Tracker Basado en Regiones de Covarianza utilizando ajuste con dos vistas

Maximiliano Guerrero, Erick Svec

Visión por Computador, Departamento de Ciencias de la Computación, Escuela de Ingeniería, Pontificia Universidad Católica de Chile
{mvguerre, evsvec} @uc.cl

Abstract. El siguiente informe presenta un método para hacer tracking de objetos usando múltiples vistas. Las cámaras correspondientes a las vistas, tienen en su campo de visión un área en común. Cada cámara esta calibrada usando puntos conocidos del espacio 3D. El método usa el sistema de tracking por Región de Covarianza y un ajuste que utiliza la información de color de ambas cámaras discriminando por la posición del objeto obtenida de cada vista. Este método ha sido probado con dos cámaras calibradas siguiendo personas, obteniendo variados desempeños que llegan hasta el 100% midiendo en frames en los que se ha seguido correctamente sobre el total del set.

Keywords: Tracking, multi view, Covariance Region, Position Adjustment.

1 Introducción

El proyecto del curso consiste en la realización de un programa que sea capaz de seguir personas en un sistema de visión por computador estéreo de dos vistas calibradas.

El usuario deberá marcar con el mouse un bounding box de una persona (o una parte de ella) en alguna imagen de la secuencia de la cámara 1 (ó 2), y el algoritmo de tracking deberá seguir a esta persona en los siguientes frames, idealmente hasta que salga del campo visual. La salida del algoritmo será un bounding box en ambas cámaras en los siguientes frames indicando dónde se encuentra la persona que se está siguiendo. El algoritmo deberá usar técnicas de tracking monocular e incluir información estéreo que haga más robusto el seguimiento.

2 Estado del Arte

La seguridad por monitoreo es una área activamente investigada en la comunidad de visión por computador. El propósito de esta investigación es alivianar la carga de las personas que están monitoreando el sistema. Este trabajo involucra procesar grandes cantidades de información generada por las cámaras. Dado que una secuencia de video puede ocupar mucho espacio de almacenamiento, muchas veces estas secuencias de video deben ser procesadas en vivo. Esto hace vulnerable al sistema de monitoreo ya que el operador debe estar procesando gran cantidad de información de todas las cámaras. Con los algoritmos de seguimiento se busca reducir la cantidad de información que el operador debe seguir y así reducir la probabilidad de que se le pasen eventos importantes. Los sistemas de monitoreo se pueden aplicar a varias industrias, como por ejemplo el monitoreo del trafico en bancos, aeropuertos o zonas concurridas.

Una vez que se tienen las cámaras y el equipo para procesar las imágenes se quiere detectar los objetos que se mueven en el rango de visión de la cámara. En nuestro caso particular, partimos de un sistema en que el usuario selecciona el objeto de interés. Si ese no fuera el caso, detectar automáticamente objetos de interés presenta varios desafíos. Entre esos desafíos destacan los cambios de apariencia de los objetos, los cambios de iluminación (provocados artificialmente o naturalmente por el movimiento del sol). Una forma de lidiar con los cambios del campo de visión es trabajar con imágenes base del entorno, entonces al tener una imagen de referencia se puede realizar una resta de imágenes para obtener posibles objetos en movimiento.

Para lograr la correspondencia de los objetos en las distintas vistas se usa una simplificación de nivel. Esta simplificación consiste en que los objetos que se presentan en el campo visual de las cámaras se encuentran sobre el mismo plano. Este supuesto permite la correspondencia de objetos entre las diferentes vistas como una simple transformación planar. Una vez que los puntos clave del objeto en cada cámara son localizados es posible inferir su localización 3D, dado que se cuenta con la calibración correspondiente a las cámaras. La calibración de las cámaras define un modelo geométrico entre el plano 2D y el sistema de coordenadas 3D. Como una consecuencia de este

supuesto el sistema debe permanecer invariable, un leve movimiento de una de las cámaras requeriría una nueva calibración. [1]

3 Método propuesto

Para graficar el método propuesto se presenta un diagrama de flujo mencionando los componentes principales de procesamiento, el cual se encuentra a continuación:

