

Fundamentos de Visión por Computador

Prof. Carlos Cerrada Somolinos

Informe elaborado por: Rafael Mártony

Septiembre 2009

Índice

1.	Introducción	3
2.	Sistema de detección de incendios	4
2.1.	Detección de movimiento lento de objetos	4
2.2.	Detección de regiones de color del humo	5
2.3.	Detección de regiones que se elevan.....	5
2.4.	Detección de sombras y su eliminación	6
3.	Conclusiones.....	7
4.	Referencias	9
Ilustración 1 - Detección de regiones de movimiento lento		8
Ilustración 2 - Segunda versión del prototipo		8

1. Introducción

La detección temprana y la rápida lucha contra incendios forestales son imprescindibles hoy en día para impedir víctimas, la devastación o destrucción de flora y fauna, y disminuir los daños materiales.

El calor solar, la deshidratación de plantas e imprudencias humanas pueden ser el comienzo de un incendio que si no se detecta a tiempo puede volverse incontrolable. El hecho de que muchas zonas no cuenten con fácil acceso agrava aún más la situación y por lo tanto la extinción del fuego resulta una tarea muy riesgosa y complicada para los bomberos.

Principalmente es posible detectar incendios forestales mediante imágenes satelitales o bien mediante torres o lugares estratégicos donde es posible ver a simple vista fuego o columnas de humo.

Las imágenes satelitales no son fáciles de obtener (se necesitan satélites de órbita baja) y la frecuencia de actualización de dichas imágenes puede ser insuficiente para que sea efectiva la detección. Además, se debe analizar no sólo el espectro visible de luz sino también distintas bandas espectrales para que sea posible la detección aún cuando hay nubes o elementos que impidan la correcta visibilidad.

Actualmente existen proyectos como MODIS [1] (abreviado de **M**oderate Resolution **I**maging **S**pectroradiometer, en inglés) de NASA y aplicaciones como “Web fire mapper” (<http://firefly.geog.umd.edu/firemap/>) que permiten determinar dónde y cuándo se producen incendios en prácticamente cualquier parte del planeta durante el día y en algunas regiones durante la noche.

MODIS es un sistema que permite proveer imágenes satelitales en tiempo cercano a real, logrando localizar un incendio unas horas después de que el satélite vuelva sobre él.

Dados los datos de MODIS se podrían utilizar por ejemplo modelos de redes neuronales para detectar humo o fuego. Existen algunos trabajos realizados [2] [3] que tienen en general más de un 50% de efectividad.

El presente trabajo se centra en cómo resolver la detección de incendios mediante la utilización de torres de vigilancia en donde deberán instalarse cámaras de vídeo conectadas a un ordenador.

Las torres de vigilancia se deben instalar en lugares adecuados cubriendo la mayor área posible estando en lugares elevados.

La localización y distribución geográfica de estas torres puede realizarse metodológicamente con la ayuda de la utilización de SIGs logrando una mayor eficiencia utilizando la cantidad mínima de torres. Se puede ver un trabajo concreto [4] en ese sentido.

La presente solución requiere un ordenador normal y cámara de vídeo fija, sin requerimientos especiales.

También se podrían utilizar cámaras profesionales que permiten mover la cámara en sus distintos ejes y realizar acercamientos o alejamientos. Este tipo de cámaras son ideales ya que se encuentran aisladas cubiertas por una cúpula y son resistentes a diversas situaciones climáticas.

El sistema permitiría la detección de fuego durante el día pero sería posible modificarlo para que permitiera la detección durante la noche.

2. Sistema de detección de incendios

La detección de fuego se realiza en el rango visible de una cámara de vídeo por lo tanto se debe primero detectar movimiento de objetos en el vídeo (1), luego la detección de regiones con colores de humo (2), detección de objetos ascendentes (3) y por último detección de sombras y eliminación de regiones con sombras (4).

Estos cuatro algoritmos darían como resultado valores reales $D_1(x,y,n)$, $D_2(x,y,n)$, $D_3(x,y,n)$, $D_4(x,y,n)$ pertenecientes a $[-1, 1]$, y cuanto más cercano a uno se encuentren, más confidente será el algoritmo (es decir, existe una alta probabilidad de presencia de fuego).

