CHAPTER X

Robust Real Time Stabilization: Estabilización de la Imagen con Aplicación en el Robot Humanoide HOAP-3

A.P MATEO[1], M.GONZÁLEZ-FIERRO[1], D.HERNÁNDEZ GARCÍA[1], P.PIERRO[1] y C. BALAGUER[1]

[1]Robotics Lab, Universidad Carlos III de Madrid.

En este artículo presentamos un nuevo algoritmo para estabilizar las imágenes que se producen en una cámara en movimiento. El algoritmo RRTS detecta las variaciones que se dan en la secuencia de imágenes y las corrige, de forma que se obtiene una secuencia suave y estabilizada. Se ha implementado en tiempo real en el robot humanoide HOAP-3 mientras camina, obteniéndose imágenes estables que pueden ser utilizadas para detectar objetos mientras el robot está en movimiento o puede mejorar la experiencia visual de un teleoperador cuando recibe la imagen del robot.

1 Introducción

En los últimos años se ha dado un fuerte impulso a la investigación en robots humanoides, desarrollándose robots con alta capacidad de manipulación, interacción y movilidad. Algunos ejemplos son el WABIAN-2 de la Universidad de Waseda (Ogura et al., 2006), el ASIMO de Honda (Sakagami et al., 2002), el HRP-2 del National Institute of Advanced Industrial Science and Technology of Japan (Kaneko et al., 2004) o el RH-1 diseñado en la Universidad Carlos III de Madrid (Arbulú, Kaynov, Cabas, y Balaguer, 2009). Pero para que estos robots interaccionen con el ser humano y se introduzcan plenamente en su entorno, es necesario aumentar su seguridad, manipulabilidad, estabilidad y su capacidad de interacción.

Para llegar a este nivel de adaptación de los robots al mundo humano, se requiere una sustancial mejora de los sistemas de visión artificial, tanto desde el punto de vista del hardware como de algoritmos, avanzando en aspectos como localización e identificación de objetos, nuevas tecnologías de video, detección de personas o estabilización de la imagen.

Para los robots humanoides, debido a las inclinaciones laterales que sufren al andar, la estabilización de la imagen es un elemento fundamental para realizar muchas tareas complejas que impliquen movilidad. Con este sistema se aumenta la capacidad de interacción con el entorno; si el robot se encuentra en un escenario desestructurado, la estabilización de la imagen permite detectar objetos con el robot el movimiento que de otra forma se perderían de vista continuamente. Así mismo, puede mejorar la manipulabilidad, con un sistema de estabilización de la imagen se puede fácilmente agarrar un objeto mientras el robot camina. Puede aumentar la capacidad de reacción ante imprevistos, especialmente en entornos dinámicos. También permite mejorar la experiencia visual de un observador que esté teleoperando al robot mediante algún HRI. Es, en definitiva, un sistema necesario y beneficioso para robot humanoide, sin el cual está ciego cuando realiza alguna tarea en movimiento.

Actualmente el uso más extendido de la estabilización de la imagen podemos encontrarlo en cámaras fotográficas o de video digitales y telescopios. Existen varios métodos de estabilización que comercializan las grandes multinacionales fotográficas, a continuación se hablará brevemente de ellos. En primer lugar está el estabilizador de imagen óptico (Cardani, 2006). Se trata de un sistema mecánico compuesto por unas lentes móviles. Si se detecta un movimiento en la cámara, las lentes se mueven de forma que la imagen es estabilizada. Algunos sistemas comerciales de estabilización óptica aparecen en las cámaras Canon IS, Nikon VR y Panasonic Lumix (y Leica) Mega OIS. El segundo método, el estabilizador de imagen electrónico, (Oshima et al., 1989), (Kinugasa et al., 1990), compensa la secuencia de imágenes mediante el uso de sensores de movimiento. Algunos ejemplos de uso de esta tecnología se pueden encontrar en las cámaras Sony Alpha, Olympus y Casio Exilim. Tanto el estabilizador óptico como el electrónico requieren de hardware adicional para obtener la estabilización, sin embargo, el método de estabilización digital de la imagen (Xu & Lin (2006)), (Pan & Ngo (2005)), no necesita ningún aparato electrónico o mecánico ya que compensa el movimiento de la imagen mediante algorit-

mos de visión. El esquema típico de esta técnica de estabilización se puede observar en la Figura 1.