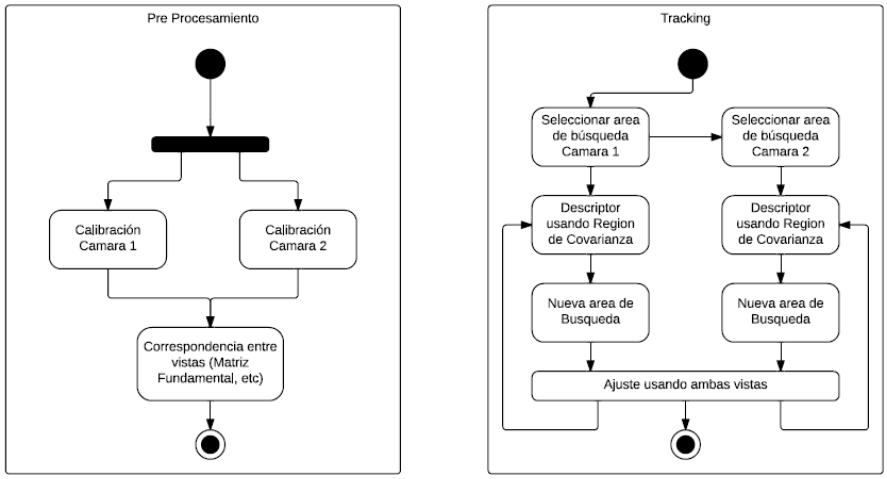

Figura 1

3.1 Calibración de las cámaras

Para extraer la calibración de las cámaras se usa un modelo que transforma las coordenadas 2D en coordenadas 3D. El modelo de calibración consiste en utilizar los parámetros intrínsecos y extrínsecos. Los parámetros intrínsecos son los que caracterizan a la cámara, estos son el punto principal, dimensión de pixeles, distancia focal y distorsión del lente. Por otro lado los parámetros extrínsecos describen la posición de la cámara y la orientación con respecto al sistema de coordenadas de la escena.

Para la obtención de estos parámetros se utilizo el toolbox de calibración para matlab de Bouguet. [2]

Se obtuvieron 25 imágenes de prueba para cada cámara, sobre las cuales se ejecuto el método de Bouguet para la calibración de cada cámara. Para esta investigación se usaron fotos tomadas en el Hall universitario del Campus San Joaquín de la Pontificia Universidad

Católica de Chile, estas fotografías fueron proporcionadas por Christian Pieringer. La idea es construir la matriz fundamental a partir de los parámetros extrínsecos de cada cámara. Es importante notar que las imágenes están a color en dos tamaños posibles, en 640x480 px para la cámara 1 y 1024x768 px para la cámara 2.

Es importante destacar que la calibración se puede realizar mediante puntos correspondientes entre las dos cámaras. Con al menos 7 puntos se puede construir la matriz fundamental.

Los parámetros [Tx Ty Tz] corresponden a vector de traslación entre la cámara y el mundo real. Los parámetros [Rx Ry Rz] corresponden a los ángulos de rotación para la transformación entre la cámara y el mundo real. Estos parámetros se obtienen luego de calibrar cada cámara usando el toolbox para Matlab. Para calibrar la cámara se deben seleccionar las esquinas de un tablero que aparece en diferentes posiciones en cada foto. Luego se pueden calcular los parámetros extrínsecos usando una imagen en donde el tablero se encuentra en el piso. Es importante destacar que dado que el tablero esta en el piso se puede suponer que la altura es z=0 en ese punto. Además se debe asegurar que el origen corresponde al mismo punto en las dos vistas. Con estos parámetros se puede construir la matriz de proyección para cada cámara.

Luego de que se tienen los parámetros extrínsecos se puede calcular las respectivas matrices de proyección como $A = [R\ T^T]$. Una vez que tiene A y B se utilizó el comando Bmv_fundamental del toolbox balu para obtener Fc.

Todos estos pasos estaban pre-calculados (dado la realización de una tarea anterior) y fueron cargador al programa para ahorrar tiempo de ejecución.

