UNIVERSITÀ
DEGLI STUDI
FIRENZE

MEDIA INTEGRATION AND
COMMUNICATION CENTER
Visual Information and Media Lab

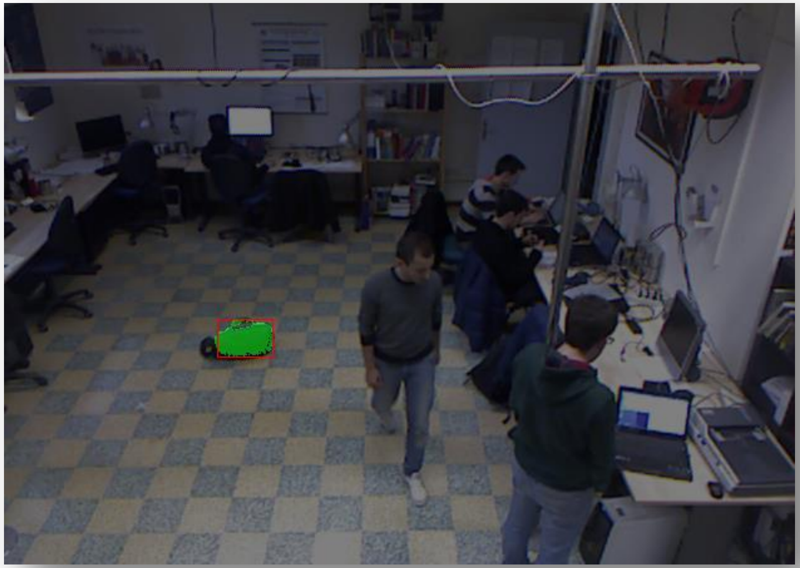
Left luggage detection
June 13th, 2014

Course of Multimedia Databases – a.a. 2013-14

Andrea Rizzo, Matteo Bruni

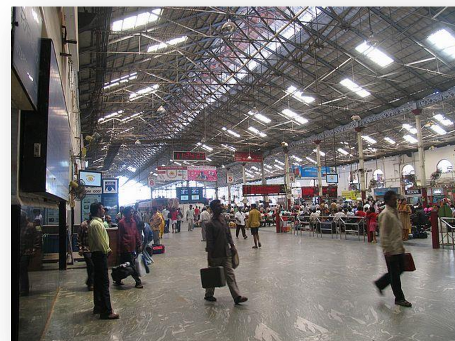
Contents

- Introduction
- Approach
  - Intensity processing
  - Depth processing
  - Combination of proposals
- Technologies
- Development
- Test
- Performance



