

Método para detección y seguimiento de objetos con aplicaciones en realidad aumentada

Christian Nicolás Pfarher

Propuesta de
PROYECTO FINAL DE CARRERA

Director
Dr. Enrique Marcelo Albornoz

Codirector
Dr. César Martínez

2 de julio de 2012



Ingeniería Informática
Facultad de Ingeniería y Ciencias Hídricas
UNIVERSIDAD NACIONAL DEL LITORAL

1. Justificación

Las imágenes y videos presentes en el mundo digital, combinados con la potencialidad que ofrecen los ordenadores actuales, han hecho más fácil la creación de aplicaciones sofisticadas en el área de procesamiento de imágenes y visión computacional [Lag11].

Desde hace tiempo, los sistemas de realidad aumentada (AR o Augmented Reality) y reconocimiento de objetos, han pasado a ser un tema pujante en el área de visión computacional, tomando gran interés en la comunidad científica y viéndose esto reflejado en diversidad de aplicaciones: en el área del e-commerce - publicidad [LC10], robótica [hCMW08, CD10], juegos [KC10, ABB⁺01], libros interactivos [KPW09], industria [BNG⁺08]; en diferentes ambientes [KM07, hCMW08, TCG⁺08, MMT⁺08]; aplicable sobre diferentes objetos [H07b, LF05, H07a]; y llevada a cabo mediante el uso de distintos dispositivos [WRM⁺10].

Un sistema de AR reemplaza parte del mundo real con objetos virtuales (generados por computadoras), los cuales parecen coexistir en el mismo espacio que el ambiente real. Si bien existen ampliaciones a esta definición, se puede definir a un sistema AR como aquel que posee las siguientes propiedades [ABB⁺01]:

- combina objetos virtuales y reales en un mismo ambiente,
- trabaja interactivamente y en tiempo real y,
- registra (alinea) objetos reales y virtuales entre sí.

La habilidad de reconocer e identificar objetos como imágenes, texto, logos, etc. en una imagen o vídeo para luego ser usados como “patrones”, es esencial en aplicaciones de AR, además de ser usada en control de calidad, vigilancia e identificación visual [LLLC11].

Existe una distinción en realidad aumentada entre dos tipos de sistemas [But11]:

- los que usan marcadores (**Marker based**), en los cuales se identifica un “marcador artificial” colocado en el ambiente, y por ende alterando el mismo y,
- los que no usan marcadores (**Markerless**), que se sustentan en usar características naturales para identificar objetos del mundo real, es decir, que no usan un “marcador artificial” para asistir al reconocimiento del objeto.

En ambos casos, el objeto 2D o 3D virtual da la apariencia de aparecer “pegado” al marcador o a la característica natural cuando se ve a través del viewport de AR.

De la misma forma, para la detección y seguimiento de objetos, existen diferentes métodos [But11], entre los cuales se pueden nombrar:

- seguimiento basado en localización (**location based tracking**): involucra sensores como GPS, acelerómetros, giroscopios, etc. para identificar el objeto y su posición (por lo general menos precisos).
- sistemas de reconocimiento (**optical tracking**): contrariamente a los anteriores, se basa en el análisis e identificación de características a partir de la imagen que permitan identificar y obtener la posición del objeto (por lo general más precisos).
- una combinación de los dos anteriores.

En este trabajo nos enmarcaremos en sistemas de realidad aumentada sin marcadores o **Markerless** y con detección y seguimiento de objetos que no involucren sensores, es decir, **optical tracking**.

Así, se propone construir un método con las características anteriormente nombradas, que permita reconocer objetos planos e identificar su posición respecto a una cámara web, en un ambiente controlado, para la posterior aplicación de realidad aumentada.

