Robust tracking-by-detection

Ivan Bogun\textsuperscript{1}  Dr. Eraldo Ribeiro\textsuperscript{1}

\textsuperscript{1}Florida Institute of Technology

CS Seminar, November, 2014
Outline

Introduction
  Motivation
  Tracking complexity

Historical perspective
  Template-based trackers
  Tracking using Discriminative classification

Structured tracker
  Structured SVM
  SVM update

Dealing with false positives and short-time occlusions
  Kalman filter
  Robust Kalman filter

Results
Problem statement

Definition

Visual tracking is a problem where, given a location of the bounding box of the object in the first frame, the task is to locate it in consecutive ones.
Problem statement

Definition

*Visual tracking* is a problem where, given a location of the bounding box of the object in the first frame, the task is to locate it in consecutive ones.

$t=1$  
$t=25$  
$t=50$
Outline

**Introduction**
- Motivation
- Tracking complexity

**Historical perspective**
- Template-based trackers
- Tracking using Discriminative classification

**Structured tracker**
- Structured SVM
- SVM update

**Dealing with false positives and short-time occlusions**
- Kalman filter
- Robust Kalman filter

**Results**
Motivation

Why tracking?

Tracking is a fundamental problem with many applications:

▶ computer-human interface (link)
Motivation

Why tracking?

Tracking is a fundamental problem with many applications:

- computer-human interface (link)
- surveillance (link)
Motivation

Why tracking?

Tracking is a fundamental problem with many applications:
- computer-human interface (link)
- surveilance (link)
- robotics (link)
Motivation

Why tracking?

Tracking is a fundamental problem with many applications:

- computer-human interface (link)
- surveillance (link)
- robotics (link)
- video processing
Outline

Introduction
  Motivation
  Tracking complexity

Historical perspective
  Template-based trackers
  Tracking using Discriminative classification

Structured tracker
  Structured SVM
  SVM update

Dealing with false positives and short-time occlusions
  Kalman filter
  Robust Kalman filter

Results
Is tracking a hard problem? (1)

Yes, since object’s appearance can change in a number of ways:

- change of illumination
- partial or complete occlusion
Is tracking a hard problem? (1)

Yes, since object’s appearance can change in a number of ways:

- change of illumination
Is tracking a hard problem? (1)

Yes, since object’s appearance can change in a number of ways:

▶ change of illumination

▶ partial or complete occlusion
Is tracking a hard problem? (2)

- change of scale
Is tracking a hard problem? (2)

- change of scale
- rotation
Is tracking a hard problem? (2)

- change of scale

- rotation
Outline

Introduction
  Motivation
  Tracking complexity

Historical perspective
  Template-based trackers
  Tracking using Discriminative classification

Structured tracker
  Structured SVM
  SVM update

Dealing with false positives and short-time occlusions
  Kalman filter
  Robust Kalman filter

Results
Early trackers

Template matching

Given initial bounding box construct a template and match it to the best location in the next frames.

- Template matching using normalized-cross correlation
- Mean Shift tracking
KLT overview

Figure: Image taken from (S. Baker and I. Matthews, 2004)
Outline

Introduction
  Motivation
  Tracking complexity

Historical perspective
  Template-based trackers
  Tracking using Discriminative classification

Structured tracker
  Structured SVM
  SVM update

Dealing with false positives and short-time occlusions
  Kalman filter
  Robust Kalman filter

Results
Tracking using Discriminative classification

Use machine learning techniques to build object's appearance model and detect the object in consecutive frames.

Examples

- Super-pixel tracking
- Multiple-instance learning tracker
- Tracking, learning, detection
- Structured tracker
Tracking using Discriminative classification

Use machine learning techniques to build object’s appearance model and detect the object in consecutive frames.
Tracking using Discriminative classification

Use machine learning techniques to build object’s appearance model and detect the object in consecutive frames.

