FAsT-Match:
Fast Affine Template Matching

Simon Korman (TAU)
Gilad Tsur (WIS)
Daniel Reichman (WIS)
Shai Avidan (TAU)

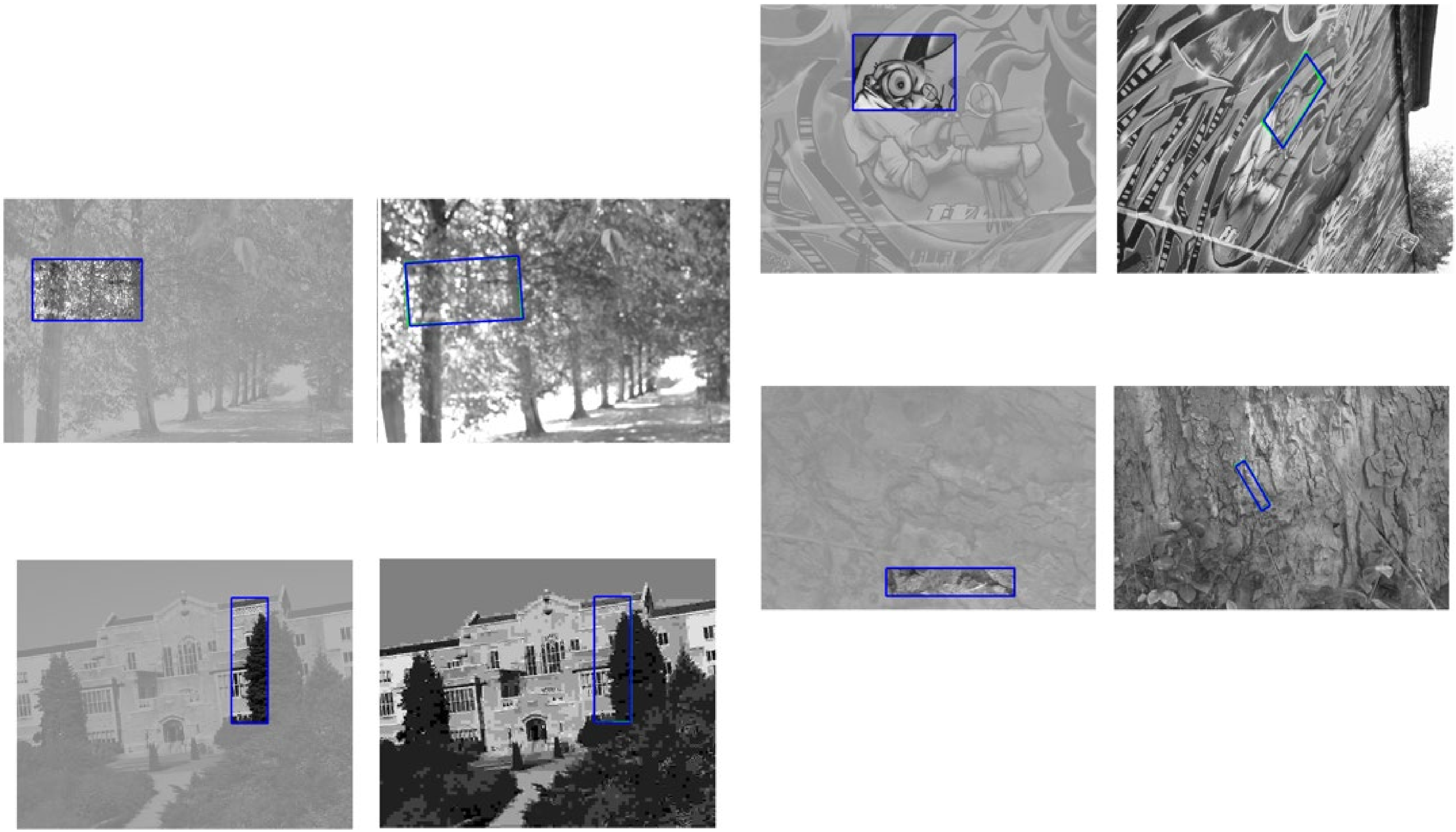
***to appear at CVPR 2013**

Generalized Template Matching

- Find the best …/Translation/Euclidean/Similarity/**Affine**/Projective/… transformation between two given images:

Some results I

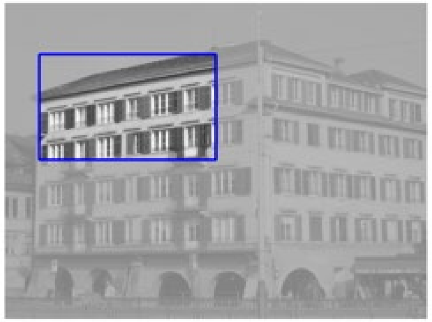
Some results II



Some results III





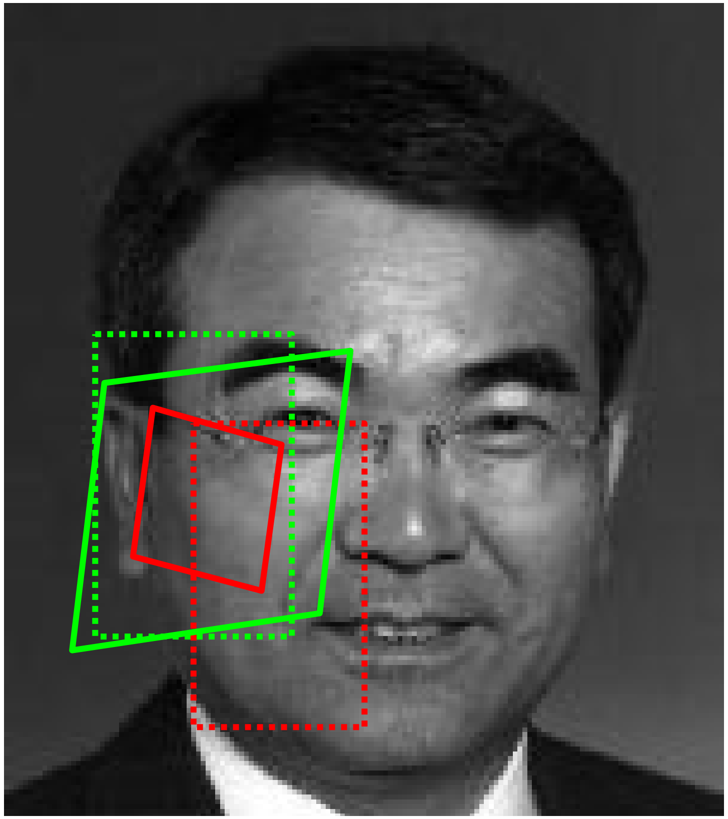



Generalized Template Matching

- The algorithm:
  1. Take a sample of the Affine transformations
  2. Evaluate each transformation in the sample
  3. Return the best

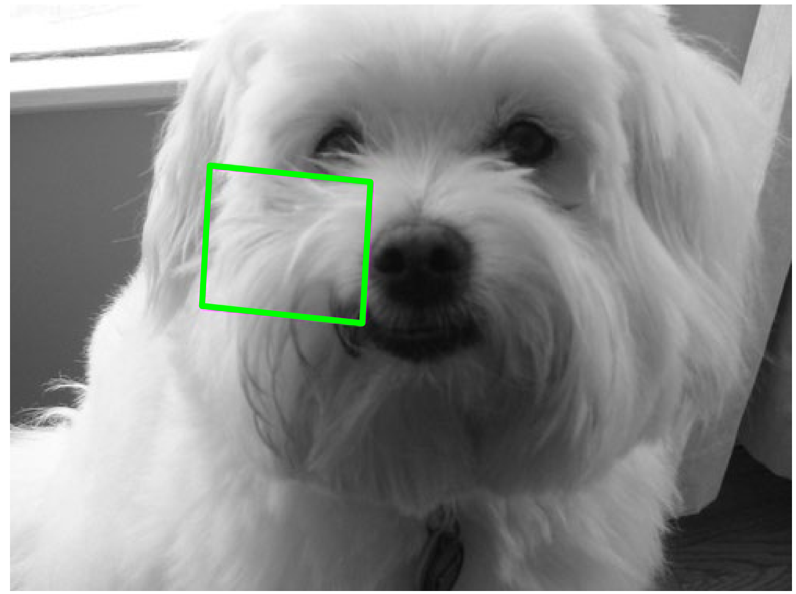
- **Questions**:
  - Which sample to use?
  - How does is guarantee a bound?

Direct methods – parametric OF



Lucas, Kanade "**An iterative image registration technique with an application to stereo vision**" [ICAI 1981]
Baker, Matthews "**Lucas-Kanade 20 years on: A unifying framework**" [IJCV 04]

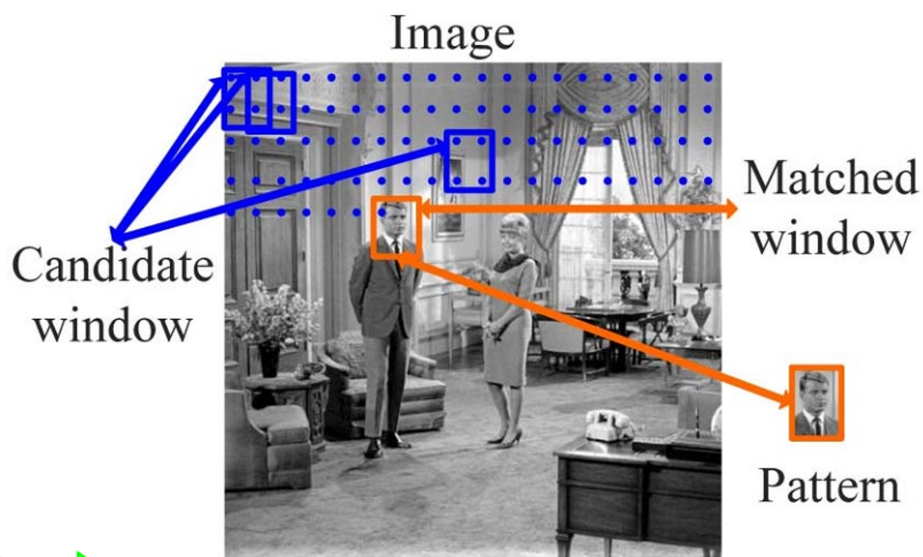
Indirect methods (feature based)

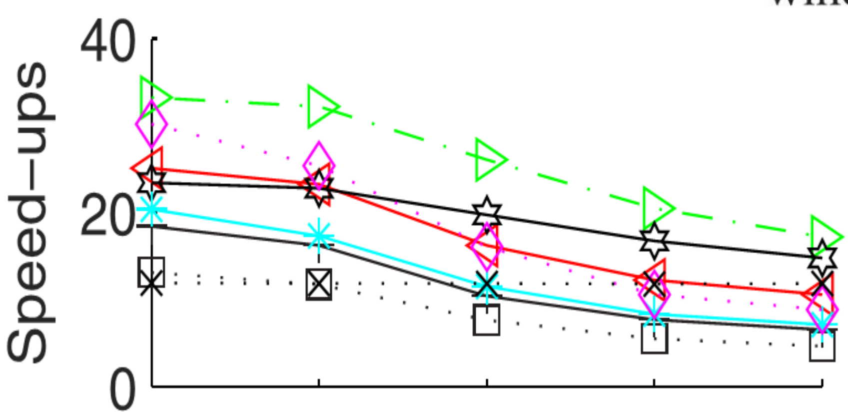


Lowe "**Distinctive image features from scale-invariant key-points**" [IJCV 04]
Morel, Yu "**Asift: A new framework for fully affine invariant image comparison**" [SIAM 09]
M.A. Fichler, R.C. Bolles "**Random sample consensus**" [Comm. of ACM 81]

Template Matching - 2D translation (2 dof)

- “**Performance Evaluation of Full Search Equivalent Pattern Matching Algorithms**” [Ouyang, Tombari, Mattoccia, Di Stefano, Cham, *TPAMI 2012*]





Template Matching – more general (3-4 dof)

- + Rotation (3 dof)
  - "**Rotation Invariant Template Matching**" [Fredrikson, 2001]
  - "**Rotation-invariant pattern matching using wavelet decomposition**" [Tsai, Chiang 2002]
- + Rotation + uniform scale (4 dof)
  - "**Grayscale template-matching invariant to rotation, scale, translation, brightness and contrast**" [Kim, Araujo 2007]
  - "**Retrieval of translated, rotated and scaled color textures**" [Yao, Chen 2003]

Fig. 6. Template images (51×51 pixels).

