Robust Visual Tracking using L1 Minimization

2009 IEEE 12th International Conference on Computer Vision (ICCV)

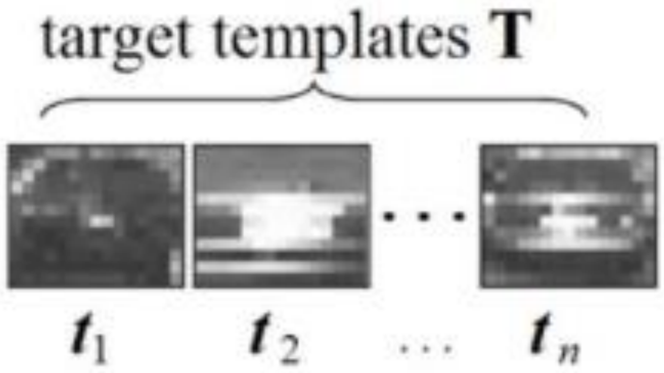
Xue Mei , Haibin Ling

Outline

- Introduction
- Related works
- L_1 Minimization Tracking
  - Sparse Representation of a Tracking Target
  - Nonnegative Constraints
  - Achieving Sparseness through L_1 Minimization
  - Template Update
- Experimental result
- Conclusion

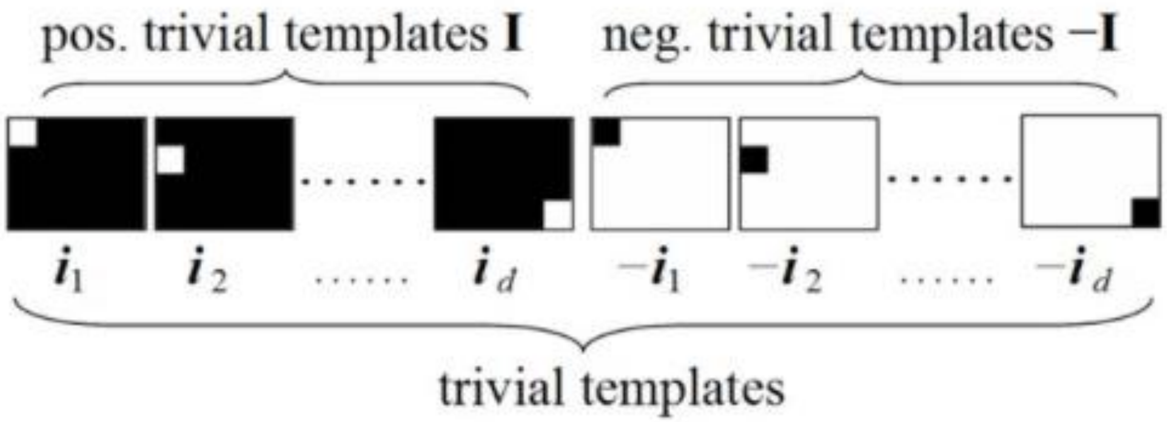
Introduction

- In this paper, we develop a robust visual tracking framework by casting the tracking problem as finding a sparse approximation in a template subspace.



Introduction

- Motivated by the work in [30], we propose handling occlusion using trivial templates.
- Each trivial template has only one non- zero element