Examples

- Super-pixel tracking
- Multiple-instance learning tracker
- Tracking, learning, detection
- Structured tracker
Intuition

Object's appearance changes in time, so ideal tracker would need to adapt and learn it on the go.

▶ Tracker should use new information so that predictions later would be more accurate.
▶ Update should be happening fast
Intuition

Object’s appearance changes in time, so ideal tracker would need to adapt and learn it on the go.
Intuition

Object’s appearance changes in time, so ideal tracker would need to *adapt* and learn it on the go.
Object’s appearance changes in time, so ideal tracker would need to *adapt* and learn it on the go.

- Tracker should use new information so that predictions later would be more accurate.
Intuition

Object’s appearance changes in time, so ideal tracker would need to *adapt* and learn it on the go.

- Tracker should use new information so that predictions later would be more accurate.
- Update should be happening fast
Discriminative tracking

Construct positive and negative examples where

- Positive: object being tracked
- Negative: everything except the object
Discriminative tracking

Construct positive and negative examples where

- Positive: object being tracked
- Negative: everything except the object
Outline

Introduction
  - Motivation
  - Tracking complexity

Historical perspective
  - Template-based trackers
  - Tracking using Discriminative classification

Structured tracker
  - Structured SVM
  - SVM update

Dealing with false positives and short-time occlusions
  - Kalman filter
  - Robust Kalman filter

Results
Structured Support Vector Machine

Predict translation into the next frame by constructing optimal separating hyperplane which separates positive and negative examples.
Structured Support Vector Machine

Predict translation into the next frame by constructing optimal separating hyperplane which separates positive and negative examples.
Structured SVM: prediction (1)

From previously known location, sample bounding boxes for translation in the ball: $S_r = \{(x, y)|x^2 + y^2 \leq 30\}$
Structured SVM: prediction (1)

From previously known location, sample bounding boxes for translation in the ball: $S_r = \{(x, y) | x^2 + y^2 \leq 30\}$
Structured SVM: prediction (1)

From previously known location, sample bounding boxes for translation in the ball: $S_r = \{(x, y)|x^2 + y^2 \leq 30\}$

- For every pixel in the ball extract bounding box where that pixel is a center.
Structured SVM: prediction (1)

From previously known location, sample bounding boxes for translation in the ball: \( S_r = \{(x, y) | x^2 + y^2 \leq 30\} \)

- For every pixel in the ball extract bounding box where that pixel is a center.
- choose bounding box which is the furthest from the hyperplane on the positive side
Structured SVM: prediction (3)

From previously known location, sample bounding boxes for translation in the ball: \( S_r = \{(x, y) | x^2 + y^2 \leq 30\} \)

- For every pixel in the ball extract bounding box where that pixel is a center.
- Choose bounding box which is the furthest from the hyperplane on the positive side.
Outline

Introduction
   Motivation
   Tracking complexity

Historical perspective
   Template-based trackers
   Tracking using Discriminative classification

Structured tracker
   Structured SVM
   SVM update

Dealing with false positives and short-time occlusions
   Kalman filter
   Robust Kalman filter

Results
Structured SVM: update (1)

New information about object appearance should be utilized.

Positive examples
Since there is only one positive example in the frame - add it.

Negative examples
Choose negative examples which are close to the hyperplane (the ones classifier could have classified incorrectly)
New information about object appearance should be utilized.

Positive examples
Since there is only one positive example in the frame - add it.
Structured SVM: update (1)

New information about object appearance should be utilized.

Positive examples
Since there is only one positive example in the frame - add it.

