Struck: Structured Output Tracking with Kernels

Sam Hare, Amir Saffari, And Philip H. S. Torr
International Conference On Computer Vision (ICCV), 2011
Motivations

Problem: tracking-by-detection

Input: target

Output: locations over times
Performance summary

Struck TLD MIL

Figure 6. Performance summary for the trackers initialized with different size of bounding box. AVG (the last one) illustrates the average performance over all trackers for each scale.

Outline

Previous works

• Tracking-by-detection
• Adaptive tracking-by-detection

Methods

• Structured output tracking
• Online optimization and budget mechanism

Experiments and results
Previous Works

Tracking problem as a detection task applied over time

Previous Works

Tracking problem as a detection task applied over time

look for the image region with the highest SVM score

Previous Works

**Adaptive tracking-by-detection**

Previous Works – Adaptive Tracking-by-detection

![Diagram showing the process of adaptive tracking]

1. Learner
2. Sampler
3. Labeller

The diagram illustrates the flow of data from Learner to Sampler and then to Labeller.
Previous Works – Adaptive Tracking-by-detection

**Adaptive tracking-by-detection**

Tracking: A classification task
Learning: A update the object model.

Previous Works – Adaptive Tracking-by-detection

Problem 1
What is the best way to generate labelled samples?
Previous Works – Adaptive Tracking-by-detection

Problem 2

Label prediction and position estimation are different objectives.
Main Idea

structured output prediction
Main Contributions

Structured output tracking

Avoid the intermediate classification step

Online learning and budgeting mechanism

Prevents too many training data
Outline

Previous work

• Tracking-by-detection
• Adaptive tracking-by-detection

Methods

• **Structured output tracking**
• Online optimization and budget mechanism

Experiments and results
Structured Output Tracking

\[ y_t = f(x_t^{P_{t-1}}) = \arg \max_{y \in \mathcal{Y}} F(x_t^{P_{t-1}}, y) \]

Structured Output Tracking

The output space is all transformations instead of the binary labels.

\[ y_t = f(x_{t}^{p_{t-1}}) = \arg \max_{y \in \mathcal{Y}} F(x_{t}^{p_{t-1}}, y) \]

Structured SVM Model

The SVM score should correlate with overlapping size with the best tracking bounding box.

Structured Output Tracking

Algorithm 2 Struck: Structured Output Tracking

Require: \( f_t, p_{t-1}, S_{t-1} \)

1: Estimate change in object location
2: \( y_t = \arg \max_{y \in y} F(x_{t}^{p_{t-1}}, y) \)
3: \( p_t = p_{t-1} \circ y_t \)
4: Update discriminant function
5: \( (i, y_+, y_-) \leftarrow \text{PROCESSNEW}(x_{t}^{p_t}, y^0) \)
6: \( \text{SMOSTEP}(i, y_+, y_-) \)
7: \( \text{BUDGETMAINTENANCE}() \)
8: for \( j = 1 \) to \( n_R \) do
9: \( (i, y_+, y_-) \leftarrow \text{PROCESSOLD}() \)
10: \( \text{SMOSTEP}(i, y_+, y_-) \)
11: \( \text{BUDGETMAINTENANCE}() \)
12: for \( k = 1 \) to \( n_O \) do
13: \( (i, y_+, y_-) \leftarrow \text{OPTIMIZE}() \)
14: \( \text{SMOSTEP}(i, y_+, y_-) \)
15: end for
16: end for
17: return \( p_t, S_t \)
Structured Output Tracking

Algorithm 2 Struck: Structured Output Tracking

Require: $f_t, p_{t-1}, S_{t-1}$

1: Estimate change in object location
2: $y_t = \arg \max_{y \in Y} F(x_t^{p_{t-1}}, y)$
3: $p_t = p_{t-1} \circ y_t$
4: Update discriminant function
5: $(i, y_+, y_-) \leftarrow \text{PROCESSNEW}(x_t^{p_t}, y^0)$
6: $\text{SMOSTEP}(i, y_+, y_-)$
7: $\text{BUDGETMAINTENANCE}()$
8: for $j = 1$ to $n_R$ do
9:   $(i, y_+, y_-) \leftarrow \text{PROCESSOLD}()$
10:  $\text{SMOSTEP}(i, y_+, y_-)$
11:  $\text{BUDGETMAINTENANCE}()$
12:  for $k = 1$ to $n_O$ do
13:     $(i, y_+, y_-) \leftarrow \text{OPTIMIZE}()$
14:     $\text{SMOSTEP}(i, y_+, y_-)$
15:  end for
16: end for
17: return $p_t, S_t$
Structured Output Tracking

Algorithm 2 Struck: Structured Output Tracking

Require: $f_t, p_{t-1}, S_{t-1}$

1: Estimate change in object location
2: $y_t = \arg \max_{y \in Y} F(x_{p_{t-1}}^t, y)$
3: $p_t = p_{t-1} \circ y_t$

