ROBUST OBJECT TRACKING AND ADAPTIVE DETECTION FOR AUTO NAVIGATION OF UNMANNED AERIAL VEHICLE

Master’s Thesis Defense

Soumyaroop Nandi

Electrical Engineering and Computer Science Department

University of Kansas

Advisor: Dr. Richard Wang

Committee Members: Dr. James Rowland, Dr. James Stiles

9th Jan, 2017
OVERVIEW

- What is Object Detection?
- What is Object Tracking?
OUTLINE

- Background
- Motivation and Goals
- Contributions
- Literature Review
  - Object Detection
  - Object Tracking
- System Overview
  - MBD Detection
  - Post Processing
  - KCF Tracking
  - Adaptive Redetection
  - Algorithm 1: Proposed Method
- Experiments and Results
  - Quantitative Evaluation
  - Qualitative Evaluation
- Limitations
- Conclusions and Future Work
BACKGROUND

BACKGROUND

Major Tasks:

1. Image Registration

2. Object Detection

3. Tracking the Detected Object
BACKGROUND

Challenges:

Background Clutter
Presence of Noise
Occlusion
Fast Camera Motion

Varying Object velocity
Rapid Background Change
Appearance Variation
Illumination Variation
MOTIVATION AND GOALS

• Till date no smart visual system is developed, helping automated UAV navigation flawlessly.
• Real-time (at least 5 frames per second), autonomous, sense and avoidance navigation system.
• Transponder based collision avoidance system fails in presence of non-cooperative rigid objects or other mobile objects.
• Vision-based methods:
  ✓ Robust to electromagnetic interference
  ✓ Compact and low power consumption
• Tracking with forward looking camera.
CONTRIBUTIONS

✓ Scale adaptive bounding box around the object in real time for all the frames.
✓ Runs automatically from the first frame till the last frame.
✓ Closed loop system with long-term, error free tracking.
✓ No computationally expensive supervised training.
✓ Adaptive redetection technique proposed
✓ Highest speed and best performance when compared to 6 other state-of-the-art trackers in 14 challenging datasets.
LITERATURE REVIEW

Salient Object Detection:

- **Top-Down Methods** [5], [6], [19]
  - Supervised training performed in previously detected frames.
  - Prior knowledge about object’s approximate location helps reduced search area.

- **Bottom-Up Methods** [7], [8], [20], [21], [22]
  - Compare low level image features (like color, text, shape, contrast, gradient, Spatio-temporal) between background and salient regions.
  - Fails in complex images.

- **Proposal Window generating Methods** [25], [26]
  - Do not provide 1 bounding box, but multiple ranked proposals.

- **Deep Learning/ Neural Network based Methods** [20], [22], [36]
  - Generate more accurate result, but require huge training datasets.
LITERATURE REVIEW

Object Tracking:

- Generative Approaches [5], [6], [11], [12]
  - Takes into account only the positive image patches from the previously tracked frames to train the classifier.

- Discriminative Approaches [3], [13], [14]
  - Takes into account both positive and negative samples from the previously tracked frames to train the classifier.

- Ng et al. [15] has mathematically shown that discriminative asymptotic error is lower than generative asymptotic error.

<table>
<thead>
<tr>
<th>TRACKING APPROACH</th>
<th>FEATURES/LIMITATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection and Tracking [16]</td>
<td>Not real-time, manual initialization, supervised training on huge dataset required.</td>
</tr>
<tr>
<td>Saliency-based Tracking [17]</td>
<td>Not real-time, automatic initialization enabled, supervised training required.</td>
</tr>
<tr>
<td>Multiple Camera Tracking [18]</td>
<td>Real-time, camera calibration required calibration, inefficient without GPS data.</td>
</tr>
</tbody>
</table>
# LITERATURE REVIEW

## Correlation Filter based Object Tracking:

<table>
<thead>
<tr>
<th>TRACKER</th>
<th>FEATURES/LIMITATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOSSE filter[28]</td>
<td>First real-time correlation filter based tracker trained on grayscale images in Fourier Domain.</td>
</tr>
<tr>
<td>CSK [29]</td>
<td>Kernel based correlation filter was introduced to MOSSE filter [28].</td>
</tr>
<tr>
<td>CN [32]</td>
<td>Used color features in CSK filter [29]. The speed was reduced by many folds.</td>
</tr>
<tr>
<td>KCF [3]</td>
<td>Introduced HOG features (31 bins added linearly in Fourier domain) and Gaussian kernels for correlation filtering. Biggest limitation is that it is scale invariant. Fails in long-term tracking.</td>
</tr>
<tr>
<td>Deep Learning/Neural Network based Trackers [37], [38]</td>
<td>use deep learning for better accuracy. But cannot be used in real-time application yet.</td>
</tr>
</tbody>
</table>
Fig 5.1: Illustration of our approach: (a) input frames, (b) auto initialization and KRLS training, (c) redetection,(from left to right) saliency map generation, binary image thresholding with post processing and bounding box drawn around redetected object.
MBD DETECTION

- Segmentation done by image properties like appearance contrast, rarity, region uniformity, spatial compactness, uniqueness, color cue, spectral cue, edge cue, super-pixel cue of pixels.

- Mathematically, these characteristics can be represented as distance functions [44].

- Let $P$ be the set of all integers $(i,j)$. The function $f$ from $P \times P$ into some non-negative integers is termed as:
  
  a) *Positive definite*, if $f(x,y) = 0$, if and only if $x = y$.
  
  b) *Symmetric*, if $f(x,y) = f(y,x)$, for all $x,y$ in $P$.
  
  c) *Triangular*, if $f(x,z) \leq f(x,y) + f(y,z)$, for all $x, y, z$ in $P$.

If $f$ satisfies all the three conditions (a-c), then $f$ can be termed as a *distance function* or a *pseudo-metric*.

- In an image, *distance function* computed as Euclidean distance or some variant of Dijkstra’s algorithm.
**MBD DETECTION**

**Image Processing and Graph Theory Equivalence**

- Consider a graph \( G = \{V,E\} \), \( G \) is a connected graph consisting of finite set of vertices \( V \) and finite set of edges \( E \).

- In image processing, \( V \) represent image intensity, color, contrast or other image characteristics.

