

Realidad Aumentada y Visión por Computador. Framework multipropósito

Alejandro Mitaritonna^{1,2}, Juan Lestani¹, Francisco Tarulla¹, Tomás Poeta¹, Silvana Olmedo¹,
Carolina Paez¹, Martín Lorenzo¹, Florencia Vela¹, Leonardo Inza¹

¹Laboratorio de Procesamiento de Imágenes y Visión Artificial
Universidad de Belgrano, Facultad de Ingeniería y Tecnología Informática
{alejandro.mitaritonna, juan.lestani, francisco.tarulla, tomas.poeta, silvana.olmedo, carolina.paez,
martin.lorenzo, florencia.vela, leonardo.inza}@comunidad.ub.edu.ar

²Instituto de Investigaciones Científicas y Técnicas para la Defensa (CITEDEF)
amitaritonna@citedef.gob.ar

RESUMEN

La línea de investigación consiste en el diseño y desarrollo de un framework de software multipropósito de Realidad Aumentada (RA) y Visión Artificial (VA) que se está llevando a cabo en el Laboratorio de Procesamiento de Imágenes y Visión Artificial de la Facultad de Ingeniería y Tecnología Informática de la Universidad de Belgrano (FITI-UB). Dicho framework servirá como base para la construcción de aplicaciones móviles y de escritorio para ser utilizadas en diferentes disciplinas. Uno de los principales objetivos del proyecto de investigación es la formación de recursos humanos y el fortalecimiento de las relaciones de I+D+i dentro y fuera de la FITI-UB ya sea con pares académicos y con la industria privada.

Palabras claves: *realidad aumentada, visión por computador, framework*

CONTEXTO

El diseño y desarrollo del framework surge como iniciativa de la FITI-UB con el fin de crear un laboratorio de Procesamiento de

Imágenes y Visión Artificial dedicado a la investigación y desarrollo de proyectos relacionados con la temática. Dentro de esta línea se está culminando una tesis doctoral en el Instituto de Investigaciones Científicas y Técnicas para la Defensa (CITEDEF) en el marco del Programa de Investigación y Desarrollo para la Defensa (PIDDEF) (aprobado con el Nro. 22/12), elaborado por la Subsecretaría de Investigación Científica y Desarrollo Tecnológico del Ministerio de Defensa.

INTRODUCCION

La línea de investigación presentada se inscribe dentro de las áreas de procesamiento de imágenes y visión por computador. En función de las necesidades de aprendizaje y de investigación que forman parte del diseño y desarrollo del framework propuesto se han creado 3 grupos de investigación conformados por alumnos de tercer año, cuarto y quinto año de las carreras de Licenciatura en Sistemas de Información e Ingeniería en Informática de la FITI-UB. Estas líneas/grupos de investigación se centraron en investigar y experimentar sobre:

estimación de posición y orientación (pose) y registración de objetos, reconocimiento y clasificación de objetos y motores de renderizado de objetos. En [Mit7] puede encontrarse una introducción del proyecto de investigación y, como parte de la experiencia adquirida en la construcción de frameworks de RA y VA, se puede encontrar un detalle en [Mit15].

Realidad Aumentada

Un sistema de RA es un tipo de sistema de visión sintética que mezcla gráficos generados por una computadora (anotaciones) con el mundo real. Las anotaciones proporcionan información destinada a ayudar a la toma de decisiones. El sistema de RA debe decidir qué anotaciones mostrar y cómo mostrarlas para asegurar que la información sea intuitiva y sin ambigüedades. El sistema de RA hace tracking de la posición y orientación del usuario y superpone, dentro del campo de visión del usuario, gráficos y anotaciones que están alineados con los objetos en el medio ambiente. Este enfoque tiene muchas ventajas ya que la información puede ser presentada de una manera intuitiva y directamente integrada con el medio ambiente [Ses00]. Cuando se implementa correctamente, la RA proporciona al usuario una experiencia de inmersión y permite la interacción entre el mundo virtual y el mundo real. [Mcd03] sostiene que la tecnología de aumentación o ampliación de la percepción humana conocida como RA es una herramienta moderna que están explorando los investigadores. Esta tecnología combina datos virtuales con el ambiente real observado por el usuario por lo que requiere la registración de información virtual con la escena real desde el punto de vista del usuario. [Mil94] introduce un término más genérico, realidad mixta, y lo definen como un *continuum* en donde en uno de los extremos persiste el ambiente real y en el

