas en este campo, enfocando la atención en el rastreo de imágenes. Los resultados muestran tres factores que afectan notoriamente la eficacia de estas aplicaciones. Consecuentemente, se infiere que el

ABSTRACT:

Augmented Reality (AR) is a powerful combination of technologies that allows obtaining a view of a real-world environment mixed with virtual objects, in real time. AR notoriously improves information visualization and user interaction. This work proposes 4 AR applications, within an academic environment, where the main development phases used in this field are described; it focuses its attention on the images tracking. The results show three factors that significantly affect the effectiveness of these applications. Consequently, it is inferred that image tracking is optional, according to the

rastreo de imágenes es opcional, según los requerimientos del campo de aplicación de la RA, ya que se evidencia un compromiso de la precisión y la velocidad de procesamiento.

Palabras clave: Realidad Aumentada, Rastreo de imágenes, Reconocimiento de imágenes, Patrones de análisis

requirements of the RA field of application, since a compromise of accuracy and processing speed is evident.

Keywords: Augmented Reality, Image tracking, Image recognition, pattern analysis

1. Introducción

Gracias al despliegue, aumento de infraestructura y cobertura del Internet a nivel mundial, y a la posterior reducción de costos del servicio ante el usuario final, en los últimos años se ha visto cómo se han popularizado un sinnúmero de tecnologías emergentes soportadas por esta mega red. Dentro de estas tecnologías se encuentra la Realidad Aumentada (RA), un campo de la ciencia computacional e ingeniería que consiste en superponer e integrar información virtual sobre información física ya existente, en tiempo real y a través de diferentes dispositivos tecnológicos (Cabero & Barroso, 2016). Es decir, mediante la RA los usuarios pueden ver un entorno real combinado con objetos virtuales superpuestos (Choi, Oh, Choi, & Yoon, 2018).

Esta tecnología se usa comúnmente para mejorar la percepción del mundo real, al agregar conocimiento adicional sobre el escenario (Marchand, Uchiyama, & Spindler, 2016); así mismo, tiene aplicación en diversos campos (Choi, Oh, Choi, & Yoon, 2018) (Fraga-Lamas, Fernández-Caramés, Blanco-Novoa, & Vilar-Montesinos, 2018) (Chavan, 2016) (Jetter, Eimecke, & Rese, 2018), tales como:

- **Medicina:** la RA ofrece mejoras en la visualización de imágenes para el diagnóstico y planeación de cirugías, brindando mayor información para un tratamiento adecuado.
- **Entretenimiento:** la RA brinda a los jugadores mayor realismo en los entornos digitales.
- **Navegación:** sistemas GPS incluyen RA para facilitar la ubicación del usuario, ya que les brinda soporte u orientación dentro de un escenario o lugar que es nuevo para ellos.
- **Educación:** la RA permite mayor interacción y mejora la experiencia sensorial de los estudiantes. Una aplicación muy común es la implementación en los e-books.
- **Comercio electrónico:** este es uno de los campos más favorecidos con la RA. Se usa en la venta y promoción de productos.

Dentro de las aplicaciones mencionadas se incluyen funciones como inspección 3D, detección de movimiento, localización de objetos, mecanismos de control de calidad, simulación, entrenamiento y apoyo al mantenimiento de sistemas (Chavan, 2016).

Como se puede apreciar, la RA ha tenido una gran acogida en diversos campos, especialmente en el industrial y se espera que siga creciendo en los próximos años, de la mano de la Realidad Virtual (RV) (Fraga-Lamas, Fernández-Caramés, Blanco-Novoa, & Vilar-Montesinos, 2018).

Cuando se desarrollan aplicaciones de RA, los sistemas requieren determinar la ubicación del usuario y de algún elemento u objeto de su interés. Por tanto, uno de los procesos fundamentales de este campo es el rastreo de objetos, el cual consiste en estimar la pose o postura de la persona u objeto con respecto a la vista de la cámara, donde la combinación de los dos mundos (real y virtual) toma lugar (Ranjan, Patel, & Chellappa, 2019). En estas aplicaciones se requieren rastrear los objetos con 6 grados de libertad (Guo, Quan, & Jiang, 2018).