- A path (\( \pi \)), in a connected graph \( G = \{V,E\} \), can be represented with a set of edges \( E \), \( \pi = \langle \pi(0), \pi(1), \ldots, \pi(n) \rangle \), for all \( \{\pi(i), \pi(i+1)\} \in E \) and \( i \in \{1,2,3,\ldots,n\} \).

- Multiple paths between two same image pixels.

- Cost function or weight \( \lambda(\pi) \geq 0 \) is assigned to each path \( \pi \).

- Minimum distance function \( f_\lambda = \min \{\pi(i)\}, i \in \{1,2,3,\ldots,n\} \).

- For every path \( \pi = \langle \pi(0), \pi(1), \ldots, \pi(n) \rangle \),
  
  a) \( \lambda(\pi) = \lambda(\langle \pi(n), \pi(n-1), \ldots, \pi(0) \rangle) \)
  
  b) \( \lambda(\pi) \leq \lambda(\langle \pi(0), \ldots, \pi(i) \rangle) + \lambda(\langle \pi(i), \ldots, \pi(n) \rangle) \) for all \( 0 \leq i \leq n \)
Cost function $\lambda(\pi)$ can either produce edge weight map or vertex weight map.

Geodesic Distance function [7] generate edge weight map with small-weight-accumulation problem.

MBD function $f_\lambda$ [8, 9, 10] generate vertex weight map.

$$f_\lambda = \min_{\pi(i) \in I_{a,b}} (\lambda^+ (\pi) - \lambda^- (\pi))$$

$$= \min_{\pi(i) \in I_{a,b}} \left( \max_{i=0,1,...,n} \{\lambda(\pi(i))\} - \min_{i=0,1,...,n} \{\lambda(\pi(i))\} \right)$$

$I_{a,b}$ represents all the possible paths from pixels $a$ to $b$.

Cost function $\lambda(\pi)$ remains constant (until higher intensity than previous seed is encountered) unlike other distance functions.

MBD cannot be computed using Dijkstra’s algorithm.

- Exact MBD [9]
- Approximate MBD [10]
- Raster scanned MBD [8].

Order of complexity is $O(mn \log n)$. $m$ is the number of distinct intensities in all the pixels and $n$ is the total number of pixels in the image.
**MBD DETECTION**

**Raster Scanned MBD:**

- The path of the raster and inverse raster scans consist of two parts –
  - Path from background seed to the adjacent pixel - $\pi_y$ and Path from adjacent pixel to the chosen pixel – $\pi_{\{y,x\}}$
- Let the entire path cost function be represented as $\lambda(\pi) = \lambda(\pi_y) + \lambda(\pi_{\{y,x\}})$, then the distance function for a single scan is given by:

  $$f_\lambda = \max_{i=0,1,...,n} \{\lambda(\pi(i))\} - \min_{i=0,1,...,n} \{\lambda(\pi(i))\}$$

  $$= \max \{A(y),I(x)\} - \min \{B(y),I(x)\}$$  \hspace{1cm} (3.4)

- $A(y)$ and $B(y)$ are the highest and lowest pixel values on the entire path $\pi$ from background seed to chosen pixel in a single scan.
**MBD DETECTION**

**Raster Scanned MBD:**

- Let $f_I$ be the final MBD map. $f_I$ is modified in each scan by choosing the minimum between previous $f_A$ and current $f_I$ as:

  $$f_I = \min (\text{current } f_I, \text{previous } f_A)$$

- $A(y)$ and $B(y)$ are modified after each raster and inverse raster scan, as the path changes every time and this in turn modifies $f_A$ and $f_I$.
- Each iteration stops when $f_A$ becomes equal to $f_I$.

---

Fig 3.4: Original Image pixel intensity distribution  
Fig 3.5: Initial $f_I$, the MBD map  
Fig 3.6: Initial Highest pixel intensity of the path $A(y)$  
Fig 3.7: Initial Lowest pixel intensity of the path $B(y)$
Otsu’s Method:

- Let $H$ be the *histogram* of saliency map $s_m$. Intensity levels $L \in [0, l-1]$, $n_i$ be the number of pixels with intensity $i \in L$

  $$H = \sum_{i=0}^{l-1} n_i$$

- Let $H_n$ be the *normalized histogram* of $s_m$, such that for every threshold value $t$, $t \in L$, we divide the normalized histogram into two groups – $C_1 \in H_n^i$, $i \in [0,t]$ and $C_2 \in H_n^i$, $i \in [t+1,l-1]$.

  $$P_1 = P(C_1) = \sum_{i=0}^{t} H_n^i$$
  $$P_2 = P(C_2) = \sum_{i=t+1}^{l-1} H_n^i = 1 - P_1$$

  $$m_1 = \sum_{i=0}^{t} i \cdot P(i/C_1) = \sum_{i=0}^{t} \frac{P(C_1/i) \cdot P(i)}{P(C_1)} = \frac{1}{P_1} \sum_{i=0}^{t} i \cdot H_n^i$$
  $$m_2 = \frac{1}{P_1} \sum_{i=t+1}^{l-1} i \cdot H_n^i$$

- $m_1$ and $m_2$ are the mean intensities of all the pixels in *group1* ($C_1$) and *group2* ($C_2$) respectively.
Otsu’s Method:

- Let $m_g$ be the *global intensity mean* and $m_t$ be the mean intensity up to $t$.

\[
\sigma_b^2 = P_1 \cdot (m_1 - m_g)^2 + P_2 \cdot (m_2 - m_g)^2 = \frac{m_g \cdot P_1 - m_t}{P_1 \cdot (1 - P_1)}
\]

- $\sigma_b^2$ is the *inter-class variance* and the optimal threshold $t_{opt}$ for the saliency map $s_m$ is given by:

\[
\sigma_b^2(t_{opt}) = \max_{0 < t < l-1} \sigma_b^2(t)
\]

- Result obtained by thresholding the saliency map $s_m$ with $\sigma_b^2(t_{opt})$ produces a *binary image*.

