VALSE Webseminar

江苏省大数据分析技术重点实验室
Jiangsu Key Laboratory of Big Data Analysis Technology

Fast Compressive Tracking

张开华，南京信息工程大学

zhkhua@gmail.com

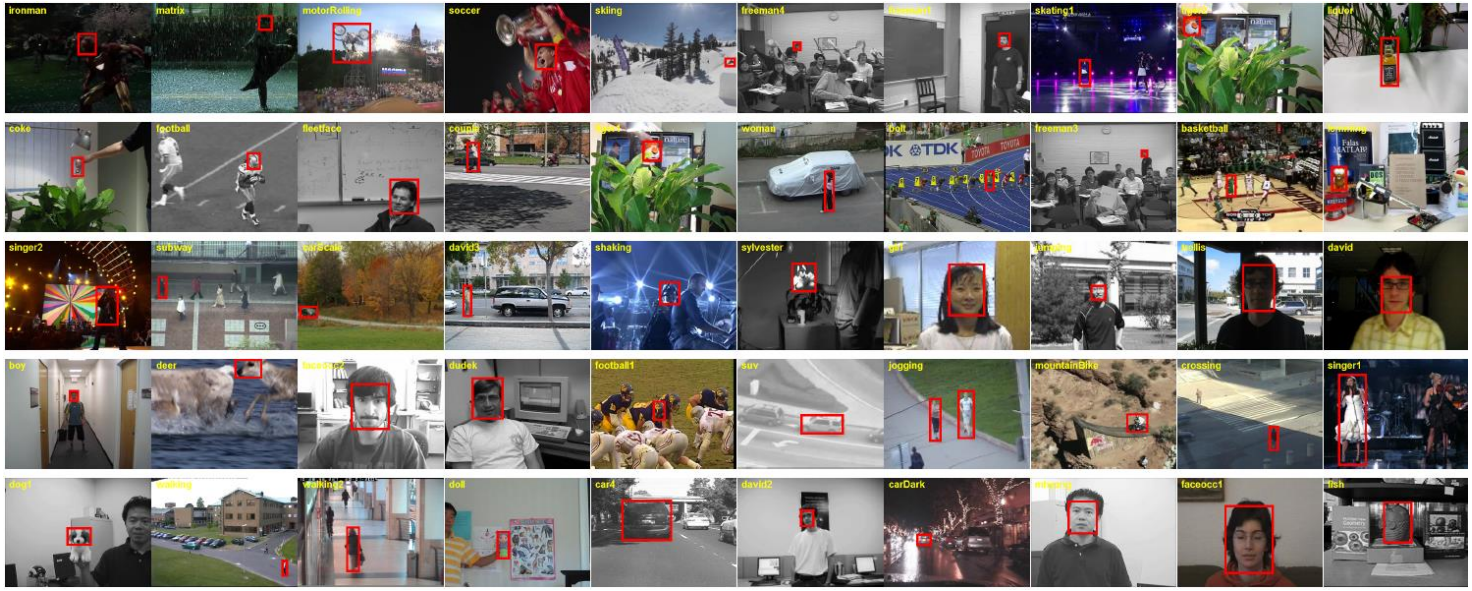
2015/1/28

Outline

- Visual tracking: a challenging task
- Tracking by detection
- The proposed algorithm
  - Fast compressive tracking

Visual Tracking

- Goal
  - To track an arbitrary object in a video given its initial location

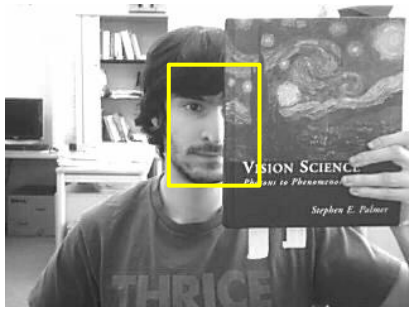


From wu yi et al., online object tracking: a benchmark, CVPR 2013

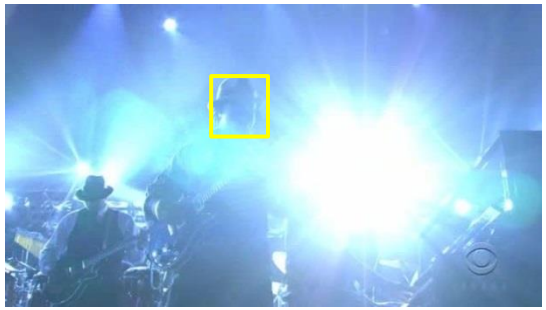
- Applications
  - Surveillance; Motion analysis; Object Recognition; Human-Computer Interaction (HCI); Traffic control; etc…

Challenges

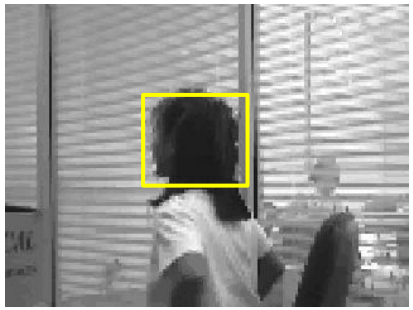
- Target appearance variation



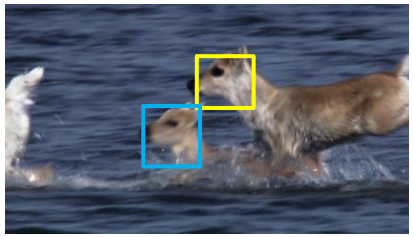
Occlusion



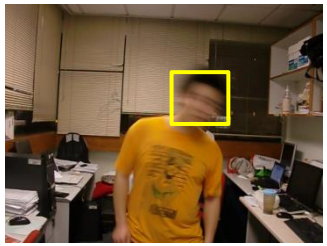
Illumination changes



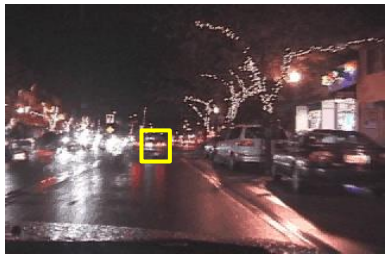
Rotation



Background distractor



Motion blur



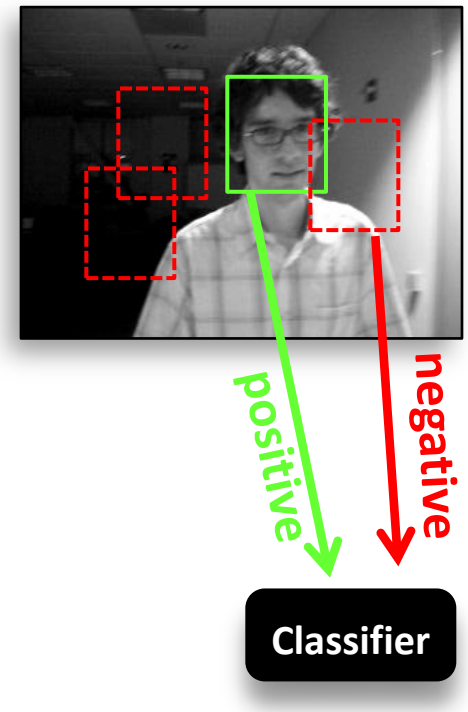
Small size

- Real-time requirement

Model complexity vs. efficiency!

