CS 103: Representation Learning, Information Theory and Control

Lecture 3, Jan 25, 2019

Seen last time

What is a nuisance for a task?

How do we design nuisance invariant representations?

Invariance, equivariance, canonization

A linear transformation is group equivariant if and only if it is a group convolution (no proof)

Today's program

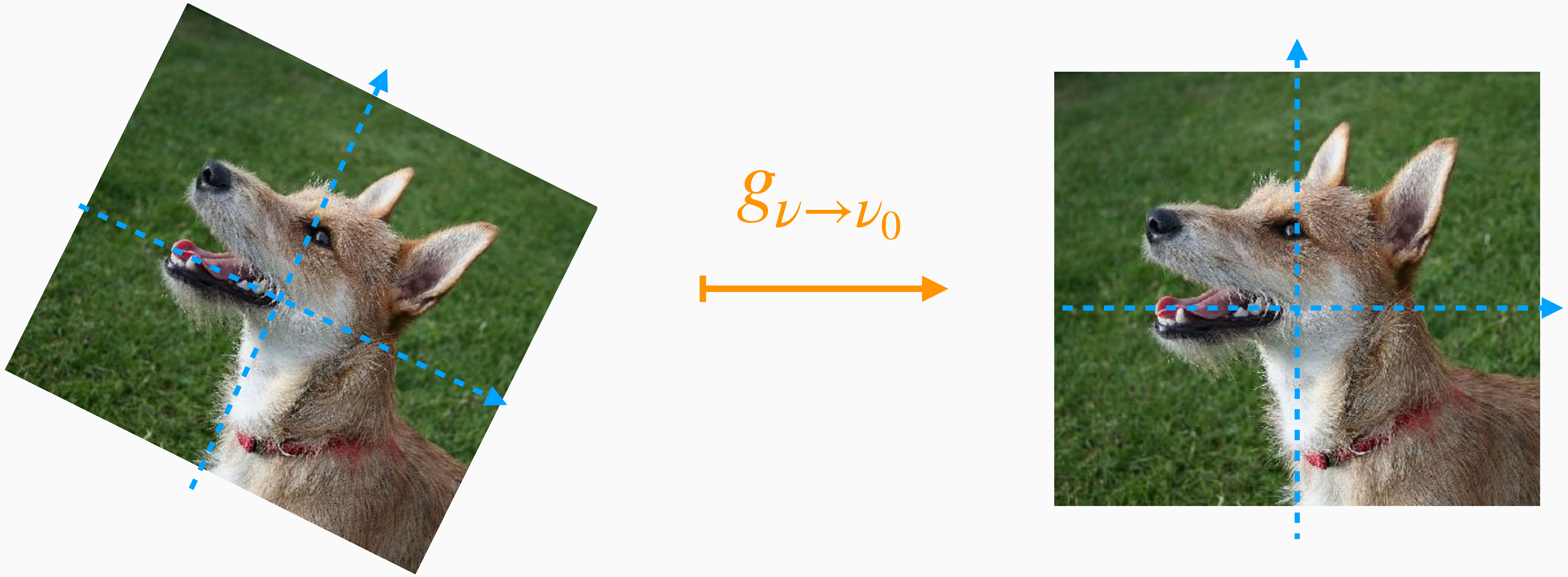
1. A linear transformation is group equivariant if and only if it is a group convolution
    - Building equivariant representations for translations, sets and graphs
2. Image canonization with equivariant reference frame detector
    - Applications to multi-object detection
3. Accurate reference frame detection: the SIFT descriptor
    - A sufficient statistic for visual inertial systems

Canonization

Invariance by canonization

Idea: Instead of finding an invariant representation, apply a transformation to put the input in a standard form.