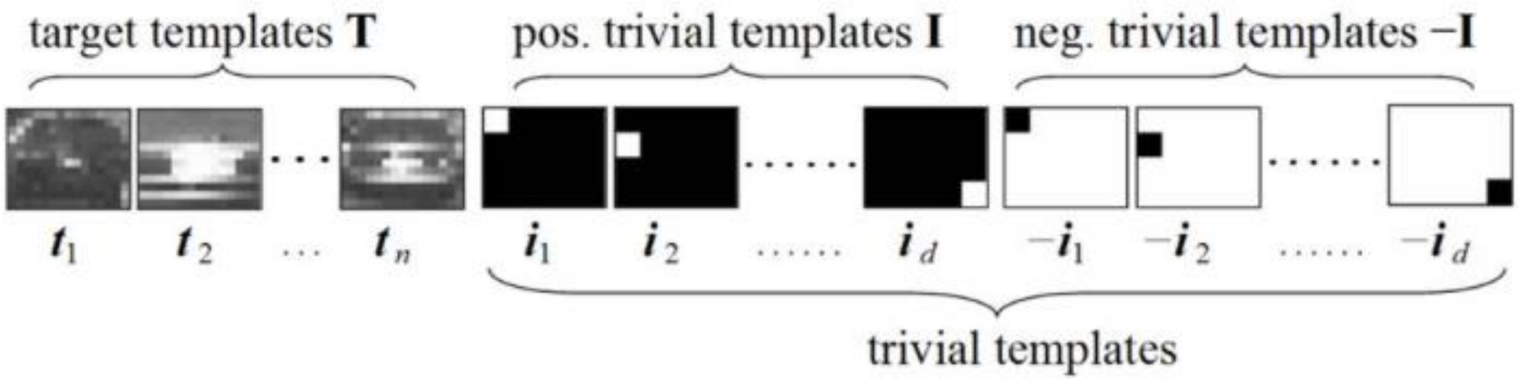


Introduction

- [30] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. “Robust Face Recognition via Sparse Representation”, *PAMI*

Introduction

- During tracking, a target candidate is represented as a linear combination of the template set composed of both target templates and trivial templates.



Introduction

- Intuitively, a good target candidate can be efficiently represented by the target templates.
- This leads to a sparse coefficient vector.
- Two additional components are included in our approach to further improve robustness.
  - Enforce nonnegative constraints to the sparse representation.
  - Dynamically update the target template set to keep the representative templates throughout the tracking procedure.

Related works

- Tracking can be considered as finding the minimum distance from the tracked object to the subspace represented by the training data or previous tracking results [4, 17].
- The most relevant work is [30] where sparse representation is applied for robust face recognition.

Related works

- [4] M. J. Black and A. D. Jepson. “Eigentracking: Robust matching and tracking of articulated objects using a view-based representation”, *IJCV*
- J. Ho, K.-C. Lee, M.-H. Yang, and D. Kriegman. “Visual tracking using learned subspaces”, *CVPR*

Sparse Representation of a Tracking Target

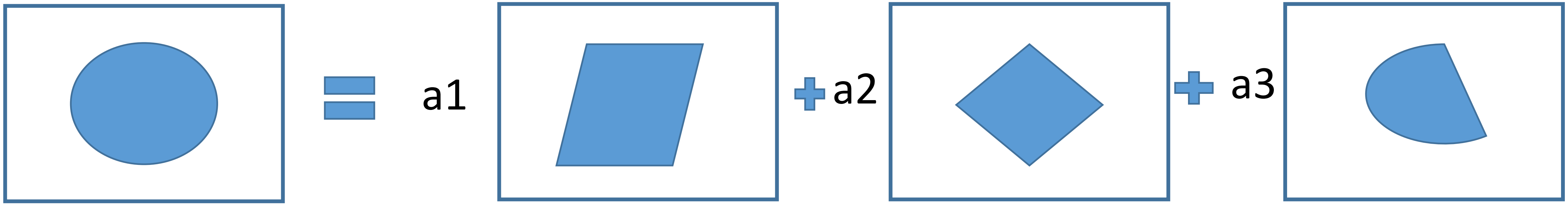
- Given target template set **T** = $[t_1 \ldots t_n] \in \mathbb{R}^{d \times n}$(d>>n), a tracking result **y** $\in \mathbb{R}^d$ approximately lies in the linear span of **T.**

$$\mathbf{y} \approx \mathbf{T}\mathbf{a} = a_1\mathbf{t}_1 + a_2\mathbf{t}_2 + \cdots + a_n\mathbf{t}_n \ , \qquad (4)$$

- where **a** = $(a1,a2,\cdots,an)^T \in \mathbb{R}^n$ is called a *target coefficient vector*.