Fig 3.8 (a) Saliency Map generated by Fast MBD detector with *airplane_016* dataset [30] (b) Bounding box around original image frame from *airplane_016* dataset [30] after detection
KCF TRACKING

Dense Sampling:
- Discriminative methods use positive samples from foreground and negative samples from background. Background is generally huge, with large negative samples. Most discriminative trackers randomly choose less number of negative samples to reduce complexity.
- KCF works with correlation filter in Fourier domain. Huge number of positive and negative samples can be used for training the classifier without increasing complexity.
- The classifier is trained with a Regularized Least Squares (RLS)
- Let $X$ and $Y$ be 2 sets of random variables such that there are $n$ sets of statistically independent and identically distributed training samples $T = \{(x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n)\}$.
- We define a binary classifier $B(x)$ to distinguish positive and negative labels from the training sample set $T$. $B(x)$ is trained with RLS as:
  \[
  \text{RLS} = \min_B \sum_{i=1}^n G(y_i, B(x_i)) + \lambda ||B||^2 \tag{4.1}
  \]
- $G(y_i,B(x_i))$ is a loss function, $B$ is the regularization trade off and $\lambda$ is a regularization parameter used to control the regularization and makes sure that there is no overfitting.
- The loss function chosen in RLS is $G(y,B(x)) = (y - B(x))^2$.
- Eigen decomposition is performed to express $B$ as:
  \[
  B = (X^TX + \lambda I)^{-1} X^T y \tag{4.2}
  \]
- $I$ is an identity matrix and $X^T$ is the transpose of matrix $X$. In the complex plane, $X^T$ becomes $X^H$.
  \[
  B = (X^H X + \lambda I)^{-1} X^H y \tag{4.3}
  \]
KCF TRACKING

Circulant Matrix:
- Base samples are represented in a Circulant matrix. Let \( x = (x_1, x_2, \ldots, x_n) \) be a vector consisting of \( n \) input samples. Circulant matrix \( M(x) \) obtained from \( x \) is given by:

\[
M(x) = \begin{bmatrix}
  x_1 & x_2 & x_3 & \cdots & x_n \\
  x_n & x_1 & x_2 & \cdots & x_{n-1} \\
  x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  x_2 & x_3 & x_4 & \cdots & x_1 \\
\end{bmatrix}
\]

\[
\Delta = \begin{bmatrix}
  0 & 0 & 0 & \cdots & 1 \\
  1 & 0 & 0 & \cdots & 0 \\
  0 & 1 & 0 & \cdots & 0 \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & 0 & \cdots & 1 \\
\end{bmatrix}
\]

- \( \Delta \) is a permutation matrix. \( M(x) \) can be obtained from \( \Delta x^T \)
- Dense samples are created by cyclically shifting base samples.
- We train the classifier with input sample vector \( x \) and test label vector \( y \).

\[
M(x).y = F^{-1} \{ F^*(x) \odot F(y) \}
\]

(4.6)

- \( M(x).y \) actually computes the convolution between vectors \( x \) and \( y \).
- Sums, products and inverses are also Circulant – so we do not have to store all of them in the memory.
- The matrix inverse operation is calculated efficiently in the Fourier domain using Eigen decomposition.

\[
B = \text{diagonal}((x^* \odot y) / (x^* \odot x + \lambda))
\]

(4.11)

- \( B \) is the binary classifier in frequency domain, \( x \) is the training vector in frequency domain, \( y \) is the testing vector sample in the frequency domain and \( \lambda \) is the regularization parameter defined in equation (4.3).
- Complexity of taking DFT is linear \( O(n \log n) \). Divisions and multiplications are element-wise with a complexity of \( O(n) \).
- Standard method of computing RLS has a complexity of \( O(n^3) \).
KCF TRACKING

Non-linear Regression:

- Computes regression function (expressed as kernels) with multiple outputs instead of binary classification.
- Non-linear classifiers using power spectrum are difficult to kernelize. Kernel trick [40] solves the problem.
- Maps input vector $x$ to higher dimensional non-linear feature space $H(x)$.
- Kernel function $k$ measures similarity between two input vectors $x_1$ and $x_2$ with a mapping in real-plane.
  $$k: X \times X \to \mathbb{R}, \quad (x_1, x_2) \to k(x_1, x_2)$$
- Simplest is inner product kernel function $k(x)$:
  $$\langle x_1, x_2 \rangle = \sum_{i,j=1}^{n} [x_i] [x_j], \quad (4.13)$$
- A distance function $f(y)$ segregates labels $y$ in the feature space.
- Distance between $f(y)$ and $y$ determines the classification. This distance can be equal for multiple kernels. So weight $\alpha_i$ is assigned representing importance of a particular kernel.
- In constrained optimization problem, this weight $\alpha_i$ is made equivalent to a Lagrange’s multiplier $\alpha_i > 0$ and the distance function can be derived from a Lagrangian.
- The distance function $f(y)$, when represented in terms of inner product kernel function $k$ in the higher dimension feature space $H(x)$ with respect to RLS binary classifier $B$ looks like:
  $$f(y) = B^T y = \sum_{i=1}^{n} \alpha \ k(y, x_i) \quad (4.17)$$
- This distance function $f(y)$ is the new kernelized classifier. A kernel matrix $K$ (nxn) stores all the inner products between all the sample pairs.
  $$K_{ij} = k(x_i, x_j) \quad (4.18)$$
Non-linear Regression: KRLS Training

- We can use the kernel matrix $K$, testing sample $y$ and the regularization parameter $\lambda$ to evaluate the dual space variable or the kernel weight $\alpha$ from equations (4.3), (4.17) and (4.18) as:
  \[ \alpha = (K + \lambda I)^{-1} y \] (4.19)

- Henrique et al. [3] states that a kernel matrix is Circulant if the kernel function satisfies:
  \[ k(x_1, x_2) = k(\Delta x_1, \Delta x_2) \] (4.20)

- $\Delta$ is the Permutation matrix.