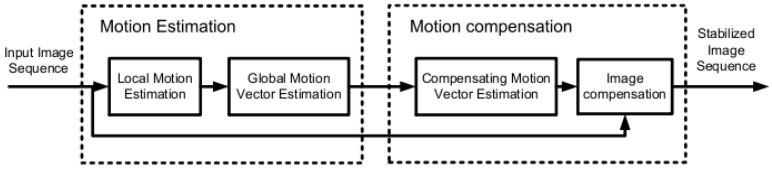


Figura 1: Esquema del estabilizador de imagen digital.

Respecto a la utilización de sistemas de estabilización de la imagen aplicados a la robótica, existen varios casos en la literatura. En (Amanatiadis et al, 2010) un sistema de estabilización tanto electrónico como digital, diseñado para implementarlo en un rover, en (Hayakashi, 2005) se presenta un sistema de control de la imagen para interfaces en robots de rescate, en (Choi et al., 2010) se presenta un sistema de detección de perturbaciones mediante inclinómetros que posteriormente se trata con el filtro de Kalman Extendido.

En este artículo presentamos un nuevo algoritmo llamado Robust Real Time Stabilization (RRTS), con él se consigue controlar y corregir las perturbaciones producidas en la imagen obtenida por una cámara en movimiento. Para probar el comportamiento del algoritmo, se ha implementado en el robot humanoide HOAP-3 mientras camina. En un momento inicial se ha grabado la imagen de la cámara del robot mientras andaba y se ha pasado el algoritmo a este video en un ordenador. Finalmente, se ha implementado en el robot directamente consiguiendo la estabilización de la imagen en tiempo real.

El documento está estructurado de la forma siguiente, en el apartado 2 se habla de la plataforma utilizada, en el apartado 3 se explica el algoritmo utilizado, en el apartado 4 se habla de la comunicación con el robot, en el apartado 5 se muestran los resultados experimentales y por último se presentan las conclusiones.

2 Plataforma utilizada

La plataforma usada para la implementación del algoritmo RRTS es el robot humanoide HOAP-3. Se trata de un humanoide de tamaño y peso medio, 60 cm y 9 kg aproximadamente, diseñado y fabricado por Fujitsu.

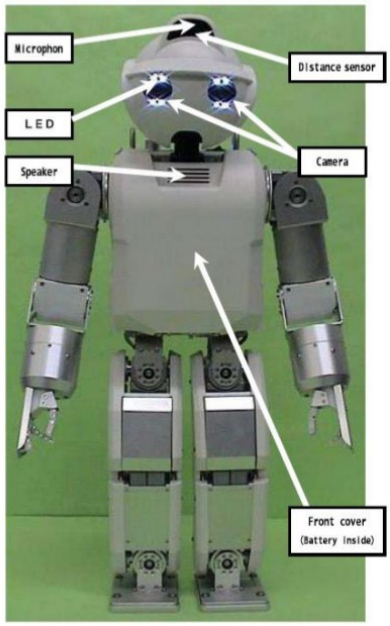


Figura 2: Sensores disponibles en el HOAP-3

El robot HOAP-3 posee 28 grados de libertad lo que le dota de gran capacidad de movimiento. Posee 6 grados de libertad en cada pierna, 6 en cada brazo, 3 en la cabeza y 1 en la cadera. Estos motores disponen de encoders relativos y existe la posibilidad de controlarlos tanto en posición como en velocidad.

El robot lleva incorporado un PC-104 embebido en su interior, exactamente en la parte de atrás. Se trata de un Pentium de 1.1 Ghz con 512 Mb de RAM y una memoria Compact Flash de 1 Gb. Posee conexión Wifi IEEE802.11g y 4 puertos USB.

El sistema operativo que incorpora el robot es un Linux en tiempo real, cuya base es un Fedora Core 1 con el kernel 2.4. Este sistema operativo en tiempo real permite un mayor dominio de la plataforma.

Para completar las funcionalidades de este humanoide se han añadido un conjunto de sensores que le permiten desenvolverse con naturalidad en cualquier entorno. Como se puede observar en la Figura 2, el robot dispone de dos cámaras para visión estereoscópica, un micrófono, un altavoz, sensores infrarrojos de distancia, y sensores de fuerza en pies y manos.