El objetivo de este trabajo consistió en desarrollar 4 aplicaciones de Realidad Aumentada, contrastando el resultado obtenido en la inclusión y en la ausencia del rastreo de imágenes con elementos como videos y objetos 3D.

El proyecto se apoyó en tres algoritmos de acceso público. El primero de ellos utiliza una cámara web para capturar una imagen; luego el sistema busca reconocer los elementos característicos que permitan analizar la información contenida en la instantánea y así, detectar si éstos coinciden o no, con los de una imagen preestablecida, llamada "marcador". Finalmente, si se encuentra que alguna zona de la instantánea concuerda con la imagen "marcador", se superpone nueva información virtual en su lugar (MathWorks, 2017).

El segundo algoritmo orienta la inclusión de objetos 3D en el desarrollo de aplicaciones de RA, permitiendo que al mover la orientación del escenario real, el objeto insertado también se reubique (GitHub, 2017).

Por último, el tercer algoritmo permite la inclusión de la función `vl_sift`, la cual se enfoca en solucionar el problema de encontrar elementos característicos par a par, cuando hay cambio de escala o rotación (VLFeat, 2017).

1.1. Fases fundamentales de desarrollo de RA

Como se mencionó anteriormente, la Realidad Aumentada (RA) está cubriendo cada vez más campos de acción y todas estas aplicaciones requieren diversos elementos para funcionar, tales como software especializado, cámaras, activadores o marcadores, dispositivos electrónicos de almacenamiento y despliegue, contenido virtual, entre otros (Ibern Ortega, 2017).

Un activador o marcador en el ámbito de la RA es un objeto físico, normalmente una imagen o símbolo, que se ubica en la escena real y se utiliza como patrón de detección para activar la RA en la ubicación del mismo (Bhattacharya & Winer, 2019). La activación es el proceso donde al detectar un marcador, se superpone el objeto virtual.

Se pueden usar tres tipos de activador (Cabero & Barroso, 2016) (Ibern Ortega, 2017): a) marcadores de posición, b) señal de geolocalización y c) códigos QR (Quick Response).

Cuando se usa el primero (marcadores de posición) se busca asociar una imagen 3D, una animación o video con una imagen impresa que se

conoce como marcador, de modo que el contenido virtual se active en el momento en que el marcador sea detectado por la cámara web. Cuando se usan las señales de geolocalización, se combinan técnicas como GPS, sistemas de búsqueda visual (CVS por su sigla en inglés) y mapeo (SLAM). Finalmente, está el caso de los códigos QR (figuras cuadradas en dos dimensiones – Ver Figura 1), donde se almacena información asociada a contenidos que puede ser desplegados con lectores que interpreten la misma tecnología (Cabero & Barroso, 2016).

Figura 1
Ejemplo de código QR



Fuente: elaboración propia con generador de códigos QR en línea.

Para que la RA tenga lugar, en el sistema desarrollado se requieren llevar a cabo 4 tareas fundamentales (Alkhamisi & Monowar, 2013):

- **Captura:** se obtiene y almacena la escena a través de un dispositivo móvil.
- **Identificación:** se localiza el marcador y se escanean sus características.
- **Procesamiento:** se analiza la escena y se detectan las coordenadas donde está ubicado el marcador.
- **Visualización:** se despliega o muestra un recurso virtual (imagen, video, objeto 3D o sonido) superpuesto en el lugar del marcador.

Los anteriores proceso o fases permiten entrever que uno de los principales focos de atención de la RA es el seguimiento del movimiento. Es más, cuando se requiere agregar información (virtual) a un escenario del mundo real, es necesario alinear con precisión los dos mundos (Marchand, Uchiyama, & Spindler, 2016).