- The kernel matrix in (4.19) is evaluated in the Fourier domain using a correlation operation and is known as the kernel correlation. The kernel correlation between two vectors $x_1$ and $x_2$ is represented as:
  \[ k_{x_1 x_2} = k(x_2, \Delta^{i-1} x_1) \] (4.21)

- Kernel autocorrelation vector of the input sample vector $x$ can be evaluated and stored in a kernel vector like (4.21) as:
  \[ k_{xx}^i = k(x, \Delta^{i-1} x) \] (4.23)

- The kernel weight vector $\alpha$ can be evaluated from input sample kernel autocorrelation by simplifying equation (4.19) and taking an inverse Fourier transform as:
  \[ \alpha = F^{-1} \left( y \left( k_{xx}^i + \lambda \right)^{-1} \right) \] (4.24)

- Equation (4.24) represents KRLS weight vector used to train the non-linear classifier.

- Conventional kernel computation has complexity $O(n^4)$. Equation (4.24) has a complexity of $O(n^2 \log n)$. 

- The kernel matrix in (4.19) is evaluated in the Fourier domain using a correlation operation and is known as the kernel correlation. The kernel correlation between two vectors $x_1$ and $x_2$ is represented as:
  \[ k_{x_1 x_2} = k(x_2, \Delta^{i-1} x_1) \] (4.21)

- Kernel autocorrelation vector of the input sample vector $x$ can be evaluated and stored in a kernel vector like (4.21) as:
  \[ k_{xx}^{i-1} = k(x, \Delta^{i-1} x) \] (4.23)
Non-linear Regression: KRLS Detection

- After training the classifier, we detect object pixels from the testing sample.
- The kernelized classifier \( f(y) \) defined in equation (4.17) classifies all the elements in testing vector \( y \) to determine them as foreground or background.
- Let \( B^y \) be the new classifier Kernel matrix for non-linear regression, similar to kernel matrix \( K \) defined in equation (4.18).
- The cyclic shifts between base input training sample vector \( x \) and the base output testing sample vector \( y \) generates all the possible training and testing sample vectors - used to develop \( B^y \).
- Let \( k_{xy}^{ij} \) is the kernel correlation between the base input training sample vector \( x \) and the base output testing sample vector \( y \). \( \Delta \) is the permutation matrix.
  \[
  k_{ij}^{xy} = k(\Delta^{i-1}y, \Delta^{i-1}x)
  \]
  \[
  (4.25)
  \]
- Let \( k^{xy} \) be the kernel correlation vector comprising of all the elements \( k_{ij}^{xy} \).
- The Circulant matrix \( M(k^{xy}) \) of \( k^{xy} \) is used to evaluate the classifier kernel matrix \( B^y \) as:
  \[
  B^y = M(k^{xy})
  \]
  \[
  (4.26)
  \]
- The kernel classifier \( f(y) \) of equation (4.17) can be termed as the KRLS classifier and defined as:
  \[
  f(y) = (B^y)^T \alpha
  \]
  \[
  (4.27)
  \]
- We can evaluate KRLS filter \( f(y) \) vector in the Fourier domain as:
  \[
  f(y) = F^{-1}(k_{xy}^{xy} \odot \alpha)
  \]
  \[
  (4.28)
  \]
KCF TRACKING

Non-linear Regression: KRLS Detection

- Radial basis kernel function following (4.20) like Gaussian kernel $k(x_1, x_2)$ is used.

$$k(x_1, x_2) = \exp\left(-\frac{1}{\sigma^2} ||x_1 - x_2||^2 \right), \quad \sigma^2 \text{ is the constant variance}$$

- Gaussian kernel correlation vector $k^{x_1x_2}$ is evaluated in the Fourier domain as:

$$k^{x_1x_2} = \exp(-\frac{1}{\sigma^2}(||x_1||^2 + ||x_2||^2 - 2F^{-1}(x_1^* \odot x_2)))$$ (4.36)

Image Pre Processing:

- Fourier transforms are periodic. But in correlation filtering with images, left edge touch right edge and top edge touch bottom edge, making them non-periodic.

- Cosine window is applied on each pixel $x_{ij}$. Boundary pixel intensity becomes zero and center pixels becomes highest magnitude pixels.

$$x_{ij} = (x_{ij}^{\text{raw}} - 0.5) \sin(\pi i/n) \sin(\pi j/n), \quad \forall (i,j) = 0,1,2,\ldots,n-1$$ (4.37)
ADAPTIVE REDETECTION

- Evaluating the saliency map in each image frame of a video sequence or evaluating the saliency map over the entire image is computationally very expensive.
- The redetection of the object in our proposed tracker is directly related to the correlation response of the KRLS filter.
- It has been seen experimentally that as the tracker starts failing to lose the object, its correlation response starts going down.
- Let $\gamma$ be the correlation filter response threshold that we will check for redetection.
- Let $S$ be the adaptive search area for redetection whenever correlation filter response goes below the threshold $\gamma$.
- Let $height$ and $width$ be the dimension of the last successful bounding box.
  - Case 1: If $0.4 \leq \gamma_1 < 0.5$, $height_1 = 1.5 \times height$, $width_1 = 1.5 \times width$ and $S_1 = height_1 \times width_1 = 2.25 \times S$
  - Case 2: If $0.3 \leq \gamma_2 < 0.4$, $height_2 = 2 \times height$, $width_2 = 2 \times width$ and $S_2 = height_2 \times width_2 = 4 \times S$
  - Case 3: If $\gamma_3 < 0.3$, $S_3 = S$ i.e. searched over entire image
- This adaptive search area $S$ is again fed as input to the fast MBD detector for redetection.
- The saliency map generation with post processing is provided as the initialization for the KRLS filter.
- Redetection is performed iteratively whenever the object is partially or absolutely outside the bounding box.
EXPERIMENTATION AND RESULTS

- The results were presented in IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI) at San Jose, 2016

**System Information:**
- Tested in Intel (R) Xeon (R) W3520 2.67 GHz CPU, 6 GB RAM
- OpenCV v.3.2.14, Visual Studio Code (C++) and MATLAB 2015b

**Datasets:**
- **Aircraft** (620 frames), **big_2** (382 frames), **airplane_001** (200 frames), **airplane_004** (200 frames), **airplane_005** (200 frames), **airplane_006** (200 frames), **airplane_007** (200 frames), **airplane_011** (300 frames), **airplane_012** (300 frames), **airplane_013** (300 frames), **airplane_015** (300 frames), **airplane_016** (300 frames) from [43], **youtube_dataset_2** (475 frames) and **youtube_dataset_3** (301 frames)

**Competing Trackers:**
- **STC** [46], **CT** [12], **CN** [32], **DSST** [33], **SAMF** [2], **KCF** [3] and **Ours** [47]
# EXPERIMENTATION AND RESULTS

**Ground Truth Annotation:**
- Manually labelled pixel coordinates of a rectangular bounding box
- $x$ and $y$ coordinates of top-leftmost corner, *height* and *width* noted
- Bounding box must cover all the object pixels and minimum background pixels
- 4,278 frames from 14 datasets were labelled

**Deciding Factors:**
- Execution Speed
- Bounding box around entire object in all frames
QUANTITATIVE EVALUATION

Comparison Metric:

- **Center Location Error (CLE)** [30] - Average Euclidean distance between the centermost pixel of the bounding box given by tracker and that by ground truth. Lower CLE better tracker.