3 Estabilización de la imagen

La estabilización de las imágenes se va a realizar en dos tipos de movimientos diferentes: rotación, teniendo en cuenta el parámetro θ y traslación, analizando los ejes X e Y. En algunas partes del algoritmo, se han utilizado algunas funciones de las librerías OpenCV.

3.1 Estructura general del algoritmo

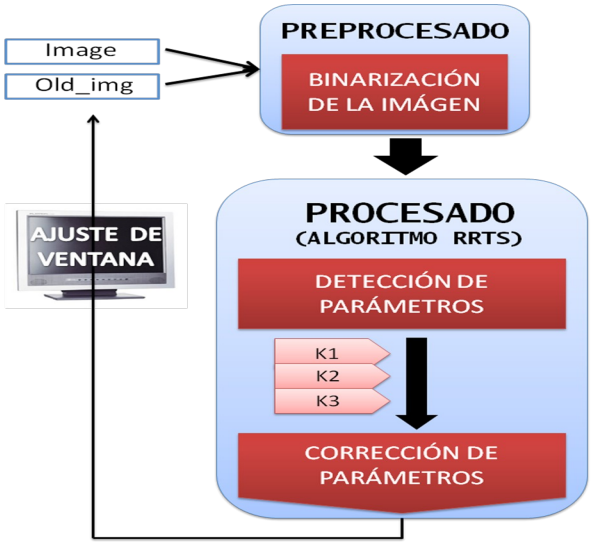


Figura 3: Esquema general de RRTS

En el diagrama de bloques de la Figura 3 se muestra el esquema general de la estabilización. Este consta de dos partes fundamentales:

- Preprocesado
- Procesado

En la parte de preprocesado se realiza una binarización de la imagen a tratar para poder llevar a cabo el algoritmo principal.

En el procesado es donde se realiza la estabilización de la imagen. Ésta a su vez se puede dividir en dos etapas fundamentales, según se puede ver en el diagrama, detección y corrección de parámetros.

Entre estas dos etapas se introducen un conjunto de valores de ajuste (K1, K2, K3), que corresponden a tres reguladores proporcionales, el objetivo fundamental es mejorar el comportamiento dinámico del sistema de forma que se adecúe a las a las especificaciones deseadas para su funcionamiento. En la Tabla 1 se observa las respuestas que puede tomar la estabilización en función de los valores de los reguladores.

VALOR K	AMORTIGUACIÓN	DESCRIPCIÓN DEL SISTEMA
0<K<1	Sistema subamortiguado	Controla el sistema y obtiene una imagen estable.
K=1	Críticamente amortiguado	Sistema críticamente controlado
K>1	Sistema sobreamortiguado	Sistema inestable

Tabla 1: Valores de K

Si los reguladores toman valores por debajo de 1, se obtiene un sistema subamortiguado, es decir, la imagen se corresponde con un sistema estable. Se darán valores por debajo de 1 cuando la estabilización se realice en el movimiento de traslación (ya sea en X o en Y), esto hará que la secuencia de imágenes fluya de manera progresiva y suave.

Cuando el valor es 1, el sistema se encuentra críticamente amortiguado, es decir, ante cualquier perturbación puede producirse una inestabilidad. Se dará valor 1 al regulador usado en la estabilización del ángulo, porque cualquier pequeña variación por debajo de la unidad produce errores significativos en la salida del sistema.

Si por el contrario, toman valores por encima de la unidad, la estabilización se produce de manera descontrolada, y el comportamiento dinámico del sistema de acuerdo a las especificaciones establecidas no es el deseado.

En el procesado hay dos partes, en la primera se lleva a cabo la identificación del desplazamiento y giro que se producen entre una imagen y la siguiente (Figura 4), para posteriormente corregirlos. Para ello se analizan las imágenes inicial y la que llega a continuación.

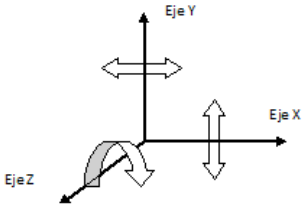


Figura 4: Variaciones que puede sufrir la imagen con respecto a los ejes de coordenadas

Tomando un primer frame de referencia estable, donde el robot tiene que encontrarse en posición erguida y parado, se compara el siguiente frame con el anterior (imagen inicial), la diferencia en los ejes X, Y y el ángulo de rotación θ entre las dos imágenes, serán los parámetros que se buscan.