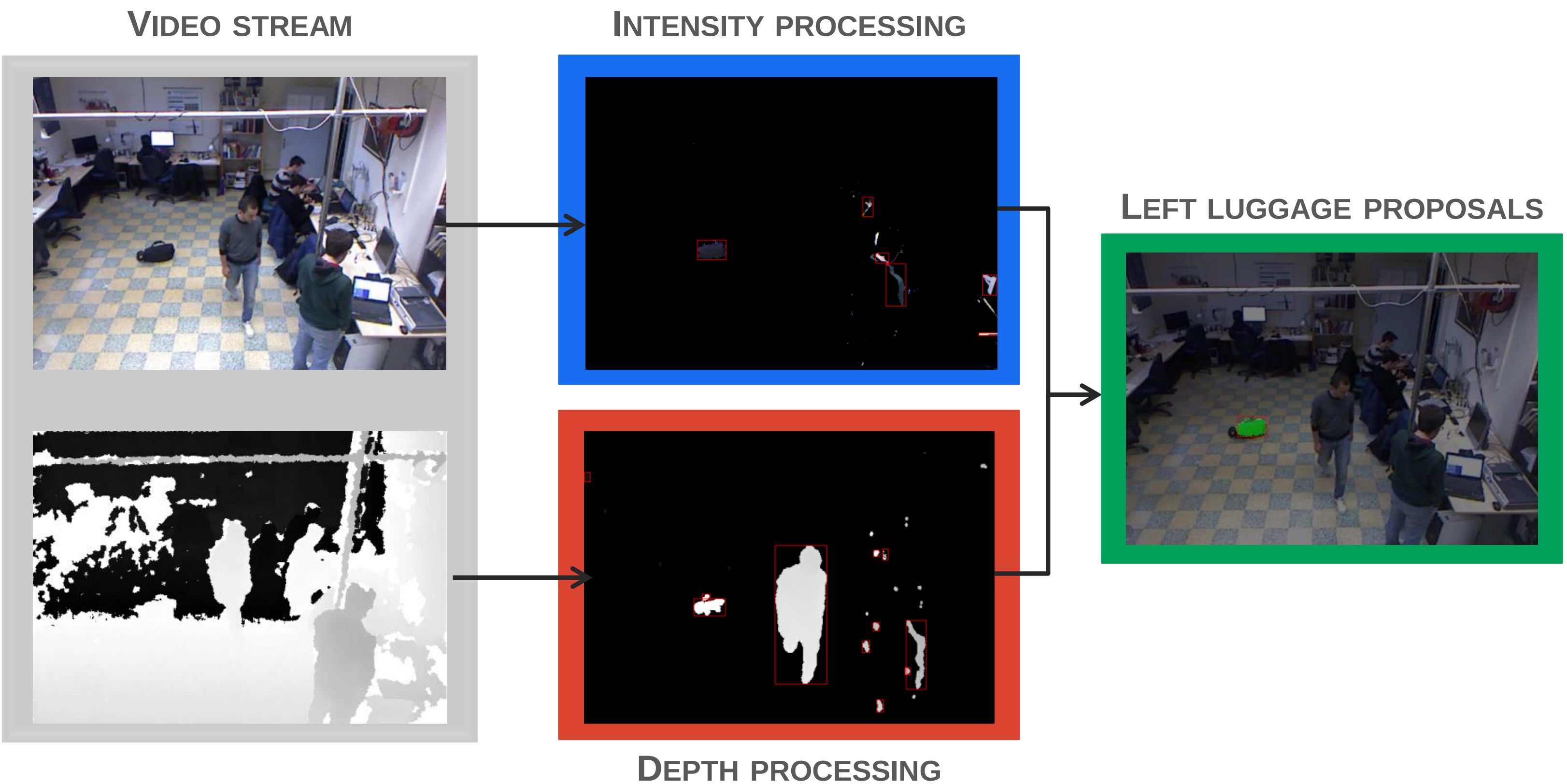
Left luggage detection

- Detection of abandoned items in public places:
  - Airports
  - Train station
- Issues:
  - Shadows
  - Occlusions
  - Illumination changes
  - Clutter



Pipeline

- It is based on ***Reliable Left Luggage Detection Using Stereo Depth and Intensity Cues*** – C. Beleznai, P. Gemeiner and C. Zinner[1] of Austrian Institute of Technology



Intensity background modelling

- The background model is computed using the **Zivkovic method** [2]:
  - Gaussian Mixture Model (GMM)
  - Select dynamically the number of components
- Left luggages are detected over time with the **dual foregrounds model** [3]:
  - Two background models are computed:
    - B_L: long-term background $\rightarrow F_L$
    - B_S: short-term background $\rightarrow F_S$
    - $\alpha_L < \alpha_S$
      - e.g.: $\alpha_L = 0.001$ and $\alpha_S = 0.01$

Dual foreground

- Using F_L and F_s we have four cases for each pixel:

$F_L(x,y)$	$F_S(x,y)$	
0	0	Background
0	1	Background pixel that was occluded
1	0	**Static object**
1	1	Foreground

- We aggregate the detection into an image $E(x,y)$
  - It aims to remove noise in the detection;
  - If $E(x,y) \geq max_e$, the pixel is marked as abandoned item.

Aggregator update rule

- The aggregator update rule is the following:

Update rule $\boldsymbol{E(x,y)}$	**Condition**
$E(x,y)+1$	$F_L(x,y)=1 \wedge F_S(x,y)=0$
$E(x,y)-PENALTY$	$F_L(x,y)\neq 1 \vee F_S(x,y)\neq 0$
max_e	$E(x,y)>max_e$
0	$E(x,y)<0$

- A set of bounding box of intensity-based proposals is obtained.

Depth processing

- Background model is built over time using the **accumulate running average method:**

$$B_t = (1 - \alpha) \cdot B_{t-1} + \alpha \cdot I_t$$

- The spatial changes are accumulated in an aggregator.
- The accumulator entry that have a number of observations above a threshold is marked as abandoned item.
- A set of bounding box of depth-based proposals is obtained.

ORIGINAL FRAME

$\alpha = 0.1$

$\alpha = 0.01$

Combination of proposals

- The two sets of proposals generated by intensity and depth-based detection are merged using:

$$R = \frac{\#pixel(P_d \cap P_i)}{\#pixel(P_d \cup P_i)}$$

- where P_d and P_i are two overlapping bounding boxes in depth and intensity.
- If $R \geq 50\%$ an abandoned item is detected.

