CS4053 Computer Vision Lab 3
Abandoned and Removed Objects

Sean Rowan, 12100404, 4th Year Electronic and Computer Engineering

December 21, 2015

1 High Level Algorithm

1. Object Change Detection
   (a) Load in the current frame from the surveillance video
   (b) Update two median background images using different ageing rates
   (c) Isolate the absolute difference between the two median background images
   (d) Convert the absolute difference image to grayscale
   (e) Manually threshold the grayscale image into a binary image
   (f) Perform a closing and an opening to remove noise
   (g) Perform connected components on the resulting binary image
   (h) If there are any connected components:
      i. Isolate the contours of the largest area connected component
      ii. Determine any frames of interest where a local maxima in contour area of the connected component occurs
      iii. Return the frame numbers and corresponding bounding rectangles that encapsulate the local maxima contour areas
2. Abandonment/Removal Classification
   (a) In a frame of interest calculate the HLS histogram of the region of interest (the bounding rectangle around the local maxima contour area as explained above)
   (b) Create a rectangle that is twice the size of the previous bounding rectangle but shares the same centre of mass as the previous bounding rectangle. Exclude the region where the two rectangles overlap and calculate the HLS histogram of this non-overlapping region.
   (c) Compare the two HLS histograms using the correlation metric
      i. If the correlation metric is greater than or equal to 0.68 (i.e. one standard deviation away from 0), then the region of interest belongs to the background and an object was removed
      ii. If the correlation metric is less than 0.68, then the region of interest belongs to the foreground and an object was abandoned

2 Detailed Algorithm Description

2.1 Object Change Detection

2.1.1 Median Background Update

After the current frame from the sequence is loaded in, the median background update is performed. This involves keeping track of the median intensity value of a particular pixel in a histogram over a series of (n) frames. The median intensity values of all pixels over the series of frames constitutes the median background image. Computing the median values for all pixels of the sequence of frames by storing every histogram in the sequence is highly computationally inefficient, and an ageing method involving the use of a variable weight scaling factor (w_k) is used.

$$h_n(i,j,p) = \sum_{k=1..n} \begin{cases} w_k & \text{if } (f_k(i,j) = p) \\ 0 & \text{otherwise} \end{cases} \qquad \text{where } w_1 = 1 \text{ and } w_k = \alpha w_{k-1} \tag{1}$$

Figure 1: Actual Frame (left), Median Background Model $\alpha = 1.0025$ (middle), Median Background Model $\alpha = 1.005$ (right). The middle frame is missing the abandoned object located roughly in the centre of the image while the right frame isn't because the middle frame's ageing rate is lower and therefore its Median Background Model will update more slowly.

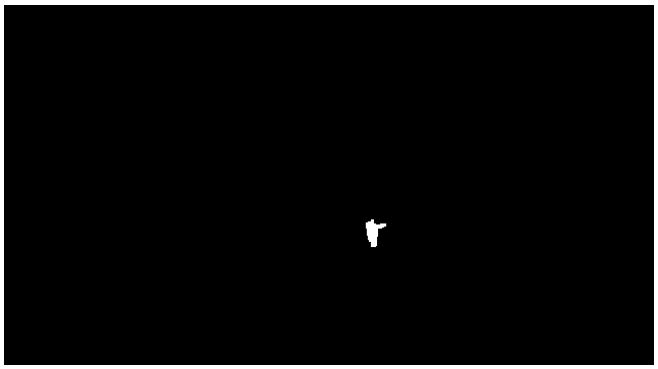

Figure 2: Closed and Opened Median Difference Image (object of interest is the white region)

The choice of the ageing factor α is what controls the rate in $frames^{-1}$ at which the median background image is updated for a given sequence. A higher value of α denotes a faster update rate, and a lower value of α denotes a slower update rate. In this project, relatively low update values are chosen, $\alpha_1 = 1.0025$ and $\alpha_2 = 1.005$, in order to be robust against any human movements entering the median background image.