2. Implementación de las aplicaciones de RA

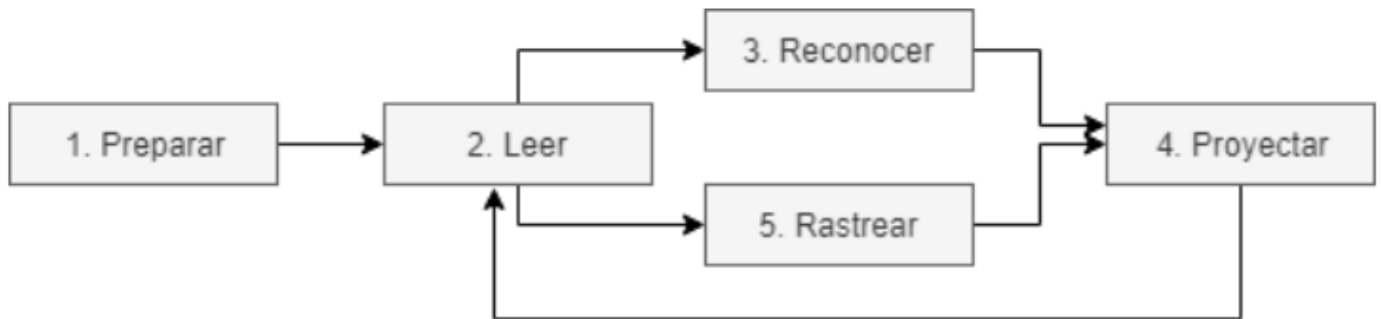
El objetivo del presente trabajo fue desarrollar 4 aplicaciones en un contexto académico, donde se generan escenarios con Realidad Aumentada, y luego contrastar la pertinencia y eficacia al incluir o no, el proceso de rastreo. Dentro de estos desarrollos también se emplearon dos opciones de elementos virtuales: un video y un objeto 3D. En esta sección se detalla cada aplicación y sus funciones básicas.

El lenguaje de programación usado fue Matlab, debido a la variedad de

funciones y recursos disponibles en el campo de visión artificial y tratamiento de imágenes.

Las aplicaciones se diseñaron incluyendo funciones propuestas por MathWorks (MathWorks, 2017), cuyo algoritmo general se compone de 5 procesos, tal como se aprecia en la Figura 2.

Figura 2
Algoritmo general de RA



Fuente: adaptado de (MathWorks, 2017)

En este algoritmo se aprecia que se agrega la fase de rastreo, respecto a las fases generales presentadas en la sección anterior.

Adicionalmente, en las aplicaciones propuestas se incluye la ejecución de la función SIFT (Scaled Invariant Feature Transform), que consiste en encontrar características propias de las escenas capturadas del mundo real, aunque cambie la escala y exista rotación (VLFeat, 2017).

En los siguientes subcapítulos, se presenta una descripción de cada aplicación propuesta dentro del proyecto.

2.1. Aplicación de RA usando rastreo con despliegue de un video

En esta propuesta se utilizan todos los elementos planteados por MathWorks, aplicando rastreo de puntos para combinar un escenario con un video. Como identificador se utilizó el logo de la UTB (Universidad Tecnológica de Bolívar) y en el escenario del mundo real se emplearon elementos con este marcador impreso.

Al momento de ejecutar esta propuesta, se observa que presenta un funcionamiento de detección rápida de características y del marcador; sin embargo, se aprecia una gran sensibilidad al movimiento de la escena. Si se presentan cambios de ubicación de los elementos dentro del escenario capturado en el mundo real, especialmente donde se encuentra el elemento de referencia, se pierde la correspondencia.

En la Figura 3 se describe cada tarea realizada como parte de las etapas implementadas en el algoritmo.

Figura 3
Etapas y funciones propuestas con rastreo

PREPARAR	<ul style="list-style-type: none"> • Cargar imagen de referencia (marcador). • Cargar video de reemplazo (element del mundo virtual). • Inicializar la cámara web • Extracción de características del escenario del mundo real.
LEER	<ul style="list-style-type: none"> • Capturar las imágenes del escenario real, a través de la cámara web.
RECONOCER	<ul style="list-style-type: none"> • Cotejar las características encontradas en el escenario real versus la imagen de referencia. • Visualizar la comparación. • Encontrar la transformación geométrica requerida.
PROYECTAR	<ul style="list-style-type: none"> • Ajustar dimensiones del video. • Insertar el video transformado (componer)
RASTREAR	<ul style="list-style-type: none"> • Rastrear puntos resultantes del movimiento. • Reubicar video de reemplazo

Fuente: adaptado de (MathWorks, 2017).

