

Proyecto Final

Reconocimiento de Objetos en Imágenes a través de Matching de
Descriptores ASIFT mediante un enfoque estadístico *A Contrario* y
MAC-RANSAC

Introducción al Reconocimiento de Patrones
Curso 2010

Manuel López Radcenco
C.I.: 4.086.135-1

Facultad de Ingeniería
Universidad de la República

14/03/2012

Índice general

| | |
|---|-----------|
| Introducción | 2 |
| Marco Teórico y Detalles de Implementación | 3 |
| Descriptores ASIFT | 3 |
| Modelo de adquisición de imágenes | 3 |
| Matching A Contrario | 7 |
| Métrica CEMD | 8 |
| Comparación de descriptores ASIFT | 10 |
| Metodología A Contrario | 11 |
| Matches significativos | 12 |
| Algoritmo MAC-RANSAC | 13 |
| El algoritmo RANSAC | 13 |
| AC-RANSAC Generalizado | 14 |
| MAC-RANSAC para detección de múltiples grupos | 16 |
| Filtrado de correspondencia espacial | 17 |
| Detección de la fusión de varias transformaciones | 17 |
| Algoritmo MAC-RANSAC | 19 |
| Algoritmo Implementado | 21 |
| Descripción | 21 |
| Puntos Críticos | 21 |
| Resultados Obtenidos | 23 |
| Matching A Contrario | 23 |
| MAC-RANSAC | 26 |
| Prueba 1 | 27 |
| Prueba 2 | 30 |
| Conclusiones | 32 |
| Bibliografía | 33 |

Introducción

Este informe pretende introducir al lector en los resultados obtenidos y la teoría detrás de un algoritmo implementado para la detección de objetos en imágenes.

Dicho algoritmo se basa en el matcheo de características entre imágenes para la detección de objetos. Para ello, el algoritmo se vale de descriptores ASIFT (Affine Scale Invariant Feature Transform), los cuales son obtenidos mediante el algoritmo presentado en [1]. El matcheo de los mismos se realizará, luego, mediante un enfoque estadístico *A Contrario* presentado en [2]. Finalmente, el algoritmo MAC-RANSAC presentado en [3] permitirá reconocer o detectar varios objetos simultáneamente a partir de detección de grupos de correspondencias coherentes con respecto a una transformación común.

Presentaremos, en primera instancia, las consideraciones teóricas y de modelado previo que serán la base para el posterior desarrollo del algoritmo.

A continuación se hará una breve descripción del algoritmo implementado, así como sus puntos críticos.

Se culminará, finalmente, presentando los resultados obtenidos al aplicar el algoritmo y las conclusiones obtenidas a partir del análisis de dichos resultados.

Marco Teórico y Detalles de Implementación

Descriptores ASIFT

Es sabido que si un objeto tiene límites bien definidos (al menos bien definidos por tramos) las imágenes obtenidas por cámaras en posiciones variables se relacionan por deformaciones de carácter “suave”, las cuales pueden ser bien aproximadas por transformaciones afines del plano de la imagen.

En consecuencia, el problema clásico de detección de objetos se reduce al cómputo de características locales invariantes frente a transformaciones afines. Para obtener dichas características nos basaremos en el algoritmo ASIFT, el cual se basa en descriptores SIFT para lograr esto.

Una transformación afín puede ser descrita completamente mediante 6 parámetros. Un algoritmo de detección de características invariantes frente a transformaciones afines debe ser capaz de cubrir estos 6 parámetros.

El algoritmo SIFT es capaz de cubrir 4 de esos 6 parámetros mediante la normalización de rotaciones y traslaciones y la simulación de zoom. A partir de ello, SIFT construye, para cada punto de interés, un descriptor compuesto por histogramas circulares de la orientación del gradiente en regiones en torno a dicho punto.

Puede demostrarse que dichos descriptores resultan invariantes ante cambios de escala, rotaciones y traslaciones, pero que los mismos no resultan robustos frente a otro tipo de transformaciones.

ASIFT (Affine-SIFT) complementa SIFT simulando los dos parámetros restantes, que representan cambios en la orientación de la cámara al momento de adquirir la imagen. De esta forma, se simulan imágenes desde una amplia gamma de posibles orientaciones de la cámara sin aumentar considerablemente el costo computacional. Se obtiene entonces un método que es completamente invariante ante transformaciones afines.

Modelo de adquisición de imágenes

En la figura 1 se muestra el modelo utilizado para representar la cámara al momento de adquirir una imagen.

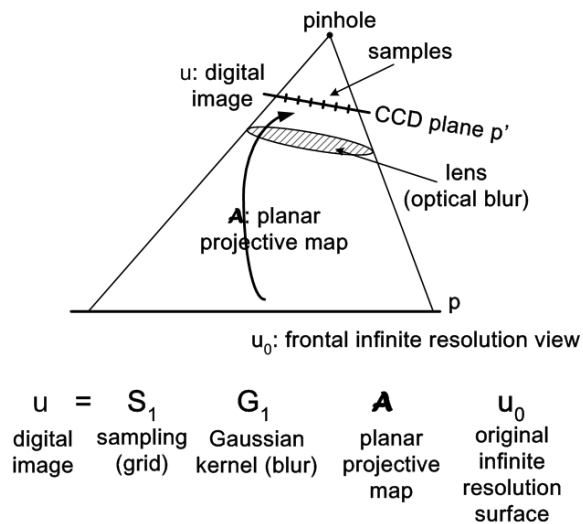


Figura 1: Modelo de una cámara para adquisición de datos

de donde puede deducirse que la adquisición de una imagen digital de un objeto plano puede describirse mediante la ecuación:

$$u = S_1 G_1 A T_{u_0} \quad (1)$$

donde u es la imagen digital y u_0 es una vista frontal ideal (de resolución infinita) de un objeto plano.

Los parámetros T y A son un transición entre planos y un mapa de proyecciones entre planos debidos al movimiento de la cámara respectivamente.

G_1 representa un modelo de convolución Gaussiana para representar el difumado óptico y S_1 es el operador estándar de muestreo sobre una rejilla regular de paso 1. Asumimos que el núcleo Gaussiano es lo suficientemente ancho para asegurar que no hay superposición espectral (aliasing), por lo cual se podrá recuperar la imagen continua a partir de su versión discreta mediante una interpolación de Shannon-Whittwaker I, teniendo así:

$$I S_1 G_1 A T_{u_0} = G_1 A T_{u_0} S_1 \quad (2)$$

Aproximación afín local

El modelo anterior puede simplificarse si se reduce A a un mapa afín. Dada la tangencialidad local de las deformaciones de perspectiva a los mapas afines, y mediante una aproximación de Taylor, cualquier deformación planar suave puede ser aproximada en torno a cada punto por un mapa afín.

La deformación en perspectiva de un objeto plano introducida por el movimiento de la cámara es una transformación homográfica plana suave, que es localmente tangente a la transformación afín $u(x, y) \rightarrow u(ax + by + e, cx + dy + f)$ en cada región de la imagen.

Cada mapa afín (con determinante estrictamente positivo) que no sea una función de similitud tiene una única descomposición de la forma:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = H_\lambda R_1(\Psi) T_t R_2(\Phi) = \begin{bmatrix} \cos(\Psi) & -\sin(\Psi) \\ \sin(\Psi) & \cos(\Psi) \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\Phi) & -\sin(\Phi) \\ \sin(\Phi) & \cos(\Phi) \end{bmatrix}$$

dónde $\lambda > 0$, t es el determinante de A , R_i son rotaciones, $\Psi, \Phi \in [0, \pi)$ y T_t es una matriz diagonal con un primer valor propio $t > 1$ y un segundo valor propio unitario, llamada *tilt*.

En la figura 2 puede apreciarse la interpretación geométrica de este modelo simplificado.

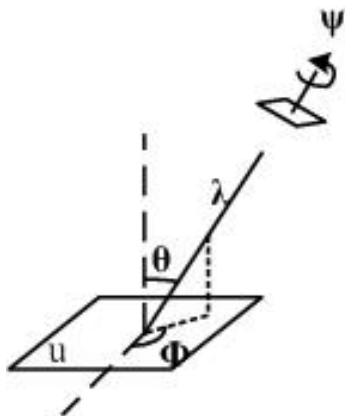


Figura 2: Interpretación geométrica de la descomposición afín

En este modelo interpretativo del movimiento de la cámara como una descomposición afín, Φ y $\Theta = \arccos(1/t)$ son los ángulos de visión, Ψ parametriza el giro de la cámara y λ representa el zoom.

Asumimos que la cámara se mantendrá lejos de la imagen u e inicialmente se encuentra en una posición de vista frontal ($\lambda = 1, t = 1, \Psi = \Phi = 0$).

Al mover la cámara en forma paralela al plano de la imagen obtenemos la traslación T . La misma puede eliminarse fácilmente sin pérdida de generalidad asumiendo que el eje de la cámara se encuentra con el plano de la imagen en un punto fijo.

El plano que contiene la normal de la imagen y el eje óptico forma un ángulo Φ con un plano vertical fijo. Este ángulo es llamado “longitud”.

El eje óptico forma un ángulo Θ con la normal al plano de la imagen u . Este ángulo es llamado “latitud”.

La cámara, además, puede rotar con respecto a su eje óptico, obteniendo entonces el parámetro de rotación Ψ .