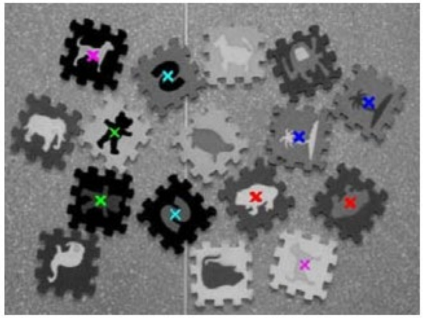
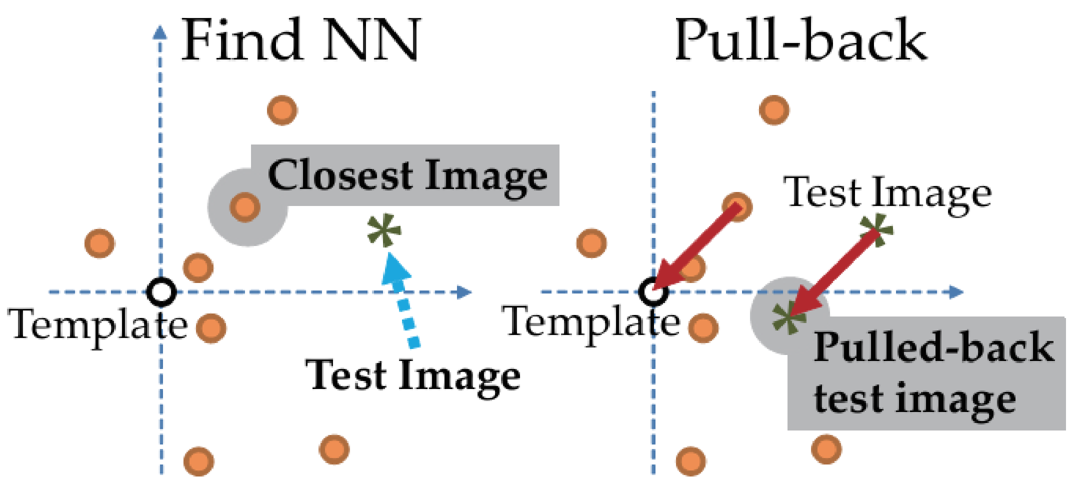


Fig. 7. Result of detection of the 5 templates.

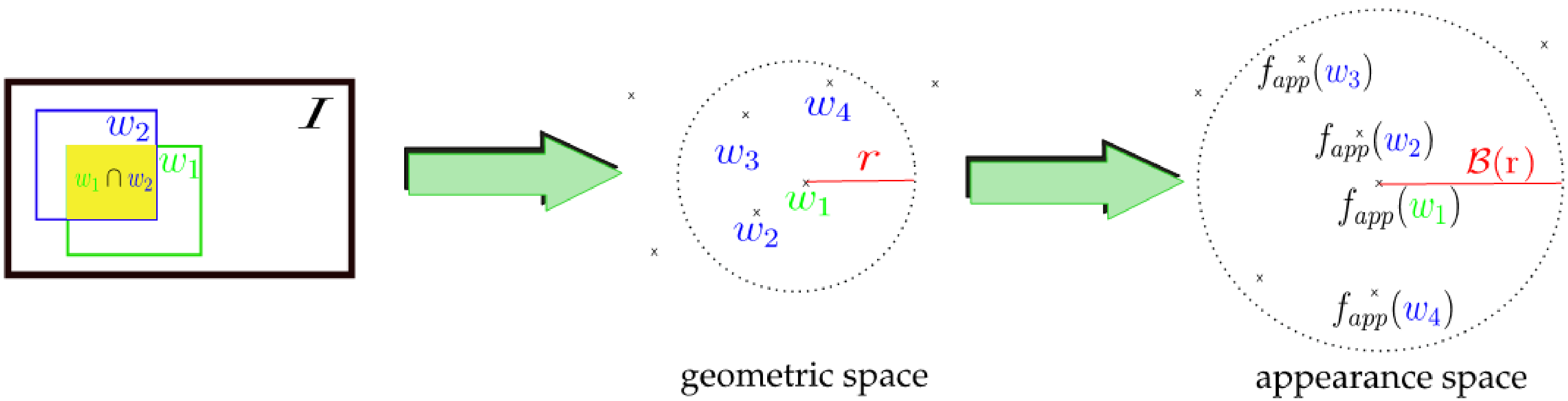
Related work

- "**A Globally Optimal Data-Driven Approach for Image Distortion Estimation**" [Tian, Narasimhan, CVPR 2010]



Related work

- "**Exploiting spatial overlap to efficiently compute appearance distances between image windows**"
  [Alexe, Petrescu, Ferrari, NIPS 2011]

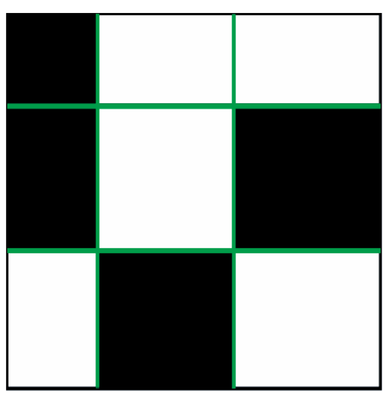


Sublinear Algorithms for Images

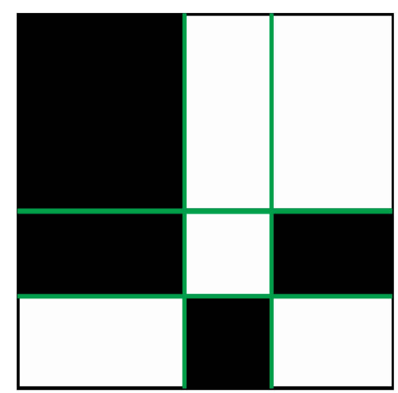
- Testing Image Properties [Raskhodnikova, RANDOM, 2003]:
  - Properties: Connectivity, Convexity, Half-Plane
  - Global properties can be checked locally

Sublinear Algorithms for Images II

- Image Partitioning [Kleiner et. al. PAMI, 2011]:
  - Test if image can be partitioned into 0/1 squares according to a template
  - Property may be adapted for different uses

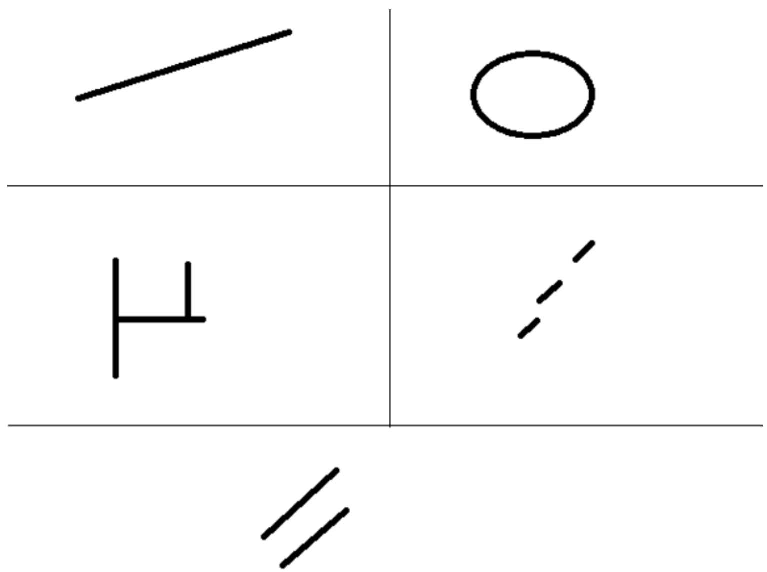


$$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$



Sublinear Algorithms for Images III

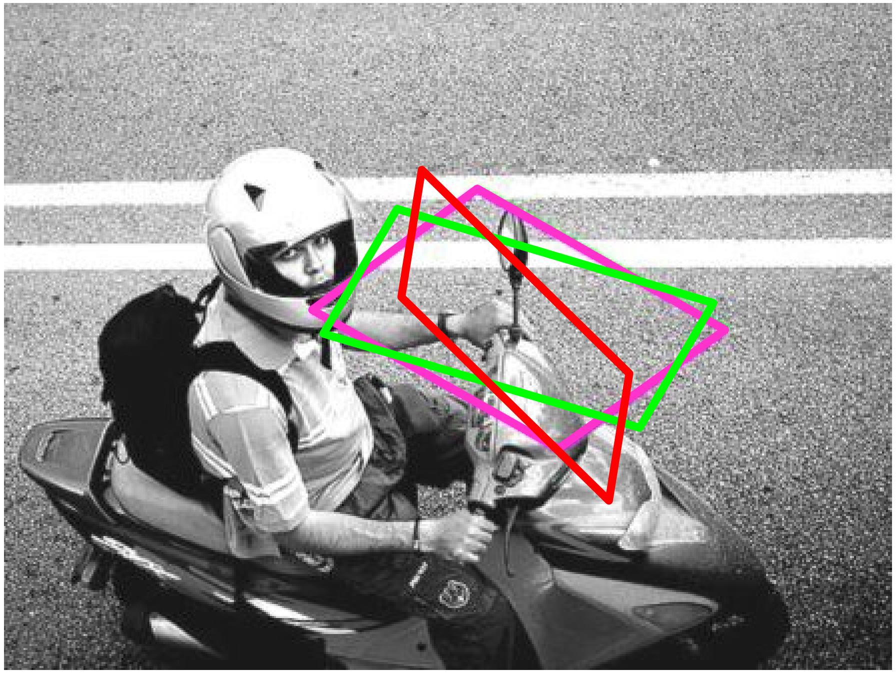
- Testing Sparse Images [Ron & Tsur, FOCS, 2010]:
  - Properties: Convexity, Connectivity etc.
  - Algorithms and analysis turn out interesting



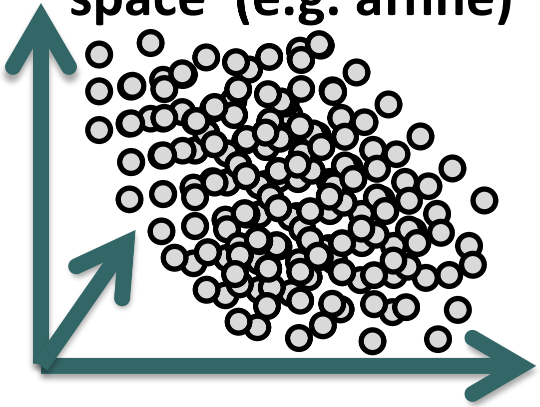
The Main Idea

template

image

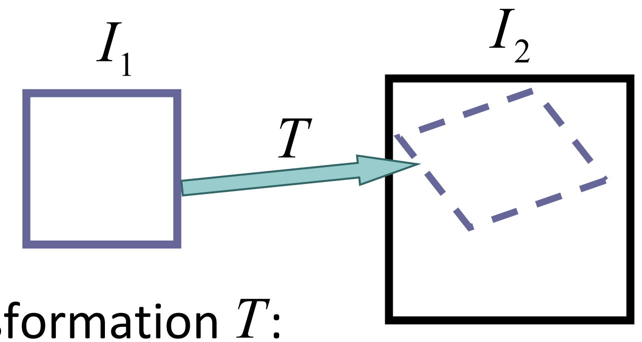


Transformation space (e.g. affine)



Formal Problem Statement

- **Input**: Grayscale image (template) I_1 $(n_1 \times n_1)$ and image I_2



- **Distance** with respect to **a <u>specific</u>** transformation T:

$$\Delta_T(I_1, I_2) = \frac{1}{n_1^2} \sum_{p \in I_1} \left| I_1(p) - I_2(T(p)) \right|$$

- **Distance** with respect to **<u>any</u>** transformation in a family Ψ (affinities):

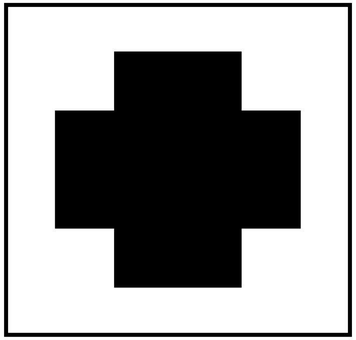
$$\Delta(I_1, I_2) = \min_{T \in \Psi} \Delta_T(I_1, I_2)$$

- **Goal**: Given $\delta > 0$, find a transformation T^* in Ψ for which:

$$\left| \Delta(I_1, I_2) - \Delta_{T^*}(I_1, I_2) \right| < \delta$$

Image Smoothness

- Total-Variation: $TV(I) = \sum_{p \in I} \max_{q \in N(p)} \left| I(p) - I(q) \right|$
- For $n \times n$ images with intensities in [*0*,*1*]: $TV \in [0, n^2]$
- **Smooth** images: $TV = O(n)$
- Intuition: In binary images *TV* = boundary-length



- Holds for grayscale as well - experiment

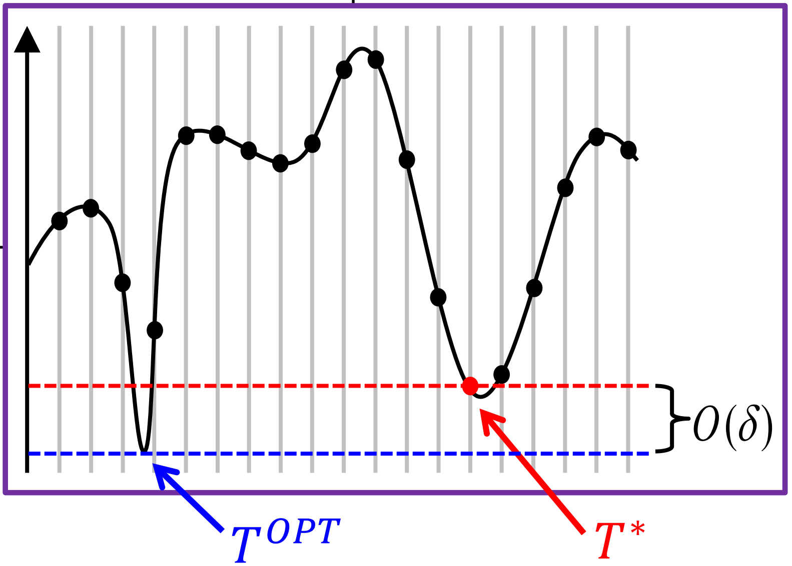
The Algorithm – take 1

- For each affine transformation T
  - Compute the distance $\Delta_T(I_1, I_2)$
- Return T^* with smallest distance

