Método para detección y seguimiento de objetos con aplicaciones en Realidad Aumentada

Christian Nicolás Pfarher

Punto de control N° 2:

Adquisición de imágenes (prototipos) para realizar las pruebas

Métodos de extracción de características en imágenes parte 1 (Análisis/estudio del método).

Director
Dr. Enrique Marcelo Albornoz

Codirector
Dr. César Martínez

29 de junio de 2012

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

1. Adquisición de imágenes (prototipos) para realizar las pruebas.

El proceso de adquisición de imágenes es aplicado en tres diferentes instancias, según la situación:

1. **Imagen patrón**: es la adquisición de la imagen que será usada como patrón para su posterior detección y seguimiento y sobre la cuál se sobre impondrá el **Objeto de Realidad Aumentada**.

   La imagen, será adquirida con la misma cámara web utilizada para la captura de la **Imagen Live**, aunque no se descarta la posibilidad del uso de otro dispositivo que permita obtener una imagen de características similares a la obtenida con la cámara web utilizada.

   La adquisición de la imagen objeto, deberá cumplir ciertas restricciones prácticas, entre las cuales se puede nombrar: el tamaño de la imagen, las condiciones de iluminación, el nivel de detalle (cantidad de detalles mínimos), entre otras.

2. **Imagen Live o Flujo de Video**: es la obtención de un fotograma del Flujo de Video adquirido en tiempo real con la cámara web, con las mismas restricciones prácticas mencionadas anteriormente. Sobre este fotograma, se detectará la **Imagen Patrón**.

   La adquisición se realizará con una cámara web con una resolución de 640x480 píxeles de una computadora portátil Toshiba Satellite A505-S6803.

3. **Objeto de Realidad Aumentada**: es la definición de una imagen (ej: foto, tapa de libro, revista, etc.) o volumen tridimensional (ej: objeto 3D dibujado con OpenGL) - dependiendo del grado de avance que se logre en el proyecto - que será sobre impuesto en el **Flujo de Video**.

2. Métodos de extracción de características en imágenes parte 1 (Análisis/estudio del método).

En visión computacional, el concepto de puntos de interés, puntos claves (Keypoints) o puntos característicos (Feature Points) es usado ampliamente en diversidad de tareas tales como: el reconocimiento de objetos, la identificación de imágenes, seguimiento de objetivos, reconstrucción de escenas 3D, etc. La idea consiste en seleccionar algunos puntos especiales de la imagen, para realizar un análisis sobre ellos. Esta aproximación es válida, en la medida en que se detecten la cantidad de puntos suficientes de tal forma

que los mismos sean distinguibles y además, formen un conjunto de características estables que permitan ser precisamente localizadas en próximas observaciones.

Cuando se trata de hacer coincidir características entre diferentes imágenes (por ejemplo para el reconocimiento de las mismas), el problema del cambio de escala se hace presente. Al ser analizadas distintas imágenes, éstas, pueden estar tomadas a diferentes distancias respecto al objeto de interés, de tal forma que los objetos aparecen de diferentes tamaños en la imagen. Es así que, si se trata de hacer coincidir las mismas características entre dos imágenes usando un tamaño fijo de píxeles vecinos, la intensidad de los patrones no coincidirá debido al cambio de escala presente en las mismas y por lo tanto, el reconocimiento fallará.

Para solucionar este problema, el concepto de características invariantes a la escala es introducido en visión computacional. La idea central es tener un factor de escala asociada con cada punto característico detectado.

Un detector de puntos de interés usado recientemente y basado en estas ideas es el algoritmo SURF: Speeded Up Robust Features (Detector rápido de características robustas) [Lag11, JG09, BETV08, MS05, TM07, BETV08, OBG11]. Éste está basado en conceptos del algoritmo SIFT: Scale Invariant Feature Transform (Transformación de características invariante a la escala) [BK08, Lag11, JG09, MS05, WRM$^+$10, TM07, LS09, NA02, OBG11, LLLC11, SL04]. El método SURF, no solamente tiene la característica de ser invariante a escala, sino que además, posee la ventaja de calcular eficientemente los resultados asociando una orientación con cada característica (indicado en la imagen de la figura 1 con una línea radial dentro de cada círculo), con el objetivo de lograr invariancia respecto a la rotación.