Por último, la cámara puede moverse hacia adelante o atrás, generándose entonces un zoom caracterizado por el parámetro λ .

Este modelo es suficiente para dar una descripción correcta de cualquier deformación proyectiva local. En caso de que la cámara no se encontrara lejos de la imagen, la transformación dada por la cámara sería una homografía. Aún así, una homografía es localmente tangente a un mapa afín, por lo cual el modelo anterior continúa siendo válido localmente.

El parámetro t es el tilt absoluto entre una imagen adquirida mediante el modelo de cámara anterior y una vista frontal del objeto. Al comparar dos o más imágenes, en general ninguna de ellas es una vista frontal del objeto, por lo cual el tilt entre estas dos imágenes estará compuesto y determinado por la combinación de los tilts absolutos de ambas imágenes así como por los ángulos de longitud de las mismas. A este nuevo tilt se le denomina tilt de transición, y es en general considerablemente mayor que los tilts absolutos de las imágenes a comparar. Este parámetro es, además, un buen indicador de la deformación producida por la transformación afín asociada al pasaje entre ambas imágenes. Será necesario, entonces, que el algoritmo de detección de características sea robusto y pueda manejar eficientemente tilts de transición elevados.

Descripción del algoritmo ASIFT

ASIFT complementa a SIFT mediante la simulación de varias vistas a partir de la variación de dos parámetros en el modelo anterior. Un esquema gráfico de esto puede apreciarse en la figura 3.

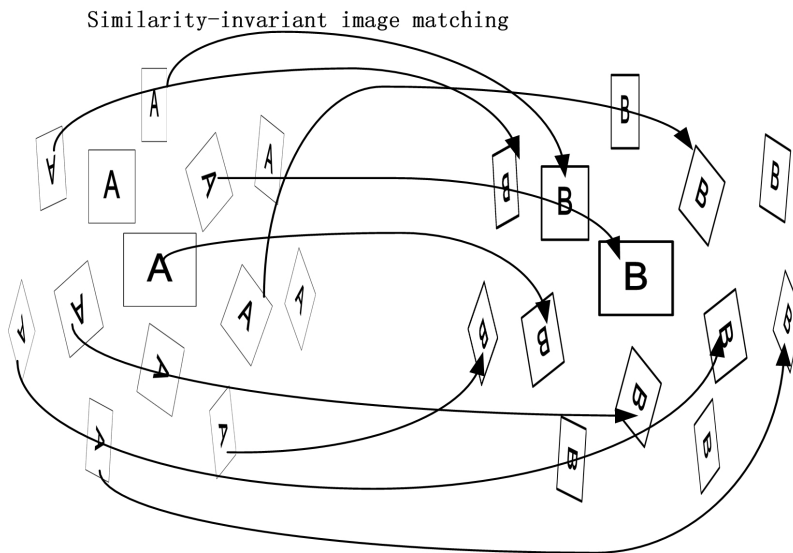


Figura 3: Esquema de funcionamiento de ASIFT

De esta forma, se simulan variaciones en los parámetros que modelan la orientación del eje óptico de la cámara (Ψ y Φ). A las imágenes simuladas se les aplica entonces SIFT y se obtiene, a partir de ello, un método que cubre los 6 parámetros del modelo afín.

En definitiva, ASIFT simula 3 parámetros (escala, longitud y latitud, siendo esto últimos dos equivalentes al tilt) y normaliza otros 3 parámetros (dos parámetros de traslación y uno de rotación).

Puede demostrarse matemáticamente que las características detectadas mediante ASIFT son totalmente invariantes frente a transformaciones afines.

Finalmente, si se eligen el muestro espacial de parámetros afines necesario para la simulación en forma adecuada (tal como puede verse en las figuras 4 y 5), el cómputo de descriptores ASIFT no resulta demasiado más complejo a nivel computacional

que el cálculo de descriptores SIFT. Por ello, el costo computacional de ASIFT, si bien es considerablemente más alto, no resulta ser un inconveniente prohibitivo.

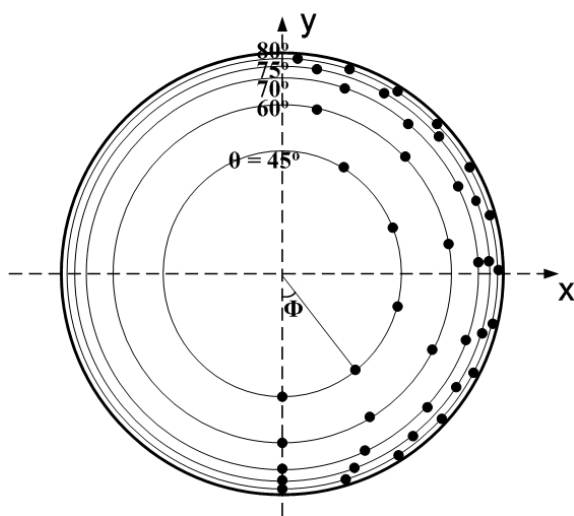


Figura 4: Muestreo espacial óptimo para ASIFT

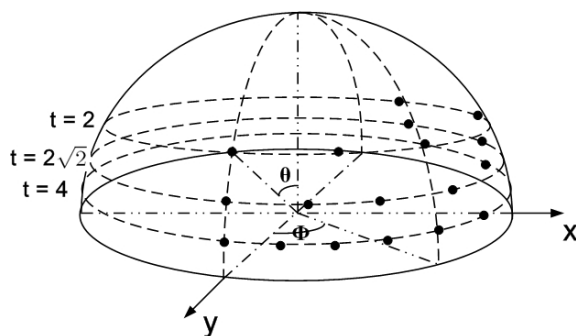


Figura 5: Muestreo espacial óptimo para ASIFT

Una vez obtenidos los descriptores, el próximo paso en un algoritmo de detección de objetos mediante matcheo de imágenes es el matcheo de características. La teoría detrás del algoritmo de matcheo se analizará a continuación.

Matching A Contrario

Uno de los aspectos más básicos e importantes en el matcheo de imágenes es definir una métrica que permita establecer la similitud entre dos características. A su vez, es importante definir un criterio mediante el cual pueda afirmarse que dos características deberían ser asociadas. Es así que la elección de la métrica y del criterio de matcheo resultan ser pasos vitales en el diseño de un algoritmo de matcheo.

La métrica elegida debe ser capaz de brindar una comparación eficiente entre características diferentes, pero a su vez ser lo suficientemente robusto como para detectar

pequeñas variaciones en las características.

El criterio de matcheo deberá, a su vez, adaptarse a la complejidad y diversidad de las características.

El algoritmo de matching utilizado trabaja en base a una nueva distancia (basada en la conocida Earth Mover Distance) que ha demostrado un buen comportamiento frente a la cuantización bin a bin, mientras que presenta un mejor comportamiento que otras distancias clásicas bin a bin en el contexto de la comparación de características locales.

Además, se utiliza un criterio estadístico con umbrales adaptivos que permite la adecuación a la complejidad de los descriptores a la vez que limita el número de detecciones, permitiendo adicionalmente un buen comportamiento si el objeto a detectar no está presente en la imagen.

Métrica CEMD

En general las métricas clásicas bin a bin (distancia Euclidiana, Manhattan o χ^2 , entre otras) suelen ser consideradas la forma más simple y menos costosa computacionalmente para comparar descriptores ASIFT (que representan las características mediante histogramas unidimensionales de orientación). Sin embargo, las mismas no resultan robustas frente a la cuantización de bins, por lo cual la elección del número de bins determina un compromiso entre sensibilidad y robustez.

Dentro de la amplia variedad de métricas posibles se opta por trabajar sobre una variante nueva de la distancia EMD planteada en [4]. El costo computacional de la misma lleva a la necesidad de desarrollar una nueva distancia optimizada para comparar histogramas circulares de una dimensión, tales como los histogramas de orientación presentes en un descriptor ASIFT.

Consideremos dos histogramas circulares normalizados $f = (f[i])_{i=1\dots N}$ y $g = (g[i])_{i=1\dots N}$. La distancia EMD entre ellos se define como:

$$EMD(f, g) = \min_{\alpha_{i,j} \in M} \sum_{i=1}^N \sum_{j=1}^N \alpha_{i,j} c(i, j) \quad (3)$$

donde

$$M = \left\{ (\alpha_{i,j}); \alpha_{i,j} \geq 0, \sum_j \alpha_{i,j} = f[i], \sum_i \alpha_{i,j} = g[j] \right\}$$

y dónde además c es una distancia entre bins. Para histogramas circulares esta distancia puede ser definida de forma natural como:

$$c(i, j) = \frac{1}{N} \min(|i - j|, N - |i - j|), \forall (i, j) \in \{1\dots N\}^2$$

Puede pensarse en la distancia EMD como un costo de transporte. El valor $c(i, j)$ representa el costo de mover una unidad de masa del bin i al bin j . Si bien esta distancia puede utilizarse en cualquier dimensión, el costo computacional se vuelve extremadamente alto para dimensiones mayores a 2. Para histogramas no-circulares

y unidimensionales, si se elige $c = \frac{1}{N}|i - j|$ puede demostrarse que $EMD(f, g) = \|F - G\|_1 = \frac{1}{N} \sum_{i=1}^N |F[i] - G[i]|$, donde F y G se definen como

$$F[i] = \sum_{j=1}^N f[j] \quad G[i] = \sum_{j=1}^N g[j] \quad (4)$$

Esta formula puede generalizarse fácilmente a histogramas circulares si los mismos son unidimensionales, obteniéndose entonces:

$$CEMD(f, g) = \min_{\mu \in [-1, 1]} \|F - G - \mu\|_1 = \frac{1}{N} \sum_i |F[i] - G[i] - \mu| \quad (5)$$

donde F y G están definidos como en 4.

