



UNIVERSIDAD AUTÓNOMA DE YUCATÁN

FACULTAD DE MATEMÁTICAS



Seguimiento de objetos usando el método de agrupamiento K-Means y la técnica de retroproyección de histograma

Reporte

Asignatura: Procesamiento de Video

Profesor: Dr. Francisco Javier Hernández López

Integrantes del equipo:

Andrés Ulloa de la Torre

Joaquín Martínez

Braian Jesús Uc Cayetano

11/12/2018

Índice

Introducción.....	3
Descripción del método	3
Centroide del momento de la imagen	4
Algoritmo de retroproyección de histograma	4
Segmentación con Fast K-Means.....	6
Coeficiente de Bhattacharyya.....	7
Mean Shift	7
Conclusiones y observaciones	9
Referencias	12

Introducción

Para la elaboración de la entrega se solicitó que dado un video con un objeto de interés; se aplicase un método de seguimiento de objetos basado en la técnica Mean Shift usando la retroproyección de histograma.

Para la implementación del código se empleó el lenguaje de programación Python en conjunción con las bibliotecas OpenCV y Numpy.

Descripción del método

El método se compone de una serie de pasos, en donde, cada uno, corresponde a un cálculo diferente y dependen del paso anterior para poder progresar. El proceso seguido por el algoritmo es el siguiente:

1. Se obtiene una imagen y se calcula el centroide del momento de la imagen.
2. Se procede a realizar la segmentación usando el algoritmo K-Means.
3. Se realiza la retroproyección del histograma de la imagen.
4. Se utiliza el coeficiente de Bhattacharyya para calcular la distancia normalizada entre los histogramas candidato y objetivo (*target*).
5. Se utiliza el método Mean Shift para calcular los cambios en el centroide del objetivo y la posición en la que se encuentra.
6. Se actualizan los parámetros de la imagen para el siguiente *frame*.



A continuación, se explica más a detalle cada uno de los pasos con los cálculos realizados.

Centroide del momento de la imagen

El primer paso consiste en calcular el centroide del momento de la imagen. Debido a que el algoritmo intenta mostrar los resultados del seguimiento del objeto en tiempo real, se calcularán únicamente los centroides para los momentos de orden 0 y 1 de la imagen. Tomando en cuenta que la fórmula general para el cálculo de momentos de una imagen es la siguiente:

$$M_{ij} = \int_S x^i y^j I(x, y) dx dy$$

Entonces, se pueden usar sumatorias que iteren sobre todos los pixeles de la imagen para realizar el cálculo de los momentos requeridos:

$$M_{00} = \sum_x \sum_y x^0 y^0 I(x, y) = \sum_x \sum_y I(x, y) \leftarrow \text{Momento de orden 0}$$

$$M_{10} = \sum_x \sum_y x^1 y^0 I(x, y) = \sum_x \sum_y x I(x, y) \leftarrow \text{Momento de orden 1 para } x$$

$$M_{01} = \sum_x \sum_y x^0 y^1 I(x, y) = \sum_x \sum_y y I(x, y) \leftarrow \text{Momento de orden 1 para } y$$

Donde $I(x, y)$ es la intensidad del pixel, x y y son las coordenadas de la imagen. Una vez calculados los momentos requeridos, se puede obtener el centroide de la imagen; para ello, se utiliza la siguiente fórmula:

$$x_c = \frac{M_{10}}{M_{00}}, y_c = \frac{M_{01}}{M_{00}}$$

Con ello se obtienen las coordenadas que corresponden al centro de la imagen o de la región en evaluación. Con ellas, puede procederse a la retroproyección del histograma.

Algoritmo de retroproyección de histograma

Con las fórmulas para calcular los momentos de la imagen de órdenes 0 y 1, se pueden calcular los momentos para el centro de masa; con estos nuevos momentos obtenidos, se puede deducir la fórmula que se usará para realizar la retroproyección de histograma.