Si se observa cuidadosamente los puntos detectados en la Fig. 1, se puede ver que el cambio en el tamaño de los círculos, es proporcional a los cambios de escala. Por ejemplo, si se considera la parte inferior de la ventana superior derecha de la fotografía, tanto en la Fig. 1a como en la Fig. 1b, la característica SURF ha sido detectada en la misma ubicación y los círculos correspondientes (de diferentes tamaños) contienen los mismos elementos visuales. Si bien este caso no se da para todas las características, la razón de repetición es lo suficientemente alta para permitir buenas coincidencias entre dos imágenes.

2.1. SURF - Detección de puntos claves

La derivada de una imagen puede ser estimada mediante el uso de filtros Gaussianos, los mismos utilizan un parámetro σ que representa la desviación estándar de la curva Gaussiana y que define la apertura del kernel.

Si se calcula el Laplaciano de un punto en una imagen usando filtros Gaussianos a diferentes escalas, se obtienen diferentes valores. Si miramos la evolución de las respuestas del filtro para diferentes factores de escala, se

(a) (b)

Figura 1: Fotografía tomada a diferentes escalas de la misma escena. (1a) - (1b)

obtiene una curva que alcanza un valor máximo en algún valor de σ específico. Extrayéndose este valor máximo para dos imágenes del mismo objeto (tomadas a dos escalas diferentes), la relación que existe entre los σ máximos se corresponderán en relación con las escalas en que fueron tomadas cada una de las fotografías. Esta observación, es el núcleo del proceso de extracción de características invariantes a la escala. Es decir, que las características invariantes a escalas, pueden ser detectadas como máximos locales en el espacio imagen (localización) y en el espacio escala (obtenido de la aplicación de filtros a diferentes escalas).

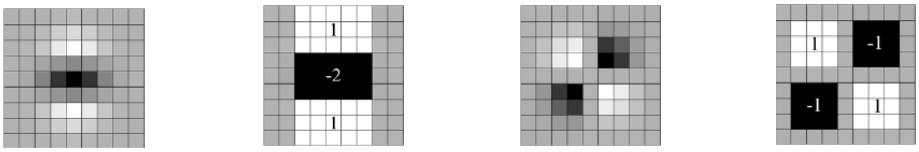
SURF implementa la misma idea. Primeramente para detectar las características, se calcula la Matriz Hessiana (Hessian matrix) para cada píxel. Esta matriz mide la curvatura local de una función y tiene la forma de la matriz de la expresión (1).

$$H(x,y) = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix} \tag{1}$$

La idea es por lo tanto, definir puntos de imágenes con valores de curvatura altos (esto es, alta variación en más de una dirección). Debido que la matriz está compuesta por derivadas de segundo orden, la misma puede ser calculada usando kernels Gaussianos Laplacianos de diferentes escalas

σ. Así, el Hessiano pasa a ser una función de tres variables: $H(x, y, \sigma)$. Una característica invariante a la escala es identificada cuando el determinante del Hessiano alcanza un máximo local en ambos espacios (imagen y escala).

El cálculo de todas las derivadas a diferentes escalas resulta ser costoso computacionalmente, por eso, para que el proceso resulte más eficiente se usa una aproximación a los Kernels Gaussianos conocidos como filtros caja (Boxlets filters) [SBHL98] que se pueden observar en la Fig. 2, de forma tal que el cálculo, sólo involucre algunas operaciones de adición de enteros, reduciendo de esta forma la complejidad y permitiendo el uso de una técnica conocida como imágenes integrales [VJ01].



(a) (b) (c) (d)

Figura 2: Derivadas parciales Gaussianas de segundo orden discretizadas y aproximadas mediante el uso de filtros caja. Las regiones grises de la imagen son iguales a cero. - (2a) discretizada en la dirección "y", (2b) aproximada en la dirección "y"; (2c) discretizada en la dirección "x-y", (2d) aproximada en la dirección "x-y".

El kernel de la Fig. 2.2d es usado para estimar las derivadas parciales segundas, mientras que el de la Fig. 2.2b estima la derivada segunda en la dirección vertical. Una versión rotada de este último, estima la derivada segunda en la dirección horizontal. Los kernels más chicos (como lo de las Fig. 2 tienen un tamaño de 9x9 píxeles, correspondiente a un $\sigma \approx 1{,}2$ y representa la escala más pequeña (máxima resolución espacial). La cantidad de filtros aplicados es especificado como un parámetro del método. Por defecto, se usan 12 diferentes tamaños de kernels (alcanzando un tamaño de 99x99).

Una vez que el máximo local es identificado, la posición precisa de cada punto de interés es obtenida a través de interpolación en ambos espacios (escala e imagen). El resultado es un conjunto de puntos característicos al que se le asocia una valor de escala.