2.2. Aplicación de RA sin rastreo con despliegue de un video

Esta propuesta aplica todas las funciones de la anterior, excepto la última que corresponde al rastreo de puntos. En esta ocasión, se realizó un bucle usando vflat hasta terminar de visualizar los frames del video superpuesto.

En comparación con el algoritmo anterior, se apreció mayor efectividad en el contraste de correspondencias (menor sensibilidad a los movimientos dentro de la escena); sin embargo, se pierde velocidad.

2.3. Aplicación de RA usando rastreo con despliegue de un objeto 3D

En esta propuesta, la aplicación se basó en un algoritmo que formó parte de un proyecto de Realidad Aumentada donde se desplegaban objetos 3D, como elemento representativo del mundo virtual (GitHub, 2017).

En esta versión, se usan las coordenadas geométricas (x,y,z) para ubicar el objeto tridimensional y se tiene en cuenta la orientación posicional que presente. El resultado fue muy similar a la primera propuesta.

2.4. Aplicación de RA sin rastreo con despliegue de un objeto 3D

Esta propuesta surge básicamente retirando la fase de rastreo de la propuesta anterior. Al igual que lo que ocurrió con las dos primeras

propuestas, el rastreo o tracking le agrega rapidez al algoritmo, pero también sensibilidad al movimiento, disminuyendo precisión cuando se altera la escena, en términos de posición de los elementos.

3. Evaluación de los resultados

Cada aplicación desarrollada se analizó por separado, generando videos de ejecución. En esta sección se mostrarán las principales capturas y el análisis de los resultados.

Cabe anotar que, para poder ejecutar las aplicaciones, y en general en aquellas que aborden captura y tratamiento de imágenes, la iluminación del recinto es fundamental. Se

pudo notar que en momentos donde la luz natural disminuía, se requerían varias fuentes de iluminación artificial adicional para asegurar una captura aceptable. Otro factor relevante dentro del proceso de captura de información, fue la resolución de la cámara web usada. En el caso de presente proyecto se hicieron pruebas con dos cámaras distintas, usadas con la resolución más alta aceptada y se lograron mejores ajustes cuando se empleó la cámara web con más alta resolución. Finalmente, un tercer factor decisivo en el momento de la ejecución de las aplicaciones fue la calidad del marcador impreso usado. Es muy importante su nitidez, para evitar información indeseada (ruido) dentro de las capturas.

Para presentar el análisis, se abordarán cada una de las fases y sus respectivas tareas, con imágenes de apoyo que permitan presentar mejor los resultados obtenidos.

3.1. Preparar

En esta fase se establece y carga la imagen de referencia, llamada marcador, y el video u objeto a superponer (mundo virtual). En la Figura 4 se muestra la imagen tomada como marcador para activar la Realidad Aumentada. Si el video presenta marcos oscuros o sombras al iniciar, se sugiere obviar la lectura de los primeros frames.

Figura 4

Imagen usada como marcador de activación.



Universidad
Tecnológica
de Bolívar

Otra tarea que se incluye en esta fase es la de inicializar la cámara web. Se emplearon dos, obteniendo mejor resultado con la primera de ellas. Las funciones usadas fueron:

```
cam = webcam('EasyCamera');  
cam = webcam('HP Webcam');
```

3.2. Leer

En esta fase del desarrollo se realizan las capturas del escenario del mundo real, lo cual se logra mediante la instrucción `cameraFrame = snapshot(cam);`

3.3. Reconocer

La principal tarea desarrollada en esta fase es la de extraer las características del marcador de referencia, con el propósito de contrastarlas o compararlas con las imágenes capturadas de la escena en el mundo real. Las características normalmente obtenidas hacen referencia a las esquinas, regiones o manchas que se destacan en los elementos de la imagen evaluada.

La Figura 5 presenta la impresión en pantalla de las características obtenidas en una de las capturas.

Figura 5

Detección de características similares dentro de la fase de Reconocimiento.



