Seminar on Sublinear Time Algorithms

FAsT-Match: Fast Affine Template Matching

Korman, S., Reichman, D., Tsur, G., & Avidan, S., 2013

Given by: Shira Faigenbaum-Golovin
Tel-Aviv University
27.12.2015

Tel Aviv University

Outline

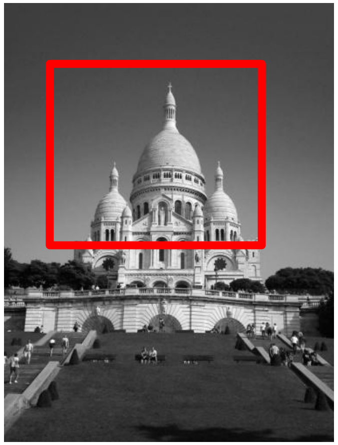
- Problem Definition
- Image processing in a nutshell
- Prior Art of image alignment
- Suggested algorithm

Problem Definition

Image matching:

Given two grayscale images, I_1 and I_2

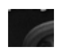
Find affine transformation T that maps pixels from I_1 to pixels to I_2.



So that the difference over pixels p between $I_1(T(p))$ and $I_2(p)$ is minimized

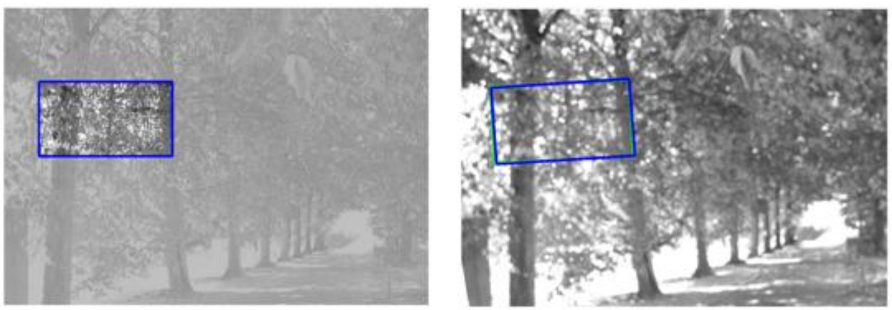
Generalized Template Matching

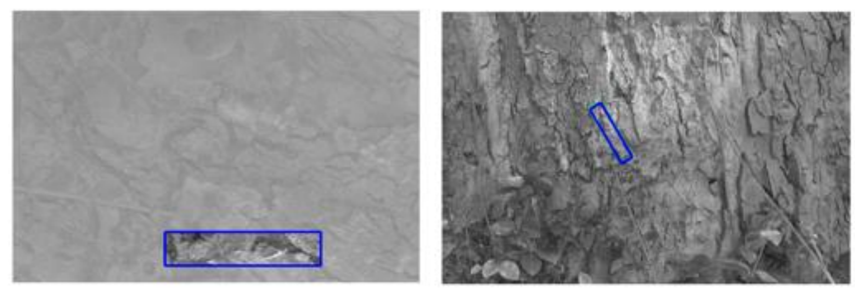
- Find the best transformation between two given images:

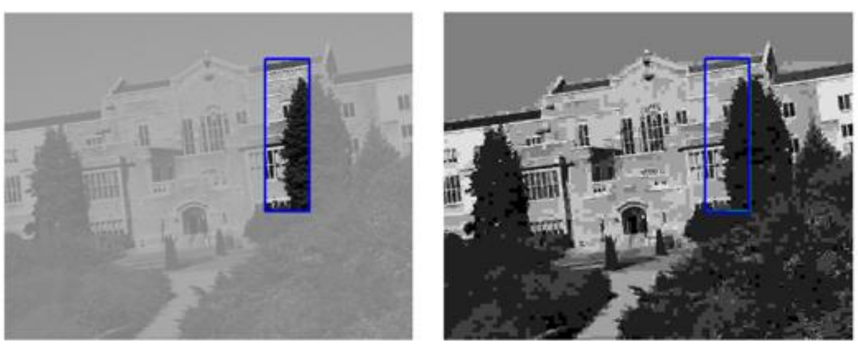


Some results I

Some results II







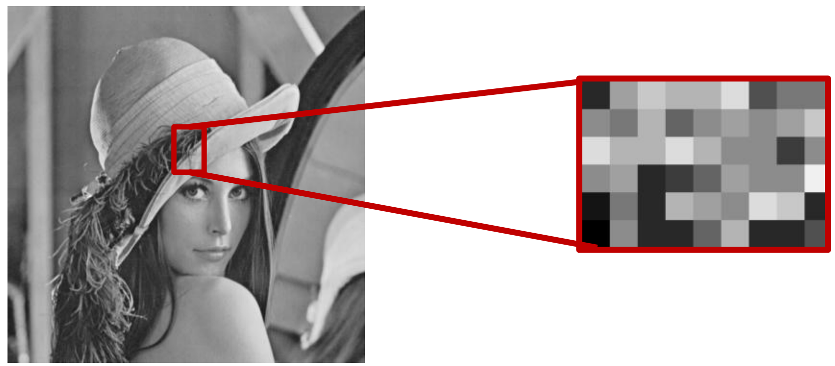
Some results III

Motivation

- Align two images before comparison
- Align for image enhancement
- Panoramic mosaics.
- Match images in a video sequence

Image Processing in a Nutshell

Gray scale image I is an nxm matrix with values between [0,1].

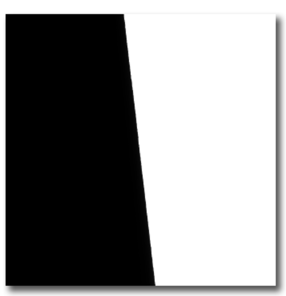


where 0 is black, and 1 is white.

The intermediate values are the gray levels

Image Processing in a Nutshell

- A pixel *p* in an *nxn* image *I* is a pair (x,y)in $\{1,\ldots,n\}^2$.
- A value of a pixel $p=(x,y)$ in an image *I* is $I(x,y)$.
- Two different pixels $p=(x,y)$ and $q=(x',y')$ are adjacent if $|x-x'|\leq 1$ and $|y-y'|\leq 1$.
- A pixel *p* is boundary in an image *I* if there is an adjacent pixel *q* s.t $I(p)\neq I(q)$.



$P_{i-1,j-1}$	$P_{i-1,j}$	$P_{i-1,j+1}$
$P_{i,j-1}$	P_{ij}	$P_{i,j+1}$
$P_{i+1,j-1}$	$P_{i+1,j}$	$P_{i+1,j-1}$

Image Transformations in 2D

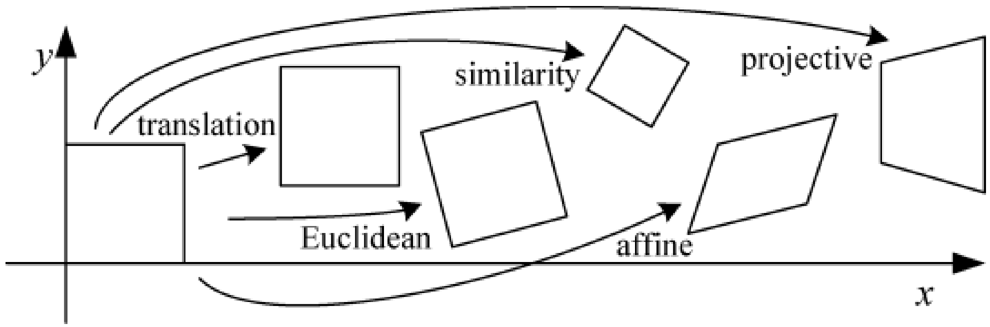
- An affine transformation matrix T can be decomposed into

$$T(I_1) = Tr \cdot R_2 \cdot S \cdot R_1 \cdot I_1$$

where Tr, R, S are translation, rotation and non-uniform scaling matrices.

$$\boldsymbol{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

- There exist 6 degrees of freedom:
- a rotation angle, x and y scales, another rotation angle and x and y translations.



Distance between Images

Given two grayscale images, I_1 and I_2

and affine transformation $T: I_1 \rightarrow I_2$.

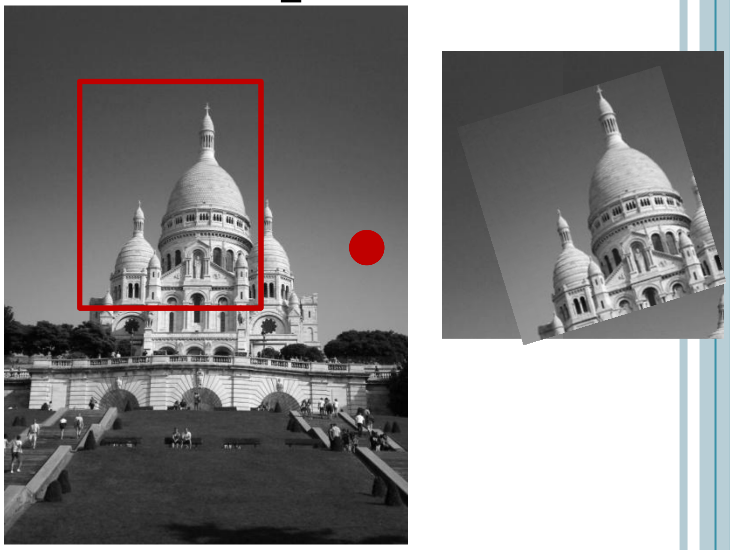
We define a sum of absolute differences (**SAD**)

and

$$d_T(I_1,I_2)=\frac{1}{n^2}\left[\sum_{p\in I_1|T(p)\in I_2}\left|I_1(T(p))-I_2(p)\right|\right]$$

The optimal transformation satisfies:

$$d(I_1,I_2)=\min_T d_T(I_1,I_2)$$



Generalized Template Matching

The algorithm:

1. Take a sample of the Affine transformations
2. Evaluate the SAD for each transformation in the sample
3. Return the best

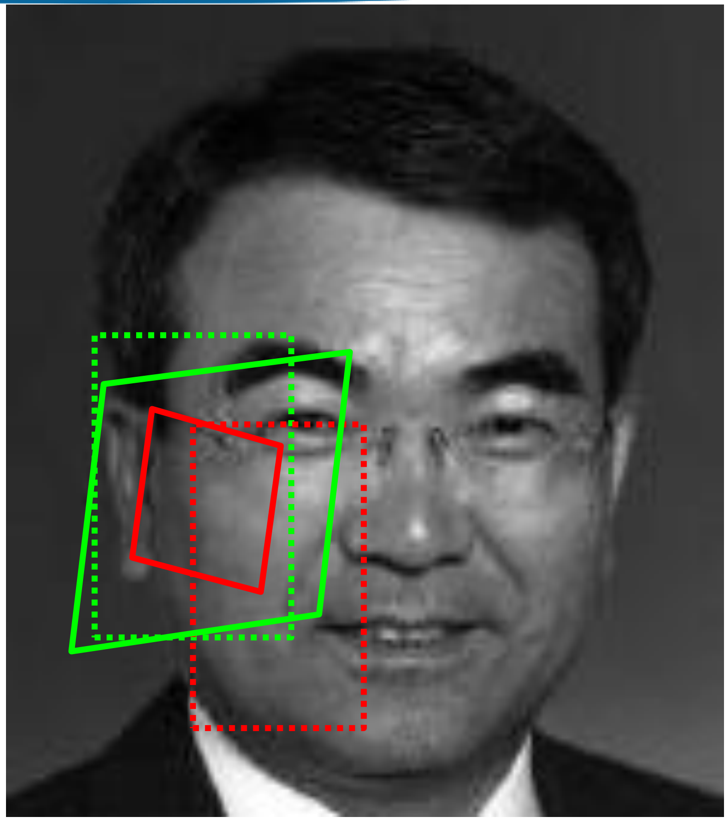
Questions:

- Which transformations to use?
- How does is guarantee a bound?

