Tracking Over Non-Overlapping Cameras

Group : Tracking 2

Members : Gan Hui Ting

Lee Chin Yin

Tan Pei Xuan Michelle

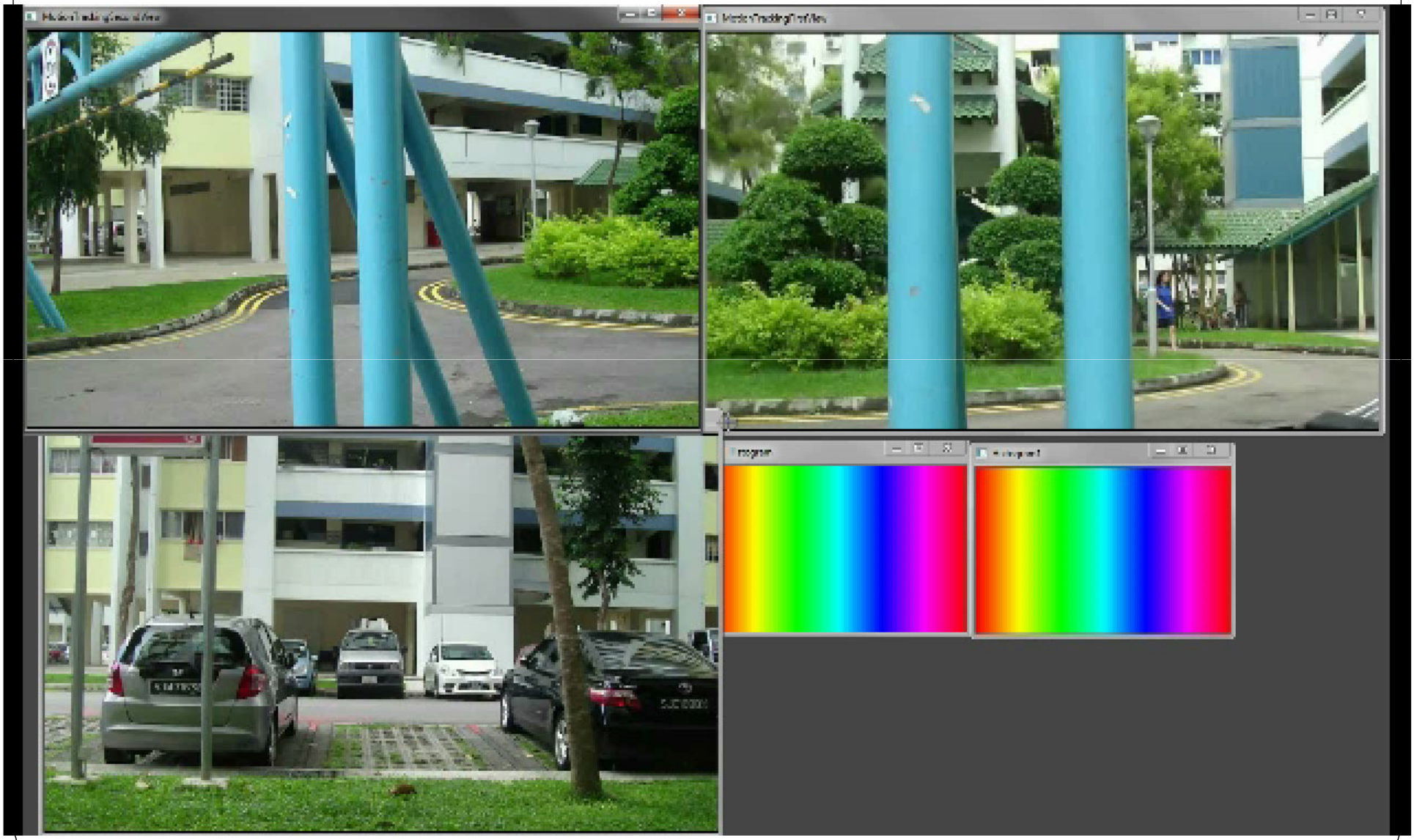
Objectives/Requirements

- To track moving objects over multiple non-overlapping cameras
  - Cameras will capture different background.
  - Tracking results should be as accurate as possible.
  - Program should have minimal user input .
  - Objects can be occluded for some time.
  - Multiple cameras (2 or more cameras)

Our program - Track-2

- ✓ **Minimal user input** - User only need to indicate the objects (humans) to track in camera 1 and the objects will be traced automatically in video 2 and 3.
- ✓ **Occlusion of objects** - The humans are occluded some of the time.
  - By objects like pillars
  - By another moving person
- ✓ **Multiple cameras** - Support more than 2 cameras – e.g. 3 cameras in our sample results.
- ✓ **Accurate tracking results**

Sampled video



Main Algorithms

1. Background removal – To facilitate accurate tracking
2. Camshift tracking – Tracking of objects
3. Blob tracking – For handling occlusion and switching between cameras

Detailed Steps

First video

1. User select the 2 humans to track – Compute the color histogram
2. Perform background removal to differentiate the moving object from the background
3. Perform camshift to track the objects moving between frames gradually
4. When occlusion occurs, camshift could not find the object in the neighboring area. Perform blob tracking to track where the object reappears and set the track window of camshift to the object and repeat step 3.

Other videos - No clues on where the objects will be entering.

