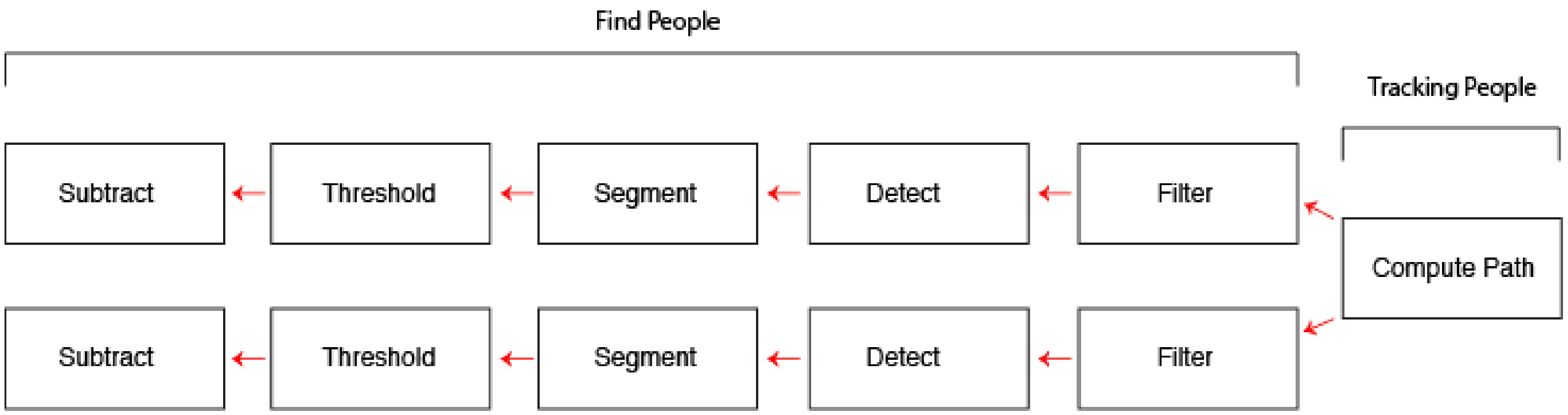
Security Camera Counter

Solon Mao and Sarah Tan

Algorithm and Dependencies on Detecting and Tracking People



The red arrows represent dependencies. For example, Threshold is dependent on Subtract. It is important to note that Tracking People is dependent on the result of two different result frames from Find People.