- **Precision Rate (PR)** [30] – Percentage of frames where CLE is lower than a threshold distance (20 pixels chosen).


  \[
  \text{Overlap Rate} = \frac{a_t \cap a_g}{a_t \cup a_g},
  \]

  Tracker successful if Overlap Rate > \( \theta \), \( \theta = 0.5 \) (chosen)

  \( a_t \rightarrow \) Area of tracker’s bounding box, \( a_g \rightarrow \) Area of tracker’s bounding box, \( \theta \rightarrow \) Threshold

  \( SR \) is the percentage of frames for which Overlap Rate > \( \theta \).
QUANTITATIVE EVALUATION

Comparison Metric:

- Trackers are sensitive to initialization.

- **One Pass Evaluation (OPE)** [30] – Tracker ran from first frame to last frame to compute all the metrics (**CLE**, **PR**, **SR**).

- **Temporal Robustness Evaluation (TRE)** [30] – Compute **OPE** for a segment of video and average all the metric (**CLE**, **PR**, **SR**) from each segment.
QUANTITATIVE EVALUATION

TABLE I

QUANTITATIVE ANALYSIS OF PROPOSED AND SIX COMPETING TRACKERS ON 14 DATASETS. THE BEST AND THE SECOND BEST RESULTS ARE HIGHLIGHTED WITH BOLD-FACE AND UNDERLINE-POINT-VALUES RESPECTIVELY.

<table>
<thead>
<tr>
<th></th>
<th>OURS</th>
<th>CT</th>
<th>STC</th>
<th>CN</th>
<th>DSST</th>
<th>SAMF</th>
<th>KCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Precision Rate (OPE)</td>
<td>0.83</td>
<td>0.15</td>
<td>0.49</td>
<td>0.44</td>
<td>0.46</td>
<td>0.48</td>
<td>0.44</td>
</tr>
<tr>
<td>Average Success Rate (OPE)</td>
<td>0.62</td>
<td>0.19</td>
<td>0.41</td>
<td>0.38</td>
<td>0.41</td>
<td>0.41</td>
<td>0.39</td>
</tr>
<tr>
<td>Average Precision Rate (TRE)</td>
<td>0.78</td>
<td>0.29</td>
<td>0.54</td>
<td>0.45</td>
<td>0.54</td>
<td>0.49</td>
<td>0.48</td>
</tr>
<tr>
<td>Average Success Rate (TRE)</td>
<td>0.56</td>
<td>0.31</td>
<td>0.41</td>
<td>0.39</td>
<td>0.46</td>
<td>0.43</td>
<td>0.42</td>
</tr>
<tr>
<td>CLE (in pixels)</td>
<td>9.02</td>
<td>232.72</td>
<td>31.97</td>
<td>84.85</td>
<td>54.92</td>
<td>45.71</td>
<td>91.88</td>
</tr>
<tr>
<td>Average Speed (fps)</td>
<td>108.94</td>
<td>22.11</td>
<td>32.87</td>
<td>30.04</td>
<td>7.09</td>
<td>5.53</td>
<td>65.28</td>
</tr>
</tbody>
</table>

- Each dataset divided into 20 segments for TRE evaluation and so 85,560 frames for each tracker (Total 5,989,200 frames).
## QUANTITATIVE EVALUATION

### One Pass Evaluation (OPE): Precision Rate

<table>
<thead>
<tr>
<th>Datasets</th>
<th>OURS</th>
<th>CT</th>
<th>STC</th>
<th>CN</th>
<th>DSST</th>
<th>SAMF</th>
<th>KCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aircraft</td>
<td>0.85</td>
<td>0.18</td>
<td>0.60</td>
<td>0.16</td>
<td>0.17</td>
<td>0.60</td>
<td>0.51</td>
</tr>
<tr>
<td>airplane_001</td>
<td>0.92</td>
<td>0.02</td>
<td>0.38</td>
<td>0.20</td>
<td>0.26</td>
<td>0.43</td>
<td>0.12</td>
</tr>
<tr>
<td>airplane_004</td>
<td>0.79</td>
<td>0.25</td>
<td>0.42</td>
<td>0.44</td>
<td>0.50</td>
<td>0.21</td>
<td>0.37</td>
</tr>
<tr>
<td>airplane_005</td>
<td>0.81</td>
<td>0.09</td>
<td>0.36</td>
<td>0.33</td>
<td>0.32</td>
<td>0.39</td>
<td>0.27</td>
</tr>
<tr>
<td>airplane_006</td>
<td>0.92</td>
<td>0.22</td>
<td>0.65</td>
<td>0.54</td>
<td>0.54</td>
<td>0.75</td>
<td>0.53</td>
</tr>
<tr>
<td>airplane_007</td>
<td>0.76</td>
<td>0.08</td>
<td>0.46</td>
<td>0.61</td>
<td>0.36</td>
<td>0.27</td>
<td>0.37</td>
</tr>
<tr>
<td>airplane_011</td>
<td>0.90</td>
<td>0.27</td>
<td>0.31</td>
<td>0.43</td>
<td>0.28</td>
<td>0.31</td>
<td>0.25</td>
</tr>
<tr>
<td>airplane_012</td>
<td>0.74</td>
<td>0.02</td>
<td>0.83</td>
<td>0.15</td>
<td>0.88</td>
<td>0.99</td>
<td>0.81</td>
</tr>
<tr>
<td>airplane_013</td>
<td>0.89</td>
<td>0.02</td>
<td>0.26</td>
<td>0.32</td>
<td>0.32</td>
<td>0.19</td>
<td>0.12</td>
</tr>
<tr>
<td>airplane_015</td>
<td>0.82</td>
<td>0.35</td>
<td>0.49</td>
<td>0.73</td>
<td>0.58</td>
<td>0.57</td>
<td>0.79</td>
</tr>
<tr>
<td>airplane_016</td>
<td>0.83</td>
<td>0.18</td>
<td>0.65</td>
<td>0.76</td>
<td>0.73</td>
<td>0.52</td>
<td>0.45</td>
</tr>
<tr>
<td>big_2</td>
<td>0.89</td>
<td>0.31</td>
<td>0.85</td>
<td>0.82</td>
<td>0.91</td>
<td>0.94</td>
<td>0.85</td>
</tr>
<tr>
<td>youtube_dataset_2</td>
<td>0.81</td>
<td>0.04</td>
<td>0.66</td>
<td>0.65</td>
<td>0.56</td>
<td>0.54</td>
<td>0.46</td>
</tr>
<tr>
<td>youtube_dataset_3</td>
<td>0.69</td>
<td>0.06</td>
<td>0.12</td>
<td>0.07</td>
<td>0.07</td>
<td>0.05</td>
<td>0.26</td>
</tr>
</tbody>
</table>
# QUANTITATIVE EVALUATION