El desarrollo de un motor propio de AR que sea fácilmente adaptable a aplicaciones específicas (comerciales, educativas, lúdicas, etc.) resulta de gran interés propio y para el grupo, ya que si bien se puede encontrar software ya desarrollado, no todos trabajan en tiempo real, otros son distribuidos bajo licencias privativas o son del tipo Marker based, entre otras características que no lo hacen de interés para su uso. Además, se estaría trabajando en una tecnología de punta que se encuentra en auge en estos tiempos y con un futuro prospero. Una de las piezas fundamentales del motor citado, son los métodos de identificación y reconocimiento de objetos los cuales serán expuestos en el presente trabajo.

Para ello, será necesario aplicar conocimientos de visión computacional [BK08, Lag11, JH00] e investigar métodos de extracción de características en imágenes [NA02]. Existen gran cantidad de investigaciones en el área, tales como:

- el estudio de descriptores visuales [OBG11] y locales [MS05, TM07, Kot11], entre los cuales se encuentran los algoritmos SURF (Speeded-Up Robust Features) [BETV08, BETV08] y SIFT (Scale Invariant Feature Transform) [LS09],
- el reconocimiento de puntos claves [OFL07],
- la detección en tiempo real de objetos [HLI⁺10],
- el seguimiento (tracking) sin marcadores [CL10],
- el análisis de características invariantes en imágenes [SL04, Low04] y
- la clasificación de puntos característicos [LF04], entre otros.

Dichos temas, se tratarán de abordar en menor o mayor medida para el correcto desarrollo del presente proyecto.

2. Objetivos

2.1. Objetivo general

- Desarrollar un método para reconocer y seguir objetos en una secuencia de vídeo digital y desarrollar un prototipo que haga uso del mismo.
- Afianzar y extender los conocimientos adquiridos en el cursado de la carrera Ingeniería en Informática.

2.2. Objetivos específicos

- Realizar el relevamiento del estado del arte en métodos utilizados para la detección y seguimiento de objetos en el Procesamiento Digital de imágenes.
- Desarrollar un método reconocedor y seguidor de objetos planos en el flujo de video tomado por una cámara web estándar, sobre un ambiente controlado.
- Implementar el método en un algoritmo computacional que pueda ser utilizado en diferentes sistemas operativos.
- Optimizar el método desarrollado para que sea aplicable en tiempo real.
- Implementar una aplicación prototipo específica (en el área de turismo, educación, publicidad, juegos u otras) aplicando el método desarrollado.

3. Alcance

Entre los alcances del proyecto se pueden enumerar los siguientes:

- El trabajo se enmarcará en un sistema del tipo Markerless con método de reconocimiento del tipo optical tracking.
- El proyecto involucra el desarrollo de un prototipo para ser utilizado en una computadora con una cámara web standard. Cabe aclarar, que no se pretende realizar una aplicación final específica y completa orientada a un usuario final.
- Se trabajará con imágenes que conformen un flujo de video en tiempo real capturado por una cámara web de computadora en un ambiente específico y controlado.
- No es un objetivo la implementación de varios métodos y su comparación. Sin embargo, en una etapa previa se evaluarán las características de varios métodos para obtener el que mejor se adecue a los requerimientos.
- Se dejará como un trabajo futuro la optimización de velocidad y/o implementación por hardware (procesamiento con GPU).

4. Metodología

4.1. Revisión y estado del arte.

Este proyecto comienza con la revisión bibliográfica y del estado de arte de los métodos disponibles para la extracción de características en imágenes/vídeo, que sean útiles para el reconocimiento y seguimiento (tracking) de objetos en tiempo real.

4.2. Identificación de características en imágenes.

Posteriormente, se buscará implementar, utilizar y/o adaptar un método o algoritmo, que permita identificar dichas características sobre objetos planos. En el proceso de detección, se utilizarán técnicas de procesamiento de imágenes, operaciones de transformaciones morfológicas, entre otras.