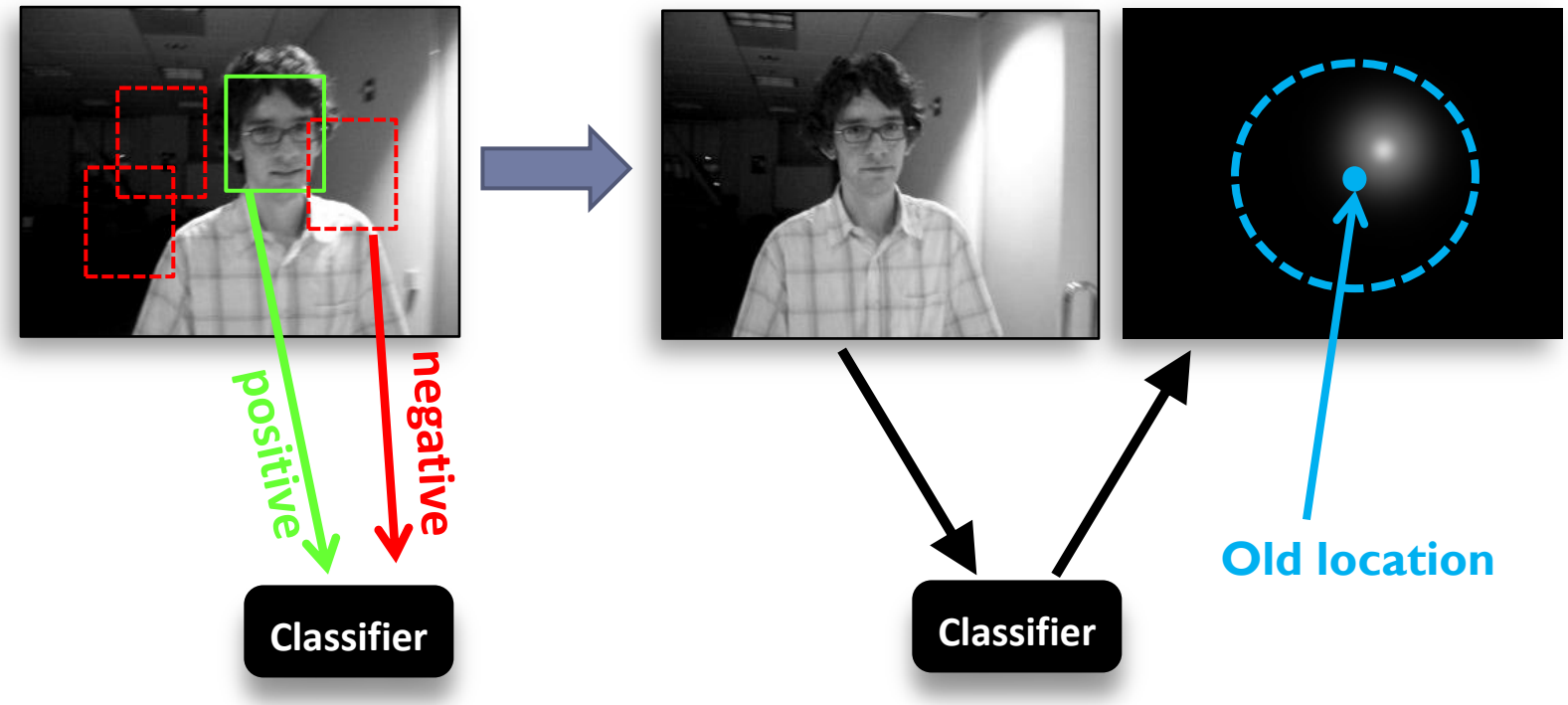
Tracking by Detection

- Train classifier



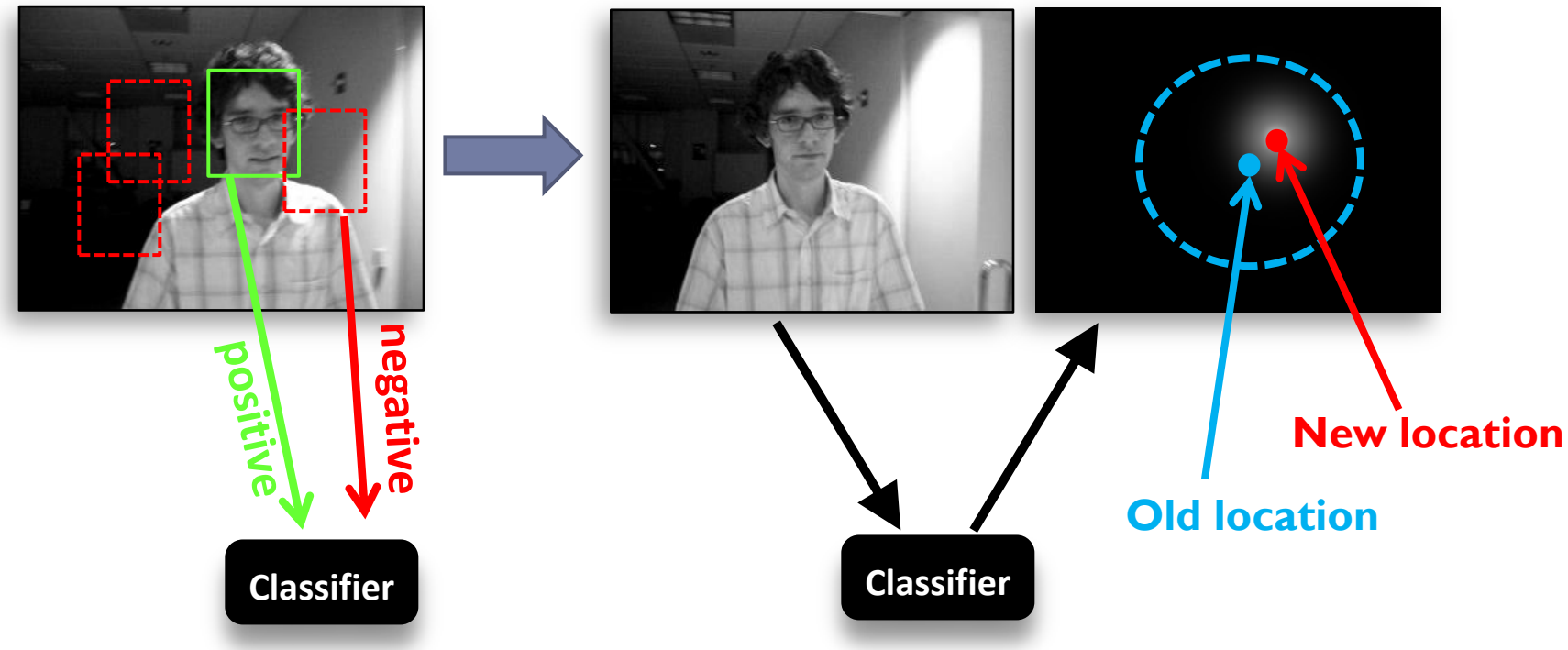
Tracking by Detection

- Evaluate classifier in some search windows



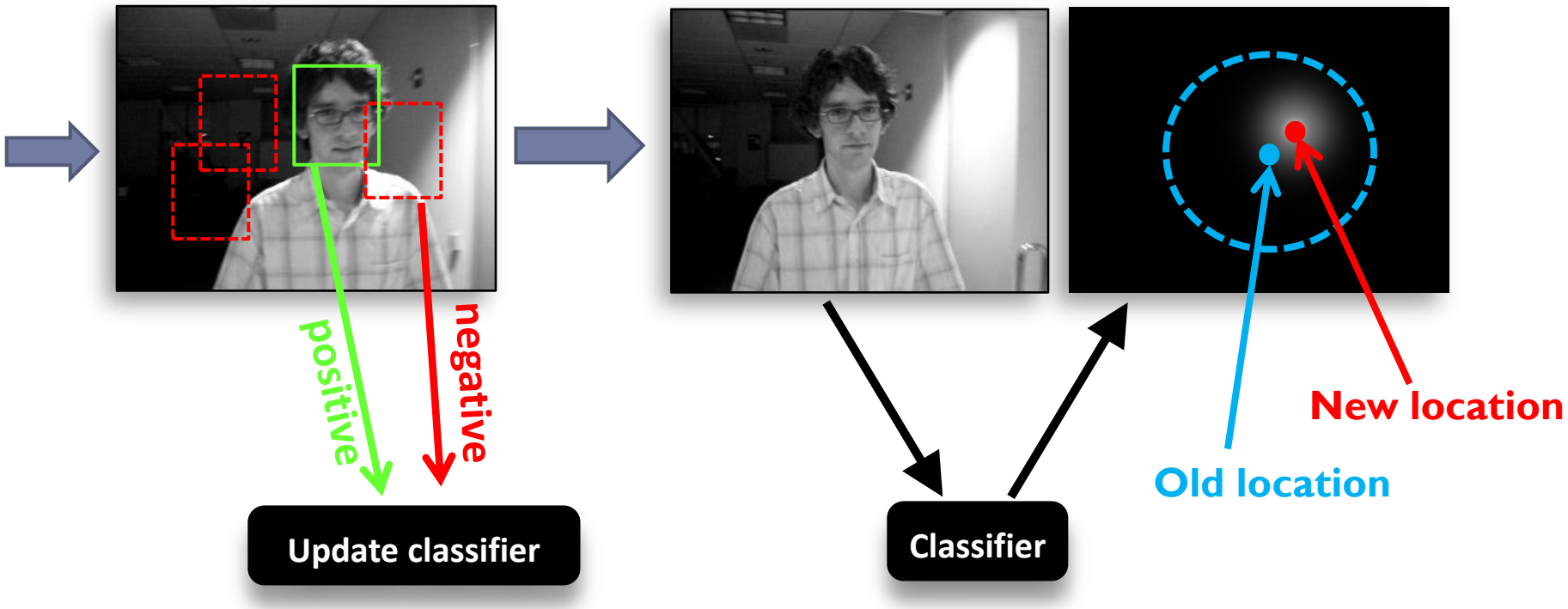
Tracking by Detection

- Find maximum classifier response



Tracking by Detection

- Repeat…



Two basic Components

- Target representation
  - Image intensity, color histogram, HOG, SIFT, Haar…
- Classifier
  - Naïve Bayes, SVM, Boosting, random forest…

Proposed Algorithms

- Fast Compressive Tracker
  - Kaihua Zhang, Lei Zhang, Ming-Hsuan Yang, Real-Time Compressive Tracking, *ECCV 2012*.
  - Kaihua Zhang, Lei Zhang, Ming-Hsuan Yang, Fast Compressive Tracking, *TPAMI 2014*.

Fast Compressive Tracker

- Target representation
  - Haar-like features
- Classifier
  - Naïve Bayes

Haar-like Features

- Definition

$$f = sum(Z \in A) + sum(Z \in C) - sum(Z \in B), \quad Z \text{ denotes image intensity}$$

Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In *CVPR* 2001.

Haar-like Features

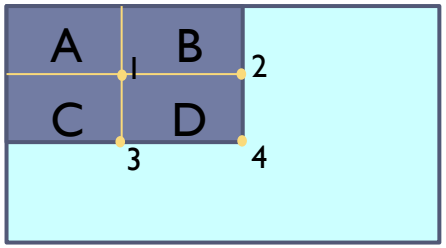
- Definition

$$f = sum(Z \in A) + sum(Z \in C) - sum(Z \in B), Z\ denotes\ image\ intensity$$

- Efficiently computed with integral image

$$sum(Z \in D) = 4 + 1 - (3 + 2)$$

Integral image



 Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In *CVPR* 2001.

Haar-like Features

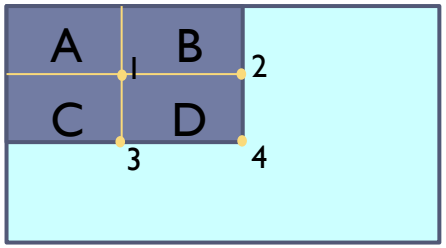
- Definition

$$f = sum(Z \in A) + sum(Z \in C) - sum(Z \in B), Z\ denotes\ image\ intensity$$

- Efficiently computed with integral image

$$sum(Z \in D) = 4 + 1 - (3 + 2)$$

Integral image



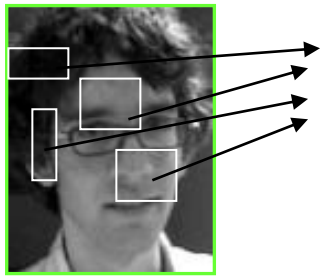
 Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In *CVPR* 2001.

Issues with Haar-like Features

- Numerous rectangles
  - All of the scales and the positions should be considered
  - For an image with size 24x24, the exhaustive set of rectangle features is over **180,000**
- Approximate method
  - Computational efficiency