Es importante observar que dicho mínimo es fácil de computar, dado que la función se minimiza en la mediana de los valores de $F[i] - G[i]$, $i = 1 \dots N$, manteniéndose la validez de la formula independientemente del bin elegido a la hora de comenzar a computar los histogramas acumulativos de f y g .

Puede afirmarse entonces:

$$CEMD(f, g) = \frac{1}{N} \min_{k \in \{1 \dots N\}} \sum_i |F[i] - G[i] - F[k] + G[k]| \quad (6)$$

Finalmente, definiendo un histograma acumulativo F_k de f según:

$$F_k[i] = \begin{cases} \sum_{j=k}^i f[j] & i \geq k \\ \sum_{j=k}^i f[j] + \sum_{j=1}^k f[j] & i < k \end{cases}$$

es posible ver que $F_k[i] = F[i] - F[k-1]$ si $k \geq i$ y $F_k[i] = F[i] + 1 - F[k-1]$ si $k < i$.

Finalmente, si se observa que $F[0] - G[0] = F[N] - G[N] = 0$ y trabajando sobre 6 se tiene

$$\begin{aligned} CEMD(f, g) &= \frac{1}{N} \min_{k \in \{1 \dots N\}} \sum_i |F[i] - G[i] - F[k] + G[k]| \\ &= \frac{1}{N} \min_{k \in \{1 \dots N\}} \sum_i |F[i] - G[i] - F[k-1] + G[k-1]| \\ &= \frac{1}{N} \min_{k \in \{1 \dots N\}} \sum_i |F_k[i] - G_k[i]| \end{aligned}$$

y por lo tanto

$$CEMD(f, g) = \min_{k \in \{1 \dots N\}} \|F_k - G_k\|_1 \quad (7)$$

por lo cual la distancia $CEMD(f, g)$ es también el mínimo en k de la distancia L^1 entre los histogramas acumulativos F_k y G_k .

Comparación de descriptores ASIFT

Se verá ahora como aplicar la distancia descrita en la sección anterior para comparar descriptores ASIFT.

Para ello es necesario recordar que un descriptor ASIFT está compuesto de M histogramas circulares a_m de orientaciones de gradientes, ponderados según la magnitud del gradiente, y calculados en una región en torno al punto de interés. Entonces, la comparación de características se reduce a comparar histogramas circulares, los cuales se asumen cuantizados en N bins y normalizados globalmente a nivel de descriptor.

En el enfoque clásico, lo usual es utilizar distancias bin a bin L^p , lo cual requiere que descriptores sean normalizados globalmente.

Para utilizar la distancia $CEMD$ para comparar dos descriptores $a = (a_1 \dots a_m)$ y $b = (b_1 \dots b_m)$ se calculará la distancia entre cada par (a_m, b_m) . Si bien en teoría esta distancia debería aplicarse a histogramas normalizados, resulta mucho más robusto normalizar todo el descriptor en forma global. Esto implicará que se deberán comparar histogramas de distintos pesos, por lo cual las ecuaciones 5 y 7 no son equivalentes.

Por este motivo se prefiere la fórmula 7, que resulta ser independiente del origen k de los histogramas circulares.

A partir de ello, la distancia entre descriptores será definida, simplemente, como la suma de las distancias entre histogramas. Esto es:

$$D_{CEMD} = \sum_{m=1}^N CEMD(f, g) \quad (8)$$

La elección de esta distancia frente a otras posibilidades se basa en que la misma a demostrado ser la opción más robusta.

Implementación y Costo Computacional

Definiendo $X[k] = F[K] - G[k]$, siendo esta la diferencia entre histogramas acumulativos. Es fácil ver que X_k puede ser escrito en función de X_1 , según:

$$X_k[i] = \begin{cases} X_1[i] & k = 1 \\ X_1[i] - X_1[k - 1] & i \geq k > 1 \\ X_1[i] - X_1[k - 1] + X_1[N] & i < k \end{cases} \quad (9)$$

Por ende, el cálculo de la distancia entre descriptores no requiere el cálculo de todos los histogramas acumulativos. De hecho, el único agregado con respecto al costo computacional necesario si se utiliza una distancia bin a bin clásica está dado por la minimización de $\|X_k\|_1$ según k .

A partir de ello, la complejidad computacional de este enfoque es N veces la complejidad computacional existente al utilizar una distancia bin a bin clásica, como la distancia Euclideana, siendo N la cantidad de bins en los cuales están cuantizados los histogramas de cada descriptor ASIFT.

Metodología A Contrario

Se introducirá ahora una forma genérica de calcular umbrales de detección para descriptores ASIFT en base a la distancia anterior. Para ello trabajaremos con N_Q descriptores $\{a^i\}$ a matchear contra N_C descriptores $\{b^j\}$ candidatos. Debemos entonces determinar para cada a^i dado cuales b^j deben ser matcheados con él. Nos basaremos, para ello, en un enfoque estadístico A Contrario.

La idea general de este enfoque consiste en detectar grupos de características de independencia poco probable bajo cierta hipótesis. Dicha hipótesis es conocida como la hipótesis nula.

En otras palabras, se detectarán aquellos grupos que no podrían aparecer debido a una coincidencia, rechazándose aquellos grupos que surgen debido al azar.

Otro aspecto importante de este tipo de enfoques es que para computar la improbabilidad de existencia de un grupo se estimará la cantidad de grupos existentes, y no la probabilidad de existencia del grupo, la cual suele ser, en general, imposible de calcular.

Para aplicar un enfoque a contrario solamente es necesaria una métrica y un supuesto bajo el cual validar los matches.

Hipótesis Nula

Para definir la hipótesis nula se asume que la distancia entre dos descriptores a^i y b^j puede ser escrita como $D(a^i, b^j) = \sum_{m=1}^M d(a_m^i, b_m^j)$, lo cual se verifica para la distancia DCEMD.

Dado un descriptor aleatorio b definimos la siguiente hipótesis nula:

\mathbf{H}_0^i : “ $d(a_m^i, b_m^j)(m \in \{1 \dots N\})$ son variables aleatorias mutuamente independientes”

Bajo la hipótesis nula la probabilidad de que la distancia entre a_i y b sea más pequeña que un umbral δ dado es:

$$\mathbb{P}(D(a^i, b) \leq \delta | \mathbf{H}_0^i) = \int_{-\infty}^{\delta} \ast_{m=1}^M p_m^i(x) dx \quad (10)$$

donde \ast denota el producto convolución y p_m^i es la densidad de probabilidad de la función $d(a_m^i, b_m^j)$.

Así, la validez de un match será determinada limitando esta probabilidad, lo cual determinará umbrales para las distancias que dependerán de a^i y la distribución observada a lo largo de los posibles candidatos a matchear.

Para computar la probabilidad expresada en 10 es necesario estimar las densidades de probabilidades p_m^i , para lo cual simplemente utilizamos histogramas de las distancias calculados a lo largo de la base de datos de candidatos disponibles, es decir, para cada $i \in \{1 \dots N_Q\}$ y cada $m \in \{1 \dots M\}$ se estima p_m^i empíricamente a partir del conjunto $\{b^1 \dots b^{N_C}\}$.

Matches significativos

Se consideran dos descriptores a^i y b^j a una distancia $\delta = D(a^i, b^j)$. Se decide matchear estos descriptores si $\mathbb{P}(D(a^i, b) \leq \delta | \mathbf{H}_0^i)$ es lo suficientemente pequeña.

Para generar umbrales automáticos para dicha probabilidad, entonces, se utiliza el esquema general de los enfoques a contrario, que consiste en controlar el promedio de falsas detecciones.

Dado que se efectúan $N_q N_c$ comparaciones al buscar matches, se define el siguiente umbral para un $\epsilon > 0$ dado:

$$\delta_i(\epsilon) = \arg \max_{\delta} \left\{ \mathbb{P}(D(a^i, b) \leq \delta | \mathbf{H}_0^i) \leq \frac{\epsilon}{N_q N_c} \right\} \quad (11)$$

Un match entre a^i y b^j se dice que es “ ϵ -meaningful” si $D(a^i, b^j) \leq \delta_i(\epsilon)$.

Este enfoque puede justificarse mediante el hecho de que al testear N_Q descriptores contra N_C candidatos bajo la hipótesis nula el número esperado de matches es menor a ϵ .

Este resultado es una consecuencia simple de la linealidad de la esperanza. Es claro que resultaría mucho más complicado establecer una cota para la probabilidad de una falsa detección, ya que las distancias entre descriptores no son necesariamente independientes.

En la práctica, para un ϵ fijo y un descriptor a^i dado:

1. Se estiman las densidades de probabilidad p_m^i de las distancias $d_m(a^i, b^j)$ a partir de histogramas calculados cuando b^j recorre la base de datos de candidatos.
2. Se calcula $\mathbb{P}(D(A^i, B^j) \leq \delta_i(\epsilon))$ mediante la fórmula 10.
3. El umbral $\delta_i(\epsilon)$ se calcula mediante la fórmula 11.
4. Para cada descriptor b^j , a^i se matchea con b^j si $D(a^i, b^j) \leq \delta_i(\epsilon)$.

