Society of Tracking Parts

Elena Burceanu
eburceanu@bitdefender.com

1^{st} Conference on Recent Advances in Artificial Intelligence
RAAI 2017

Tracking

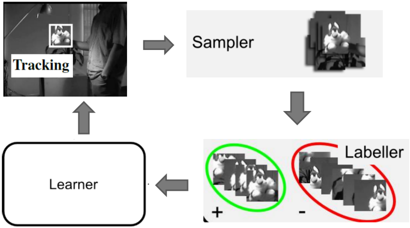
- the root for (m)any video application (e.g. medical-posture apps, self-driving cars, smart houses, surveillance)
- Types:
  - specific classes: pedestrians, cars, etc (integrate prior knowledge about a certain class)
  - generic (no special treatment, works with "undefined" objects)
- Challenges in tracking
  - integrate changes in appearance, but keep the model learned so far
  - problems: bkg clutter, fast or complex motion, deformation, etc
  - drifting: accumulating small errors (eg. bkg as positive sample)
  - decide bounding box based on detection map (weight and height)

Related Work

- Key-components in a tracker:
  - appearance model: features for parts [4, 11, 12, 16] or for all object
  - mathematical formulation, optimization
  - motion model
  - target region: bbox, ellipse [10], superpixels [18], blobs [5]
  - features: pixel level or region descriptors invariant to several transformations (but more expensive)
- Best current trackers [9]
  - CNNs (supervised or/and pre-trained: [3, 14, 15, 17])
    - problems with unseen/uncommon objects
    - small datasets (prone to overfit)
  - Correlation Filters (unsupervised: [1, 2, 6, 8])
    - Fourier formulation for "1 vs all" classifier
    - enables very fast computing for one obj descriptor

Bitdefender

Tracking by Detection



- use a discriminative appearance model
- sampler and labeler
  - chooses patches to update on (near previous detection)
  - ex. label = threshold on the distance from the max activation
- learner (appearance model)
  - binary classifier (foreground vs background)
  - trains with samples based on previous frame detection
- tracker
  - use the learner activations to choose the next object location
  - choose the maximum activation zone

Image from [7]

Learning a Robust Society of Tracking Parts

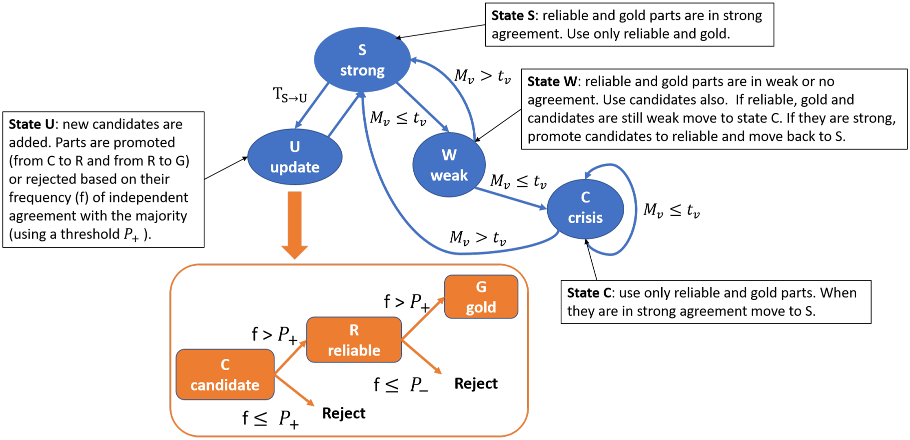
Elena Burceanu, Marius Leordeanu
eburceanu@bitdefender.com, marius.leordeanu@imar.ro

https://arxiv.org/abs/1705.09602

Intuition and Motivation

- Purpose:
  - adapt the current knowledge of the object model to continuous changes
  - don't forget valuable info
- Our solution
  - many parts of the object (like members in a company)
  - parts vote for the target object center (decision making)
  - each part has its own reliability and follows different rules
  - each part is validated in time (founders = initiators are by default valid)
- **Stability**: only "founders" and parts validated in time will vote (robust against noisy variations)
- **Adaptation**: over time, new candidates come in, old reliables get out
- **Never forget**: gold members (consistent behavior) are never removed

