Robust Fusion Tracking

Md. Rasheduzzaman & Amirul Islam

North South University

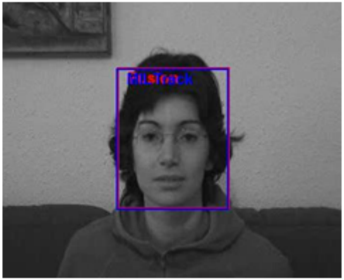
rashed091nsu@gmail.com
amirul@gmail.com

October 10, 2013

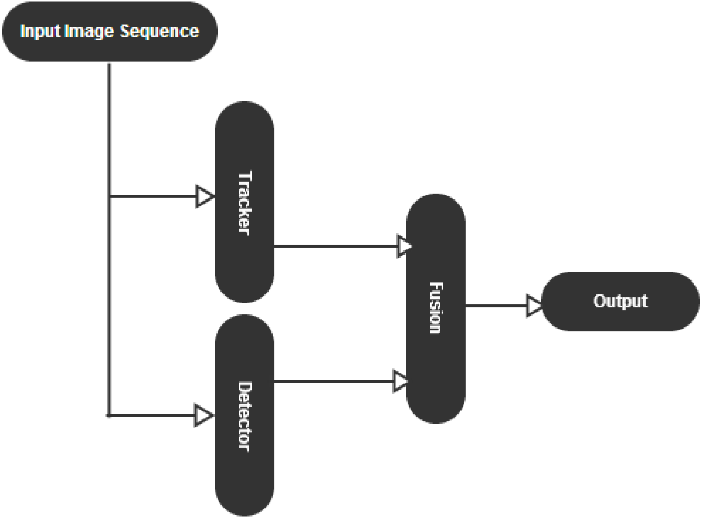
Overview

Problem Definition

The main objective of our project is to develop a robust tracking system that can detect partially occluded object in video sequence.

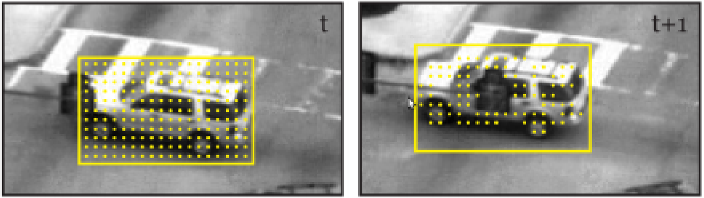


System Illustraion



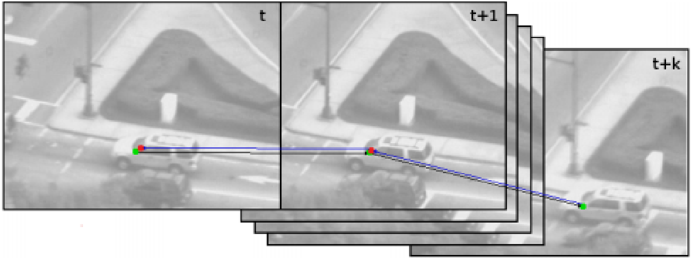
Tracking

- Lucas-Kanade(LK)tracker is iterative deferential method for optical flow estimation.
- For two given images $I(x)$ and $J(x)$ and a template image $T(x)$ in image $I(x)$, LK tracker estimate position of this template $T(x)$ in image $J(x)$.



Forward-Backward Tracking

- The idea of the forward-backward error measure lies in the observation that certain points cannot be re-tracked to their original location.