Negative examples
Choose negative examples which are close to the hyperplane (the ones classifier could have classified incorrectly)
Structured SVM: update (2)

Update hyperplane to be as far as possible from the closest positive and negative example.
Structured SVM: update (2)

Update hyperplane to be as far as possible from the closest positive and negative example.
Issues

1. Introduction

Model-free tracking is one of the earliest problems in computer vision. Problem settings are trivially easy: given a bounding box in the first frame locate position of the object. In applications include surveillance, video analysis and traffic monitoring results of the detections with Robust Kalman filter. Occlusion handling is done by thresholding the reconstruction error. If the patch is deemed occluded its sparse coefficient is set to zero. In this case, the object is assumed to be occluded. We extend structured tracker to be robust towards false positives and short-time occlusions. We achieve that by comparing the results of the detector and filter in every frame. Based on the agreement between the detector and filter, we either update the tracker or look for the object in an extended region in an attempt to recover it from the occlusion or incorrect detection. We argue that thresholding cannot be made robust regardless of the form because it is impossible to distinguish between true negatives and false positives and short-time occlusions.

2. What if the object is partially or completely occluded?

What is not an object (e.g. false positive)?

3. What if the tracker made a mistake and incorrectly identified an object?

Recent benchmarks showed that state-of-the-art trackers adapt to the changing appearance of the tracking and model-free tracking. In a tracker known as Struck, a new appearance patch is added to the object model if the label from a nearest-neighbor classifier did not return a value larger than predefined threshold. The weight of each such support vector is calculated as contribution to the objective function. In addition, for the tracker to be updated, it is also important to be robust towards noise and partial or complete occlusions. In addition, the computational cost of tracking has to be low as real-time performance is often desired.
Issues

- what if the tracker made a mistake and incorrectly identified what is not an object (e.g. false positive)?
Issues

- what if the tracker made a mistake and incorrectly identified what is not an object (e.g. false positive)?
Issues

► what if the tracker made a mistake and incorrectly identified what is not an object (e.g. false positive)?

► what if the object is partially or completely occluded?
Intuition

Use more information than just appearance.
Intuition

Use more information than just appearance.

Velocity

- Velocity in-between frames can be assumed constant
- use velocity to extrapolate the location in the next frame
Intuition

Use more information than just appearance.

Velocity

▶ Velocity in-between frames can be assumed constant
▶ use velocity to extrapolate the location in the next frame

Prediction

▶ Predict best location using structured tracker
Intuition

Use more information than just appearance.

Velocity

- Velocity in-between frames can be assumed constant
- use velocity to extrapolate the location in the next frame

Prediction

- Predict best location using structured tracker
- Predict location by extrapolating velocity
Intuition

Use more information than just appearance.

Velocity

- Velocity in-between frames can be assumed constant
- Use velocity to extrapolate the location in the next frame

Prediction

- Predict best location using structured tracker
- Predict location by extrapolating velocity
- Detect a mistake if two do not coincide
Outline

Introduction
  Motivation
  Tracking complexity

Historical perspective
  Template-based trackers
  Tracking using Discriminative classification

Structured tracker
  Structured SVM
  SVM update

Dealing with false positives and short-time occlusions
  Kalman filter
  Robust Kalman filter

Results
Kalman filter

- Online linear quadratic estimator capable of estimating unobserved (latent) variables from observed measurements.

Tracking
In our case location of the bounding box from the tracker is observed variable while true location is hidden.

Good
Estimation is optimal in least squares sense

Bad
Performs bad during outliers (e.g. false positives)
Classic Kalman filter demo
Outline

Introduction
  Motivation
  Tracking complexity

Historical perspective
  Template-based trackers
  Tracking using Discriminative classification

Structured tracker
  Structured SVM
  SVM update

Dealing with false positives and short-time occlusions
  Kalman filter
  Robust Kalman filter

Results
Robust Kalman filter: trade optimality for robustness towards outliers

Method
Replace Kalman gain with robust counterpart, like Huber function:

\[ h_{b}(x) = x \min\{1, \frac{b}{\|x\|}\} \] (1)

If Kalman correction is far away - decrease it's contribution to the estimate (if tracker made a mistake - do not use that bounding box).
Solution

Robust Kalman filter: trade optimality for robustness towards outliers

Method
Replace Kalman gain with robust counterpart, like Huber function:

\[ h_b(x) = x \min \{1, \frac{b}{\|x\|} \} \]  

(1)

If Kalman correction is far away - decrease it’s contribution to the estimate (if tracker made a mistake - do not use that bounding box).
Classical vs Robust Kalman filter
Take trajectory of the bounding box from the basketball sequence and add noise to every third frame.
Classical vs Robust Kalman filter
Take trajectory of the bounding box from the basketball sequence and add noise to every third frame.