Fuente: captura propia en el momento de ejecución

Para esta labor, se usó el detector de características SIFT, el cual es invariante a la rotación y cambio de escala. La función encargada de esta tarea es `relacion`, cuyos parámetros de entrada son las dos capturas a comparar. En las Figuras 6, 7 y 8 se muestra esta función:

Figura 6

Función relacion, encargada de capturar y contrastar características.

```
1 function [ CI1,CI2 ] = relacion(I1,I2)
2 %I1 imagen de referencia
3 %I2 imagen comparativa
4 %CI1 Correspondencia I1
5 %CI2 Correspondencia I2
6 %% close all
7 % clear all
8 % cld
9
10 %I1 = imread('logomenor.jpg');%imagen de referencia
11 - if size(I1,3)>1
12 - I1 = rgb2gray(I1); % If color, convert to grayscale
13 - end
14 - I1 = single(I1);% Convert to single precision floating point
15 - imshow(I1,[]);
16 % These parameters limit the number of features detected
17 - peak_thresh = 0; % increase to limit; default is 0
18 - edge_thresh = 10; % decrease to limit; default is 10
19 - [f1,d1] = vl_sift(I1, ...
20 'PeakThresh', peak_thresh, ...
21 'edgethresh', edge_thresh );
22 - fprintf('Number of frames (features) detected: %d\n',size(f1,2));
23 % Show all SIFT features detected
24 - h = vl_plotframe(f1);
25 - set(h,'color','y','linewidth',2);
```

Fuente: captura propia del código de la propuesta

Figura 7

Función relacion (continuación).

```

28 % Second image
29 I2 = imread('montar.jpg'); %imagen de referencia
30 - if size(I2,3)>1
31 - I2 = rgb2gray(I2); % If color, convert to grayscale
32 - end
33 - I2 = single(I2); % Convert to single precision floating point
34 - imshow(I2, []);
35 % These parameters limit the number of features detected
36 - peak_thresh = 0; % increase to limit; default is 0
37 - edge_thresh = 5;%10; % decrease to limit; default is 10
38 - [f2,d2] = vl_sift(I2, ...
39 'PeakThresh', peak_thresh, ...
40 'edgethresh', edge_thresh );
41 - fprintf('Number of frames (features) detected: %d\n', size(f2,2));
42 % Show all SIFT features detected
43 - h = vl_plotframe(f2) ; set(h,'color','y','linewidth',2);
44 %*****
45 % Threshold for matching
46 % Descriptor D1 is matched to a descriptor D2 only if the distance d(D1,D2)
47 % multiplied by THRESH is not greater than the distance of D1 to all other
48 % descriptors
49 - thresh = 2.0; % default = 1.5; increase to limit matches
50 - [matches, scores] = vl_ubcmatch(d1, d2, thresh);
51 - fprintf('Number of matching frames (features): %d\n', size(matches,2));
52 - indices1 = matches(1,:); % Get matching features

```

Fuente: captura propia del código de la propuesta

Figura 8

Función relacion (continuación).

```

53 - flmatch = f1(:,indices1);
54 - d1match = d1(:,indices1);
55 - indices2 = matches(2,:);
56 - f2match = f2(:,indices2);
57 - d2match = d2(:,indices2);
58 %*****
59 % Show matches
60
61 %interpretando los datos
62 - xp1=flmatch(1,:);
63 - yp1=flmatch(2,:);
64 - Pimg1=[xp1' yp1'];
65 - xp2=f2match(1,:);
66 - yp2=f2match(2,:);
67 - Pworld1=[xp2' yp2'];
68 - CI1=Pimg1;
69 - CI2=Pworld1;
70 %graficando las correspondencias
71 %showMatchedFeatures(I1, I2,Pimg1,Pworld1,'Montage');
72 |
73 - end

```

Fuente: captura propia del código de la propuesta

En este punto del algoritmo se hace necesario encontrar la transformación que se efectuó en la escena, la cual coincide con el tipo predeterminado llamado "similar". Este tipo de transformación se presenta cuando hay rotación o escalamiento de la imagen.