Detection: Parallelizing each Pixel in Frame Subtraction and Thresholding

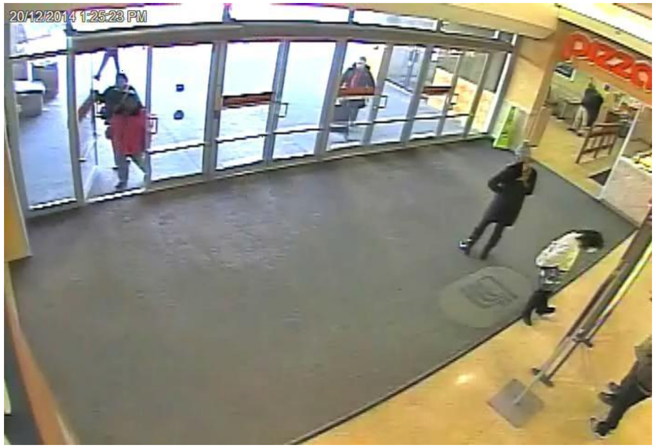


Figure 1a: Frame 1

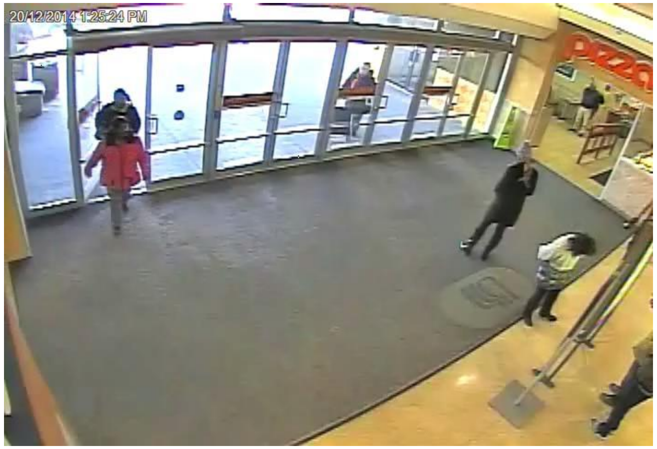


Figure 1b: Frame 2

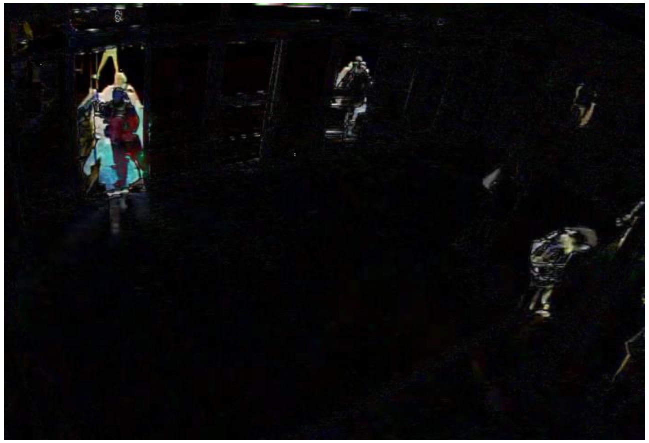


Figure 2a: Frame Subtraction

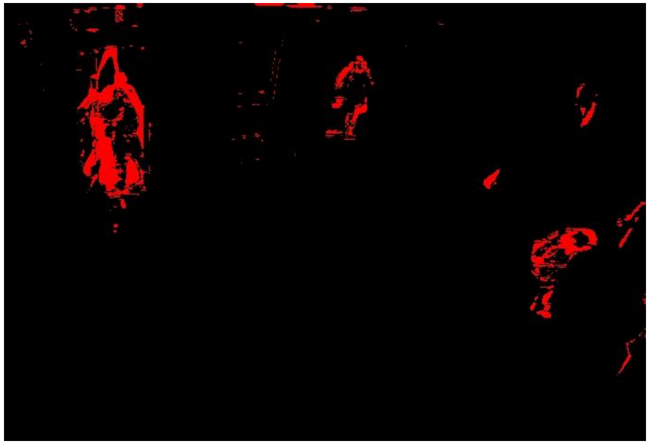
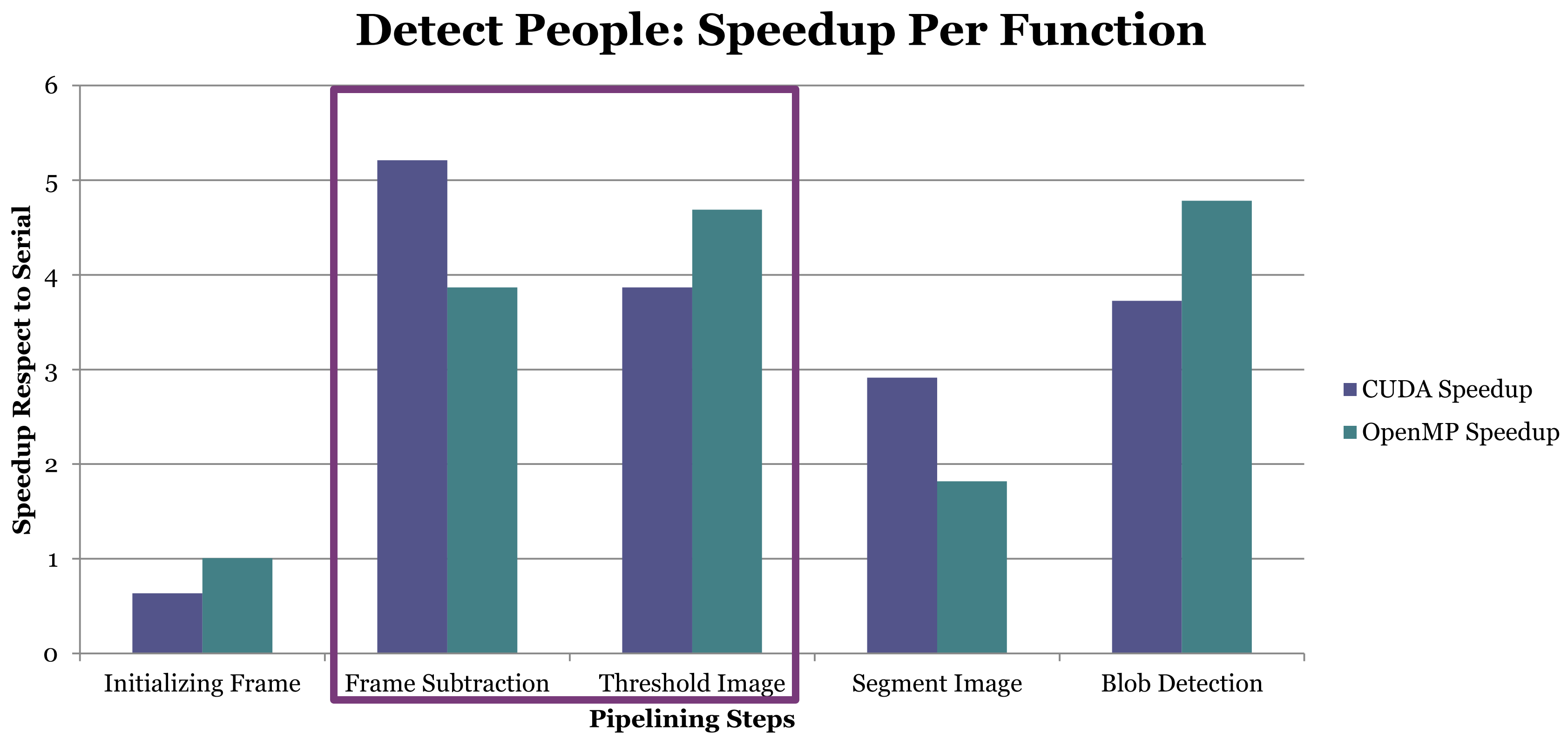