Para el cálculo de momentos para el centro de masa, se utiliza la siguiente fórmula:

$$\mu_{pq} = \sum_{x=1}^N \sum_{y=1}^M (x - x_c)^p (y - y_c)^q f(x, y)$$
$$\text{con } x_c = \frac{M_{10}}{M_{00}}, y_c = \frac{M_{01}}{M_{00}}$$

Como se puede notar, utiliza los valores del centroide calculado en el paso anterior. Y con esa fórmula, se pueden generar los momentos para el centro de masa de orden 1 para x y y , orden 2 para x y orden 2 para y . En las siguientes ecuaciones se utiliza la misma integral con la que se pudieron sacar los momentos de la imagen, pero la diferencia está en la resta que se realiza después del cálculo del cociente con los momentos, en la cual se multiplican las x y las y por los números que representan a los momentos; por ejemplo, en μ'_{11} , se resta $x_c y_c$ pues es orden 1 para x y orden

1 para y ; en μ'_{20} , se multiplica x_c al cuadrado debido a que es de orden 2 para x y no se usa y_c pues es el momento de orden 0 de y .

Ahora, se procede a calcular los histogramas del *target* y de la imagen actual. Los cálculos de estos histogramas deben realizarse en imágenes a color, es decir, que se compongan de los canales RGB y no sean de tipo *grayscale*.

Para comenzar se declaran las siguientes variables para los histogramas calculados:

$$\begin{aligned} H_T &\rightarrow \text{Histograma del target} \\ H_F &\rightarrow \text{Histograma del frame} \end{aligned}$$

Conociendo esto, se procede mediante el siguiente proceso:

1. Para cada bin j del histograma hacer lo siguiente:

$$R(j) = \frac{H_T(j)}{H_F(j)} \text{ con } j = 0, \dots, no_bins$$

Este cálculo nos dice qué tan importante es un *target* en una cierta imagen, y la cantidad designada para j varía de acuerdo a las necesidades.

2. Para cada píxel \vec{x} hacer:

$$B(\vec{x}) = \min \left(R \left(bin(f(\vec{x})) \right), 1 \right)$$

3. Se calcula $\hat{B} = B \otimes D^r$

$$\text{con } D^r(\vec{x}) = \begin{cases} 1 & \text{si } \sqrt{x^2 + y^2} < r \\ 0 & \text{en otro caso} \end{cases}$$

Ahora, los picos en \hat{B} indican la posición del objeto (*target*).

4. Finalmente, se calcula la posición $\vec{x}' = \arg \max_{\vec{x}} (\hat{B}(\vec{x}))$ para $\vec{x} \in \Omega$

Ahora, se buscará la fórmula que permitirá encontrar el largo y el ancho de la caja que envolverá al objeto y que nos permita verificar la robustez de este algoritmo para el seguimiento de los objetos. Para encontrar la fórmula, se deben acomodar los momentos para el centro de masa de orden 1 y de orden 2 para x y y de la siguiente manera:

$$Cov(R_T) = \begin{bmatrix} \mu'_{20} & \mu'_{11} \\ \mu'_{11} & \mu'_{02} \end{bmatrix}$$

A continuación, se procede con su resolución utilizando las posiciones de los valores en la matriz:

$$\begin{aligned} A\vec{v} &= \lambda\vec{v} \\ A\vec{v} - \lambda\vec{v} &= 0 \\ (A - \lambda I)\vec{v} &= 0 \\ \det(A - \lambda I) &= 0 \\ A &= \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \\ \det \begin{pmatrix} a_{11} - \lambda & a_{12} \\ a_{21} & a_{22} - \lambda \end{pmatrix} &= 0 \\ a_{11}a_{22} - a_{11}\lambda - a_{22}\lambda + \lambda^2 - a_{12}a_{21} &= 0 \\ \lambda^2 - (a_{11} + a_{22})\lambda + a_{11}a_{22} - a_{12}a_{21} &= 0 \end{aligned}$$

Finalmente, se obtiene una ecuación de segundo grado, que puede resolverse utilizando la fórmula general de las ecuaciones:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Sustituyendo a , b y c con los valores obtenidos en la ecuación de la matriz de covarianza, se procede de la siguiente manera:

$$\lambda_{1,2} = \frac{(a_{11} + a_{22}) \pm \sqrt{(a_{11} + a_{22})^2 - 4(a_{11}a_{22} - a_{12}a_{21})}}{2}$$

$$\lambda_{1,2} = \frac{(a_{11} + a_{22}) \pm \sqrt{(a_{11} - a_{22})^2 + 4a_{12}a_{21}}}{2}$$

Ahora, se sustituyen nuevamente las posiciones con los valores correspondientes en la matriz de covarianza:

$$\lambda_{1,2} = \frac{(\mu'_{20} + \mu'_{02}) \pm \sqrt{(\mu'_{20} + \mu'_{02})^2 - 4\mu'^2_{11}}}{2}$$

$$\Rightarrow l = 2\sqrt{\lambda_1}, w = 2\sqrt{\lambda_2}$$

Tras la resolución de la ecuación, se obtienen los valores l (largo) y w (ancho) de lo que será el *bounding box* o caja que encerrará al objeto y mostrará visualmente la exactitud del algoritmo para calcular su posición en la imagen.