$$I(\xi, \nu) \longmapsto g_{\nu \to \nu_0} \circ I(\xi, \nu) = I(\xi, \nu_0)$$



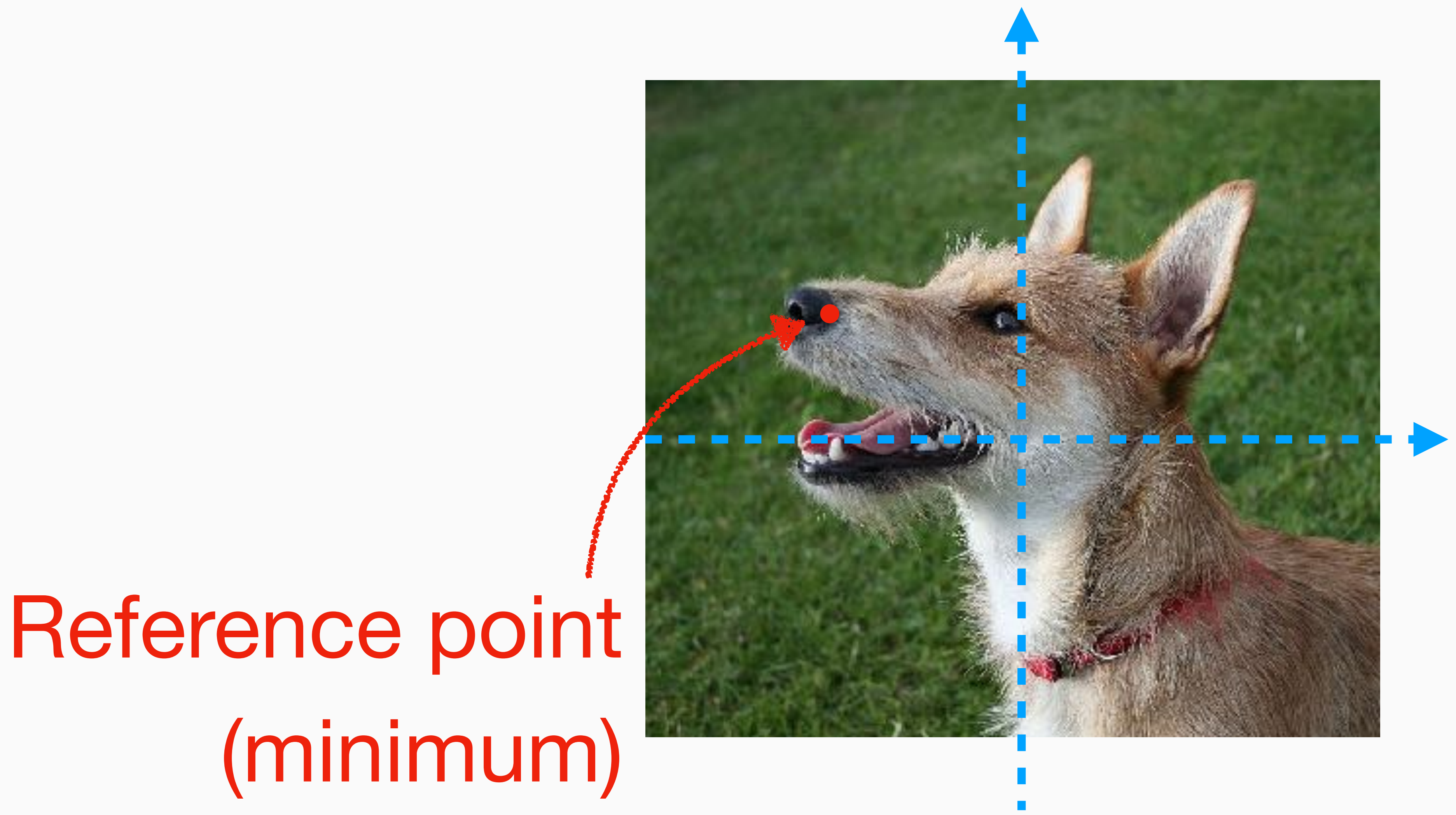
$$g_{\nu \to \nu_0}$$

Canonization for translations

Suppose we want to canonize the image with respect to translations.

1. Decide a reference point that is equivariant for translations.

   Examples: The barycenter of the image, the maximum (assuming it's unique)
2. Find the position of the reference point
3. Center the reference point

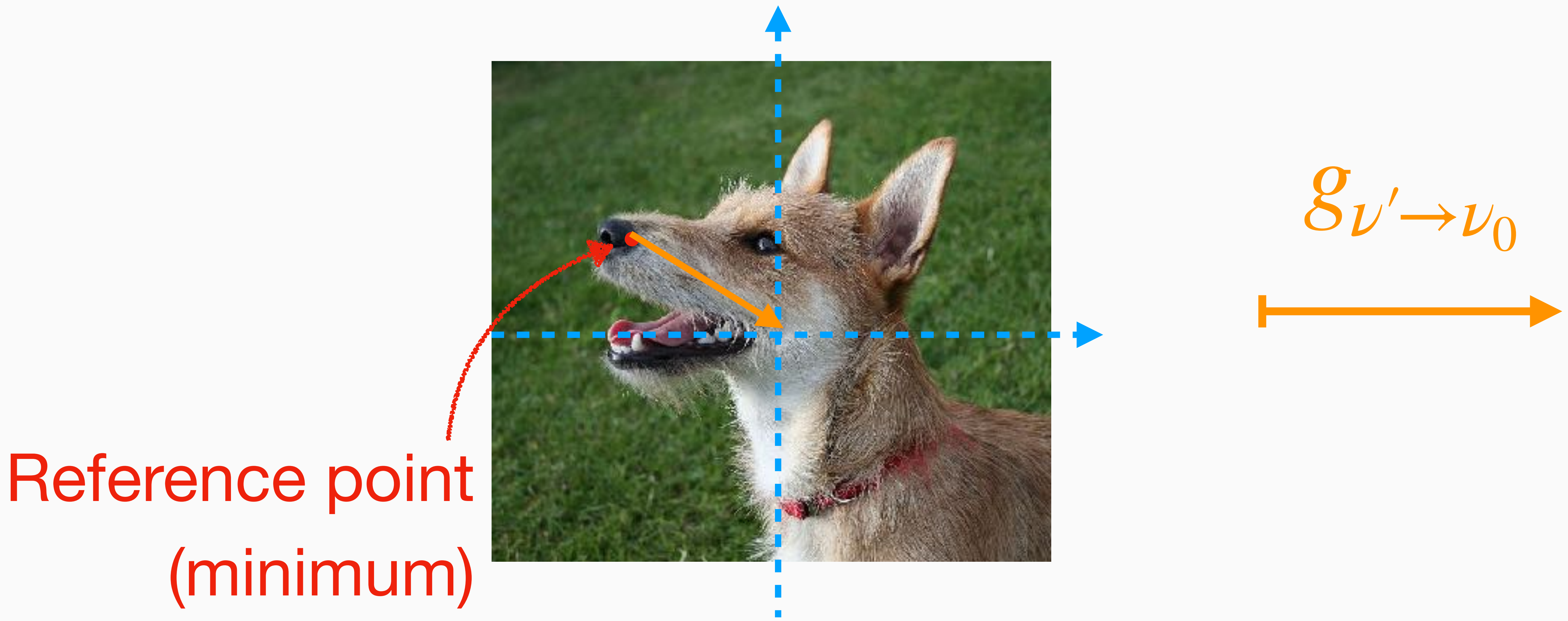


Canonization for translations

Suppose we want to canonize the image with respect to translations.