Prior art

- Direct methods – parametric OF
- Indirect methods (feature based)

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]

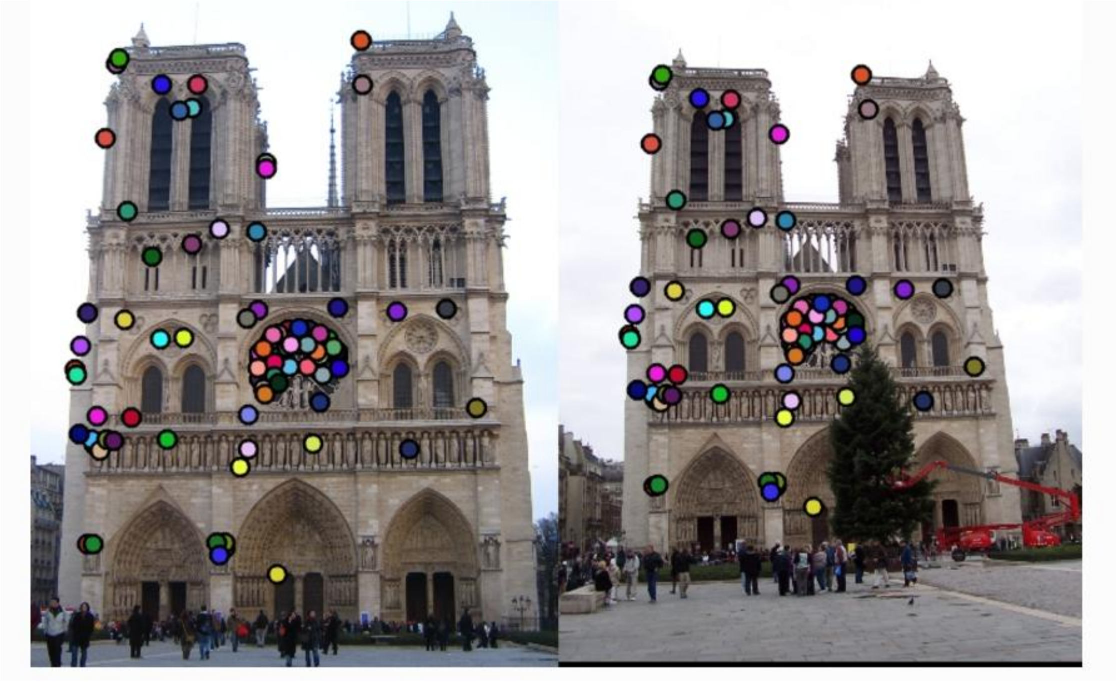
DIRECT METHODS – PARAMETRIC OF

- ∞ transformations – need to discretize
- **"Combinatorial bounds and algorithmic aspects of image matching under projective transformations"** [Hundt & Liskiewicz MFCS, 2008]

Enumerate ~ n^{18} affine transformation (for *nxn* images)

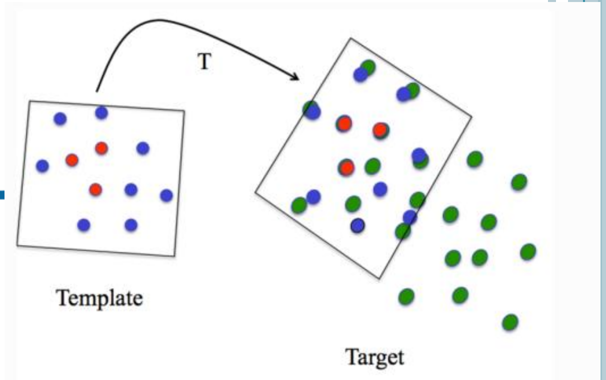
INDIRECT METHODS (FEATURE BASED)

e.g. SIFT



Computational complexity $\Theta(n^2)$

RANSAC

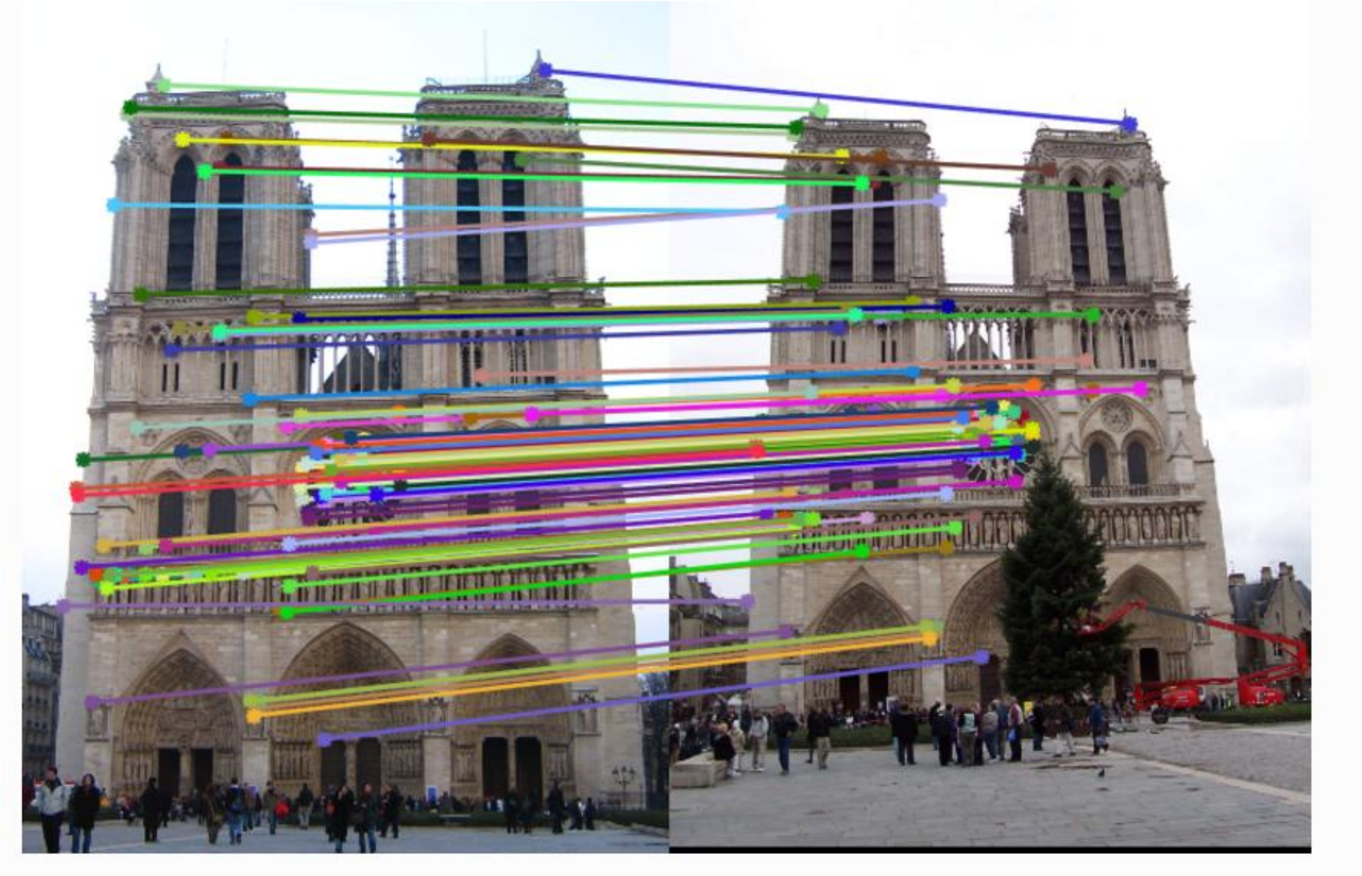


- Select (at random) a subset of *k* pairs
- Compute a motion estimate T
- By using least squares, to minimize the sum of squared residuals.
- Counts the number of inliers that are within ε of their predicted location
- The random selection process is repeated *m* times,
- The sample set with largest number of inliers is kept as the final solution

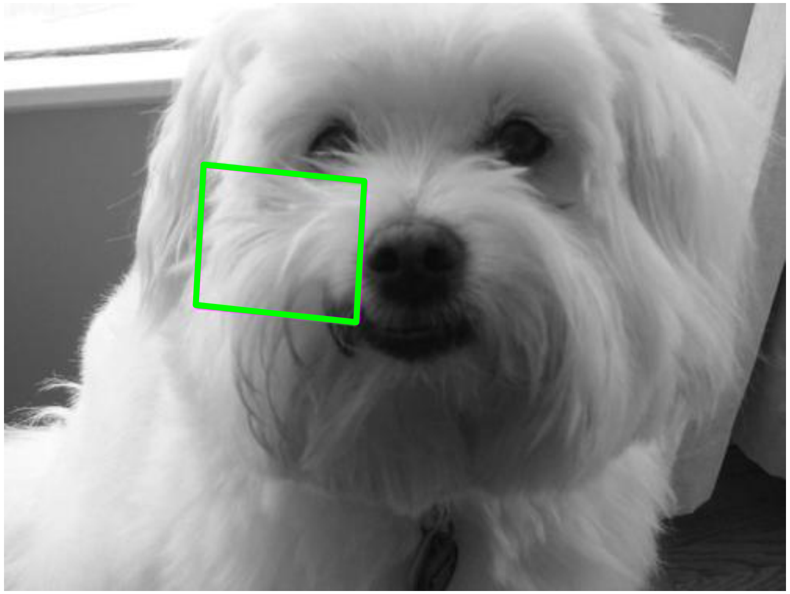
Computational complexity $\Theta(k(t+mN))$

k=#samples
N=#data points
t= time of single model
m=avg # of models per sample

INDIRECT METHODS (FEATURE BASED)



INDIRECT METHODS (FEATURE BASED)