Segmentación con Fast K-Means

La idea de este algoritmo es tratar de acelerar el algoritmo de K-Means.

1. Calcular el histograma $h(r)$, $0 \leq r \leq L$, con L el número de bins con el que se normaliza la imagen (Si son 8 bins, entonces $L = 7$).
2. Fast K-means for gray image
 - 2.1. $t = 0$
 - 2.2. Para $r = 1, 2, \dots, L$ hacer:
 - 2.3. Para $k = 1, 2, \dots, K$ hacer:
 - 2.4. $d(r, c_k^{(t)}) = |r - c_k^{(t)}|$ o también $(r - c_k^{(t)})^2$
Fin Para 2.3.
 - 2.5. $\tilde{k} = \arg \min_k [d(r, c_k^{(t)})]$
 - 2.6. $c_k^p = c_k^p + r \times h(r) \rightarrow$ Niveles de gris total del \tilde{k} -ésimo clúster
Para imágenes a color: $\vec{c}_k^p = \vec{c}_k^p + \vec{r} \times h(\vec{r})$
 - 2.7. $c_k^N = c_k^N + h(r) \rightarrow$ Número de pixeles total del \tilde{k} -ésimo clúster
Para imágenes a color: $\vec{c}_k^N = \vec{c}_k^N + 1$, que sería $\vec{c}_k^N = \vec{c}_k^N + h(r_1, r_2, r_3)$
Fin Para 2.2.
 - 2.8. $t = t + 1, flag = 0$
 - 2.9. Para $k = 1, 2, \dots, K$

- 2.10. $c_k^{(t)} = \frac{c_k^p}{c_k^N}$
- 2.11. Calcular $\delta(c_k^{(t)}, c_k^{(t-1)}) = |c_k^{(t)} - c_k^{(t-1)}|$
 Para imágenes a color: $\|\vec{c}_k^{(t)} - \vec{c}_k^{(t-1)}\|_2^2$
- 2.12. Si $\delta(c_k^{(t)}, c_k^{(t-1)}) \geq \delta T$, hacer
 $flag = 1$
 Fin Si
- Fin Para
- 2.13. Si ($flag == 1$) hacer:
 Ir al paso 2.2.
 Fin Si
- 2.14. De lo contrario, hacer:
 $c_k^{(op)} = c_k^{(t)} \forall k = 1, \dots, k$
- 2.15. *Output* ($c_k^{(op)}$)
- 2.16. Fin

Para notar un poco más la diferencia entre el Fast K-Means y el K-Means “clásico”, se presenta a continuación, la complejidad de ambos algoritmos:

K-means $O(t \times N \times M \times k)$	Fast K-means $O(t \times L + 1 \times k)$
--	--

Coeficiente de Bhattacharyya

El coeficiente de Bhattacharyya se utiliza como medida de similitud sin considerar los pixeles de *foreground* y *background*, sino que solo se usará un target. Entonces, se utilizará el Mean Shift para visualizar la dirección hacia donde se va a recorrer el cuadro delimitador; y éste, se alimentará de los datos del mapa de la retroproyección del histograma.

La fórmula para calcular el coeficiente de Bhattacharyya es la siguiente:

$$C_{Bha}(p_c^i, q) = \sum_{u=1}^m \sqrt{p_{cu}^i \times q_u}$$

Donde:

- p_c^i y q son los vectores de características del *target* candidato i y del *target* modelo, respectivamente.
- m es la dimensión del histograma
- u es el índice de cada bin

El cálculo de este coeficiente se realiza en dos ocasiones durante la ejecución del algoritmo Mean Shift, en la siguiente sección se mostrarán las partes en las que se utiliza.