Sparse Representation of a Tracking Target

$$\mathbf{y} \approx \mathbf{T}\mathbf{a} = a_1\mathbf{t}_1 + a_2\mathbf{t}_2 + \cdots + a_n\mathbf{t}_n \ , \qquad (4)$$



Sparse Representation of a Tracking Target

- In many visual tracking scenarios, target objects are often corrupted by noise or partially occluded.
- To incorporate the effect of occlusion and noise, Equation 4 is rewritten as

$$\mathbf{y} = \mathbf{T}\mathbf{a} + \epsilon \qquad (5)$$

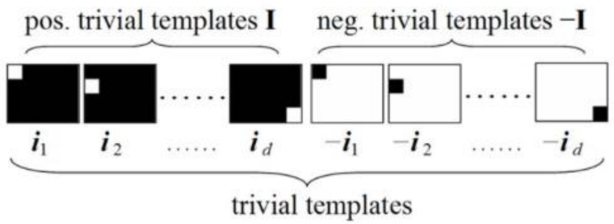
- Where ε is the error vector, the nonzero entries of ε indicate the pixels in **y** that are corrupted or occluded.

Sparse Representation of a Tracking Target

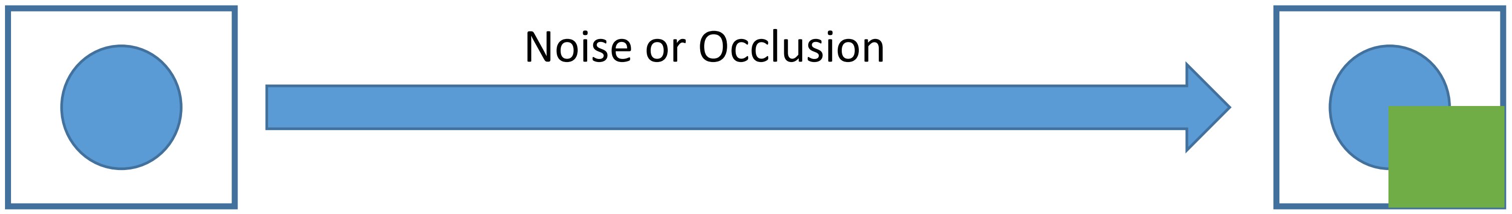
- Following the scheme in [30], we can use trivial templates **I** = $[i_1, i_2$, ..., i_d] $\in \mathbb{R}^{d\ x\ d}$ to capture the occlusion as

$$\mathbf{y} = [\mathbf{T}, \quad \mathbf{I}] \begin{bmatrix} \mathbf{a} \\ \mathbf{e} \end{bmatrix} , \qquad (6)$$

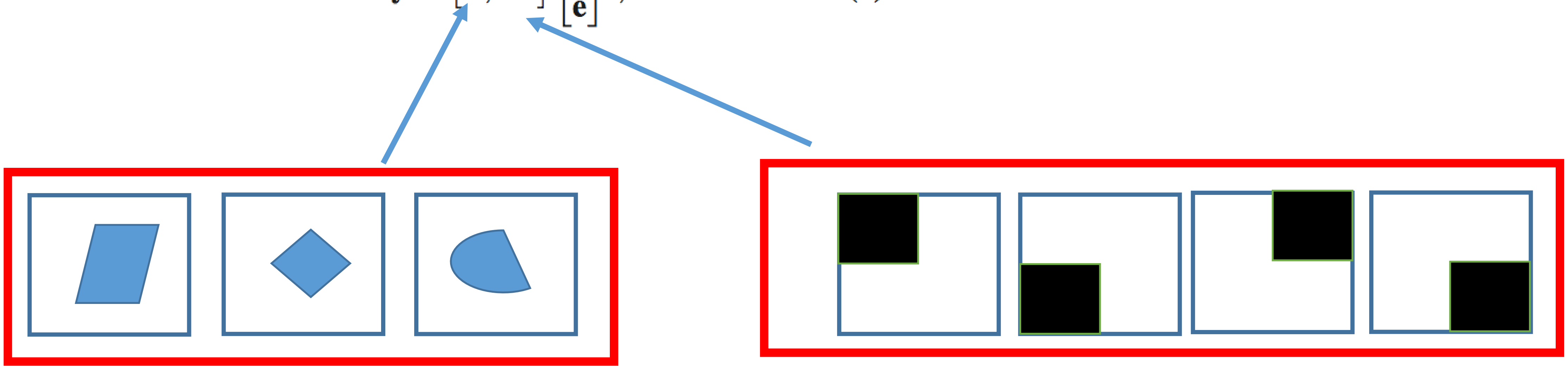
- Where a *trivial template* $i_1 \in \mathbb{R}^d$ is a vector with only one nonzero entry.(I is a identity matrix) and **e** = $(e_1, e_2, \cdots, e_d)^T \in \mathbb{R}^d$ is called a *trivial coefficient vector.*



Sparse Representation of a Tracking Target



$$\mathbf{y} = [\mathbf{T}, \quad \mathbf{I}] \begin{bmatrix} \mathbf{a} \\ \mathbf{e} \end{bmatrix}, \qquad (6)$$



Nonnegative Constraints

- In principle, the coefficients in **a** can be any real numbers if the target templates are taken without restrictions.