Tras la identificación de los datos necesarios, se realiza la segunda parte del procesado, la corrección del desplazamiento y giro, es decir, una transformación afín genérica en 2D (Figura 5).

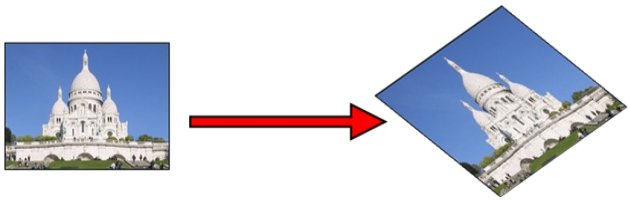


Figura 5: Transformación afín genérica.

Esta corrección solo se producirá cuando las condiciones iniciales sean las adecuadas y no en todos los casos, pues solo se busca que el siste-

ma estabilice cuanto el robot se encuentra en movimiento y no si se producen cambios muy grandes de ángulo, como podría ser el caso de que el humanoide se cayera.

Una vez realizada la estabilización, se saca por pantalla la imagen estabilizada. Al realizar las trasformaciones aparecerá un borde negro, que se produce a causa de píxeles no definidos, que toman valor 0 en el proceso de estabilización. Estos bordes serán eliminados recortando la imagen y que posteriormente será ampliada hasta que tenga el mismo tamaño que la imagen sin estabilizar. Esto permitirá una visualización del video mucho más cómoda obteniendo un resultado muy parecido al que vería un ser humano al caminar.

Estos pasos se irán produciendo a lo largo de todas las imágenes formando un ciclo, tomando como referencia siempre la imagen anterior, que ya estará estabilizada. Con el algoritmo presentado se consigue una imagen estable con el humanoide en movimiento.

3.2 Procesado: El algoritmo RRTS

Después de la binarización realizada en la parte del preprocesado, se pasa a la parte de procesado, compuesto por dos grandes bloques: la detección de parámetros de estabilización (X,Y,θ) y la corrección de estos para obtener la imagen estabilizada. A continuación se explicará detalladamente cada uno de ellos.

La detección de parámetros de estabilización está basada en el algoritmo SURF (Speeded Up Robust Teatures) (Bay et al., 2006) el cual es una adaptación eficiente y rápida del algoritmo SIFT (Lowe, 1999). El algoritmo SURF consta de tres partes, un detector de puntos característicos, un descriptor que asocia con vectores esos puntos, para definir el contenido de la imagen y finalmente, se realiza la correspondencia de puntos entre una imagen y la siguiente.

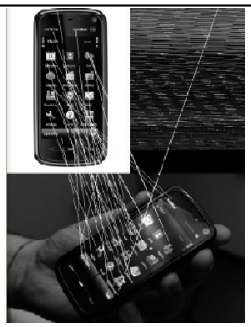


Figura 6: Aplicación del algoritmo SURF

Este algoritmo se suele usar en visión por computador en tareas como reconocimiento o identificación de objetos. En la parte superior de la Figura 6 se puede observar la imagen del objeto que se desea detectar, detrás de esta, vemos una imagen general que contiene a ese objeto. Los puntos característicos en una imagen y otra serán detectados, se hará la correspondencia entre ellas y por último se unirán mediante líneas.

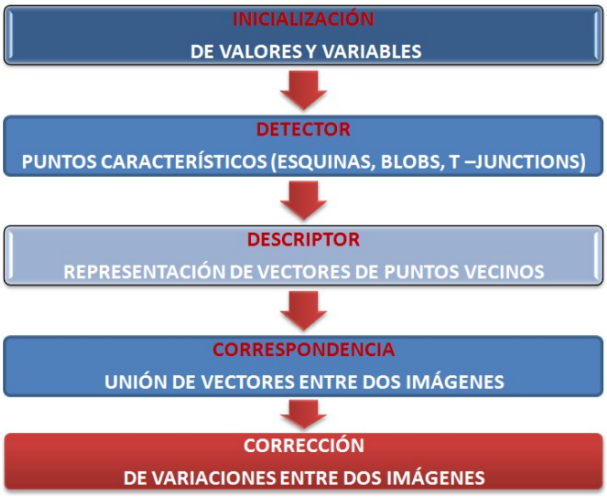


Figura 7: Diagrama del algoritmo RRTS

El nuevo algoritmo RRTS (Robust Real Time Stabilization) se basa en SURF para estabilizar una imagen en tiempo real. En este algoritmo no sólo se va a realizar una identificación, sino que también se calculan las variaciones en el eje X, Y y el ángulo θ entre dos imágenes, y por último se hace una traslación afín genérica. En la Figura 7 se observan las fases del algoritmo que a continuación se explicarán.

