Scalable Kernel Correlation Filter with Sparse Feature Integration

Andrés Solís Montero, Jochen Lang and Robert Laganière.

uOttawa

University of Ottawa

December 12, 2016

Outline

Motivation

- Fast object tracking with live learning
- Object representation, independent of the type of object
- Live estimation of location and scale changes
- General solution for tracking objects??

Problem

- Tracking object with a moving camera
- No information of the object except an initial selection
- Challenging scenarios and object representations, i.e., partial occlusions, noise, and small and low textured objects
- Estimating location and change of scale
- Speed performance and scalability

Objectives

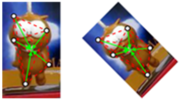
- Develop a fast and accurate tracking framework
- Estimate changes in location and scale
- Uses a general object representation
- Benchmark the solution: Visual Benchmark and VOT Challenges
  - Precision, Success, Accuracy, and Robustness

Contributions

- Extended the KCF framework to add on-line scale estimation
- Improved object/background separation.
- Combines sparse and dense object representations to estimate location and scale on-line
- Improved real-time frame rates and low latency using fHOG (SSE2) and Intel's CCS format for Fourier spectrums
- Improved precision, success, accuracy, and robustness
- Possibility of processing high dimensional data with different feature/scale/correlation estimation methods

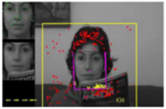
Tracking Learning and Detection (TLD)
Z.Kalal, K. Mikolajcyk and J. Matas 2012.

Consensus-based Match. Track. of Keypoints
(CMT) G. Nebehay & Roman Pflugfelder, 2014.



Structured Output Tracking with Kernels
(Struck) S. Hare, A. Saffari and P. H. Torr, 2011

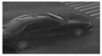
Object Tracking by Oversampling Local Features
(Alien) F. Pernici and A. del Bimbo, 2014.

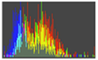


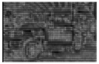
Object Representations

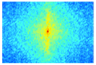
Dense Representations


Color : RGB, HSV, HLS ....


Single channel: gray , tir, ...


Histogram

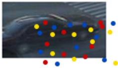

Gradient, HOG, ...

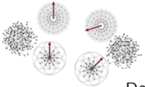

Frequency Domain

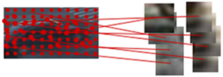
Sparse Representations

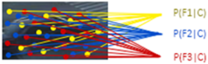

Keypoints


Grid points


Random points


Descriptors


Patches



Classifiers

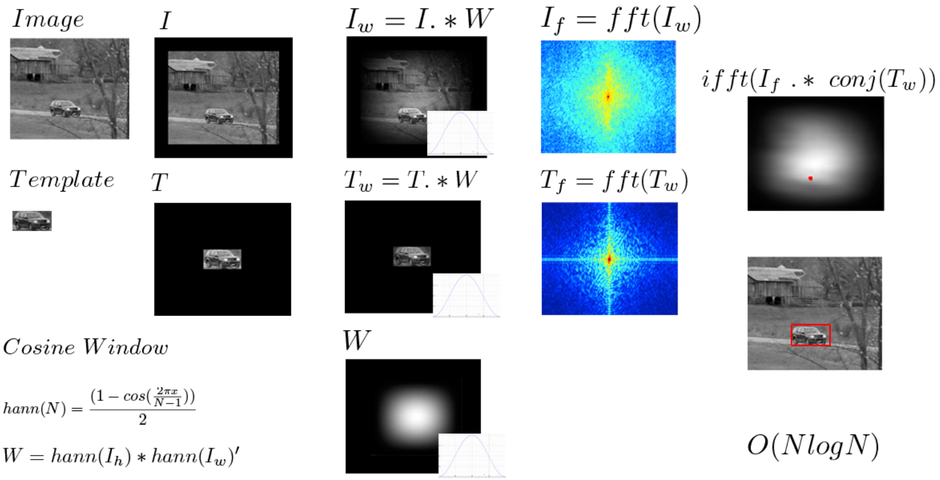
Related Work

- Visual Tracker Benchmark: 29 Trackers.
- VOT Challenges: 27 Trackers (2013), 38 Trackers (2014) ...
- Among most relevant work:
  - TLD, SCM, Struck, CMT, Alien, KCF, CSK, SAMF, etc