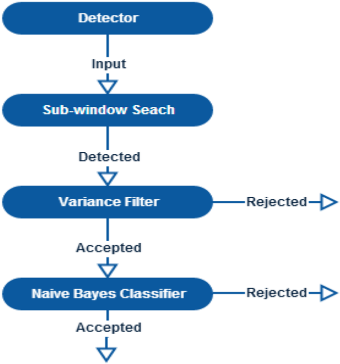
Median Flow Algorithm

Require: $Image\ I_i, I_{i-1}\ \ Patch\ BB_i$

for $i = 1$ **to** $All\ image\ sequence$ **do**
    $p_1 .... p_n \leftarrow select_{Points}(BB_i)$
    for $p_i = 1$ **to** $n\ points$ **do**
        $p'_i \leftarrow LK_{FORWOARD}(p_i)$
        $p''_i \leftarrow LK_{BACKWOARD}(p_1)$
        $e \leftarrow |p_i - p''_i|$
        $d \leftarrow NCC(W(p_i), W(p'_i))$
    end for
    $med_{NCC} \leftarrow median(d_1 .... d_n)$
    $med_{FB} \leftarrow median(e_1 .... e_n)$
    for $i = 1$ **to** $n\ points$ **do**
        if $d_i <= med_{NCC}\ and\ e_i <= med_{FB}$ **then**
            $target \leftarrow p_i$
        end if
    end for
    $predictBB(traget, BB_{i+1})$
end for

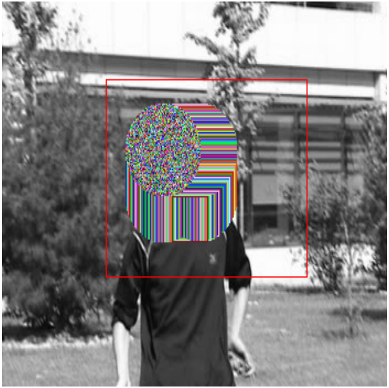
Detection

- Our detection system has two separate filter system.



Detection

- search radius and detected sub-windows



Variance Filter

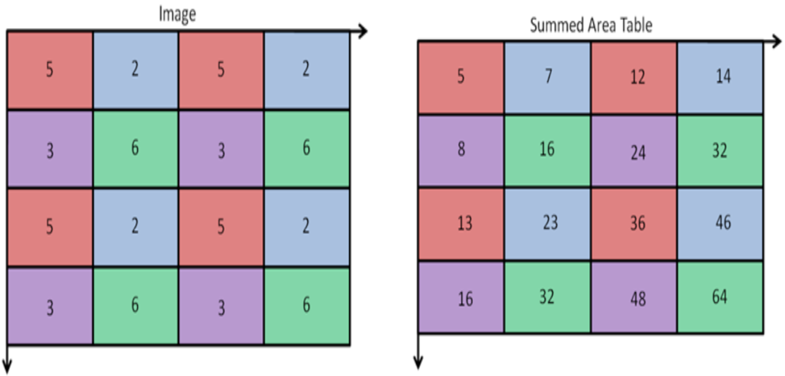
- First we calculate the threshold variance from first frame of input image sequences.

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2 \quad \mathit{with} \quad \mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

$$\sigma^2 = \frac{1}{n}\left(\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2\right) \tag{2}$$

- Both the sum in variance equation can be calculated from integral image with few memory lookup.

Integral Image



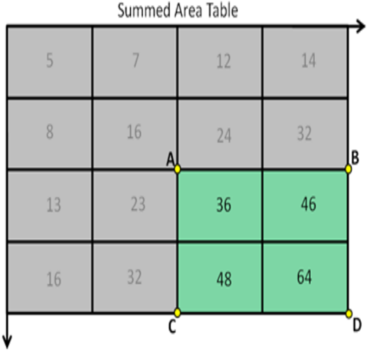
(a) Original Image (b) Integral Image

Figure : Integral Image Calculation.

Integral Image (cnt'd)

The sum within the rectangle *ABCD* in the figure can be computed as

$$S = I'(A) - I'(B) - I'(C) + I(D) \tag{3}$$



Haar-like Features

- Using the above described method of integral image we also calculate Haar-like features, Which is used as input feature for the naive Bayes classifier

