

Sistema de Realidad Aumentada para Planificación Neuroquirúrgica Basado en Dispositivos Móviles de Uso Masivo

Pablo Roncagliolo^{1,2}, Agustín González¹, Antonio Orellana³, Paolo Massaro³

¹Departamento de Electrónica, Universidad Técnica Federico Santa María, Chile

²Departamento de Ingeniería Biomédica, Universidad de Valparaíso, Chile

³Servicio de Neurocirugía, Hospital Carlos Van Buren, Valparaíso, Chile

pablo.roncagliolo@uv.cl

Resumen— La Realidad Aumentada, consiste en la tecnología capaz de añadir información virtual sobre un contexto real. En la década de los '90 se realizaron numerosas investigaciones que permitieron implementar complejos sistemas basados en diferentes medios de visualización (pantallas, lentes, cascos) y captura de imagen (cámaras, sensores). Actualmente se abren nuevas posibilidades de desarrollo y masificación de los sistemas de Realidad Aumentada, por cuanto es factible implementarlos en dispositivos móviles de uso masivo. El presente trabajo, analiza algunos aspectos relevantes en la implementación de un sistema de realidad aumentada en tiempo real sobre dispositivos móviles. Específicamente se determina la capacidad de capturar y procesar una imagen para determinar la transformación de coordenadas del sistema. El sistema se programó en lenguaje Java, utilizando un equipo Nokia 6670. Dada las restricciones técnicas y de rendimiento de estos dispositivos, se tuvo especial cuidado en la utilización de algoritmos eficientes en el procesamiento de las imágenes y en particular se propone una modificación a uno de los algoritmos clásicos de calibración de cámaras para este tipo de sistemas. Los resultados del trabajo indican que es posible implementar sistemas de realidad aumentada en equipos móviles de nueva generación.

Palabras Claves— Realidad Aumentada, Procesamiento de Imágenes, Dispositivos Móviles, Planificación Neuroquirúrgica.

I. INTRODUCCION

Los sistemas de Realidad Aumentada, corresponden a un conjunto de tecnologías de adquisición, procesamiento y despliegue de imágenes, que permiten superponer información virtual, sobre un contexto real. Esta información consiste generalmente en textos, imágenes y objetos 3D, la cual está relacionada con las imágenes o videos reales, y donde la perspectiva y posición de los objetos virtuales tiene directa relación con el sistema de coordenadas "real".

En conjunto con el desarrollo de la realidad virtual, durante las décadas pasadas, se desarrollaron complejos sistemas de realidad aumentada, que estaban compuestos por diversos sistemas de captura de imágenes y sensores de posición, computadores para el procesamiento de los datos y sistemas de visualización basados en diversas tecnologías como monitores, lentes, cascos, proyectores, etc.

La etapa fundamental para la implementación de un sistema de realidad aumentada, es la capacidad de relacionar el "eje de coordenadas virtuales" con el "eje de coordenadas reales". Para ello, el sistema debe extraer de manera manual o automática, información desde las imágenes reales que permitan determinar la posición u orientación relativa del dispositivo de captura respecto del contexto real. Este proceso se denomina "calibración de cámara" o POSE. El resultado final del proceso de calibración es obtener una matriz de transformación entre ambos sistemas de coordenadas. Existen muchos métodos, algunos basados en algoritmos iterativos y otros en métodos directos [1]. Muchas de las publicaciones iniciales relacionadas con realidad aumentada abordaron esta temática. Posteriormente la mayoría de las publicaciones en el área se orientaron a las aplicaciones, entre las que destacan las aplicaciones médicas, en museos, en juegos, en arquitectura, en diseño, entre otras.

Sin embargo, actualmente se abren nuevas posibilidades de desarrollo y masificación de los sistemas de Realidad Aumentada, por cuanto es factible implementarlos en dispositivos móviles de uso masivo. Los nuevos dispositivos móviles como celulares y PDA poseen una mayor capacidad de cómputo, mejores interfaces gráficas y periféricos incorporados, como cámaras y tarjetas de comunicación inalámbrica. Además, tanto los sistemas operativos, como las herramientas de desarrollo son cada vez más compatibles y abiertos, lo que ha potenciado el desarrollo de aplicaciones más complejas que requieren de procesamiento en tiempo real, adquisición de imágenes y capacidad de comunicación.