Es posible combinar linealmente los resultados de estos cuatro algoritmos y darle diferente peso a cada uno de ellos para lograr mejores resultados.

2.1. Detección de movimiento lento de objetos

Asumiendo que la cámara está fija, se deberá analizar el vídeo para detectar regiones que se encuentran en movimiento.

Es necesario separar elementos que están en el primer plano de los que pertenecen al fondo de una imagen. Un aproximación sencilla para estimar el fondo de la imagen es tomando un promedio de los marcos (frames) observados en el vídeo.

Como los objetos en movimiento y las regiones sólo ocupan una parte de la imagen, ellas ocultan parte de la escena del fondo y su efecto es cancelado en el tiempo al promediar.

Existe numerosa literatura [5] [6] acerca de los diferentes métodos de detección de movimiento.

Como en este caso se quiere realizar un sistema de detección en tiempo real, se implementa un algoritmo que no resulta tan costoso en cuanto a recursos informáticos. Este algoritmo [5] fue presentado en la reunión internacional de actualidad sobre robótica de la universidad de Carnegie Mellon. En ese documento (*System for Video Surveillance and Monitoring* (VSAM)), se utiliza un método recursivo de estimación del fondo de la imagen que veremos como funciona.

Sea $I_n(x,y)$ el valor de la intensidad (brillo) en la posición (x,y) de la n -ésima imagen I_n .

El valor estimado de intensidad del fondo de la imagen en la misma posición, $B_{n+1}(x,y)$, se calcula como:

$$B_{n+1}(x,y) = \begin{cases} aB_n(x,y) + (1-a)I_n(x,y), & \text{si } I_n(x,y) \text{ no está en movimiento} \\ B_n(x,y), & \text{si } I_n(x,y) \text{ está en movimiento} \end{cases} \quad (2.1)$$

donde $B_n(x,y)$ es la anterior estimación de la intensidad del fondo de la imagen en la misma posición del píxel.

El valor de actualización a es un número real positivo cercano a uno.

Para saber si un píxel está en movimiento se debe evaluar la siguiente desigualdad

$$|I_n(x,y) - I_{n-1}(x,y)| > T_n(x,y) \text{ y } |I_{n-1}(x,y) - I_{n-2}(x,y)| > T_n(x,y) \quad (2.2)$$

donde $T_{n+1}(x,y)$ se calcula de la siguiente manera

$$T_{n+1}(x,y) = \begin{cases} aT_n(x,y) + (1-a)(c|I_n(x,y) - B_n(x,y)|), & \text{si } I_n(x,y) \text{ no está en movimiento} \\ T_n(x,y), & \text{si } I_n(x,y) \text{ está en movimiento} \end{cases} \quad (2.3)$$

$T_0(x,y)$ se determina experimentalmente. El valor c es un valor real mayor que uno y cuanto mayor sea éste, menor será la sensibilidad del detector. En este trabajo se utilizó c con valor 5.

Inicialmente todos los píxeles se asumen que no están en movimiento.

Según [7], para la detección de movimiento lento de objetos se utilizan dos imágenes del fondo $B_{fast}(x,y,n)$ y $B_{slow}(x,y,n)$ que se renuevan con diferentes tasas de actualización, donde (x,y,n) es la posición del píxel en la imagen en el instante de tiempo n . B_{fast} se actualiza en cada marco y B_{slow} se actualiza cada un segundo, utilizando el valor $a = 0,7$ y $0,9$ respectivamente (a aparece en (2.1) y (2.3)).

Inicialmente $B_{fast}(x,y,0)$ y $B_{slow}(x,y,0)$ se pueden tomar como $I_0(x,y)$.

Comparando B_{fast} y B_{slow} y evaluando si existen grandes diferencias en un período de tiempo se puede activar una alarma de movimiento.

Entonces la función de decisión $D_I(x,y,n)$ se define como:

$$D_I(x, y, n) = \begin{cases} -1 & \text{si } |B_{fast}(x, y, n) - B_{slow}(x, y, n)| \leq T_{low} \\ 2 \frac{|B_{fast}(x, y, n) - B_{slow}(x, y, n)|}{T_{high} - T_{low}} - 1 & \text{si } T_{low} \leq |B_{fast}(x, y, n) - B_{slow}(x, y, n)| \leq T_{high} \\ 1 & \text{si } |B_{fast}(x, y, n) - B_{slow}(x, y, n)| \leq T_{high} \end{cases}$$

donde $0 < T_{low} < T_{high}$ son umbrales determinados experimentalmente.