2.1.2 Median Background Difference Image

Two median background models are updated for every frame, with ageing rates 1.0025 and 1.005. This means that changes to the scene that subsequently remain static will be updated in the 1.005 ageing rate median background model before the 1.0025 ageing rate median background model. This means that there will be a window of frames where the object change is reflected in 1.005 model but not in the 1.0025; therefore, to capture this difference, the absolute difference between the 1.005 model B_{1n} and the 1.0025 model B_{2n} is calculated with every new frame in the sequence and the previous difference image D_{n-1} is updated to the current difference image D_n.

$$D_n = |B_{1n} - B_{2n}| \tag{2}$$

2.1.3 Conversion To Grayscale

The Median Background Difference Image obtained in the previous step is a colour image, and this image is now converted to grayscale using a simple weighting formula that is based on the human visual system's higher sensitivity to green light and lower sensitivity to blue light.

$$Y = 0.2126R + 0.7152G + 0.0722B \tag{3}$$

2.1.4 Manual Binary Threshold

Many binary thresholding techniques rely on minimising the in-class variance between the sets of white and black pixels in an image (e.g. Otsu's Method), which makes sense when the image is somewhat featured with a high variance of grayscale values. However, in this project using the Median Background Difference Image, most of the updated difference images will be almost entirely zero-valued because the difference image only contains discernible information when an object abandonment/removal occurs. Therefore, instead of allowing noise pixels

to skew the in-class variance such as in Otsu's Method when an object event is not occurring, a manual threshold is set that removes noise pixels by setting them to 0 and isolates region of interest pixels by setting them to 255.

$$B(i,j) = \begin{cases} 255 & \text{if } G(i,j) \geq 30 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

2.1.5 Closing and Opening

This is a noise removal sequence of operations. A closing is a dilation followed by an erosion, while an opening is an erosion followed by a dilation.

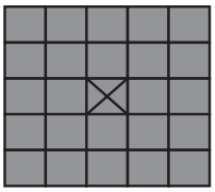

Figure 3: 5x5 Structuring Element used in the Erosion and Dilation operations

Both an erosion and a dilation make use of a structuring element (displayed above) that is shifted around an input image. For an erosion, the centre of the structuring element is placed over every 255-valued pixel in the input binary image, and if the surrounding pixels in the input image that overlap with the structuring element are all 255-valued, then the centre pixel remains 255-valued in the output image; otherwise the centre pixel is set to 0. For a dilation, the centre of the structuring element is placed over every 255-valued pixel in the input image as before, except now all of the surrounding pixels in the input image that overlap with the structuring element are set to 255 in the output image.

The closing operation (dilation followed by erosion) in effect closes small gaps that exist within a tightly clustered group of white pixels (the words tightly clustered can be defined by the size of the structuring element) without necessarily increasing the perimeter of the clustered group of white pixels. The opening operation does the opposite by more clearly defining separated regions in tightly clustered groups of white pixels. By combining the closing and the opening respectively in that order, two erosions will be performed back to back. This causes noise pixels which do not belong to tightly clustered groups of white pixels to be taken out of the image while still preserving the shape and more clearly defining the tightly clustered groups of white pixels.

2.1.6 Connected Components

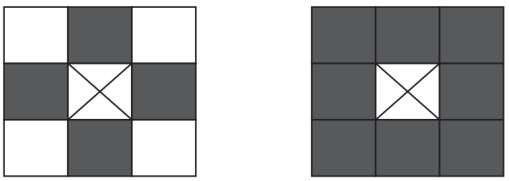

Figure 4: 4-adjacency (left) and 8-adjacency (right) structuring elements

Figure 5: 8-adjacency previous neighbours (left) and 4-adjacency previous neighbours (right) structuring elements

At this stage of the current frame processing, there may be multiple disconnected white regions which either correspond to different objects in the original frame or a separation between one object that could occur due to the object colour matching the background colour in a particular region of pixels, therefore causing it to be binary thresholded to black in a previous stage because this particular region of pixels would be similar to noise

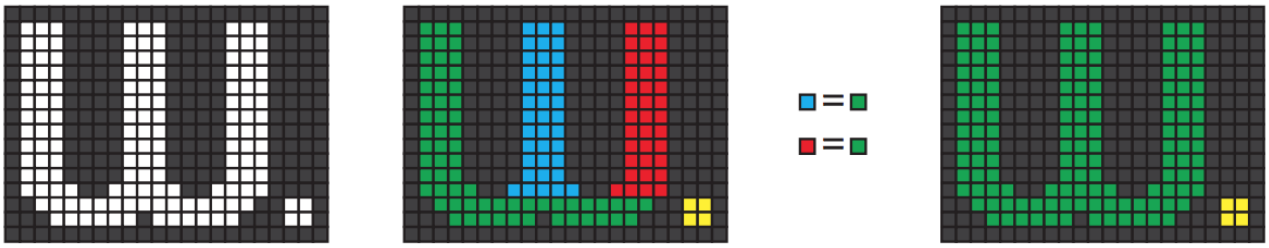

Figure 6: Connected Components Algorithm

pixels.