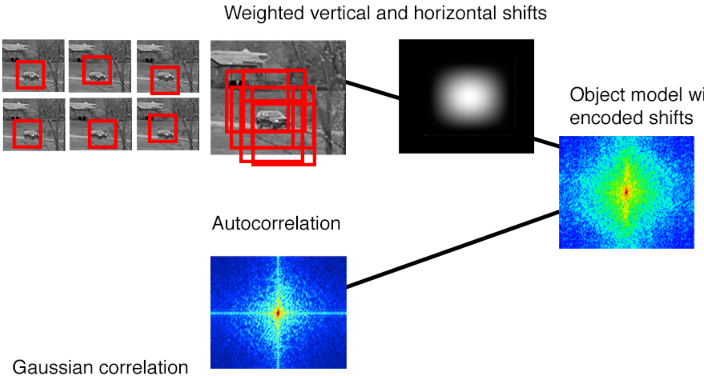
Selected Work

- Henriques, J. F. et al., High-Speed Tracking with Kernelized Correlation Filters, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015.

Algorithm Overview - KCF- Estimating Location



Algorithm Overview - KCF - Estimating Location



$$\alpha_f = \frac{y_f}{k_f^{xx'} + \lambda}$$

Learning Formula

Gaussian correlation

$$k^{xx'} = e^{\left(-\frac{1}{\sigma^2}\left(\|x\|^2 + \|x'\|^2 - 2F^{-1}\left(\sum_c x_f^c \odot (x_f'^c)^*\right)\right)\right)}$$

Algorithm Overview - KCF

Algorithm 1 : KCF.

Variables with subscript f are in the frequency domain. Circled operators represent element-wise operations (i.e., $\odot$ and $\oslash$).

- w_sz: size of the tracked region, (W×H).
- pos: center location of the tracker in spatial domain.
- $patch$: region of img centered at pos with size w_sz, (W×H×C).
- $features(x)$: extracted features (e.g., HoG), (m×n×c).
- cos_window: cosine window weights each feature channel, (m×n×1).

for each img **in** $sequence$:
  if **not** first image:
    $patch \leftarrow$ region(img, pos, w_sz)
    $z_f \leftarrow F$(features($patch$) $\odot$ cos_window)
    $k_f^{z\tilde{x}} \leftarrow F$(correlation($z_f$, $\tilde{x}_f$)) ▷ Eq. (2)
    $pos \leftarrow pos + \underset{loc}{\text{argmax}}(r(k_f^{z\tilde{x}}))$ ▷ Eq. (3)
  $patch \leftarrow$ region(img, pos, w_sz)
  $x_f \leftarrow F$(features($patch$) $\odot$ cos_window)
  $k_f^{xx} \leftarrow F$(correlation(x_f, x_f)) ▷ Eq. (2)
  $\alpha_f \leftarrow y_f \oslash (k_f^{xx} + \lambda)$ ▷ Eq. (1)
  if first image: $f \leftarrow 1$ **else** $f \leftarrow factor$
  $\tilde{\alpha}_f \leftarrow f \times \alpha_f + (1 - f) \times \tilde{\alpha}_f$
  $\tilde{x}_f \leftarrow f \times x_f + (1 - f) \times \tilde{x}_f$

Learning Formula: Eq. 1

$$\alpha_f = \frac{y_f}{k_f^{xx'} + \lambda}$$

Gaussian Correlation: Eq. 2

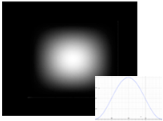
$$k^{xx'} = e^{\left(-\frac{1}{\sigma^2}\left(\|x\|^2+\|x'\|^2-2F^{-1}\left(\sum_c x_f^c \odot (x'^c_f)^*\right)\right)\right)}$$