Para encontrar la transformación se utilizó la siguiente instrucción:

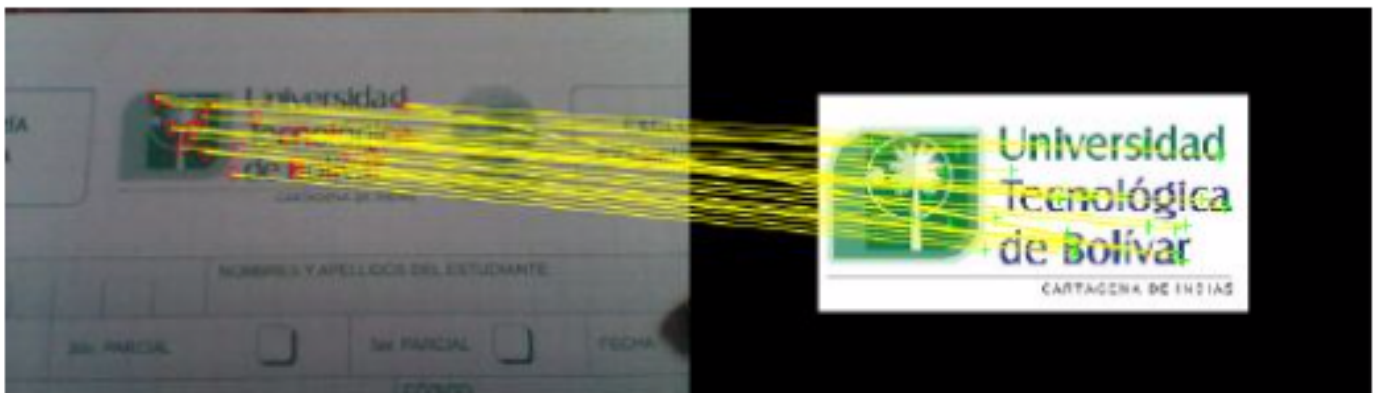
```
[referenceTransform, inlierReferencePts, inlierCameraPts]=  
estimateGeometricTransform(...
```

```
matchedReferencePts, matchedCameraPts, 'Similarity');
```

Como apoyo para analizar las correspondencias encontradas, se muestra la impresión de este resultado en la Figura 9.

Figura 9

Comparación de características entre el marcador (imagen de referencia) e instantánea capturada de la escena real.



Fuente: captura propia de la ejecución del código de la propuesta

3.4. Proyectar

Para poder proyectar la Realidad Aumentada (video superpuesto en la escena original), es necesario realizar una adaptación del contenido (reescalamiento), aplicar la transformación y finalmente componer (unir los dos mundos: real y virtual). En las Figuras 10 y 11 se encuentra apartes del código para esta finalidad y en la Figura 12 y 13 se muestran las capturas del resultado obtenido (con video y con el objeto 3D).

Figura 10

Adaptación de contenido y aplicación de la transformación.

```

77 %% Reescala el video a reemplazar
78 %carga el video a reemplazar
79 - videoFrame = step(video);
80 %obtiene las dimensiones
81 - repDims = size(videoFrame(:,:,1));
82 - refDims = size(referenceImage);
83 - tform=fitgeotrans(matchedReferencePts, matchedCameraPts, 'Similarity');
84
85 %encuentra la escala para transformar
86 - scaleTransform=tform;% fitgeotrans(repDims,refDims,'projective');
87
88 %%
89 - videoFrameScaled= imresize(videoFrame, [size(referenceImage,1) size(referenceImage,2)]);
90 - figure(2)
91 - imshowpair(referenceImage, videoFrameScaled,'Montage');
92 - %'outputView',...
93 - %         outputView);
94
95 - outputView = imref2d(size(cameraFrame));
96 - videoFrametransformed= imwarp(videoFrameScaled, referenceTransform,'OutputView',outputView);%,'OutView', outputView);
97
98 - figure(2)
99 - imshowpair(cameraFrame, videoFrametransformed,'Montage');

```

Fuente: captura propia del código de la propuesta.