extremo opuesto persiste el ambiente virtual. La *Realidad Aumentada* y la *Virtualidad Aumentada* constituyen la realidad mixta o intermedia entre estos dos extremos [Ler10].

[Azu97] define tres requerimientos para las aplicaciones de RA:

1. Se deben combinar los aspectos real y virtual
2. La aplicación debe ser en tiempo real e interactiva, y
3. Los objetos deben estar registrados en el mundo tridimensional (3D)

De acuerdo a [Aba11] una aplicación de realidad aumentada tiene las siguientes partes:

- Captura de la escena real: El video capturado puede utilizarse para tracking basado en visión, es decir basado en el análisis de la imagen mediante algoritmos de visión
- Tracking del usuario: puede realizarse mediante dispositivos específicos o puede realizarse tracking basado en visión para lo cual es necesaria la captura de la escena real.
- Generación de la escena virtual: se tiene un mundo virtual, con la información de la posición y orientación del participante se genera una vista acorde del mismo
- Rendering: se combinan las imágenes del mundo real con los objetos virtuales. Los objetos virtuales se *renderizan* y se proyectan en el dispositivo de visualización.

Basándonos en lo detallado anteriormente las líneas de investigación se centraron en 3 grupos: estimación de pose y registración de objetos, clasificación de objetos y renderizado de objetos.

Estimación de Pose y Registración

En términos generales la estimación de la pose (posición y orientación) es un evento en el que se quiere detectar a partir de una serie de puntos dados en un plano, la traslación y rotación de

un objeto en función del movimiento de la cámara entre *frames*. Por otro lado, la registración es tomar estas coordenadas en el plano y proyectar un modelo 3D en éste. El proceso de registro es la superposición de objetos virtuales en una escena real utilizando información extraída de la escena [Pan06]. Como menciona [Aba11] la información virtual tiene que estar vinculada espacialmente al mundo real de manera coherente, lo que se denomina *registro* de imágenes (*registration*, en inglés). Por esto se necesita saber en todo momento la posición del usuario, *tracking*, con respecto al mundo real. Cuando se habla de pose (posición y orientación) de un objeto o de una cámara debe existir un sistema de referencia en base al cual se expresan. En RA, al hablar de imágenes registradas significa que tanto las imágenes sintéticas como el mundo real estén en referencia al mismo sistema de coordenadas.

Con el objetivo de ir entendiendo y experimentando con la librería OpenCV, se realizaron pruebas de concepto tales como: procesamiento sobre imágenes, dibujo de líneas, círculos y otros (figura 1). Se realizaron pruebas sobre espacios de colores tales como BGR (Blue, Green, Red), escalas de grises, HSV (Hue, Saturation, Value) y el espacio YUV.

Se desarrolló una aplicación para detectar la presencia de un objeto de color utilizando técnicas de visión por computador. Se realizaron pruebas de *tracking* mediante la implementación de una aplicación de reconocimiento facial (figura 2). Otra prueba en la que se trabajó fue en el calibrado de la cámara. Por otro lado utilizando los datos obtenidos por la calibración, se pudo estimar la pose en un tablero de ajedrez (figura 3). Adicionalmente se evaluó estimar la pose utilizando redes neuronales, más precisamente Redes Convolucionales (*CNN*, *Convolutional*

Neural Network, en inglés) [Tom14]. Como se verá en la subsección de *Clasificación de objetos*, el uso de Aprendizaje Automático no sólo es útil para clasificar un objeto y a partir de la detección realizar la pose y posterior registración de objetos digitales, sino también se ha experimentado para estimar la pose, por ejemplo, de un cuerpo humano o de otro tipo de objeto.