Response: Eq. 3

$$r(k_f^{z\tilde{x}}) = F^{-1}(k_f^{z\tilde{x}} \odot \tilde{\alpha}_f)$$

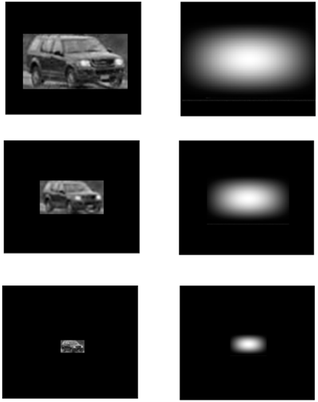
Algorithm Overview - Adjustable Windows

Cosine Window



Scale Changes

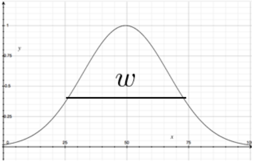
Adjustable Windows



Gaussian window

$$g(N) = e^{-\frac{1}{2}\left(\frac{i}{\sigma(N-1)}\right)^2}, \quad 0 \leq i \leq N.$$

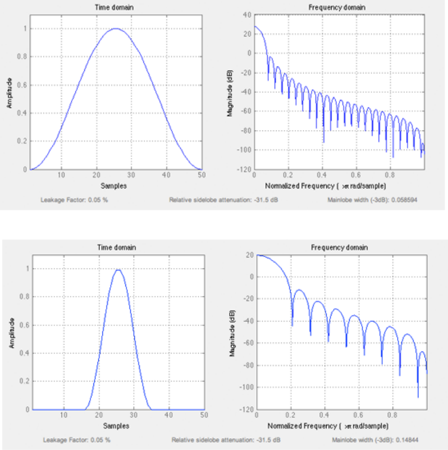
$$\sigma = \frac{w}{N}$$



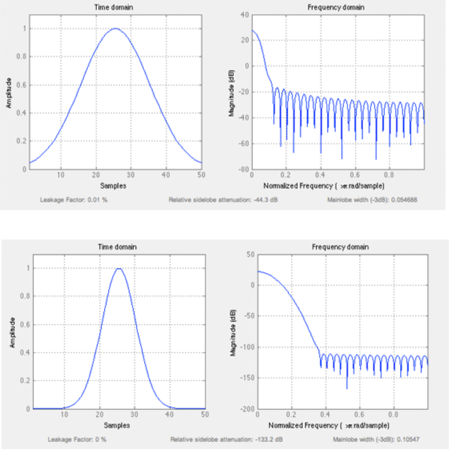
Algorithm Overview - Adjustable Windows [examples]

Cosine vs Gaussian Window

Cosine window



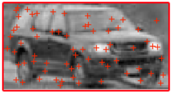
Gaussian window



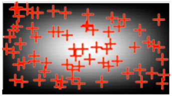
Algorithm Overview - Estimating Scale

$$\boxed{} * scale = \boxed{}$$

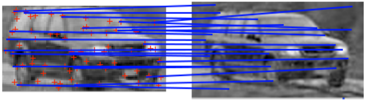
$keypoints\ k_i$



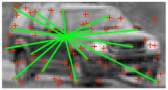
$weights\ w_i$



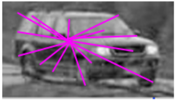
$optical\ flow$



$\|k_1 - k_j\|_2$



$\|m_1 - m_j\|_2$



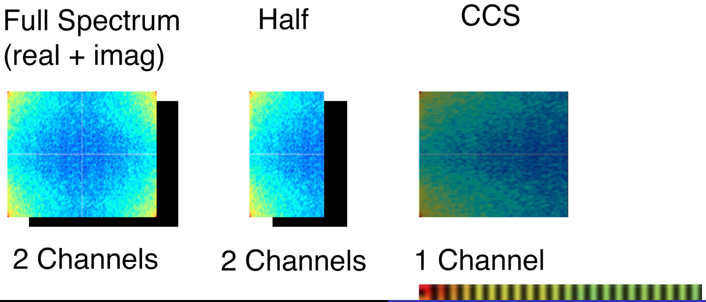
$Weighted\ arithmetic\ mean$

$$scale = \frac{\sum_i \sum_j w_i w_j * \frac{\|m_i - m_j\|_2}{\|k_i - k_j\|_2}}{\sum_i \sum_j w_i w_j}$$