$D_I(x,y,n)$ toma valores en el rango $[-1,1]$ y es 1 (-1) si la diferencia $|B_{fast}(x,y,n) - B_{slow}(x,y,n)|$ es mayor (menor) que el umbral T_{high} (T_{low}).

Por lo tanto, los píxeles (x,y) que cumplen con $D_I(x,y,n)$ cercano a uno son candidatos a pertenecer a una región de fuego o humo.

2.2. Detección de regiones de color del humo

El color en el comienzo de un fuego es en general grisáceo principalmente debido al vapor de agua y partículas de carbon en la composición del foco ígneo.

Estas regiones pueden ser detectadas mediante umbrales en el espacio de color YUV. Los valores de Y son en general altos, en cambio los valores de crominancia son bajos.

Entonces la función de decisión $D_2(x,y,n)$ se define como:

$$D_2(x, y, n) = \begin{cases} 1 - \frac{|U(x, y, n) - 128 + V(x, y, n) - 128|}{128} & \text{si } Y(x, y, n) > T_I \\ -1, & \text{en otro caso} \end{cases}$$

en donde $Y(x,y,n)$, $U(x,y,n)$ y $V(x,y,n)$ son valores de luminancia y crominancia en el pixel ubicado en (x,y) de la imagen del marco al instante de tiempo n , respectivamente.

Los valores $Y(x, y, n)$, $U(x, y, n)$ y $V(x, y, n) \in [0,255]$.

Como las regiones de humo son mayormente incoloras, al tener bajos valores de crominancia causan que los valores de $D_2(x,y,n)$ se aproximen a 1 y a -1 si el valor de $Y(x,y,n)$ es menor a T_I .

T_I es un umbral de intensidad de pixel para evitar que píxeles muy oscuros aparezcan en la detección.

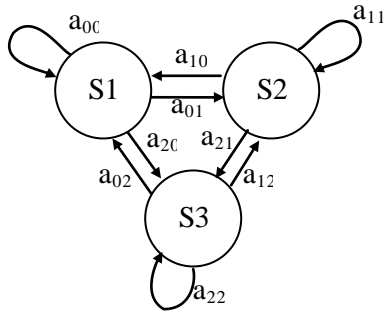
2.3. Detección de regiones que se elevan

Las regiones de humo tienden a subir en el cielo en las primeras etapas del fuego.

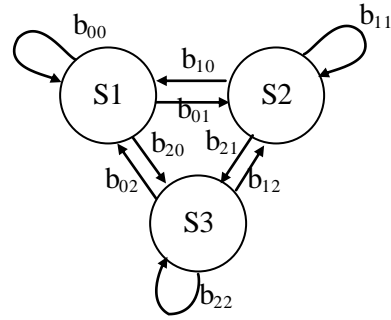
Este comportamiento característico de las columnas de humo se pueden modelar mediante modelos ocultos de cadenas de Markov (HMM) de tres estados.

Existe abundante información acerca de los modelos ocultos de Markov [12] [13] que son muy utilizados en aplicaciones de reconocimiento de patrones.

La variación temporal del número de fila del pixel superior perteneciente a la región de movimiento lento es utilizado como una señal unidimensional, $F = f(n)$, alimentando los modelos de Markov que se muestran a continuación.



λ_1 (humo de incendio)



λ_2 (nubes)

El estado S1 es alcanzado cuando el valor de la fila del pixel superior de la región en movimiento es menor que el del marco anterior (elevación). Si se alcanza S2 quiere decir que la región en movimiento descende y S3 cuando la región permanece igual.