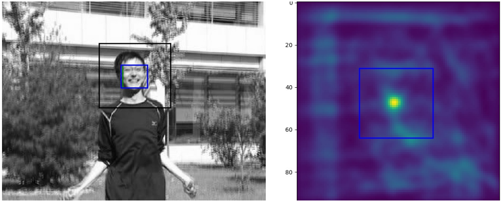
Society of Tracking Parts



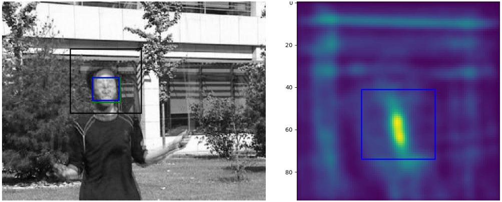
Algorithm details

- Voting
  - at each frame, each part contribute with its activation map, displaced by its relative to center position
  - add individual voting map and choose maximum activation (center)
  - maximum is proportional with the number of parts in agreement
- Update Parts
  - reliability: the frequency of part agreement with the majority
  - long time candidate and reliable, promote to reliables and gold
- Weak Tracker State
  - reliable and gold parts can't decide
  - allow new candidates (sampled from previous several frames) to vote
  - promote them to reliable if they were able to build a strong vote

Vote Examples I

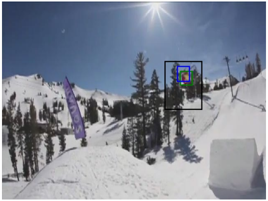
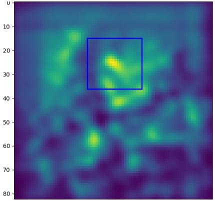


Strong vote. Votes for the object position concentrate in the same center.

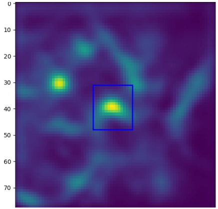


Voting map when the frame is moved (motion blur).

Vote Examples II

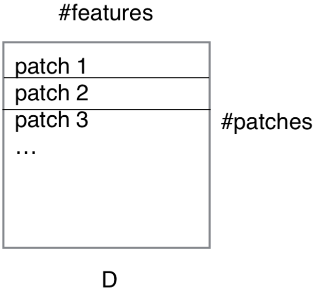
Weak voting in frames over the video.

Distractors in the frame.

Classifiers for Parts I

- Choose patches
  - centered on a thin grid over the searching zone
  - build data matrix D (with one patch per row)



- Pixel level features: HSV and edges for 0, $\pi/4$, $\pi/2$, π

Classifiers for Parts II

- Build linear "1 vs all" classifiers
  - $c_i = (\mathbf{D}^\top \mathbf{D} + \lambda \mathbf{I}_k)^{-1} \mathbf{D}^\top \mathbf{y}_i$
  - balance positive vs negatives: weighted linear ridge regression, in closed form: $\theta_i = (\mathbf{D}^\top \mathbf{W}_i \mathbf{D} + \lambda \mathbf{I}_k)^{-1} \mathbf{D}^\top \mathbf{W}_i \mathbf{y}_i$
  - $y_i^\top = \begin{bmatrix} 0 & 0 & \dots & 1 & \dots & 0 & 0 \end{bmatrix}$
  - **novelty**: for "1 vs all" case, the solution vector (θ_i) has the same direction with the one for the linear ridge regression (c_i), having the ratio $q_i = \frac{n}{1+(n-1)\mathbf{d}_i^\top \mathbf{c}_i}$, so $\theta_i = q_i c_i$
- Advantages
  - c_i can be computed in one operation for all "i"s (not possible for the weighted case)
  - bonus: invert a smaller matrix ($\mathbf{D}\mathbf{D}^\top$ instead of $\mathbf{D}^\top \mathbf{D}$)
  - $c_i = \mathbf{D}^\top (\mathbf{D}\mathbf{D}^\top + \lambda \mathbf{I}_n)^{-1} \mathbf{y}_i$ [1], invert a matrix with 2 orders of magnitude smaller
- we can compute all positive and all negative classifiers with **only one** (**small**) matrix inversion

[1] Matrix Inversion Lemma, see [13], Ch. 4.3.4.2

Bitdefender

Dataset and Metrics

- OTB50 dataset [19]
  - 50 videos, 51 targets
  - 100 - 3000 frames
  - labeled with difficulty: Illumination Variation, Scale Variation, OCClusion, DEFormation, Motion Blur, Fast Motion, Out-of-Plane Rotation/In-Plane Rotation, Out-of-View, Background Clutter, Low Resolution
  - Ground Truth rectangles for all frames
- Metrics
  - Average Overlap
    - IoU for area of bounding boxes (GT and predicted) $> 60\%$
    - mean over all frames
    - very sensitive (small values for ok trackers)
    - disadvantage for scale agnostic trackers
  - Mean Precision
    - distance between GT center and predicted center < 20 px
    - mean over all frames