$O(KN)$

Improving Performance

- fast HOG descriptors (SSE instructions)
  Felzenszwalb et al. Object detection with discriminatively trained part, TPAMI 2010.
- Intel's CCS packed format
- Optimal search area $N = 2^p \times 3^q \times 5^r$
  (e.g., 300x300 $= 5^2 \times 3 \times 2^2$, closer power of two is 512x512).



Algorithm sKCF

Algorithm 2 : sKCF.

Changes to the KCF pipeline are showed in different color.

- w_sz: size of the tracked region, (W×H).
- t_sz: size of the target, (w×h).
- $features(x)$: extracted features (e.g., HoG), (m×n×c).

for each img **in** $sequence$:
  if not first image:
    $p2 \leftarrow$ region(img, pos, w_sz)
    $z_f \leftarrow F$(features($p2$) $\odot$ cos_window)
    $k_f^{z\tilde{x}} \leftarrow F$(correlation($z_f$, $\tilde{x}_f$)) ▷ Eq. (2)
    $pos \leftarrow pos + \underset{loc}{\mathrm{argmax}}(r(k_f^{z\tilde{x}}))$ ▷ Eq. (3)
    $t\_sz \leftarrow t\_sz$ * scale(K^{p1}, K^{p2}) ▷ Eq. (7)
  $p1 \leftarrow$ region(img, pos, w_sz)
  $gauss \leftarrow$ G(m, n, t_sz, w_sz) ▷ Eq. (4, 5)
  $x_f \leftarrow F$(features($p1$) $\odot$ $gauss$)
  $k_f^{xx} \leftarrow F$(correlation(x_f, x_f)) ▷ Eq. (2)
  $\alpha_f \leftarrow y_f \oslash (k_f^{xx} + \lambda)$ ▷ Eq. (1)
  if first image: $f \leftarrow 1$ **else** $f \leftarrow factor$
  $\tilde{\alpha}_f \leftarrow f \times \alpha_f + (1 - f) \times \tilde{\alpha}_f$
  $\tilde{x}_f \leftarrow f \times x_f + (1 - f) \times \tilde{x}_f$

Learning Formula: Eq. 1

$$\alpha_f = \frac{y_f}{k_f^{xx'} + \lambda}$$

Gaussian Correlation: Eq. 2

$$k^{xx'} = e^{\left(-\frac{1}{\sigma^2}\left(\|x\|^2 + \left\|x'\right\|^2 - 2F^{-1}\left(\sum_c x_f^c \odot ({x'}_f^c)^*\right)\right)\right)}$$

Response: Eq. 3

$$r(k_f^{z\tilde{x}}) = F^{-1}(k_f^{z\tilde{x}} \odot \tilde{\alpha}_f)$$

Gaussian Window : Eq. 4

$$g(N, \sigma) = \exp -\frac{1}{2}\left(\frac{i}{\sigma(N-1)}\right)^2, \quad 0 \leq i \leq N.$$

Scale Estimation: Eq. 7

$$scale(K^{p1}, K^{p2}) = \frac{\sum_i^T \sum_j^T w_i w_j * \frac{\left\|K_i^{p2} - K_j^{p2}\right\|^2}{\left\|K_i^{p1} - K_j^{p1}\right\|^2}}{\sum_i^T \sum_j^T w_i w_j}.$$

Datasets

Tracker Benchmark v1.0 [Yi Wu et al. 2013]

