Laboratory of Image Processing

Abandoned Object Detection

Pier Luigi Mazzeo

pierluigi.mazzeo@cnr.it

Institute of Applied Sciences and Intelligent Systems

ScienceApp

National Research Council of Italy

Abandoned Object Detection

- This example shows how to track objects at a train station and it determines which ones remain stationary. Abandoned objects in public areas concern authorities since they might pose a security risk. Algorithms, such as the one used in this example, can be used to assist security officers monitoring live surveillance video by directing their attention to a potential area of interest.

- This example illustrates how to use the BlobAnalysis System object to identify objects and track them. The example implements this algorithm using the following steps:

- Extract a region of interest (ROI), thus eliminating video areas that are unlikely to contain abandoned objects.
  - Perform video segmentation using background subtraction.
  - Calculate object statistics using the blob analysis System object.
  - Track objects based on their area and centroid statistics.
  - Visualize the results.

Initialize Required Variables and System Objects

```
roi = [100 80 360 240];
% Maximum number of objects to track
maxNumObj = 200;
% Number of frames that an object must remain stationary before
an alarm is raised
alarmCount = 45;
% Maximum number of frames that an abandoned object can be
hidden before it is no longer tracked
maxConsecutiveMiss = 4;
areaChangeFraction = 13;      % Maximum allowable change in
object area in percent
centroidChangeFraction = 18; % Maximum allowable change in
object centroid in percent
% Minimum ratio between the number of frames in which an object
is detected and the total number of frames, for that object to
be tracked.
minPersistenceRatio = 0.7;
% Offsets for drawing bounding boxes in original input video
PtsOffset = int32(repmat([roi(1), roi(2), 0, 0],[maxNumObj 1]));
```

Initialize Required Variables and System Objects

- Create a VideoFileReader System object to read video from a file.

```
hVideoSrc = vision.VideoFileReader;
hVideoSrc.Filename = 'viptrain.avi';
hVideoSrc.VideoOutputDataType = 'single';
```

- Create a ColorSpaceConverter System object to convert the RGB image to Y'CbCr format.

```
hColorConv =
vision.ColorSpaceConverter('Conversion', 'RGB to
YCbCr');
```

- Create an Autothresholder System object to convert an intensity image to a binary image.

```
hAutothreshold =
vision.Autothresholder('ThresholdScaleFactor',
1.3);
```

Initialize Required Variables and System Objects

- Create a MorphologicalClose System object to fill in small gaps in the detected objects.

```
hClosing = vision.MorphologicalClose('Neighborhood',
strel('square',5));
```

- Create a BlobAnalysis System object to find the area, centroid, and bounding box of the objects in the video.

```
hBlob = vision.BlobAnalysis('MaximumCount', maxNumObj,
'ExcludeBorderBlobs', true);
hBlob.MinimumBlobArea = 100;
hBlob.MaximumBlobArea = 2500;
```

Initialize Required Variables and System Objects

- Create System objects to display results.

```
pos = [10 300 roi(3)+25 roi(4)+25];
hAbandonedObjects = vision.VideoPlayer('Name', 'Abandoned
Objects', 'Position', pos);

pos(1) = 46+roi(3); % move the next viewer to the right

hAllObjects = vision.VideoPlayer('Name', 'All Objects',
'Position', pos);

pos = [80+2*roi(3) 300 roi(3)-roi(1)+25 roi(4)-roi(2)+25];

hThresholdDisplay = vision.VideoPlayer('Name', 'Threshold',
'Position', pos);
```

Video Processing Loop

- Create a processing loop to perform abandoned object detection on the input video. This loop uses the System objects you instantiated above

```
firsttime = true;
while ~isDone(hVideoSrc)
    Im = step(hVideoSrc);

    % Select the region of interest from the original video
    OutIm = Im(roi(2):end, roi(1):end, :);

    YCbCr = step(hColorConv, OutIm);
    CbCr  = complex(YCbCr(:,:,2), YCbCr(:,:,3));

    % Store the first video frame as the background
    if firsttime
        firsttime = false;
        BkgY      = YCbCr(:,:,1);
        BkgCbCr   = CbCr;
    end
```

Video Processing Loop

```
%Background Subtraction
SegY    = step(hAutothreshold, abs(YCbCr(:,:,1)-BkgY));
SegCbCr = abs(CbCr-BkgCbCr) > 0.05;


 % Fill in small gaps in the detected objects
    Segmented = step(hClosing, SegY | SegCbCr);

    % Perform blob analysis
    [Area, Centroid, BBox] = step(hBlob, Segmented);

% Call the helper function that tracks the identified objects and
returns the bounding boxes and the number
of the abandoned objects

[OutCount, OutBBox] = videoobjtracker(Area, Centroid, BBox,
maxNumObj, ...
areaChangeFraction, centroidChangeFraction, maxConsecutiveMiss,...
       minPersistenceRatio, alarmCount);
```