4.3. Proyección, matching y procesamiento.

Se procederá con la aplicación de proyecciones y/o transformaciones perspectivas o afines, matching de características entre imágenes u otras acciones relacionadas con proyecciones que se evalúen durante el desarrollo y que sean necesarias para obtener la ubicación del objeto.

En esta etapa también se pondrá en consideración la posible implementación de un pre-procesamiento para tratar de mejorar la velocidad del algoritmo.

4.4. Procedimiento de aplicación y finalización.

La aplicación del código desarrollado, se hará sobre imágenes capturadas en tiempo real, con una cámara web estándar. El procedimiento tendrá dos pasos principales: en una primera instancia y como parte del proceso de configuración, se capturará una imagen del objeto a detectar, seguidamente, se capturará flujo de vídeo en tiempo real, para ser analizado y así llevar a cabo la detección del objeto y el cálculo de su posición respecto a la cámara.

En lo que respecta a herramientas a utilizar para la codificación, existe variedad de software y librerías para la manipulación de videos e imágenes, sin embargo OpenCV (Open Source Computer

Vision)¹ es una librería open source² de visión computacional, escrita en C y C++, la cual puede ser ejecutada sobre diferentes sistemas operativos: Linux, Windows y Mac OS X, que ha sido ampliamente adoptada como la herramienta para el desarrollo en la comunidad de investigadores y desarrolladores en el campo de visión computacional [Lag11], debido a que fue diseñada para lograr gran eficiencia computacional, haciendo foco en el desarrollo de aplicaciones en tiempo real. Es por ello, que como base para el desarrollo del código fuente de nuestro método, se utilizará la citada librería. Además, se considerará la utilización de OpenGL si el proyecto así lo requiriese.

Finalmente y para la conclusión del trabajo, se aplicará el método a un prototipo de aplicación específica.

5. Plan de trabajo y de tareas propuesto

1. Revisión y estado del arte. (*Total final: 75 hs.*)

Recopilación bibliográfica:

- a) técnicas de reconocimiento y seguimiento (tracking) de objetos, (30 hs.)
- b) procesamiento de flujo de vídeo, (10 hs.)
- c) pre-procesamiento de imágenes, (10 hs.)
- d) cálculo de homografías, proyecciones y transformaciones. (25 hs.)

2. Identificación de características en imágenes. (*Total final: 145 hs.*)

- a) Adquisición de imágenes para formar un conjunto de pruebas que sirva de prototipo. (5 hs.)
- b) Implementación, uso o desarrollo de un método de extracción de características en imágenes. (*Total: 140 hs.*)
 - Análisis/estudio del método. (75 hs.)
 - Implementación o uso en el código. (75 hs.)

3. Proyección, matching y procesamiento. (*Total final: 90 hs.*)

- a) Matching de características entre imágenes. (10 hs.)
- b) Proyecciones. (10 hs.)
- c) Transformaciones. (10 hs.)
- d) Pre-procesamiento para posibles mejoras de velocidad en tiempo de ejecución. (30 hs.)
- e) Análisis homográfico. (30 hs.)

4. Procedimiento de aplicación y finalización. (*Total final: 240 hs.*)

- a) Superposición de contenido (imágenes o videos) sobre el objeto detectado. (40 hs.)
- b) Redacción del informe final del proyecto. (200 hs.)

Total del plan de trabajo y tareas propuesto: 550 hs.

¹<http://SourceForge.net/projects/opencvlibrary>

²<http://opensource.org>

6. Puntos de seguimiento

Comienzo de actividad: Noviembre de 2011.

1. Punto de control n° 1, diciembre de 2011:

- Informe de la bibliografía consultada (recopilación bibliográfica).

2. Punto de control n° 2, febrero de 2011:

- Adquisición de imágenes (prototipos) para realizar las pruebas.
- Métodos de extracción de características en imágenes parte 1 (Análisis/estudio del método).