1. Decide a reference point that is equivariant for translations.

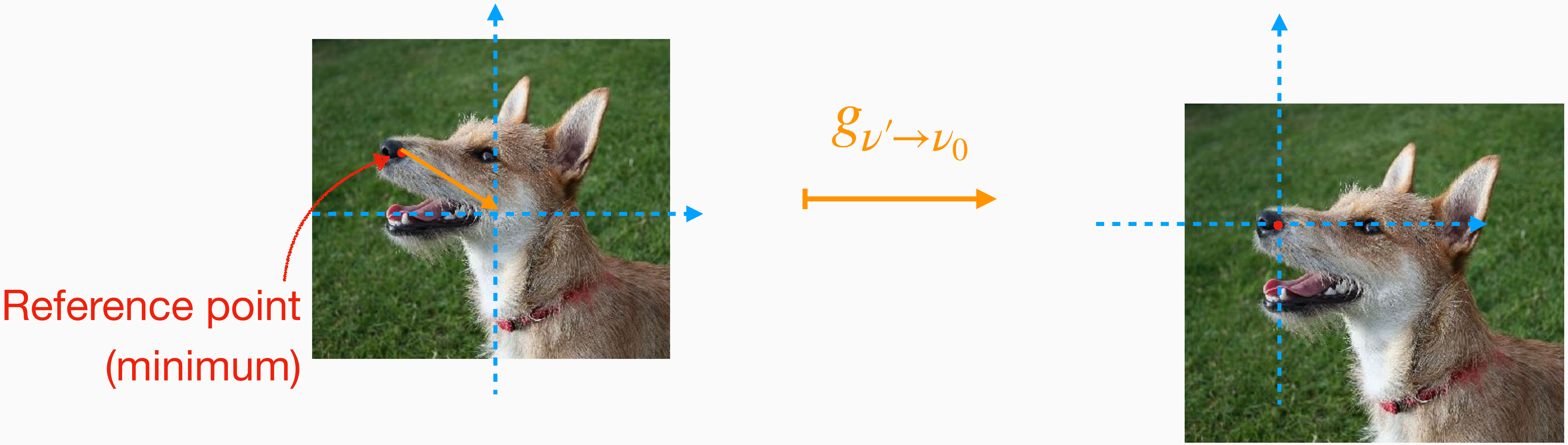
   Examples: The barycenter of the image, the maximum (assuming it's unique)
2. Find the position of the reference point
3. Center the reference point



Canonization for translations

Suppose we want to canonize the image with respect to translations.

1. Decide a reference point that is equivariant for translations.
   Examples: The barycenter of the image, the maximum (assuming it's unique)
2. Find the position of the reference point
3. Center the reference point



Equivariant reference frame detector

A reference frame detector *R* for a group *G* is any function *R(x): X* → *G* such that

$$R(g \cdot x) = g \cdot R(x)$$

That is, a reference frame detector is any equivariant function from *X* to *G.*

Example: Let *G* = $\mathbf{R}^2$ be the group of translations. Then R(x) = “position of the maximum of x” is a reference frame, assuming the maximum is unique.

From equivariant frame detector to invariant representations

Proposition. Let R be a reference frame detector for the group G. Define a representation $f(x)$ *as:*

$$f(x) = R(x)^{-1} \cdot x$$

Then $f(x)$ is a G-invariant representation.

From equivariant frame detector to invariant representations

Proposition. Let *R* be a reference frame detector for the group *G*. Define a representation *f(x) as:*

$$f(x) = R(x)^{-1} \cdot x$$

Then *f(x)* is a *G*-invariant representation.

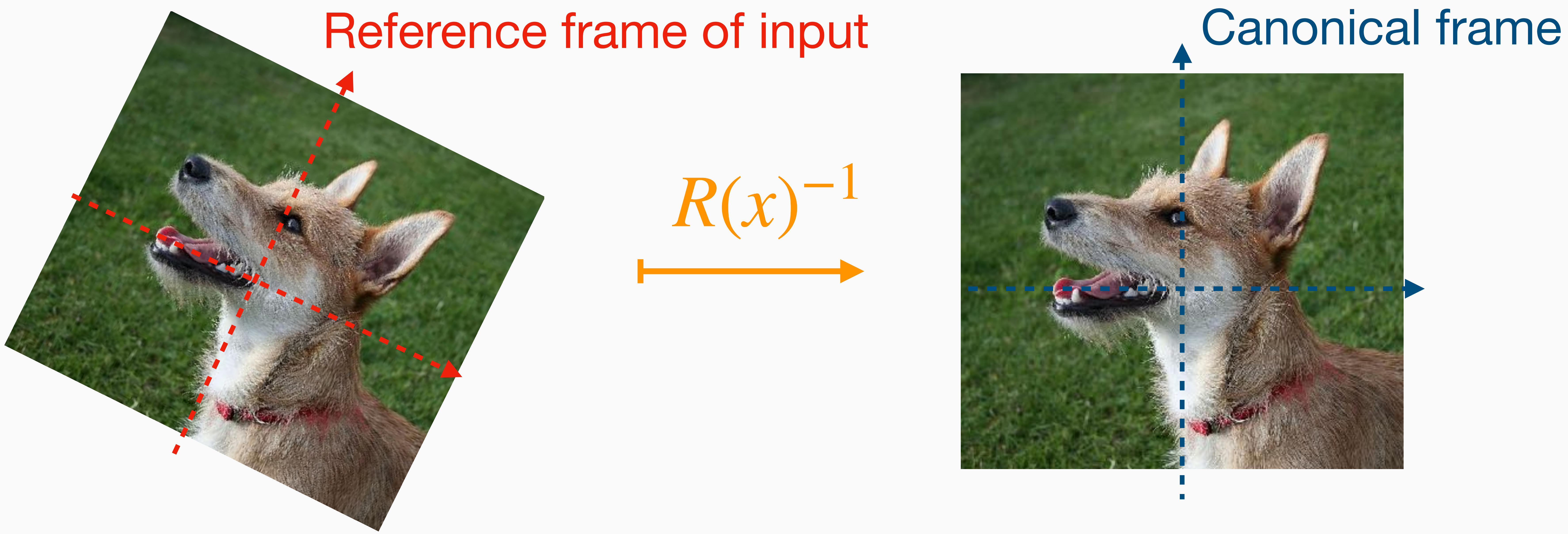
Proof:

$$\begin{aligned} f(g \cdot x) &= R(g \cdot x)^{-1} \cdot (g \cdot x) \\ &= (g \cdot R(x))^{-1} \cdot g \cdot x \\ &= R(x)^{-1} \cdot g^{-1} \cdot g \cdot x \\ &= R(x)^{-1} \cdot x \\ &= f(x) \end{aligned}$$

The canonization pipeline

Canonization consists of the following steps

1. Build an equivariant **reference frame detector**
2. Choose a "**canonical**" reference frame
3. Find the reference frame of the input image
4. Invert the transformation to make the reference frame canonical



Some examples of canonization in vision

Document analysis: Find border of the document and un-warp the image prior to analysis.