## One Pass Evaluation (OPE): Success Rate

<table>
<thead>
<tr>
<th>Datasets</th>
<th>OURS</th>
<th>CT</th>
<th>STC</th>
<th>CN</th>
<th>DSST</th>
<th>SAMF</th>
<th>KCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aircraft</td>
<td>0.52</td>
<td>0.15</td>
<td>0.34</td>
<td>0.18</td>
<td>0.13</td>
<td>0.50</td>
<td>0.42</td>
</tr>
<tr>
<td>airplane_001</td>
<td>0.78</td>
<td>0.02</td>
<td>0.32</td>
<td>0.17</td>
<td>0.27</td>
<td>0.46</td>
<td>0.12</td>
</tr>
<tr>
<td>airplane_004</td>
<td>0.54</td>
<td>0.49</td>
<td>0.47</td>
<td>0.53</td>
<td>0.42</td>
<td>0.43</td>
<td>0.44</td>
</tr>
<tr>
<td>airplane_005</td>
<td>0.57</td>
<td>0.15</td>
<td>0.39</td>
<td>0.20</td>
<td>0.26</td>
<td>0.28</td>
<td>0.23</td>
</tr>
<tr>
<td>airplane_006</td>
<td>0.54</td>
<td>0.19</td>
<td>0.49</td>
<td>0.43</td>
<td>0.45</td>
<td>0.46</td>
<td>0.43</td>
</tr>
<tr>
<td>airplane_007</td>
<td>0.53</td>
<td>0.13</td>
<td>0.47</td>
<td>0.46</td>
<td>0.55</td>
<td>0.34</td>
<td>0.49</td>
</tr>
<tr>
<td>airplane_011</td>
<td>0.77</td>
<td>0.21</td>
<td>0.33</td>
<td>0.34</td>
<td>0.29</td>
<td>0.31</td>
<td>0.20</td>
</tr>
<tr>
<td>airplane_012</td>
<td>0.47</td>
<td>0.08</td>
<td>0.48</td>
<td>0.31</td>
<td>0.59</td>
<td>1.00</td>
<td>0.70</td>
</tr>
<tr>
<td>airplane_013</td>
<td>0.70</td>
<td>0.04</td>
<td><strong>0.27</strong></td>
<td>0.19</td>
<td>0.30</td>
<td>0.11</td>
<td>0.06</td>
</tr>
<tr>
<td>airplane_015</td>
<td>0.71</td>
<td>0.45</td>
<td>0.49</td>
<td><strong>0.65</strong></td>
<td>0.59</td>
<td>0.38</td>
<td>0.54</td>
</tr>
<tr>
<td>airplane_016</td>
<td><strong>0.73</strong></td>
<td>0.22</td>
<td><strong>0.62</strong></td>
<td>0.58</td>
<td>0.56</td>
<td>0.66</td>
<td>0.58</td>
</tr>
<tr>
<td>big_2</td>
<td>0.61</td>
<td>0.30</td>
<td>0.58</td>
<td>0.58</td>
<td><strong>0.65</strong></td>
<td>0.56</td>
<td><strong>0.63</strong></td>
</tr>
<tr>
<td>youtube_dataset_2</td>
<td><strong>0.56</strong></td>
<td>0.05</td>
<td>0.25</td>
<td><strong>0.45</strong></td>
<td>0.40</td>
<td>0.27</td>
<td>0.43</td>
</tr>
<tr>
<td>youtube_dataset_3</td>
<td><strong>0.62</strong></td>
<td>0.10</td>
<td>0.16</td>
<td>0.24</td>
<td>0.22</td>
<td>0.06</td>
<td><strong>0.29</strong></td>
</tr>
</tbody>
</table>
QUANTITATIVE EVALUATION