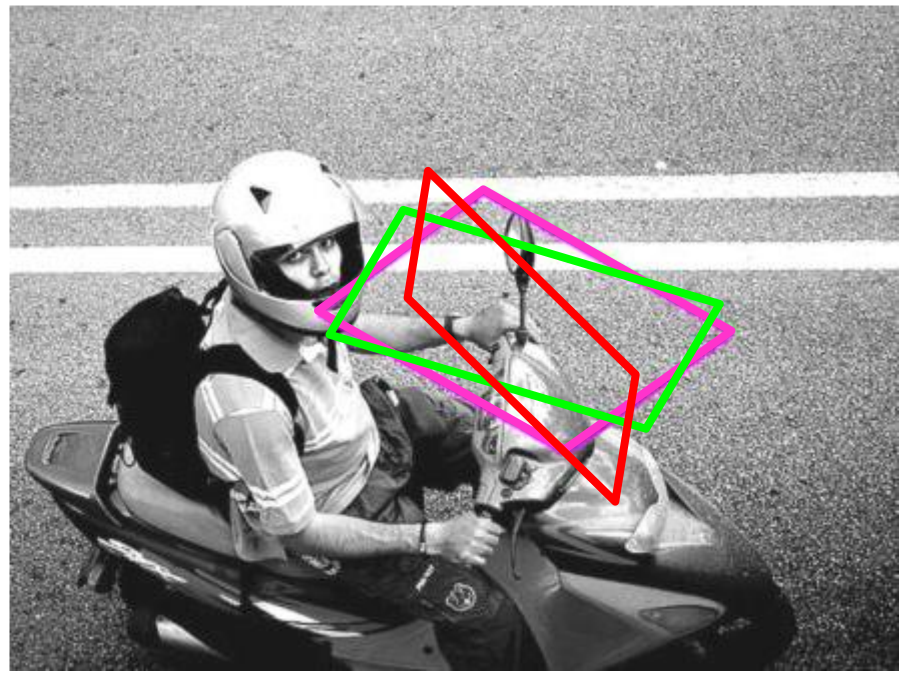


Lowe "**Distinctive image features from scale-invariant key-points**" [IJCV 04]
Morel, Yu "**Asift: A new framework for fully affine invariant image comparison**" [SIAM 09]
M.A. Fichler, R.C. Bolles "**Random sample consensus**" [Comm. of ACM 81]

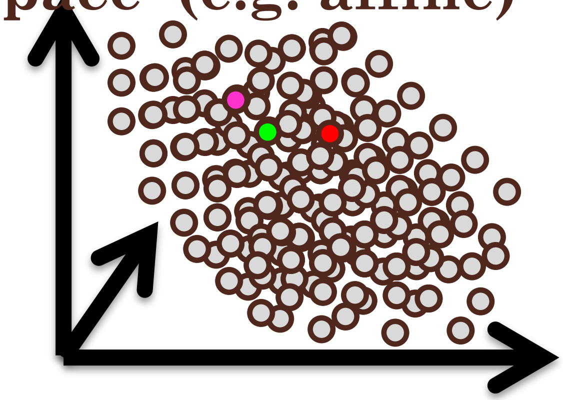
The Main Idea

template

image



Transformation space (e.g. affine)

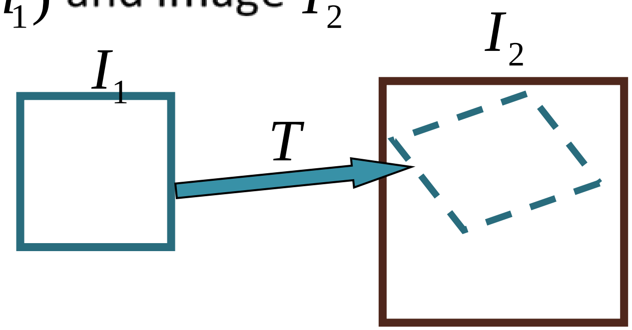


<u>Observation:</u>

Due to <u>image smoothness assumption</u>, the SAD measure will not change significantly, when small variations in the parameters of the transformation

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:

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

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

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

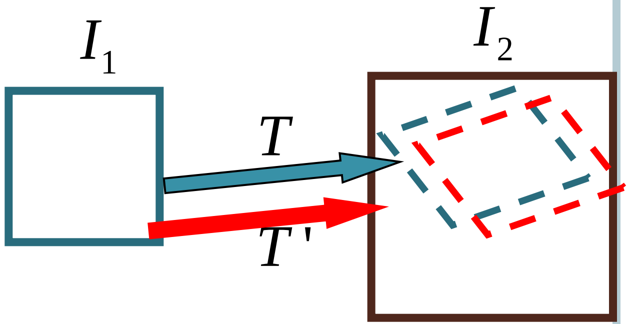
- **Goal**: Given $\delta > 0$, find a transformation T^* in Ψ for which:

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

NET OF AFFINE TRANSFORMATIONS

Given two transformations *T* and *T'* , we define $l_\infty(T,T')$ as:

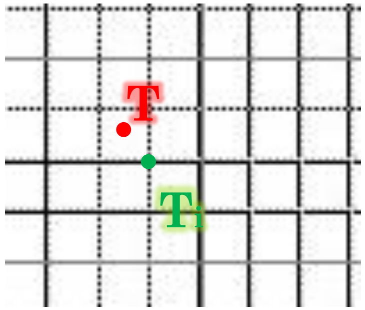
$$l_\infty(T,T') = \max_{p \in I_1} \|T(p) - T'(p)\|_2$$



For a positive *α*, <u>a net of (affine) transformations</u> $\tau = \{T_i\}$ is an *α*-cover if

$$\forall \quad T \notin \tau$$

$$\exists T_i \in \tau \quad l_\infty(T,T_i) = O(\alpha)$$



The Algorithm – take 2

- Create a net $\mathcal{N}_{\delta/2}$ that is a $(\delta n_1)/2$-cover of the set of affine transformations
- For each $T \in \mathcal{N}_{\delta/2}$ approximate $\Delta_T(I_1, I_2)$ to within precision of $\delta/2$. Denote the resulting value d_T
- Return the transformation T with the minimal value d_T

Create a Net

- An affine transformation matrix T can be decomposed into

$$T(I_1) = Tr \bullet R_2 \bullet S \bullet R_1 \bullet I_1$$

- There exist 6 degrees of freedom: a rotation angle, x and y scales, another rotation angle and x and y translations.

The basic idea is to discretize the space of Affine transformations, by dividin each of the dimensions into $\Theta(\delta)$ equal segments, such that for any two consecutive transformations T and T' on any of the dimensions it will hold that

$$l_\infty(T, T') < \Theta(\delta n_1)$$

Create a Net

transformation	step size	range	num. steps
x translation	$\Theta(\delta n_1)$ pixels	$[-n_2, n_2]$	$\Theta(\frac{n_2}{n_1}/\delta)$
y translation	$\Theta(\delta n_1)$ pixels	$[-n_2, n_2]$	$\Theta(\frac{n_2}{n_1}/\delta)$
1st rotation	$\Theta(\delta)$ radians	$[0, 2\pi]$	$\Theta(1/\delta)$
2nd rotation	$\Theta(\delta)$ radians	$[0, 2\pi]$	$\Theta(1/\delta)$
x scale	$\Theta(\delta)$ pixels	$[1/c, c]$	$\Theta(1/\delta)$
y scale	$\Theta(\delta)$ pixels	$[1/c, c]$	$\Theta(1/\delta)$

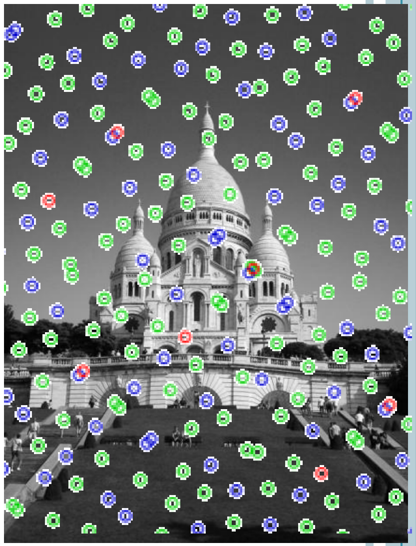
$$\Theta\left(\frac{1}{\delta^6} \cdot \left(\frac{n_2}{n_1}\right)^2\right)$$

APPROXIMATE $\Delta_T(I_1, I_2)$

Input: Grayscale images I_1 and I_2, a precision parameter δ and a transformation T

Output: An estimate of the distance $\Delta_T(I_1, I_2)$

- Sample m = $\Theta(1/\delta^2)$ values of pixels $p_1 \dots p_m$ in I_1
- Return $d_T = \sum_{i=1}^{m} |I_1(p_i) - I_2(T(p_i))|/m.$



Claim: Given images I_1 and I_2 and an affine transformation T, the algorithm returns a value d_T such that $|d_T - \Delta_T(I_1, I_2)| < \delta$ with probability 2/3. It performs $(1/\delta^2)$ samples.

Estimate the SAD to within $O(1/\delta^2)$

The Algorithm

- Create a net $\mathcal{N}_{\delta/2}$ that is a $(\delta n_1)/2$-cover of the set of affine transformations
- For each $T \in \mathcal{N}_{\delta/2}$ approximate $\Delta_T(I_1, I_2)$ to within precision of $\delta/2$. Denote the resulting value d_T
- Return the transformation T with the minimal value d_T

$$\Theta\left(\frac{1}{\delta^6} \cdot \left(\frac{n_2}{n_1}\right)^2\right)$$

$O(1/\delta^2)$

Total runtime is: $|A_\delta| \cdot \Theta(1/\delta^2) = \Theta\left(\frac{1}{\delta^8} \cdot \left(\frac{n_2}{n_1}\right)^2\right)$

PROBLEM

Achieving a satisfactory error rate would require using a net N_δ where δ is small. Thus causing the execution time to grow.

Therefore branch-and-bound scheme is used, by running the algorithm on subset of the net N_δ. and refining the δ parameter.

Branch-and-Bound

Input: Grayscale images I_1, I_2, a precision parameter δ^*
Output: A transformation T.

1. Let S^0 be the complete set of transformations in the net $\mathcal{N}_{\delta_0}$ (for initial precision δ_0)
2. Let $i = 0$ and repeat while $\delta_i > \delta^*$
   (a) Run algorithm 1 with precision δ_i, but considering only the subset S^i of $\mathcal{N}_{\delta_i}$
   (b) Let T_i^{Best} be the best transformation found in S^i
   (c) Let $Q^i = \{q \in S^i : \Delta_q(I_1, I_2) - \Delta_{T_i^{Best}}(I_1, I_2) < L(\delta_i)\}$
   (d) Improve precision: $\delta_{i+1} = fact \cdot \delta_i$ (by some constant factor $0 < fact < 1$)
   (e) Let $S^{i+1} = \{T \in Net_{\delta_{i+1}} : \exists q \in Q_i \ s.t. \ \ell_\infty(T, q) < \delta_{i+1} \cdot n_1\}$
3. Return the transformation T_i^{Best}

Results

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}

Image degradations

Example of Lossy Compression

Original Lena Image (12KB size)

Lena Image, Compressed (85% less information, 1.8KB)

Lena Image, Highly Compressed (96% less information, 0.56KB)