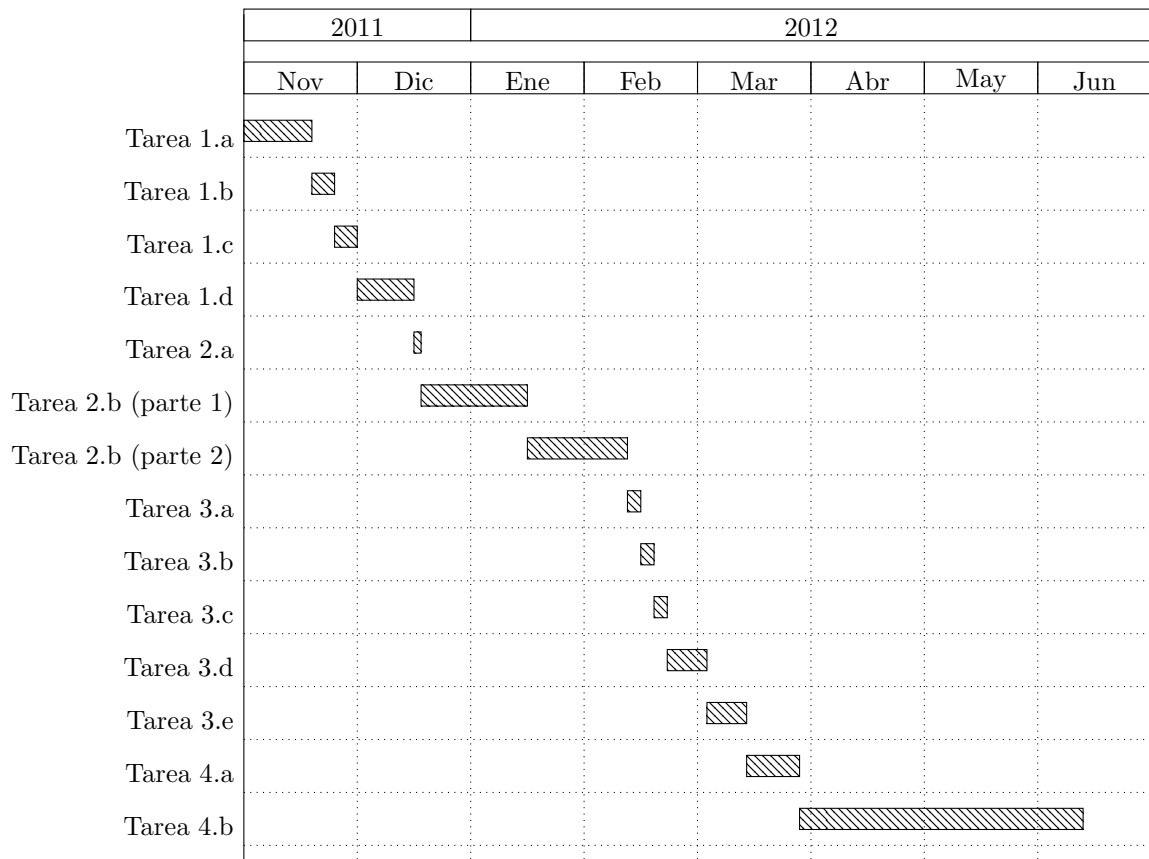
3. Punto de control n° 3, marzo de 2012:

- Métodos de extracción de características en imágenes parte 2 (Implementación o uso en el código).
- Matching de características entre imágenes.

4. Punto de control n° 4, abril de 2012:

- Pre-procesamiento para lograr posibles mejoras de velocidad, en tiempo de ejecución.
- Análisis homográfico, proyecciones y transformaciones.
- Superposición de contenido sobre el objeto detectado.

7. Cronograma tentativo



Cabe aclarar que la Tarea 8 incluye tiempos de redacción (redacciones parciales) durante el desarrollo del proyecto.

