Video Segmentation

Sung Soo Hwang

Introduction

- What is image/video segmentation?
  - Process of partitioning a digital image into multiple regions
  - Application
    - Chroma-keying
    - Surveillance camera

Background subtraction

- Given a video, identify the foreground objects in that video
  - In most cases, objects are of interest, not the scene

Image at time t:

$I(x, y, t)$

$-$

Background at time t:

$B(x, y, t)$

$| > Th$

From "Foreground Background detection from video"

Background subtraction

- Concept
  - Assume we have two image frames $f(x, y, t)$ and $\mathrm{B}(x, y, t)$
  - We detect changes between two images pixel by pixel
  - $d(x, y, t) = \begin{cases} 1 \; or \; 255 & if \mid f(x, y, t_i) - B(x, y, t) \mid > T \\ 0 & otherwise \end{cases}$
  - The difference with value 1 are considered the result of object motion
  - Assumption
    - Two images are registered spatially
    - Illumination is relatively constant

Background subtraction

- Key to successful background subtraction
  - We should handle sudden or gradual illumination changes
  - Repetitive motion
    - Tree leaves
    - waves
  - Long-term scene change
    - unattended bag
    - parked car

➔Estimating good background is the key!

Background subtraction

- Background estimation
  - Mean filter
    - Background is the mean of the previous n frames
    - $B(x,y,t) = \frac{1}{n}\sum_{i=0}^{n-1} f(x,y,t-i)$ or
    - $B(x,y,t) = \frac{1}{n}\sum_{i=0}^{n-1} f(x,y,i)$

n=10

n=20

n=50

From "Foreground Background detection from video"

Background subtraction

- Background estimation
  - Median filter
    - Background is more likely to appear in a scene
    - $B(x, y, t) = median(f(x, y, t - i))$ or
    - $B(x, y, t) = median(f(x, y, i))$

n=10

n=20

n=50

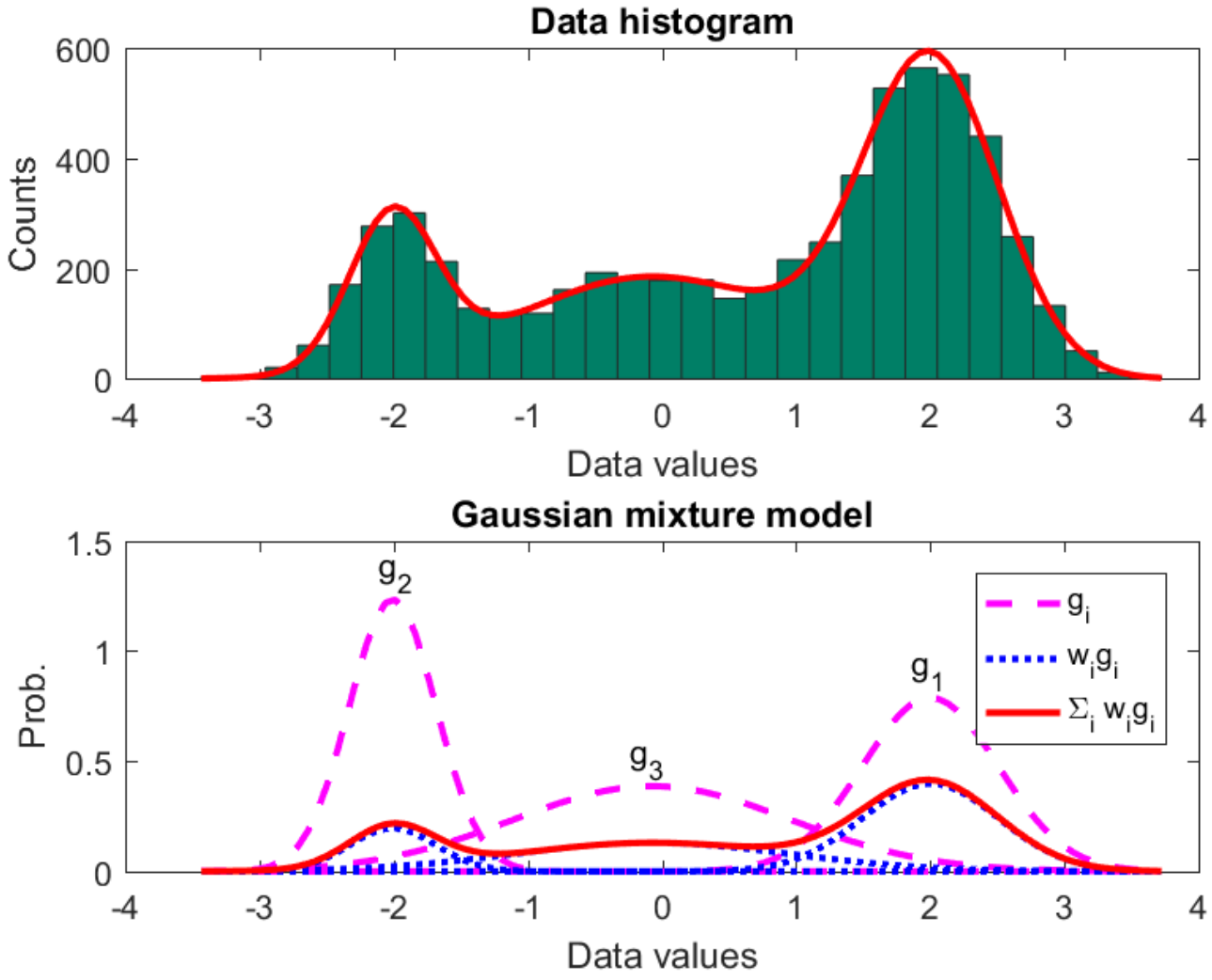
From "Foreground Background detection from video"