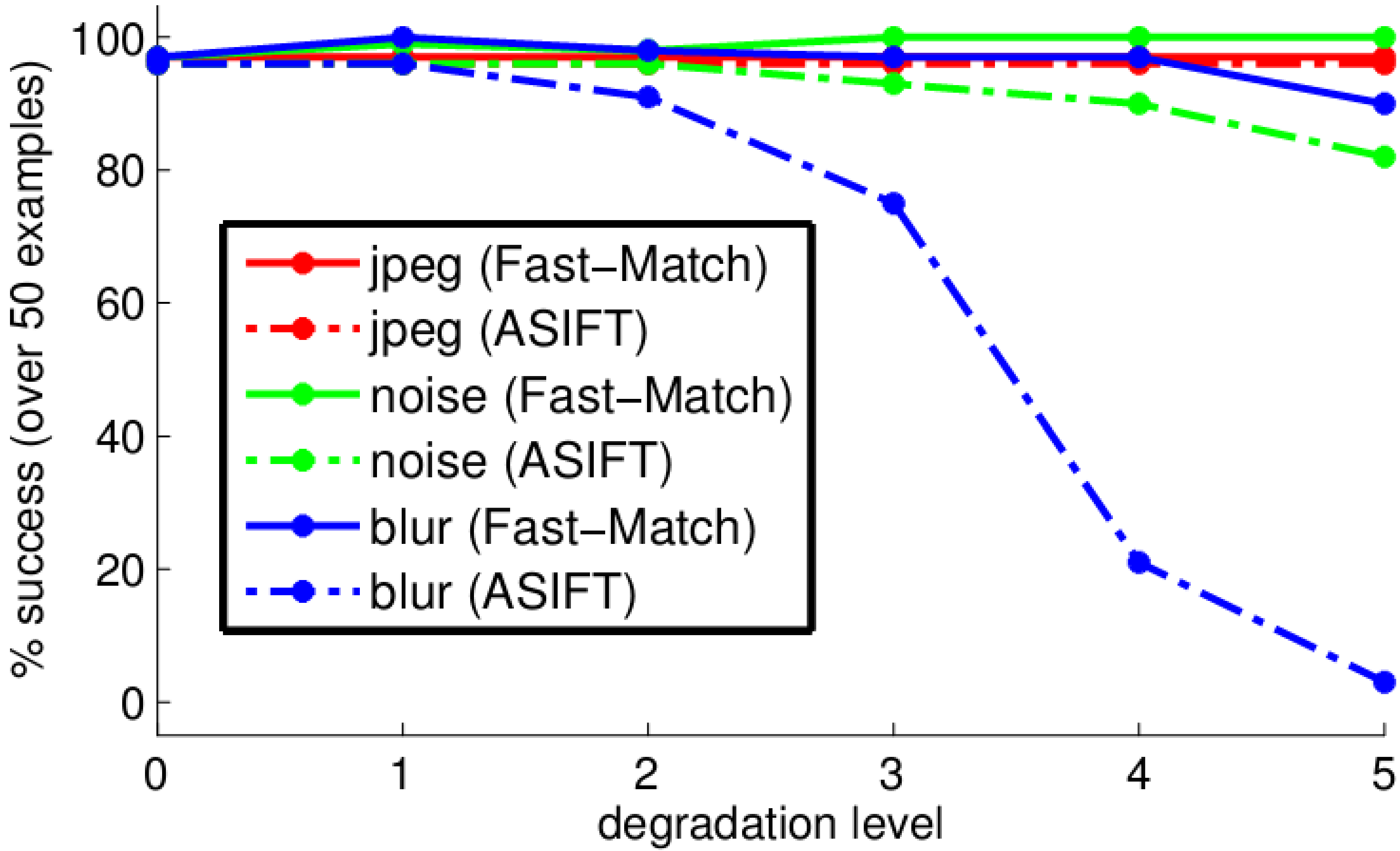
Experiment 1: Affine Template Matching

- 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%

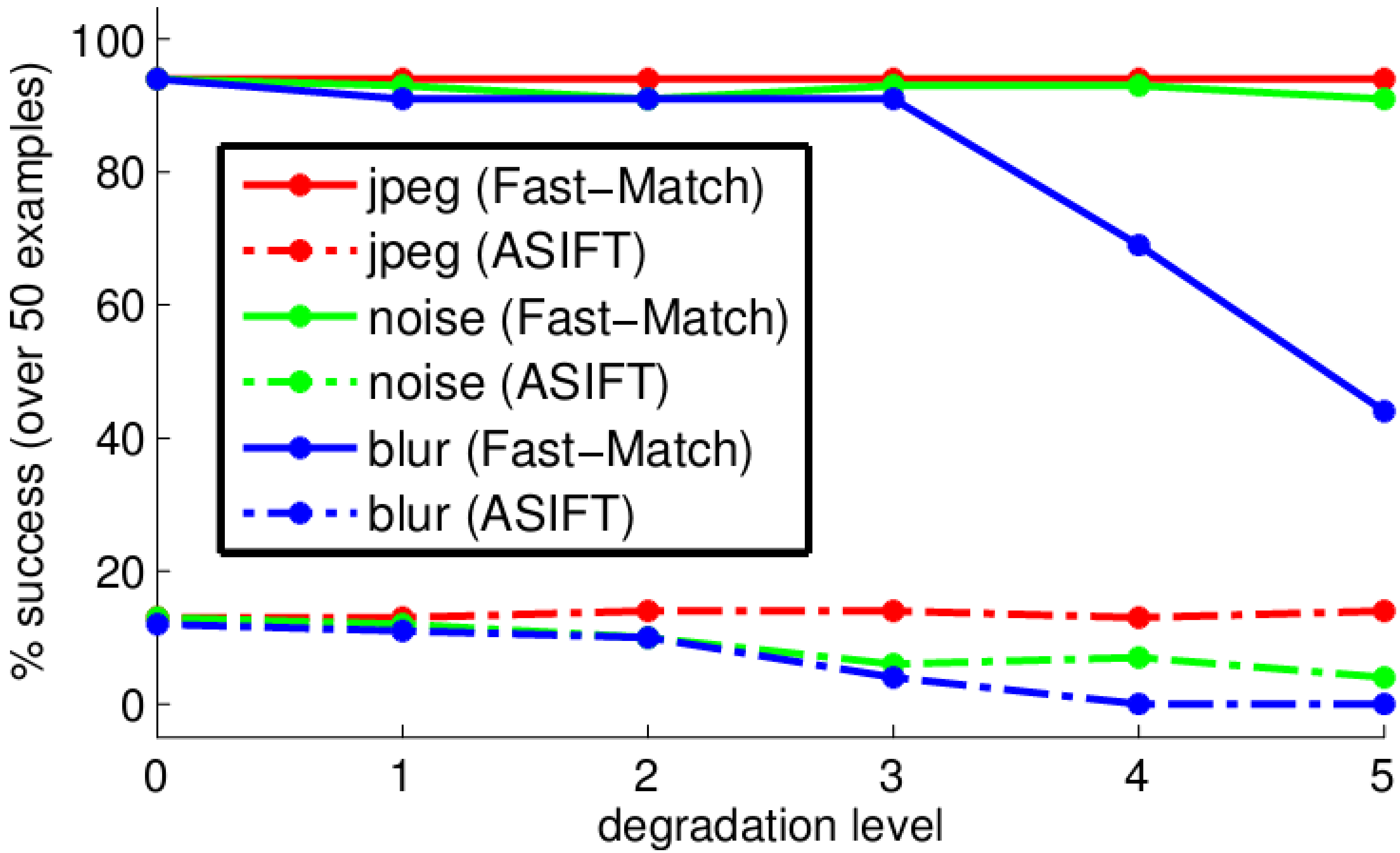
Experiment 1: Affine Template Matching

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



Experiment 1: Affine Template Matching

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

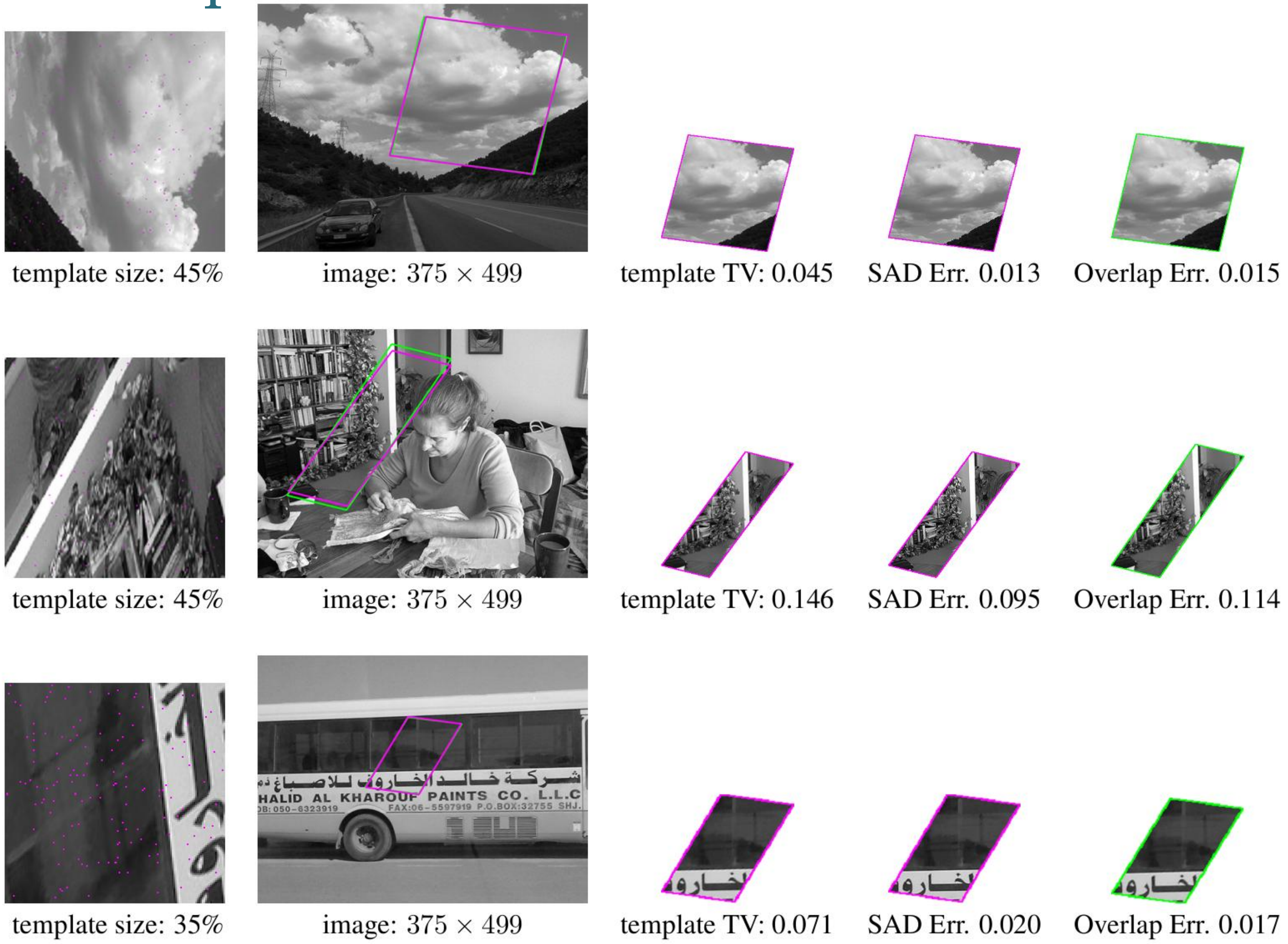


Experiment 1: Affine Template Matching

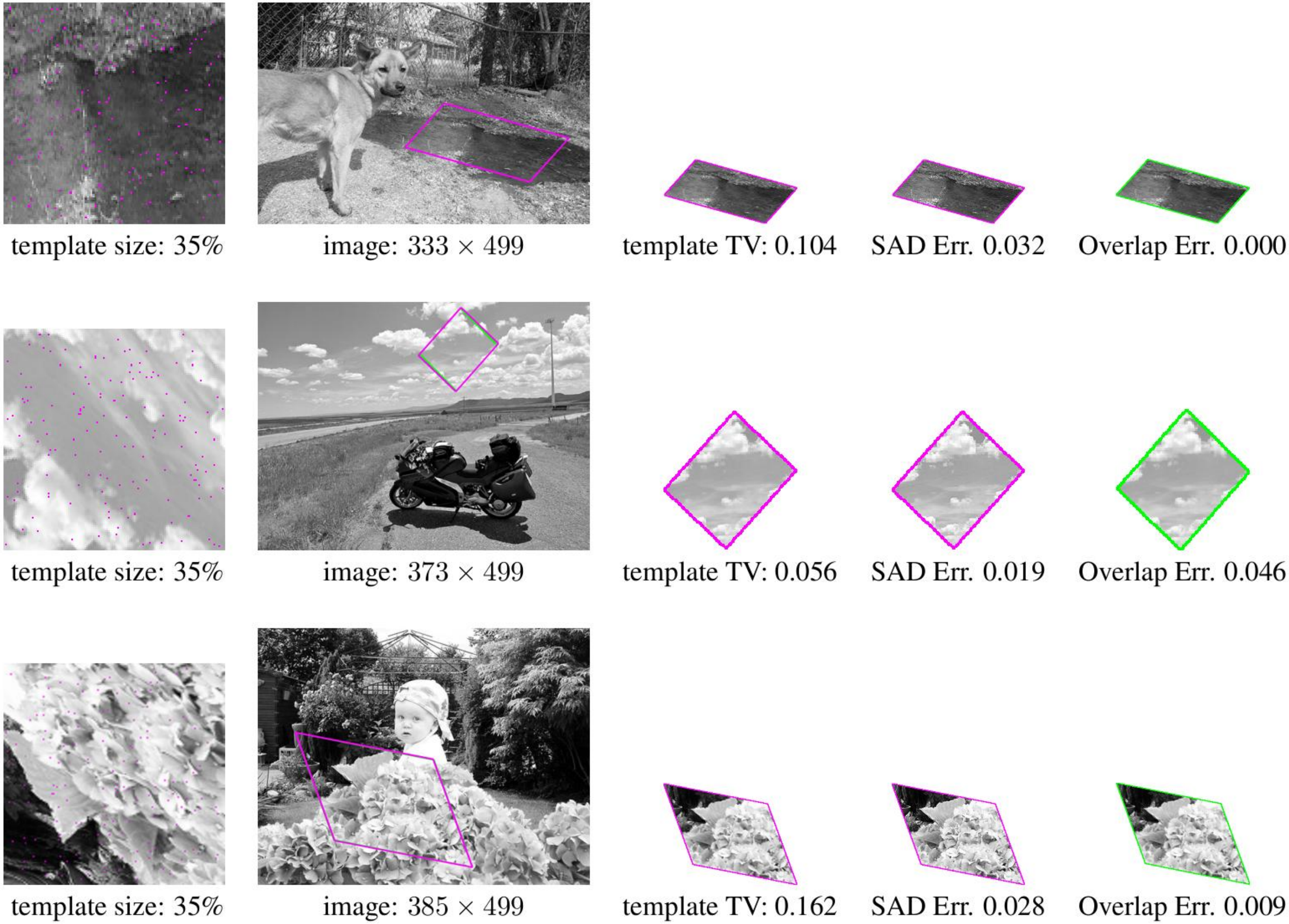