4: Update discriminant function
5: $(i, y_+, y_-) \leftarrow \text{PROCESSNEW}(x_{p_t}^t, y^0)$
6: $\text{SMOSTEP}(i, y_+, y_-)$
7: $\text{BUDGETMAINTENANCE}()$
8: for $i$ from 1 to $n$
9:    $\text{SMOSTEP}(i, y_+, y_-)$
10: end for
11: return $p_t, S_t$

Come back later
Structured output SVM

$$y_t = f(x_{t}^{P_{t-1}}) = \operatorname*{arg\,max}_{y \in Y} F(x_{t}^{P_{t-1}}, y)$$

$$\min_{w} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i$$

s.t. $\forall i: \xi_i \geq 0$

$$\forall i, \forall y \neq y_i: \langle w, \delta \Phi_i(y) \rangle \geq \Delta(y_i, y) - \xi_i$$

Efficient SMO optimization (CS229, EE364)
Kernels (CS229)

Structured output SVM

Gaussian kernel between image feature vectors (CS229)

\[ k(x, \bar{x}) = \exp(-\sigma \|x - \bar{x}\|^2), \]

Haar-like features (CS231A, CS232)

The responses of the Haar features are the input vectors of the kernel
Online optimization

Algorithm 2 Struck: Structured Output Tracking

Require: \( f_t, p_{t-1}, S_{t-1} \)
1: Estimate change in object location
2: \( y_t = \arg \max_{y \in Y} F(x_t^{p_{t-1}}, y) \)
3: \( p_t = p_{t-1} \circ y_t \)
4: Update discriminant function
5: \((i, y_+, y_-) \leftarrow \text{PROCESSNEW}(x_t^{p_t}, y^0)\)
6: \( \text{SMOSTEP}(i, y_+, y_-) \)
7: \( \text{BUDGETMAINTENANCE}() \)
8: for \( j = 1 \) to \( n_R \) do
9: \((i, y_+, y_-) \leftarrow \text{PROCESSOLD}()\)
10: \( \text{SMOSTEP}(i, y_+, y_-) \)
11: \( \text{BUDGETMAINTENANCE}() \)
12: for \( k = 1 \) to \( n_O \) do
13: \((i, y_+, y_-) \leftarrow \text{OPTIMIZE}()\)
14: \( \text{SMOSTEP}(i, y_+, y_-) \)
15: end for
16: end for
17: return \( p_t, S_t \)
Outline

Previous work

- Tracking-by-detection
- Adaptive tracking-by-detection

Methods

- Structured output tracking
- **Online optimization and budget mechanism**

Experiments and results
Online optimization

PROCESSNEW():
  • Processes a new example

PROCESSOLD():
  • Processes an existing support pattern

OPTIMIZE():
  • Processes an existing support pattern chosen at random
Budget mechanism

The number of support vectors increase over time.

Computational and storage costs grow linearly with the number of support vectors.
Incorporating a budget

A budget (limit) of the number of supporting vectors.

Remove the support vector which results in the smallest change to the weight vector.

Outline

Previous works

• Tracking-by-detection
• Adaptive tracking-by-detection

Methods

• Structured output tracking
• Online optimization and budget mechanism

Experiments and results
Experiments

- Haar-like features
  - 6 different types arranged on a grid at 2 scales on a 4 x 4 grid, resulting in 192 features

- Search radius 60, 5 radial and 16 angular divisions.

- Budget size is as low as \( B = 20, 50, 100, \text{ inf.} \)
Dataset

http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml;
Overlap criterion

Jaccard similarity of bounding boxes

\[
S^O_{p_t}(y^i_t, y^j_t) = \frac{(p_t \circ y^i_t) \cap (p_t \circ y^j_t)}{(p_t \circ y^i_t) \cup (p_t \circ y^j_t)}
\]

Metric 2: Jaccard Similarity

\[
100 \cdot \frac{\sum_{i=1}^{n} [\alpha_i = 1 \land f_i = 1]}{\sum_{i=1}^{n} [\alpha_i = 1 \lor f_i = 1]}
\]

\(\alpha_i\): estimated foreground/background label
\(f_i\): ground truth foreground/background label
Results

http://www.samhare.net/research/struck
Visualization of the support vector set

(a) girl

(b) david
Comparison

http://www.samhare.net/research/struck
Results

Struck with the smallest budget size ($B = 20$) outperforms the state-of-the-art.

Average frames per second: $12 – 21$. 
Extensions

- Used more objection representations
  - Haar-like features
  - Raw pixel features
  - Histogram features
- Combining multiple kernels seems to improve results, but not significantly.

- Use key points and associated descriptors for object detection.
- Consider other machine learning algorithms.
Main Contributions

Structured output tracking
  Avoid the intermediate classification step

Online learning and budgeting mechanism
  Prevents too many training data
References

Sam Hare, Amir Saffari Philip H. S. Torr Struck: Structured Output Tracking with Kernels International Conference on Computer Vision (ICCV), 2011


Thank you!