Es claro que fijar el valor de ϵ (el cual representa el número esperado de falsas detecciones y, a su vez, termina determinando un umbral para las distancias) resulta considerablemente más sencillo que determinar un umbral para las distancias en forma directa, dado que las mismas pueden variar considerablemente, tanto entre descriptores como entre imágenes.

Este umbral dependerá tanto del descriptor a^i como de los candidatos de los que se disponga en la base de datos.

Es esperable, entonces, que el desempeño del algoritmo sea bueno si se cuenta con una base de datos suficientemente grande.

Es interesante observar, finalmente, que los matches no están restringidos al vecino más cercano (como sucede en una gran parte de los enfoques clásicos), a pesar de que dicha restricción puede resultar útil y debería ser agregada en ciertas aplicaciones. No contar con dicha restricción, sin embargo, no implicará un gran aumento en la cantidad de matches erróneos.

Por último, es destacable el hecho de que el único parámetro que es necesario setear es ϵ . Siguiendo el enfoque clásico de los esquemas a contrario, este parámetro puede setearse sin problemas a 1. Sin embargo, dado que en un descriptor ASIFT en general los histogramas que lo componen no son completamente independientes, resulta más robusto elegir valores de ϵ uno o dos ordenes de magnitud menores ($\epsilon = 10^{-1}$, $\epsilon = 10^{-2}$)

Algoritmo MAC-RANSAC

Una vez que se han detectado todos los matches es necesario discriminar las distintas transformaciones involucradas, con el fin de detectar así diversos objetos o múltiples ocurrencias de un mismo objeto. Cada transformación detectada permitirá determinar si el objeto buscado se encuentra en la imagen analizada, así como determinar y diferenciar múltiples apariciones del mismo, ya que cada una estará asociada a una única transformación.

Para realizar esta detección de transformaciones nos basaremos en el algoritmo MAC-RANSAC descrito en [3]. El mismo es una generalización del algoritmo RANSAC clásico, el cual se modifica a partir de un enfoque estadístico A Contrario, para ser luego adaptado para detectar múltiples transformaciones en forma secuencial.

El algoritmo RANSAC

Consideremos $\mathbb{C} = \{(m_i, m'_i) | i = 1 \dots N\}$ un conjunto de correspondencias de puntos entre dos imágenes I e I' , dónde es posible que un punto m_i de I sea matcheado con varios puntos m'_i de I' y viceversa. Deseamos detectar si algunas de esas correspondencias pueden ser explicadas por una transformación geométrica común. Bajo esta premisa, el algoritmo RANSAC (RANdom SAMple CONsensus) intentará encontrar una transformación afín T (la cual puede ser determinada totalmente a partir de $n = 3$ puntos) que permita explicar de la mejor forma posible un máximo de correspondencias de \mathbb{C} .

En cada paso, un subconjunto S' de n correspondencias es sacado de \mathbb{C} para estimar la transformación.

Luego, para cada correspondencia de $C \setminus S$ se define un error que permita medir la adecuación de dicha correspondencia con la transformación.

Se definen, entonces, los inliers como aquellas correspondencias cuyo error es menor a un umbral δ_R . El error total del cada conjunto está directamente relacionado con el tamaño del mismo. Este proceso se repite i_{MAX} veces, eligiéndose como grupo al mayor de los grupos detectados, siempre y cuando este tenga un tamaño mayor a un umbral N_C .

Es importante notar que existen 3 parámetros que deben ser seteados por el usuario (δ_R , i_{MAX} y N_C), los cuales resultan ser críticos en el desempeño del algoritmo.

AC-RANSAC Generalizado

Para lidiar con las limitaciones del algoritmo RANSAC clásico, se generaliza dicho algoritmo mediante un enfoque A Contrario que se encargará de fijar los parámetros de dicho algoritmo en forma automática. Para ello, se combina el algoritmo clásico con un test de hipótesis que permite decidir, para cada S' y T estimada a partir de él, si un conjunto dado S puede considerarse bien descrito por dicha transformación.

Marco de trabajo A Contrario

Dado un conjunto $\mathbb{C} : \{(m_i, m'_i) | i = 1 \dots N\}$ de correspondencias entre dos imágenes I e I' , se busca decidir si un subgrupo de dichas correspondencias se puede explicar por una única transformación.

Para ello, AC-RANSAC se basa en cierta hipótesis nula \mathbb{H}_0 que describe una distribución genérica de correspondencias al azar $(m_i, m'_i), i = 1 \dots N$ para las cuales se acepta que no podrá encontrarse una transformación que las explique de forma consistente.

Así, un conjunto significativo puede ser encontrado buscando aquellos grupos de correspondencias cuya probabilidad de existencia es pequeña bajo la hipótesis nula.

Un conjunto \mathbb{C} de N correspondencias aleatorias $(m_i, m'_i), i = 1 \dots N$ satisface la hipótesis nula si:

- Los puntos m_i y $m'_j, i, j = 1 \dots N$ son variables aleatorias mutuamente independientes
- Los puntos $m_i, i = 1 \dots N$ se distribuyen uniformemente en la imagen I
- Los puntos $m_j, j = 1 \dots N$ se distribuyen uniformemente en la imagen I'

Si consideramos entonces un conjunto \mathbb{C} de N correspondencias que verifican la hipótesis nula, un subconjunto S' de \mathbb{C} que verifique $\#S' = n$ y un subconjunto S de \mathbb{C} tal que $S \cap S' = \emptyset$ puede definirse la rigidez del conjunto S con respecto a la transformación $T_{S'}$ estimada a partir de S' como:

$$\alpha(S, T_{S'}) = \max_{(m, m') \in S} \max \left(\frac{\pi}{A'} d(T_{S'} m, m')^2, \frac{\pi}{A} d(m, T_{S'}^{-1} m')^2 \right) \quad (12)$$

donde A y A' son las áreas de las imágenes I e I' respectivamente.

La rigidez de un conjunto S mide el nivel de adecuación del conjunto con respecto a la transformación $T_{S'}$. Adicionalmente, se verifica que bajo la hipótesis nula, la probabilidad de observar una rigidez $\alpha(S, T_{S'})$ menos que un valor dado α está acotada por $\alpha^{\#S}$:

$$P_{\mathbb{H}_0} [\alpha(S, T_{S'}) \leq \alpha] \leq \alpha^{\#S}$$

Entonces, un conjunto S se considera bien explicado por una transformación $T_{S'}$ asociado a un conjunto S' en la medida que $P_{\mathbb{H}_0} [\alpha(S, T_{S'}) \leq \alpha]$ sea lo suficientemente pequeña. En otras palabras, S se considera bien explicado por $T_{S'}$ si es poco probable observar una rigidez pequeña bajo la hipótesis nula.

Para definir más formalmente cómo determinar que tan pequeña debe ser la rigidez consideremos un conjunto $\mathbb{C} = (m_i, m'_i, 1 \dots N)$ de N correspondencias entre dos imágenes I e I' y un subconjunto S de \mathbb{C} tal que $\#S = K$. Para un $\epsilon > 0$ dado decimos que el conjunto S es ϵ -meaningful si existe un subconjunto S' de \mathbb{C} tal que $\#S' = n$, $S \cap S' = \emptyset$ y además:

$$NFA(S, S') = (N - n) \binom{N}{K} \binom{N - K}{n} ((\alpha(S, T_{S'}))^K \leq \epsilon \quad (13)$$

La cantidad $NFA(S, S')$ da una medida de la calidad de la adecuación del conjunto S y la transformación $T_{S'}$, combinando en un mismo término las rigidez de S con respecto a $T_{S'}$ y su tamaño.

La fórmula 13 nos permitirá computar umbrales adaptivos sobre las rigideces $\alpha(S, T_{S'})$.

Algoritmo AC-RANSCAC

El algoritmo AC-RANSAC, desarrollado a partir de lo expuesto anteriormente sorteá, en cada iteración y en forma aleatoria, un subconjunto S de \mathbb{C} para luego estimar la transformación $T_{S'}$. El resto de las correspondencias $c_i = (m_i, m'_i)$ de $\mathbb{C} \setminus S'$ son ordenadas de acuerdo al valor de sus rigideces $\alpha_i = \alpha(c_i, T_{S'})$.

Hallar el conjunto S que minimice el NFA es sencillo. Para cada K menor a $N - n$, el conjunto de tamaño K que minimiza $NFA(S, S')$ está formado por las K correspondencias c_i con la menor rigidez α_i .

El algoritmo continúa hasta completar i_{MAX} iteraciones o hasta que un subconjunto S que verifique $NFA(S, S') < 1$ es hallado. En caso de ser así, se realiza a continuación una optimización ORSA (Optimal RANdom Sample Algorithm) que se concentra en los inliers hallados para optimizar la detección.

Limitaciones de AC-RANSCAC

El algoritmo AC-RANSAC parte de una hipótesis nula que asume que los matches existentes son variables aleatorias mutuamente independientes, lo cual no es cierto para todos los casos posibles. De hecho, existen dos razones estructurales claras por las cuales los matches obtenidos a partir de descriptores ASIFT no siempre verifican esta suposición.