Mean Shift

El algoritmo Mean Shift requiere de las siguientes representaciones:

Sea $\{\vec{x}_i^*\}_{i=1,\dots,n} \rightarrow$ posiciones del pixel normalizados en la región target

Sea $\{\vec{x}_i\}_{i=1,\dots,n_h} \rightarrow$ posiciones del pixel normalizados en la región candidata

La representación del modelo del *target* se muestra a continuación:

$$\begin{aligned}\hat{q} &= \{\hat{q}_u\}_{u=1,\dots,m} \quad \sum_{u=1}^m \hat{q}_u = 1 \\ \hat{q}_u &= C \sum_{i=1}^n k(\|\vec{x}_i^*\|^2) \delta[b(\vec{x}_i^*) - u], \\ \text{con } C &= \frac{1}{\sum_{i=1}^n k(\|\vec{x}_i^*\|^2)} \\ b(\vec{x}_i) &\rightarrow \text{valor del pixel } [1, \dots, m]\end{aligned}$$

Mientras que la representación del modelo candidato, es la siguiente:

$$\begin{aligned}\hat{p}(\vec{y}) &= \{\hat{p}_u(\vec{y})\}_{u=1,\dots,m} \quad \sum_{u=1}^m \hat{p}_u = 1 \\ \hat{p}_u(\vec{y}) &= C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{\vec{y} - \vec{x}_i}{h}\right\|^2\right) \delta[b(\vec{x}_i) - u], \\ \text{con } C_h &= \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{\vec{y} - \vec{x}_i}{h}\right\|^2\right)}\end{aligned}$$

Dado el modelo del target $\{\hat{q}_u\}_{u=1,\dots,m}$ y su posición inicial \hat{y}_0 en el frame $t - 1$, hacer lo siguiente:

1. Inicializar la posición del target en el frame actual con \hat{y}_0 , calcular $\{\hat{p}_u(\hat{y}_0)\}_{u=1,\dots,m}$ y evaluar:

$$\rho[\hat{p}(\vec{y}_0), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\vec{y}_0) \hat{q}_u} \quad \text{Coeficiente de Bhattacharyya}$$

Este es el primer paso en el que se utiliza el coeficiente de Bhattacharyya, y se usa como una medida de similitud entre el *target* y el candidato.

2. A continuación, se calculan los pesos $\{w_i\}_{i=1,\dots,n_h}$, utilizando la siguiente fórmula:

$$w_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\vec{y}_0)}} \delta[b(\vec{x}_i) - u]$$

3. Se calcula la nueva posición del *target* candidato:

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{\vec{y} - \vec{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\vec{y} - \vec{x}_i}{h}\right\|^2\right)}$$

Con $g(z) = -k'(z)$, es decir, la derivada negativa del kernel k

4. Calcular $\{\hat{p}(\vec{y}_1), \hat{q}\}_{u=1,\dots,m}$ y evaluar:

$$\rho[\hat{p}(\vec{y}_1), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\vec{y}_1) \hat{q}_u}$$

En esta parte se utiliza por segunda vez, el coeficiente de Bhattacharyya; en este caso, se usa para evaluar el coeficiente del centro de masa y compararlo con el centro de masa obtenido antes de evaluar el primero.

5. Mientras $\rho[\hat{p}(\vec{y}_1), \hat{q}] < \rho[\hat{p}(\vec{y}_0), \hat{q}]$:

a. Actualizar $\vec{y} \leftarrow \frac{1}{2}(\vec{y}_0 + \vec{y}_1)$

b. Evaluar $\rho[\hat{p}(\vec{y}_1), \hat{q}]$

6. Si $\|\vec{y}_1 - \vec{y}_0\|_2 < \epsilon$, entonces **parar**.

De lo contrario:

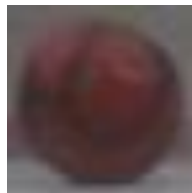
a) Actualizar $\vec{y}_0 \leftarrow \vec{y}_1$

b) Ir al **paso 2**

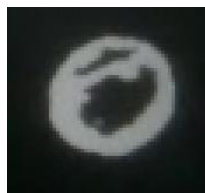
Conclusiones y observaciones

Como base para el desarrollo de las conclusiones se parte con el hecho de que, el nombre de la técnica da cabida a interpretaciones ambiguas, pues si bien, los mapas de probabilidades generados por la retroproyección de *frames* filtrados con la técnica de K-Means poseen menos falsas incidencias de “altas probabilidades” del ROI en el frame en el que se busca éste, dichas mejoras no se trasladan del todo a un *tracking* estable, pues el filtrado con k-medias actúa como un “bottleneck” para información importante de la escena procesada dándose en numerosas veces el caso de que los juegos de sombras en una escena eran agrupadas por k-medias como intensidades mucho más oscuras que la contraparte no ensombrecida de la misma intensidad.