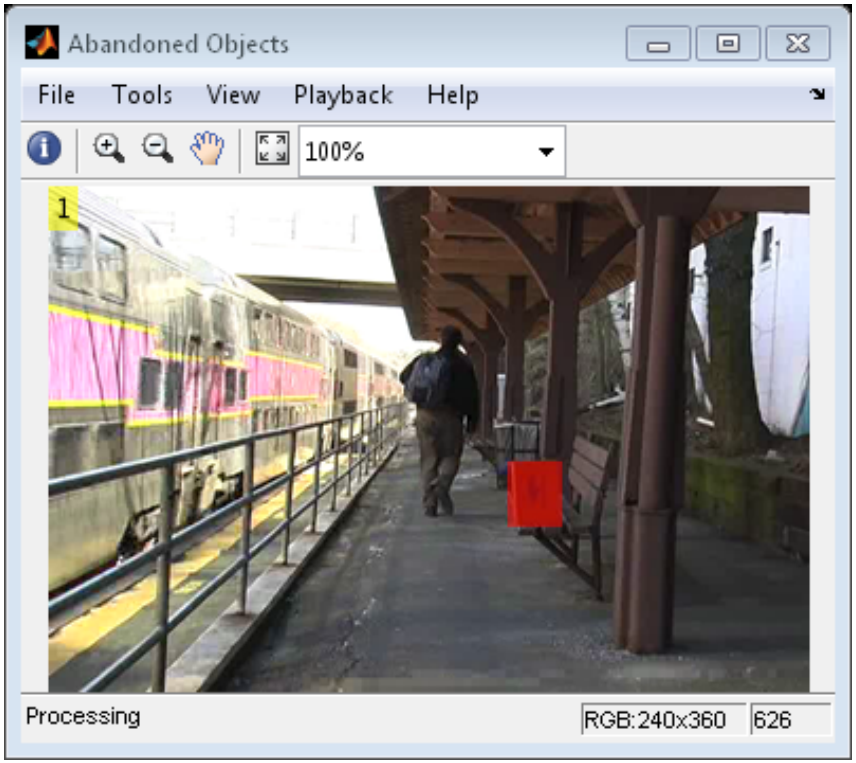
Video Processing Loop

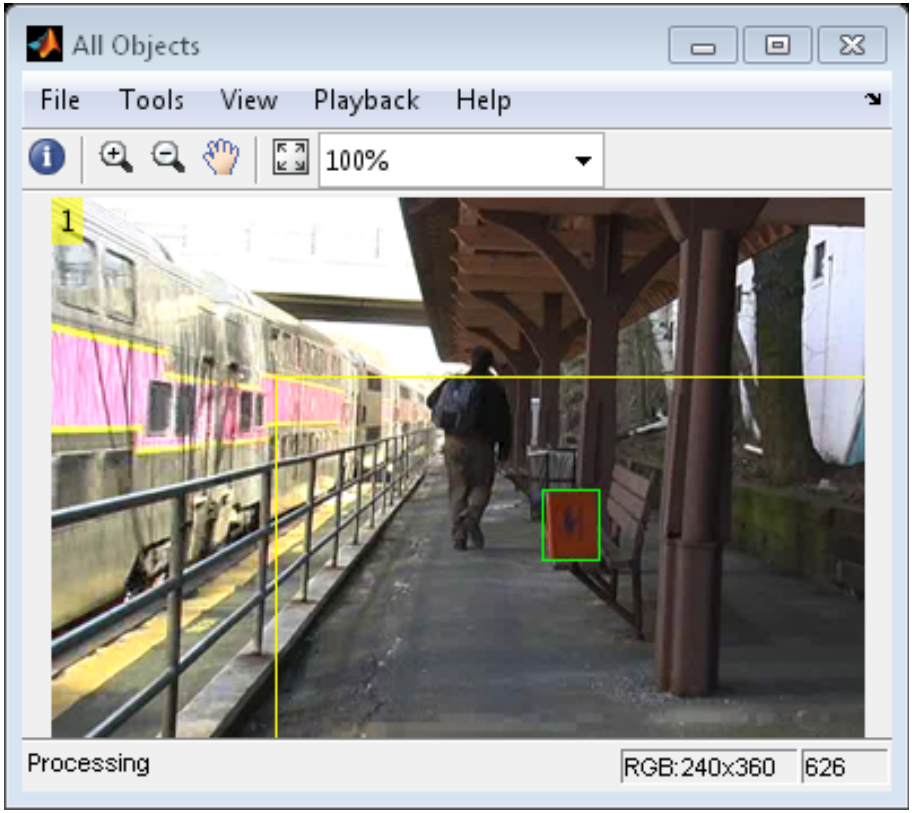
```
% Display the abandoned object detection results
    Imr = insertShape(Im,'FilledRectangle',OutBBox
+PtsOffset,...
        'Color','red','Opacity',0.5);
    % insert number of abandoned objects in the frame
    Imr = insertText(Imr, [1 1], OutCount);
    step(hAbandonedObjects, Imr);

    BlobCount = size(BBox,1);

    BBoxOffset = BBox + int32(repmat([roi(1) roi(2) 0  0],
[BlobCount 1]));
    Imr =
insertShape(Im,'Rectangle',BBoxOffset,'Color','green');
```

Video Processing Loop

```
 % Display all the detected objects

    % insert number of all objects in the frame
    Imr = insertText(Imr, [1 1], OutCount);
    Imr = insertShape(Imr,'Rectangle',roi);
    %Imr = step(hDrawBBox, Imr, roi);
    step(hAllObjects, Imr);

    % Display the segmented video
    SegBBox = PtsOffset;
    SegBBox(1:BlobCount,:) = BBox;
    SegIm = insertShape(double(repmat(Segmented,[1 1
3])),'Rectangle', SegBBox,'Color','green');
   step(hThresholdDisplay, SegIm);
end

release(hVideoSrc);
```

Results


Abandoned Objects
File Tools View Playback Help
100%
1
Processing
RGB:240x360
626


All Objects
File Tools View Playback Help
100%
1
Processing
RGB:240x360
626

Results

The Abandoned Objects window highlights the abandoned objects with a red box. The All Objects window marks the region of interest (ROI) with a yellow box and all detected objects with green boxes. The Threshold window shows the result of the background subtraction in the ROI.