- Runtimes

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%



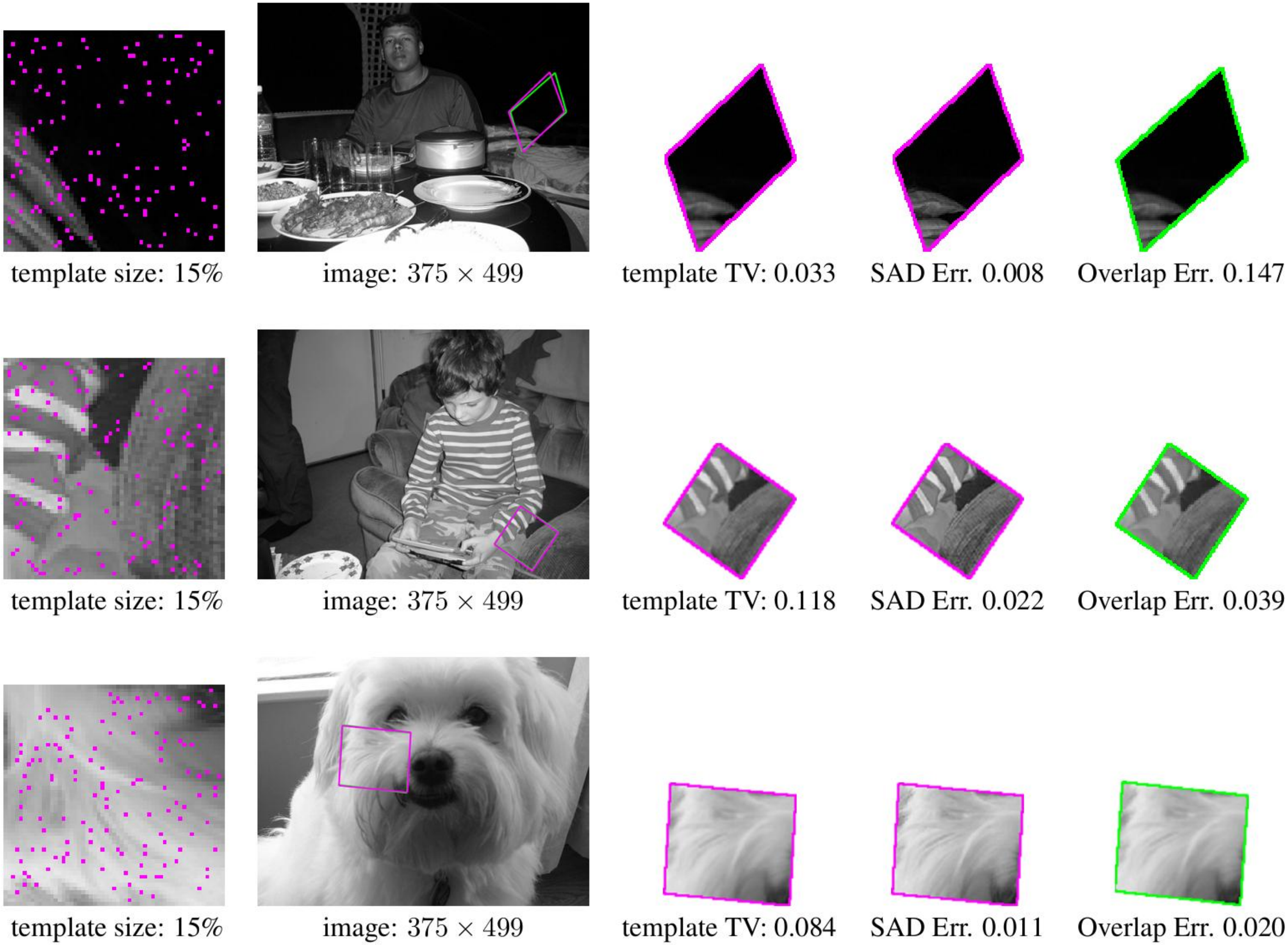
Template Dim: 25%

template size: 25% | image: 375 × 499 | template TV: 0.113 | SAD Err. 0.030 | Overlap Err. 0.000

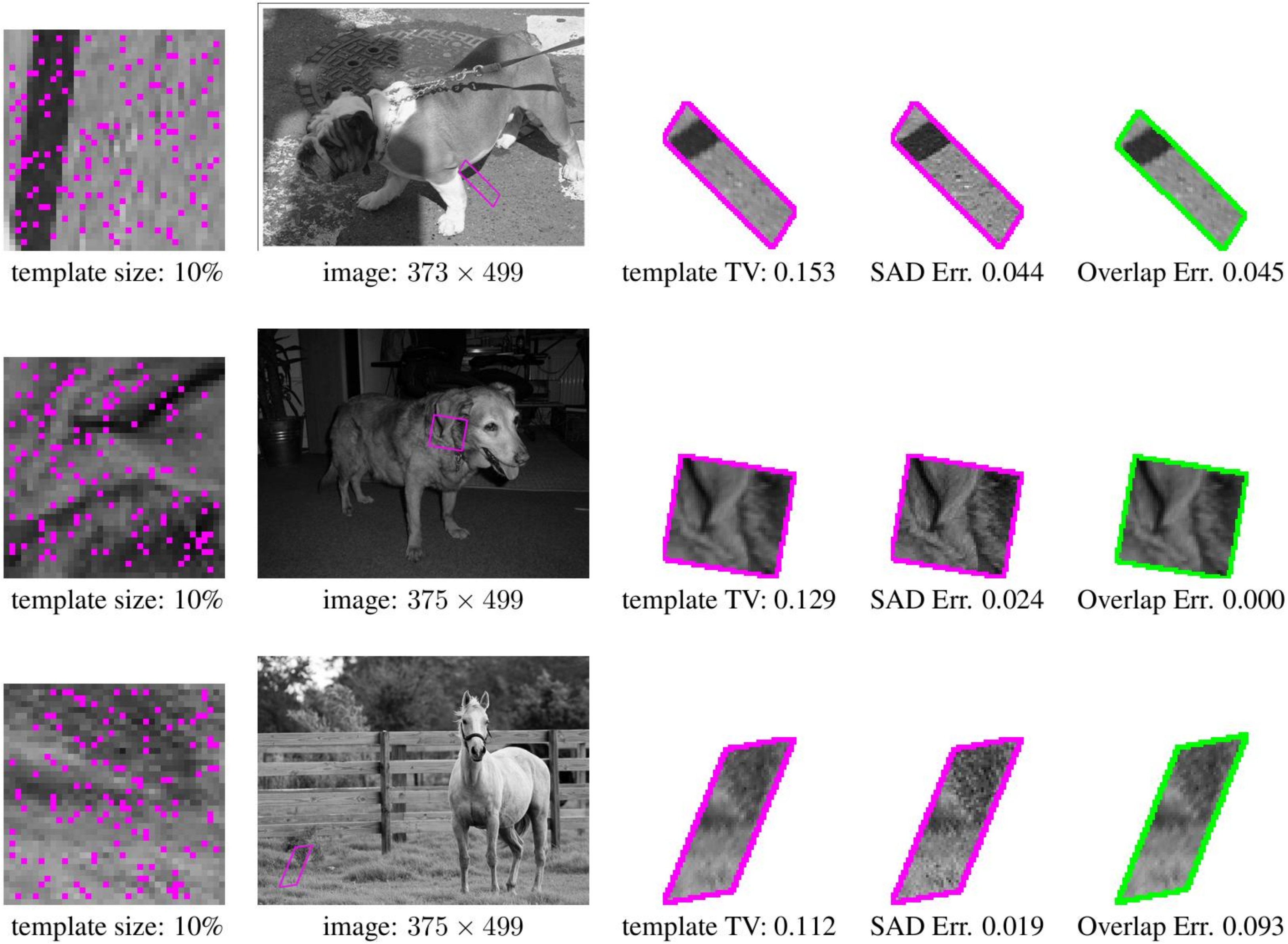
template size: 25% | image: 323 × 499 | template TV: 0.132 | SAD Err. 0.043 | Overlap Err. 0.067

template size: 25% | image: 375 × 499 | template TV: 0.066 | SAD Err. 0.037 | Overlap Err. 0.030