3.2.1 Inicialización

En la inicialización se definen las funciones y variables que van a ir formando parte del algoritmo:

- Este algoritmo comienza con la selección del objeto que se desea identificar en la siguiente imagen. Si tenemos una imagen inicial se hará un recuadro centrado de aproximadamente el 80% de la imagen. Se hace así para que a la hora de analizar la imagen el objeto sea recuadrado con vectores que nos servirán posteriormente para analizar y corregir el giro detectado.
- El centro de la imagen será el punto de intersección entre las líneas que definen el ancho de la imagen entre 2 y el alto de la imagen entre 2. Este coincidirá como centro de rotación.
- Se definen las matrices de rotación y traslación para poder realizar la traslación afín.
- Se inicializan los valores de los reguladores proporcionales, K1, K2 y K3 con un valor de 1, 0.5 y 0.7 respectivamente. Estos valores se han estimado de forma heurística de acuerdo a la situación de la cámara, ángulo de giro del robot al andar y factores del terreno. Para la aplicación del algoritmo con un movimiento diferente a caminar sería necesario reajustar estos valores.
- Para que la imagen estabilice, el ángulo de giro se supone no mayor de 30°. Cuando se produzcan giros muy elevados, como por ejemplo una caída del humanoide, no interesa estabilizar la imagen.

3.2.2 Detector

Una vez seleccionada la región de interés, la cual se obtiene maximizando el criterio de máximo de esquina (Figura 8), se buscan los puntos característicos del rectángulo y se analizan puntos vecinos mediante la aproximación de cajas de filtrado y se toman vectores.

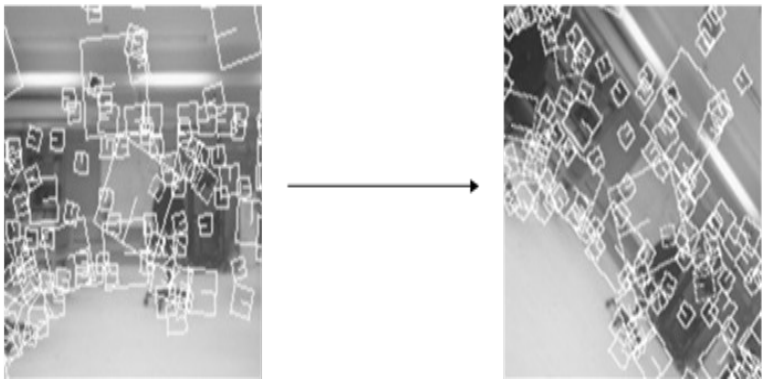


Figura 8: Identificación de puntos característicos mediante búsqueda de máximos de esquina locales

Los puntos característicos encontrados serán esquinas, blobs y T-junctions. Es necesario usar escenarios donde al menos se encuentren 2 puntos característicos, lo que sería suficiente para dibujar el rectángulo de la región.

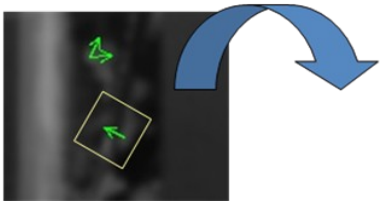
3.2.3 Descriptor

El descriptor se utiliza para caracterizar la región de interés de la imagen que rodea a los puntos característicos. Estos descriptores son invariantes ante traslación, rotación y cambios en la resolución de la imagen. RRTS se basa en SURF para calcular los descriptores, este algoritmo es computacionalmente rápido y da la posibilidad de implementarlo en tiempo real en el robot.

Para obtener el descriptor el primer paso es calcular la escala, esta es el resultado del determinante de la matriz Hessiana (se utiliza esta puesto que se reduce significativamente el tiempo de computación). Luego se calcula la orientación del punto de interés, para posteriormente calcular el descriptor SURF.

Para obtener un punto invariante a la orientación del punto se calcula el “Haar-wavelet” en todas las direcciones, esto se calcula para todos sus puntos vecinos y posteriormente se realiza la suma de todos los resultados.