- However, a tracking target can always represented by the target templates dominated by nonnegative coefficients.

Nonnegative Constraints

- The templates that are most similar to the tracking target are positively related to the target.
- In new frames the appearance of targets may change, but new templates will be brought in.
- And the coefficients will still be positive for the most similar target templates in the following frames.

Nonnegative Constraints

- Another important argument for including nonnegative coefficients comes from their ability to filter out clutter that is similar to target templates at "reversed intensity patterns".

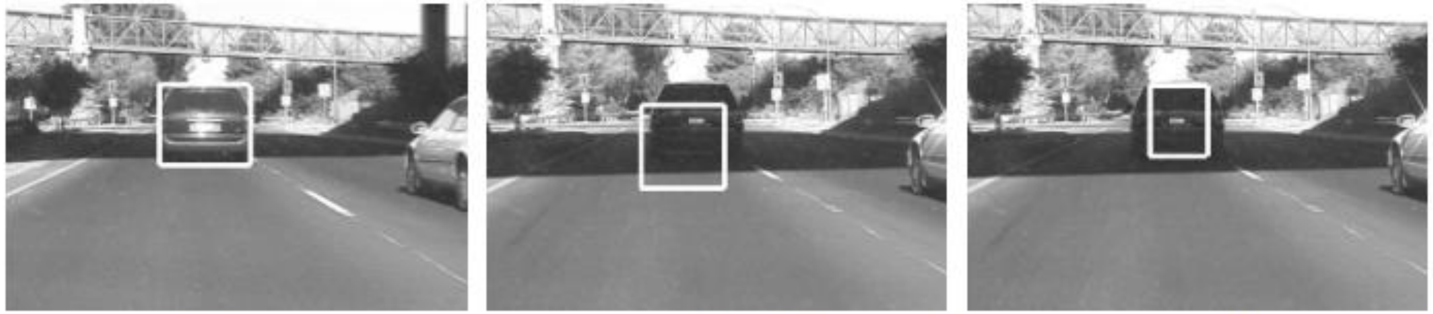


Figure 2. Left: target template. Middle: tracking result without non-negativity constraint. Right: tracking result with non-negativity constraint.

Nonnegative Constraints

- It is unreasonable to put such constraints directly on the trivial coefficient vector **e**.
- For this reason, we propose extending the trivial templates by. including *negative trivial templates* as well

$$\mathbf{y} = [\mathbf{T}, \quad \mathbf{I}] \begin{bmatrix} \mathbf{a} \\ \mathbf{e} \end{bmatrix} , \tag{6}$$