Background subtraction

- Conditional probabilities
  - $p(A \cap B) = p(A|B)p(B) = p(B|A)p(A)$
- Bayes rule
  - $p(B|A) = \frac{p(A \cap B)}{p(A)} = \frac{p(A|B)p(B)}{p(A)}$
- Assume A is pixel value, and B is background
  - If we can figure out(or estimate) $p(A|B)$, then we can find out the probability of a certain pixel value being background ➔$p(B|A)$

Background subtraction

- Gaussian mixture model(or mixture of Gaussian)

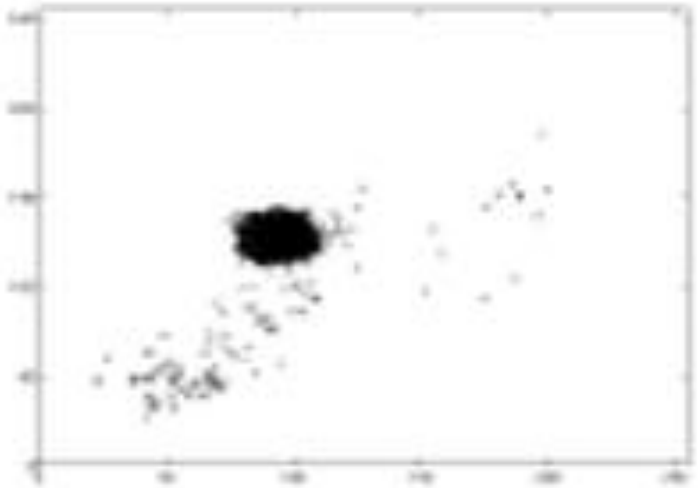


http://mirlab.org/jang/books/dcpr/example/output/gmm1d02.png

Background subtraction

- Background estimation using GMM
  - Determine the number of mode of GMM
  - At the training stage, estimate mean and variance of each Gaussian model with the training data

    ➔estimate $p(A|B)$
  - Each pixel is classified into background/foreground by calculating $p(B|A)$

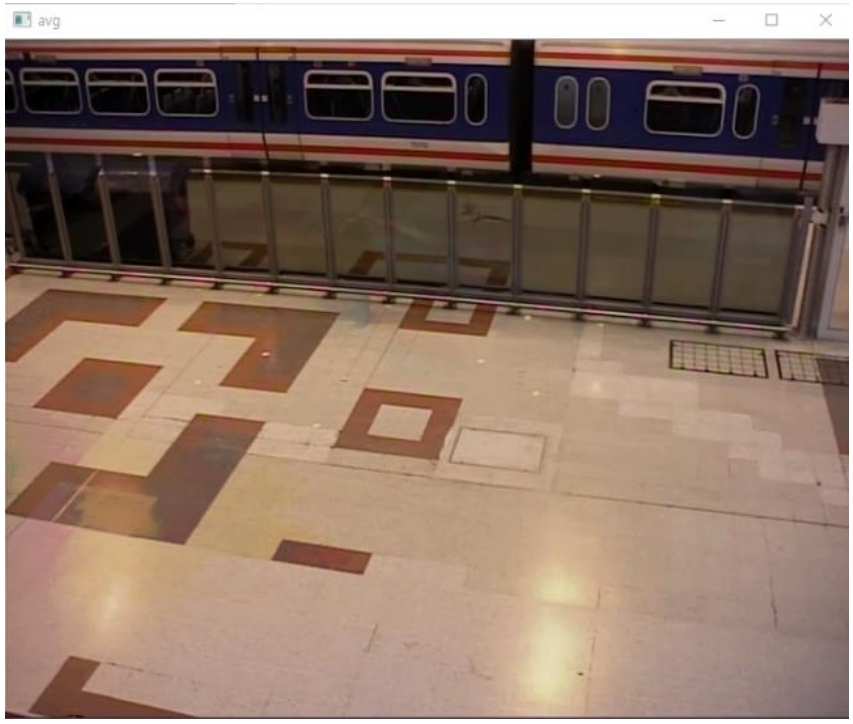


From "Foreground Background detection from video"