1. Perform background removal to differentiate the moving object from background.
2. Perform blob tracking to track where the object appears for the first time. Set track window of camshift to the object.
3. Perform camshift tracking.

1. Background removal

- With clean plate

Our method – Find the absolute different

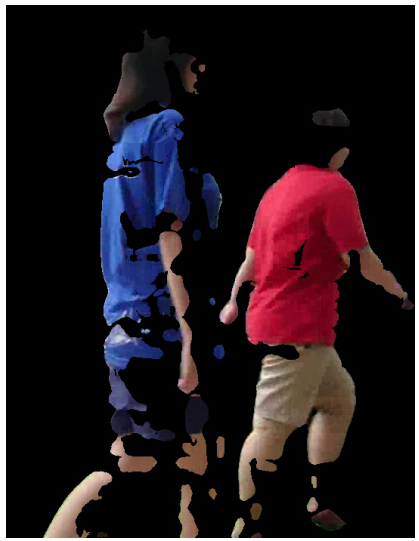
- Compute the color difference between the clean plate and current frame
- When the pixel difference is large, set as foreground
- When the pixel difference is small, set as background

Why we use this method –

- Simple – Given: Stationary cameras with clean plate

Other possible methods:

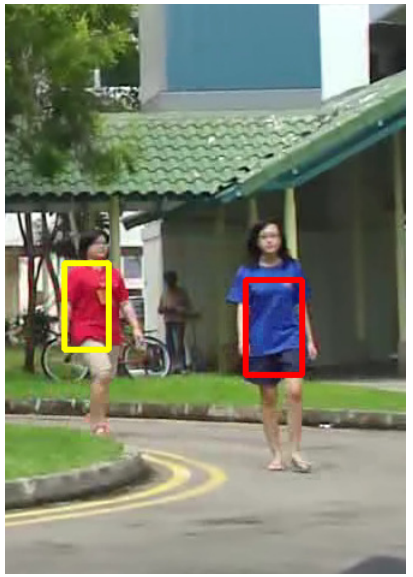
- K-mean clustering



2. Camshift tracking

Our method -

- When the user select the object to track, create a color histogram.
- Using the histogram, calculate the object probability for each pixel in incoming video frame – back projection.
- Shift the location of the object rectangle in each video frame – To find the new location, start at the previous location and search in the neighbour pixels (in negative gradient). The center of the rectangle will be the highest concentration of pixels in the object-probability image.
- Calculate the size - by selecting the scale that are the best fit to the object-probability pixels



Why we use this method –

- Can handle the change of object size

Other possible methods –

- Mean shift tracking

3. Blob tracking

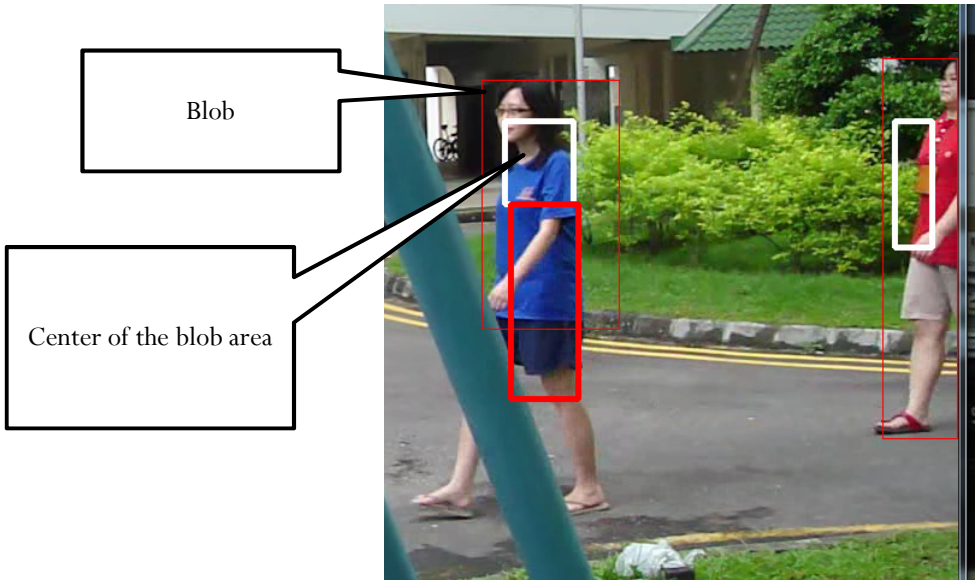
Our method -

- Compute the difference between the previous frame and current frame.
- Track the 2 largest blobs (collection of pixels) which is most likely to be the 2 humans
- Compute the histogram for the center area of the blob (which is most likely the shirt)
- Compare the blob histogram with the object color histogram
- If high similarity, means object found. Else, continue to find the object in next frame.

Why use this method –

- Stationary cameras so the two moving objects

  is likely be observed in blob tracking



Other possible methods -

- Kalman Filtering
- Lucas Kanade tracking

References

- For Background removal
  - **Brian's Python Dev Blog :** http://bitsofpy.blogspot.com/2009/09/greenscreen.html
- Grouping of blobs
  - **Motion Tracking with a Webcam :** http://appdelegateinc.com/blog/2010/08/02/motion-tracking-with-a-webcam/
- Camshift –
  - **OpenCV2.1** Samples "camshift.py"
  - **How OpenCV's Face Tracker Works :**

    http://www.cognotics.com/opencv/servo_2007_series/part_3/sidebar.html