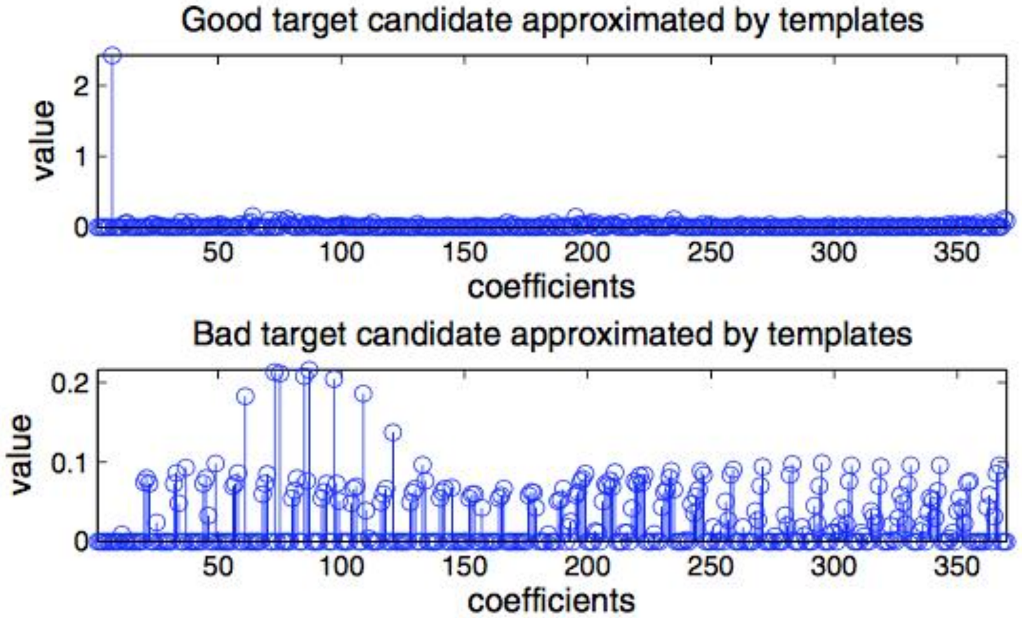
$$\mathbf{y} = [\mathbf{T}, \quad \mathbf{I}, \quad -\mathbf{I}] \begin{bmatrix} \mathbf{a} \\ \mathbf{e}^{+} \\ \mathbf{e}^{-} \end{bmatrix} \hat{=} \mathbf{B}\mathbf{c} \ , \quad \text{s.t. } \mathbf{c} \geq 0 \ , \tag{7}$$

Achieving Sparseness through L_1 Minimization

- For a good target candidate, there are only a limited number of nonzero coefficients in e^+ and e^- that account for the noise and occlusion.
- The target can be presented perfectly by a limited number of target templet.

Achieving Sparseness through L_1 Minimization

- Good and Bad target candidates.



Achieving Sparseness through L_1 Minimization

- Consequently, we want to have a sparse solution to (7).

$$\mathbf{y} = [\mathbf{T}, \quad \mathbf{I}, \quad -\mathbf{I}] \begin{bmatrix} \mathbf{a} \\ \mathbf{e}^+ \\ \mathbf{e}^- \end{bmatrix} \hat{=} \mathbf{B}\mathbf{c} \ , \quad \text{s.t.} \ \mathbf{c} \geq 0 \ , \qquad (7)$$

- We exploit the com- pressibility in the transform domain by solving the problem as an L_1-regularized least squares problem.

$$\min ||\mathbf{B}\mathbf{c} - \mathbf{y}||_2^2 + \lambda||\mathbf{c}||_1 \ , \qquad (8)$$

Achieving Sparseness through L_1 Minimization

- We then find the tracking result by finding the smallest residual after projecting on the target template subspace, i.e., $||\mathbf{y} - \mathbf{Ta}||_2$

Template Update

- The object is tracked by extracting a template from the first frame and finding the object of interest in successive frames.
- A fixed appearance template is not sufficient to handle recent changes in the video.

Template Update

- Object appearance remains the same only for a certain period of time.
- But eventually the template is no longer an accurate model of the object appearance.
- How often should we update the template?
  1. If we do not update the template, the template cannot capture the appearance variations.(illumination and pose changes)
  2. If we update the template too often, small errors are introduced each time the template is updated.

Template Update

- We tackle this problem by dynamically updating the target template set **T**.

Template Update

- One important feature for L_1 minimization is that it favors the template with larger norm.

$$\min ||\mathbf{Bc} - \mathbf{y}||_2^2 + \lambda||\mathbf{c}||_1 \ , \qquad (8)$$

- We exploit the characteristic by introducing a weight W_i=$||t_i||_2$
- The larger the weight is, the more important the template is.