Para calcular el descriptor se construye una región cuadrada, esta se divide en 4 subregiones, se buscan puntos característicos en cada uno de ellos, se aplica "Haar-wavelet" y se suavizan los resultados con una Gaussiana, en cada subregión se suman los resultados y se calcula su valor absoluto, esos valores serán los elementos del vector que proporciona cada subregión. La unión de los vectores de cada subregión forma el descriptor. (Figura 9)



$$\vec{v} = \begin{pmatrix} v_1 \\ v_2 \\ \dots \\ v_{128} \end{pmatrix}$$

Figura 9: Vector que proporciona una subregión y su valor

3.2.4 Correspondencia

Se puede definir una correspondencia como una transformación geométrica que lleva un punto de interés de la imagen prototipo a la imagen de prueba. Siendo el objetivo encontrar un gran grupo de correspondencias que compartan la misma transformación. En este caso tenemos dos tipos de correspondencias, traslación (Figura 10) y rotación (Figura 11):

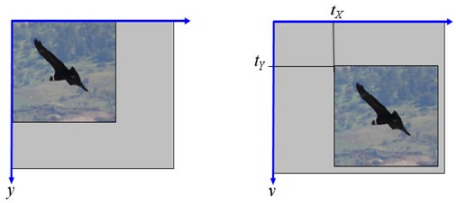


Figura 10: Traslación

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_X \\ t_Y \end{pmatrix} \quad (1)$$

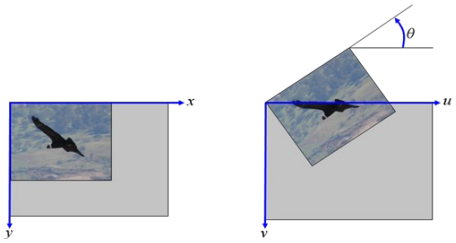


Figura 11: Rotación

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{2}$$

Uniendo estas dos correspondencias, obtenemos una imagen con translación afín genérica (Figura 12). Obteniendo gráficamente el siguiente resultado:

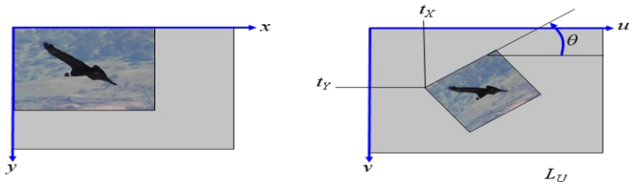


Figura 12: Traslación afín genérica

$$\begin{pmatrix} u \\ v \end{pmatrix} = e\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_X \\ t_Y \end{pmatrix} \tag{3}$$

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} e\cos\theta & e\sin\theta & t_X \\ -e\sin\theta & e\cos\theta & t_Y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{4}$$

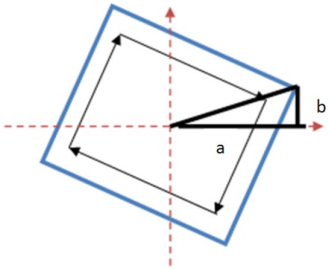


Figura 13: Cálculo del ángulo de una imagen rotada

El ángulo de rotación será calculado, según la Figura 13 como:

$$\theta = tg^{-1}\frac{p.y - centro.y}{p.x - centro.x} = tg^{-1}\frac{b}{a} \tag{5}$$

donde p.x y p.y es la distancia X e Y del punto desplazado.

3.2.5 Corrección

La última parte del algoritmo RRTS será la inversión de los valores obtenidos, mediante una trasformación afín genérica en 2D. En esta parte la variación de valores en X, Y y θ serán corregidas hasta que los ejes de la imagen siguiente coincidan con los ejes de la imagen anterior. Los valores a compensar serán θ, X=a y Y=b (Figura 13).

Con el algoritmo presentado, la estabilización de la imagen cuando el robot humanoide camina se produce perfectamente en la mayoría de los casos. Mediante la implementación del algoritmo en vídeos grabados (off-fline) y posteriormente en tiempo real, se ha podido realizar un análisis sobre los casos en los que el resultado es el esperado.

4 Comunicación con el robot

El algoritmo RRTS para la estabilización de la imagen descrito anteriormente fue implementado en el robot humanoide HOAP-3 para estabilizar

la imagen percibida por un operador remoto en la interfaz de teleoperacion, mejorando la comodidad y el desempeño del teleoperador del robot.