En primer lugar, consideremos el caso de múltiples correspondencias, en el cual un punto m de una imagen I puede tener varias correspondencias en la imagen I' y viceversa.

Esta propiedad presenta la ventaja de permitir la detección de varias ocurrencias de un objeto, o de objetos que presenten patrones repetitivos, como las texturas. Sin embargo, es también una clara contradicción a la hipótesis nula.

Para lidiar con esto, basta introducir un principio de maximalidad mediante el cual, en cada conjunto detectado S , se permite sólo una correspondencia por punto. De esta forma, para cada punto se selecciona aquella correspondencia que minimiza la rigidez α con respecto a la transformación $T_{S'}$ considerada.

El segundo fenómeno incompatible con la hipótesis nula es la existencia de matches redundantes. El algoritmo usado para computar los descriptores ASIFT suele representar algunas estructuras mediante varios puntos que difieren apenas en ubicación y escala, por lo cual generarán correspondencias no independientes.

Para lidiar con ello es necesario detectar los matches redundantes y eliminar los mismos. En primer lugar, se identifican los matches redundantes como aquellos que comparten un punto en una imagen mientras que sus correspondientes matches en la segunda imagen están lo suficientemente cerca.

Más formalmente, decimos que dos matches $c_i = (m_i, m'_i)$ y $c_j = (m_j, m'_j)$ se dicen redundantes si se verifica alguna de las siguientes propiedades:

- $m_i = m_j$ y $\|m'_i - m'_j\|_2 < \min\{\sigma_i, \sigma_j\}$
- $m'_i = m'_j$ y $\|m_i - m_j\|_2 < \min\{\sigma'_i, \sigma'_j\}$

donde σ_k representa la escala característica del punto de interés m_k .

Para descartar matches redundantes se opta por utilizar un criterio basado en la distancia entre descriptores calculada en la etapa de detección. De esta forma, cada vez que un match redundante es detectado se descarta aquel match c_j que presente la mayor distancia $DCEMD$. Esto es, para dos matches redundantes c_i y c_j , se descartará c_j en la medida que se verifique que $DCEMD(c_i) \leq DCEMD(c_j)$.

MAC-RANSAC para detección de múltiples grupos

Tanto RANSAC como AC-RANSAC solamente permiten detectar una transformación a partir de un conjunto de correspondencias. Dado que es necesario detectar múltiples transformaciones (esto es, varios grupos de correspondencias), se utilizará un enfoque de uso secuencial de RANSAC.

AC-RANSAC Secuencial

El algoritmo básico de RANSAC secuencial consiste en aplicar RANSAC en forma iterativa a un conjunto, sacando del mismo, en cada iteración, el conjunto detectado por RANSAC. El algoritmo se detiene cuando ya no pueden detectarse nuevos grupos.

Este enfoque presenta las siguientes limitaciones:

- **Detección de falsas transformaciones:** Validación de grupos compuestos de outliers.
- **Fusión de transformaciones cercanas:** En muchos casos dos o más transformaciones son detectadas como una única transformación.
- **Segmentación de una transformación:** Una única transformación es detectada como varias transformaciones. Muchas veces este fenómeno se produce al trabajar con una tolerancia espacial demasiado baja.
- **Transformaciones fantasmas:** Se produce al detectarse varias transformaciones espúreas producto del eco de una única transformación. Generalmente es debida a la detección de objetos con una estructura repetitiva (fachadas de edificios, texturas, etc).

Al combinar este enfoque con un esquema a contrario como el dado por AC-RANSAC se logran considerables mejoras. El criterio de validación de un grupo mediante su NFA aporta un buen criterio de parada para la detección secuencial. A su vez, el problema de sobre-segmentación se soluciona al contar con umbrales adaptivos que se setean en forma automática en base a las rigideces de los puntos del conjunto, sin necesidad de contar con ningún conocimiento a priori sobre los datos. En contraposición, los problemas de fusión de transformaciones y la existencia de transformaciones fantasmas no desaparecen al utilizar un enfoque a contrario, por lo cual deberán tomarse medidas para lidiar con ellos.

Filtrado de correspondencia espacial

La existencia de transformaciones fantasmas es debida, principalmente, a la auto-similitud de algunos objetos a detectar. De esta forma, se detectan varias transformaciones (que resultan de la existencia de correspondencias repetitivas que hacen eco a una única transformación real) y el objeto es detectado varias veces en poses diferentes.

Generalmente la transformación correcta es aquella que mejor describe al conjunto de puntos detectado y es detectada primero. A partir de ello, pueden descartarse las repeticiones de esta transformación. Se puede definir entonces un criterio para detectar correspondencias repetidas.

Consideremos un conjunto detectado S , siendo \mathbb{C} el conjunto de las restantes correspondencias (esto implica que $S \cap \mathbb{C} = \emptyset$). Una correspondencia $c_i = (m_i, m'_i)$ de \mathbb{C} se considera repetitiva con respecto a S si se cumplen las siguientes condiciones en forma simultánea:

- $\exists m \in S / \|m - m_i\|_2 < \min(\sigma, \sigma_i)$
- $\exists m' \in S / \|m' - m'_i\|_2 < \min(\sigma', \sigma'_i)$

donde σ_k es, nuevamente, la escala característica del punto m_k .

Mediante este criterio, cada vez que una nueva transformación es detectada y validada, las correspondencias del conjunto de las restantes correspondencias \mathbb{C} que son repetitivas con respecto al conjunto validado S son descartadas.

Detección de la fusión de varias transformaciones

Dado que AC-RANSAC muchas veces detecta varias transformaciones como una única transformación es necesario utilizar algún algoritmo que permita la detección de dichas fusiones.

Detección de la fusión de 2 grupos

Consideremos, en primer lugar, el caso de la fusión de 2 transformaciones en una única transformación detectada.

Sean S_0 un grupo de correspondencias, S_1 y S_2 dos subgrupos disjuntos de S_0 (esto es $S_1 \cup S_2 \subset S_0$ y $S_1 \cap S_2 = \emptyset$). Decimos que el conjunto S_0 resulta de la fusión de

los subconjuntos S_1 y S_2 si está bien descrito por alguna transformación detectada a la vez que las transformaciones T_1 y T_2 que describen a los subconjuntos S_1 y S_2 respectivamente son lo suficientemente diferentes.

La razón por la cual se podría llegar a validar al conjunto S_0 es que su NFA resulta menor que la de los subconjuntos S_1 y S_2 , mientras que estos dos subconjuntos presentan rigideces menores. Es decir,

$$\begin{cases} NFA(S_0, S'_0) < \min(NFA(S_1, S'_1), NFA(S_2, S'_2)) \\ \alpha(S_0, T_{S'_0}) \geq \max(\alpha(S_1, T_{S'_1}), \alpha(S_2, T_{S'_2})) \end{cases}$$

Este resultado se debe al comportamiento “avaro” de AC-RANSAC, que intenta detectar el mayor grupo que minimice NFA.

Para detectar la fusión de grupos, entonces, debe definirse un criterio para comparar el conjunto S_0 con los conjuntos S_1 y S_2 simultáneamente.

Consideremos un subconjunto $S_0(\subset \mathbb{C})$ ϵ -meaningful, con una transformación asociada $T_{S'_0}$. Si existen dos subconjuntos disjuntos S_1 y S_2 de S_0 , y también existen dos subconjuntos disjuntos S'_1 y S'_2 de $S_0 \setminus \{S_1 \cup S_2\}$ que verifiquen

- $NFA(S_1, S'_1) \leq \epsilon$ y $NFA(S_2, S'_2) \leq \epsilon$
- $NFA(S_1, S'_1) \times NFA(S_2, S'_2) \leq NFA(S_0, S'_0)$

entonces los dos subgrupos S_1 y S_2 son dos subconjuntos válidos de correspondencias, en lugar de un único conjunto válido S_0 .

Para poder aplicar este criterio, debe definirse una estrategia para dividir S_0 en forma óptima, así como un sistema recursivo que permita detectar la fusión de varias transformaciones.

División binaria recursiva en subgrupos

Se opta por identificar solamente uno de los subgrupos en un esquema de partición binaria como el anterior y aplicar esta estrategia en forma recursiva al subconjunto inicial S_0 . Definiremos, entonces, la estrategia de división de S_0 en dos subgrupos S_1 y S_2 .

Observemos, en primer lugar, que si S_1 y S_2 son dos subconjuntos de S_0 , y si S_1 es el más pequeño de ambos, entonces $\#S_1 \leq \#S_0/2 \leq \#S_2$.

Entonces el subconjunto S_1 se obtiene buscando (mediante muestreo aleatorio) un subconjunto $S'_1 \subset S_0$ que minimice $NFA(S_1, S'_1)$ bajo la restricción $\#S_1 \leq \frac{\#S_0}{2}$.

El subconjunto S_2 se obtiene en forma similar, buscando un subconjunto S'_2 que minimice $NFA(S_2, S'_2)$ entre las restantes correspondencias de S_0 .

Finalmente, el criterio anterior permite determinar si esta partición será considerada como válida o no.

Mientras la partición detectada sea validada, esta estrategia se continúa aplicando en forma recursiva al subconjunto $S_1^{(k)}$ identificado en la iteración previa k .

El proceso se detiene cuando el último subconjunto encontrado $S_1^{(k)}$ corresponde a una única transformación. Esto es, cuando no se pueden encontrar una partición $(S_1^{(k+1)}, S_2^{(k+1)})$ o si la misma no es validada.