One Pass Evaluation (OPE): Precision Plot
QUANTITATIVE EVALUATION

One Pass Evaluation (OPE): Success Plot
## QUANTITATIVE EVALUATION

**Temporal Robustness Evaluation (TRE): Precision Rate**

<table>
<thead>
<tr>
<th>Datasets</th>
<th>OURS</th>
<th>CT</th>
<th>STC</th>
<th>CN</th>
<th>DSST</th>
<th>SAMF</th>
<th>KCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aircraft</td>
<td>0.85</td>
<td>0.09</td>
<td>0.59</td>
<td>0.03</td>
<td>0.08</td>
<td>0.31</td>
<td>0.07</td>
</tr>
<tr>
<td>airplane_001</td>
<td>0.92</td>
<td>0.22</td>
<td>0.30</td>
<td>0.44</td>
<td>0.49</td>
<td>0.46</td>
<td>0.45</td>
</tr>
<tr>
<td>airplane_004</td>
<td>0.78</td>
<td>0.35</td>
<td>0.50</td>
<td>0.61</td>
<td>0.61</td>
<td>0.63</td>
<td>0.46</td>
</tr>
<tr>
<td>airplane_005</td>
<td>0.49</td>
<td>0.47</td>
<td>0.59</td>
<td>0.37</td>
<td>0.53</td>
<td>0.39</td>
<td>0.48</td>
</tr>
<tr>
<td>airplane_006</td>
<td>0.92</td>
<td>0.37</td>
<td>0.60</td>
<td>0.53</td>
<td>0.55</td>
<td>0.55</td>
<td>0.47</td>
</tr>
<tr>
<td>airplane_007</td>
<td>0.76</td>
<td>0.20</td>
<td>0.37</td>
<td>0.52</td>
<td>0.59</td>
<td>0.54</td>
<td>0.54</td>
</tr>
<tr>
<td>airplane_011</td>
<td>0.89</td>
<td>0.43</td>
<td>0.79</td>
<td>0.24</td>
<td>0.60</td>
<td>0.26</td>
<td>0.34</td>
</tr>
<tr>
<td>airplane_012</td>
<td>0.76</td>
<td>0.50</td>
<td>0.81</td>
<td>0.25</td>
<td>0.89</td>
<td>0.27</td>
<td>0.84</td>
</tr>
<tr>
<td>airplane_013</td>
<td>0.89</td>
<td>0.17</td>
<td>0.27</td>
<td>0.43</td>
<td>0.46</td>
<td>0.45</td>
<td>0.27</td>
</tr>
<tr>
<td>airplane_015</td>
<td>0.70</td>
<td>0.24</td>
<td>0.43</td>
<td>0.66</td>
<td>0.59</td>
<td>0.68</td>
<td>0.65</td>
</tr>
<tr>
<td>airplane_016</td>
<td>0.81</td>
<td>0.27</td>
<td>0.58</td>
<td>0.80</td>
<td>0.70</td>
<td>0.82</td>
<td>0.63</td>
</tr>
<tr>
<td>big_2</td>
<td>0.54</td>
<td>0.40</td>
<td>0.81</td>
<td>0.80</td>
<td>0.85</td>
<td>0.82</td>
<td>0.85</td>
</tr>
<tr>
<td>youtube_dataset_2</td>
<td>0.82</td>
<td>0.18</td>
<td>0.46</td>
<td>0.39</td>
<td>0.38</td>
<td>0.41</td>
<td>0.30</td>
</tr>
<tr>
<td>youtube_dataset_3</td>
<td>0.73</td>
<td>0.13</td>
<td>0.39</td>
<td>0.22</td>
<td>0.29</td>
<td>0.24</td>
<td>0.34</td>
</tr>
</tbody>
</table>
### QUANTITATIVE EVALUATION

**Temporal Robustness Evaluation (TRE): Success Rate**

<table>
<thead>
<tr>
<th>Datasets</th>
<th>OURS</th>
<th>CT</th>
<th>STC</th>
<th>CN</th>
<th>DSST</th>
<th>SAMF</th>
<th>KCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aircraft</td>
<td>0.52</td>
<td>0.08</td>
<td>0.34</td>
<td>0.05</td>
<td>0.10</td>
<td>0.46</td>
<td>0.06</td>
</tr>
<tr>
<td>airplane_001</td>
<td>0.78</td>
<td>0.21</td>
<td>0.19</td>
<td>0.36</td>
<td>0.45</td>
<td>0.37</td>
<td>0.40</td>
</tr>
<tr>
<td>airplane_004</td>
<td>0.54</td>
<td>0.44</td>
<td>0.46</td>
<td>0.53</td>
<td>0.56</td>
<td>0.54</td>
<td>0.46</td>
</tr>
<tr>
<td>airplane_005</td>
<td>0.34</td>
<td>0.38</td>
<td>0.49</td>
<td>0.29</td>
<td>0.45</td>
<td>0.30</td>
<td>0.41</td>
</tr>
<tr>
<td>airplane_006</td>
<td>0.52</td>
<td>0.32</td>
<td>0.44</td>
<td>0.40</td>
<td>0.41</td>
<td>0.41</td>
<td>0.36</td>
</tr>
<tr>
<td>airplane_007</td>
<td>0.53</td>
<td>0.28</td>
<td>0.44</td>
<td>0.43</td>
<td>0.57</td>
<td>0.44</td>
<td>0.56</td>
</tr>
<tr>
<td>airplane_011</td>
<td>0.77</td>
<td>0.43</td>
<td>0.53</td>
<td>0.32</td>
<td>0.53</td>
<td>0.33</td>
<td>0.41</td>
</tr>
<tr>
<td>airplane_012</td>
<td>0.35</td>
<td>0.50</td>
<td>0.56</td>
<td>0.40</td>
<td>0.78</td>
<td>0.41</td>
<td>0.78</td>
</tr>
<tr>
<td>airplane_013</td>
<td>0.71</td>
<td>0.20</td>
<td>0.30</td>
<td>0.36</td>
<td>0.44</td>
<td>0.37</td>
<td>0.29</td>
</tr>
<tr>
<td>airplane_015</td>
<td>0.59</td>
<td>0.32</td>
<td>0.56</td>
<td>0.59</td>
<td>0.54</td>
<td>0.60</td>
<td>0.48</td>
</tr>
<tr>
<td>airplane_016</td>
<td>0.66</td>
<td>0.45</td>
<td>0.49</td>
<td>0.55</td>
<td>0.51</td>
<td>0.56</td>
<td>0.52</td>
</tr>
<tr>
<td>big_2</td>
<td>0.38</td>
<td>0.39</td>
<td>0.51</td>
<td>0.60</td>
<td>0.51</td>
<td>0.61</td>
<td>0.59</td>
</tr>
<tr>
<td>youtube_dataset_2</td>
<td>0.56</td>
<td>0.19</td>
<td>0.18</td>
<td>0.30</td>
<td>0.27</td>
<td>0.31</td>
<td>0.27</td>
</tr>
<tr>
<td>youtube_dataset_3</td>
<td>0.61</td>
<td>0.17</td>
<td>0.25</td>
<td>0.27</td>
<td>0.33</td>
<td>0.28</td>
<td>0.31</td>
</tr>
</tbody>
</table>
QUANTITATIVE EVALUATION