Results

Algorithm	OPR	MB	BC	OCC	SV	IV	LR	OV	FM	DEF	All	FPS
OURS (STP)	75.9	72.5	68.4	78.5	76.9	73.2	63.7	73.4	69.8	80.1	78.7	30
KCF on HOG [13]	72.9	65	73.3	74.9	67.9	71.1	38.1	65	60.2	74	73.2	172
Struck [11]	59.7	55.1	58.5	56.4	63.9	55.8	54.5	53.9	60.4	52.1	65.6	20
KCF on pixels [13]	54.1	39.4	50.3	50.5	49.2	44.8	39.6	35.8	44.1	48	56	154
TLD [16]	59.6	51.8	42.8	56.3	60.6	53.7	34.9	57.6	55.1	51.2	60.8	28
ORIA [31]	49.3	23.4	38.9	43.5	44.5	42.1	19.5	31.5	27.4	35.5	45.7	9
MIL [2]	46.6	35.7	45.6	42.7	47.1	34.9	17.1	39.3	39.6	45.5	47.5	38
MOSSE [4]	39	24.4	33.9	39.7	38.7	37.5	23.9	22.6	21.3	36.7	43.1	615
CT [34]	39.4	30.6	33.9	41.2	44.8	35.9	15.2	33.6	32.3	43.5	40.6	64

- Mean Precision:
- 22% improvement over pixel level features (tracker vs features)
- 5% better than stronger features (HOG)

Algorithm	STP-S	STP-SW	STP-full
Precision (20px)	63.97	70.48	**78.7**

- results for: Strong-only, Strong-Weak only, full (tracker states)
- detect and recover from failures helps (+6%, +8%)

Bitdefender

Conclusions and Future work

- Conclusion
  - our tracker is functioning as a society of parts, each one with a different state and role
  - the solution for linear ridge regression classifier has the same vector direction with the weighted least squares (one sample versus all case)
  - SoTA results without using deep features (better emphasize the value of our tracker, not the greatness of the features)
  - online ("pure") unsupervised learning from video (unlike CNNs solutions - few tracking datasets, supervised approaches might overfit)
- Future work
  - use negative classifiers to vote where the object is not (combine them with the positive vote)
  - use stronger (CNN) features
  - add rotation invariance
  - VOT contest

Thank you!

References I

[1] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr. Staple: Complementary learners for real-time tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1401–1409, 2016.

[2] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4310–4318, 2015.

[3] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *European Conference on Computer Vision*, pages 472–488. Springer, 2016.

[4] D. A. Forsyth. Object detection with discriminatively trained part-based models. *IEEE Computer*, 47:6–7, 2010.

References II

[5] M. Godec, P. M. Roth, and H. Bischof. Hough-based tracking of non-rigid objects. *Computer Vision and Image Understanding*, 117:1245–1256, 2011.

[6] E. Gundogdu and A. A. Alatan. Spatial windowing for correlation filter based visual tracking. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 1684–1688. IEEE, 2016.

[7] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M. Cheng, S. L. Hicks, and P. H. S. Torr. Struck: Structured output tracking with kernels. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(10):2096–2109, 2016.

[8] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3):583–596, 2015.

References III

[9] G. Hua and H. Jégou, editors. *Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II*, volume 9914 of *Lecture Notes in Computer Science*, 2016.

[10] C.-H. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? In *CVPR*, 2011.

[11] J. Kwon and K. M. Lee. Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 1208–1215, 2009.

References IV

[12] Y. Li, J. Zhu, and S. C. Hoi. Reliable patch trackers: Robust visual tracking by exploiting reliable patches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 353–361, 2015.

[13] K. P. Murphy. *Machine learning - a probabilistic perspective*. Adaptive computation and machine learning series. MIT Press, 2012.

[14] H. Nam, M. Baek, and B. Han. Modeling and propagating cnns in a tree structure for visual tracking. *arXiv preprint arXiv:1608.07242*, 2016.

[15] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4293–4302, 2016.

References V

[16] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based multiple-person tracking with partial occlusion handling. In *CVPR*, 2012.

[17] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3119–3127, 2015.

[18] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *ICCV*, 2011.

[19] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37:1834–1848, 2015.