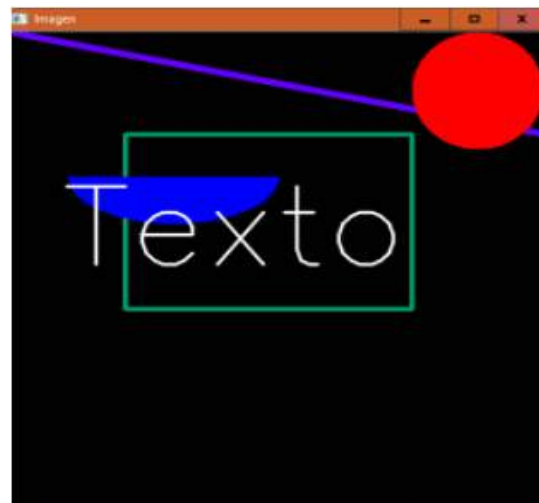


Figura 1

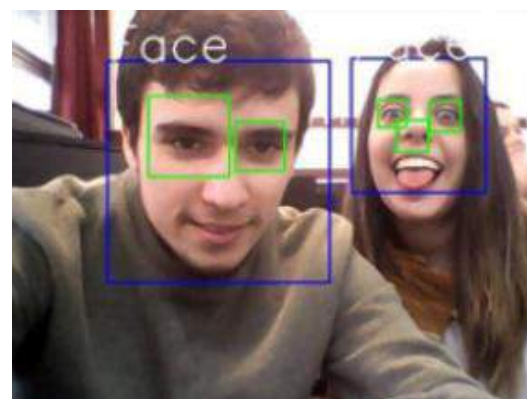


Figura 2

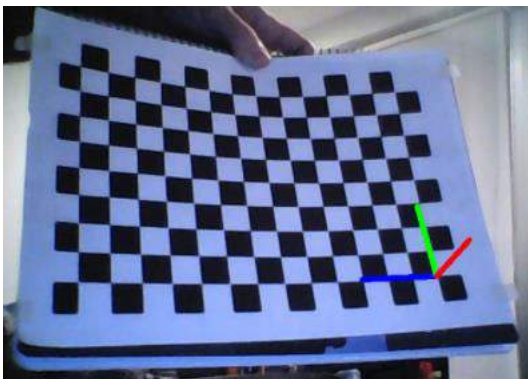


Figura 3

Clasificación de objetos

La acción de clasificar se conoce como “ordenar o dividir un conjunto de elementos en clases a partir de un criterio determinado”¹. Estos criterios pueden definirse según las diferentes características que poseen los objetos a clasificar, por ejemplo los colores, las formas, el tamaño, la textura. Las principales utilidades de los clasificadores son:

- Segmentación de imágenes.
- Reconocimiento de objetos.
- Control de calidad.
- Reconocimiento óptico de caracteres.

Una técnica utilizada para clasificar objetos es “Machine Learning”, o en español “Aprendizaje Automático”. Los sistemas de Aprendizaje Automático se utilizan para identificar objetos en imágenes, reconocimiento de voz, reconocimiento de patrones, etc. Como se menciona en [Yal15], cada vez más, estas aplicaciones hacen uso de una clase de técnicas llamadas “Deep Learning”, o en español “Aprendizaje Profundo”. Para las tareas de clasificación, las capas más altas de representación amplifican aspectos de la entrada que son importantes para la discriminación y suprimen las variaciones irrelevantes. Una imagen, por ejemplo, se presenta en forma de una matriz de píxeles, y las características aprendidas en la primera capa