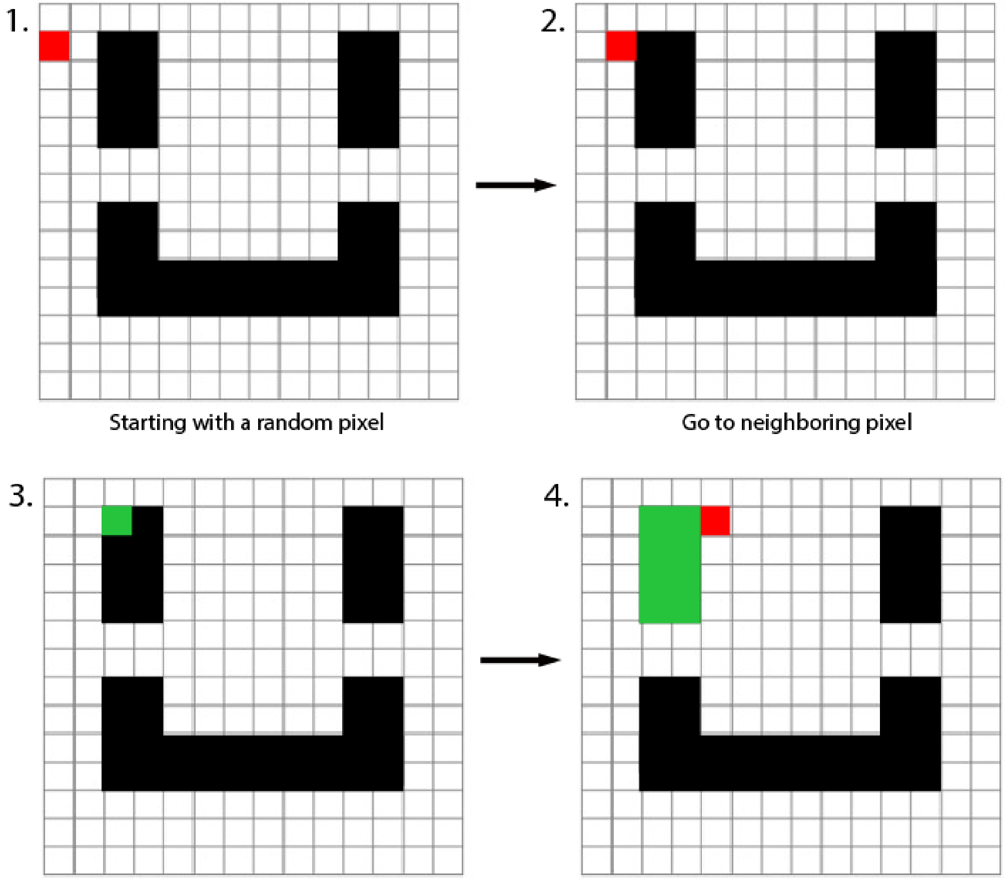
Figure 2b: Frame Threshold

Average Speedup for Frame Subtraction and Thresholding Compared to Serial



Detection: Parallelizing through Tiling and Starting Points in Watershed Segmentation

Red pixel is the current pixel we are looking at
We continue to a neighboring pixel until we reach an on pixel (black ones)
Once we hit a black pixel, we look and label all the neighboring black pixels with a tag. This is one segment.