In order to provide more context to the binary image, the individual clusters of white pixels are given labels using the connected components algorithm as follows:

- Every non-zero pixel in the input binary image is first indexed for every row and subsequently for every column
- Using the 8-adjacency structuring element (shown above), and the previous neighbours mask of the 8-adjacency element, the previous neighbour pixels from the current white pixel are selected.
- If these previous neighbour pixels are all black, then the current pixel forms a new object is given a new label
- Else pick one of the labels (if there are multiple different labels) from the previous neighbouring pixels and assign it to the current pixel.
  - If there are multiple labels in the previous step, note the equivalence of the labels.
- After this is completed for both the rows and the columns indexing paths, update the labels for equivalence. This is shown in the far right image of the Connected Components Algorithm image above.

2.1.7 Contour Area

For every frame that a connected component exists, calculate the area for this connected component using the contour area algorithm. For every vertex from the connected component (i_r, j_r), $r \epsilon [0, n-1]$, where n is the number of vertices, the area of the connected component is:

$$area = \frac{1}{2}\left|\sum_{k=0}^{n-1}(i_k j_{k+1} - i_{k+1} j_k)\right| \tag{5}$$

2.1.8 Find Local Maxima Contour Area In a Sequence of Frames

When an object is abandoned or removed from the scene, the object pixels which are more different from the corresponding background model pixels will appear sooner in the median update difference image than object pixels which more closely match the corresponding background model. Subsequently, the same order holds for the disappearance of object pixels from the median update difference frame. Therefore, a region of interest is classified by a 'breathing' (i.e. expanding and shrinking) contour area that occurs in a certain location over a sequence of frames. To isolate the region of interest, the local maxima of the contour area in a sequence of frames is found as follows:

- In the current frame (n) of a sequence of frames:
  - If the current max contour area in the frame is > 0 AND if the current max contour area $>$ the previous frame's max contour area
    - Update local maxima contour area and frame in which it occurs
  - If the current max contour area in the frame is $= 0$ AND if the previous frame's max contour area > 0
    - Current value of local maxima contour area (and it's frame) is stored as the local maxima for this sequence of frames of interest and the temporary variables are reset

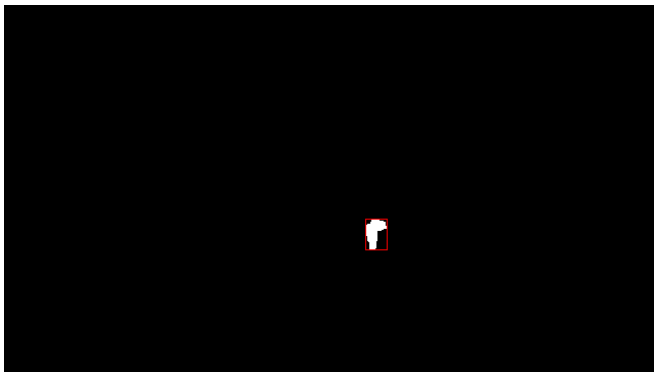

Figure 7: Local Maxima Contour Area Detection

2.2 Object Abandonment/Removal Classification

2.2.1 Conversion From RGB to HLS Colour Space

$$L = Max(R,G,B) + \frac{Min(R,G,B)}{2} \tag{6}$$

$$S = \begin{cases} Max(R,G,B) - \frac{Min(R,G,B)}{Max(R,G,B)+Min(R,G,B)} & \text{if } L < 0.5 \\ Max(R,G,B) - \frac{Min(R,G,B)}{2-(Max(R,G,B)+Min(R,G,B))} & \text{if } L \geq 0.5 \end{cases} \tag{7}$$

$$H = \begin{cases} 60\frac{(G-B)}{S} & \text{if } R = Max(R,G,B) \\ 120 + 60\frac{(B-R)}{S} & \text{if } G = Max(R,G,B) \\ 240 + 60\frac{(R-G)}{S} & \text{if } B = Max(R,G,B) \end{cases} \tag{8}$$

