Feature Matching

Correspondence

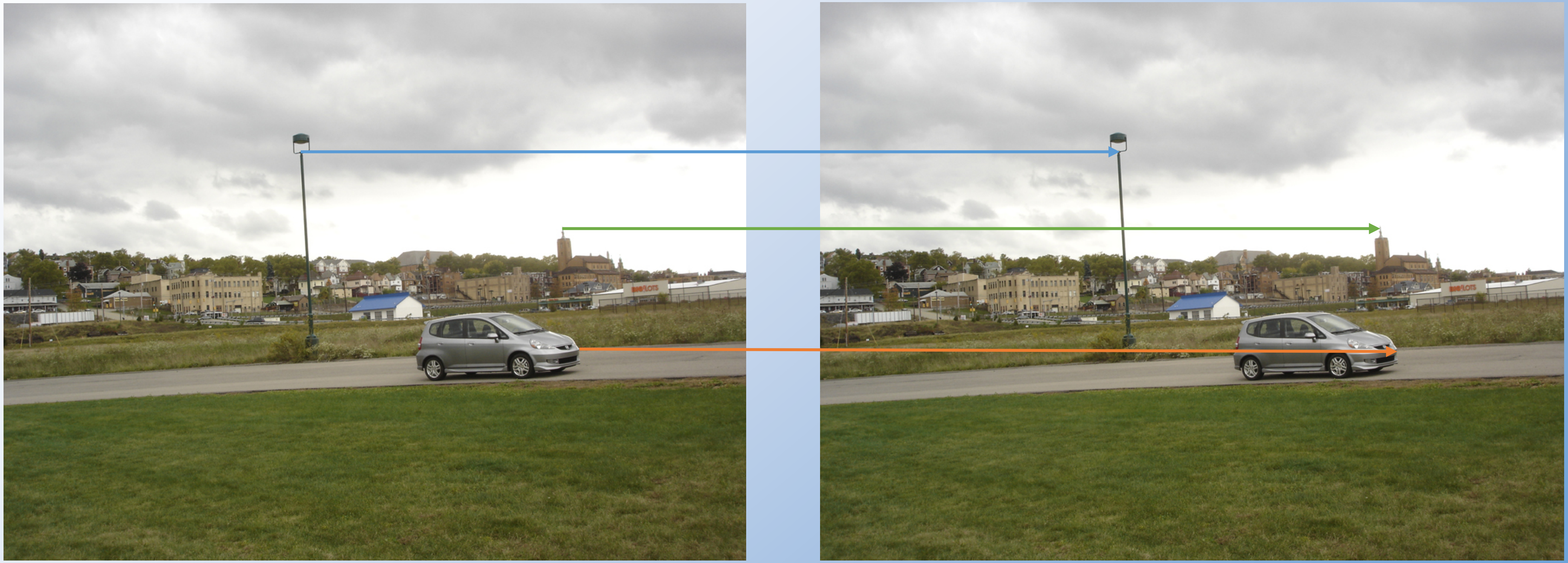
The correspondence problem:

- Given a pixel p1 in image I1, find the pixel p2 in image I2 such that p1 and p2 represent the same point in the world.

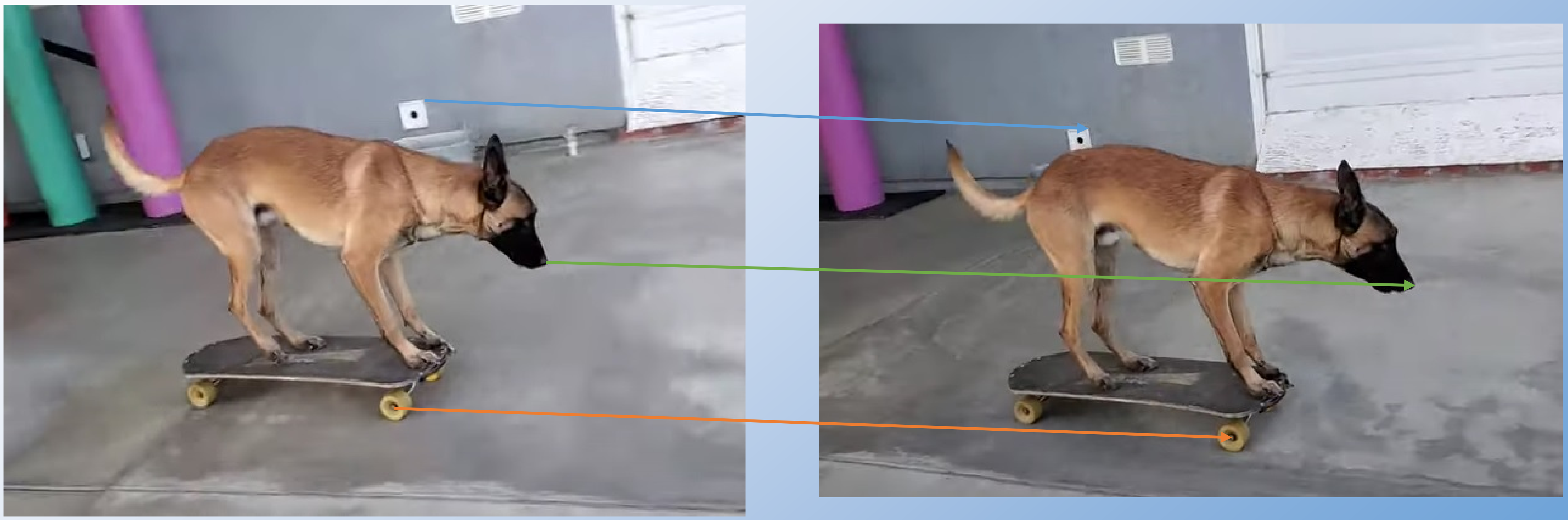
Necessary for:

- Alignment
- 3D reconstruction
- Tracking
- ...

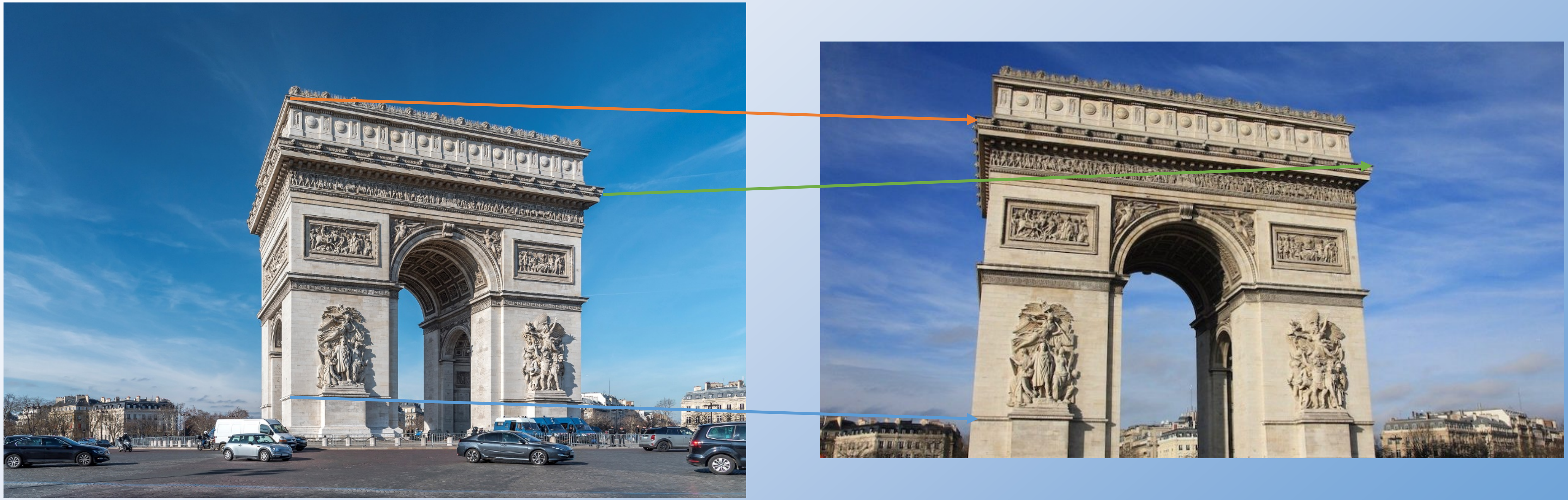
Correspondence



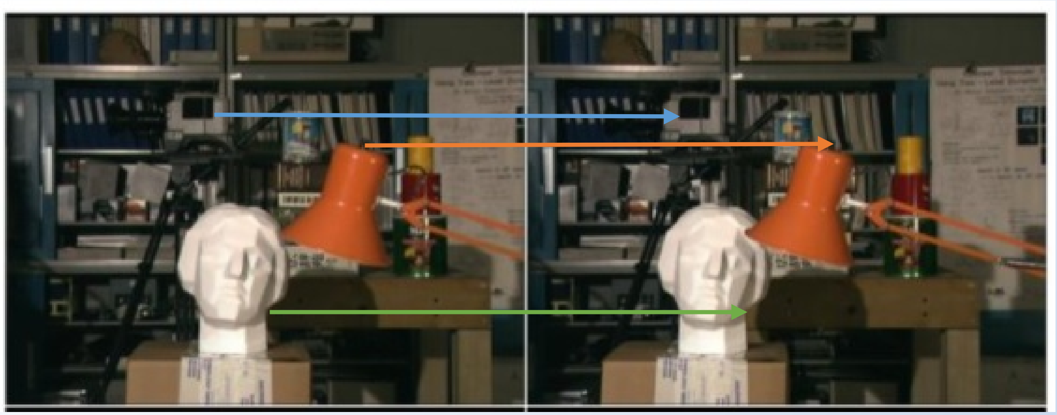
Correspondence



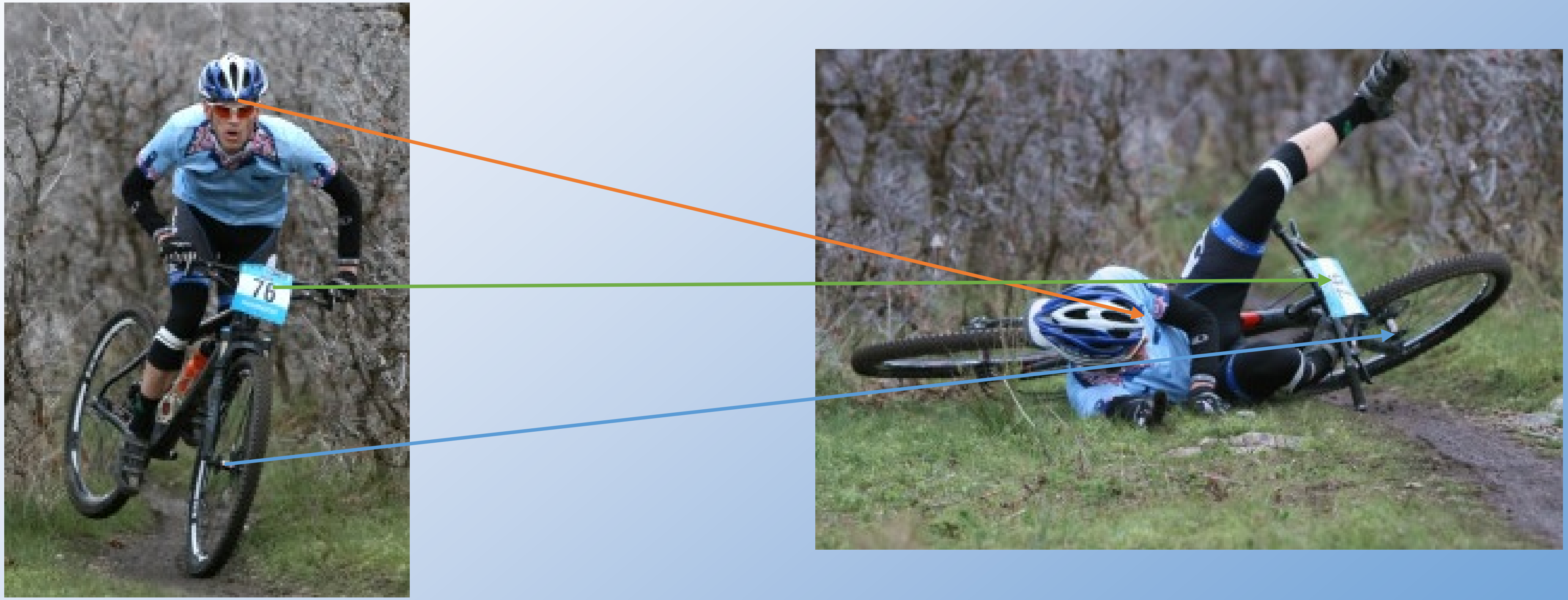
Correspondence



Correspondence



Correspondence



Correspondence

- Solution requirements:
  - Invariant to:
    - Rotation
    - Scale
- In general, the problem is unsolvable
- ‘Probably correct’ correspondences can be found for certain **salient regions** – regions with high gradient in at least two directions (a.k.a. corners)