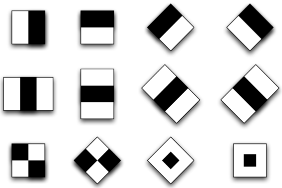


Figure : Set of Haar-like features

Detection Algorithm

Require: $Image\ I_i\ \ Output\ D_t$

$I' \leftarrow integral_{image}(I)$

$I'' \leftarrow integral_{image}(I^2)$

for $minrow$ **to** $maxrow$ **do**

  for $mincol$ **to** $maxcol$ **do**

    $box_x \leftarrow mincol$

    $box_y \leftarrow minrow$

    $box_{width} \leftarrow width$

    $box_{height} \leftarrow height$

    if $variance_{filter}(I', I'', box)$ **then**

      $B_i \leftarrow box$

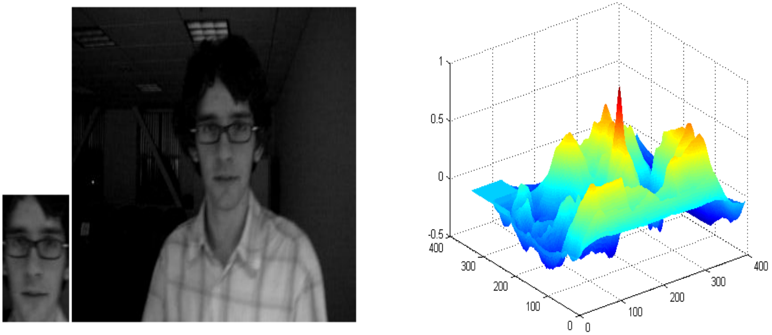
    end if

  end for

end for

$D_t \leftarrow Bayes_{classifier}(B)$

Template Matching



(a) Patch and Searched Image (b) Normalized cross- correlation plot

Figure : Template matching process.

Fusion Algorithm

Require: $BoundingBox\ R_t, D_t \ \ Output\ B_t$

$P^{+}_{R_t} \leftarrow template_{matching}(R_t)$

$P^{+}_{D_t} \leftarrow template_{matching}(D_t)$

if $P^{+}_{R_t} > P^{+}_{D_t}$ **then**

  $B_t \leftarrow D_t$

end if

if $P^{+}_{R_t} < P^{+}_{D_t}$ **then**

  $B_t \leftarrow R_t$

end if

Key-point Descriptor

- Scale invariant feature transform (SIFT).
- Speed up robust features (SURF).
- Maximally Stable Extremal Regions (MSER).
- Oriented BRIEF (ORB).
- Good Feature To Track (GFTT).
- Harris key-point.
- Random key-point.
- FAST Corner detection.
- Star key-point detector.

Testing Image Sequence

(a) David

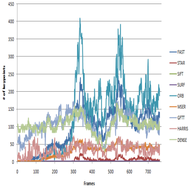
(b) Jumping

(c) Pedestrian

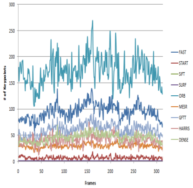
(d) Panda

Figure : First frame of test image sequences

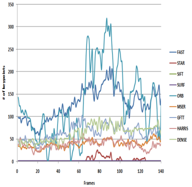
Key-points Found



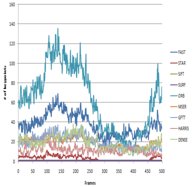
(a) David



(b) Jumping



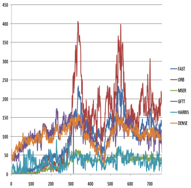
(c) Pedestrian



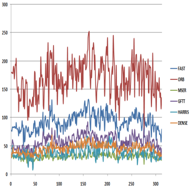
(d) Panda

Figure : First frame of test image sequences

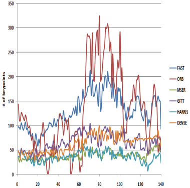
Forward-Backward Error



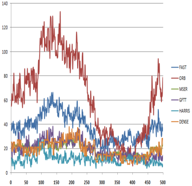
(a) David



(b) Jumping



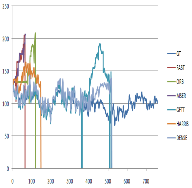
(c) Pedestrian



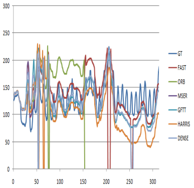
(d) Panda

Figure : First frame of test image sequences

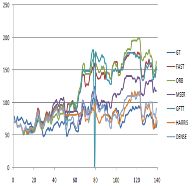
Central Location Error



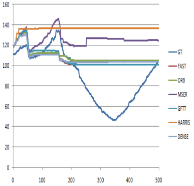
(a) David



(b) Jumping



(c) Pedestrian



(d) Panda

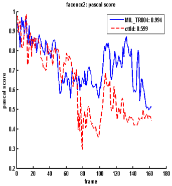
Figure : First frame of test image sequences