Nota: Anteriormente se ha mencionado que el algoritmo SURF ha sido desarrollado como una variante del detector de características SIFT. Este último, también detecta características como máximos locales en el espacio imagen y escala, pero usa las respuestas del filtro Laplaciano, en vez de el Determinante Hessiano. Este Laplaciano es calculado a diferentes escalas usando el filtro de Diferencia de Gaussianos. Como el cálculo de los puntos característicos esta basado en Kernels de punto flotante, el algoritmo SIFT, es considerado generalmente más preciso que SURF en términos de localización de las características en el espacio imagen y escala, pero resulta computacionalmente más costoso.

2.2. Descripción de las características SURF - Vector descriptor

Como se mencionó anteriormente, el algoritmo SURF define la localización y escala para cada una de las características detectadas. Este factor de escala, puede ser usado para definir el tamaño de una ventana alrededor de cada punto característico, de tal manera de poder definir un área vecina que incluya la misma información visual, sin importar la escala en que el objeto fue fotografiado. Así, esta información visual incluida en esa vecindad, resulta útil para caracterizar el punto y ayuda en la distinción del mismo de otros similares.

Para describir el área vecina de los puntos característicos, se usan descriptores, que usualmente (en el caso de comparación entre imágenes) son vectores N-dimensionales que resultan ser invariantes a cambios de iluminación (en el caso ideal) y a pequeñas deformaciones de perspectivas. Además, resultan potencialmente usables para ser comparados mediante el uso de una métrica de distancia, como por ejemplo: la distancia euclídea.

En el caso del algoritmo SURF, el descriptor por defecto posee 64 elementos (SIFT utiliza 128 elementos) y este vector caracteriza el área que rodea a un punto característico. Cuanto más similares sean los 2 puntos característicos, más cercanos serán sus vectores descriptores.

El buen desempeño de SIFT en comparación con otros descriptores es notable. Su mezcla de información de localización y la distribución de las características relacionadas con el gradiente, lo hacen poderoso para lograr la diferenciación de características, mientras sufre errores de localización en términos de espacio y escala. El uso de los puntos fuertes mencionados, junto con la orientación del gradiente, reduce los efectos de los cambios fotométricos. El descriptor SURF, se basa en propiedades similares, pero con su complejidad simplificada. El primer paso, consiste en fijar una orientación reproducible basada en la información de una región circular alrededor del punto de interés. Luego, se construye una región cuadrada alineada con la orientación seleccionada, y se extrae el descriptor SURF de ella.

2.2.1. Asignación de la orientación

Con el objetivo de lograr invariancia a la rotación, se identifica una orientación que sea reproducible para los puntos de interés. Para este propósito, primeramente se calculan las respuestas de la Wavelet Harr en la dirección "x" y "y" con los kernels de la Fig. 3 en una vecindad circular de radio 6σ alrededor del punto de interés (donde σ representa la escala a la que fue detectado el punto de interés).

Una vez que han sido calculadas las respuestas wavelets y ponderadas por una Gaussiana centrada en el punto de interés (con un valor de $2{,}5\sigma$) estas son representadas mediante un vector en el espacio, con las respuestas de

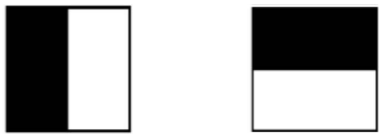
intensidad horizontales a lo largo de la abscisa y las verticales a lo largo de la ordenada. Para una orientación dada, las respuestas dentro de un intervalo angular ($\pi/3$) se suman, y la orientación que da el mayor vector es definido como la orientación dominante.

Cabe aclarar que el tamaño de la ventana deslizante es un parámetro seleccionado experimentalmente.

2.2.2. Componentes del descriptor

Para la extracción del descriptor, el primer paso consiste en construir una región cuadrada centrada alrededor del punto de interés y orientada en la dirección de la orientación calculada en la subsección 2.2.1. El tamaño de la ventana es 20σ.

La región es dividida regularmente en subregiones cuadradas más pequeñas de 4x4. Estas mantienen información espacial importante en ellas. Por razones de simplicidad, llamaremos dx a la respuesta de la wavelet Harr en la dirección horizontal Fig. 3a y dy la respuesta en la dirección vertical Fig. 3b (tamaño del filtro: 2σ). "Horizontal" y "Vertical" aquí son definidas en relación a la orientación seleccionada del punto. Para incrementar la robustez frente a deformaciones geométricas y errores de localización, las repuestas dx y dy son primero ponderadas con un gaussiano ($3{,}3\sigma$) centrado en el punto de interés.