¹ <https://es.oxforddictionaries.com/definicion/clasificar>

de representación típicamente representan la presencia o ausencia de bordes en una orientación y ubicación particular en la imagen. La segunda capa típicamente detecta arreglos particulares de bordes, independientemente de pequeñas variaciones en las posiciones de los bordes. La tercera capa puede combinar patrones más grandes que corresponden a partes de objetos familiares, y las capas posteriores detectarían objetos como combinaciones de estas partes. Para investigar esta línea se optó por realizar “Estudios exploratorios o Formulativos”. Se dio inicio al proceso de investigación y exploratorio formulando una hipótesis, y a partir de ahí se fue investigando de distintas fuentes, reuniendo la mayor cantidad de información posible para concluir con el objetivo. Las pruebas iniciales fueron ejecutadas sobre *Keras*. *Keras* [Ker18] es una API de Redes Neuronales de alto nivel, escrita en Python y capaz de ejecutarse sobre TensorFlow², CNTK (Microsoft Cognitive Toolkit)³ o Theano⁴. Fue desarrollado con un enfoque que posibilita la experimentación rápida. Al utilizar Keras:

- Permite la creación de prototipos fácil y rápida (a través de la facilidad de uso, la modularidad y la extensibilidad).
- Admite Redes Convolucionales y Redes Recurrentes, así como combinaciones de las dos.
- Se ejecuta sin problemas en la CPU y la GPU.

Renderizado

Se entiende por renderizado (*render*, en inglés) al proceso de generar una imagen, fotorrealista o no, partiendo de un modelo en 2D o 3D. El objetivo principal de esta línea de investigación

² TensorFlow: <https://www.tensorflow.org/>

³ CNTK: <https://docs.microsoft.com/en-us/cognitive-toolkit/>

⁴ Theano: <http://deeplearning.net/software/theano/>

se centró en el análisis de 4 motores de renderizado (incluido motores de video juegos) y cuyo objetivo fue determinar, de acuerdo a un conjunto de parámetros de performance e integración, cuál motor o motores eran los más aptos para integrarlos al framework propuesto. La renderización es el proceso final de creación de la imagen o animación 2D real de la escena preparada. Como menciona [Chu07a], en computación gráfica, el renderizado es el proceso de producir una imagen en la pantalla a partir de la descripción del modelo. Para ello es necesario un motor de renderizado, el cual tiene la función de procesar todas estas imágenes y devolver como resultado el diseño 3D [Chu07b]. Cada uno de estos motores tiene un conjunto de características que hacen que se diferencien unos de otros y delimitan los ámbitos de aplicación de cada uno de ellos. Los 4 motores de renderizado 3D utilizados en la investigación fueron:

- Unity 3D
- Ogre
- OpenScenegrph
- Panda

El objetivo principal de la evaluación consiste en identificar las características que abarca una aplicación 3D; especialmente aquellas que tienen impacto directo en la performance, tiempo de respuesta y fluidez de ejecución de las aplicaciones generadas con el motor [Uni18].

Los parámetros de evaluación fueron agrupados en 2 grupos, por un lado aquellos que están relacionados con la performance, y por el otro, aquellas características que representan otro tipo de ventajas a la hora de integrar dichos motores. Dentro de los parámetros de performance para evaluar en cada motor se seleccionó: velocidad de renderizado y calidad del renderizado. En cuanto a los parámetros de integración se optó por evaluar: lenguajes de programación que soporta, facilidad de

integración, documentación técnica, software libre y ejemplos existentes (cuadro 1). En [Fed17] aportan el criterio de evaluación y experimentación que fueron utilizados para analizar cada uno de los parámetros de evaluación de los motores seleccionados.

	Unity 3D	Ogre	OpenScen	Panda 3D
Velocidad	Medio	Alto	Medio/Bajo	Alto
Calidad	Muy buena	Buena	Buena	Buena
Lenguajes	C#, JavaScript	C#, C++, Java, Python, .NET	C++, Java, .NET, C#, Python, LUA	Python, C++
Integración	Baja	Media	Media	Media
Documentación	Alta	Alta	Media	Alta
Ejemplos	Muy buena	Buena	Buena	Muy buena
Soft. Libre	No	Si	Si	Si

Cuadro 1

Habiendo evaluados los motores de renderizado llegamos a la conclusión que los 2 motores que se adaptarían e integrarían adecuadamente al framework serian Ogre y Panda 3D. Si bien Unity 3D es muy poderoso con suficiente documentación y ejemplos, resultó de difícil integración con nuestro entorno de framework.