Also: Normalize contrast and illumination

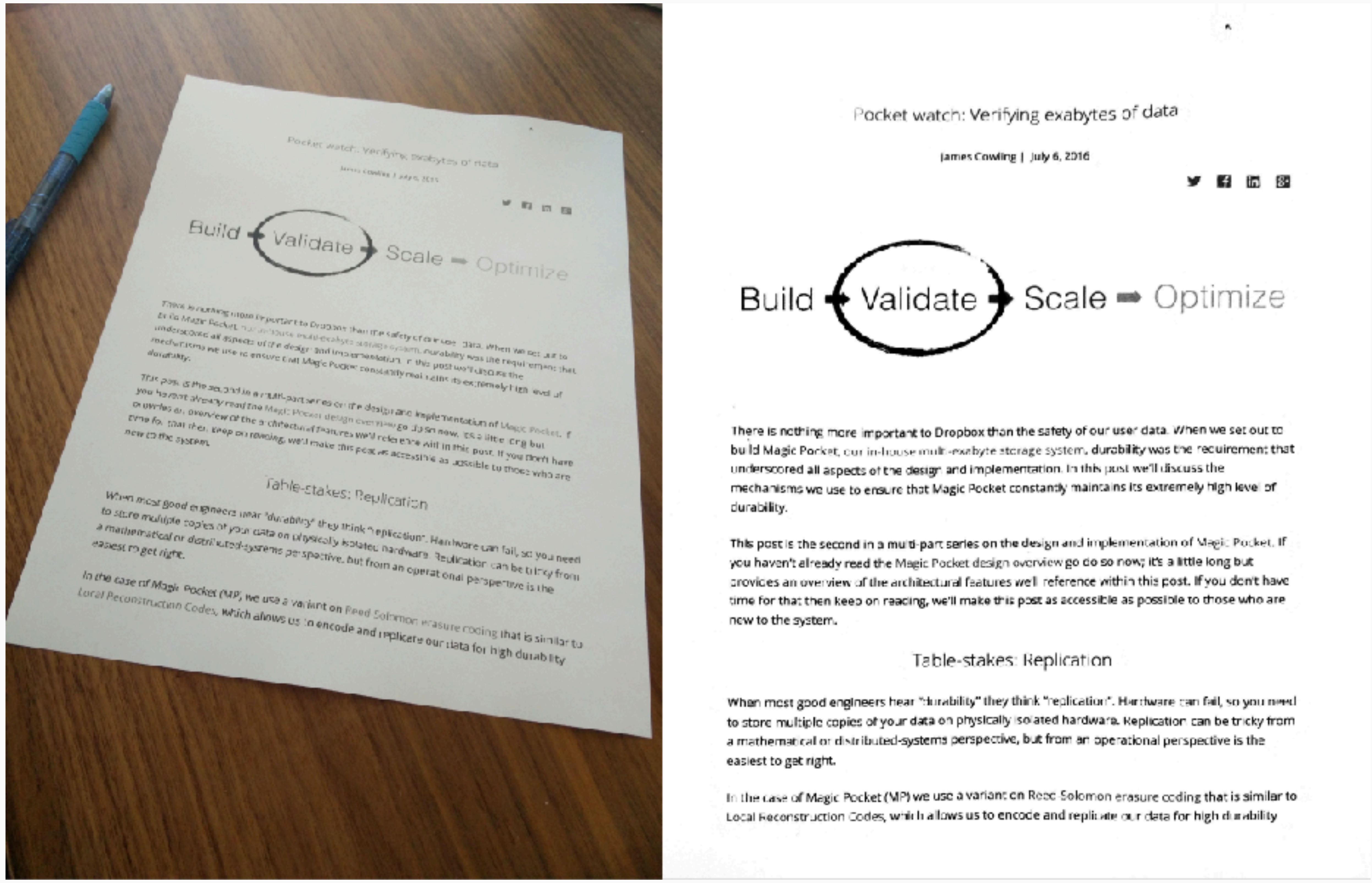
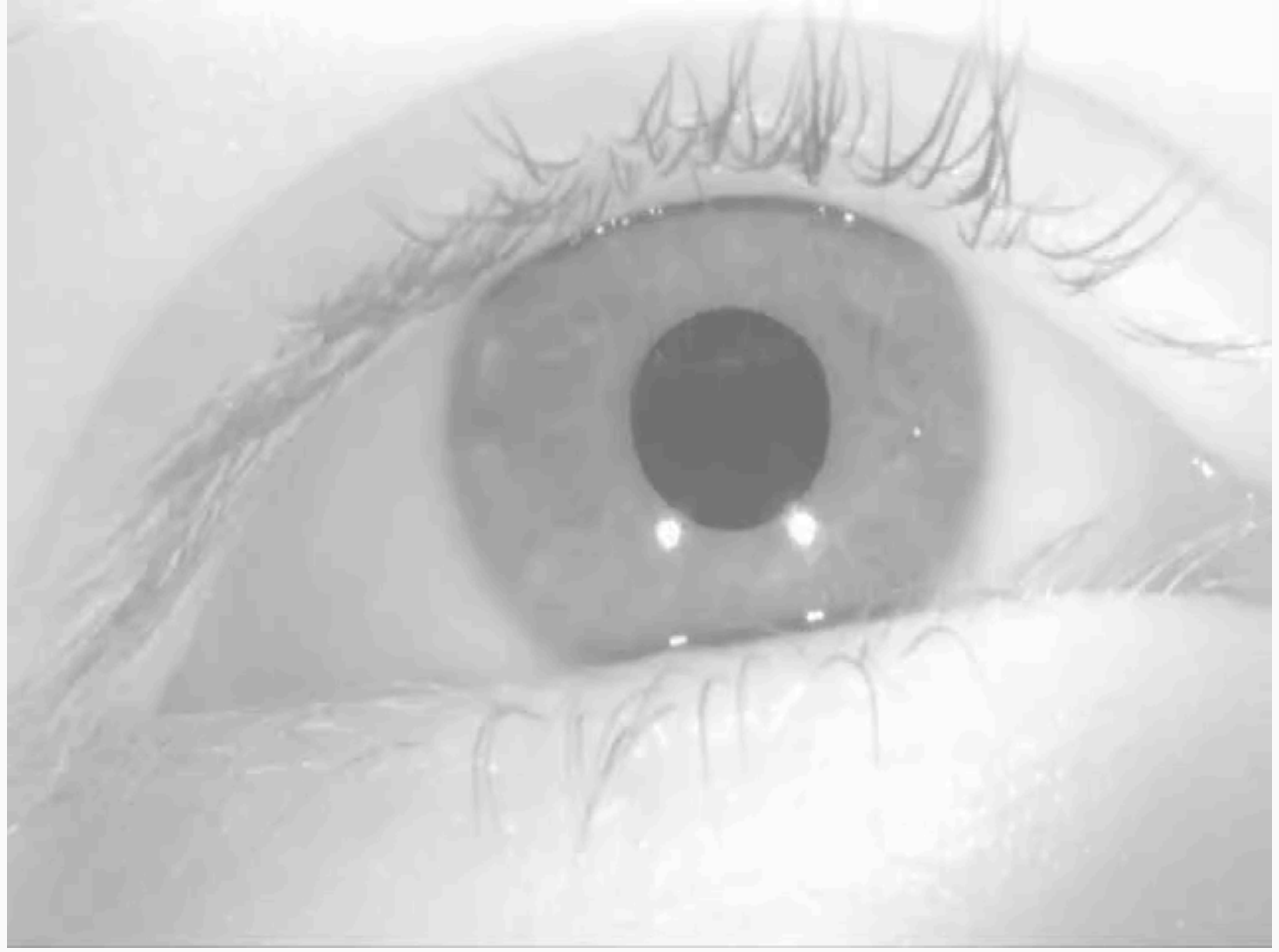