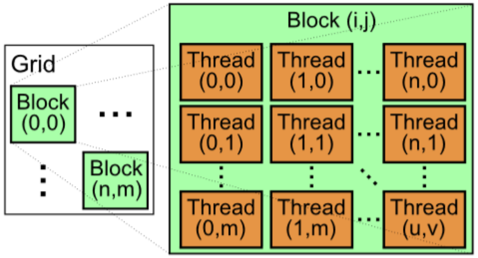


1. Starting with a random pixel

2. Go to neighboring pixel

3. Reach an on pixel

4. Create a segment. Continue to a neighboring pixel.



CUDA

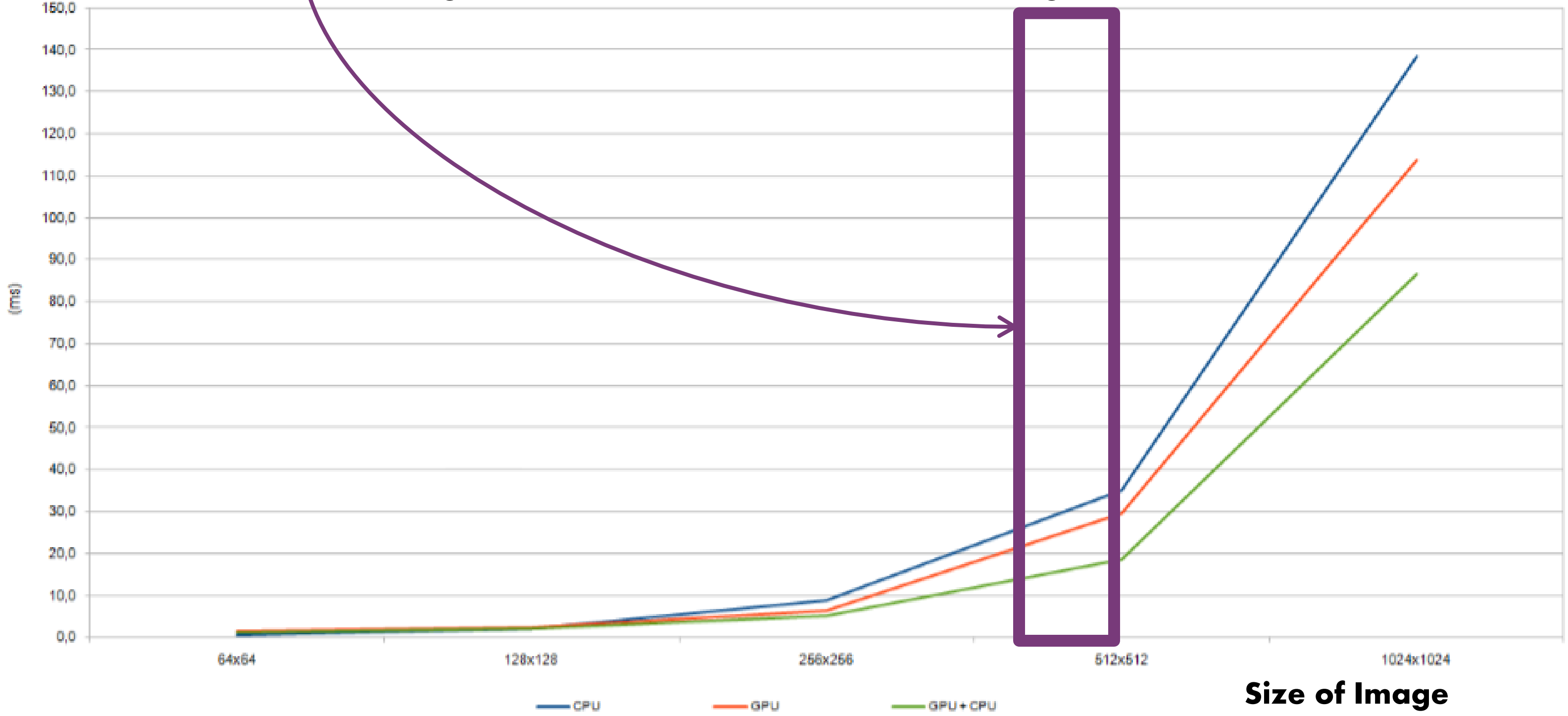
```
#pragma parallel for
For each pixel x:
  if ((Value(x) == 1) && Seen(x) != 1):
    Seen(x) = 1;
    Push(x);
    while(isNotEmpty(Neighbors(x)):
      Seen(Neighbors(x)) = 1;
      Push(Neighbors(x));
  while (isNotEmpty(Stack)):
    y = Pop(Stack);
    Label(x) = label;
  label++
```