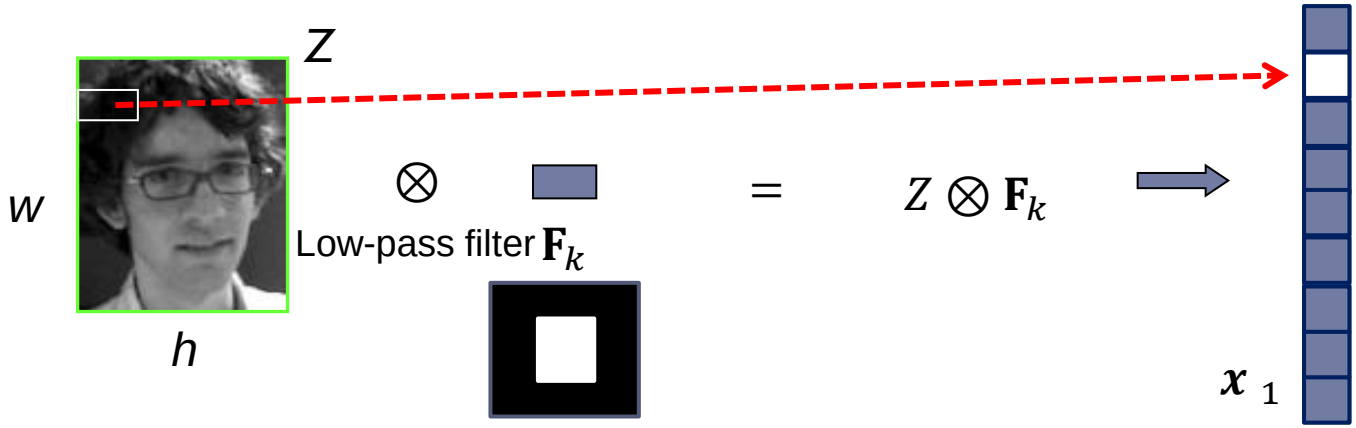
Compressive Tracking: Formulation

- Haar-like features



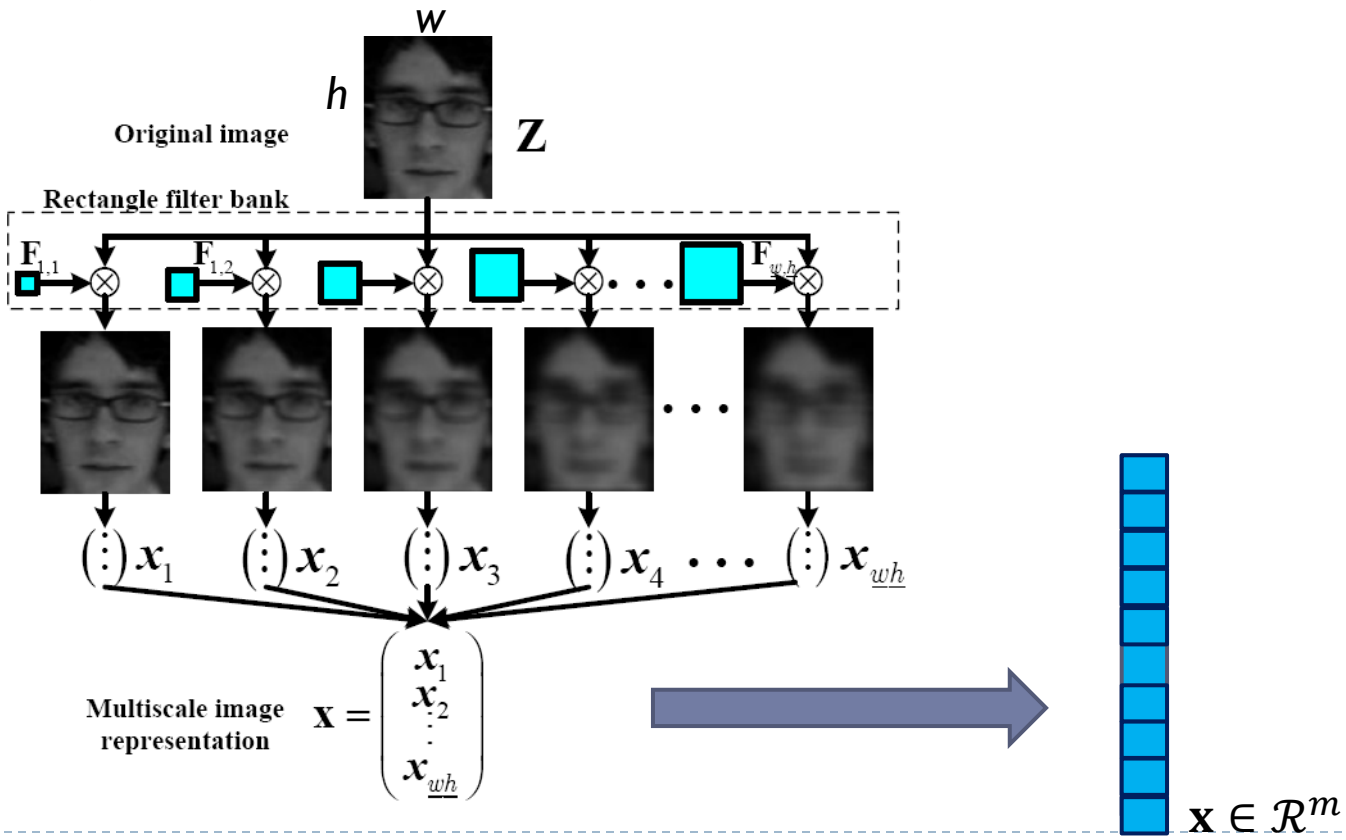
$$f_k = w_{k1}rec_{k1} + w_{k2}rec_{k2} + w_{k3}rec_{k3} + w_{k4}rec_{k4}$$

- Convolution to yield a rectangle feature



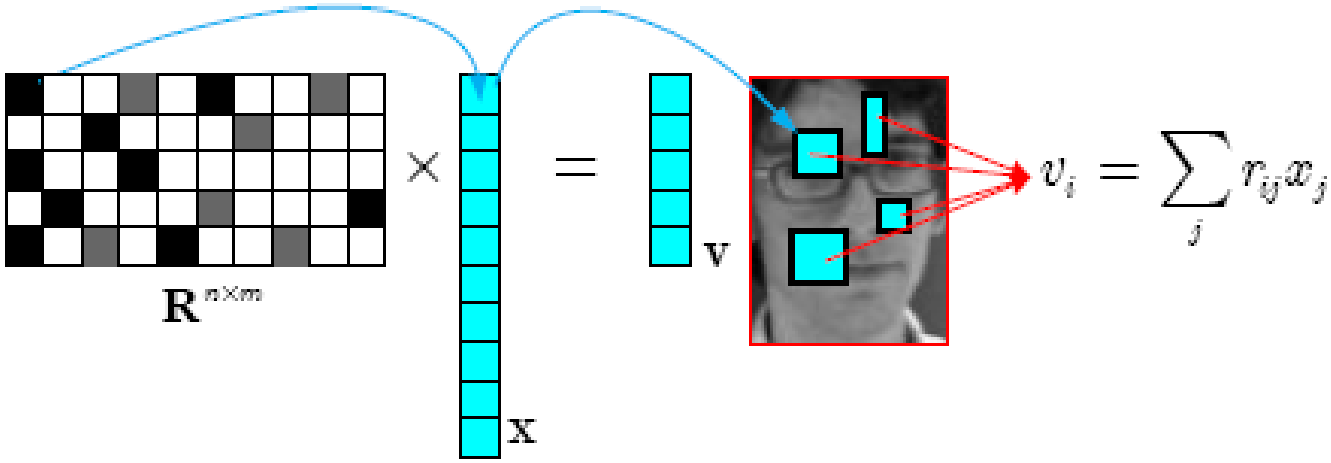
Compressive Tracking: Formulation

- High-dimensional multi-scale feature vector
  - The dimension of the final feature vector is $\mathrm{m} = (wh)^2$. If w=50, h=50, $m = 6.25 \times 10^6$!



Dimension Reduction

- Random projection



- **R** is a very sparse random matrix
  - The number of nonzero entries is less than $4 << m$
  - Only need to store nonzero entries ($< 4n, n = 100$)
  - Low memory requirement
  - Use integral images to efficiently compute features

How to Design Random Matrix **R** ?

- Sparsify **R** with compressive sensing theory

$$r_{ij} = \sqrt{\rho} \begin{cases} 1 \ \ with\, prob. \dfrac{1}{2\rho} \\ 0 \ \ with\, prob. 1 - \dfrac{1}{\rho} \\ -1\, with\, prob. \dfrac{1}{2\rho}. \end{cases}$$

- Set $\rho = o(m)$ to let each row of **R** only have at most 4 non-zero entries, e.g.,

$$\rho = \frac{m}{\log(m)}$$

The above very sparse random matrix satisfies compressive sensing theory (Restricted Isometry Property (RIP)).

Ping Li et al., very sparse random projections. KDD 2006.

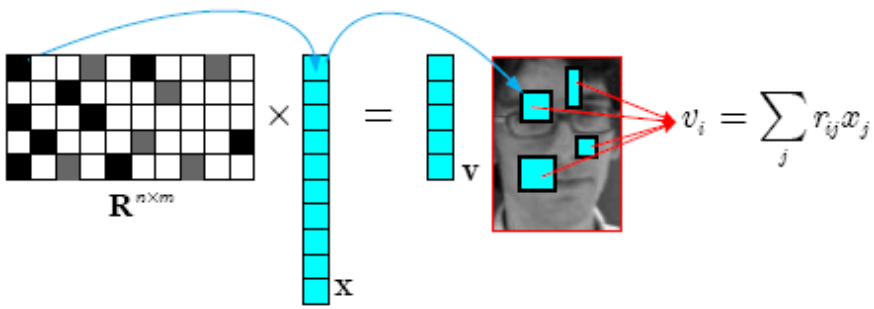
JL vs. RIP

Lemma 1 *(JL lemma) [18] Let Q be a finite collection of d points in $\mathbb{R}^m$. Given $0 < \epsilon < 1$ and $\beta > 0$ let $R \in \mathbb{R}^{n \times m}$ be a random matrix as (4) with $s = 3$ projecting from $\mathbb{R}^m$ to $\mathbb{R}^n$ with*

$$n \geq \left(\frac{4 + 2\beta}{\epsilon^2/2 - \epsilon^3/3} \right) \ln(d). \tag{2}$$

If $n \leq m$, then, with probability exceeding $1 - d^{-\beta}$, the following statement holds: For every $\mathbf{u}, \mathbf{v} \in Q$,