Image from https://blogs.dropbox.com/tech/2016/08/fast-document-rectification-and-enhancement/

Saccades

Eyes move rapidly while looking at a fixed object.



Image

Trace of saccades

Can we consider this a form of translation invariance by canonization?

Video and Images from https://en.wikipedia.org/wiki/Saccade

Saccades

Eyes move rapidly while looking at a fixed object.

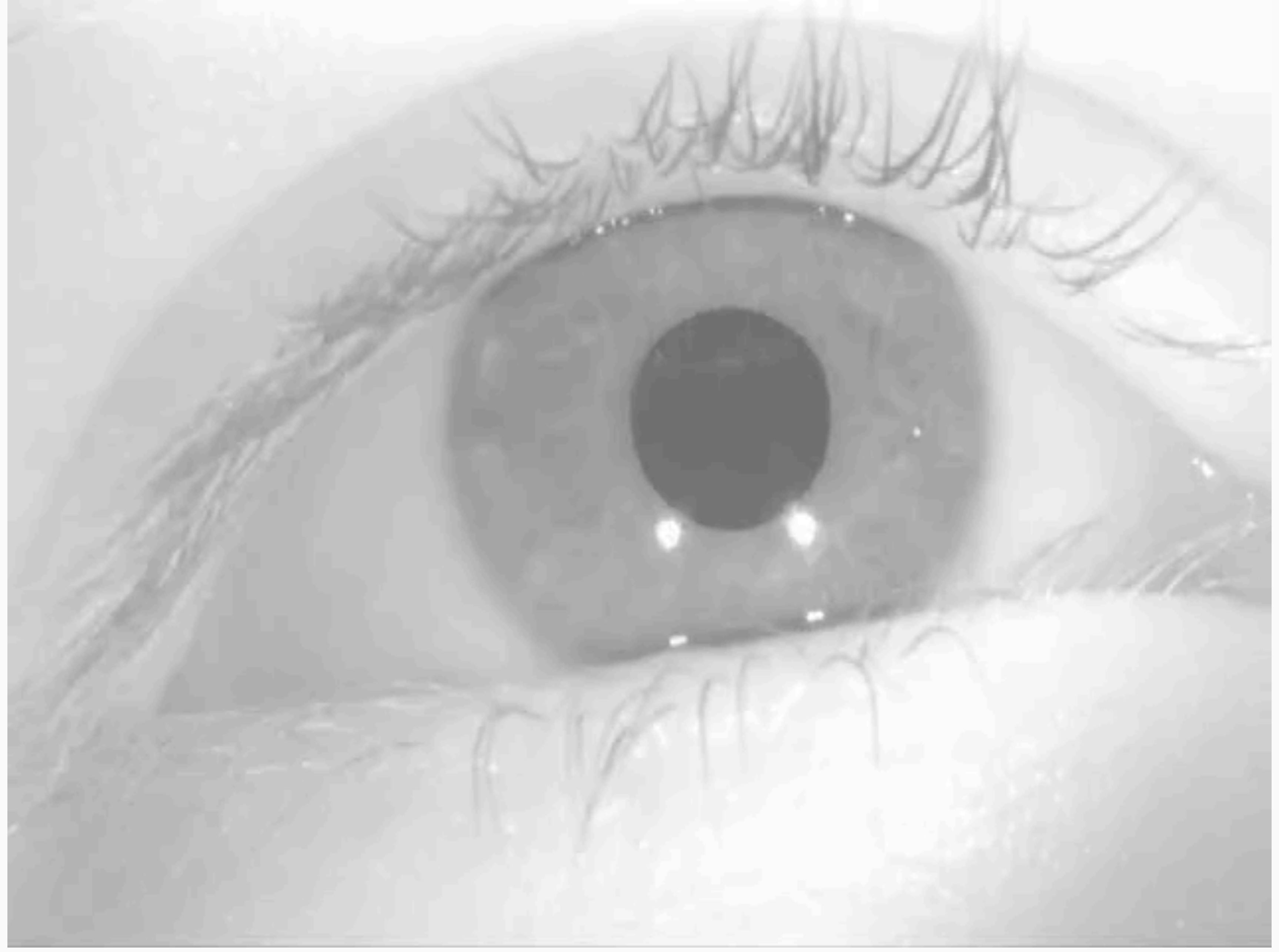


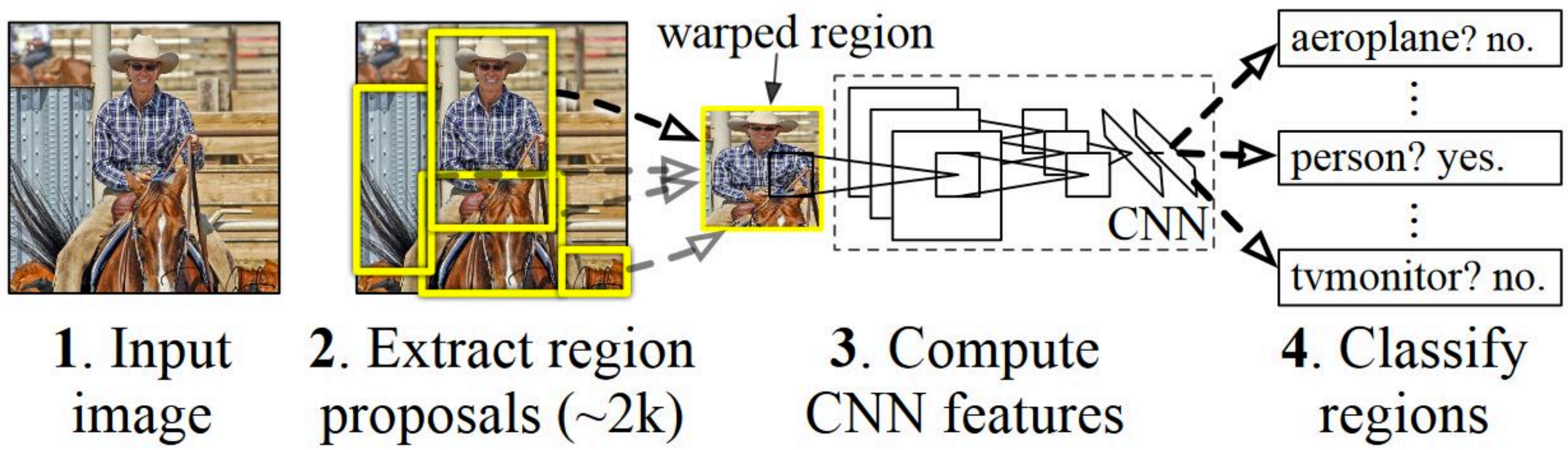
Image   Trace of saccades

Can we consider this a form of translation invariance by canonization?

The R-CNN model for multi-object detection

Region proposal: find regions of the image that may contain an interesting object (i.e., reference frame proposal)

CNN classifier: warp the region to put it in canonical form (invariance) and feed it to a classifier



Region proposal + CNN classifier = R-CNN

Image from Girshick et al., 2014

Region Proposal

Selective Search for Object Recognition, Uijlings et al., 2013

Originally: hand-crafted proposal mechanisms based on saliency, uniformity of texture, scale, and so on.

Region Proposal

Selective Search for Object Recognition, Uijlings et al., 2013

Originally: hand-crafted proposal mechanisms based on saliency, uniformity of texture, scale, and so on.