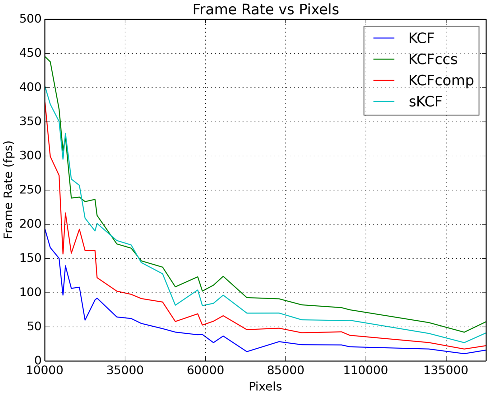
- 50 sequences with 29 trackers
- Measures: precision and success

VOT Challenge [Kristan et al.]

- VOT2013: 16 sequences with 27 trackers
- VOT2014: 25 sequences with 37 trackers
- VOT2015: 60 sequences
- VOTTIR2015: 20 sequences
- Measures: accuracy and robustness/reliability

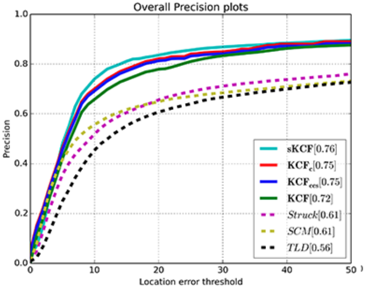
Speed

Frame rate expressed in frames per second (y-axis of the plot) measured by the number of pixels processed (x-axis of the plot).

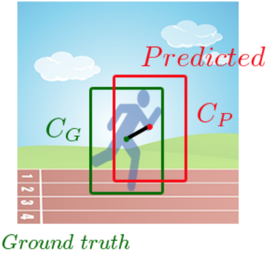


Precision [Yi Wu et al.]

Precision plot shows the ratio of successful frames whose tracker output is within the given threshold (x-axis of the plot, in pixels) from the ground-truth, measured by the center distance between bounding boxes.

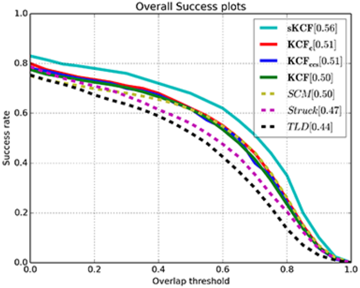


$$||C_p - C_c|| < T_p$$

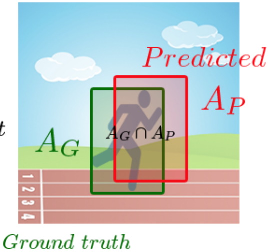


Success [Yi Wu et al.]

For an overlap threshold (x-axis of the plot), the success ratio is the ratio of the frames whose tracked box has more overlap with the ground-truth box than the threshold.

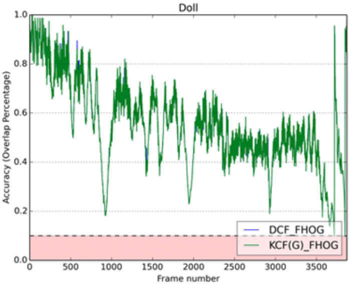


$$\frac{A_G \cap A_P}{A_G \cup A_P} \geq S_t$$

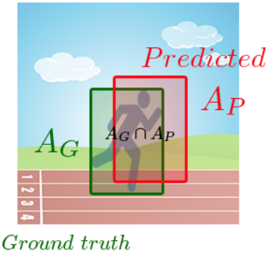


Accuracy [Kristan et al.]

Overlap between the ground-truth AG and the area predicted by a tracker, i.e., AP. The overall accuracy of a sequence is the average accuracy of all the frames in the sequence.