Cabe destacar que al dividir S_0 en S_1 y S_2 el único grupo validado es S_1 . Dado que el subconjunto S_2 es ϵ -meaningful se lo considera como una detección válida y se pasa directamente a la etapa de optimización ORSA.

En la imagen 6 se ilustra el proceso de detección de subconjuntos para un conjunto S_0 formado por la fusión de 5 transformaciones.

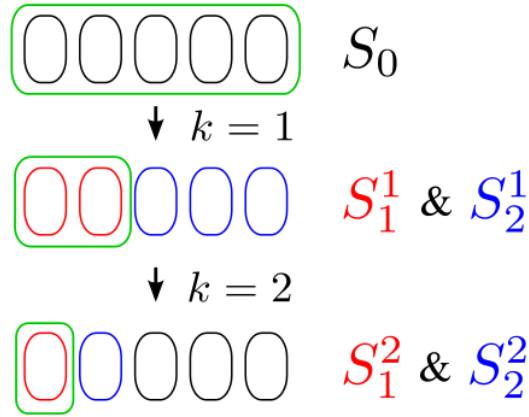


Figura 6: Esquema de detección de fusión de transformaciones

Algoritmo MAC-RANSAC

En base a todo lo explicado a lo largo de esta sección definiremos ahora el esquema general del algoritmo MAC-RANSAC.

Cabe destacar que el único parámetro seteable es la cantidad máxima de iteraciones i_{MAX} . El parámetro ϵ , que fija un umbral máximo sobre los valores de NFA , se fija a priori (como suele hacerse en los algoritmos con enfoque a contrario). Si bien dicho parámetro suele fijarse en 1, resultados empíricos demuestran que setear el mismo en 10^{-1} o 10^{-2} resulta ser más robusto. Esto determinará tanto un umbral de tolerancia espacial como un número mínimo de inliers para validar un conjunto (y su transformación asociada) detectado.

El algoritmo MAC-RANSAC puede ser resumido mediante el siguiente esquema:

Entradas: Conjunto de correspondencias no redundantes \mathbb{C} y número máximo de iteraciones i_{MAX}

Inicialización: $i := 0, \mathbb{S} := \emptyset$

1) Detección: Mientras $i < i_{MAX}$, muestreo aleatorio de $S' \subset \mathbb{C}$ y búsqueda de $S \subset \mathbb{C} \setminus S'$ que minimice $NFA(S, S')$

- Si $NFA(S, S')$, $(S_{opt}, S'_{opt}) := (S, S')$ y paso a **2)**

- De lo contrario $i := i + 1$

2) Optimización ORSA: Repetir $i_{MAX}/10$ veces:

Muestreo de S' en S_{opt} y búsqueda de $S \subset \mathbb{C} \setminus S'$ que minimice $NFA(S, S')$

- Si $NFA(S, S') < NFA(S_{opt}, S'_{opt}), (S_{opt}, S'_{opt}) := (S, S')$

3) Detección de fusión: Búsqueda de un subconjunto óptimo incluido en S_1

- Si se detecta una fusión, identificar dos subconjuntos ϵ -meaningful (S_1, S'_1) y (S_2, S'_2)
- De lo contrario $S_1 := S_1$ y $S_2 = \emptyset$

4) Filtrado Espacial: Descartar correspondencias de \mathbb{C} que sean repeticiones de S_1 .

5) Iteración: Agrego S_1 a la lista \mathbb{S} . Definición de $\mathbb{C} := \mathbb{C} \setminus S_1$. Reinicializo $i := 0$.

- Si $S_2 = \emptyset$ ir a **1)**
- De lo contrario $(S_{opt}, S'_{opt}) := (S_2, S'_2)$ e ir a **2)**

Salida: Lista de conjuntos disjuntos detectados \mathbb{S}

Algoritmo Implementado

A modo ilustrativo, se presenta un breve resumen del algoritmo implementado. La descripción detallada de los pasos del mismo fue realizada en forma exhaustiva en las secciones anteriores.

Descripción

El algoritmo implementado parte de dos imágenes con objetos comunes, en los cuales se quiere detectar dichos objetos en común.

En primer lugar se utiliza ASIFT para calcular los puntos clave, es decir, aquellos que resultan invariantes ante transformaciones afines. La implementación de ASIFT fue extraída de IPOL (http://www.ipol.im/pub/algo/my_affine_sift/) y utilizada sin realizarse ninguna modificación sobre ella.

Una vez calculados los puntos claves, se utiliza una implementación propia del algoritmo de Matching A Contrario para determinar las correspondencias entre dichos puntos claves.

Finalmente, habiendo determinado las correspondencias, se utiliza una implementación propia de MAC-RANSAC para agrupar dichas correspondencias en torno a transformaciones comunes, con lo cual cada grupo detectado estará asociado a una transformación dada. Esto permitirá diferenciar los distintos objetos así como las distintas ocurrencias de un mismo objeto.

Puntos Críticos

La criticidad del algoritmo implementado se encuentra principalmente en la etapa de matcheo.

El algoritmo de Matching A Contrario recorre en forma exhaustiva el espacio de características a lo largo de ambas imágenes, por lo cual el costo computacional del mismo es considerablemente alto.

Es esta etapa la que determina casi completamente el desempeño computacional del algoritmo. Si bien existen lenguajes de programación más adecuados a este tipo de aplicaciones (C, C++, entre otros) la implementación fue realizada en MatLab, el cual resulta ser un poco menos eficiente computacionalmente que dichos lenguajes. También es posible paralelizar varias de las tareas del algoritmo si se cuenta con varios procesadores e hilos de ejecución, pero esto no se hizo debido a restricciones de tiempo.

Dado que el algoritmo implementado resulto ser computacionalmente costoso, se opto por realizar pruebas con imágenes relativamente pequeñas (entre 100x100 y 200x200 pixeles). A partir de los resultados teóricos anteriores (en particular a partir de la ecuación 12), es fácil ver que cuanto más pequeña sea la imagen menor será la capacidad de detectar matches y establecer correspondencias entre los mismos. El tamaño de imagen seleccionado es lo suficientemente grande como para permitir verificar el correcto funcionamiento del algoritmo sin requerir excesivos tiempos de cómputo.

Un segundo inconveniente encontrado fue que, en ciertos casos, se produjeron problemas computacionales asociados a la existencia de matrices con un número de condición alto en ciertas partes del algoritmo. Esto da cierta inestabilidad numérica al algoritmo, por lo cual es posible que se produzcan resultados inexactos o se presenten errores de cómputo.

Existen técnicas para prevenir estos problemas numéricos, y si bien ninguna fue implementada en este trabajo, resulta claro que sería una mejora no sólo considerable, sino necesaria, si es que se desea utilizar el algoritmo como parte de un sistema de mayores dimensiones.

Resultados Obtenidos

Tal como se explicó anteriormente, los resultados presentados aquí fueron obtenidos considerando $\epsilon = 10^{-2}$, tanto para la etapa de matcheo como en el algoritmo MAC-RANSAC. Esta elección está basada principalmente en la mayor robustez que dicha elección da al algoritmo.

Dada la poca disponibilidad de imágenes de tamaño reducido disponibles para testear el algoritmo, se optó por generar una base de datos propia que se adapte a las necesidades de la etapa de validación de este trabajo. Para ello, se generaron imágenes de prueba a partir de aplicar alguna transformación afín aleatoria a una imagen base.

Matching A Contrario

Se presentan a continuación algunas imágenes mostrando los resultados obtenidos al aplicar el algoritmo de Matching A Contrario. En las mismas se colocaron marcadores sobre los puntos clave matcheados por el algoritmo.