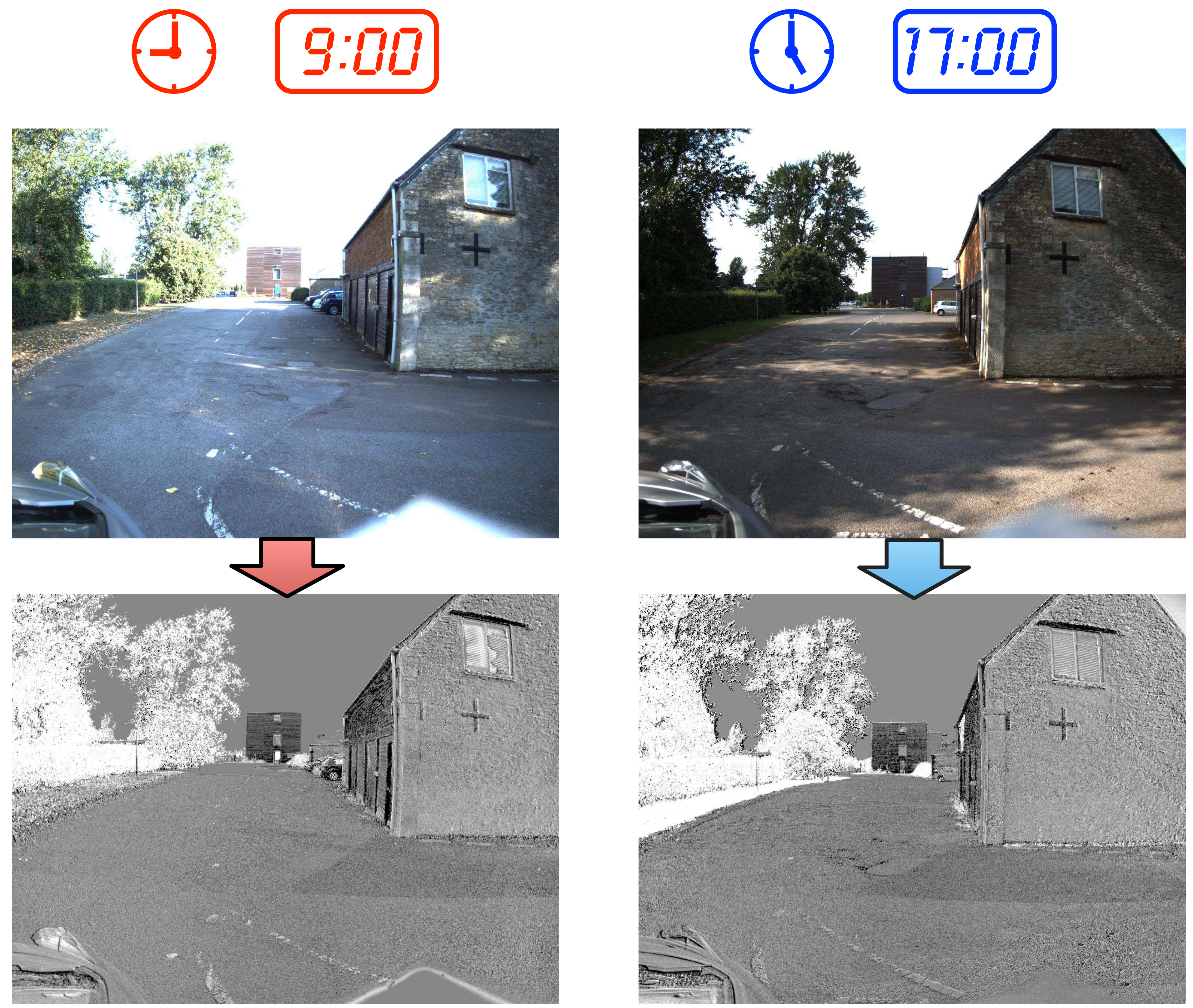
Illumination invariant colorspace

Region Proposal

Selective Search for Object Recognition, Uijlings et al., 2013

Originally: hand-crafted proposal m exture, scale, and so on.

Illumination invariant colorspace



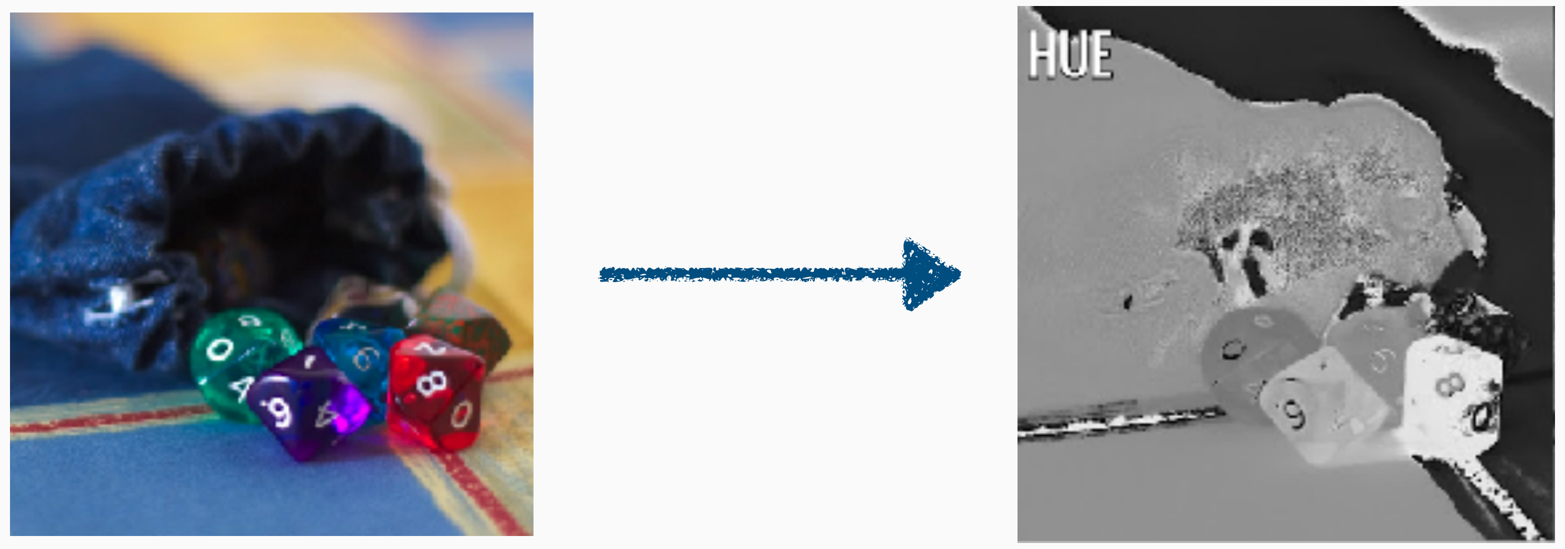
Maddern et al., ICRA 2014

Region Proposal

Selective Search for Object Recognition, Uijlings et al., 2013

Originally: hand-crafted proposal mechanisms based on saliency, uniformity of texture, scale, and so on.

Illumination invariant colorspace



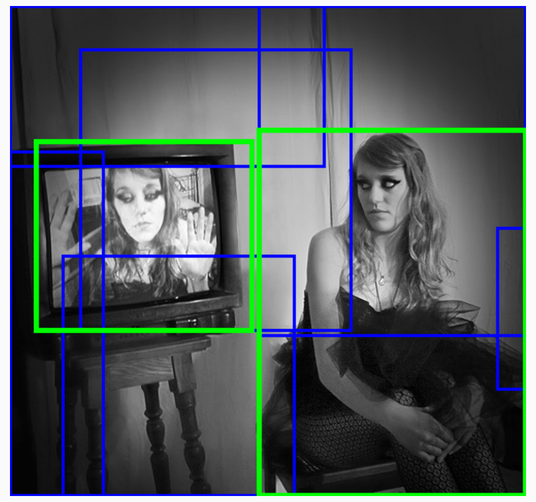
Region Proposal

Selective Search for Object Recognition, Uijlings et al., 2013

Originally: hand-crafted proposal mechanisms based on saliency, uniformity of texture, scale, and so on.