- ∞ transformations – need to discretize
- "**Combinatorial bounds and algorithmic aspects of image matching under projective transformations**" [Hundt & Liskiewicz MFCS, 2008]
  - Enumerate $\approx n^{18}$ affine transformations (for $n \times n$ images)
- **Guarantee**: best possible transformation

The Algorithm – take 2

- For each affine transformation T in a Net A_δ
  - Compute the distance $\Delta_T(I_1, I_2)$
- Return T^* with smallest distance

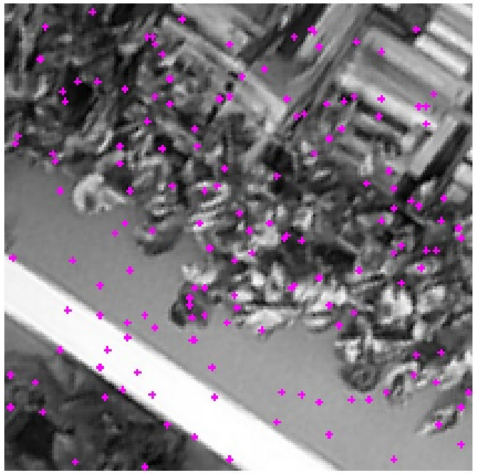


- Sample transformation space
  - build a Net A_δ of transformations
- **Guarantee**
  - 'δ – away' from best possible distance

$$\boxed{\left|\Delta_{T^{OPT}}(I_1, I_2) - \Delta_{T^*}(I_1, I_2)\right| = O(\delta)}$$

The Algorithm – take 3

- For each affine transformation T in a Net A_δ
  - ~~Compute~~ *Estimate* the distance $\Delta_T(I_1, I_2)$
- Return T^* with smallest ~~distance~~ *estimate*



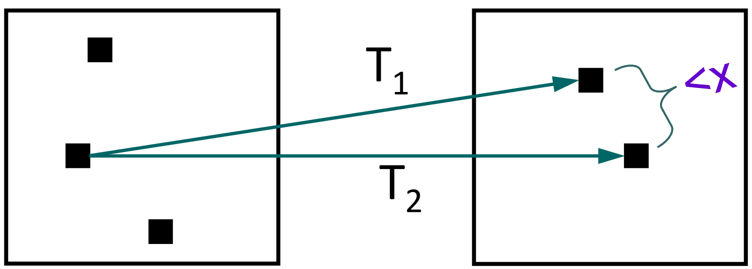
- Estimate the SAD to within O(δ)
- By sampling $\Theta(1/\delta^2)$ pixels
- Thus – total runtime is: $|A_\delta| \cdot \Theta(1/\delta^2) = \Theta(\frac{1}{\delta^8} \cdot (\frac{n_2}{n_1})^2)$

$$\Delta_{T^{OPT}}(I_1, I_2) = O(\delta)$$

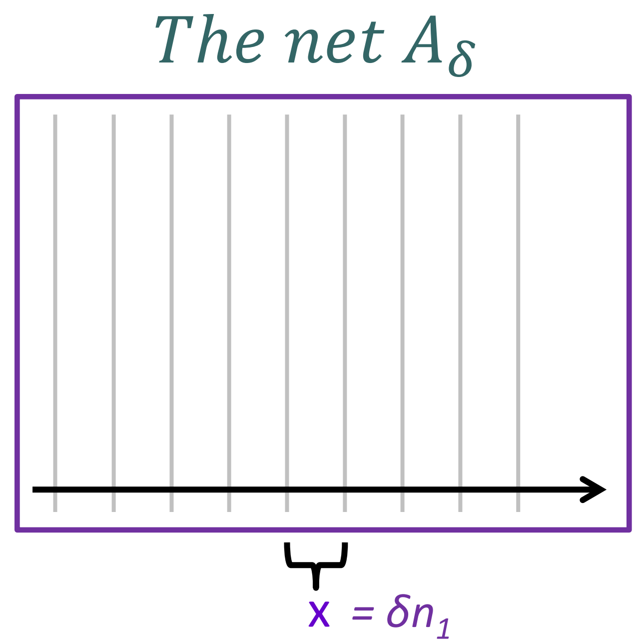
$$|\Delta_{T^{OPT}}(I_1, I_2) - \Delta_{T^*}(I_1, I_2)| = O(\delta)$$

The net A_δ

- Transformations T_1 and T_2 are x-close

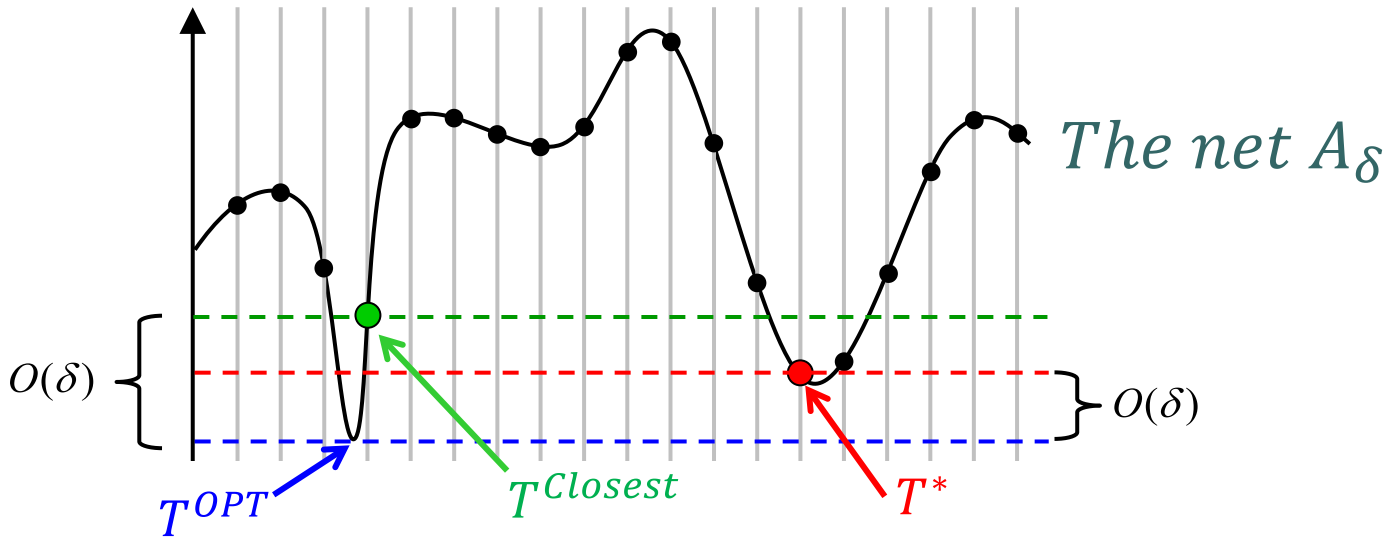


$$L_\infty(T_1, T_2) = \max_{p \in I_1} \left\| T_1(p) - T_2(p) \right\|_2 \leq x$$



- The Net A_δ
  - Any affine transformation is δn_1-close to some trans. in A_δ
    - (A_δ is a δn_1-cover of affine transformations)
  - Possible construction with size: $\Theta(\frac{1}{\delta^6} \cdot (\frac{n_2}{n_1})^2)$

Approximation – Proof Outline



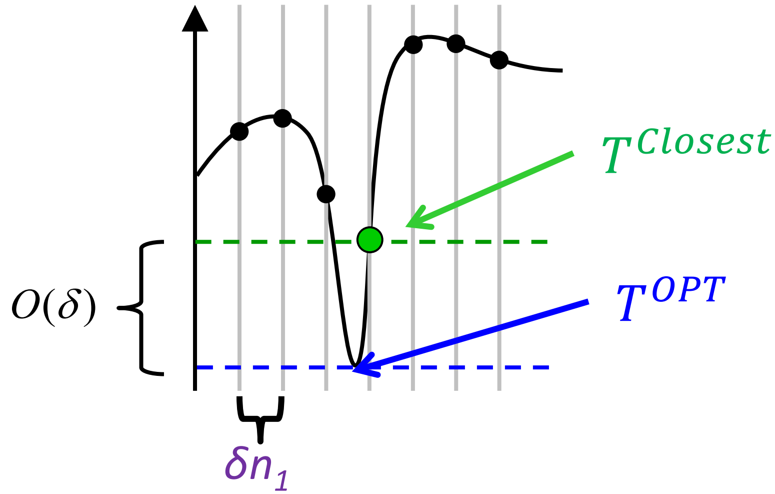
- Goal: $\boxed{\left|\Delta_{T^{OPT}} - \Delta_{T^{*}}\right| = O(\delta)}$

Main Thm: $\boxed{\left|\Delta_{T^{OPT}} - \Delta_{T^{Closest}}\right| = O(\delta)}$ ⇐
- A_δ construction
- Image smoothness

- Using full transformation evaluation: $\boxed{\Delta_{T^{*}} < \Delta_{T^{Closest}}}$
- Using full estimated evaluation: $\boxed{\Delta_{T^{*}} < \Delta_{T^{Closest}} + O(\delta)}$

Approximation – Proof Outline

The net A_δ



$$\boxed{\left|\Delta_{T^{OPT}} - \Delta_{T^{Closest}}\right| = O(\delta)}$$

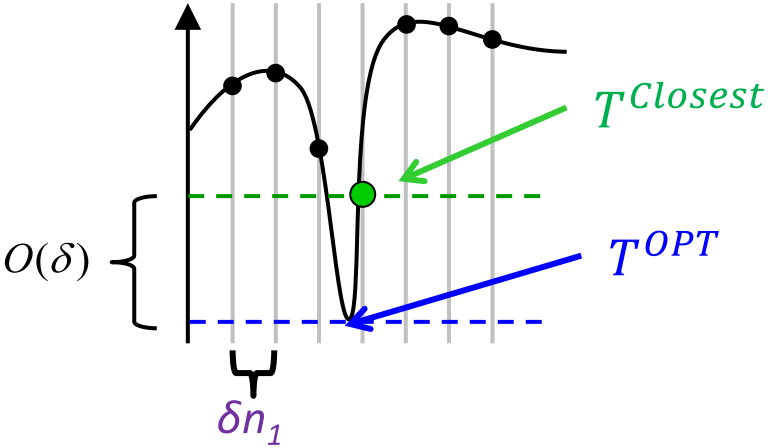
- A_δ construction
- Image smoothness

- Recall that $T^{Closest}$ and T^{OPT} are δn_1-close
- How worse can $T^{Closest}$ be compared with T^{OPT}?
- Not much – for **smooth** images