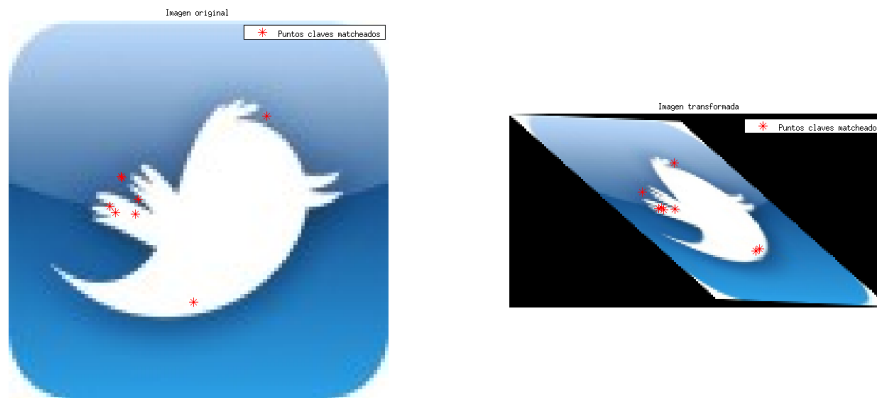


Figura 7: Resultado de Matching A Contrario 1



Figura 8: Resultado de Matching A Contario 2



Figura 9: Resultado de Matching A Contario 3

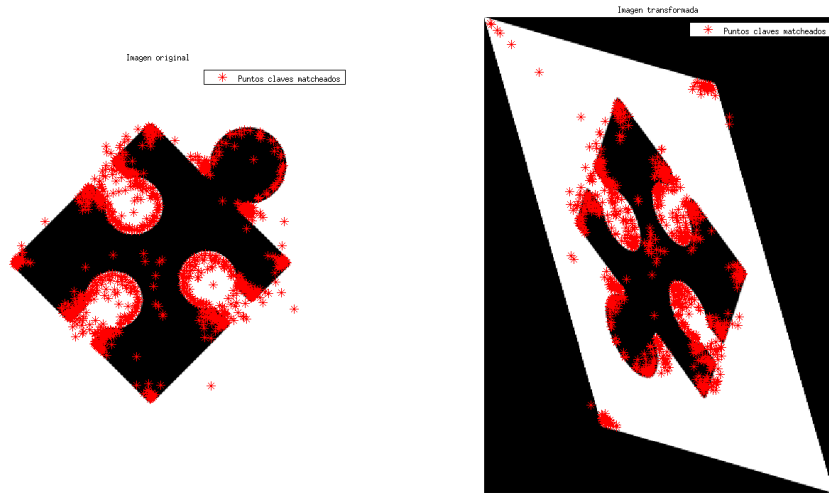


Figura 10: Resultado de Matching A Contario 4

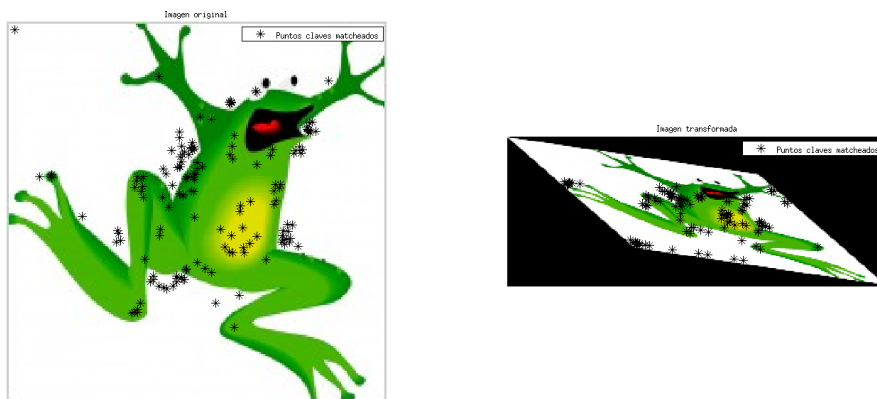


Figura 11: Resultado de Matching A Contario 5

Tal como se analizó durante el desarrollo teórico, el algoritmo implementado tiende a producir matches redundantes entre los puntos claves matcheados. A modo ilustrativo se incluye una segunda versión de los resultados presentados en 11 dónde se han eliminado esos matches redundantes (la eliminación de matches redundantes puede apreciarse con más claridad en la imagen transformada).

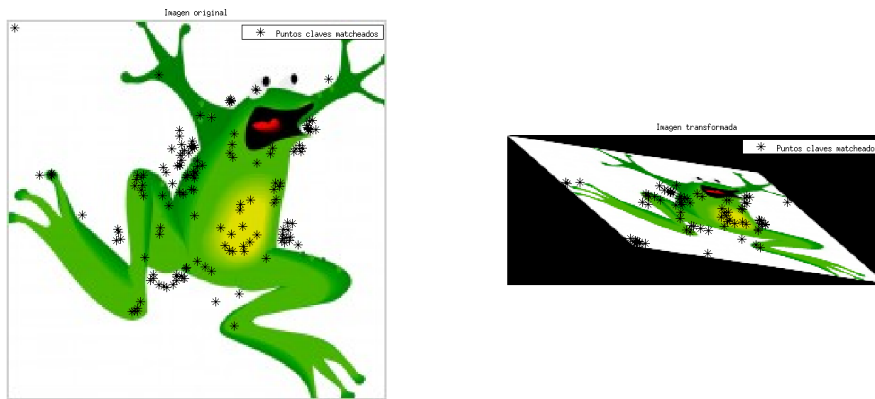


Figura 12: Resultado de Matching A Contario 5 sin correspondencias redundantes

MAC-RANSAC

El algoritmo MAC-RANSAC implementado presenta algunas diferencias con respecto al MAC-RANSAC presentado en [3]. La principal diferencia es que, a la hora de buscar el conjunto que minimiza el NFA para una transformación estimada, este no intenta incluir la mayor cantidad de puntos posibles en dicho conjunto. Por el contrario, esta versión de MAC-RANSAC termina determinando el menor conjunto que minimiza el NFA bajo la restricción $NFA < 1$.

Este enfoque presenta tanto ventajas como desventajas. En primer lugar, es claro que el problema de la segmentación de una transformación en varios conjuntos no siempre es resuelto mediante este enfoque. Sin embargo, presenta la ventaja de ser capaz de diferenciar dos transformaciones aún si las mismas resultan ser considerablemente iguales. El algoritmo MAC-RANSAC original sólo es capaz de detectar la fusión de transformaciones a partir de un nivel mínimo de diferenciación entre transformaciones.

El enfoque adoptado aquí, entonces, tendrá un mejor nivel de diferenciación entre transformaciones muy similares, a expensas de detectar una única transformación segmentada en varias transformaciones iguales, representadas a través de varios subconjuntos detectados en forma separada.

A continuación se presentan varias imágenes que reflejan los resultados obtenidos al aplicar el algoritmo. Nuevamente, se trabaja con imágenes generadas a partir de una imagen base y transformaciones afines aleatorias. Se colocan marcadores sobre los puntos claves de los conjuntos detectados en cada caso.