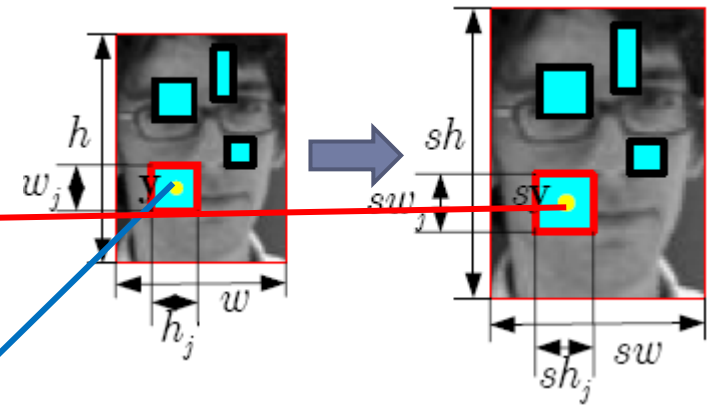
$$\boxed{(1-\epsilon)||\mathbf{u}-\mathbf{v}||_2 \leq ||R\mathbf{u} - R\mathbf{v}||_2 \leq (1+\epsilon)||\mathbf{u}-\mathbf{v}||_2.} \quad \text{RIP} \tag{3}$$



JL = RIP

Scale Invariant Property

- For the *j*-th feature, we have



$$
\begin{aligned}
x_j(s\mathbf{y}) &= \mathbf{F}_{sw_j,sh_j}(s\mathbf{y}) \otimes \mathbf{Z}(s\mathbf{y}) \\
&= \mathbf{F}_{sw_j,sh_j}(\mathbf{a}) \otimes \mathbf{Z}(\mathbf{a})|_{\mathbf{a}=s\mathbf{y}} \\
&= \frac{1}{s^2 w_i h_i} \int_{\mathbf{u}\in\Omega_s} \mathbf{Z}(\mathbf{a}-\mathbf{u})d\mathbf{u} \\
&= \frac{1}{s^2 w_i h_i} \int_{\mathbf{u}\in\Omega} \mathbf{Z}(\mathbf{y}-\mathbf{u})|s^2|d\mathbf{u} \\
&= \frac{1}{w_i h_i} \int_{\mathbf{u}\in\Omega} \mathbf{Z}(\mathbf{y}-\mathbf{u})d\mathbf{u} \\
&= \mathbf{F}_{w_j,h_j}(\mathbf{y}) \otimes \mathbf{Z}(\mathbf{y}) \\
&= x_j(\mathbf{y}),
\end{aligned}
$$

Classifier

- Naïve Bayes classifier.

$$H(\mathbf{f}) = log\left(\frac{\prod_{k=1}^{n} p(f_k|y=1)p(y=1)}{\prod_{k=1}^{n} p(f_k|y=0)p(y=0)}\right) = \sum_{k=1}^{n} log\left(\frac{p(f_k|y=1)}{p(f_k|y=0)}\right)$$

- PDF for each feature

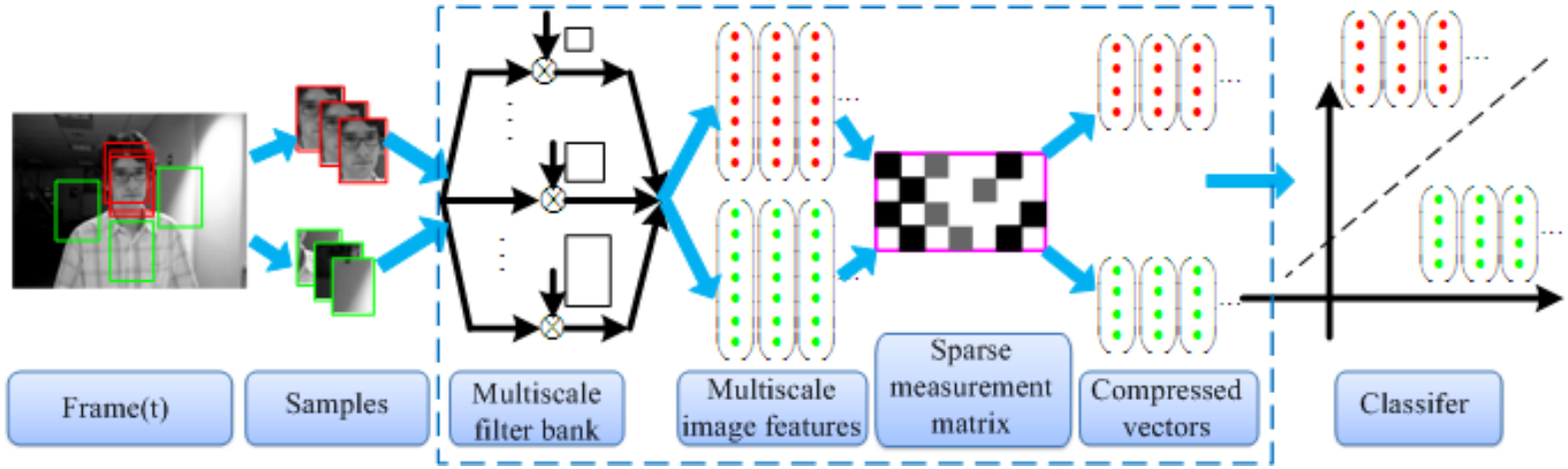
$$p(f_k|y=1) \sim N(\mu_k, \sigma_k),$$
$$p(f_k|y=0)\ has\ similar\ formulation.$$

- Parameter estimation

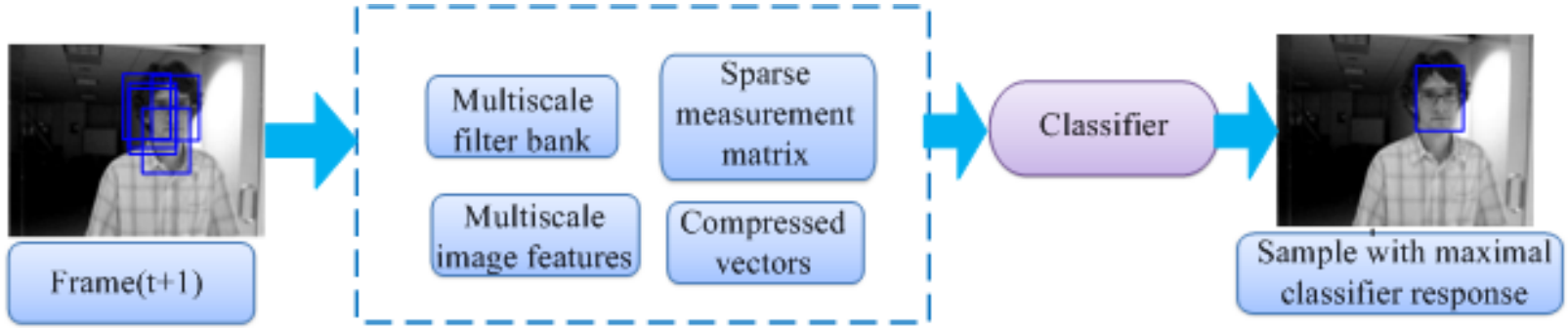
$$\mu_k \leftarrow \eta\mu_k + (1-\eta)\mu$$

$$\sigma_k \leftarrow \sqrt{\eta(\sigma_k)^2 + (1-\eta)\sigma^2 + \eta(1-\eta)(\mu_k - \mu)}$$

Algorithm Overview



(a) Update classifier at the t-th frame



(b) Tracking procedure at the (t+1)-th frame

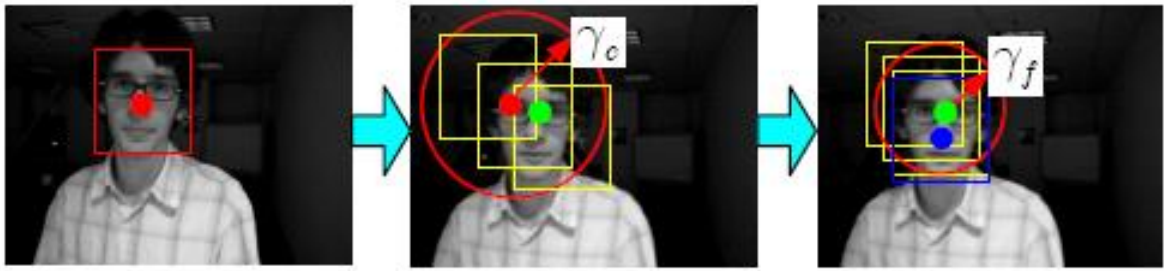
Computation Cost Reduction

- Which part takes the most cost ?
  - Training: 50 positive samples and 50 negative samples.
  - Detection: search radius is 25 pixels and step is 1 pixel. 1962 test samples!