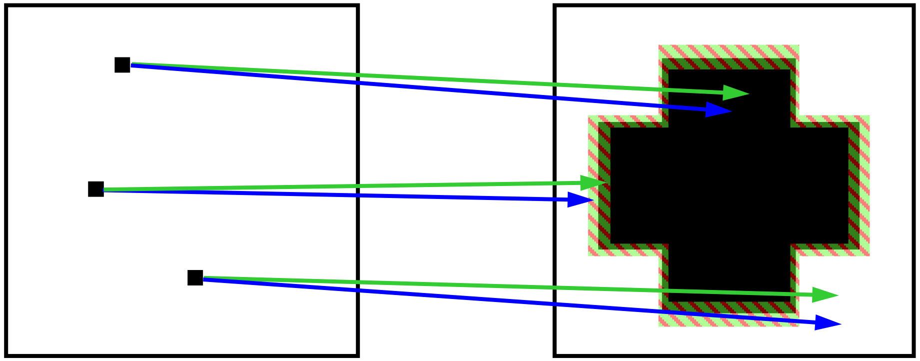
Approximation – Proof Outline

- A_δ construction
- Image smoothness

$$\boxed{\left|\Delta_{T^{OPT}} - \Delta_{T^{Closest}}\right| = O(\delta)}$$

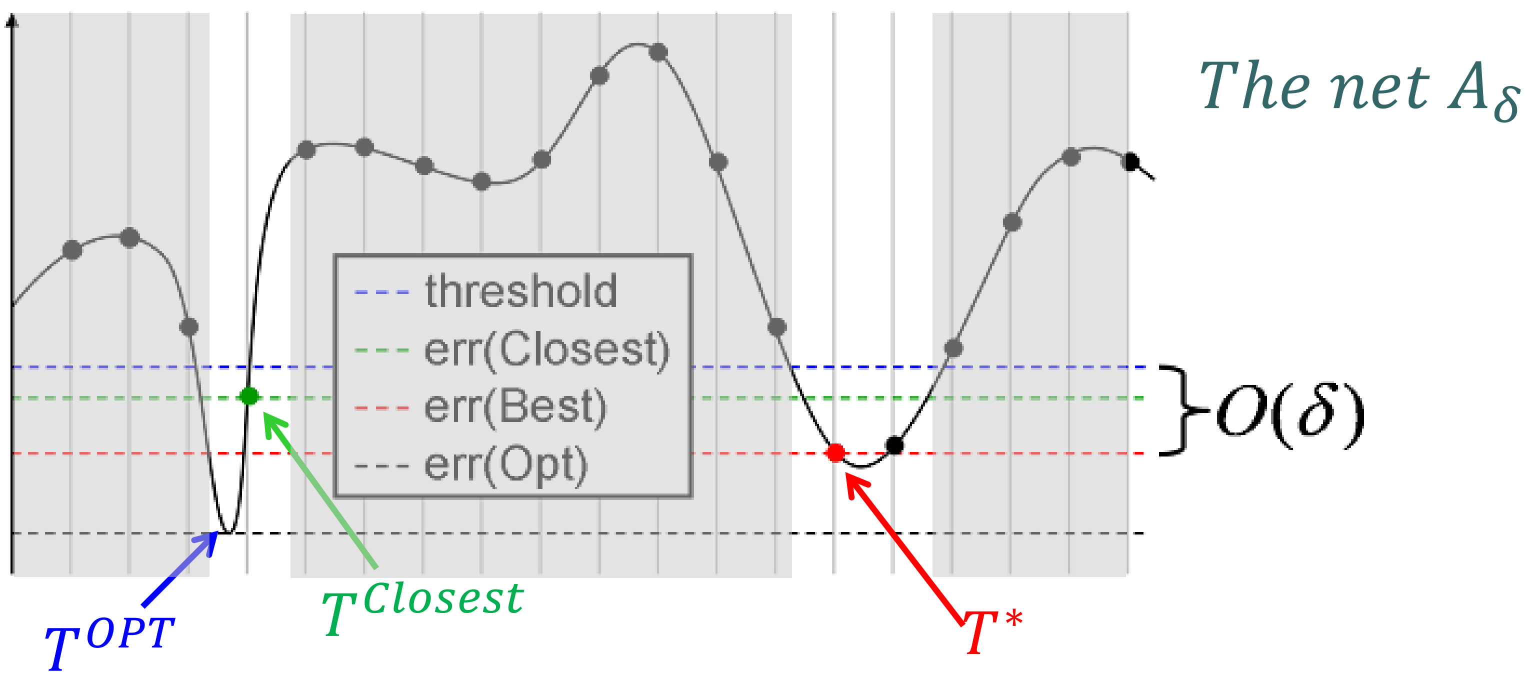


- $T^{Closest}$ and T^{OPT} are δn_1-close
- Intuition:
  - Error addition



Fast-Match:
a Branch-and-Bound Scheme

- Iteratively increase Net-precision (decrease δ)
- Throw away irrelevant transformation regions
  - $T^{Closest}$ is guaranteed to move to next round
  - (off-net neighbors of above- threshold points are worse than T^*)



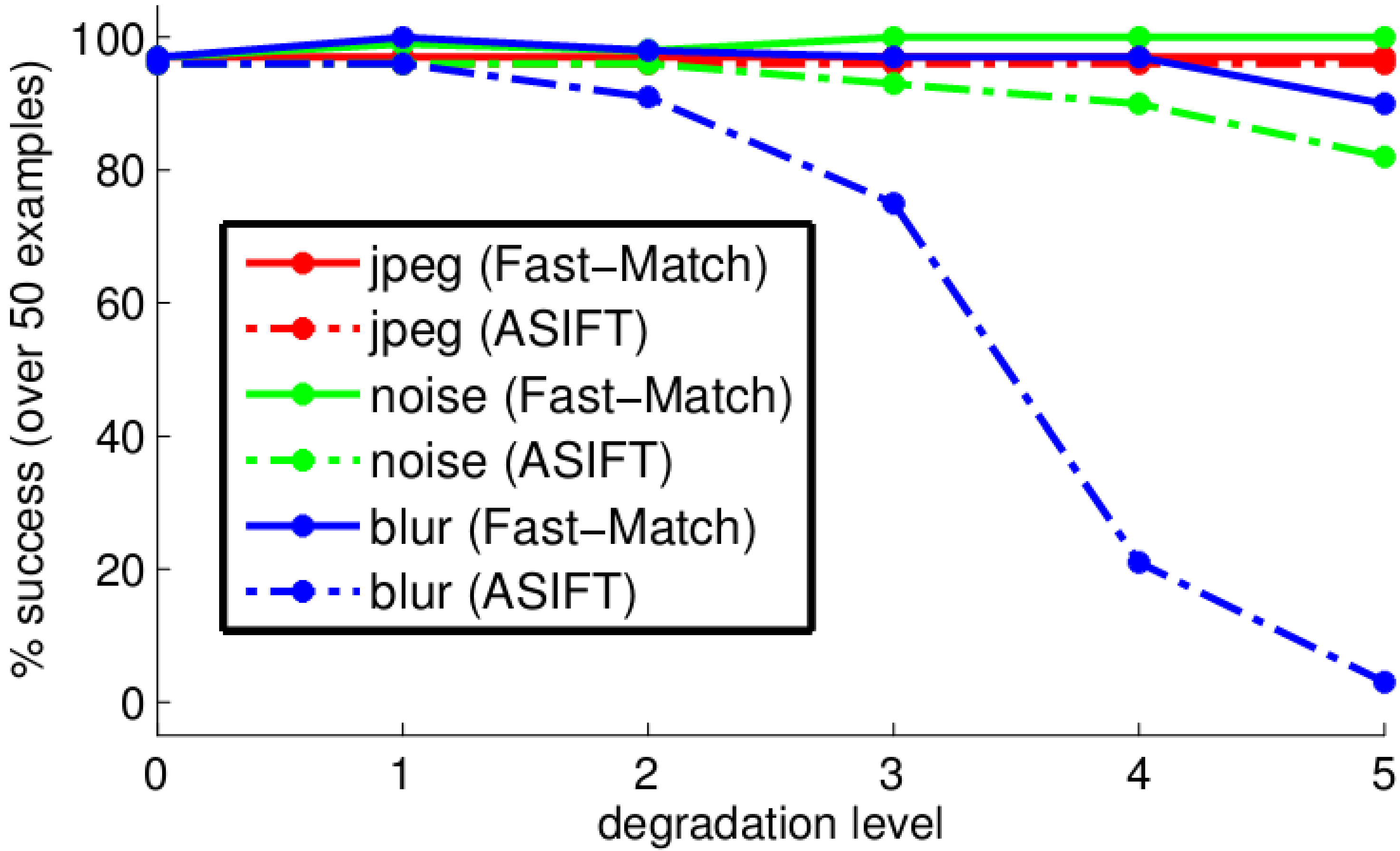
Experiment 1: Affine Template Matching

- Pascal VOC 2010 data-set
  - 200 random image/templates
  - Template dimensions of 10%, 30%, 50%, 70%, 90%
  - 'Comparison' to a feature-based method - ASIFT
  - Image degradations (template left in-tact):
    - **Gaussian Blur** with STD of {0,1,2,4,7,11} pixels
    - **Gaussian Noise** with STD of {0,5,10,18,28,41}
    - **JPEG compression** of quality {75,40,20,10,5,2}

Template Dimension	90%	70%	50%	30%	10%
avg. Fast-Match SAD err.	5.5	4.8	4.4	4.3	4.8
avg. ground truth SAD err.	4.1	4.1	4.0	4.4	6.1
avg. Fast-Match overlap err.	3.2%	3.3%	4.2%	5.3%	13.8%