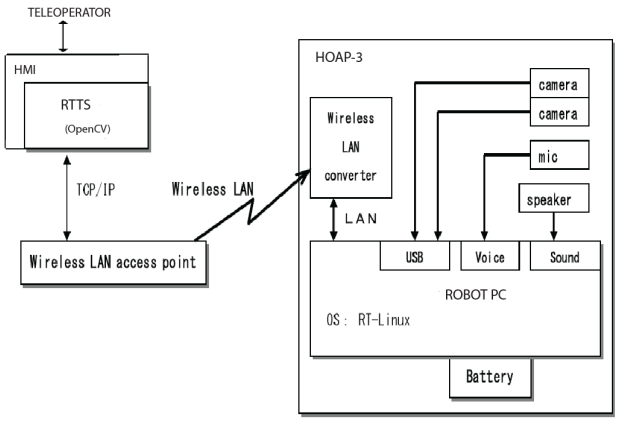


Figura 14. Sistema de teleoperacion del robot HOAP-3

La Figura 14 muestra la muestra la configuración del sistema de teleoperacion para el robot HOAP-3. La comunicación entre y el robot y la interfaz de control del teleoperador se realiza mediante de enlaces de TCP/IP a travez de una red inalámbrica independiente.

La imagen capturada de las cameras del robot es previamente procesada mediante el algoritmo RRTS antes de su visualización por el teleoperador. El sistema de estabilización de la imagen recibe del servidor del HOAP-3 imágenes de la cámara del robot, a un ratio de 6-10 fps aporx., en formato YUV240p del cual es transformado al formato de imagen IplImage de openCV para su adecuado procesamiento mediante el algoritmo RRTS. Finalmente una la imagen estabilizada resultante se presenta al teleoperador.

5 Resultados experimentales

En este aparatado se muestran y explican los resultados de la estabilización obtenidos en el humanoide. Para ello primero se ha hecho un estudio offline. Esto quiere decir que se ha ejecutado el algoritmo en un video previamente grabado, con el fin de disponer de material con el que trabajar. Posteriormente, como se detalla en el siguiente apartado se implementa en tiempo real.

5.1 Discusión de resultados en la implementación offline

A continuación, se muestran tres figuras en las que aparece la imagen original y el resultado de la aplicación del algoritmo. En la Figura 15 se puede apreciar que el algoritmo ha realizado una corrección en la rotación, es decir, se compensa el ángulo de giro sufrido en el eje z.

Figura 15: Estabilización en la rotación

En el siguiente caso, no solo se produce una compensación el movimiento de rotación, sino que también se puede observar una corrección en el eje y. El algoritmo ha sido implementado para que también compense el movimiento ascendente/descendente de la imagen, como puede verse en la Figura 16.

Figura 16: Estabilización en rotación y traslación en y

Finalmente, tal y como se muestra en la Figura 17 se compensan los dos tipos de movimientos: rotación en Z y traslación en X e Y.

Figura 17: Estabilización en la rotación y traslación en x e y

Los resultados obtenidos en la mayoría de los casos son los esperados. Hay una mejoría notable comparando la imagen original y la estabilizada.

5.2 Discusión de resultados en la implementación online

En este apartado, se exponen los resultados obtenidos tras la implementación online del algoritmo en el humanoide. Como se puede ver a continuación se han capturado tres imágenes donde se pueden apreciar los resultados obtenidos. En el primer caso se realiza una compensación en la rotación Figura 18, en el segundo no solo se compensa la rotación, también la traslación en el eje y Figura 19 , en el último caso se corrige el movimiento en los tres ejes Figura 20.

Figura 18: Estabilización en la rotación

Figura 19: Estabilización en la rotación y en y

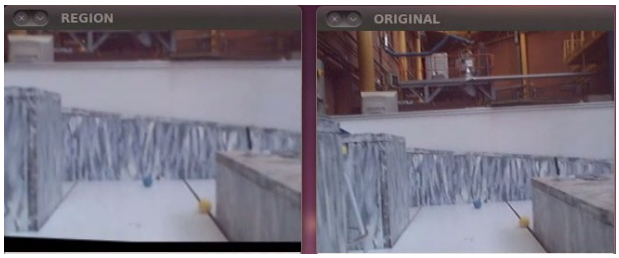


Figura 20: Estabilización en la rotación y traslación en x e y

Tras la evaluación de las imágenes obtenidas en tiempo real, el resultado es satisfactorio pero menos preciso que en el caso de la implementación a partir de un video grabado. Esto se debe a que en tiempo real hay que asegurarse de que los primeros fotogramas que se tomen sean estáticos, ya que si no corrige con respecto a una primera imagen desestabilizada.