Cómo ejemplo considerar los juegos de sombras presentes en el objeto rastreado:



Notar cómo el balón presenta tonalidades más oscuras en el centro debido a las sombras proyectadas como consecuencia de recibirse iluminación directa desde el techo del escenario. Dicha diferencia en intensidades provocando que el mapa de probabilidades generado por la retroproyección tuviese el siguiente aspecto para la región correspondiente al sitio en el que el objeto se encuentra:



Perjudicando por esta razón a los centros de masa propuestos por mean shift, pues el punto central del balón se encuentra apagado.

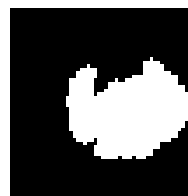
Se destaca además que, el grosor del anillo “prendido” es proporcional a la cantidad de k 's definidas siendo entonces el anillo de la imagen mostrada con anterioridad el correspondiente a $k = 50$, esto necesariamente significando que para conseguir la “suficiente masa” necesaria como para generar *shifts* en el objeto hacía falta tomar un penalty de rendimiento en lo que a tiempo respecta, pues el incremento en centroides de K-Means vuelve computacionalmente más costoso cada ciclo de *track* efectuado por el algoritmo.

Dicho lo anterior, se tiene que la técnica hace un *trade-off* entre la información provista por las intensidades de la escena y el que tan “limpio” esté el mapa de probabilidades generado por la retroproyección. Pues, si se da el caso de que los colores en el objeto de interés no estén completamente homogeneizados, entonces es posible que éstos sean afectados por la segmentación del paso de filtrado con K-Means provocando escenarios como el que hallamos con el video con el que se elaboraron las pruebas.

Se agrega que es posible obtener un mejor rendimiento al momento de generar el mapa de retroproyección si se procesa manualmente el video con herramientas de edición que homogenizan la imagen para retirar las sombras; esto siendo prácticamente equivalente a segmentar la imagen manualmente. Para esto, se aplicó un filtro de “caricaturización” al ROI que, en consecuencia, generó cambios dramáticos en el rendimiento del seguimiento, pues no solo se pudo reducir la cantidad de centroides en el filtrado a 15 (aumentando la velocidad de ejecución), sino que también se obtuvieron mapas de probabilidad mucho más “limpios” sin huecos “apagados” en el centro del objeto de interés:



ROI caricaturizado:



Retroproyección del ROI caricaturizado sobre el frame target:

Importante además es considerar que el paso de pre-filtrado involucrando la segmentación con K-Means reduce de manera importante el ruido presente en el mapa de probabilidades.

Retroproyección empleando pre filtrado con K-Means ($k = 15$) (sin el ROI caricaturizado):



Retroproyección empleando pre filtrado sin K-Means (sin el ROI caricaturizado):



Como conclusión final, se asevera qué la técnica de seguimiento de objetos con kernel y pre filtrado con k medias ofrece un rendimiento razonable siempre y cuando el objeto de interés no esté contaminado con efectos de sombras.

Referencias

Khalid MS, Ilyas MU, Sarfanaz MS, Ajaz MA (2006) Bhattacharyya coefficient in correlation of gray-scale objects. *J Multimed* 1:56–61

Leichter I, Lindenbaum M, Rivlin E (2010) Mean shift tracking with multiple reference color histograms. *Comput Vis Image Und* 114:400–408

Li J, Li X, Tao D (2005) KPCA for semantic object extraction in images. *Pattern Recogn* 41:3244–3250

Lin C, Chen C, Lee H, Liao J (2014) Fast k-means algorithm based on a level histogram for image retrieval. *Expert Syst Appl* 41:3276–3283

Lu N, Freng Z (2008) Mathematical model of blob matching and modified Bhattacharyya coefficient. *Image Vis Comput* 26:1421–1434

Sim KS, Tso CP, Tan YY (2007) Recursive sub-image histogram equalization applied to gray scale images. *Pattern Recogn Lett* 28:1209–1221

Wang F, Yu S, Yang J (2010) Robust and efficient fragments-based tracking using mean shift. *AEU-Int J Electron C* 64:614–623

Firat Hardalaç, Uğurhan Kutbay, İsa Şahin, Anıl Akyel (2017). A novel method for robust object tracking with K-means clustering using histogram back-projection technique. *Springer* 77:24059–24072