3.2 Cargado de frames

El siguiente paso es cargar las imágenes con las que se va a realizar el seguimiento. En este caso se puede ingresar el path de la carpeta que contiene las imágenes o también se puede ingresar el nombre del archivo a ser cargado que contenga toda la secuencia de imágenes. En nuestro caso usamos la segunda opción y cargamos todas las imágenes a partir de un archivo *.mat.

De lo contrario se ejecutara un código que cargara las imágenes en memoria a partir del path especificado por el usuario. El archivo smLoad preguntara el path del directorio que tiene las imágenes y luego

preguntará la extensión de las imágenes. El resultado de la ejecución de esta función será que todas las imágenes que se encuentren en la carpeta seleccionada serán cargadas a la variable frames. Adicionalmente las imágenes será re escaladas al tamaño 640x480 px para facilitar y agilizar el trabajo con una gran cantidad de imágenes.

3.3 Selección de objeto a seguir

El siguiente paso es seleccionar en un bounding box el objeto a ser seguido por el algoritmo. Para esto llamamos al método de msStart. Este método muestra la primera imagen cargada para cada cámara y deja al usuario seleccionar con el mouse sobre la imagen el objeto que se desea seguir. Dado que se obtiene un primer bounding box se puede empezar a hacer el seguimiento con el método de seguimiento que será luego descrito.

Inicialmente se tenía que hacer un llamado por cada cámara. La idea es implementar un modificación del método que permita seleccionar el objeto en una de las vistas y mediante geometría epipolar obtener el bounding box del objeto en la otra vista sin necesidad del input del usuario.

Se implemento un método que calcula el punto 3D a partir de un punto 2D, de la matriz de proyección de la cámara y de una altura h que es definida al momento de llamar al método.

Dado que se tiene la siguiente relación entre la vista 2D y 3D $\lambda m_1 = AM$, se puede conocer la ubicación 3D dado un punto en 2D. Para poder resolver este sistema se asume una altura h a la que se encuentra el punto 2D. Con ese supuesto se obtiene un sistema de 3 incógnitas y 3 ecuaciones. Cabe notar que el punto m1 es de 3x1 (en su forma homogénea), A es de 3x4, M es de 4x1 y lambda es un escalar. Luego de plantear las ecuaciones y resolver por X y Y (las primeras dos componentes del vector M se obtiene el siguiente resultado:

$$\alpha = a8 * a2 + m1 * a12 * a6 - m1 * a10 * a8 - a6 * a4 - m1 * a10 * a7 * h$$

$$\beta = m1 * a11 * h * a6 - a2 * m2 * a12 + a7 * h * a2 - a6 * a3 * h + a4 * m2 * a10 - a2 * m2 * a11 * h + a3 * h * m2 * a10$$

$$\gamma = -m2 * a9 * a2 - a5 * a10 * m1 + a5 * a2 + m2 * a1 * a10 + a9 * a6 * m1 - a6 * a1$$

$$\theta = -m2 * a9 * a3 * h - m2 * a9 * a4 - a5 * m1 * a11 * h - a5 * m1 * a12 + a5 * a3 * h + a5 * a4 + m2 * a11 * h * a1$$

$$\mu = m2 * a12 * a1 + a7 * h * a9 * m1 - a7 * h * a1 + a8 * a9 * m1 - a8 * a1$$

$$\sigma = -m2 * a9 * a2 - a5 * a10 * m1 + a5 * a2 + m2 * a1 * a10 + a9 * a6 * m1 - a6 * a1$$

$$X = \frac{-(\alpha + \beta)}{\gamma}$$
$$Y = \frac{-(\theta + \mu)}{\sigma}$$

Considerando

$$m = [m1\ m2\ 1]^T$$
$$A = \begin{matrix} a1 & a2 & a3 & a4 \\ a5 & a6 & a7 & a8 \\ a9 & a10 & a11 & a12 \end{matrix}$$
$$M = [X\ Y\ h\ 1]^T$$

Luego que se obtiene el punto M se puede usar la matriz de proyección de la segunda cámara para obtener el punto correspondiente de la cámara 1 en la cámara 2. El siguiente paso es dibujar el bounding box con la información de los puntos correspondientes y comenzar el proceso de tracking en ambas cámaras.