Salient Regions

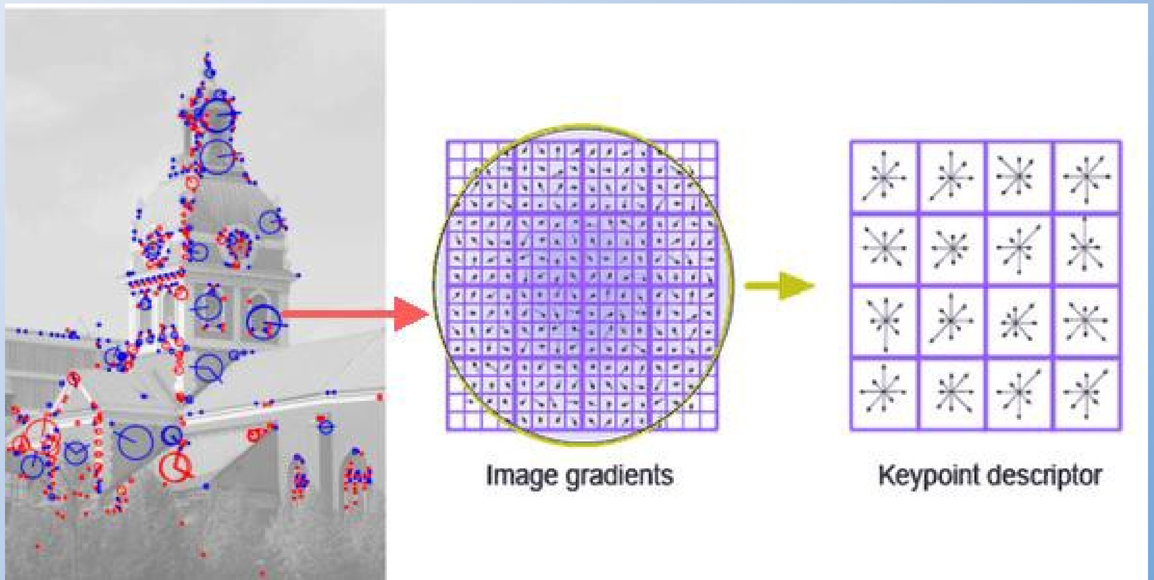
Several algorithms have been proposed to find salient regions in an image:

- SIFT – Scale invariant feature transform
- SURF- Speeded up robust features
- ORB - Oriented FAST and rotated BRIEF

Salient Regions

Several algorithms have been proposed to find salient regions in an image:

- SIFT – Scale invariant feature transform
- SURF- Speeded up robust features
- ORB - Oriented FAST and rotated BRIEF



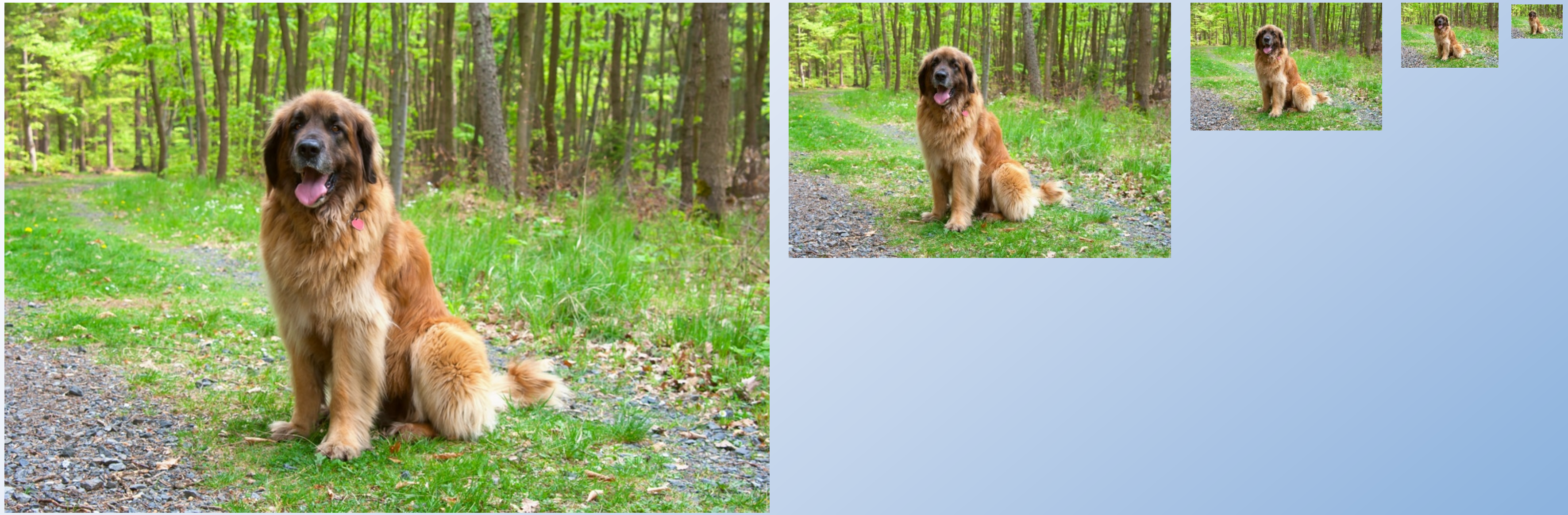
Salient Regions

Main Ideas:

- Scale space
- Corner detection
- Non-local maxima suppression
- Rotational invariance

Scale space

Salient region can appear at any scaled, so image is progressively downsampled and analyzed



Corner Detection

Corner detection:

- Define saliency metric
  - Usually dependent on the product of vertical and horizontal gradients
  - Select all regions with saliency above a given threshold

Non-local-maxima suppression

Eliminate all regions that have neighbors with higher saliency in space or in scale

Rotational invariance

Rotate region so highest gradient corresponds to a fixed axis (usually x or y)

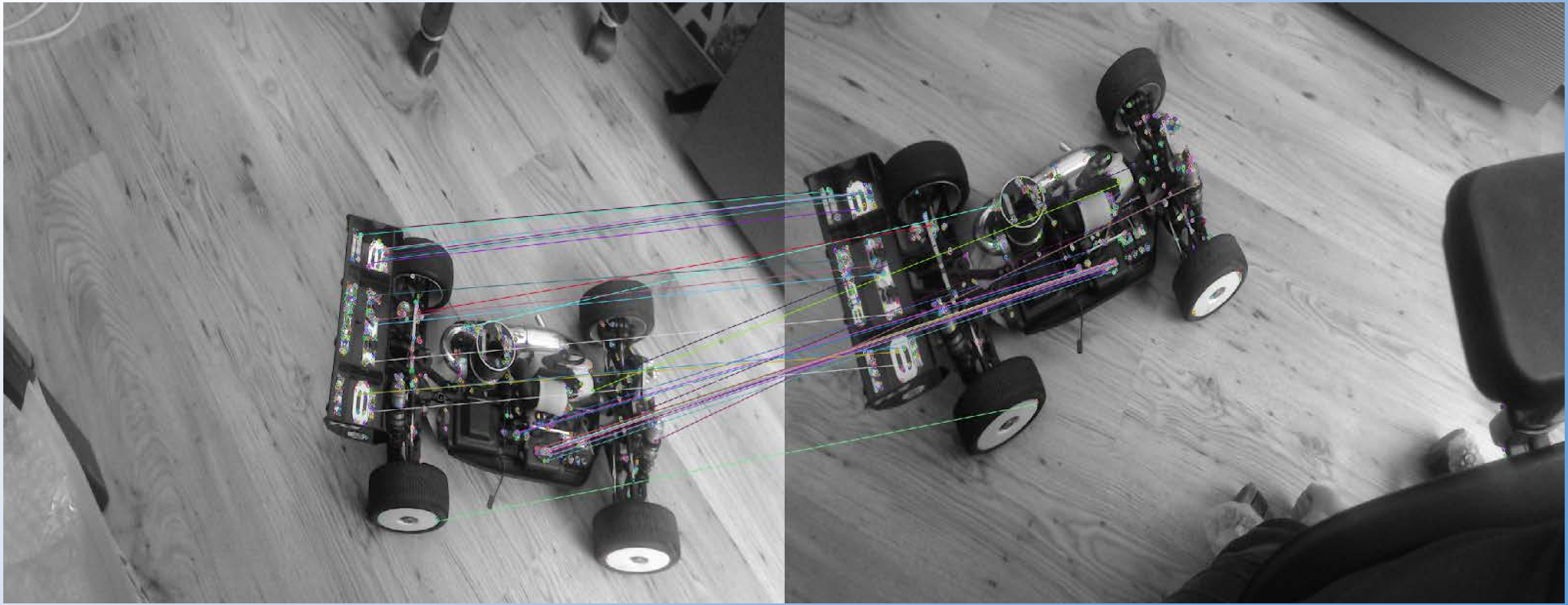
Notice that rotation is equivalent to shifting the histogram of gradients

Matching

For every salient region in image 1 find the most similar salient region in image 2

Matching

For every salient region in image 1 find the most similar salient region in image 2



Matching

Not all regions will be matched

Greedy algorithms often used, but don’t guarantee optimal correspondences

O(n^3) solution implemented in Scipy and OpenCV – the Hungarian algorithm

Finding a global match from point matches

The RANSAC Algorithm

```
Best Model = [ ]
For i = 1 to max_iterations
    • Select four point correspondences at random
    • Generate a model M using those points
    • Determine how many of the points not chosen can be approximated by M
    • Generate M’ using the points that could be approximated by M
    • If M’ is better than Best Model, replace Best Model by M’
Return Best Model
```