8. Recursos necesarios

- Acceso a Internet.
- Material bibliográfico.
- PC con cámara web y sistema operativo GNU/Linux.
- Software de libre distribución (IDE, compiladores, bibliotecas, control de versiones).

Los recursos mencionados anteriormente se encuentran disponibles en el Centro de Investigación en Señales, Sistemas e Inteligencia Computacional (Sinc), donde se lleva a cabo el Proyecto.

9. Presupuesto

Tipo	Detalle	Cant.	P. Unit.	P. Total
<i>Librería</i>				
	Resma de hojas blancas A4 (Entregables, Informe Proyecto Final y Borradores)	2	\$23.00	\$46.00
	Anillados	6	\$13.00	\$78.00
	Lapiceras y lápices	10	\$3.75	\$37.50
<i>Fotocopias</i>				
	Material bibliográfico	3	\$65.00	\$195.00
<i>Hardware</i>				
	Monitor LCD 23 pulgadas	1	\$1900.00	\$1900.00
	Notebook con webcam	1	\$3500.00	\$3500.00
	Webcam externa	1	\$150.00	\$150.00
<i>Recursos humanos</i>				
	Honorarios	8	\$3300.00	\$26400.00
<i>Servicios</i>				
	Acceso a internet	8	\$99.90	\$799.20
<i>Software</i>				
	Ide, librerías, compiladores, control de versión, etc	1	\$0.00	\$0.00
			Total	\$36105.70

- La totalidad del software utilizado en el proyecto es de distribución libre y gratuita, con lo cual no se aplican gastos de licencias para su uso.
- A pesar que el hardware está disponible en el SINC, se incluye en el presupuesto, ya que son recursos utilizados cuando no se está el mismo.

Referencias

- [ABB⁺01] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. Recent advances in augmented reality. *IEEE Comput. Graph. Appl.*, 21(6):34–47, November 2001.
- [BETV08] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up Robust Features (SURF). *Computer Vision and Image Understanding: CVIU*, 110(3):346–359, June 2008.
- [BK08] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O’Reilly Media, 1st edition, October 2008.
- [BNG⁺08] Selim Benhimane, Hesam Najafi, Matthias Grundmann, Yakup Genc, Nassir Navab, and Ezio Malis. Real-time object detection and tracking for industrial applications. In Alpesh Ranchordas and Helder Araújo, editors, *VISAPP (2)*, pages 337–345. INSTICC - Institute for Systems and Technologies of Information, Control and Communication, 2008.
- [But11] Ben Butchart. Augmented reality for smartphones - a guide for developers and content publishers. In *JISC Observatory*, 2011.
- [CD10] Daniel Conrad and Guilherme N. DeSouza. Homography-based ground plane detection for mobile robot navigation using a modified EM algorithm. In *ICRA* [CD10], pages 910–915.
- [CL10] Zhuo Chen and Xinyu Li. Markless tracking based on natural feature for augmented reality. In *Educational and Information Technology (ICEIT), 2010 International Conference on*, volume 2, pages V2–126 –V2–129, sept. 2010.
- [H07a] Taehee Lee 0002 and Tobias Höllerer. Handy AR: Markerless inspection of augmented reality objects using fingertip tracking. In *ISWC* [H07a], pages 83–90.
- [H07b] Taehee Lee 0002 and Tobias Höllerer. Initializing markerless tracking using a simple hand gesture. In *ISMAR* [H07b], pages 259–260.
- [hCMW08] Ian Y h Chen, Bruce Macdonald, and Burkhard Wünsche. Markerless augmented reality for robots in unprepared environments, 2008.
- [HLI⁺10] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Pascal Fua, and Nassir Navab. Dominant orientation templates for real-time detection of texture-less objects. In *CVPR*, pages 2257–2264. IEEE, 2010.
- [JH00] B. Jahne and H. W. Haussecker. *Computer Vision and Applications: A Guide for Students and Practitioners*. Academic Press, April 2000.
- [KC10] Jong-Hyoun Kim and T. Cho. The initiative experiments for utilizing real cards in online trading card game. In *Computer Sciences and Convergence Information Technology (ICCIT), 2010 5th International Conference on*, pages 182 –185, 30 2010-dec. 2 2010.
- [KM07] Georg Klein and David W. Murray. Parallel tracking and mapping for small AR workspaces. In *ISMAR* [H07b], pages 225–234.