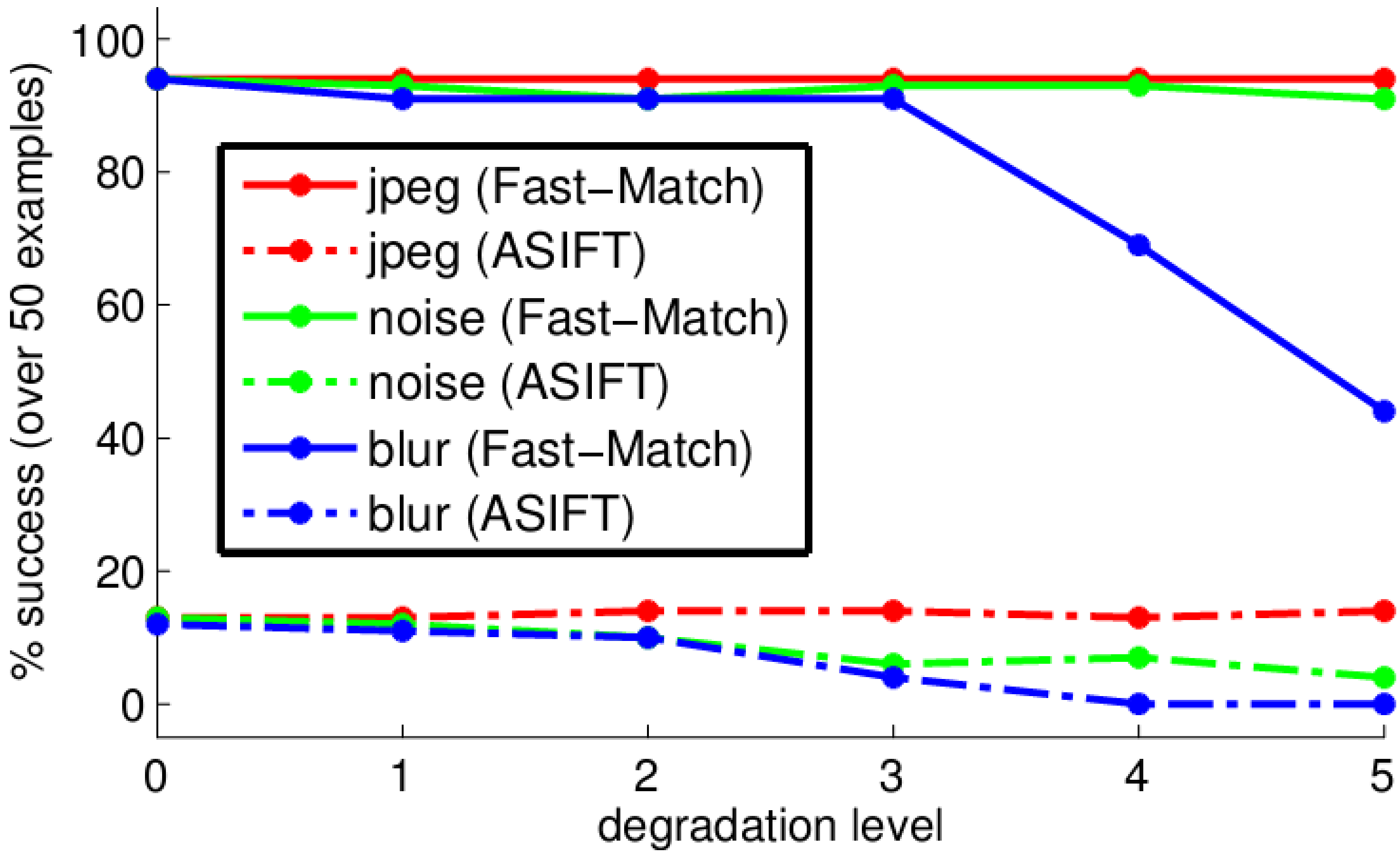
Experiment 1: Affine Template Matching

- Fast-Match vs. ASIFT – template dimension 50%



Experiment 1: Affine Template Matching

- Fast-Match vs. ASIFT – template dimension 20%

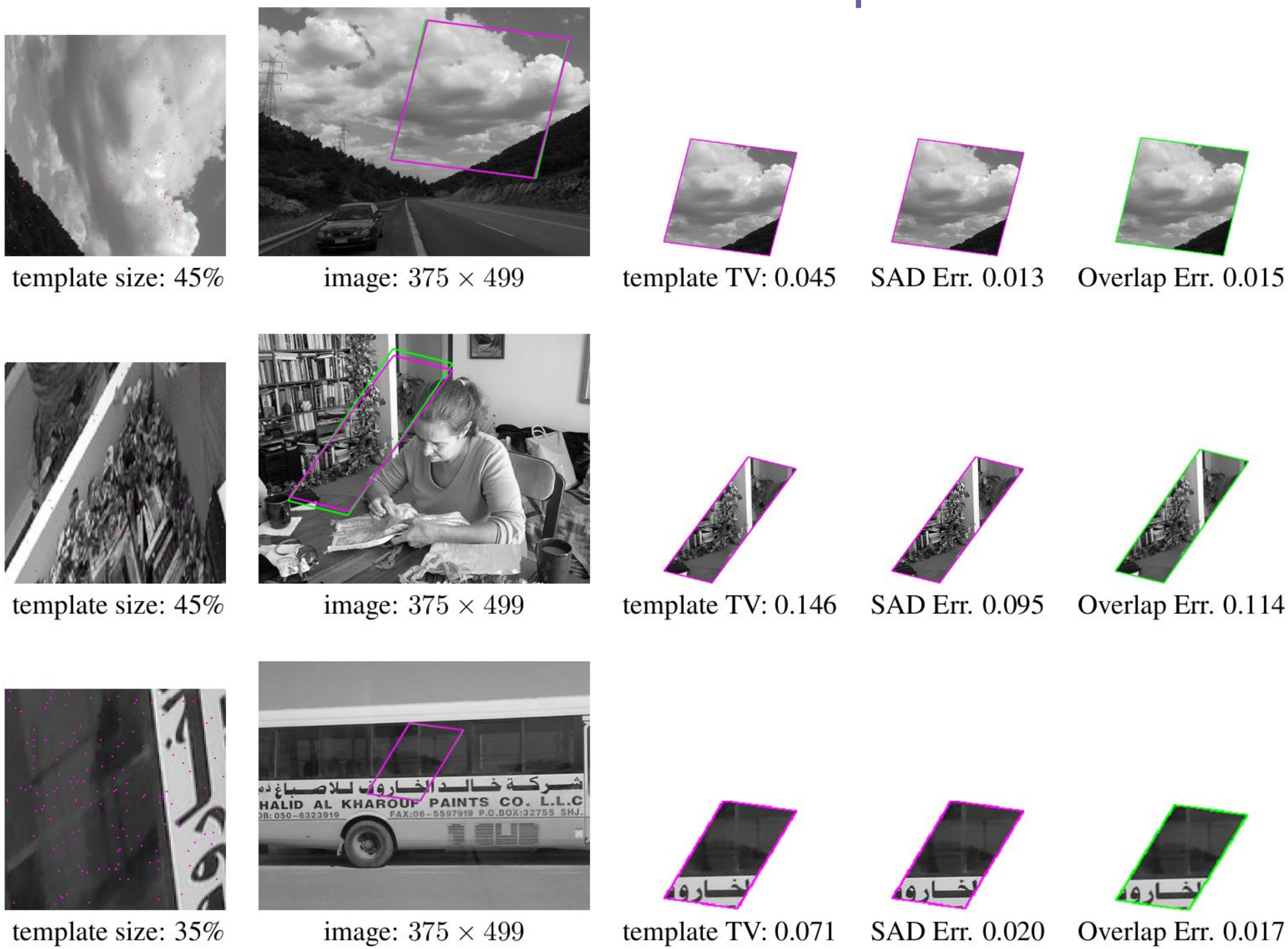


Experiment 1: Affine Template Matching

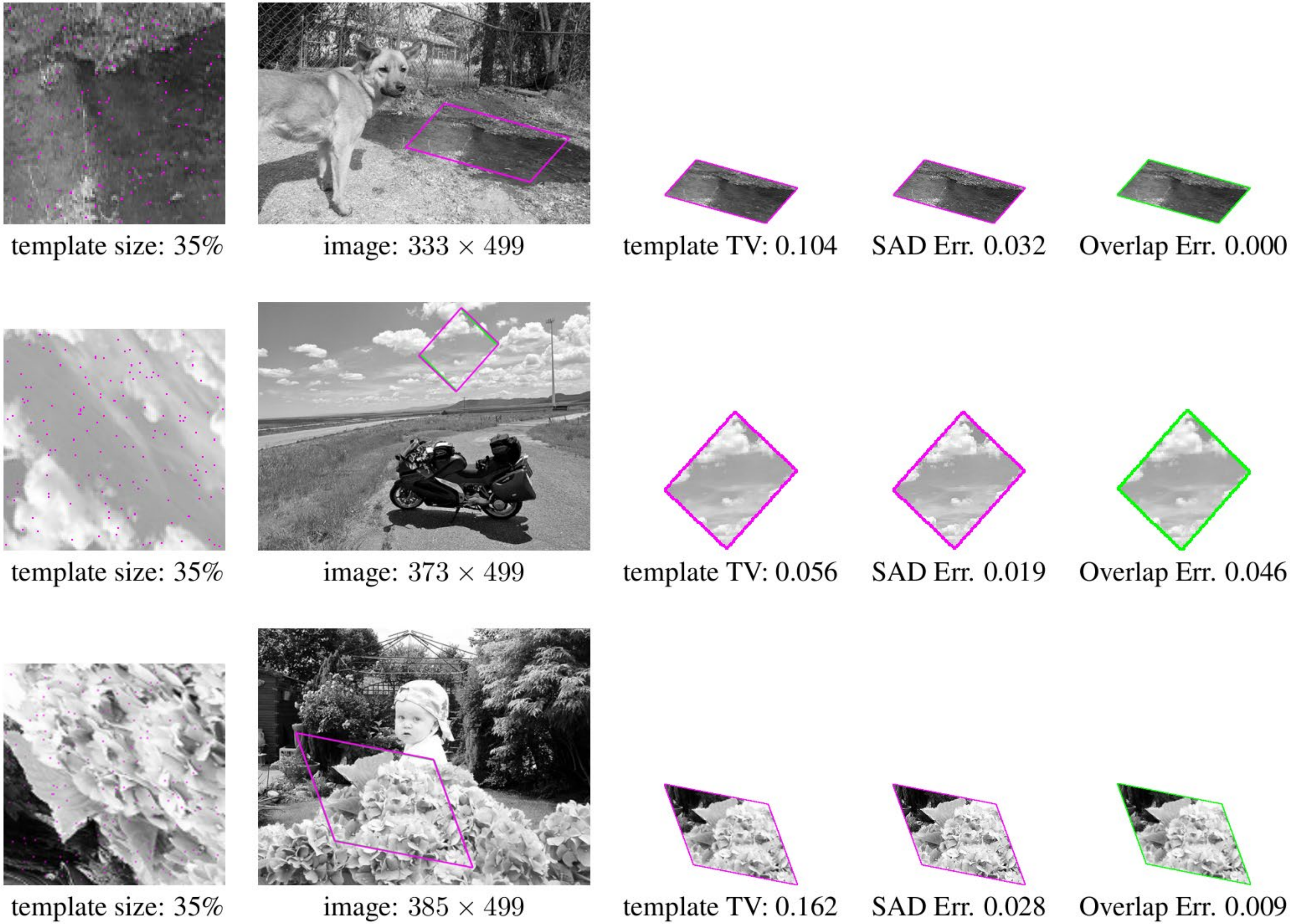
- Runtimes

Template Dimension	90%	70%	50%	30%	10%
ASIFT	12.2 s.	9.9 s.	8.1 s.	7.1 s.	NA
Fast-Match	2.5 s.	2.4 s.	2.8 s.	6.4 s.	25.2 s.