El presente trabajo, aborda un aspecto central en los sistemas de realidad aumentada como lo es el método de calibración de la cámara.

II. DESARROLLO

Se implementó un sistema de realidad aumentada para dispositivos móviles, que consiste en un software de adquisición y procesamiento de las imágenes capturadas, desde las cuales se extrae la información de una marca de referencia externa o patrón, que permite calibrar la posición relativa del dispositivo móvil en cada iteración. Sobre la

imagen capturada se despliega información virtual, por ejemplo una imagen o un objeto 3D, cuya perspectiva mantiene total coherencia con la referencia externa, como se observa en la figura 1.

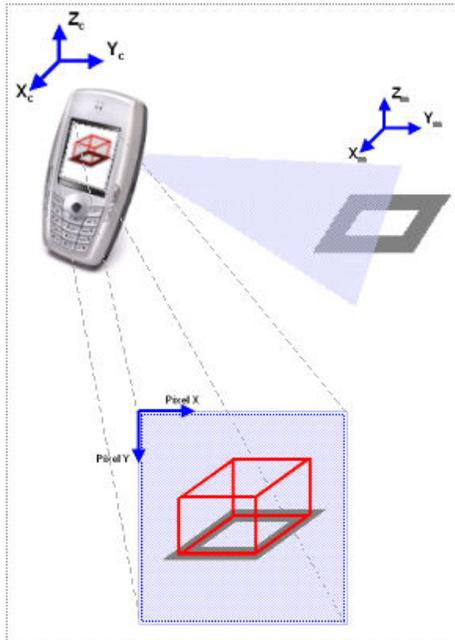


Fig. 1. Esquema General del Sistema de Realidad Aumentada utilizando un dispositivo móvil.

El modelo básico de calibración del sistema, considera la utilización de un dispositivo móvil (cámara) y un patrón de referencia. Este patrón de referencia posee dimensiones conocidas y se le asigna un sistema de coordenadas propio con origen en el centro del patrón. Por ejemplo para un patrón rectangular, las cuatro esquinas (conocidas) corresponderán a las "marcas de referencia" para la calibración del sistema. Como se observa en la figura 2 las marcas de referencia se denominan p1, p2, p3 y p4, y el sistema de coordenadas del patrón se denomina Xm/Ym/Zm. A su vez la cámara posee un sistema de coordenadas propio denotado Xc/Yc/Zc con origen en el centro del sensor sobre el cual se proyecta la imagen. El modelo de la cámara corresponde a un modelo simplificado denominado "pin-hole" donde no se considera el efecto del uso de lentes, por el contrario, se modela en base a una apertura infinitesimal.

Finalmente el modelo considera los ejes de coordenadas propios de la imagen digital resultante. Esta imagen obviamente sólo posee un eje X e Y, en unidades de píxeles.

El modelo planteado contempla dos transformaciones principales.

- Transformación nº1: La transformación desde el eje de coordenadas de las marcas de referencias hacia el eje de coordenadas de la cámara.
- Transformación nº2: Desde el eje de coordenadas de la cámara hasta el eje de coordenadas de la imagen digital.