Tracking Testing

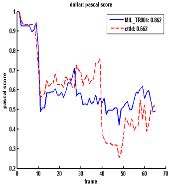
Video Clip	OAB	Semi Boost	Fragment	MIL Track	Fusion
Sylvester	25	16	11	11	9
David Indoor	49	39	46	23	15
Cola Can	25	13	63	20	20
Occluded Face	43	7	6	27	18
Occluded Face 2	21	23	45	20	21
Tiger 1	35	42	39	16	10
Tiger 2	33	61	37	18	15
Coupon	25	67	56	15	20

Figure : Central Location Error

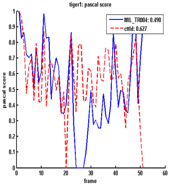
Tracking Precision plot



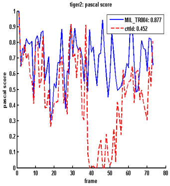
(a) Face2



(b) Dollar



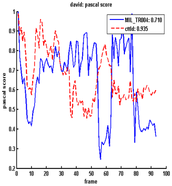
(c) Tiger1



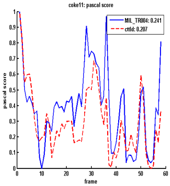
(d) Tiger2

Figure : Precision plot of test image sequences

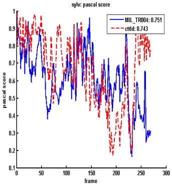
Tracking Precision plot



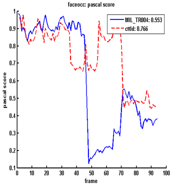
(a) David



(b) Coke



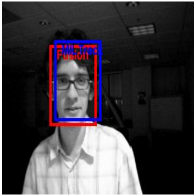
(c) Sylvester



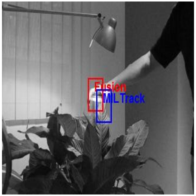
(d) Face1

Figure : Precision plot of test image sequences

Tracking Output



(a) David



(b) Coke

(c) Sylvester

(d) Face1

Figure : Ouput of MIL Track and Fusion tracking

Positive

- Tracks successfully from frame 1 .
- Learns weak transformations (such as small rotations) .
- Less sensitive to things that look similar.
- Robust to some partial occlusion.
- Good at face recognition.
- Better detection on smaller patch sizes.

Negative

- Sensitive to the objects spatial surroundings .
- Sensitive to transformations (such as rotation).
- Detection affected by object disappearance or complete occlusion.
- Affected by object scale.

Neutral

- Better at tracking slow motion than fast motion.
- Tracks some objects much better than others (those with more detail are probably tracked better as the features should be more selective on these)
- Better at tracking than re-detecting.

Feature work

- We have integrated a small part of our tracking system with Ar. Drone quadricopter.
- We want to navigate the drone autonomously and detect object.

Related Work

Babenko, B. and Ming-Hsuan Yang and Belongie, S. (2011)
Robust Object Tracking with Online Multiple Instance Learning
Pattern Analysis and Machine Intelligence, IEEE Transactions on 33(8), 1619-1632.

Adam, A. and Rivlin, E. and Shimshoni, I. (2006)
Robust Fragments-based Tracking using the Integral Histogram
Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on 3, 798-805.

Grabner, Helmut and Leistner, Christian and Bischof, Horst (2008)
Semi-supervised On-Line Boosting for Robust Tracking
Proceedings of the 10th European Conference on Computer Vision: Part I (14), 234–247.

Ross, David A. and Lim, Jongwoo and Lin, Ruei-Sung and Yang, Ming-Hsuan (2008)
Incremental Learning for Robust Visual Tracking
Int. J. Comput. Vision 77(1-3), 125–141.

The End