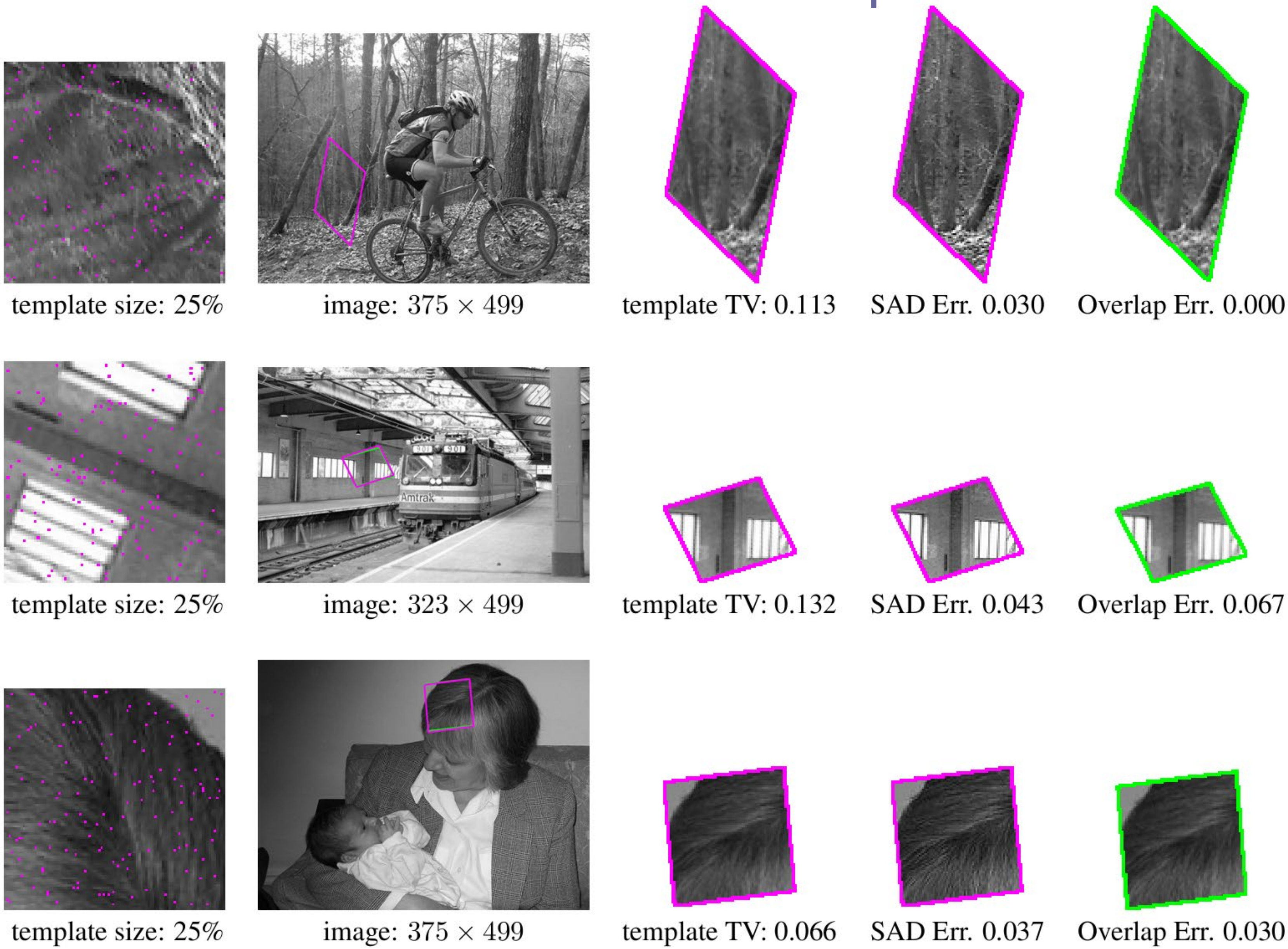
Template Dim: 45%



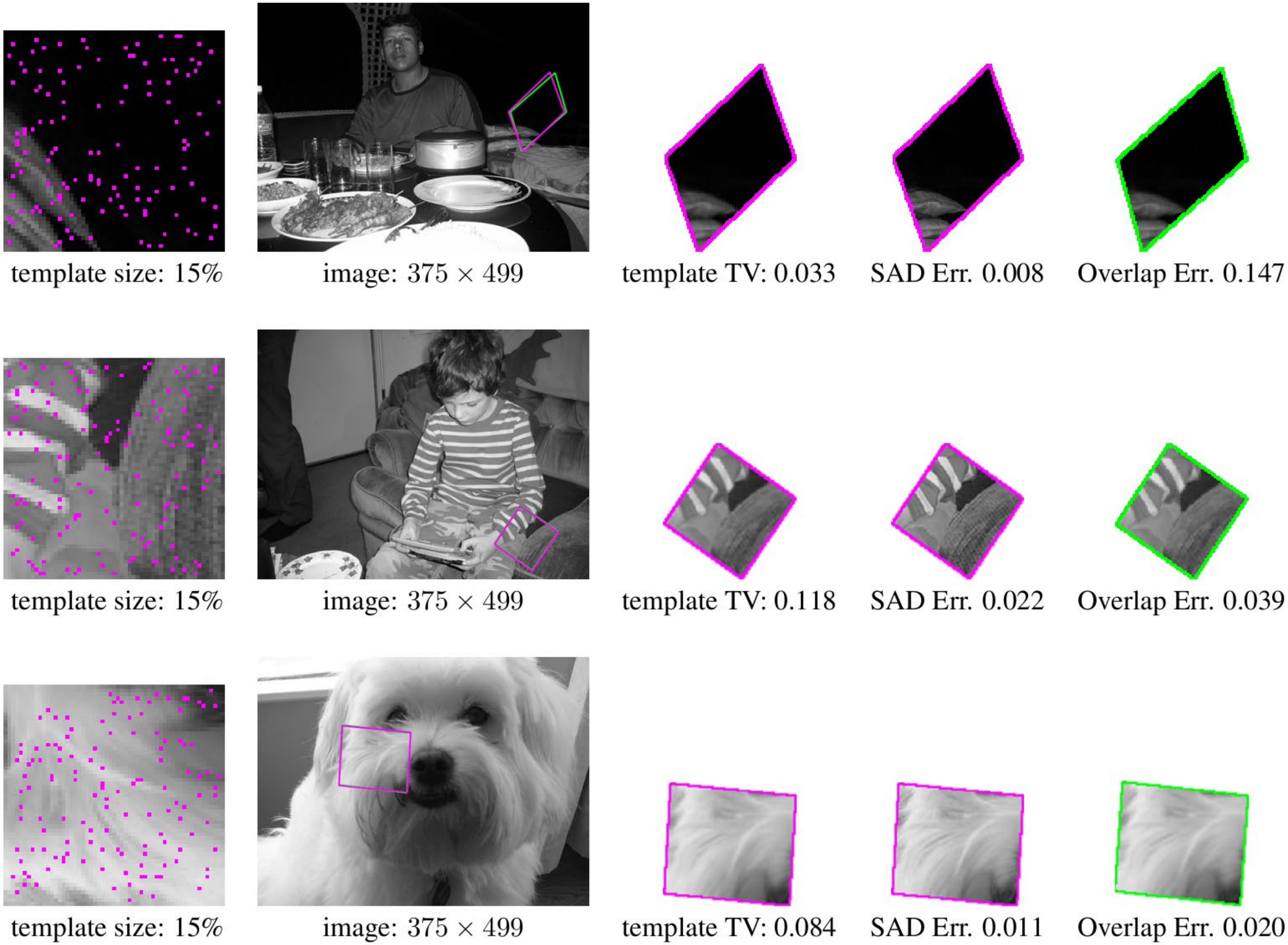
Template Dim: 35%



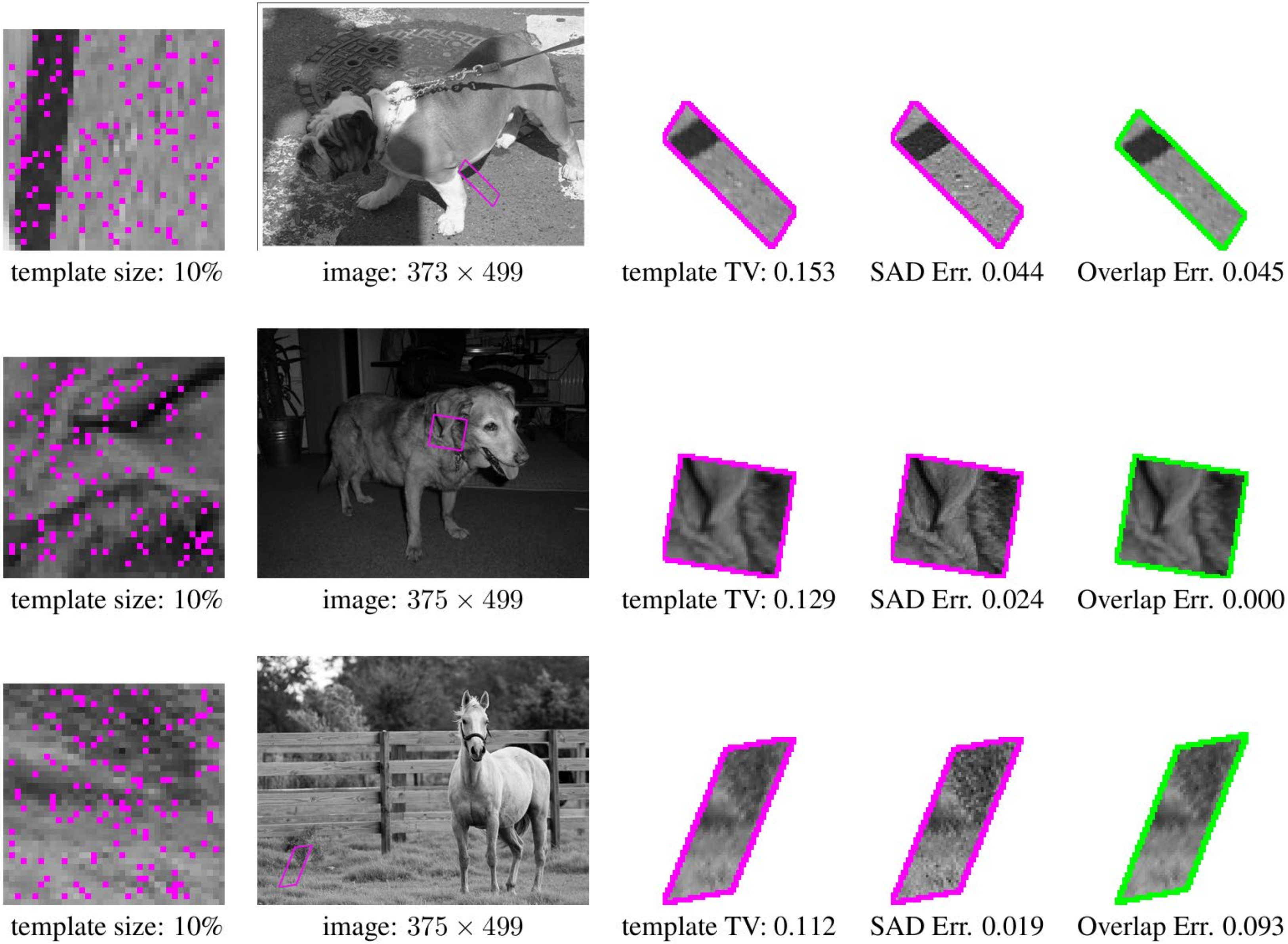
Template Dim: 25%



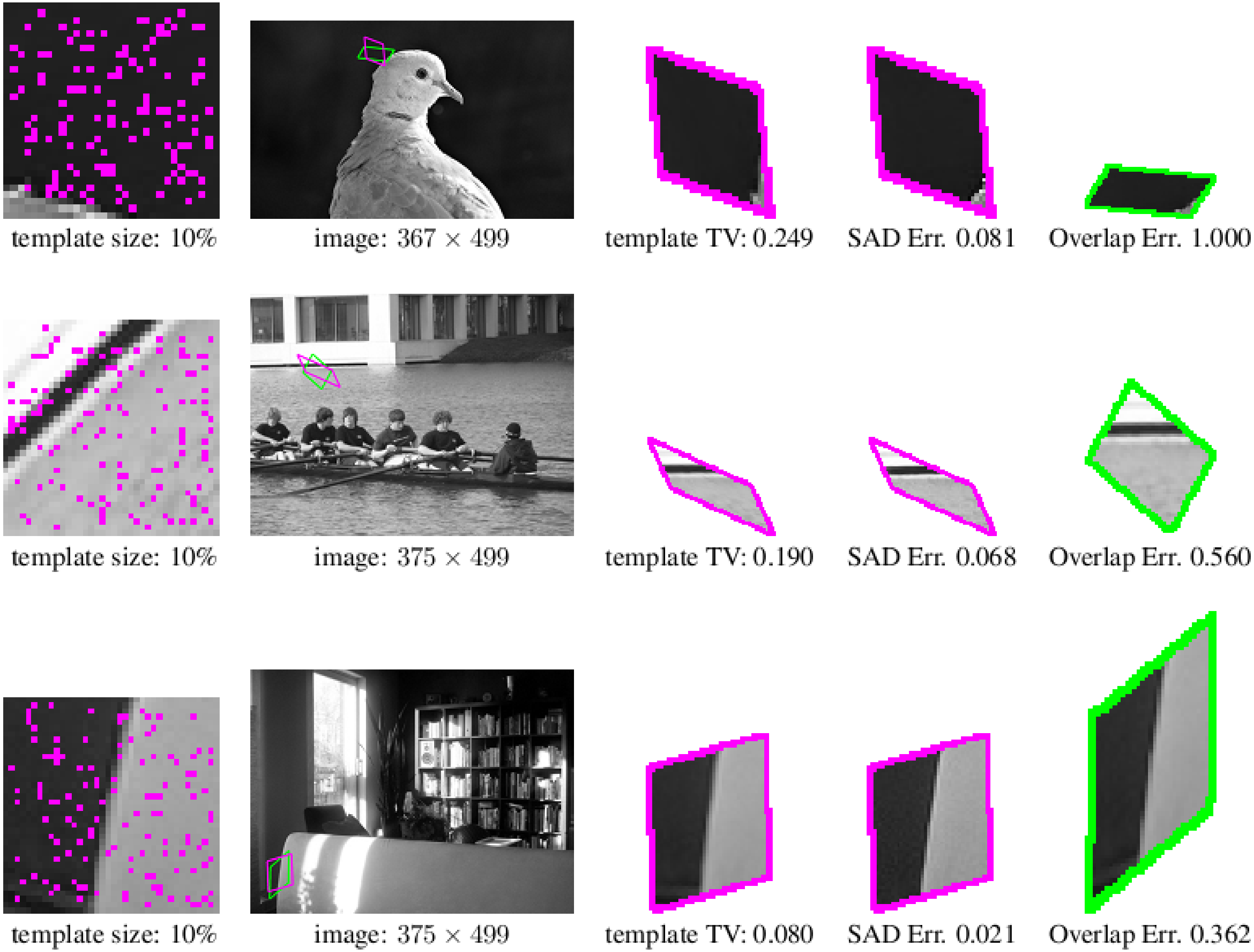
Template Dim: 15%



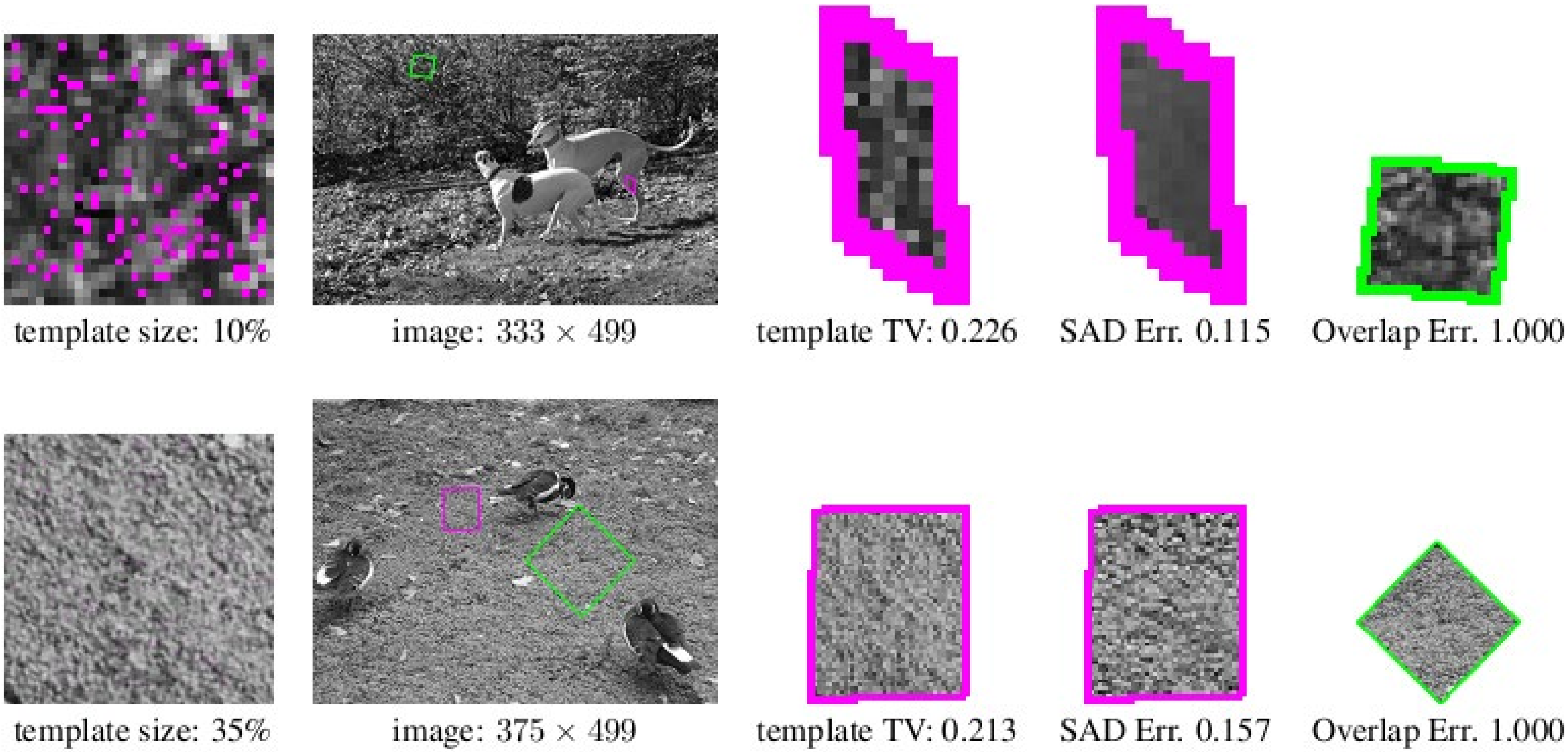
Template Dim: 10%



Bad overlap due to ambiguity



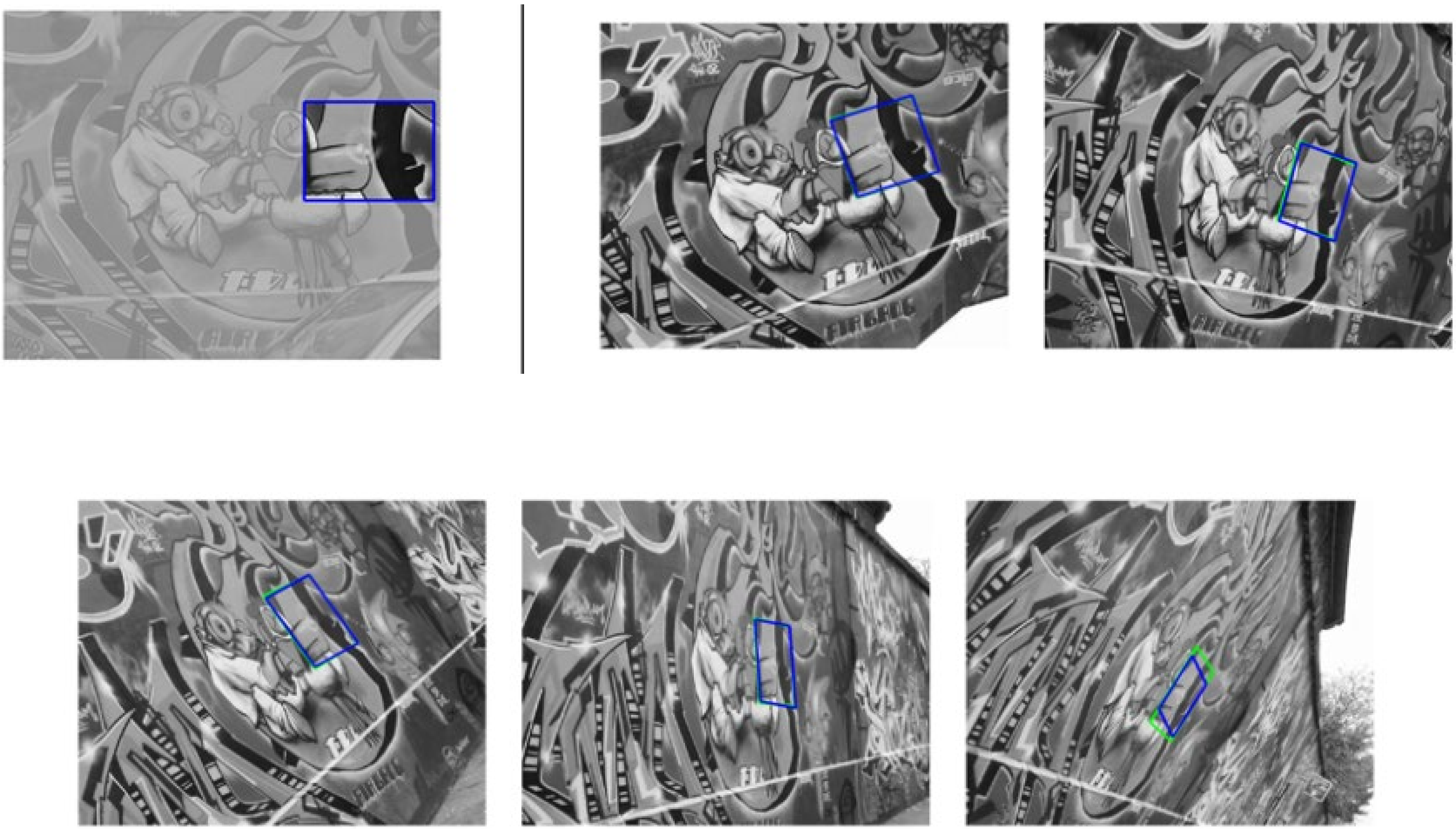
High SAD due to high TV and ambiguity



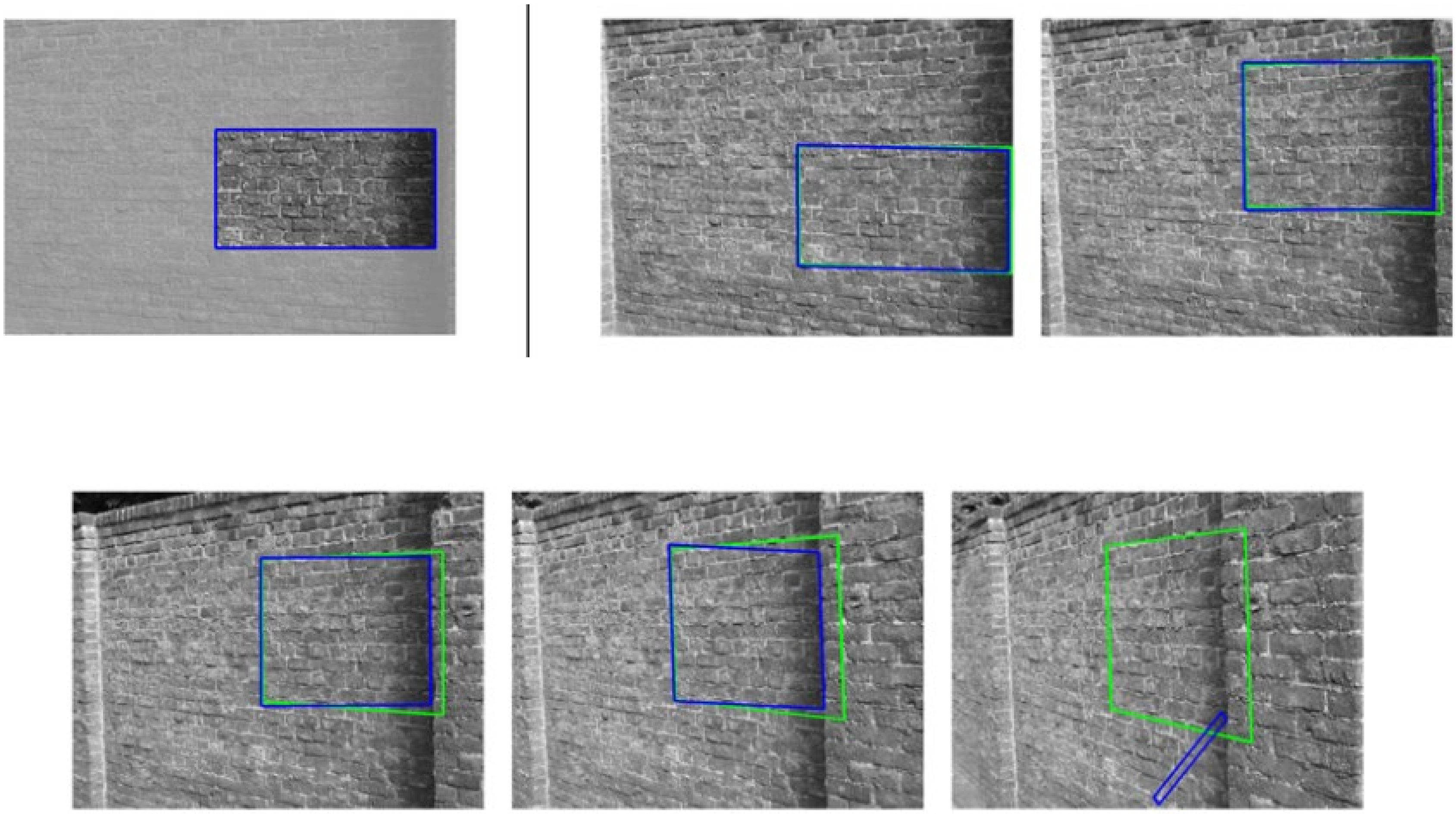
Experiment 2: Varying conditions