Figure: First raw corresponds to results with classical Kalman filter, while bottom with Robust counterpart
Results: visual
1. Introduction

Recent benchmarks have introduced challenging sequences for the object tracking task. Although there exist many state-of-the-art trackers, they are often sensitive to noise and occlusions. Making correct predictions despite noisy observations is an important aspect. Metal detectors are often deployed in the wild, making the approach robustness towards noise and partial or complete occlusions important. Noise appears in various forms due to changes of illumination, rotation and occlusion. A common problem is the drift of the bounding boxes if the detector makes a mistake, i.e., false positive, or when the bounding box from one video sequence in the dataset is much smaller than the other. Making correct predictions despite noisy observations is an important aspect.

To see if Robust Kalman filter is capable of filtering out noise while still making correct predictions despite noisy observations, we start by introducing the intuition behind using Robust Kalman filter. We then present a synthetic experiment. Then, we show how our implementation of the method performs in real-time application. We include results compared to the ground truth. For the frame where a ground truth trajectory was corrupted, notice how the robust Kalman filter is able to recover. This is done by using Robust Kalman filter where we decrease the weight of each such support vector is calculated as a contribution to the objective function.

Online structural SVM framework is defined as the distance between center of the tracked object, \( r_t \), and ground truth, \( r_{gt} \):

\[
pt = ||r_{gt} - r_t|| \quad (6)
\]

We assume that the data from the detector is noisy. Making correct predictions despite noisy observations is an important aspect. Metal detectors are often deployed in the wild, making the approach robustness towards noise and partial or complete occlusions important. Noise appears in various forms due to changes of illumination, rotation and occlusion. A common problem is the drift of the bounding boxes if the detector makes a mistake, i.e., false positive, or when the bounding box from one video sequence in the dataset is much smaller than the other. Making correct predictions despite noisy observations is an important aspect.

To see if Robust Kalman filter is capable of filtering out noise while still making correct predictions despite noisy observations, we start by introducing the intuition behind using Robust Kalman filter. We then present a synthetic experiment. Then, we show how our implementation of the method performs in real-time application. We include results compared to the ground truth. For the frame where a ground truth trajectory was corrupted, notice how the robust Kalman filter is able to recover. This is done by using Robust Kalman filter where we decrease the weight of each such support vector is calculated as a contribution to the objective function.
Results: quantitative

Success plot

Precision plot

Success plot

Precision plot

Figure 7. Results on dataset \([0, 1]\), SCM \([0.451]\) and orig-

Corr \([0.291]\)

STR \([0.52]\)

STR+f \([0.534]\)

TLD \([0.35]\)

Corr \([0.691]\)

SCM \([0.543]\)

STR \([0.67]\)

STR+f \([0.688]\)

TLD \([0.457]\)

Overlap threshold

Location error threshold

Overlap threshold

Location error threshold

Success rate,

Overlap threshold

Precision

Overlap threshold

Precision

Overlap threshold

Precision

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold

Overlap threshold
Tracking demo
Summary

- Discussed visual tracking: applications, historical approaches
- Introduced Structured tracker
- Improved structured tracker to be more resilient towards false positives and partial occlusions
Summary

- Discussed visual tracking: applications, historical approaches
- Introduced Structured tracker
- Improved structured tracker to be more resilient towards false positives and partial occlusions

Source code

https://github.com/ibogun/Robust-tracking-by-detection
Tested on OS X, Linux (under development)
Thank you for your attention! Questions?