Una región de movimiento lento se clasifica como una región que se eleva cuando la probabilidad de obtener la señal $F = f(n)$ dado el modelo de probabilidad λ_1 es mayor que la probabilidad de obtener la señal $F = f(n)$ dado el modelo de probabilidad λ_2 . Cuando el pixel superior perteneciente a la región de movimiento lento tiende a exhibir características de elevación se cumple:

$$p_1 = P(F | \lambda_1) > p_2 = P(F | \lambda_2)$$

donde F es la señal observada, λ_1 y λ_2 representan el modelo de Markov para humo de incendio y nubes, respectivamente.

Las probabilidades a_{ij} y b_{ij} de ambos modelos λ_1 y λ_2 se estiman fuera de línea. Es importante tener varios casos de vídeos de incendios y nubes para poder realizar el aprendizaje de los modelos de Markov.

El incorrecto aprendizaje impide la correcta detección de regiones de humo o aumenta la posibilidad de que ocurran falsas alarmas.

Para el aprendizaje se utiliza una secuencia de observaciones y el algoritmo de Baum-Welch que permite calcular las probabilidades a_{ij} y b_{ij} de ambos modelos.

Cuando la probabilidad p_1 (p_2) toma valores mayores que p_2 (p_1), el nivel de confianza del algoritmo se incrementa (disminuye). Entonces la función de decisión $D_3(x,y,n)$ se determina normalizando la diferencia de las siguientes probabilidades:

$$D_3(x,y,n) = \frac{p_1 - p_2}{p_1 + p_2}$$

Cuando la región de movimiento lento es clasificada como una región en elevación, se cumple $p_1 \gg p_2$, $D_3(x,y,n)$ es cercano a 1. En otro caso, $D_3(x,y,n)$ es cercano a -1.

2.4. Detección de sombras y su eliminación

Las sombras provenientes de nubes que se mueven lentamente producen falsas alarmas en la detección de humo. Las sombras de nubes toman valores de U y V muy bajos, similares a las regiones de humo.

Para poder determinar si existen sombras o no, siguiendo la metodología vista en [8], primero se calcula el promedio de valores RGB para regiones en movimiento para la imagen actual y el fondo.

Sea $S(n)$ la región de movimiento lento en la imagen I del marco n.

Entonces el vector de color promedio $\overrightarrow{C_{LS}}(n)$, de esta región en la imagen I en el marco n se calcula como sigue:

$$\overrightarrow{C_{I,S}}(n) = \frac{1}{A_{S(n)}} \left(\sum_{x \in S(n)} r_I(x, y, n), \sum_{x \in S(n)} g_I(x, y, n), \sum_{x \in S(n)} b_I(x, y, n) \right)$$

donde $A_{S(n)}$ es el área de la región $S(n)$ y $r_I(x, y, n)$, $g_I(x, y, n)$ y $b_I(x, y, n)$ son los valores de los canales rojo, verde y azul en la posición (x, y) en la imagen I del marco n .

En forma similar, el vector de color promedio $\overrightarrow{C_{B,S}}$ en la misma región en la imagen de fondo, B , se calcula como sigue:

$$\overrightarrow{C_{B,S}}(n) = \frac{1}{A_{S(n)}} \left(\sum_{x \in S(n)} r_B(x, y, n), \sum_{x \in S(n)} g_B(x, y, n), \sum_{x \in S(n)} b_B(x, y, n) \right)$$

donde $r_B(x, y, n)$, $g_B(x, y, n)$ y $b_B(x, y, n)$ son los valores de los canales rojo, verde y azul en la posición (x, y) en la imagen I del marco n . En esta implementación se utilizó B_{slow} como fondo de imagen.

En regiones de sombras, el ángulo $\theta(x, y)$, entre los vectores $\overrightarrow{C_{I,S}}(n)$ y $\overrightarrow{C_{B,S}}(n)$, debe ser pequeño y la magnitud del vector en la imagen actual debe ser menor que la del vector en el fondo de la imagen, es decir, $|\overrightarrow{C_{I,S}}(n)| < |\overrightarrow{C_{B,S}}(n)|$. Esto es así debido a que las regiones de sombra retienen en cierto grado el color y la textura subyacente.

Por lo tanto, según [7], $D_4(x, y, n) = \begin{cases} \frac{4|\theta(x, y)|}{\pi} - 1, & \text{si } |\overrightarrow{C_{I,S}}(n)| < |\overrightarrow{C_{B,S}}(n)| \\ -1, & \text{si } |\overrightarrow{C_{I,S}}(n)| > |\overrightarrow{C_{B,S}}(n)| \end{cases}$

donde $\theta(x, y)$ es el ángulo entre los dos vectores de color.