Figura 11
Composición de la Realidad Aumentada

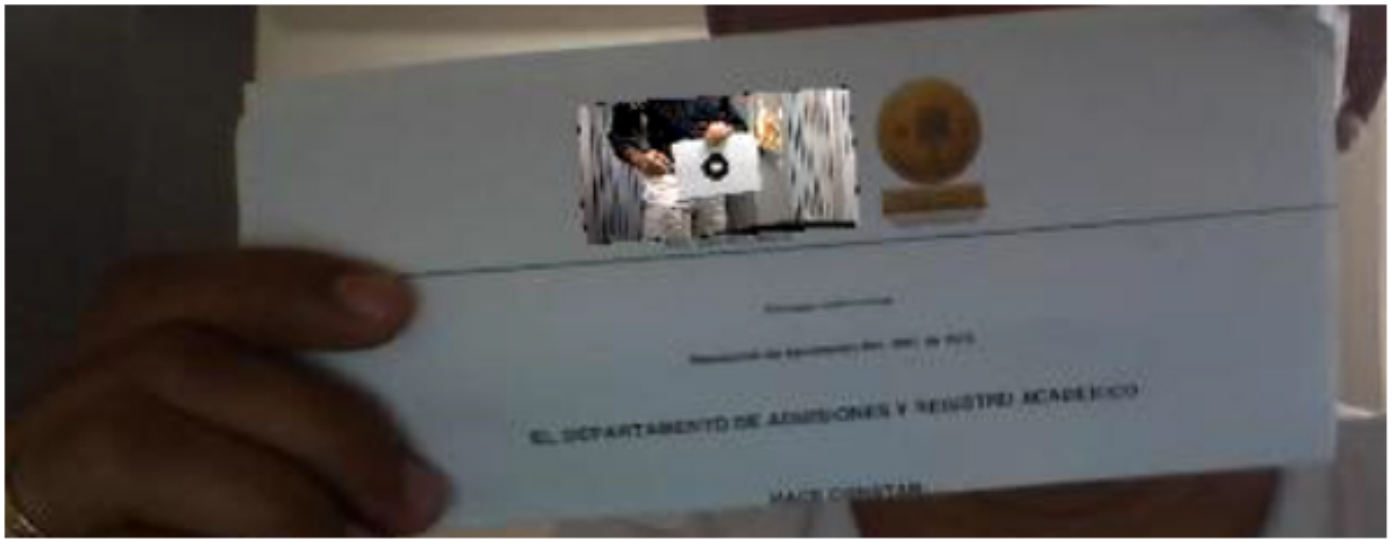
```

101 %%insertamos el video transformado
102 - alphaBlender = vision.AlphaBlender('Operation','Binary mask', ...
103     'MaskSource', 'Input port');
104
105 - mask= videoFrametransformed(:,:,1) | ...
106     videoFrametransformed(:,:,2) | ...
107     videoFrametransformed(:,:,3) > 0;
108
109 - outputFrame = step(alphaBlender, cameraFrame, videoFrametransformed, mask);
110
111 - figure(2)
112 - imshow(outputFrame);
113

```

Fuente: captura propia del código de la propuesta

Figura 12
Composición obtenida de RA con video



Fuente: captura propia de la ejecución del código de la propuesta

Figura 13

Composición obtenida de RA con objeto 3D



Fuente: captura propia de la ejecución del código de la propuesta.

3.5. Rastrear

Finalmente, en esta sección se realiza un ciclo, donde se iteran las fases de lectura, reconocimiento y proyección, haciendo un rastreo de puntos entre el marcador y cada imagen capturada. Esta fase agrega velocidad, pero disminuye precisión al alterar la escena con algún movimiento.

4. Conclusiones

La Realidad Aumentada es una tecnología de una cobertura impresionante, que ha venido tomando fuerza en los últimos años, debido a su gran versatilidad. Aunque existen numerosos estudios en el área, aún existen campos donde no se ha explorado su implementación y podría ser de gran aporte. Si bien, la Realidad Virtual surgió con anterioridad, la Realidad Aumentada ha mostrado más campos de acción aparte del entretenimiento.

En el presente proyecto se pudieron abordar las principales tareas que se deben procesar en una aplicación de RA. Se logró a pequeña escala una primera aproximación a la generación de recursos que detectan marcadores y proyectan contenido digital. Se logró apreciar la potencia de la función SIFT y en general, de la herramienta Matlab, cuando de tratamiento de imágenes y objetos digitales se trata.

La fase de rastreo es muy útil en situaciones donde la velocidad de respuesta es un factor clave. Sin embargo, si se requiere alta precisión, es recomendable omitir esta fase y agregar una función que se ocupe de los cambios dentro del escenario real, proporcionando efectividad en las correspondencias, aunque se sacrifique la velocidad de respuesta.