Dense search, pixel by pixel

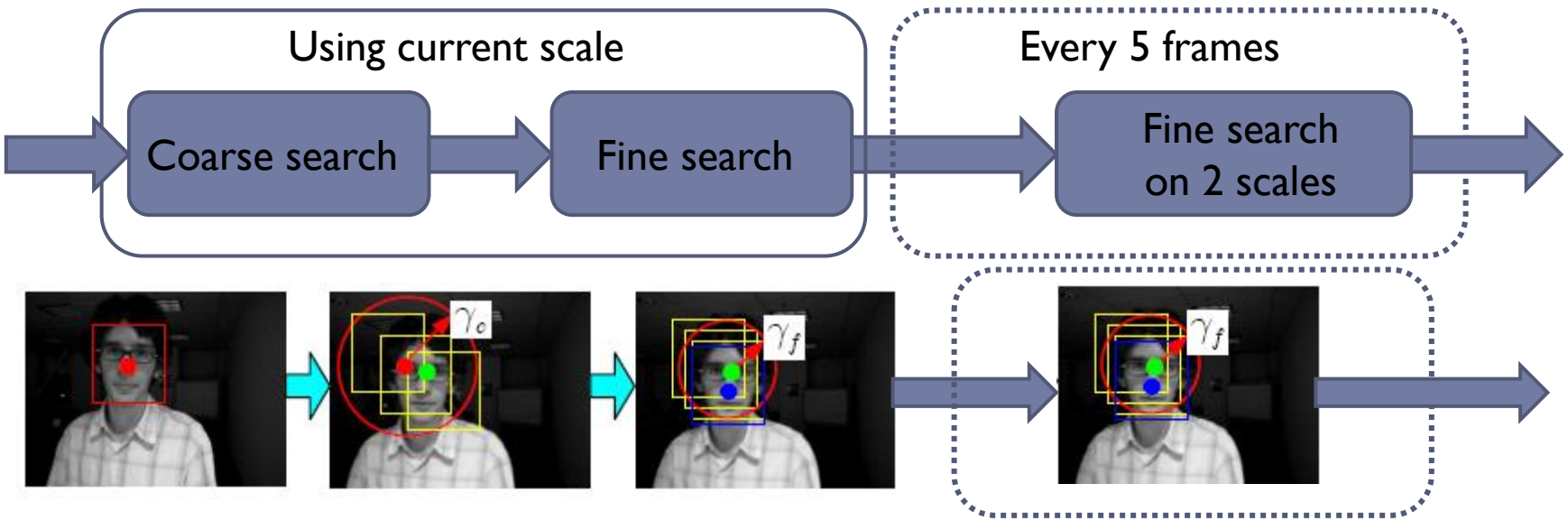
Computation Cost Reduction

- Coarse-to-fine search to reduce test samples
  - First, based on the former object location, make coarse search find the target location, and then based on the coarse target location, make fine search to find the final object location
  - Coarse search radius is 25, and search steps is 4 pixels. Fine search radius is 10, and search steps is 1 pixel.
  - Total search samples are 436<1962, significantly reducing cost.



Cost Reduction for Multi-scale Tracking

- First, coarse + fine search on one scale. There are 436 samples
- Second, every 5 frames, fine search on 2 scales with radius 10 and step 1 pixel. 628 samples.
- Total number is far less than our original method (1962)

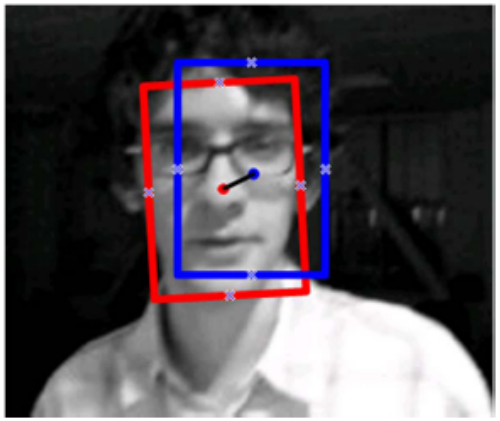


Experiments

- Quantitative comparison criteria
  - Center location error
  - Success rate

Experiments

- Quantitative comparison criteria
  - Center location error: the distance between the central locations of the tracked target and the manually labeled ground truth.
  - Success rate

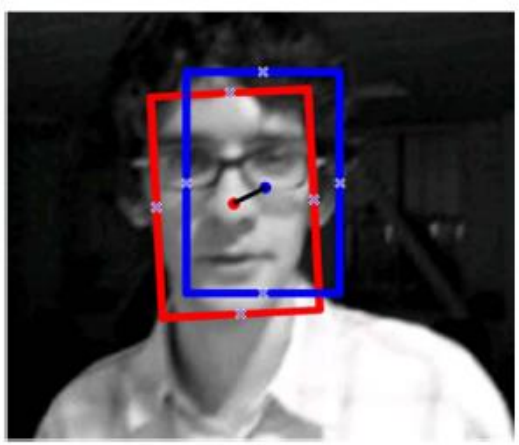


$$CenterError = \sqrt{\left(x_g - x_t\right)^2 + \left(y_g - y_t\right)^2}$$

Experiments

- Quantitative comparison criteria
  - Center location error: the distance between the central locations of the tracked target and the manually labeled ground truth.
  - Success rate: tracking in one frame is considered as success when the $OverlapRate \geq 0.5$.

$$SR = \frac{Number\ of\ the\ success\ ful\ frames}{Number\ of\ all\ the\ frames}$$



$$OverlapRate = \frac{area\left(R_G \cap R_T\right)}{area\left(R_G \cup R_T\right)}$$

Competing Methods

- Evaluate with 15 state-of-the-art tracking methods on 20 video sequences (8762 frames)
  - IVT [Ross et al, '08]
  - MILTrack [Babenko et al, '09]
  - VTD [Kwon et al, '10]
  - L1-tracker [Mei et al, '09]
  - MTT [Zhang et al, '12]
  - Struck [Hare et al, '11]
  - TLD [Kalal et al, '10]
  - SCM [Zhong et al, '12]
  - …

Center Location Errors

TABLE 3: Center location error (CLE)(in pixels) and average frame per second (FPS). **Bold** fonts indicate the best performance while the *italic* fonts indicate the second best ones. The total number of evaluated frames is 8,762.