Template Update

Algorithm 1 Template Update

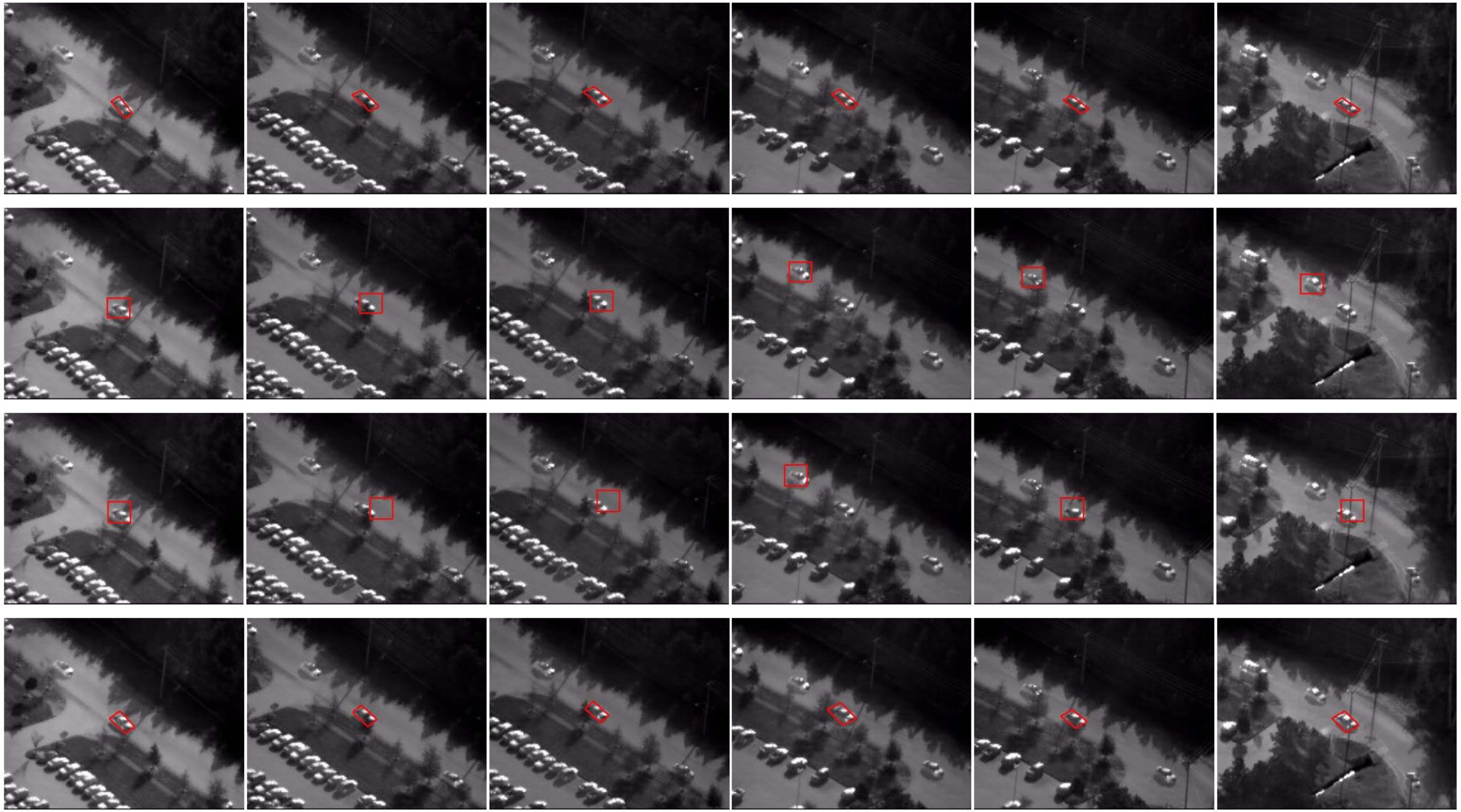
```
y is the newly chosen tracking target.
a is the solution to (8).
w is current weights, such that w_i ← ||t_i||_2.
τ is a predefined threshold.
Update weights according to the coefficients of the target templates. w_i ← w_i * exp(a_i).
if ( sim(y, t_m) < τ ), where sim is a similarity function. It can be the angle between two vectors or SSD between two vectors after normalization. t_m has the largest coefficient a_m, that is, m = arg max_{1≤i≤n} a_i then
    i_0 ← arg min_{1≤i≤n} w_i
    t_{i_0} ← y,    /*replace an old template*/.
    w_{i_0} ← median(w),    /*replace an old weight*/.
end if
Normalize w such that sum(w) = 1.
Adjust w such that max(w) = 0.3 to prevent skewing.
Normalize t_i such that ||t_i||_2 = w_i.
```

$$\min ||\mathbf{B}\mathbf{c} - \mathbf{y}||_2^2 + \lambda||\mathbf{c}||_1 \ , \qquad (8)$$