Prueba 1

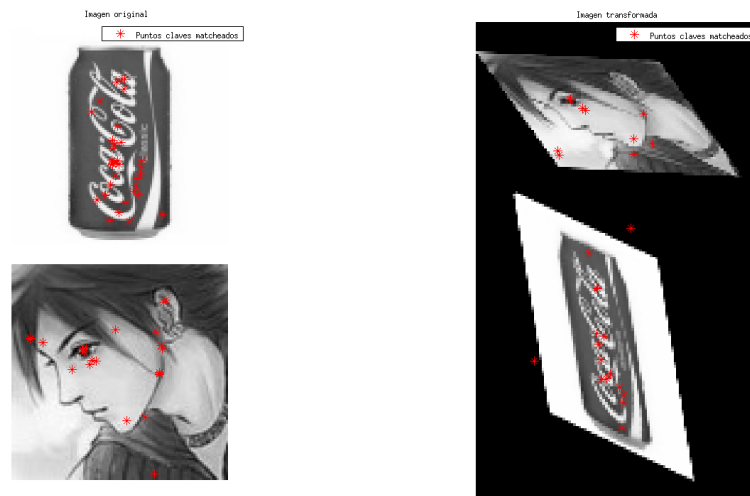


Figura 13: Total de puntos clave detectados

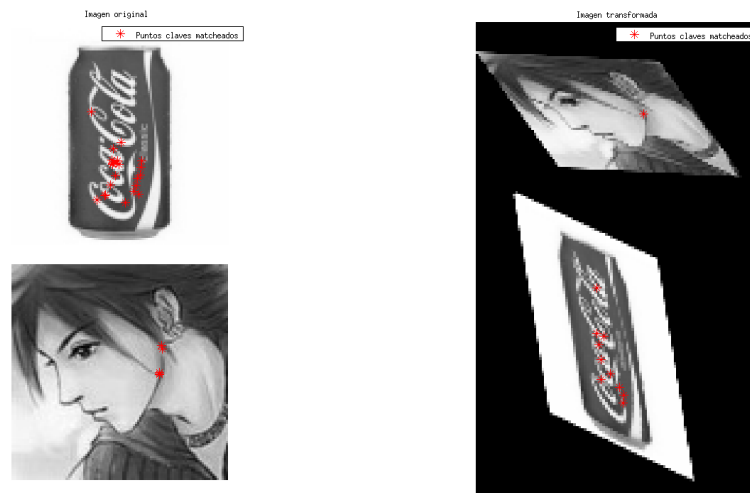


Figura 14: Conjunto detectado 1

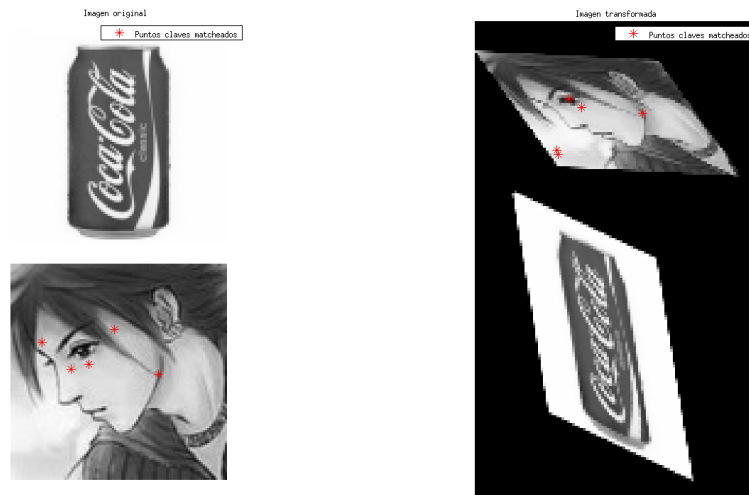


Figura 15: Conjunto detectado 2

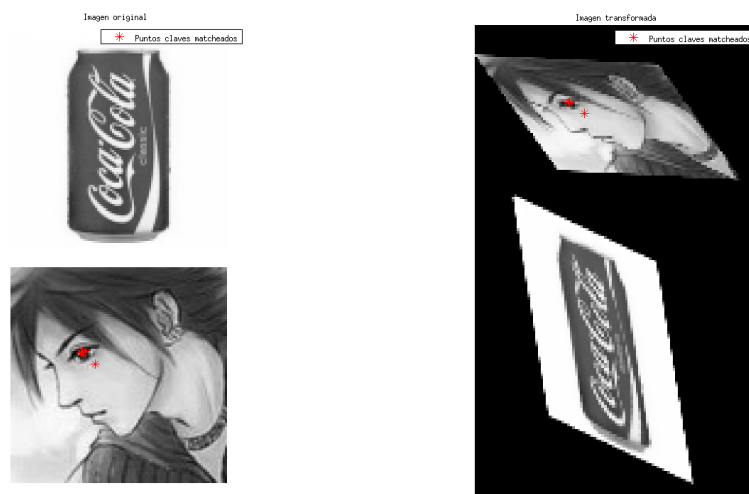


Figura 16: Conjunto detectado 3

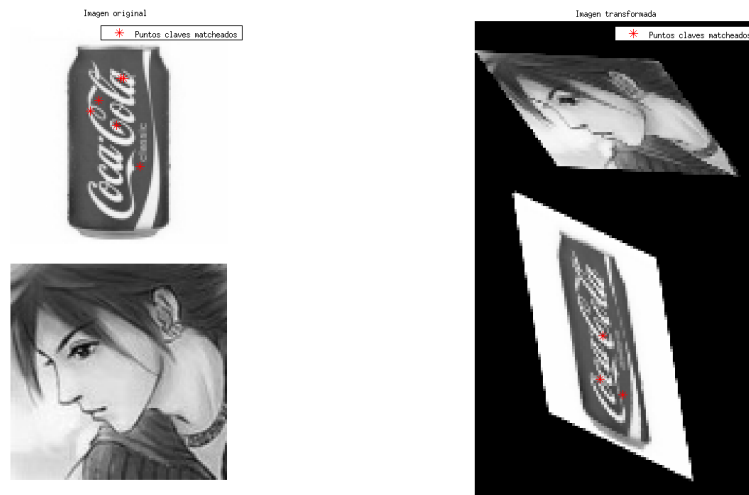


Figura 17: Conjunto detectado 4



Figura 18: Conjunto detectado 5

A partir de las imágenes anteriores, es claro que si bien el algoritmo detecta las dos transformaciones en forma segmentada, la diferenciación entre ambas transformaciones es clara, detectándose correctamente las correspondencias pertenecientes a cada una de las dos imágenes a detectar.

Prueba 2

Para la imagen utilizada en esta prueba se detectaron varios conjuntos para cada transformación involucrada. No se considera relevante presentarlos todos, por lo cual sólo se presentarán dos a modo ilustrativo.



Figura 19: Total de puntos clave detectados

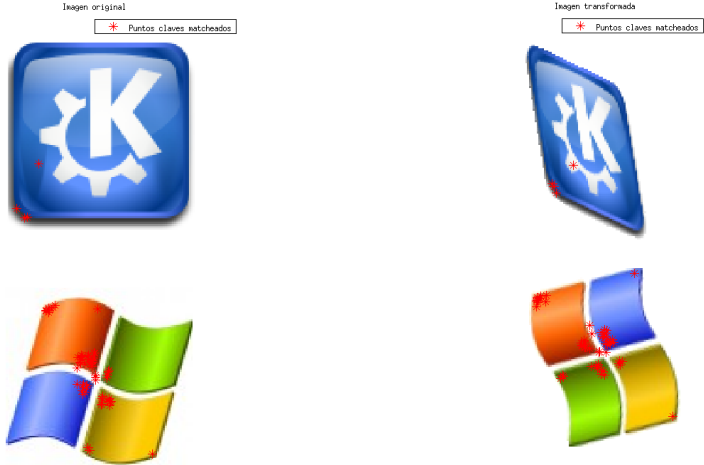


Figura 20: Conjunto detectado 1

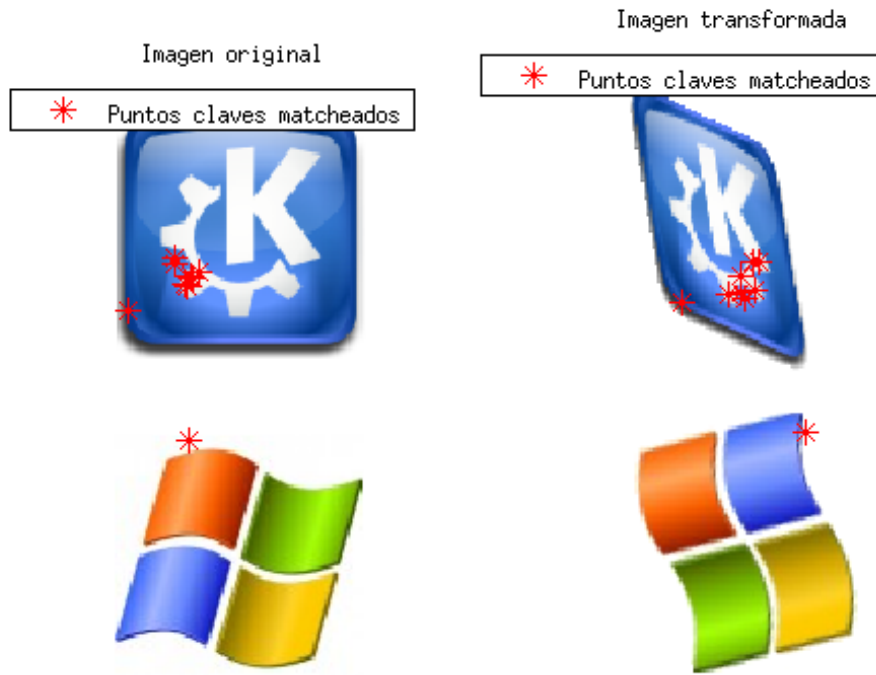


Figura 21: Conjunto detectado 2

De los resultados anteriores resulta claro que si bien tener un mayor nivel de diferenciación entre transformaciones muy similares resulta una característica deseable, el hecho de que una única transformación sea segmentada en varios conjuntos puede ser un inconveniente importante, sobre todo si se desea utilizar el algoritmo como parte de un sistema más complejo.

Conclusiones

De la sección anterior puede concluirse que, si bien el algoritmo de Matching A Contrario no siempre logra establecer todas las correspondencias en forma correcta, el desempeño del mismo resulta, en general, aceptable. Es claro que el correcto desempeño de dicha etapa brinda una base sólida la próximo paso del algoritmo.

El algoritmo MAC-RANSAC implementado parece funcionar en forma adecuada, detectando en forma correcta las transformaciones. En contraposición, haber sacrificado la capacidad de evitar la segmentación de transformaciones con el fin de tener una mayor capacidad de diferenciación entre transformaciones similares puede resultar una mala elección en una gran cantidad de contextos y aplicaciones.

Se concluye que, si bien el algoritmo implementado funciona correctamente, el mismo no resulta demasiado aplicable en gran cantidad de contextos debido a los inconvenientes de segmentación y costo computacional que presenta.

Se plantea la posibilidad de mejorar el presente algoritmo mediante la implementación de técnicas y medidas que permitan lidiar con los problemas presentados por el algoritmo (paralelización, utilización de otro lenguaje de programación, optimización y fusión de los conjuntos detectados para obtener transformaciones no segmentadas, etc).

También se plantea la posibilidad de estudiar la dependencia de los resultados obtenidos en función de algunos parámetros característicos internos al algoritmo implementado con el fin de analizar si una mejor elección de ellos podría traducirse en mejores resultados. En particular resulta interesante preguntarse si introducir una cota menor a 1 para el NFA durante la etapa de matcheo resultaría en un mejor desempeño del algoritmo.

Finalmente, se destaca que si se desea utilizar el algoritmo implementado en alguna aplicación concreta resulta claro que la mejora y optimización del mismo será no sólo un paso clave, sino uno fundamental.

Bibliografía

- [1] - ASIFT: An Algorithm for Fully Affine Invariant Comparison. G. Yu, J. Morel
- [2] - A Statistical Approach to the Matching of Local Features. J. Rabin, J. Delon, Y. Gousseau
- [3] - MAC-RANSAC: A robust algorithm for the recognition of multiple objects. J. Rabin, J. Delon, Y. Gousseau, L. Moisan
- [4] - The Earth Mover's Distance as a metric for image retrieval. Y. Rubner, C. Tomasi, L.J. Guibas
- [5] - Distinctive Image Features from Scale-Invariant Keypoints. D.G. Lowe
- [6] - A Unified Framework for Detecting Groups and Application to Shape Recognition. F. Cao, J. Delon, A. Desolneux, P. Musé, F. Sur
- [7] - Transportation Distances on the Circle. J.Rabin, J. Delon, Y. Gousseau
- [8] - An A Contrario Decision Method for Shape Element Recognition. P. Musé, F. Sur, F. Cao, Y. Gousseau, J.M. Morel
- [9] - Wikipedia: RANSAC (<http://en.wikipedia.org/wiki/RANSAC>)
- [10] - Wikipedia: Scale-invariant feature transform (http://en.wikipedia.org/wiki/Scale-invariant_feature_transform)