Sequence	SFCT	FCT	CT	CS	Frag	OAB	SemiB	IVT	MIL	VTD	L1T	TLD	DF	MTT	Struck	CST	SCM	ASLA
Animal	**13**	*15*	16	271	100	62	26	207	32	16	122	125	252	17	19	*15*	16	**13**
Biker	**6**	12	**6**	176	107	*10*	14	111	44	86	89	166	76	68	95	53	227	109
Bolt	**8**	10	*9*	152	44	227	102	60	**8**	146	261	286	277	293	148	12	200	210
Cliff bar	*7*	**6**	8	69	34	33	56	37	14	31	40	70	52	25	46	*7*	99	49
Chasing	9	10	12	9	56	9	44	*5*	13	23	9	47	31	**4**	*5*	**4**	61	47
Coupon book	*5*	**4**	7	175	62	9	74	**4**	6	74	75	81	23	72	6	21	73	23
David indoor	*8*	11	14	72	73	57	37	**6**	19	**6**	17	*8*	56	125	64	18	150	57
Dark car	*7*	9	10	89	116	11	11	8	9	20	8	13	**6**	*7*	9	8	45	8
Football	*8*	13	14	43	144	37	58	10	13	**6**	39	15	33	9	26	17	200	207
Goat	20	*18*	103	137	140	71	77	94	109	92	88	103	86	99	22	**9**	75	94
Occluded face	**11**	*12*	16	29	57	36	39	14	17	36	17	24	22	19	15	13	24	20
Panda	**6**	**6**	10	157	56	8	9	58	*7*	61	9	16	64	47	11	46	156	9
Pedestrian	*7*	**6**	70	78	160	91	86	84	71	74	76	211	90	76	72	104	210	93
Skating	16	14	21	207	176	74	76	144	136	**9**	87	204	174	78	15	*10*	42	72
Shaking 1	13	*10*	14	119	55	22	134	122	12	**6**	72	232	*10*	115	24	21	47	*10*
Shaking 2	*14*	15	46	255	119	18	124	109	58	41	113	144	**7**	16	48	84	18	27
Sylvester	*9*	*9*	14	84	47	12	14	138	*9*	66	50	**8**	56	18	10	**8**	10	*9*
Tiger 1	*13*	23	25	48	39	42	38	45	27	**8**	37	24	30	61	**8**	25	146	49
Tiger 2	16	**10**	17	84	36	22	30	44	18	47	48	40	13	24	*11*	22	230	36
Twinnings	11	10	15	44	15	**7**	70	23	11	19	11	*8*	12	12	9	11	9	29
Average CLE	**9**	*10*	18	96	60	31	48	56	22	42	45	65	48	57	24	23	87	45
Average FPS	135	*149*	80	40	6	22	11	33	38	6	0.5	28	13	5	20	**362**	1	7

The proposed tracker runs at above 150 frames per second.

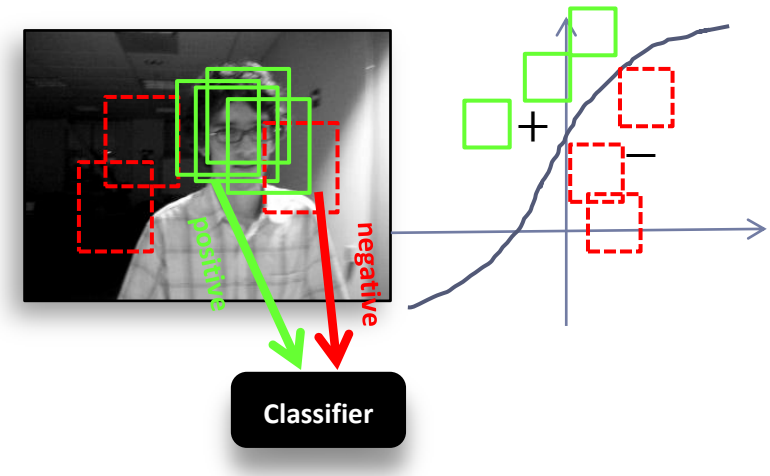
Success Rate

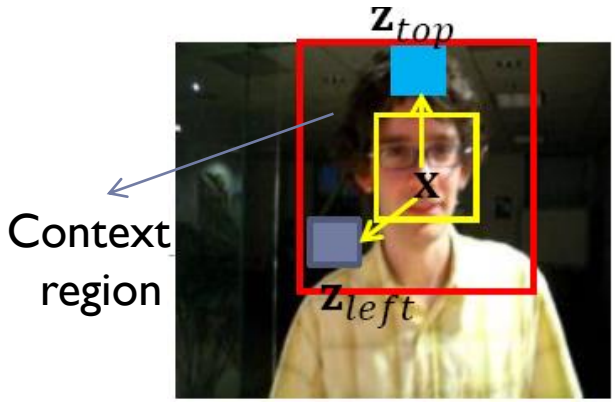
TABLE 2: Success rate (SR)(%). **Bold** fonts indicate the best performance while the *italic* fonts indicate the second best ones. The total number of evaluated frames is $8,762$.

Sequence	SFCT	FCT	CT	CS	Frag	OAB	SemiB	IVT	MIL	VTD	L1T	TLD	DF	MTT	Struck	CST	SCM	ASLA
Animal	*99*	92	96	4	3	17	51	4	83	96	6	37	6	87	96	**100**	98	96
Biker	**85**	35	*84*	5	3	66	39	10	1	15	3	2	6	9	9	9	5	10
Bolt	**99**	*94*	90	5	41	0	18	17	92	0	2	0	1	1	8	92	1	1
Cliff bar	95	**99**	89	24	24	66	24	47	71	53	24	63	26	55	44	*96*	41	40
Chasing	88	79	47	67	21	71	62	*91*	65	70	72	76	70	**96**	85	**96**	64	63
Coupon book	**99**	*98*	97	17	26	*98*	23	*98*	*98*	17	16	31	34	39	*98*	81	32	71
David indoor	**99**	*98*	94	8	8	32	46	*98*	71	*98*	83	*98*	51	41	33	66	30	34
Dark car	*75*	36	53	6	0	14	19	54	48	25	46	67	**78**	59	18	48	47	57
Football	*77*	76	74	35	26	31	17	64	*77*	**83**	35	59	56	67	62	69	17	7
Goat	75	*77*	26	26	14	46	43	37	27	39	24	48	44	39	59	**89**	57	37
Occluded face	*98*	**99**	**99**	39	54	49	41	96	97	79	96	87	78	88	*97*	**99**	76	93
Panda	**91**	84	*90*	1	9	83	71	11	80	7	63	34	13	11	43	15	29	71
Pedestrian	*82*	**83**	13	1	0	1	3	0	1	3	4	0	7	4	1	1	5	1
Skating	*96*	**97**	83	7	11	68	39	8	21	*96*	65	37	19	10	84	9	76	61
Shaking 1	72	84	80	9	25	39	30	1	83	*93*	3	15	84	2	48	36	54	**98**
Shaking 2	**97**	88	55	12	34	74	46	39	41	80	36	56	*95*	93	53	43	86	82
Sylvester	*83*	77	69	57	34	65	66	45	77	33	40	**89**	32	67	80	*83*	76	82
Tiger 1	49	52	50	62	19	24	29	8	34	*78*	18	40	36	25	**87**	42	31	14
Tiger 2	61	**72**	48	11	12	36	16	19	44	13	11	24	*65*	34	62	37	2	24
Twinnings	72	*98*	70	41	73	**99**	23	49	83	75	82	91	82	77	95	86	89	63
Average SR	**86**	*82*	73	29	33	56	43	49	68	51	47	57	50	50	64	66	51	58

Future Work

- Intrinsic problem in tracking by detection
  - To estimate the target location, not a classification problem that outputs binary labels