$D_4(x, y, n)$ se aproxima a -1 cuando es una región de sombras.

3. Conclusiones

Los resultados de las funciones $D_1(x, y, n)$, $D_2(x, y, n)$, $D_3(x, y, n)$, $D_4(x, y, n)$ se pueden combinar linealmente para obtener una decisión final si un píxel pertenece a una región de humo o no.

Se pueden asignar pesos iguales a cada una de las funciones pero también sería posible una combinación adaptativa de estas funciones. La combinación adaptativa queda pendiente para un trabajo futuro.

Sin embargo, los resultados son satisfactorios, aunque no se ha probado el sistema para casos reales de incendios.

Se realizó una aplicación prototipo en Java utilizando JMF (Java Media Framework), JavaVis (una biblioteca de visión por computador en Java) y Jahmm (una implementación en Java de Modelos ocultos de Markov).

Si bien el rendimiento de la aplicación es aceptable, dados los recursos informáticos de hoy en día, es posible mejorarlo mediante la utilización de lenguajes como C++ y bibliotecas de visión por computador para dicho lenguaje como OpenCV.

En este trabajo no se priorizó el rendimiento sino la experiencia con la que contaba en desarrollo de aplicaciones en Java.

Hubiera ayudado mucho contar con material en formato digital de vídeos reales de inicios de incendios forestales para poder ajustar mejor los algoritmos. Lamentablemente en Internet no se encuentra fácilmente este tipo de material con la calidad y características que se requieren (cámara fija).

Se probó el prototipo utilizando un vídeo descargado de Internet y como se puede ver en la siguiente ilustración, aparece a la izquierda el fondo B_{slow} , en el centro B_{fast} y a la derecha la región en movimiento en un instante dado.



Ilustración 1 - Detección de regiones de movimiento lento

En una segunda versión se incorporó el aprendizaje del modelo de Markov. Como se puede ver a continuación la aplicación permite realizar el aprendizaje y luego obtener las probabilidades a_{ij} .



Ilustración 2 - Segunda versión del prototipo

Como se puede ver arriba, el subalgoritmo $D_3(x,y,n)$ da cercano a 1 para la región en movimiento.

Queda trabajo pendiente por realizar para que el prototipo sea una aplicación real, pero nos da una idea de cómo la visión por computador puede resolver problemas y ayudarnos en la vida real.

4. Referencias

- [1] MODIS - modis.gsfc.nasa.gov/
- [2] <http://www.scielo.cl/pdf/bosque/v28n2/art04.pdf>
- [3] <http://www2.computer.org/portal/web/csdl/doi/10.1109/JCAI.2009.79>
- [4] <http://www.scielo.br/pdf/rarv/v26n3/a12v26n3.pdf>
- [5] R.Collins, A.Lipton, and T.Kanade. *A system for video surveillance and monitoring*. In Proceedings of the 8-th International Topical Meeting on Robotics and Remote Systems. American Nuclear Society, April 1999.
- [6] C.Stauffer and W.Grimson. *Adaptive background mixture models for real-time tracking*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 2, 1999.
- [7] Behçet Uğur Töreyn. *Fire detection algorithms using multimodal signal and image analysis*. Febrero, 2009.
- [8] T. Horprasert, D. Harwood, L. S. Davis. *A Statistical Approach for Real-time Robust Background Substraction and Shadow Detection*
- [9] JMF, *Java Media Framework*, <http://java.sun.com/javase/technologies/desktop/media/jmf/>
- [10] JavaVis, *A computer Vision Library in Java*, <http://javavis.sourceforge.net/>
- [11] Jahmm, *An implementation of Hidden Markov Models in Java*, <http://code.google.com/p/jahmm/>
- [12] Modelo oculto de Markov, Wikipedia, http://es.wikipedia.org/wiki/Modelo_oculto_de_M%C3%A1rkov
- [13] Lawrence R. Rabiner, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, <http://people.cs.ubc.ca/~murphyk/Bayes/rabiner.pdf>