- Mikolajczyk data-set (for features and descriptors)
- 8 sequences of 6 images, with increasingly harsh conditions
- Including:
  - Zoom+Rotation (bark)
  - Blur (bikes)
  - Zoom+rotation (boat)
  - Viewpoint change (graffiti)
  - Brightness change (light)
  - Blur (trees)
  - Jpeg compression (UBC)
  - Viewpoint change (wall)

Mikolajczyk– graffiti (viewpoint)



Mikolajczyk– 'wall' (viewpoint)

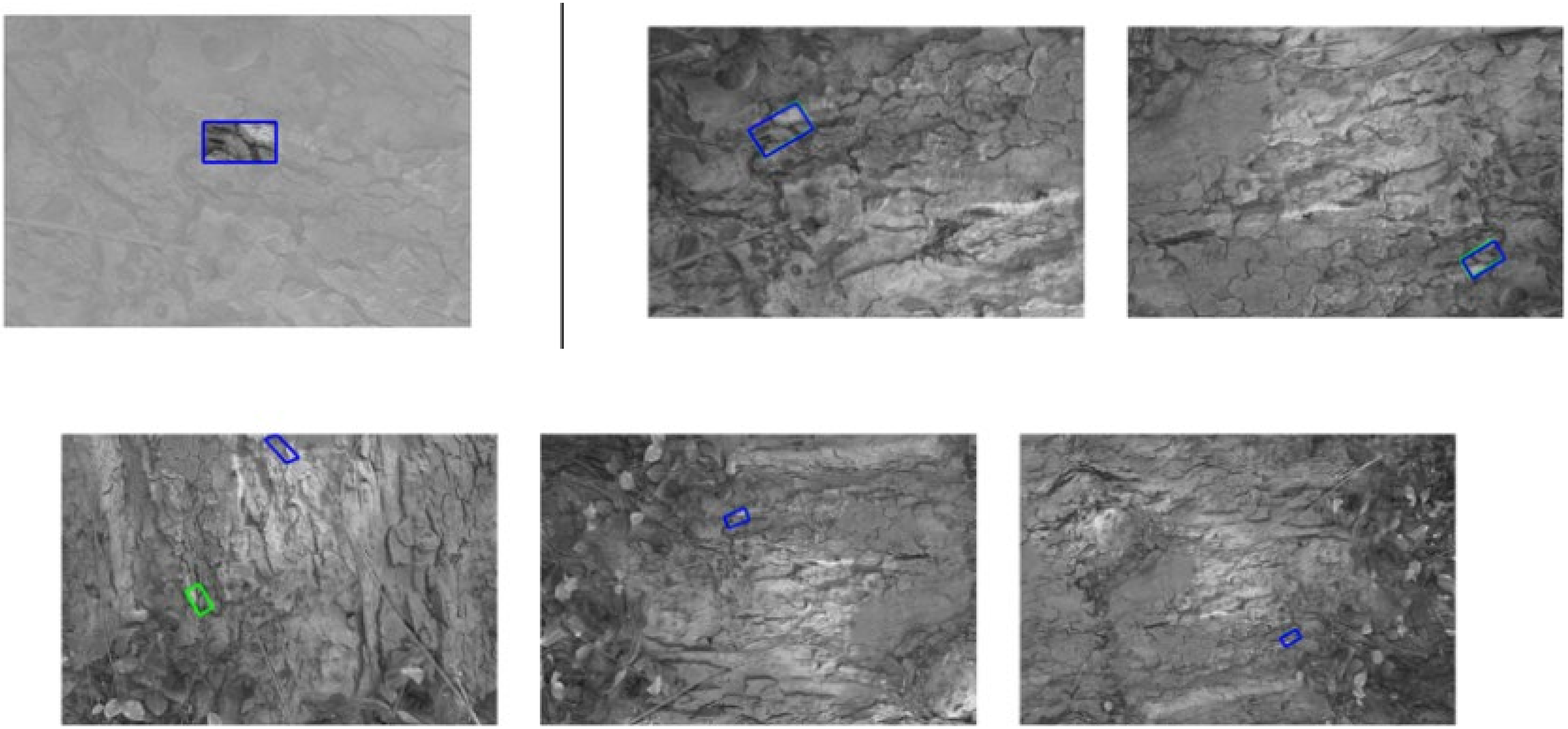


Mikolajczyk– ‘trees’ (blur)