Experimental Result

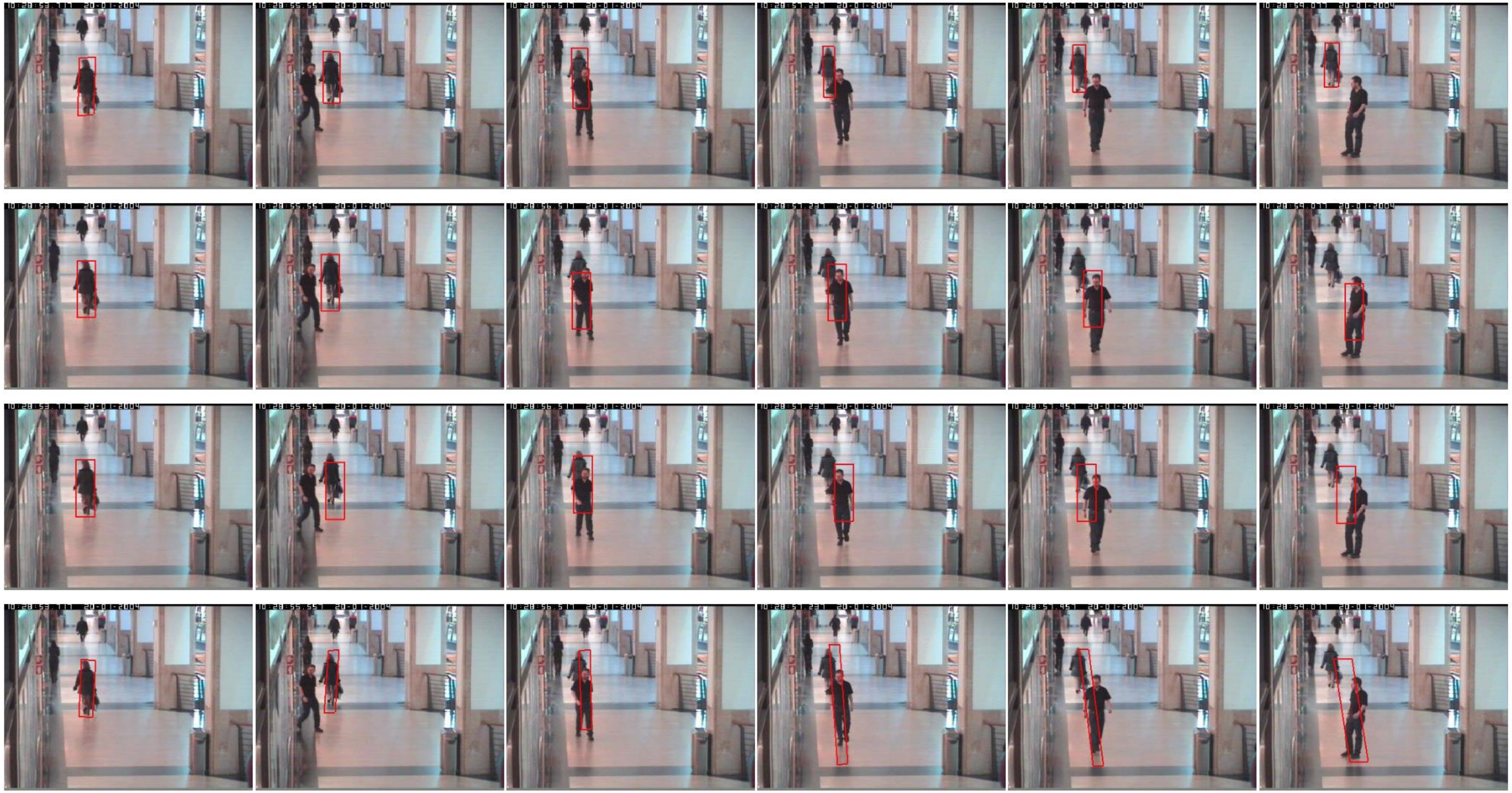
- The first two videos consist of 8-bit gray scale images while the last two are composed of 24-bit color images.
- We compare the tracking results of our proposed method with those of state-of-the-art standard Mean Shift (MS) tracker [9], covariance (CV) tracker [27], and appearance adaptive particle filter (AAPF) tracker [34].