OpenMP Pseudocode

Watershed (Partially) Serialized

Average Amount of Data in 480 x 720 image after Frame Subtraction

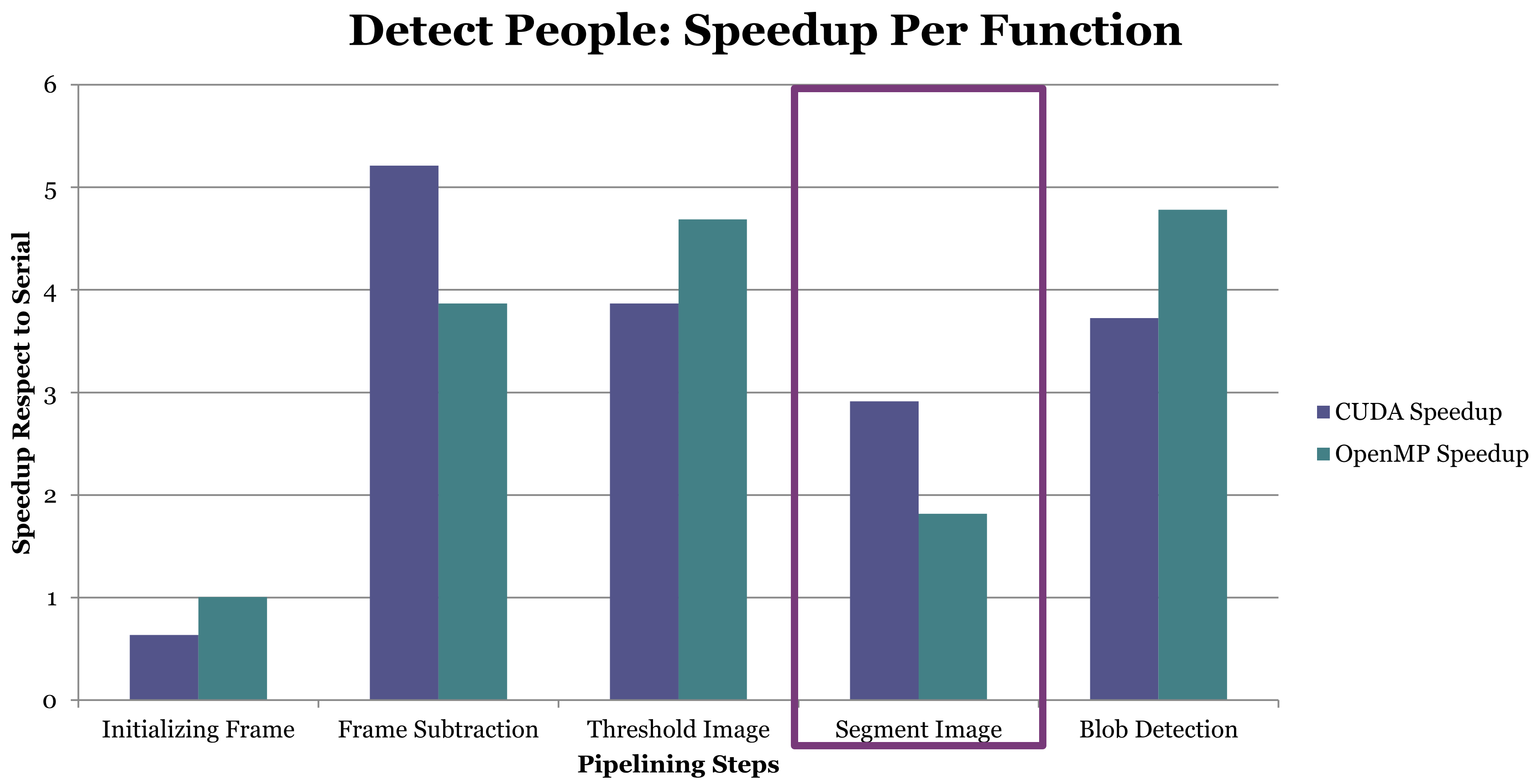
Average Time Taken to Perform Watershed Segmentation



Size of Image

Figure 3: Graphic comparison of average time between algorithms [1]