3.4 Tracking Región de Covarianza

El método utilizado para realizar el tracking es el método de región de covarianza. El primer paso es construir un vector de características, en nuestro caso usamos un vector que incluye los valores de rojo, azul y verde de la región que está siendo seguida. El paso siguiente es extraer la covarianza de estos vectores (uno para el frame actual y otro para el frame siguiente).

Ahora nos interesa comparar estos vectores de covarianza, para eso se usa la descomposición SVD, dado que este descriptor de covarianza no es un elemento euclidiano se usa la métrica de matrices semi-positivas Log-eucliadianas que se define como:

$\rho(X, Y) = \log(X) - \log(Y)$ [3]

Donde log(X) es el mapa logarítmico de la matriz de covarianza, el cual es definido por la descomposición SVD de la matriz X. Así $SVD(X) = U\Sigma U^T$ es la descomposición SVD de X, donde U es una matriz ortonormal y $\Sigma = diag(\lambda 1, ..., \lambda n)$ es una matriz diagonal con los valores propios.

Entonces el mapa logarítmico queda definido como:

$\log(X) = U[diag(\log(\lambda 1), ..., \log(\lambda n))]U^T$ [3]

El último paso es recorrer el bounding box en donde estamos buscando el posible movimiento del objeto que está siendo seguido y seleccionar el bounding que tenga el menor rho, donde rho es el error cuadrático medio. Este paso de seleccionar el menor error es conocido como elegir el vecino más cercano en cuanto a los descriptores de los bounding boxes.

3.5 Tracking vista dual

El tracking de vista dual se realiza en dos etapas principales, ambas corresponden a una iteración o búsqueda del objeto en el frame siguiente. En la primera etapa, las cámaras utilizan el método de Región de covarianza explicado en la sección 3.4 de forma independiente retornando un nuevo bounding box correspondiente al objeto que se está siguiendo. En la segunda etapa, se mide el error medido basado en la distancia en pixeles de las imágenes utilizando geometría epipolar [4], este proceso se denomina ajuste de seguimiento y se describe con detalle en la sección 3.6.

3.6 Ajuste de seguimiento

En este paso detectamos si alguna de las cámaras se perdió de su objetivo. Para este paso se compara la posición 3D del centroide del bounding box que sigue a cada objeto. Si su distancia con respecto a la línea epipolar varia de forma drástica quiere decir que uno de los dos bounding boxes no está siguiendo bien al objetivo (o por lo menos quiere decir que uno de ellos realizo un movimiento brusco con respecto al otro). Dado que se está siguiendo al mismo objeto con las dos vistas, es de esperar que los movimientos sean más o menos coordinados. Así, si la distancia epipolar varia en un 5% o más se realiza un ajuste al seguimiento de una de las cámaras.

Para detectar que cámara debe ser ajustada con respecto a la otra, se compara el bounding box actual con uno seleccionado de un frame anterior y se toma el promedio de los valores de intensidad de ese bounding box y se compara con el frame anterior. La cámara que presenta la mayor diferencia es ajustada. De esta forma el método determina que bounding box se movió de tal forma que su contenido ya no contiene elementos similares a los bounding box anteriores.

El ajuste de la cámara seleccionada es realizado mediante un método similar al del tracking por covarianza, se selecciona un vector (en este caso el de valor promedio de la intensidad) y se compara con los del bounding box grande, para luego elegir el que presente el menor error.

4 Experimentos y Resultados

Los resultados y experimentos serán presentados en dos subseciones, la primera corresponde a los resultados del pre-procesamiento, el cual es independiente del objeto a seguir. Mientras que la segunda subsección corresponde a los experimentos y resultados del método de tracking usando dos vistas descrito en la sección 3.5.