Entorno de Desarrollo

Con el fin de estandarizar todas las líneas de investigación se decidió utilizar como sistema operativo Ubuntu 16.04 lts, de modo que se creó y configuró una máquina virtual con el sistema operativo indicado, a través de VirtualBox. La misma se instaló sobre los host Windows 8.1, Windows 10 y MacOS. Utilizar esta distribución resultó útil pues se cuenta con

Python 2.7 instalado junto con las librerías Numpy y Matplotlib, de gran utilidad para poder hacer desarrollos de RA y VA. Por otro lado se instaló de manera separada, OpenCV 3.2.0. Para realizar la investigación y pruebas sobre Deep Learning se descargó el software Anaconda Cloud y se procedió a instalar Keras.

LINEAS DE INVESTIGACIÓN, DESARROLLO E INNOVACIÓN

- Diseño y desarrollo de un framework de RA y VA multipropósito (aplicado a diferentes disciplinas)
- Aplicaciones de RA y VA utilizando el framework propuesto
- Aplicaciones de Deep Learning aplicables al procesamiento de imágenes y clasificación de objetos
- Modelado y renderizado de objetos 3D utilizando los motores propuestos

RESULTADOS Y OBJETIVOS

A continuación se detallan los resultados obtenidos y los objetivos generales del proyecto:

- Creación del Laboratorio de Procesamiento de imágenes y Visión Artificial. Facultad de Ingeniería y Tecnología Informática. Universidad de Belgrano.
- Curso de Posgrado en Interfaces Avanzadas: Del mundo Real al Virtual⁵. Departamento de Posgrado y Educación Continua. Facultad de Ingeniería y Tecnología Informática. Universidad de Belgrano.
- Diseño y desarrollo del Framework de RA y VA multipropósito. Laboratorio de Procesamiento de imágenes y Visión Artificial. Facultad de Ingeniería y

Tecnología Informática. Universidad de Belgrano.

- Desarrollo de aplicaciones, utilizando el Framework para ser probadas en la Facultad de Ingeniería y Tecnología Informática. Universidad de Belgrano y en empresas privadas de acuerdo a las tratativas que se están gestionando con el sector.
- Aprendizaje de los alumnos en los fundamentos de procesamiento de imágenes, visión artificial y deep learning.
- Aprendizaje del uso de la librería de Procesamiento de Imágenes y Visión por Computador: *OpenCV*.
- Aprendizaje del uso del framework de Deep Learning: *Keras*.

FORMACION DE RECURSOS HUMANOS

La formación de recursos humanos es prioritaria en esta línea de investigación (procesamiento de imágenes y visión artificial) y es por esta razón que se está capacitando a un grupo de alumnos de las carreras de Licenciatura en Sistemas de Información e Ingeniería Informática de la FITI-UB. Uno de los principales objetivos del Laboratorio de Procesamiento de imágenes y Visión Artificial es la formación de recursos altamente calificados en esta temática, pues la intención de la FITI-UB es posicionarse como referente académico en esta área del conocimiento. Adicionalmente con el Curso de posgrado (Interfaces Avanzadas: Del mundo Real al Virtual) se buscará transferir a la comunidad los conocimientos obtenidos en el proceso de construcción del framework, las experiencias obtenidas en dicho proceso y los conocimientos adquiridos en materia de Procesamiento de imágenes y Visión Artificial.