Otro problema presentado es el hecho de que al trabajar con un mismo video no es necesario el ajuste de parámetros del algoritmo, sin embargo en el desarrollo en tiempo real, las condiciones en el entorno (estado del robot, iluminación, terreno, etc.) no siempre son las mismas. El único inconveniente posible que se puede presentar es que se trabaje en un entorno que no posea puntos característicos suficientes para el análisis de la imagen por parte del algoritmo.

Con todo esto la estabilización obtenida resulta altamente satisfactoria, ya que se consigue estabilizar movimientos muy bruscos. Siendo el método teórico estudiado el adecuado en todos los casos que se nos han presentado.

6 Conclusiones

El algoritmo de estabilización de imagen propuesto, RRTS, permite estabilizar una cámara en movimiento mediante la corrección de los saltos bruscos en la imagen. Este algoritmo permite estabilizar una imagen incluso ante perturbaciones producidas por amplios movimientos de traslación y rotación, como por ejemplo en el caso de la caminata de un humanoide. Se presenta también un sencillo sistema de control de la estabilización mediante un conjunto de parámetros obtenidos heurísticamente que permiten obtener una secuencia de imágenes suave y nítida. Este algoritmo se ha implementado en el robot humanoide HOAP-3 mientras camina, consiguiéndose estabilizar la imagen de la cámara del robot. En el futuro trataremos de detectar objetos mientras el robot está en movimiento.

Referencias

A. Amanatiadis, A. Gasteratos, S. Papadakis and V. Kaburlasos. "Image Stabilization in Active Robot Vision" in Robot Vision, edited by: Ales Ude. Vukovar, Croatia, 2010.

M. Arbulu, D. Kaynov, L. M. Cabas, and C. Balaguer. The RH-1 full-size humanoid robot: design, walking pattern generation and control. Journal of Applied Bionics and Biomechanics, 6(3):301–344, 2009.

H. Bay, T. Tuytelaars and Luc Van Gool. SURF: Speeded Up Robust Features. In book "Computer Vision - ECCV 2006". Lecture Notes in Computer Science. Springer Berlin / Heidelberg. 2006.

Cardani, B. (2006). Optical image stabilization for digital cameras, IEEE Control Syst. Mag. 26(2): 21–22.

Hayashi, K.; Yokokohji, Y.; Yoshikawa, T.; Tele-existence Vision System with Image Stabilization for Rescue Robots. Proceedings of IEEE International Conference on Robotics and Automation. 50 – 55. 2005

K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, and T. Isozumi. Humanoid robot HRP-2. Proceedings of IEEE International Conference on Robotics and Automation, pp. 1083–1090. 2004.

Kinugasa, T., Yamamoto, N., Komatsu, H., Takase, S. & Imaide, T. Electronic image stabilizer for video camera use, IEEE Trans. Consum. Electron. 36(3): 520–525. 1990.

D. G. Lowe. Distinctive image features from scale-invariant key-points. International Journal of Computer Vision, 2:91–110, 2004.

Y. Ogura, H. Aikawa, H. Kondo, A. Morishima, H. Lim, and A. Takanishi. Development of a new humanoid robot WABIAN-2. International Conference on Proceedings of IEEE Robotics and Automation, pp 76–81. 2006.

Oshima, M., Hayashi, T., Fujioka, S., Inaji, T., Mitani, H., Kajino, J., Ikeda, K. & Komoda, K. VHS camcorder with electronic image stabilizer, IEEE Trans. Consum. Electron. 35(4): 749–758. 1989.

Pan, Z. & Ngo, C. Selective object stabilization for home video consumers, IEEE Trans. Consum. Electron. 51(4): 1074–1084. 2005.

Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura. The intelligent Asimo: System overview and integration. IEEE/RSJ international conference on intelligent robots and systems, pages 2478–2483. 2002.

Xu, L. & Lin, X. Digital image stabilization based on circular block matching, IEEE Trans. Consum. Electron. 52(2): 566–574. 2006.