4.1 Calibración de las cámaras

Calibrando cada cámara por separado usando el método de Bouguet y a partir de los parametros intrinsecos y extrinsecos se obtienen las siguientes matrices:

La matriz de proyección A de la cámara 1:

$$A = \begin{bmatrix} 0.0704 & 0.9916 & -0.1088 & 1.154E3 \\ 0.6139 & -0.1291 & -0.7788 & -1.1198E3 \\ -0.7863 & -0.0119 & -0.6178 & 9.3086E3 \end{bmatrix}$$

La matriz de proyección B de la cámara 2:

$$B = \begin{bmatrix} 0.0013 & -0.2749 & 0.9615 & 14.57E2 \\ -0.8691 & 0.4752 & 0.1317 & -11.94E2 \\ -0.4946 & -0.8358 & -0.2383 & 12.381E3 \end{bmatrix}$$

La matriz fundamental entre la cámara 1 y 2:

$$F = \begin{bmatrix} -0.2573 & -0.4438 & 0.6984 \\ -0.1828 & 0.4395 & 0.6417 \\ 0.0296 & 0.7787 & 0.0599 \end{bmatrix}$$

4.2 Tracking

Como métrica de desempeño se utiliza la cantidad de frames con seguimiento correcto del total de frames utilizados. Un ejemplo de seguimiento correcto se muestra a continuación:

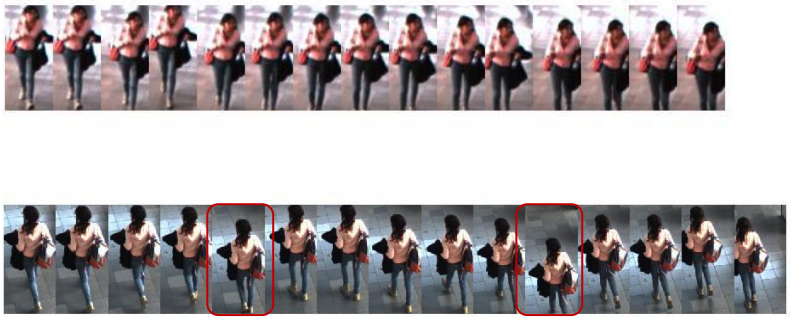

Figura 2

En la figura 2 se muestran los bounding box del seguimiento en cada cámara, cabe recalcar el ajuste, el cual se ve en la cámara 2 (abajo) en los frames marcados con rojo.

Para un set de 5 experimentos, los cuales incluyen personas con diferentes trayectorias rectas, e incluso cambiando su trayectoria en el mismo set. Con esto, se obtuvieron los siguientes resultados:

	Cantidad de frames con seguimiento correcto				
	Set 1	Set 2	Set 3	Set 4	Set 5
Camara 1	33	43	29	173	332
Camara 2	105	15	87	173	329
Frames Totales	133	51	90	173	332

Tabla 1

	Porcentaje de frames correctos				
	Set 1	Set 2	Set 3	Set 4	Set 5
Camara 1	25%	84%	32%	100%	100%
Camara 2	79%	29%	97%	100%	99%

Tabla 2

A continuación se presentan los graficos respectivos, reflejando el desempeño del algoritmo en los 5 set de entrenamientos.