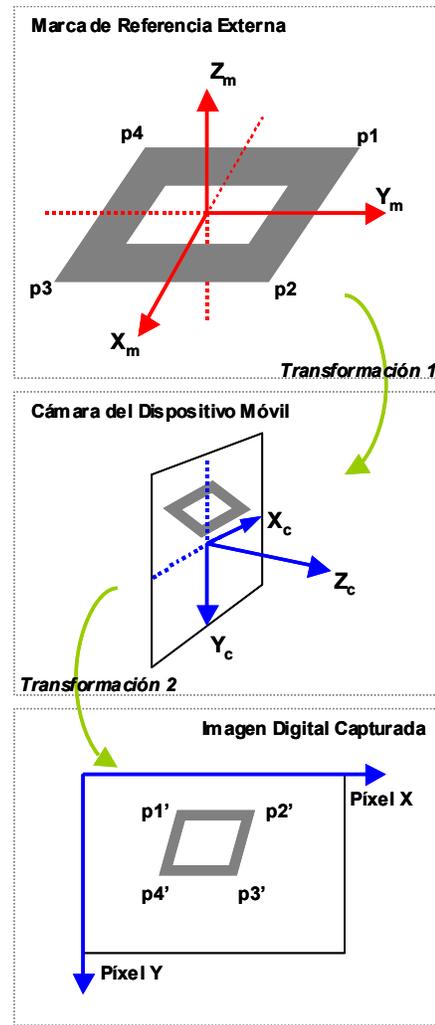


Fig. 2. Transformaciones de sistema de coordenadas.

La transformación nº1 requiere una operación de rotación y una de traslación. Utilizando coordenadas homogéneas:

$$\begin{bmatrix} wXc \\ wYc \\ wZc \\ w \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Xm \\ Ym \\ Zm \\ 1 \end{bmatrix}$$

En la transformación nº2, al sistema de coordenadas de la cámara con origen en el centro del sensor y con unidades en sistema métrico, se aplican dos operaciones: i) traslación del origen hasta la esquina del sensor; ii) cambio de escala y de unidades, de unidades métricas a píxeles. Esta transformación se modela con la siguiente matriz utilizando coordenadas homogéneas:

$$\begin{bmatrix} h \cdot P \\ h \cdot Q \\ h \end{bmatrix} = \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Xc \\ Yc \\ Zc \end{bmatrix}$$

Donde f_x y f_y son los factores de cambio de escala para los ejes P y Q respectivamente. Las constantes x_0 e y_0 corresponden a la traslación del origen hacia la esquina superior izquierda de la imagen.

Considerando ambas transformaciones en serie, el esquema general de la transformación queda representado por la siguiente matriz, donde existen 12 variables a determinar.

$$\begin{bmatrix} h \cdot P \\ h \cdot Q \\ h \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix}$$

La expresión anterior se puede expresar como el siguiente sistema de ecuaciones lineales:

$$\begin{aligned} h \cdot P &= a_{11} X_m + a_{12} Y_m + a_{13} Z_m + a_{14} \\ h \cdot Q &= a_{21} X_m + a_{22} Y_m + a_{23} Z_m + a_{24} \\ h &= a_{31} X_m + a_{32} Y_m + a_{33} Z_m + a_{34} \end{aligned}$$

Reemplazando la tercera ecuación en las dos primeras se obtiene:

$$\begin{aligned} a_{11} X_m + a_{12} Y_m + a_{13} Z_m + a_{14} - a_{31} X_m P - a_{32} Y_m P - a_{33} Z_m - a_{34} P &= 0 \\ a_{21} X_m + a_{22} Y_m + a_{23} Z_m + a_{24} - a_{31} X_m Q - a_{32} Y_m Q - a_{33} Z_m - a_{34} Q &= 0 \end{aligned}$$

Las 12 incógnitas presentes en estas dos ecuaciones se pueden obtener teóricamente si se tienen a lo menos 12 ecuaciones linealmente independientes, para lo que se requiere un mínimo de 6 puntos (X_m, Y_m, Z_m) conocidos y sus respectivas proyecciones medidas en píxeles (P, Q) .

Ambas ecuaciones se pueden expresar de forma matricial, como un sistema de ecuaciones homogéneo donde todos los términos independientes de las ecuaciones son nulos. Sin embargo este sistema homogéneo se puede transformar en un sistema no homogéneo, suponiendo que el valor de alguna incógnita es distinto de cero, por ejemplo a_{34} , y fijando momentáneamente su valor en 1. Se obtiene así el siguiente sistema de ecuaciones no homogéneo con 11 incógnitas:

$$\begin{bmatrix} X_m & Y_m & Z_m & 1 & 0 & 0 & 0 & 0 & -X_m P & -Y_m P & -Z_m P \\ 0 & 0 & 0 & 0 & X_m & Y_m & Z_m & 1 & -X_m Q & -Y_m Q & -Z_m Q \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \\ a_{31} \\ a_{32} \\ a_{33} \end{bmatrix} = \begin{bmatrix} P \\ Q \end{bmatrix}$$

Como existen dos ecuaciones independientes se requieren a lo menos 6 puntos conocidos del sistema para estructurar el sistema de ecuaciones final en base a 11 o 12 ecuaciones.