Average Speedup for Watershed Segmentation Compared to Serial



Detection: Parallelizing each Pixel per Detected Person

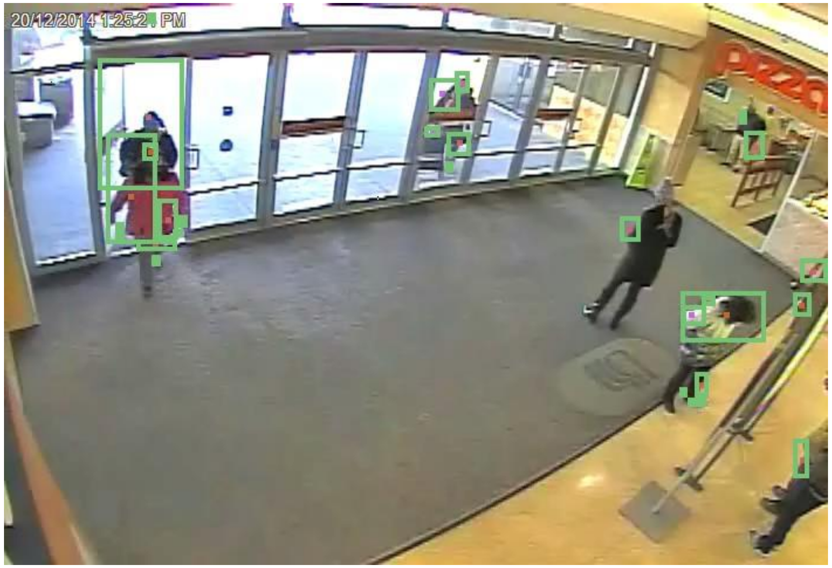


Figure 3: Creating Boxes around Segmentation

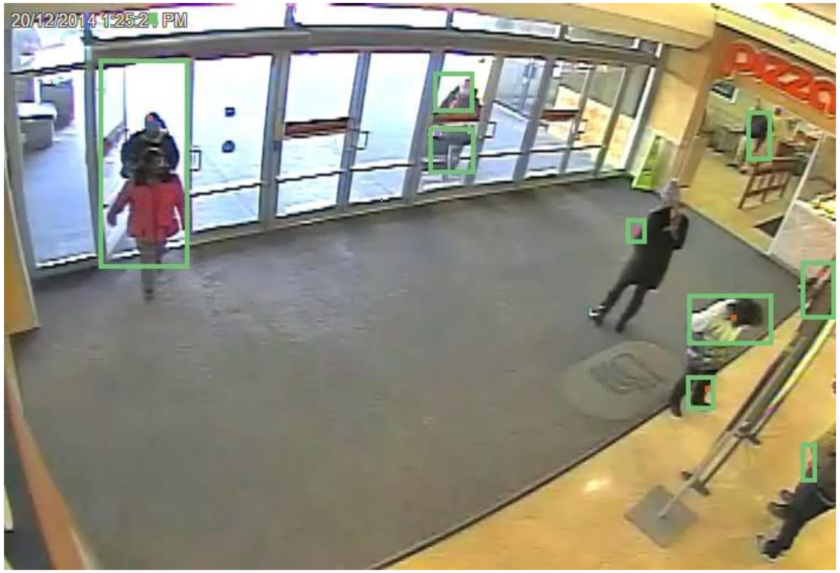
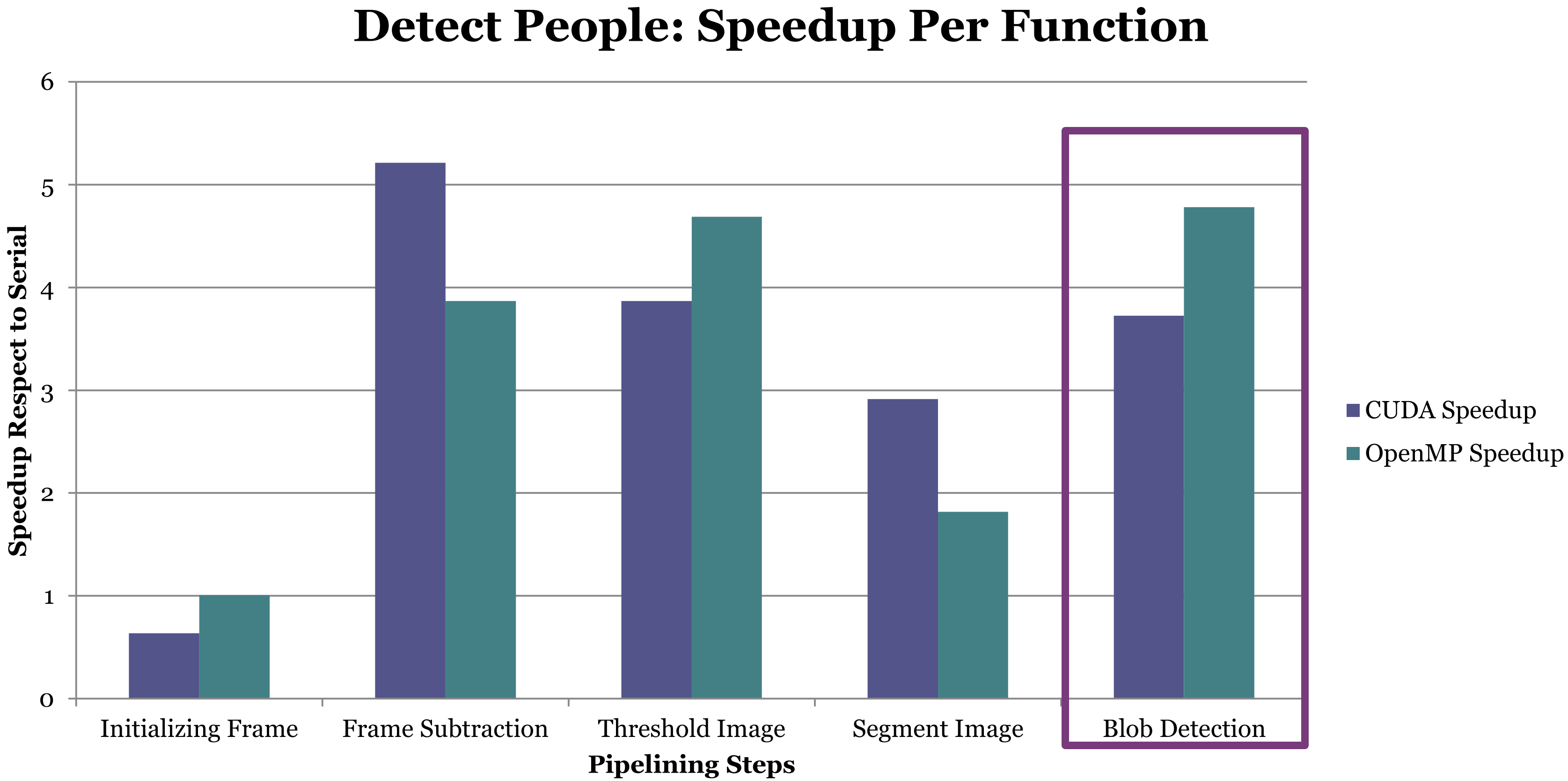