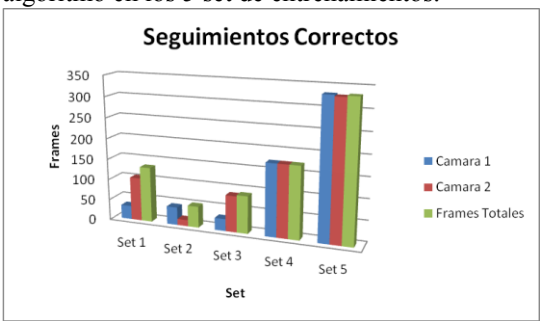

Grafico 1

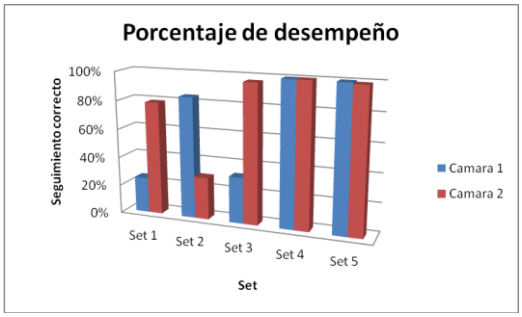

Grafico 2

5 Conclusiones

Los sistemas de seguimiento con una cámara han sido investigados por la comunidad de visión por computador y se han visto considerables avances en las métricas de seguimiento y rendimiento. Esta investigación se concentra específicamente en utilizar estos métodos y construir un seguimiento utilizando la información entregada por dos cámaras. Nuestro método no es capaz de ejecutar en tiempo real pero aun así revisa alrededor de 100 fotos en 200 segundos. Para esta investigación asumimos que las cámaras comparten una proporción importante del campo de visión. Cabe notar que la secuencia de imágenes entregadas no está en sincronía de tiempo, por lo que por medio de inspección se logro determinar el desfase temporal entre las imágenes. La cámara 1 esta 35 frames adelantados a la cámara 2.

En este informe se han descrito métodos de calibración, de estimación de homografía, de seguimiento mono-vista y seguimiento multi-vista.

En el método de calibración los resultados no fueron los esperados, en un comienzo se realizo la calibración y no se tenían los orígenes de los parámetros extrínsecos en el mismo punto 3D. Luego de que se resolvió ese problema los ejes X e Y estaban rotados. Para solucionar este problema se aplico una transformación a una

de las matrices para así tener los ejes del espacio 3D en el mismo punto y en la misma orientación.

Una vez que se obtuvo las matrices de proyección usando el toolbox de Bouguet, se construyo la matriz fundamental. Acá también nos encontramos frente a problemas ya que algunas líneas epipolares pasan cerca del punto que correspondiente y otras definitivamente no. Se intento ajustar la matriz mediante fuerza bruta con el fin de lograr mejores resultados pero no se logro lo esperado. Finalmente se decidió ajustar el resto del código a este error. Esto quiere decir que si los objetos que están siendo seguidos presentan el mismo error el seguimiento va bien, de lo contrario una de las vistas presenta un movimiento inesperado con respecto al objeto que se está siguiendo. En futuras investigaciones quizás sea mejor idea intentar encontrar la homografía entre las dos vistas, como forma de reducir esta ambigüedad que se presenta con en análisis epipolar.

Este sistema no resuelve el caso de oclusiones complejas. Si ocurre que se cruza el objeto que estamos siguiendo con otro, el bounding box se ajusta para mantener la selección inicial.

Para el seguimiento, inicialmente se comenzó a trabajar con el background substraction. Lo cual funcionaba en algunos casos, pero en muchos otros no. Esto se debe a que este método no es robusto frente a los cambios de iluminación. Por otro lado si la persona deja de moverse o se mueve más lento se pierde el seguimiento. Otro imprevisto que presenta este método es que cuando se cruzan dos objetos el seguimiento tiende a cambiarse de objeto.

El siguiente método intentado fue el seguimiento por Lucas Kanade, el cual funciona bien con el método de corrección que fue implementado pero fue abandonado ya que el seguimiento por covarianza presento mejores resultados.

La razón por la cual el seguimiento por covarianza funciona mejor es que se asigna un peso a cada matriz de covarianza entonces todas las ambigüedades son ponderadas de tal forma que se selecciona el bounding box que tiene el menor valor de ambigüedad.

Luego de concluido esta investigación podemos concluir que se podrían obtener mejores resultados si se realiza la calibración de las cámaras usando puntos correspondientes de los puntos relevantes que se presentan en el campo compartido de visión en el momento que no hay objetos (cuando se toma el background).

Bibliografía

[1] J.-Y. Bouguet, «Camera Calibration Toolbox for Matlab,» 9 Julio 2012. [En línea]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html. [Último acceso: 10 12 2012].

[2] J. Black, «Mutli View Image Surveillance and Tracking,» City University, London, 2004.

[3] D. M. L. S. Pedro Cortez, «Object tracking based on Covariance Descriptors and on-line Naive Bayes Nearest Neighbor Classifier,» Pontificia Universidad Catolica de Chile, Santiago, 2010.

[4] «Epipolar geometry,» [En línea]. Available: http://en.wikipedia.org/wiki/Epipolar_geometry.