Como futuras mejoras al trabajo realizado, se pueden abordar escenarios al aire libre y mejorar la calidad de proyección de objetos 3D.

Agradecimientos

Este proyecto ha sido financiado con el plan de fortalecimiento de grupos de Investigaciones de la Universidad de Cartagena. Agradecemos también al doctor Andrés Guillermo Marrugo Hernández por su orientación en el curso Visión 3D impartido en la Universidad Tecnológica de Bolívar.

Referencias bibliográficas

Alkhamisi, A. O., & Monowar, M. M. (2013). Rise of Augmented Reality: Current and Future Application Areas. *International Journal of Internet and Distributed Systems*, 1(4), 25-34.

Bhattacharya, B., & Winer, E. H. (2019). Augmented reality via expert demonstration authoring (AREDA). *Computers in Industry*, 105, 61-79.

Cabero, J., & Barroso, J. (2016). The educational possibilities of Augmented Reality. *Journal of New Approaches in Educational Research*, 5(1), 44-50.

Chavan, S. R. (2016). Augmented Reality vs. Virtual Reality: Differences and Similarities. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 5(6), 1947-1952.

Choi, H., Oh, S., Choi, S., & Yoon, J. (2018). Innovation Topic Analysis of Technology: The Case of Augmented Reality Patents. *IEEE Access*, 6,

16119-16137.

Fraga-Lamas, P., Fernández-Caramés, T. M., Blanco-Novoa, Ó., & Vilar-Montesinos, M. A. (2018). A Review on Industrial Augmented Reality Systems for the Industry 4.0 Shipyard. *IEEE Access*, 6, 13358-13375.

GitHub. (2017, 12 10). *Augmented-Reality-3D-Room-Decore*. Retrieved from <https://github.com/parvparkhiya/Augmented-Reality-3D-Room-Decore>

Guo, Q., Quan, Y., & Jiang, C. (2018). Object Pose Estimation in Accommodation Space using an Improved Fruit Fly Optimization Algorithm. *Journal of Intelligent & Robotic Systems*, 1-13.

Ibern Ortega, L. (2017). *3D virtual object tracking and registration based on marker technology for AR apps (Master's thesis)*. Barcelona: Universitat Politècnica de Catalunya.

Jetter, J., Eimecke, J., & Rese, A. (2018). Augmented reality tools for industrial applications: what are potential key performance indicators and who benefits? *Computers in Human Behavior*, 87, 18-33.

Marchand, E., Uchiyama, H., & Spindler, F. (2016). Pose Estimation for Augmented Reality: A Hands-On Survey. *IEEE transactions on visualization and computer graphics*, 22(12), 2633-2651.

MathWorks. (2017, 12 10). *Object Recognition and Tracking for Augmented Reality*. Retrieved from <https://www.mathworks.com/videos/object-recognition-and-tracking-for-augmented-reality-90546.html>

Ranjan, R., Patel, V. M., & Chellappa, R. (2019). Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1), 121-135.

VLFeat. (2017, 12 9). *VLFeat.org*. Retrieved from <http://www.vlfeat.org/>

1. Candidata a doctor en Ingeniería con énfasis en Electrónica y Computación. Magíster en Ingeniería de Sistemas e Informática, Especialista en Telecomunicaciones, Ingeniera de Sistemas e Informática. Universidad de Cartagena. mcontreraso@unicartagena.edu.co

2. Magíster en Ingeniería con énfasis en Ingeniería Eléctrica y Electrónica. Especialista en Automatización y Control de Procesos Industriales. Ingeniero Electrónico. Universidad Nacional Abierta y a Distancia. erik.barrios@unad.edu.co

3. Doctor en Ingeniería de Sistemas Telemáticos. Especialista en Gestión para el desarrollo empresarial. Ingeniero de Sistemas. Universidad de Cartagena. jrodriguezr@unicartagena.edu.co

Revista ESPACIOS. ISSN 0798 1015
Vol. 40 (Nº 34) Año 2019

[[Índice](#)]

[En caso de encontrar algún error en este website favor enviar email a [webmaster](#)]