- [Kot11] Michal Kottman. Planar object detection using local feature descriptors. In *Information Sciences and Technologies Bulletin of the ACM Slovakia, Special Section on Student Research in Informatics and Information Technologies*, pages 59–63, 2011.
- [KPW09] Kiyoungh Kim, Jonghee Park, and Woontack Woo. Marker-less tracking for multi-layer authoring in AR books. In Stéphane Natkin and Jérôme Dupire, editors, *ICEC*, volume 5709 of *Lecture Notes in Computer Science*, pages 48–59. Springer, 2009.
- [Lag11] Robert Laganière. *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Publishing, June 2011.
- [LC10] Xinyu Li and Dongyi Chen. Augmented reality in e-commerce with markerless tracking. In *Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference on*, pages 609–613, april 2010.
- [LF04] Vincent Lepetit and Pascal Fua. Towards recognizing feature points using classification trees, 2004.
- [LF05] Vincent Lepetit and Pascal Fua. Monocular model-based 3d tracking of rigid objects. *Found. Trends. Comput. Graph. Vis.*, 1(1):1–89, January 2005.
- [LLLC11] Ahyun Lee, Jae-Young Lee, Seok-Han Lee, and Jong-Soo Choi. Markerless augmented reality system based on planar object tracking. In *Frontiers of Computer Vision (FCV), 2011 17th Korea-Japan Joint Workshop on*, pages 1–4, feb. 2011.
- [Low04] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [LS09] T. W. R. Lo and J. P. Siebert. Local feature extraction and matching on range images: 2.5D SIFT. *Computer Vision and Image Understanding*, 113(12):1235–1250,, December 2009.
- [MMT⁺08] Tsutomu Miyashita, Peter Georg Meier, Tomoya Tachikawa, Stephanie Orlic, Tobias Eble, Volker Scholz, Andreas Gapel, Oliver Gerl, Stanimir Arnaudov, and Sebastian Lieberknecht. An augmented reality museum guide. In *ISMAR*, pages 103–106. IEEE, 2008.
- [MS05] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, October 2005.
- [NA02] Mark S. Nixon and Alberto S. Aguado. *Feature extraction and image processing*. Newnes, Oxford, 2002.
- [OBG11] Òscar Boullosa Garcìa. Estudio comparativo de descriptores visuales para la detección de escenas cuasi-duplicadas. In *Proyecto Fin De Carrera, Universidad Autónoma De Madrid Escuela Politécnica Superior*, 2011.
- [OFL07] M. Ozuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *CVPR*, pages 1–8, 2007.
- [SL04] Iryna Skrypnyk and David G. Lowe. Scene modelling, recognition and tracking with invariant image features. In *ISMAR [SL04]*, pages 110–119.

- [TCG⁺08] Gabriel Takacs, Vijay Chandrasekhar, Natasha Gelfand, Yingen Xiong, Wei-Chao Chen, Thanos Bimpigiannis, Radek Grzeszczuk, Kari Pulli, and Bernd Girod. Outdoors augmented reality on mobile phone using loxel-based visual feature organization. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, MIR '08, pages 427–434, New York, NY, USA, 2008. ACM.
- [TM07] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2007.
- [WRM⁺10] Daniel Wagner, Gerhard Reitmayr, Alessandro Mulloni, Tom Drummond, and Dieter Schmalstieg. Real-time detection and tracking for augmented reality on mobile phones. *IEEE Trans. Vis. Comput. Graph.*, 16(3):355–368, 2010.