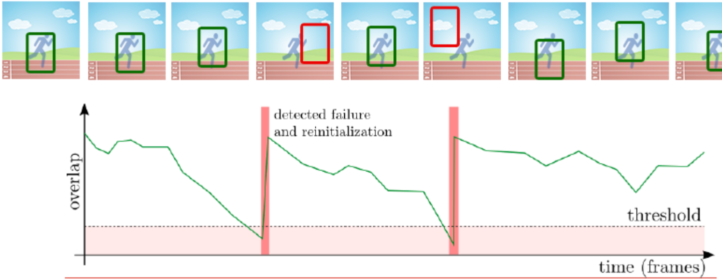


$$\frac{A_G \cap A_P}{A_G \cup A_P}$$



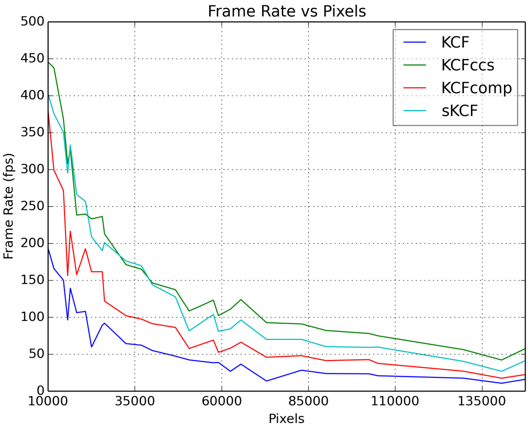
Robustness/Reliability

Counts the number of times the tracker failed and had to be reinitialized. Failure occurs when the overlap drops below a threshold.



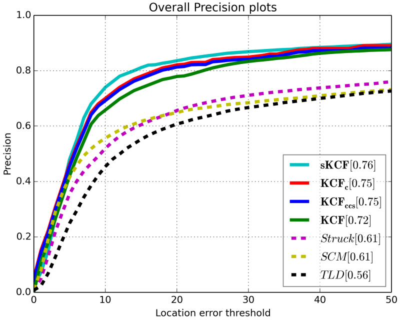
Speed Benchmark

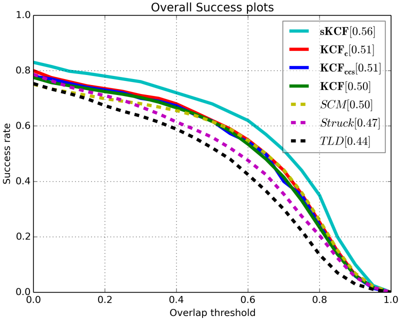
Comparison between KCF implementation [Henriques et al. 2015] and our solution



Precision and Success

Dataset: Visual Tracker Benchmark [Yi Wu et al. 2013]





VOT 2014

Table : VOT 2014 Results

	Overall		Rank			
	Acc.	Fail.	Acc.	Rob.	Overall	fps
DSST	**0.65**	**16.90**	5.44	**12.17**	**8.81**	5.8
SAMF	**0.65**	19.23	**5.23**	12.94	9.09	1.6
sKCF	0.61	18.44	7.68	13.14	10.41	**65.4**
KCF	0.56	27.14	13.14	18.02	15.58	20.3

VOT 2015

Table : VOT and VOT TIR 2015 Results

VOT 2015	Overall		Rank			
	Acc.	Fail.	Acc.	Rob.	Overall	fps
sKCF	**0.50**	**2.49**	**2.22**	**2.60**	**2.41**	**64.5**
KCF	0.47	2.61	3.29	2.68	2.99	24.4
VOT TIR 2015						
sKCF	0.58	**5.28**	2.92	**2.50**	**2.71**	**215.0**
KCF	0.56	5.66	3.40	2.65	3.02	94.8

Conclusions

- Scalable KCF solution that reacts better to object transformations and changes of scale
- Gaussian Window filtering for better object/background separation.
- Combines sparse and dense object representations to estimate location and scale on-line
- Improved real-time frame rates and low latency using fHOG (SSE2) and CCS format for Fourier spectrums
- Improved precision, success, accuracy, and robustness
- Possibility of processing high dimensional data with different scale estimation methods

Future Work

- Including rotation
- Improve learning methodology, tracker should drop information while occluded
- Improve speed performance
- Compare adjustable filtering functions