(a) (b)

Figura 3: Kernels aplicados en la vecindad de un punto característico.

Las respuestas wavelet dx y dy son sumadas sobre cada subregión y forman un primer conjunto de entradas para el vector de características. Con el propósito de dar información acerca de la polaridad de los cambios de intensidad, también se extrae la suma de los valores absolutos de las respuestas: $|dx|$ y $|dy|$. Así, cada subregión tiene un vector descriptor de cuatro dimensiones cuya expresión es:

$$\left[\sum dx \qquad \sum dy \qquad \sum |dx| \qquad \sum |dy|\right] \tag{2}$$

Esto, resulta en un vector descriptor para todas las subregiones de 4x4 de un tamaño de 64 elementos. Se debe tener en cuenta que las respuestas de las wavelets son invariantes a un sesgo en la iluminación y que la invariancia de contraste (un factor de escala) se consigue transformando el vector en uno unidad.

La Fig. 4, muestra las propiedades del descriptor para tres imágenes distintas, con diferentes patrones de intesidad, en una subregión. Una combinación de dichos patrones locales de intensidad, daría como resultado un descriptor posible de distinguir.

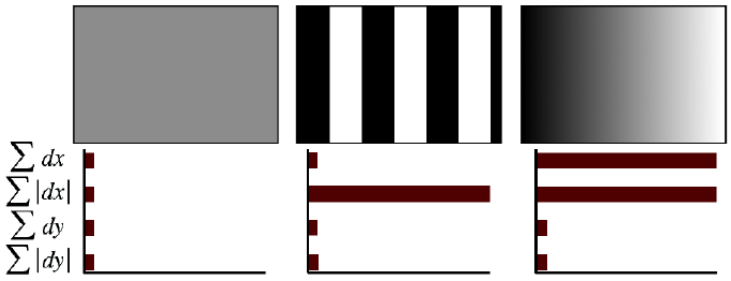


Figura 4: Las entradas del descriptor de una subregión, representan la naturaleza del patrón de intensidad subyacente. Izquierda: en el caso de una región homogénea, todos los valores son relativamente bajos. Centro: En presencia de frecuencias en la dirección "x", el valor $\sum |dx|$ es alto, mientras los demás son bajos. Derecha: Si la intensidad se incrementa gradualmente en la dirección "x", ambos valores: $\sum dx$ y $\sum |dx|$ son altos.

De esta forma, buscar correspondencias con invariancia a escala entre imágenes es alcanzable mediante las características y descriptores que se obtiene con SURF.

Aclaración respecto del algoritmo SIFT: El algoritmo SIFT, también define su propio descriptor. Se basa en la magnitud del gradiente y orientación calculado en la escala del punto clave considerado. Como en el caso de los descriptores SURF, el área vecina escalada del punto clave es dividido en 4x4 subregiones. Para cada una de estas regiones, se construye un histograma de 8 clases de las orientaciones del gradiente (ponderados por su magnitud y por una ventana gaussiana global centrada en el punto clave). Luego, el vector descriptor es construido de las entradas de este histograma. Hay 4x4 regiones y 8 clases por histograma, lo que nos da un descriptor de una longitud de 128 elementos.
En cuanto a la detección de características, la diferencia entre los descriptores SIFT y SURF es principalmente la velocidad y precisión. Mientras que los descriptores SURF están mayormente basados en diferencias de intensidades, son más rápidos de calcular; por otro lado, los descriptores SIFT son considerados generalmente más precisos en buscar la característica correcta (coincidencia más exacta), pero llevando más tiempo de cálculo.

Referencias

[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.

[JG09] Luo Juan and Oubong Gwon. A comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing (IJIP)*, 3(4):143–152, 2009.

[Lag11] Robert Laganière. *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Publishing, June 2011.

[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.

[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.

[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.

[SBHL98] Patrice Simard, Léon Bottou, Patrick Haffner, and Yann LeCun. Boxlets: A fast convolution algorithm for signal processing and neural networks. In Michael J. Kearns, Sara A. Solla, and David A. Cohn, editors, *NIPS*, pages 571–577. The MIT Press, 1998.

[SL04] Iryna Skrypnyk and David G. Lowe. Scene modelling, recognition and tracking with invariant image features. In *ISMAR*, pages 110–119. IEEE Computer Society, 2004.

[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.

[VJ01] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proc. CVPR*, 1:511–518, 2001.

[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.