Figure 4: Filter and Detect People

Average Speedup for Blob Detection and Filtering for People Compared to Serial



Visualization of Each Method Relative Time Processing if Input Video Stream has No Delays

Serial

OpenMP

CUDA

Common Questions

- Why do we only detect moving people?
  - Make assumption people will move in security camera footage
  - Increase speedup because less-data to work with
- Why is the detection accuracy so poor?
  - Comparing only moving portions of the image
  - Generalize tracking other cameras and orientations without max size for person
- How would we improve accuracy?
  - Get video footage of average running background

Tracking: Compute Path of Person through Pipelining, Centroids, and Direction of Travel

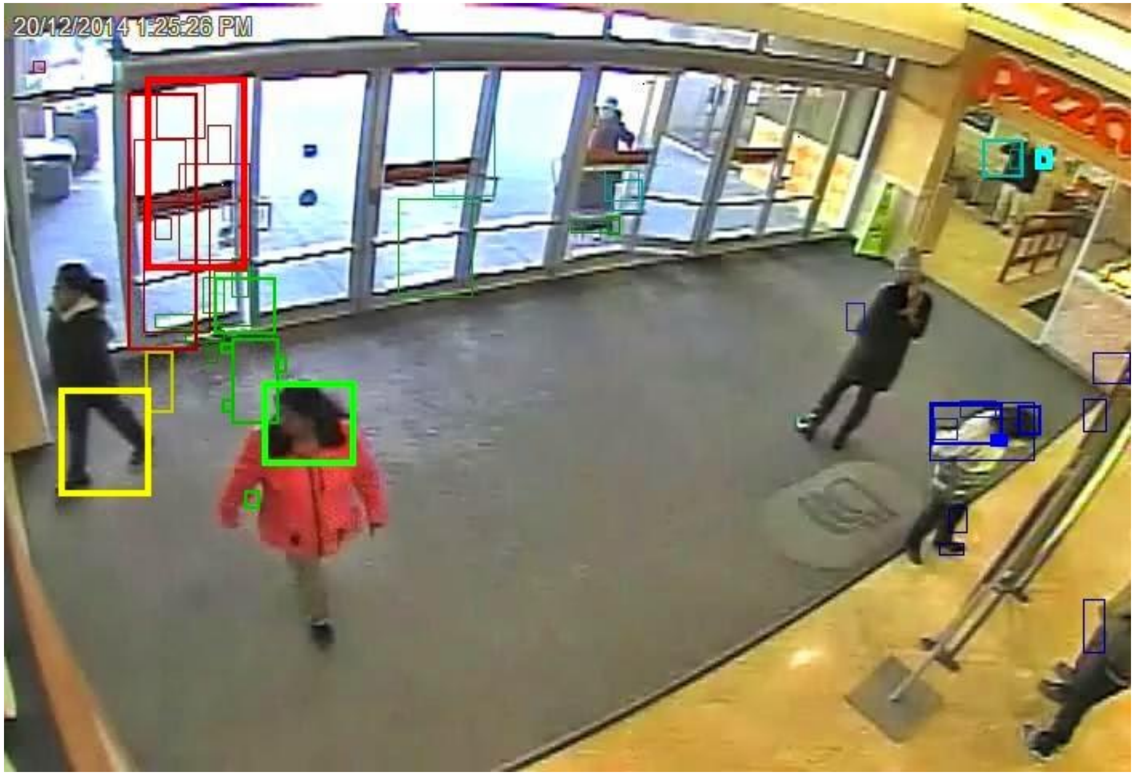
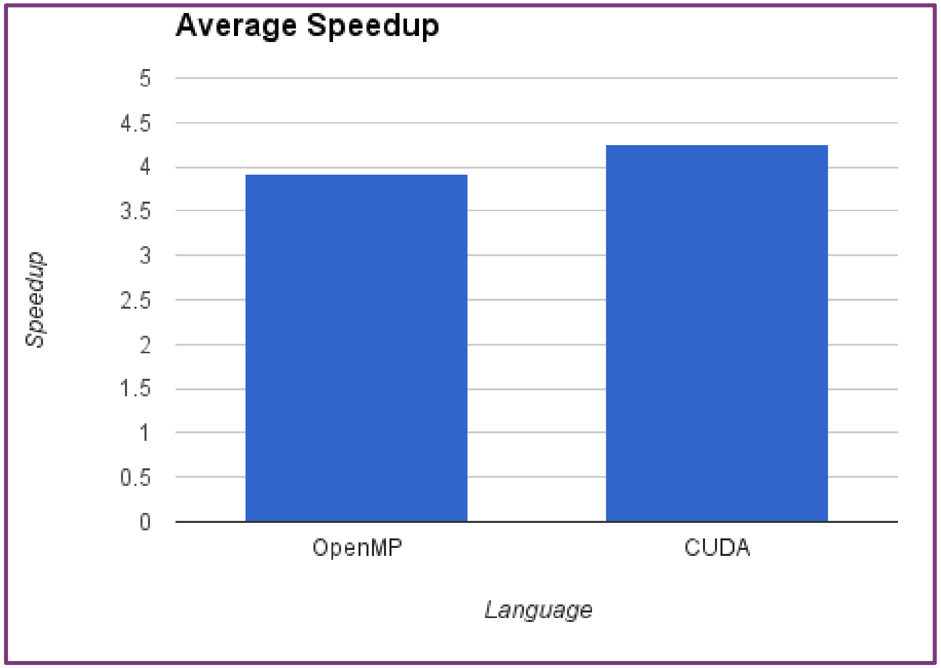
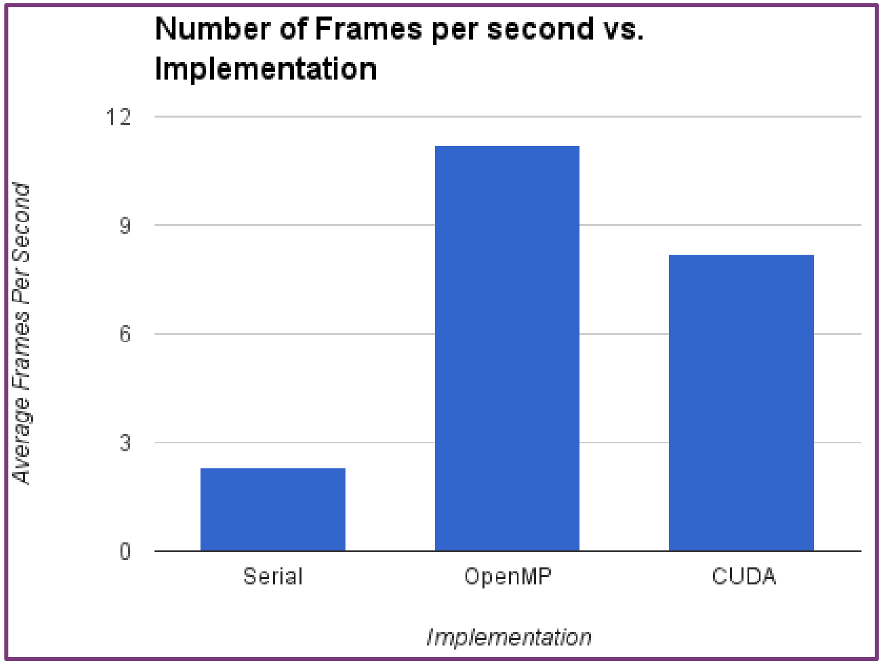


Figure 4: Path of person over 3 frames

Average Speedup and Throughput (Compared to Serial) Per Method





References

- [1] Vitor, Giovani Bernardes, Janito Vaqueiro Ferreira, and André Korbes. "Fast image segmentation by watershed transform on graphical hardware." *XXX Iberian Latin American Congress on Computational Methods in Engineering-CILAMCE 2009*. Vol. 1. 2009.

- [2] Jiang, Shu. "Parallelization of Image Segmentation Algorithms.“

- [3] Farhadi, Masoud, Seyed Ahmad Motamedi, and Saeed Sharifian. "Efficient human detection based on parallel implementation of gradient and texture feature extraction methods." *Machine Vision and Image Processing (MVIP), 2011 7th Iranian*. IEEE, 2011.

- [4] Bilgic, Berkin, Berthold KP Horn, and Ichiro Masaki. "Fast human detection with cascaded ensembles on the GPU." *Intelligent Vehicles Symposium (IV), 2010 IEEE*. IEEE, 2010.

- [5] Comaniciu, Dorin, Visvanathan Ramesh, and Peter Meer. "Kernel-based object tracking." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 25.5 (2003): 564-577.