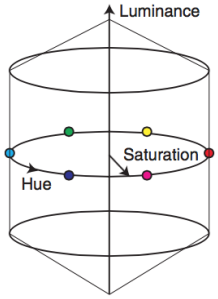

Figure 8: Hue, Luminance and Saturation Colour Space

2.2.2 HLS Histogram Calculation and Normalisation

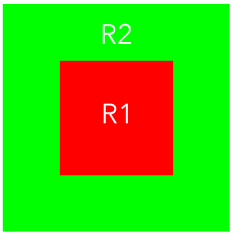

Figure 9: R1: Region where object change occurs, R2: Background region

A hue-luminance-saturation (HLS) histogram of an image is a 3-dimensional representation of the number of pixels in the image that are at a certain intensity for each of the 3 channel types (hue, luminance and saturation). The HLS histogram is chosen instead of the RGB histogram because the HLS space tends to provide

better comparison results for some images than the RGB space does.

An HLS histogram is calculated for region R1 as seen in Figure 9 which is the bounding rectangle around the local maxima contour area. A region R2 is created which is twice the area of R1 but shares the same centre of mass as R1. The region of R2 that overlaps with R1 is removed from R2. The HLS histogram is now calculated for the resulting region R2.

Both histograms are now normalised by dividing every column on the intensity axis for each colour channel by the total number of pixels, ensuring that every column is now contained within [0,1], i.e. the histograms now become probability density functions.

2.2.3 Correlation Metric Calculation Between Two Histograms

$$D_{correlation}(h_1, h_2) = \frac{\sum_i ((h_1(i) - \bar{h_1})(h_2(i) - \bar{h_2}))}{\sqrt{\sum_i (h_1(i) - \bar{h_1})^2 \sum_i (h_2(i) - \bar{h_2})^2}} \tag{9}$$

The correlation metric is now computed between the two normalised histograms, and this value is contained within [-1,1]. Simply, if the absolute value of the correlation value is greater than or equal to 0.68 (which represents one standard deviation away from 0 correlation), then R1 and R2 are sufficiently similar to suggest that the object has been removed. If the correlation value is less than one standard deviation, then the object was abandoned.

3 Results

Figure 10: Video 1, Region of Interest 1, Abandoned

Figure 11: Video 1, Region of Interest 2, Removed

Figure 12: Video 2, Region of Interest 1, Abandoned

Figure 13: Video 2, Region of Interest 2, Removed

Video/Event	Dice Coefficient	Detection Time (s)	Abandonment(A) or Removed(R)	TP/FP/FN
1/1	0.75	3	A - Correct	TP
1/2	0.75	5	R - Correct	TP
2/1	0.90	2	A - Correct	TP
2/2	0.78	5	R - Correct	TP

Precision = 1, Recall = 1

3.1 Discussion

A value of the dice coefficient closer to 1 indicates a better object detection has occurred. This is because ideally the overlap area exactly would exactly equal the ground truth area and the experimental area, which in terms of the dice coefficient causes it to be equal to 1. All of the dice coefficients in this project are greater than 0.75, which indicates a good overlap has occurred.

The detection times for the object abandonment/removal events are reasonable, as surveillance systems generally don't need to work in real time (also an object must remain stationary from the time it is placed for a period of time before it can be classified as abandoned).

The abandonment/removal classification method worked well too, causing the precision and recall to be equal to 1, which is the best possible value. The abandonment/removal problem is quite an interesting problem and there are many solutions to it with varying degrees of complexity. Some solutions check for edges in the frame of interest that closely match the arbitrary edge around the region of interest picked up by the median difference local maxima area, however this will encounter problems where part of the object closely matches the background (as happens in the second test video) and the object is split into multiple connected components. Another interesting solution is to perform 'region growing' by first eroding the region of interest, then expanding the region in the direction of the original region of interest's edge orientations until an edge is encountered. Then, if the total area of the expanded area region plus the original eroded area region is less than or equal to the original region of interest's area, the object was abandoned and vice versa for removal. Many of these techniques can also be combined with optical flow algorithms (e.g. feature based) in order to track humans as they abandon/remove objects to provide more robustness to the system.

4 Citation

Dawson-Howe, Kenneth. A Practical Introduction to Computer Vision with OpenCV. Print.