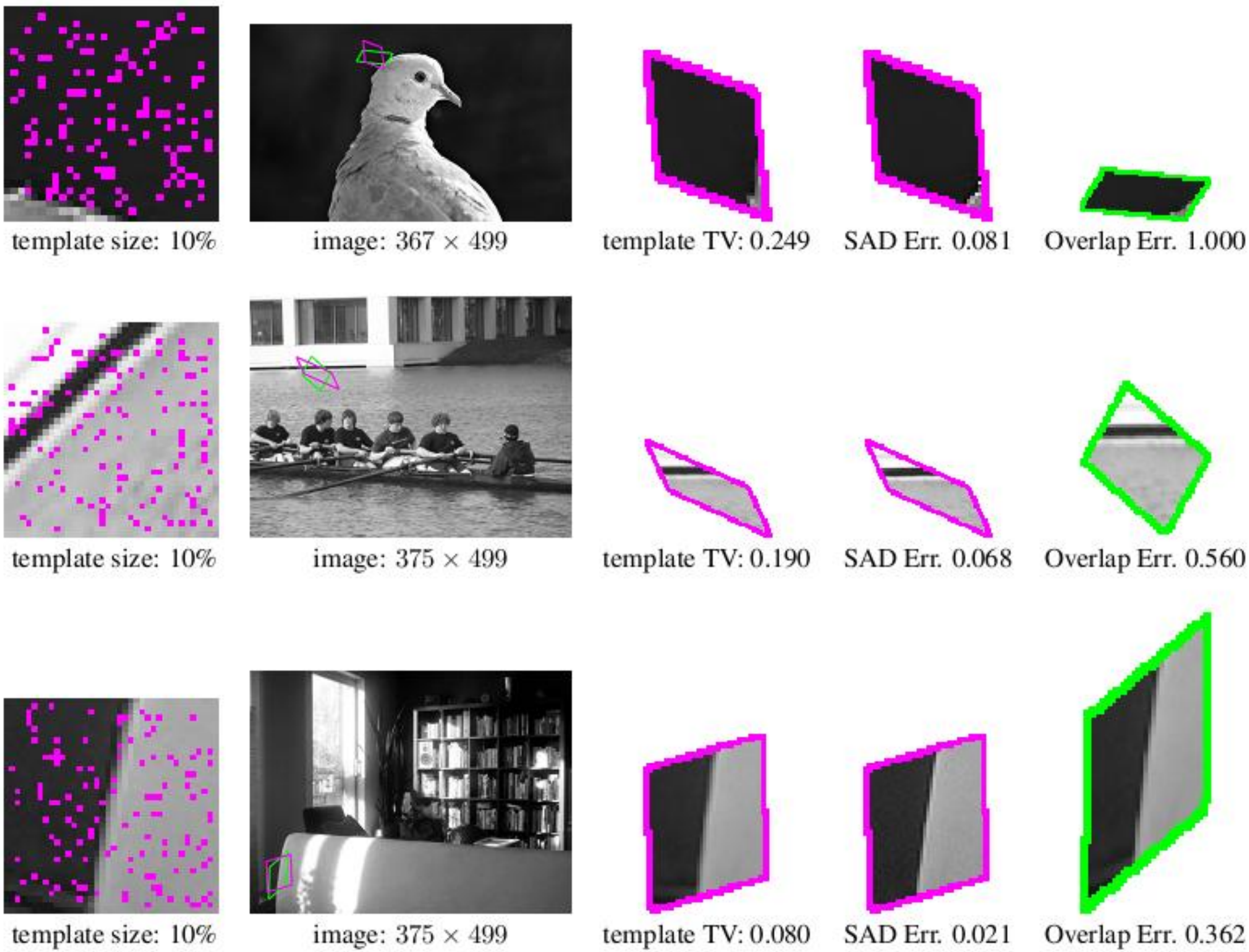
Template Dim: 15%



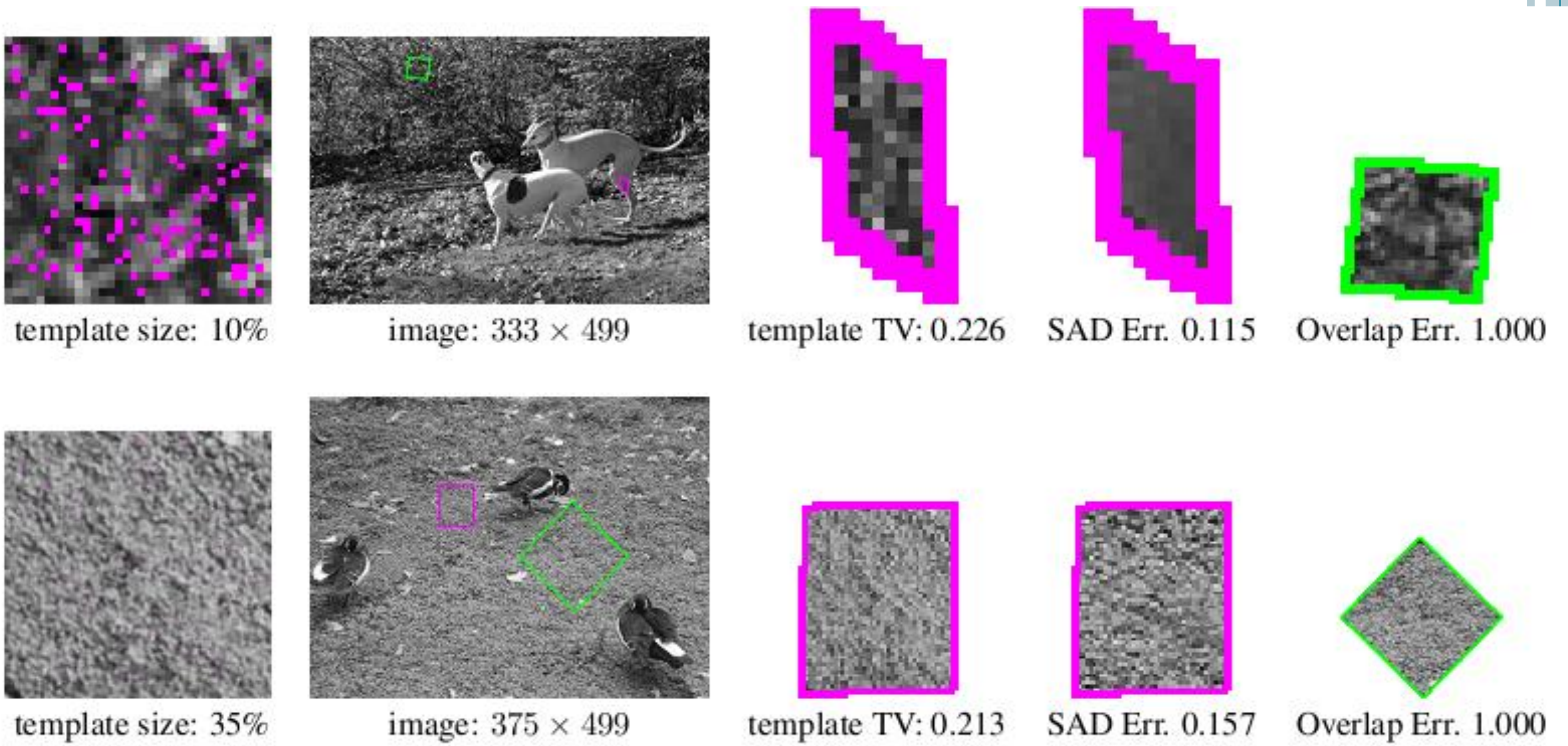
Template Dim: 10%



Bad overlap due to ambiguity



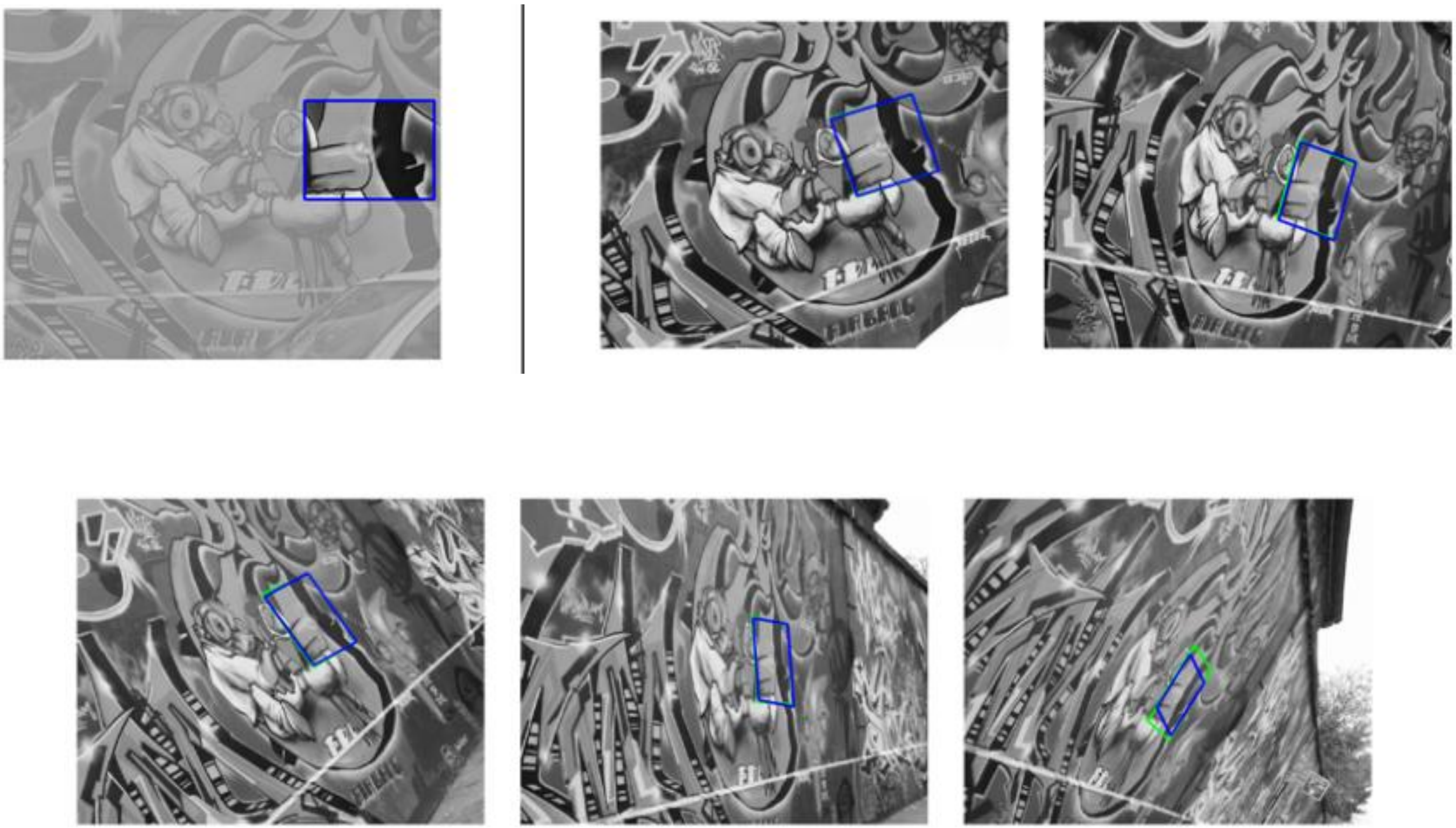
High SAD due to high TV and ambiguity



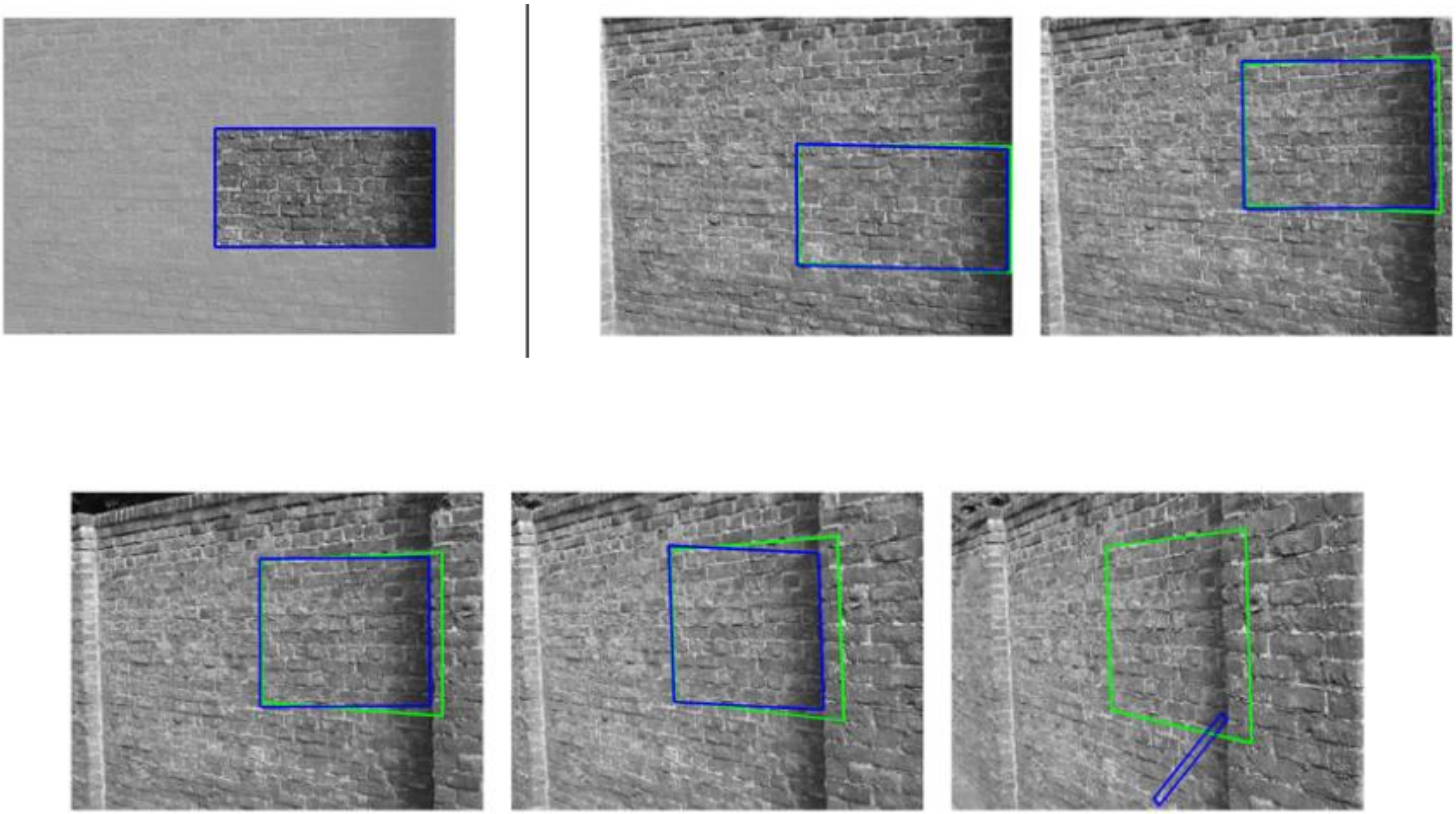