En la actualidad hay en curso una tesis de posgrado, una tesina de grado y capacitación a los alumnos que participan en el proyecto en el marco de esta línea de investigación:

⁵

http://www.ub.edu.ar/sites/default/files/interfaces_avanzadas_final.pdf

- Alejandro Mitaritonna. “Realidad Aumentada para la Identificación de Objetivos Militares” (culminación de tesis de doctorado en la UNLP)
- Silvana Olmedo. “Sistema de Reconocimiento de objetos reciclables” (tesina de grado en Ingeniería Informática en la FITI-UB)
- Carolina Páez, Florencia Vela, Tomás Poeta, Martín Lorenzo y Leonardo Inza (alumnos que forman parte del grupo de investigación del proyecto *Framework multipropósito de Realidad Aumentada y de Visión Artificial*. Laboratorio de Procesamiento de imágenes y Visión Artificial. FITI-UB)

BIBLIOGRAFIA

- [Aba11] M. J. Abásolo Guerrero, C. Manresa Yee, R. Más Sansó y M. Vénere, *Realidad virtual y realidad aumentada. Interfaces avanzadas.*, La Plata, Buenos Aires, Argentina: Editorial de la Universidad Nacional de La Plata (Edulp). 47 N° 380 / La Plata B1900AJP / Buenos Aires, Argentina, 2011.
- [Azu97] R. T. Azuma. *A Survey of Augmented Reality. Teleoperators and Virtual Environments* 6, 4, 355-385, 1997.
- [Chu07a] Chua Hock Chuan, Chang Lanlan, Tan Yap-Peng and Chua Hock-Chuan. *Detection and Removal of Rainbow Effect Artifacts*. IEEE International Conference on Image Processing; San Antonio, Texas, USA. 2007
- [Chu07b] Seong-Ping Chuah, Zhenzhong Chen, Tianxiao Ye, Yap-Peng Tan and Hock-Chuan Chua. *Throughput adaptation for scalable video multicast in wireless networks*. IEEE International Symposium on Circuits and Systems, Taipei, TAIWAN (pp. 1453-1456). 2009
- [Ker18] Keras: The Python Deep Learning library. <https://keras.io/>
- [Ler10] W. Le Roux, *The use of Augmented Reality in Command and Control Situation Awareness*, South Africa: Council for Scientific and Industrial Research, 2010.
- [Mcd03] C. McDonald, *Hand Interaction in Augmented Reality*, Ottawa, Ontario, Canada: The Ottawa-Carleton Institute for Computer Science. School of Computer Science. Carleton University, 2003
- [Mil94] P. Milgram and F. Kishino, “A Taxonomy of Mixed Reality Visual Displays”, *IEICE Transactions on Information Systems*, Vol E77-D, Nr 12, Dec 1994.
- [Mit17] Mitaritonna, A.; Lestani, Juan. *Framework multipropósito de Realidad Aumentada y de Visión Artificial*. Congreso TE&ET, Universidad Nacional de La Matanza, Buenos Aires, Argentina, 2017.
- [Mit15] Mitaritonna, A.; Abásolo Guerrero, M. J. (2015) *Improving Situational Awareness in Military Operations using Augmented Reality*. Proceedings of WSCG 2015. ISBN N°:978-80-86943-72-5, 2015
- [Pan06] Y. Pang, M.L. Yuan, A.Y.C. Nee, S.K. On y Kamal Youcef-Toumi, *A Markerless Registration Method for Augmented Reality based on Affine Properties*, Australian Computer Society, Inc. Seventh Australasian User Interface. 2006.
- [Ses00] S. Sestito, S. Julier, M. Lanzagorta y L. Rosenblum. *Intelligent Filtering for Augmented Reality*. Sydney, Australia, 2000.
- [Tom14] Jonathan Tompson, Arjun Jain, Yann LeCun, Christoph Bregler. *Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation*. Advances in

Neural Information Processing Systems 27.
NIPS, 2014

[Uni18] Optimizing Graphics Performance.
<https://docs.unity3d.com/Manual/OptimizingGraphicsPerformance.html>

[Yal15] Yann LeCun, Yoshua Bengio &
Geoffrey Hinton. Deep learning. Nature volume
521, pages 436–444, 28 May 2015