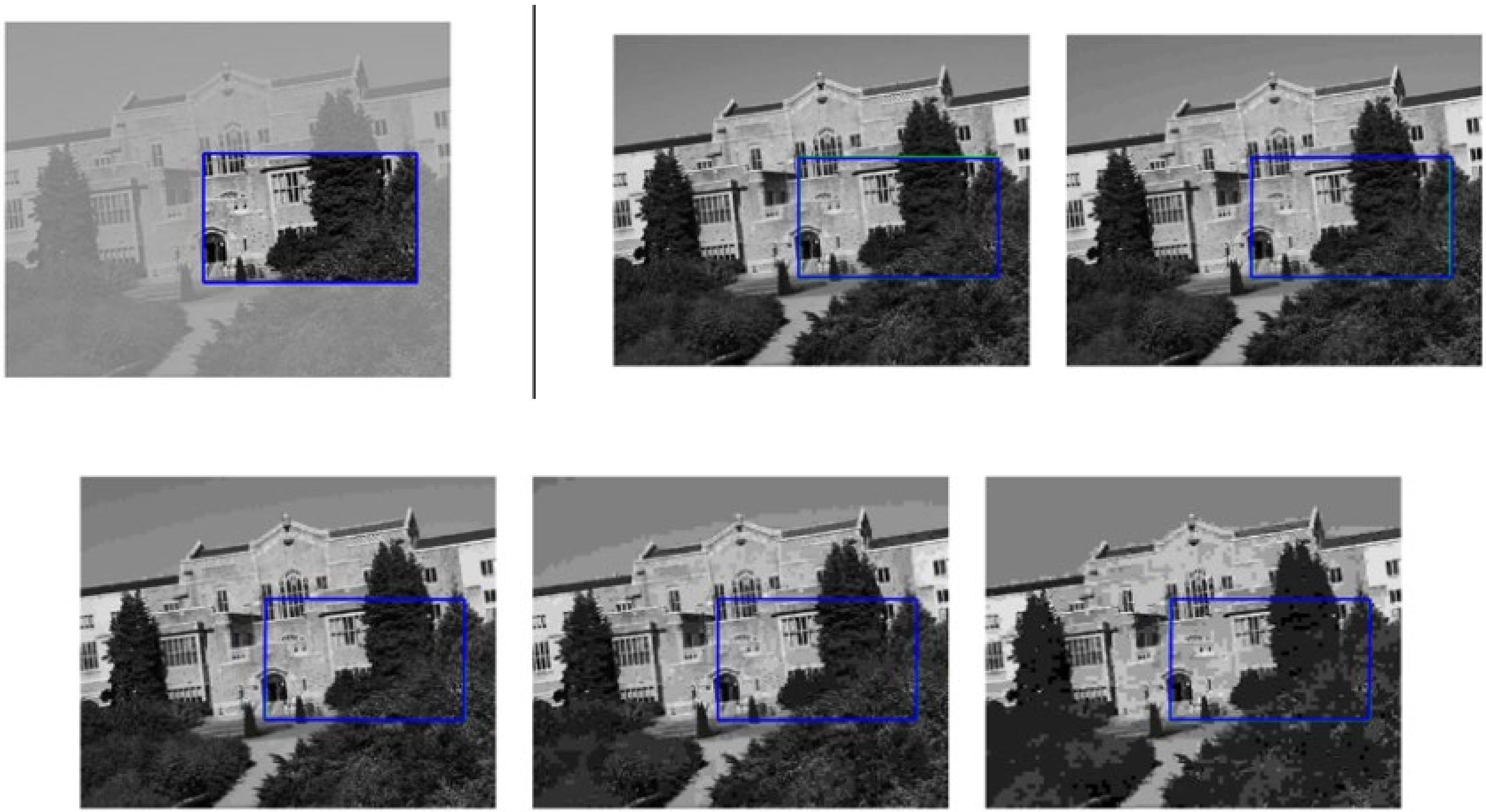





Mikolajczyk – ‘bark’ (zoom+rot)



Mikolajczyk – ‘UBC’ (JPEG)



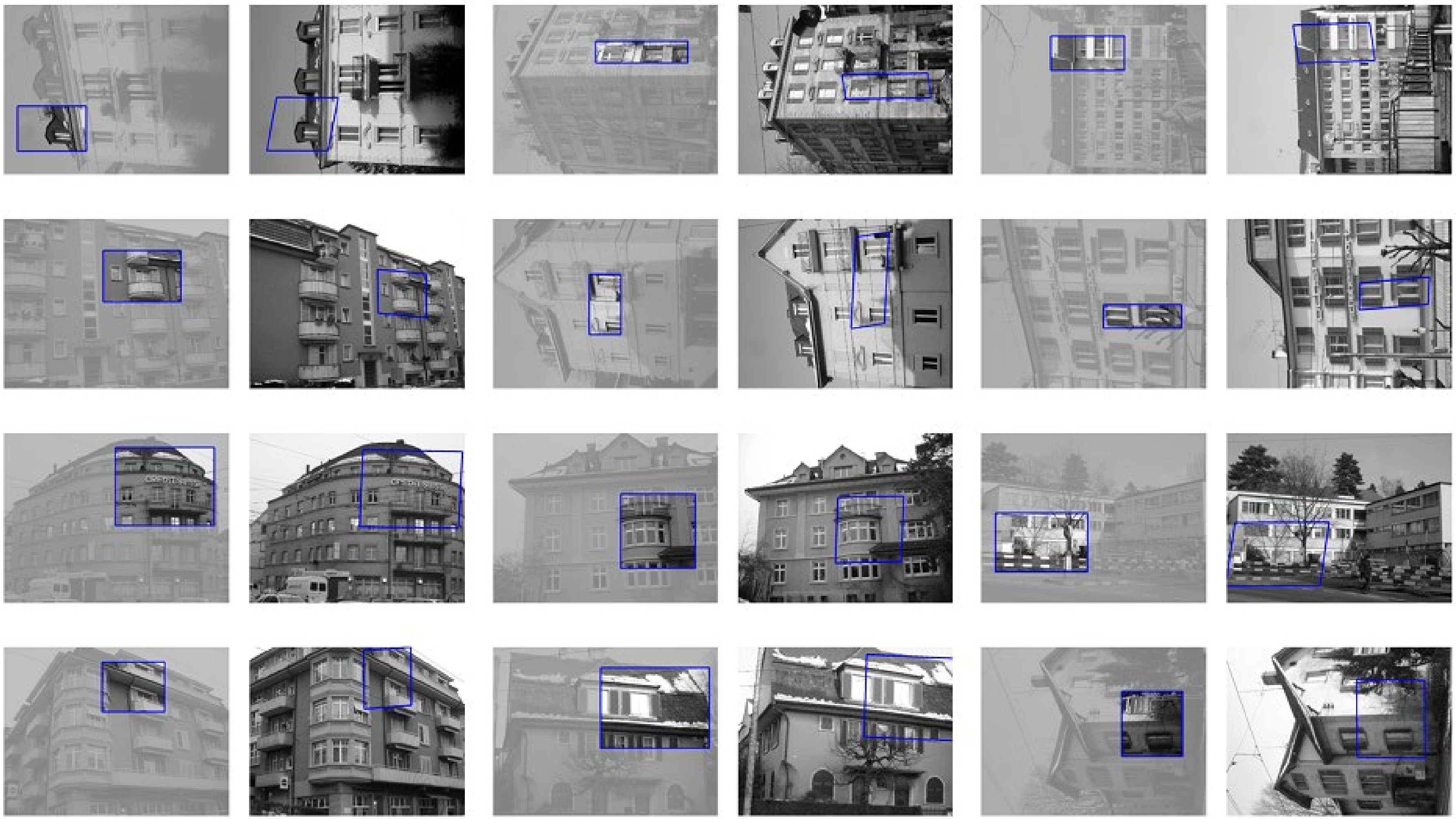
Experiment 3: Matching in real-world scenes

- The Zurich Building Data-set
  - 200 buildings, 5 different views each
  - 200 random instances
    - Random choice of building, 2 views, template in one view
- We seek the best possible affine transformation
- In most cases homography or non-rigid is needed

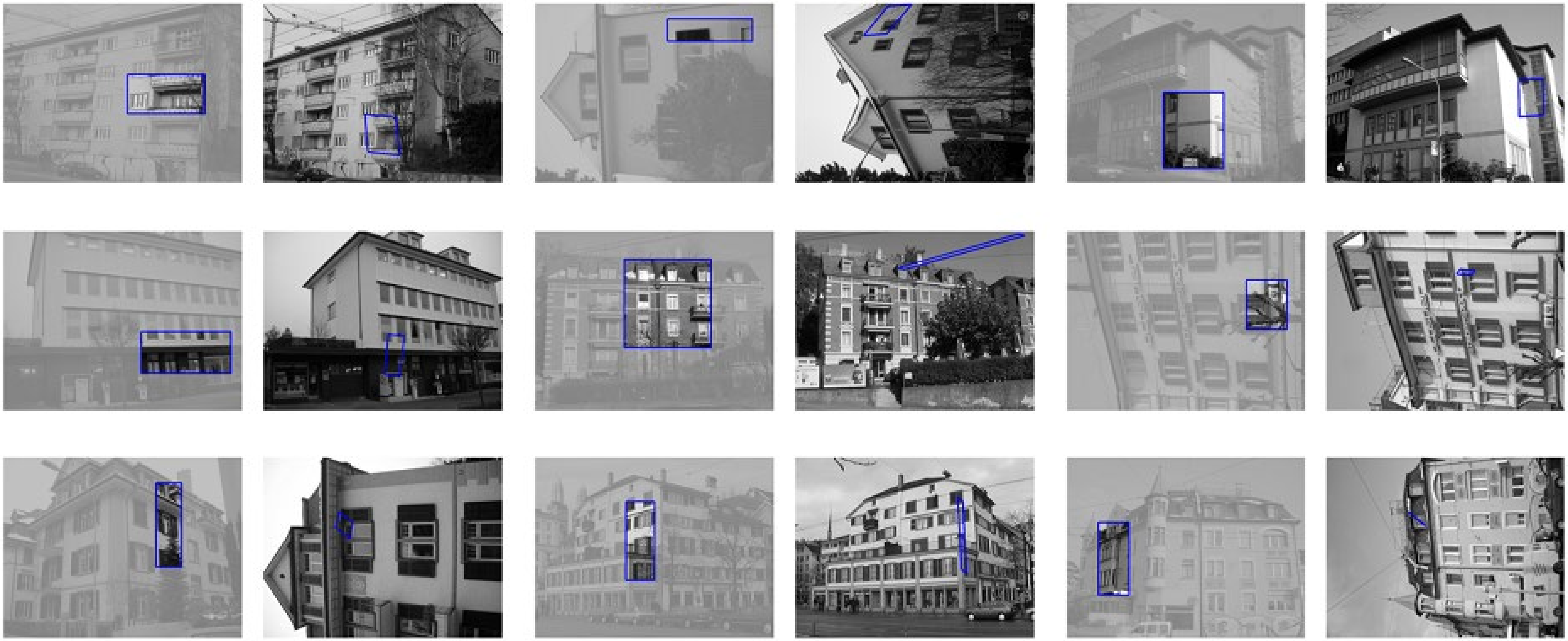
- **Results**:
  - 129 cases - ‘good’ matches
  - 40 cases – template doesn’t appear in second image
  - 12 cases – bad occlusion of template in second image
  - 19 cases – ‘failure’ (none of the above)

Experiment 3: Good cases

Experiment 3: Good cases



Experiment 3: failures, occlusions, out of img.



Fast-Match: Summary

- Handles template matching under arbitrary **Affine** (6 dof) transformations with
  - Guaranteed error bounds
  - Fast execution

- Main ingredients
  - Sampling of transformation space (based on variation)
  - Quick transformation evaluation (‘property testing’)
  - Branch-and-Bound scheme

Fast-Match: Summary

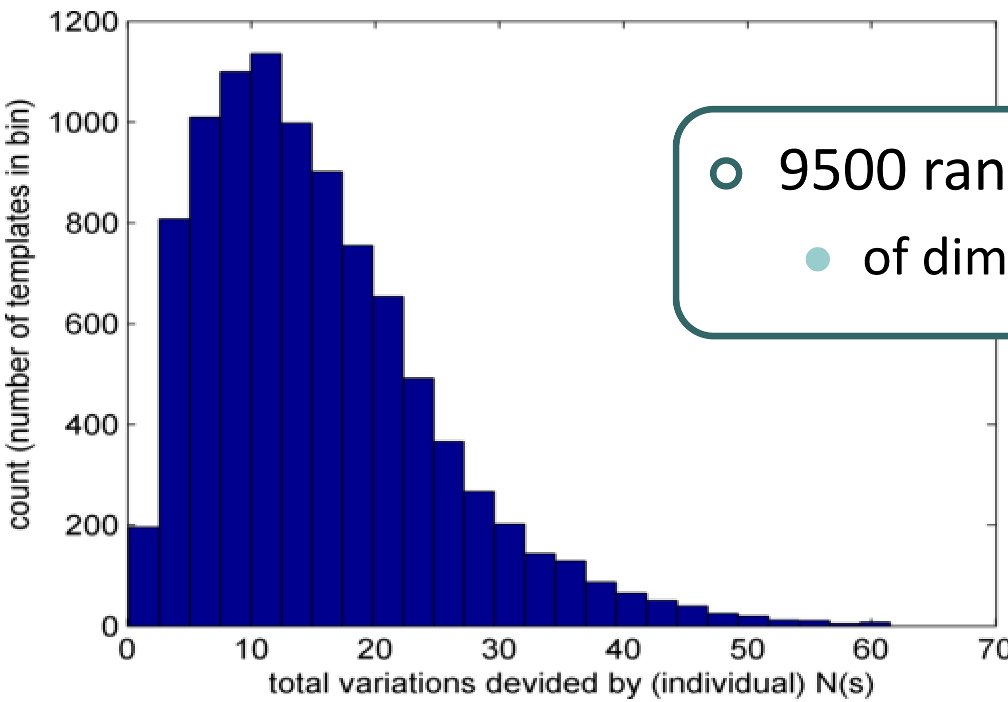
- Limitations
  - Smoothness assumption
  - Global transformation
  - Partial matching
- Extensions
  - Higher dimensions - Matching 3D shapes
  - Other registration problems
  - Symmetry detection

Thanks!

eng.tau.ac.il/~simonk

Image Smoothness

- Total-Variation: $TV(I) = \sum_{p \in I} \max_{q \in N(p)} \left| I(p) - I(q) \right|$
- For $n \times n$ images with intensities in [*0,1*]: $TV \in [0, n^2]$
- **Smooth** images: $TV = O(n)$



- 9500 random templates
  - of dim: $n = 200 \pm 100$

$$\frac{TV}{n} \approx O(1)$$

link back