Temporal Robustness Evaluation (TRE): Precision Plot
QUANTITATIVE EVALUATION

Temporal Robustness Evaluation (TRE): Success Plot
## QUANTITATIVE EVALUATION

### Speed Comparison:

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Tracker</th>
<th>OURS</th>
<th>CT</th>
<th>STC</th>
<th>CN</th>
<th>DSST</th>
<th>SAMF</th>
<th>KCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aircraft</td>
<td></td>
<td>105.62</td>
<td>35.73</td>
<td>63.32</td>
<td>38.58</td>
<td>9.88</td>
<td>4.20</td>
<td>49.18</td>
</tr>
<tr>
<td>airplane_001</td>
<td></td>
<td>50.41</td>
<td>17.92</td>
<td>37.30</td>
<td>89.34</td>
<td>20.41</td>
<td>3.99</td>
<td>84.61</td>
</tr>
<tr>
<td>airplane_004</td>
<td></td>
<td>99.26</td>
<td>17.93</td>
<td>28.90</td>
<td>17.81</td>
<td>4.49</td>
<td>6.20</td>
<td>64.11</td>
</tr>
<tr>
<td>airplane_005</td>
<td></td>
<td>49.26</td>
<td>18.06</td>
<td>28.62</td>
<td>17.73</td>
<td>4.25</td>
<td>5.92</td>
<td>60.58</td>
</tr>
<tr>
<td>airplane_006</td>
<td></td>
<td>58.14</td>
<td>18.01</td>
<td>31.289</td>
<td>26.14</td>
<td>6.36</td>
<td>7.75</td>
<td>90.87</td>
</tr>
<tr>
<td>airplane_007</td>
<td></td>
<td>43.73</td>
<td>17.94</td>
<td>29.65</td>
<td>21.65</td>
<td>4.60</td>
<td>7.93</td>
<td>104.29</td>
</tr>
<tr>
<td>airplane_011</td>
<td></td>
<td>140.40</td>
<td>18.54</td>
<td>34.81</td>
<td>35.21</td>
<td>8.94</td>
<td>2.77</td>
<td>37.85</td>
</tr>
<tr>
<td>airplane_012</td>
<td></td>
<td>246.45</td>
<td>18.55</td>
<td>10.03</td>
<td>3.91</td>
<td>0.99</td>
<td>1.66</td>
<td>10.62</td>
</tr>
<tr>
<td>airplane_013</td>
<td></td>
<td>39.53</td>
<td>18.11</td>
<td>39.09</td>
<td>72.71</td>
<td>16.16</td>
<td>3.55</td>
<td>60.62</td>
</tr>
<tr>
<td>airplane_015</td>
<td></td>
<td>134.72</td>
<td>17.68</td>
<td>24.40</td>
<td>14.01</td>
<td>3.29</td>
<td>5.68</td>
<td>53.43</td>
</tr>
<tr>
<td>airplane_016</td>
<td></td>
<td>131.93</td>
<td>17.80</td>
<td>12.79</td>
<td>5.33</td>
<td>1.25</td>
<td>2.45</td>
<td>16.60</td>
</tr>
<tr>
<td>big_2</td>
<td></td>
<td>160.82</td>
<td>33.84</td>
<td>37.76</td>
<td>15.02</td>
<td>3.79</td>
<td>7.57</td>
<td>68.54</td>
</tr>
<tr>
<td>youtube_dataset_2</td>
<td></td>
<td>83.23</td>
<td>18.76</td>
<td>32.80</td>
<td>25.91</td>
<td>6.12</td>
<td>9.07</td>
<td>124.93</td>
</tr>
<tr>
<td>youtube_dataset_3</td>
<td></td>
<td>72.72</td>
<td>18.58</td>
<td>16.57</td>
<td>7.16</td>
<td>1.60</td>
<td>3.19</td>
<td>22.47</td>
</tr>
<tr>
<td>mean</td>
<td></td>
<td>108.94</td>
<td>22.11</td>
<td>32.87</td>
<td>30.04</td>
<td>7.09</td>
<td>5.53</td>
<td>65.28</td>
</tr>
</tbody>
</table>
QUALITATIVE EVALUATION

Scale Variation: youtube_dataset_3 dataset
QUALITATIVE EVALUATION

Partial Occlusion: airplane_005 dataset [43]
QUALITATIVE EVALUATION

Fast Motion: airplane_001 dataset [43]
QUALITATIVE EVALUATION

Illumination Variation: airplane_006 dataset [43]
QUALITATIVE EVALUATION

In-plane Rotation: Aircraft dataset [43]
QUALITATIVE EVALUATION

Out-of-plane Rotation: big_2 dataset [43]
LIMITATIONS

1. Failure with multiple objects

2. Bad result with MBD detection failure
CONCLUSIONS AND FUTURE WORK

- We have defined a new computer vision problem of object detection and tracking to guide an UAV in its navigation.
- Extensive research was performed to track down the most relevant approaches.
- Proposed a method for tracking objects from a forward looking camera and examined thoroughly with challenging datasets.
- Six other state-of-the-art trackers were extensively tested and compared (both qualitatively and quantitatively).
- Obtained best result for most of the datasets and second best for few of them in terms of speed and accuracy.
CONCLUSIONS AND FUTURE WORK

Other Applications:

- Collision Avoidance in Airplanes
- Automatic Navigation in Water
- Fire Detection
- Automatic Traffic Surveillance
- Object Recognition and Classification
- Face Detection
REFERENCES


doi: 10.1109/TPAMI.2015.2509974


REFERENCES

REFERENCES


44. A. Rosenfeld, J. L. Pfaltz, Distance functions on digital pictures, *Pattern Recognition* 1 (1968) 33-61.


ACKNOWLEDGEMENTS

- Dr. Richard Wang, EECS Department, University of Kansas.
- Dr. James Rowland, EECS Department, University of Kansas
- Dr. James Stiles, EECS Department, University of Kansas.
- My sincere respect is shown to all the faculty and staff members of EECS Department, University of Kansas.
- I want to express my gratitude for all the students and researchers at the Computer Vision group, University of Kansas.
- I want to thank all my friends in Lawrence for their support in this journey.
Thank you