Experimental Result

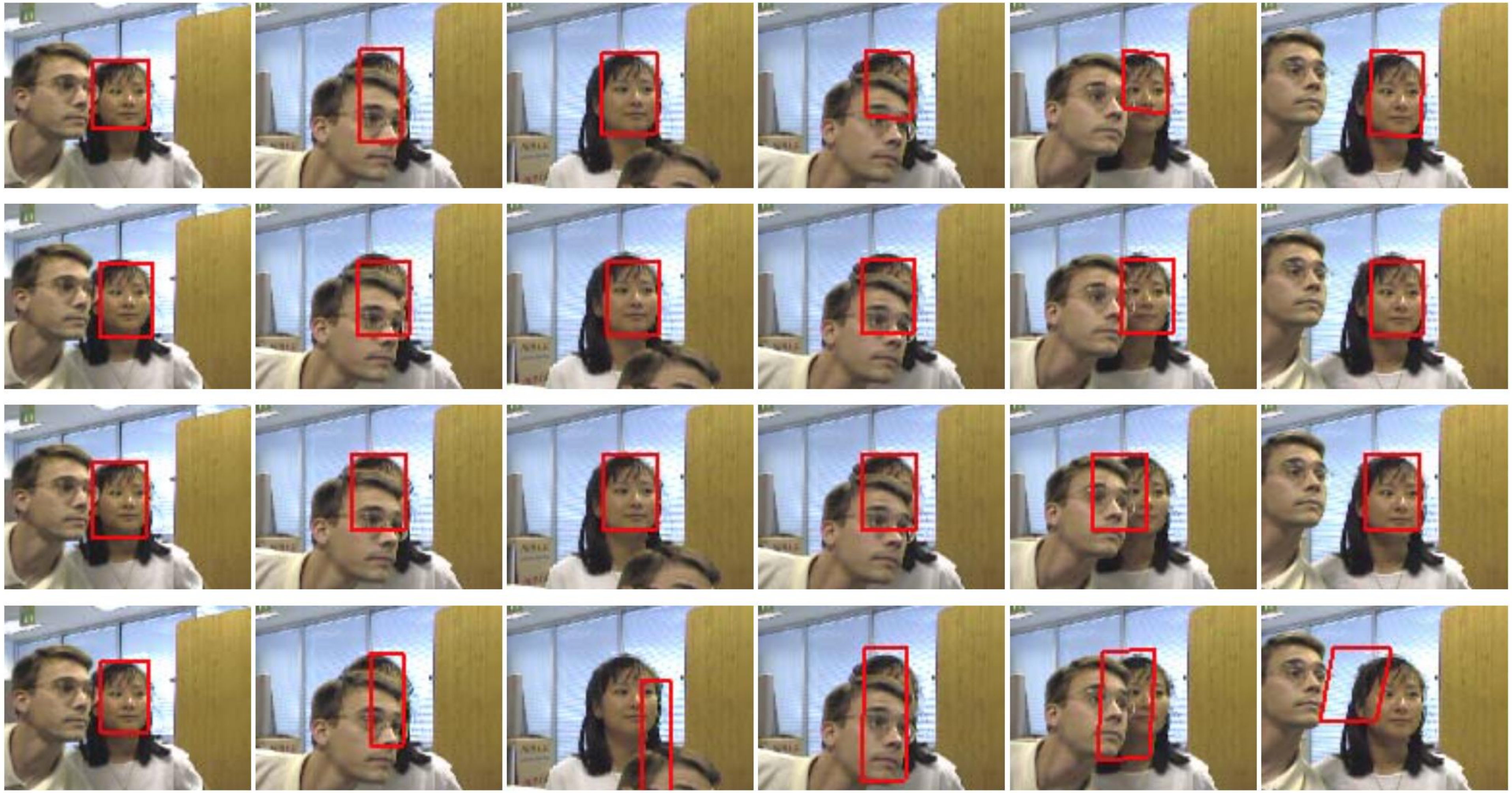


Experimental Result

Experimental Result



Experimental Result



Conclusion

- In this paper we propose using a sparse representation for robust visual tracking.
- We model tracking as a sparse approximation problem and solve it through an l1-regularized least squares approach.
- For further robustness, we introduce nonnegative constraints and dynamic template updating in our approach.