Technologies

- Video capture with Kinect device
  - RGB camera:
    - resolution 640×480
    - 8 bit quantization
  - Depth sensor:
    - resolution 640×480
    - 11 bit quantization
  - USB 2.0 interface
- OpenKinect
  - Open source library
  - Multiplatform
- OpenCV

OPENKINECT

python

PyCharm

OpenCV

Intensity processing

- We implement an intensity processing module.
- The RGB processing routine can be summed as following:

```
# get next video frame
rgb.current_frame = cam.get_image()

# get rgb dual foreground (long and short term)
rgb.compute_foreground_masks(rgb.current_frame)

# update rgb aggregator
rgb.update_detection_aggregator()

# extract bounding box proposals
rgb_proposal_bbox = rgb.extract_proposal_bbox()
```

- A bounding box contains a set of connected pixel in which the aggregator value is max_e.
- The bounding boxes with area less than 50 pixel are filtered:
  - So small objects are discarded.

Depth processing

- The depth processing routine can be summed as following:

```
# get next depth frame (11-bit precision)
depth.current_frame = cam.get_depth_matrix()
depth.current_frame_holes = bg_models.compute_holes_mask_in_frame(depth.current_frame)

# get depth background
depth.update_background_model(depth.current_frame, depth.current_frame_holes)

# get depth foreground
depth.extract_foreground_mask_from_run_avg(depth.current_frame)

# apply opening to remove noise
depth.foreground_mask = bg_models.apply_opening(depth.foreground_mask, BG_OPEN_KSIZE,
                                                                        cv2.MORPH_ELLIPSE)
depth_proposal_bbox = depth.extract_proposal_bbox(depth.ACCUMULATOR)

# cut foreground with real values
foreground_depth_proposal = bg_models.cut_foreground(depth.current_frame, depth.foreground_mask)
```

- Is it that simple?   Would be nice!!!

Depth challenge

- The depth video stream is **noisy**:
  - We apply the opening morphological operator.
- The depth video stream is **not** defined everywhere:
  - it's available for the image regions that are close enough to the device;
  - for black objects the sensor can't measure the depth value;
  - A proper **management** of N/D pixel has been implemented.

Depth-based proposals

- The spatial changes over time are accumulated in an image aggregator:
  - If the aggregator exceeds a threshold is segmented[4] with a bounding box;
  - The spatial region is marked as left item proposal.
- We provide 3 methods to accumulate the depth changes:
  - IMAGE ACCUMULATOR
  - BOUNDING BOX ACCUMULATOR
  - BEST BOUNDING BOX ACCUMULATOR

Combination of proposals

- The intensity and depth-based proposals sets are merged by using:

$$R = \frac{\#pixel(P_d \cap P_i)}{\#pixel(P_d \cup P_i)}$$

- If $R \geq 50\%$ an abandoned item is detected:
  - The bounding box is saved in the set $Result_t$

Combination of proposals - improvements

- Since in the RGB foreground model the left luggage eventually become background in the long run the intensity-based proposals are discarded:
  - The last frame $frame_{t-1}$ is saved;
  - The last set of bounding boxes of proposals $Result_{t-1}$ is saved;
    - If the ratio R between a bounding box $bb_t \in Result_t$ and a bounding box $bb_{t-1} \in Result_{t-1}$ is over 50% then the two bounding box are considered the same;
    - Else if $R < 50\%$ we compute the **Normalized Cross-Correlation** C_N between the region of $frame_t$ bounded by the bounding box $bb_{t-1}$and the same region of $frame_{t-1}$
      - If $C_N > 0.90$ means that the regions is the same then we keep it;
      - Else we discard the bb_{t-1}
- The final proposals are segmented by using watershed algorithm instead region growing algorithm.

Test

- We have tested the application using more than 20 videos;
- The videos of interest are available for download through this link;
  - Video04: no clutter, no occlusions, one person, one left luggage; ✔✔✔
  - Video05: no clutter, two luggages, occlusions, standing people; ✔✔✔
  - Video12: no clutter, three luggages, illumination; ✔✔✘
  - Video13: clutter, two luggages, occlusions, illumination; ✔✘✘
  - Video16: clutter, two luggages, occlusions, illumination; ✔✘✘

Performance

- The approach used in this application is demanding in term of performance. The proposed framework runs at 5 fps on a modern notebook, just like the one proposed in the paper from which the authors were inspired.

References

- [1] C. Beleznai, P. Gemeiner and C. Zinner, Reliable Left Luggage Detection Using Stereo Depth and Intensity Cues
- [2] Z. Zivkovic and F. Heijden, Efficient adaptive density estimation per image pixel for the task of background subtraction
- [3] F. Porikli, Y. Ivanov and T. Haga, Robust abandoned object detection using dual foregrounds
- [4] Suzuki, S. and Abe, K., Topological Structural Analysis of Digitized Binary Images by Border Following. CVGIP 30 1, pp 32-46 (1985).

UNIVERSITÀ DEGLI STUDI FIRENZE

MEDIA INTEGRATION AND COMMUNICATION CENTER
Visual Information and Media Lab

Left luggage detection March xxth, 2014

Course of Multimedia Databases – a.a. 2013-14

Andrea Rizzo, Matteo Bruni