Background subtraction

- Example code
  - Generating average image

```
int main() {
	VideoCapture capture("background.mp4");
	Mat image, sum, avg;
	int cnt = 2;

	capture >> avg;

	while (true) {
		if (!capture.read(image)) break;
		add(image / cnt, avg*(cnt - 1) / cnt, avg);

		imshow("avg", avg);
		cnt++;
		waitKey(33);
	}
}
```



Background subtraction

- Example code(using absdiff)

```
int main() {
	VideoCapture capture("background.mp4");
	Mat background, image, gray, result, foregroundMask, foregroundImg;

	//set the first frame as background
	capture >> background;
	cvtColor(background, background, CV_BGR2GRAY);

	while (true) {
		if (capture.grab() == 0) break;
		capture.retrieve(image);
		cvtColor(image, gray, CV_BGR2GRAY);

		absdiff(background, gray, foregroundMask);
		threshold(foregroundMask, foregroundMask, 50, 255, CV_THRESH_BINARY);
		foregroundMask.copyTo(foregroundImg);
		gray.copyTo(foregroundImg, foregroundMask);

		imshow("foregroundImg", foregroundImg);
		imshow("foregroundMask", foregroundMask);
		imshow("background", background);

		waitKey(33);
	}
}
```

Background subtraction

- Example code(using absdiff)

Background subtraction

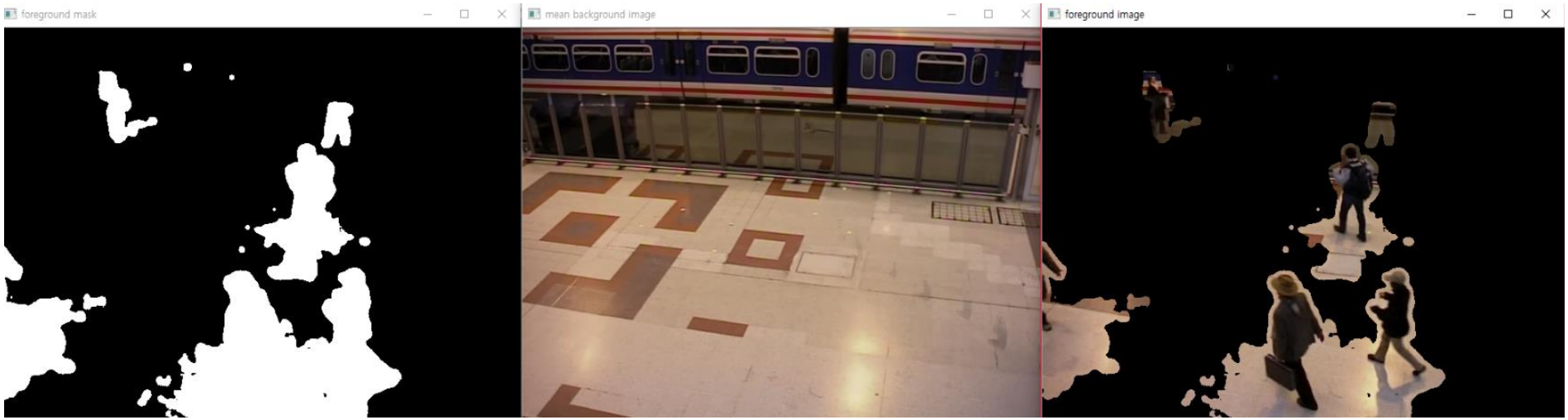
- Example code
  - openCV MoG2

```
int main() {
	Ptr<BackgroundSubtractor> bg_model = createBackgroundSubtractorMOG2();
	Mat image, foregroundMask, backgroundImg, foregroundImg;
	VideoCapture cap("background.mp4");

	while (true) {
		cap >> image;
		resize(image, image, Size(640, 480));

		if (foregroundMask.empty())
			foregroundMask.create(image.size(), image.type());

		bg_model->apply(image, foregroundMask);
		GaussianBlur(foregroundMask, foregroundMask, Size(11, 11), 3.5, 3.5);
		threshold(foregroundMask, foregroundMask, 10, 255, THRESH_BINARY);
		foregroundImg = Scalar::all(0);
		image.copyTo(foregroundImg, foregroundMask);
		bg_model->getBackgroundImage(backgroundImg);
```

Background subtraction

- Example code
  - openCV MoG2

```
            imshow("foreground mask", foregroundMask);
            imshow("foreground image", foregroundImg);

            if (!backgroundImg.empty()) {
                imshow("mean background image", backgroundImg);
            }
            waitKey(33);
        }
}
```



Background subtraction

- Counting the number of objects

```
int main() {
	Mat gray = imread("contours.jpg", 0);
	Mat result;
	threshold(gray, result, 180, 255, THRESH_BINARY_INV);
	vector<vector<Point>> contours;
	vector<Vec4i>hierarchy;
	findContours(result, contours, hierarchy, CV_RETR_EXTERNAL, CV_CHAIN_APPROX_SIMPLE);

	putText(result, format("contour count: %d", contours.size()), Point(50, 80),
	FONT_HERSHEY_SIMPLEX, 1, Scalar(128), 4);

	imshow("contours", result);
	waitKey(0);
}
```