Experiment 2: Varying conditions

- 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)

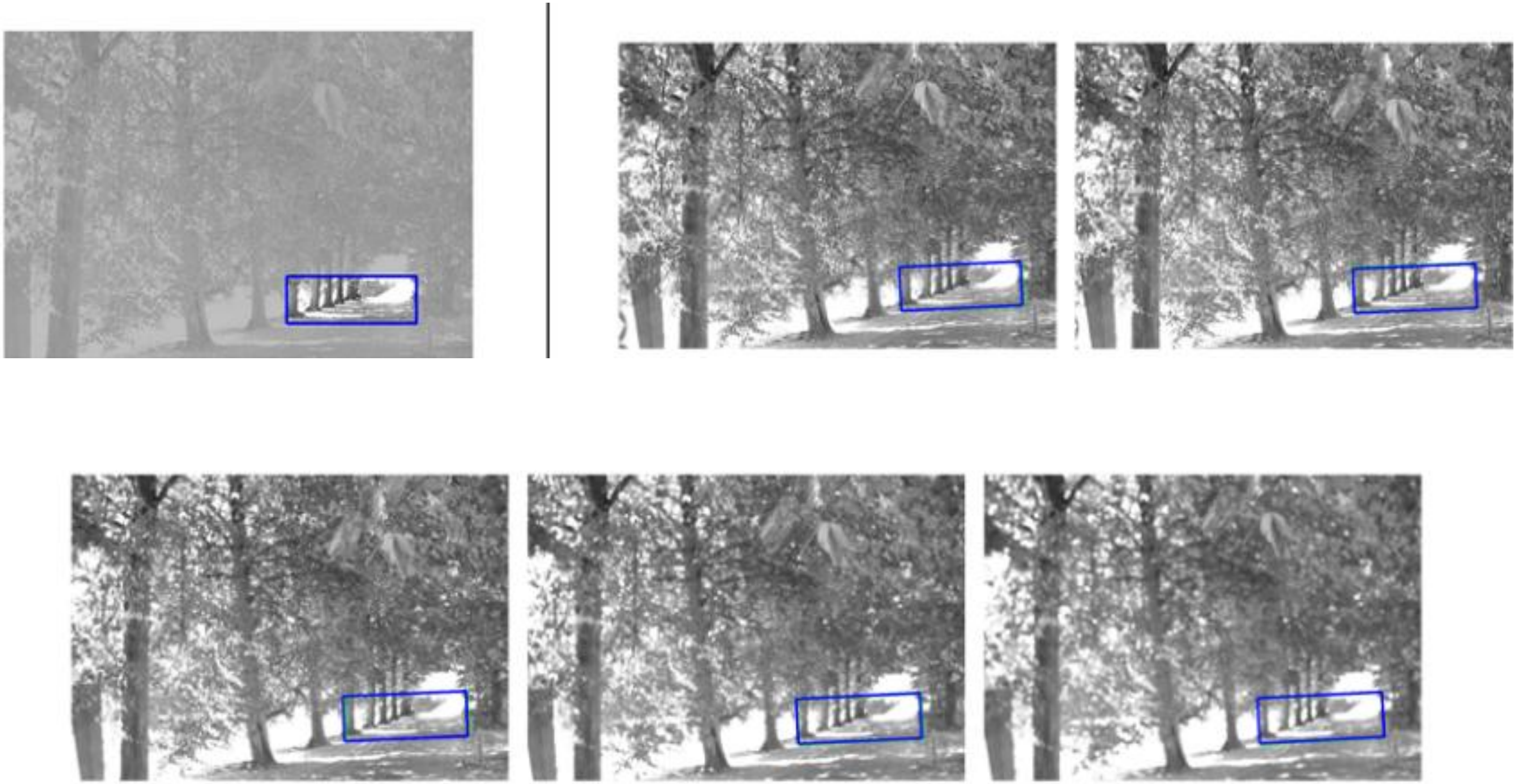
MIKOLAJCZYK– GRAFFITI (VIEWPOINT)



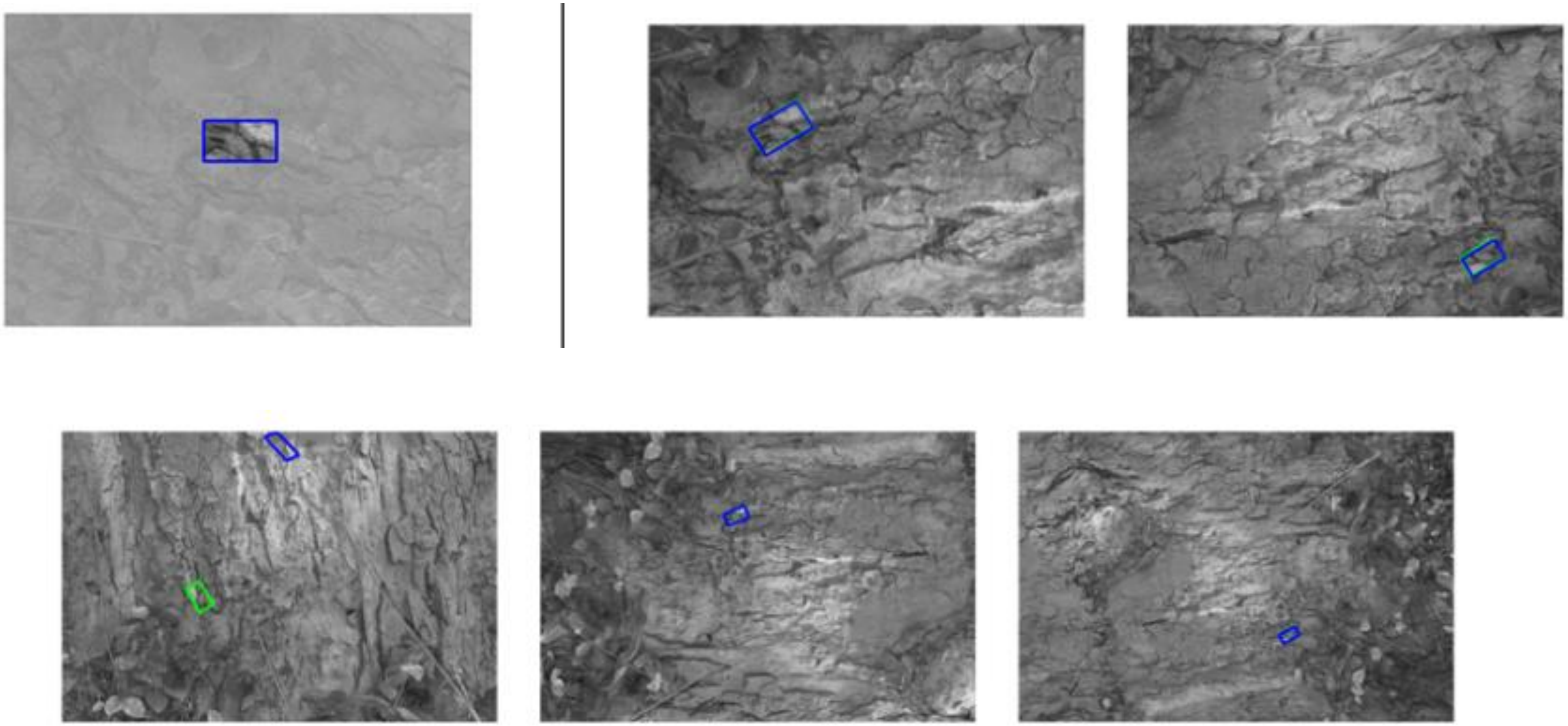
MIKOLAJCZYK– 'WALL' (VIEWPOINT)