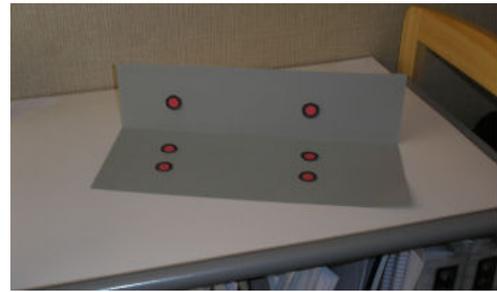


Fig. 3. Patrón de referencia de 6 puntos.

Para evaluar este método se diseñó el patrón de referencia que se observa en la figura 3, con 6 puntos distantes unos de otros, con lo cual se logra aumentar la independencia de las ecuaciones. De esta misma manera, los puntos no pueden pertenecer a un mismo plano, de lo contrario, el sistema no tiene solución.

Mediante técnicas de procesamiento de imágenes es posible determinar los 6 puntos de referencia del patrón en el plano de la imagen P,Q. Dado que las dimensiones del patrón deben ser conocidas, entonces los 6 puntos en el sistema X_m, Y_m, Z_m , también son conocidos. Con esta información se logra el sistema final de ecuaciones.

Para resolver este sistema es posible utilizar algunos de los métodos directos entre los cuales los más utilizados son: Eliminación de Gauss, Gauss-Jordan, Matriz Inversa, Cramer (Determinantes). Además existen otros métodos "iterativos" como Jacobi y de Gauss-Seidel, y los métodos denominados de "descomposición" como LU, de Crout, de Gauss-Crout y de Choleski. Sin embargo, dado que el requerimiento principal de la aplicación propuesta en este trabajo (Realidad Aumentada en Equipos Móviles) es lograr un algoritmo de calibración en tiempo real, se debe priorizar aquel método con el menor tiempo de cálculo para resolver las 11 incógnitas.

Para ello se realizaron diferentes pruebas en MATLAB, analizando el número de operaciones (Flops) requeridas por cada uno de los siguientes métodos:

Método	Nº Flops
Gauss (Matlab)	1518
Gauss (Implementado)	884
Gauss-Jordan (Matlab)	5338
Gauss-Jordan (Implementado)	4092
Inversa (Matlab)	2419
Inversa (Implementado)	7746
Cramer (Implementado)	9283

Los resultados medidos en Flops para una imagen estándar se observan en la tabla anterior y en la figura 4. Además en la figura 4 se presentan tres imágenes que utilizan los resultados obtenidos a través del método de Gauss implementado. La matriz de proyección calculada en base a las 6 marcas de referencia, permite realizar una proyección correcta de un elemento tridimensional.

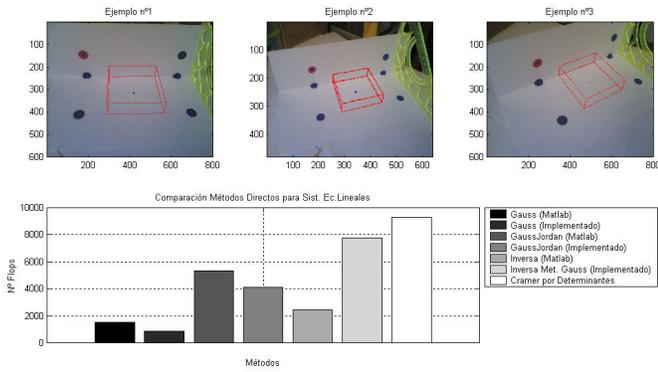


Fig. 4. Resultados

Como era de esperar el algoritmo más eficiente corresponde al método de Eliminación de Gauss. Los otros métodos poseen otras ventajas y entregan diferentes resultados adicionales como la matriz inversa y las matrices L y U. Además algunos métodos como el Gauss-Jordan presentan ventajas desde el punto de vista de la simpleza del algoritmo. El método clásico de Gauss posee dos fases: en primer lugar se debe obtener una matriz triangular superior de coeficientes; y en segundo lugar se debe realizar un proceso de "sustitución hacia atrás" de las incógnitas. No obstante, desde el punto de vista del número de operaciones requeridas por cada método, no hay duda que el método de Gauss es el más eficiente.