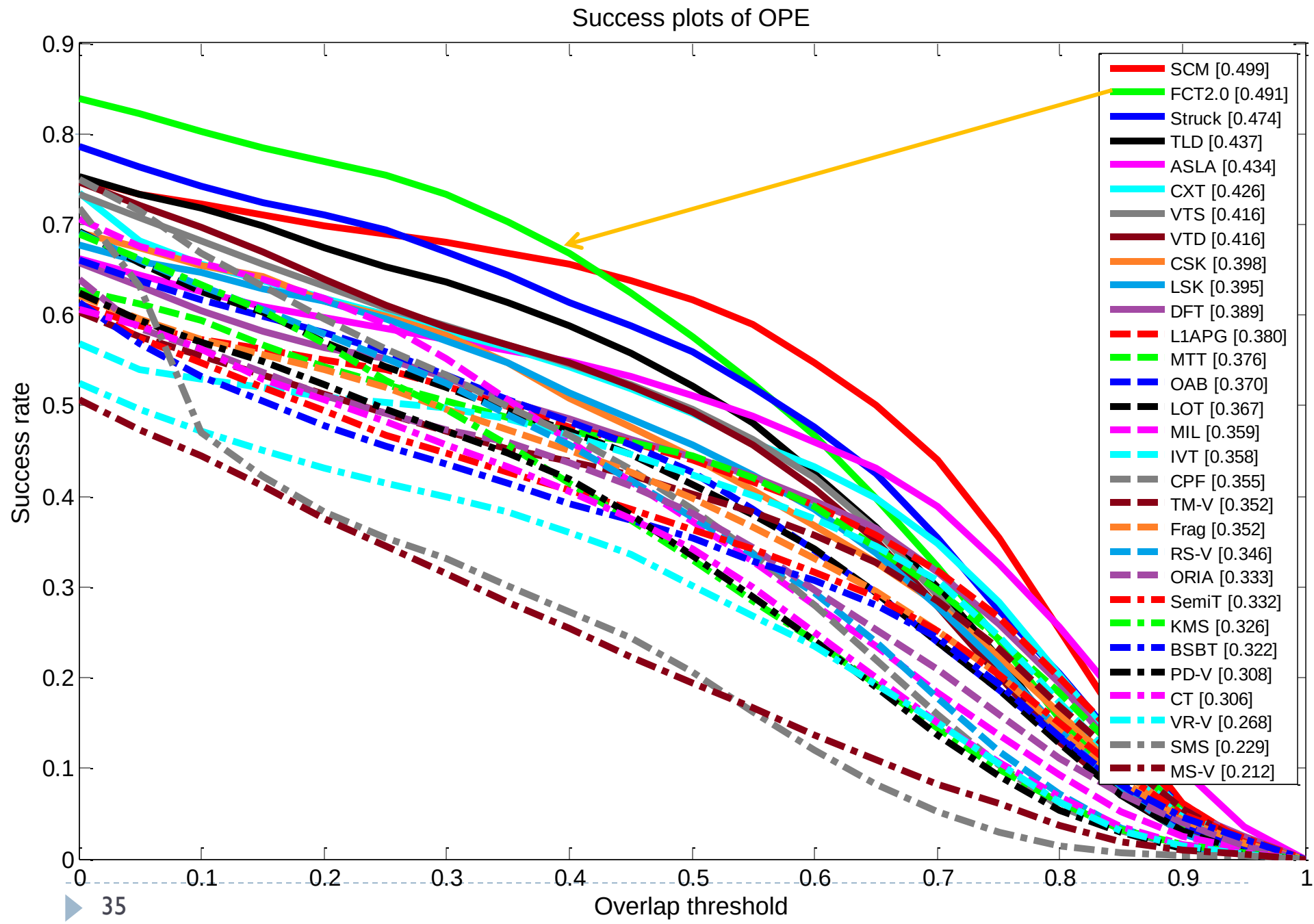




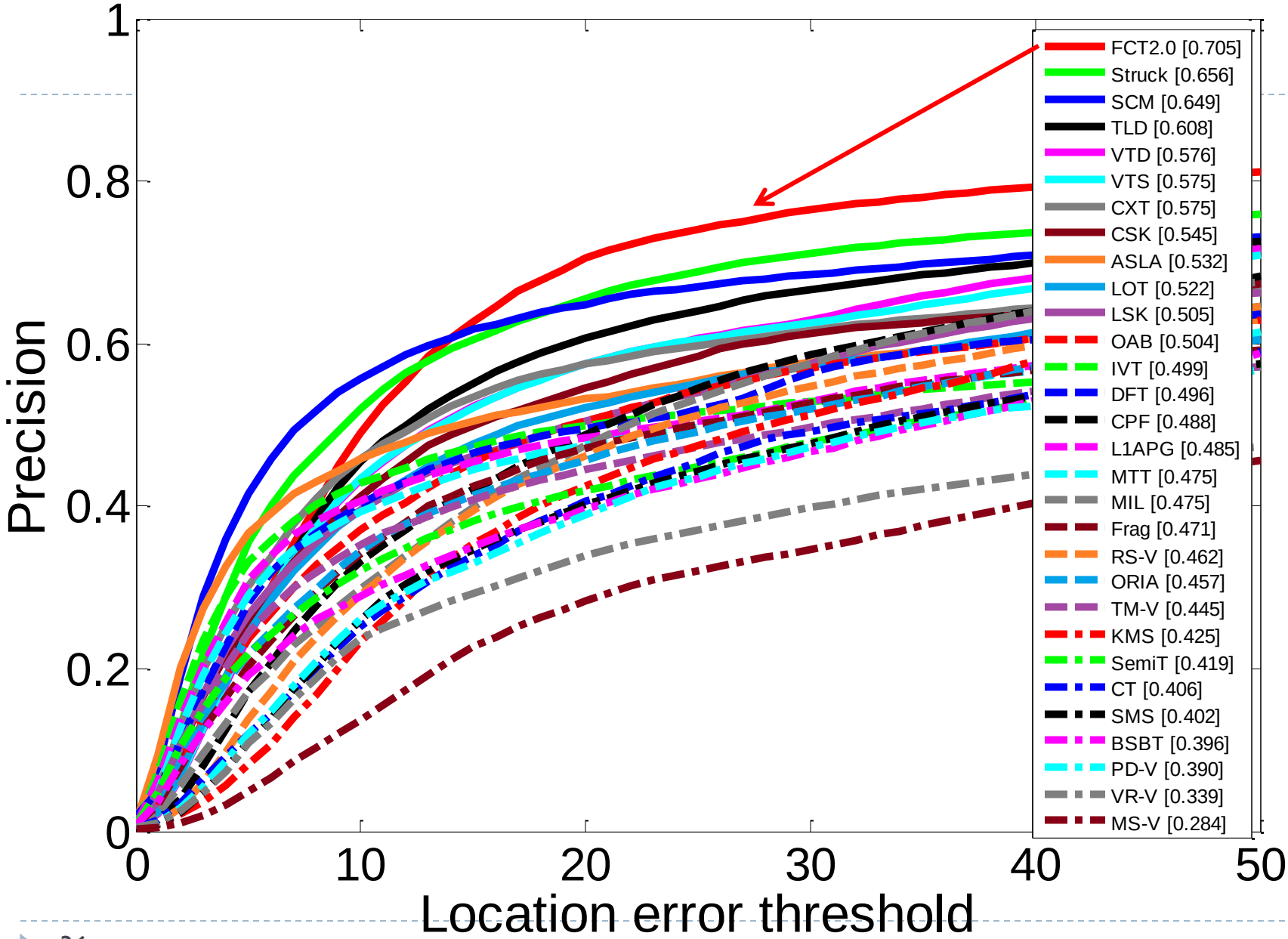
- The spatial relationship between the target and its surrounding background (context) does not fully taken into accounted. (refer to Kaihua Zhang et al., fast visual tracking via dense spatio-temporal context learning, ECCV2014)

Future Work

- Performance on benchmark
  - OPE score is about 32% that is far from 50% (top 3). However, we have some preliminary results that the OPE score can be reached at about 50% based on this simple framework.


Success plots of OPE
Success rate
Overlap threshold
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
SCM [0.499]
FCT2.0 [0.491]
Struck [0.474]
TLD [0.437]
ASLA [0.434]
CXT [0.426]
VTS [0.416]
VTD [0.416]
CSK [0.398]
LSK [0.395]
DFT [0.389]
L1APG [0.380]
MTT [0.376]
OAB [0.370]
LOT [0.367]
MIL [0.359]
IVT [0.358]
CPF [0.355]
TM-V [0.352]
Frag [0.352]
RS-V [0.346]
ORIA [0.333]
SemiT [0.332]
KMS [0.326]
BSBT [0.322]
PD-V [0.308]
CT [0.306]
VR-V [0.268]
SMS [0.229]
MS-V [0.212]

Precision plots of OPE



Thank you!

Code is available from http://www4.comp.polyu.edu.hk/~cslzhang/FCT/FCT.htm