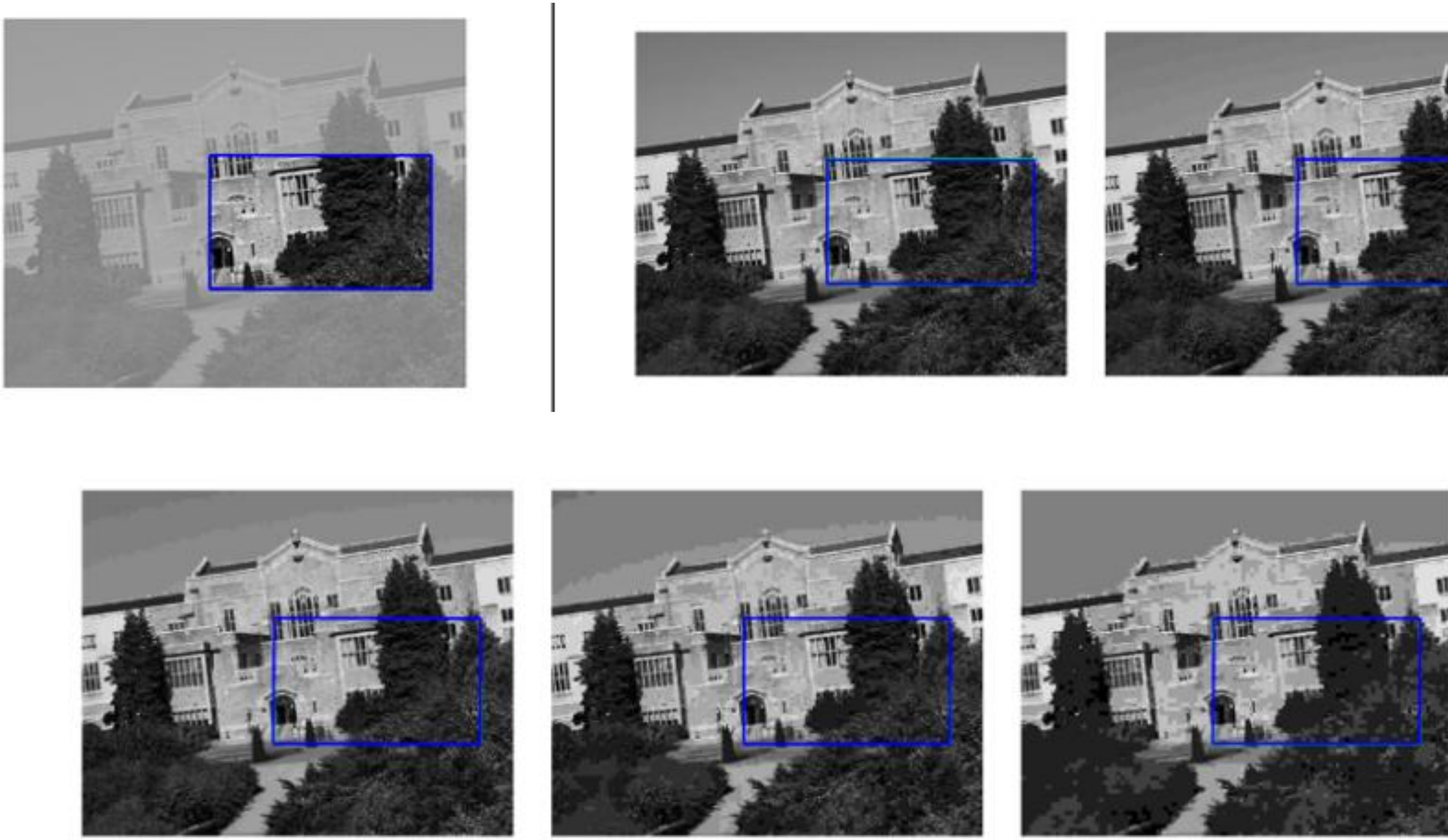
Mikolajczyk– ‘trees’ (blur)



MIKOLAJCZYK – ‘BARK’ (ZOOM+ROT)



MIKOLAJCZYK – ‘UBC’ (JPEG)



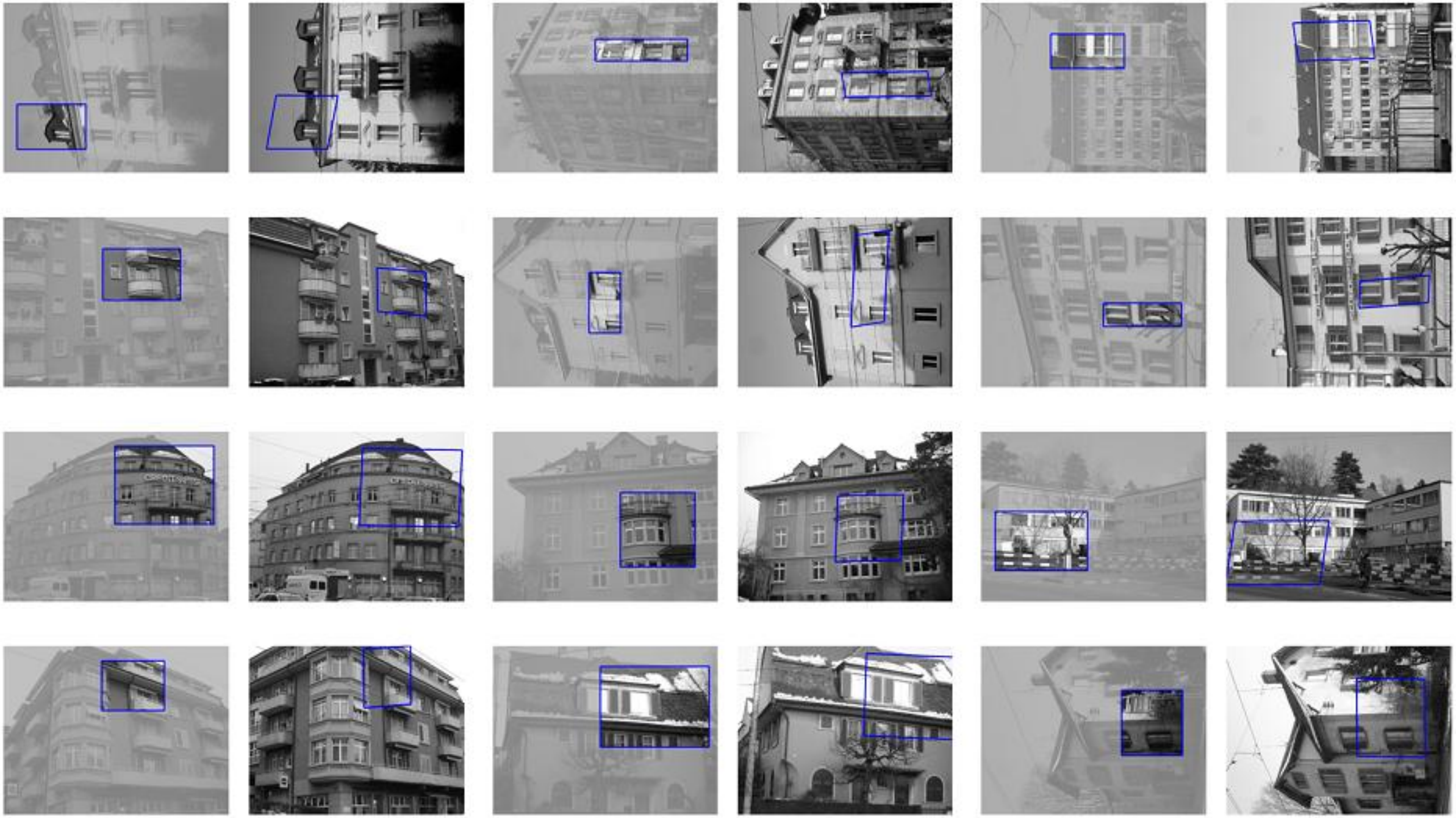
Experiment 3: Matching in real-world scenes

- 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

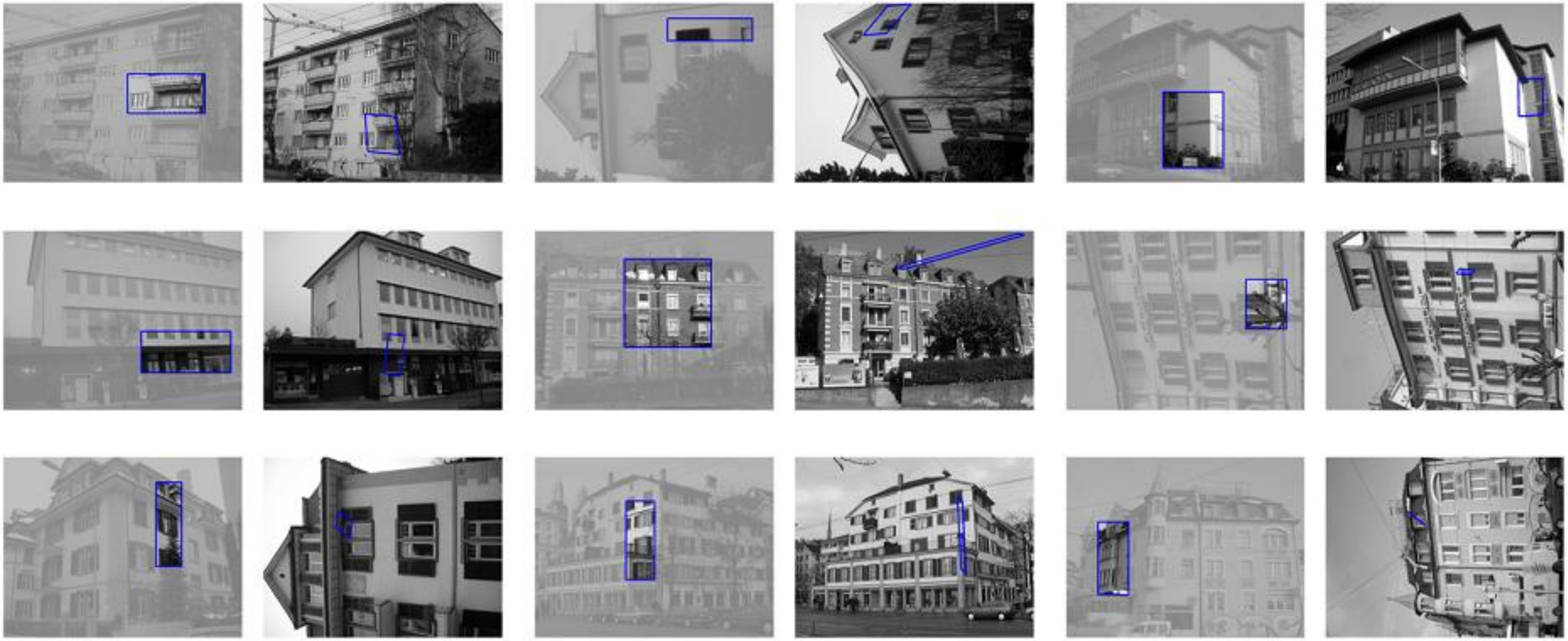
- **Results**:
  - 129 cases - ‘good’ matches
  - 40 cases – template doesn’t appear in second image
  - 12 cases – bad occlusion of template in second image
  - 19 cases – ‘failure’ (none of the above)

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

HTTP://WWW.ENG.TAU.AC.IL/~ORON/

Thank you for your attention