Los resultados indican que el algoritmo más eficiente fue el implementado en este trabajo, por cuanto requiere un menor número de operaciones (flops). Sin realizar un análisis exhaustivo de la implementación incluida en el software MATLAB, pues corresponde a una función compilada del software, se puede constatar que se logró una mejora significativa. Esto se debe a una simple modificación del algoritmo que se detalla a continuación.

A. Mejora del Método de Eliminación de Gauss

Si se observa la matriz triangular inferior de coeficientes, se puede constatar que existe un número significativo de coeficientes iguales a "0". Además, dado que la tarea principal de la primera fase del método consiste en la eliminación de coeficientes de la matriz triangular inferior, existe un número cercano al 50% de iteraciones en las que no es necesario realizar la multiplicación de la fila pivote y la posterior substracción a la fila analizada. Para ello agregó una simple instrucción de comparación "IF" al método tradicional de Gauss, lo que permitió disminuir el número de operaciones como se observa en los resultados.

III. APLICACIÓN Y TRABAJOS FUTUROS

El desarrollo anterior se enmarca en un proyecto mayor que consiste en un sistema de "Realidad Aumentada" para planificación neuroquirúrgica, que permite la superposición referenciada de leyendas u objetos virtuales sobre la imagen de un paciente capturada en tiempo real por la cámara. Mediante algoritmos de procesamiento de imágenes la aplicación decodifica la distancia y ángulo de visión del dispositivo móvil, la cual es retransmitida a un computador

(servidor) que posee las imágenes médicas del paciente (TAC), extrayendo la "vista" correspondiente de la imagen según la posición espacial del equipo móvil. De esta forma, por ejemplo, es posible visualizar de manera interactiva a través de un equipo móvil, la posición de una marca, por ejemplo un tumor. Dicha marca se debe realizar previamente en el software de planificación. El médico podría utilizar este sistema para evaluar al paciente *in situ* antes y durante la operación y marcar las zonas de abordaje. Existe un gran potencial para esta aplicación dada la masificación y bajo costo de los dispositivos móviles, lo que permite vislumbrar un futuro cercano donde cada cirujano pueda visualizar y confirmar ciertos procedimientos prequirúrgicos, simplemente "observando" a través de su celular o PDA.



Fig. 5. Resultado esperado.

IV. CONCLUSIONES

Se evaluó el correcto funcionamiento del sistema implementando el algoritmo en lenguaje J2ME [3], y utilizando un equipo Nokia 6670. Este equipo se puede considerar como un dispositivo móvil de uso masivo y representativo del mercado actual de teléfonos celulares. Dada las restricciones técnicas y de rendimiento de estos dispositivos, se tuvo especial cuidado en la utilización de algoritmos eficientes en el procesamiento de las imágenes y en particular se propuso una modificación a uno de los algoritmos clásicos de calibración de cámaras para este tipo de sistemas.

Los resultados preliminares del trabajo indican que es posible implementar sistemas de realidad aumentada en equipos móviles estándares.

REFERENCIAS

[1] D. Aracena, P. Campos, C. Tozzi. "Comparación de técnicas de calibración de cámaras digitales", Rev. Fac. Ing. - Univ. Tarapacá, Vol.13, #1, pp.57-67. Abril, 2005.
 [2] Hirokazu Kato, "ARToolKit Project", Fecha de consulta Nov. 2006 <http://www.hitl.washington.edu/artoolkit/>
 [3] M. Firtman. "Programación para Celulares con Java", MP Ediciones, Ed. #1, Buenos Aires, Argentina. 2004