Illumination invariant colorspace

Initial region proposal

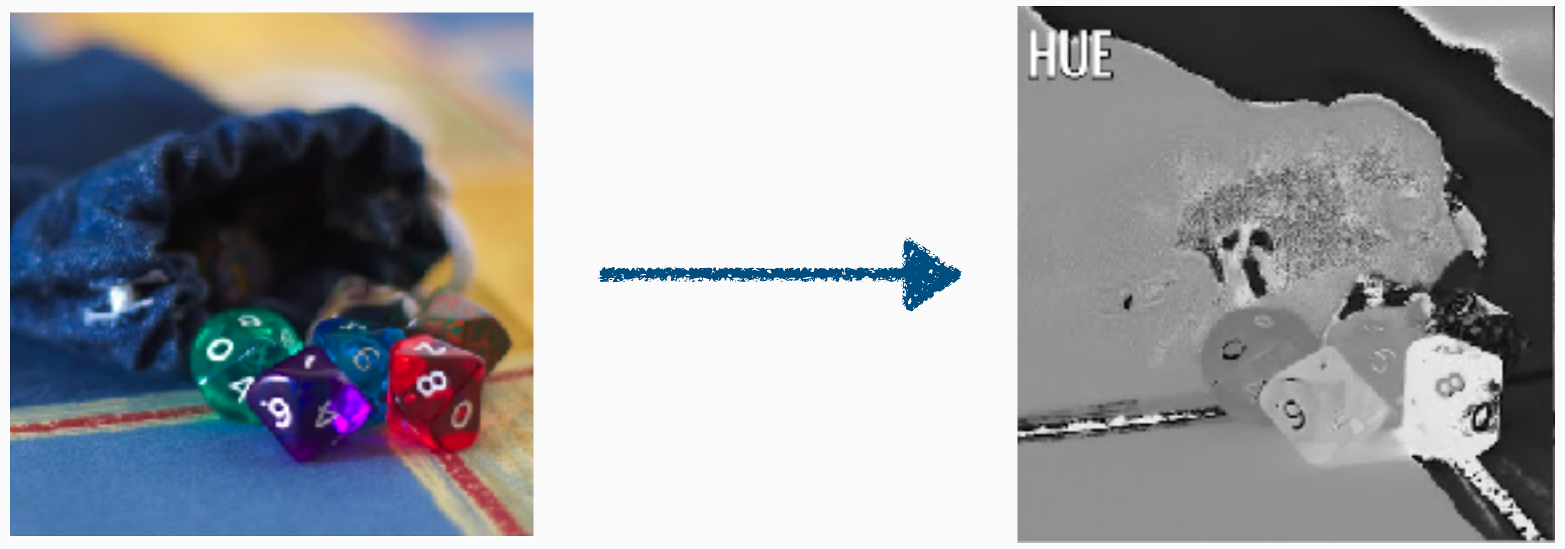


Region Proposal

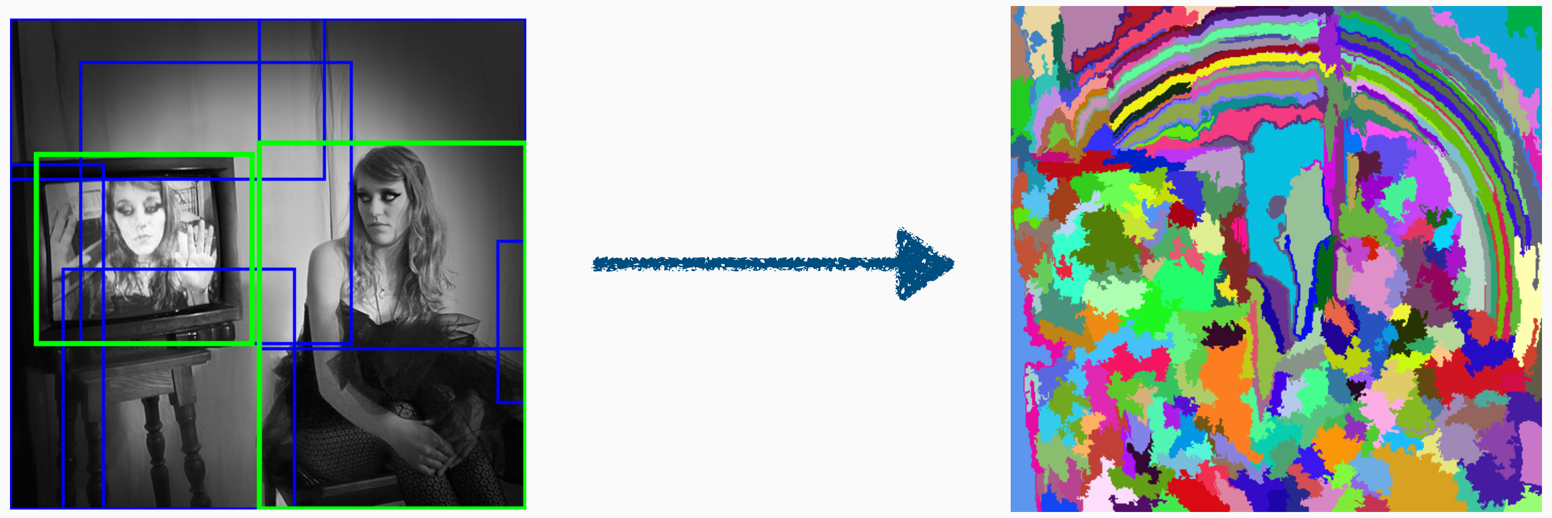
Selective Search for Object Recognition, Uijlings et al., 2013

Originally: hand-crafted proposal mechanisms based on saliency, uniformity of texture, scale, and so on.

Illumination invariant colorspace



Initial region proposal



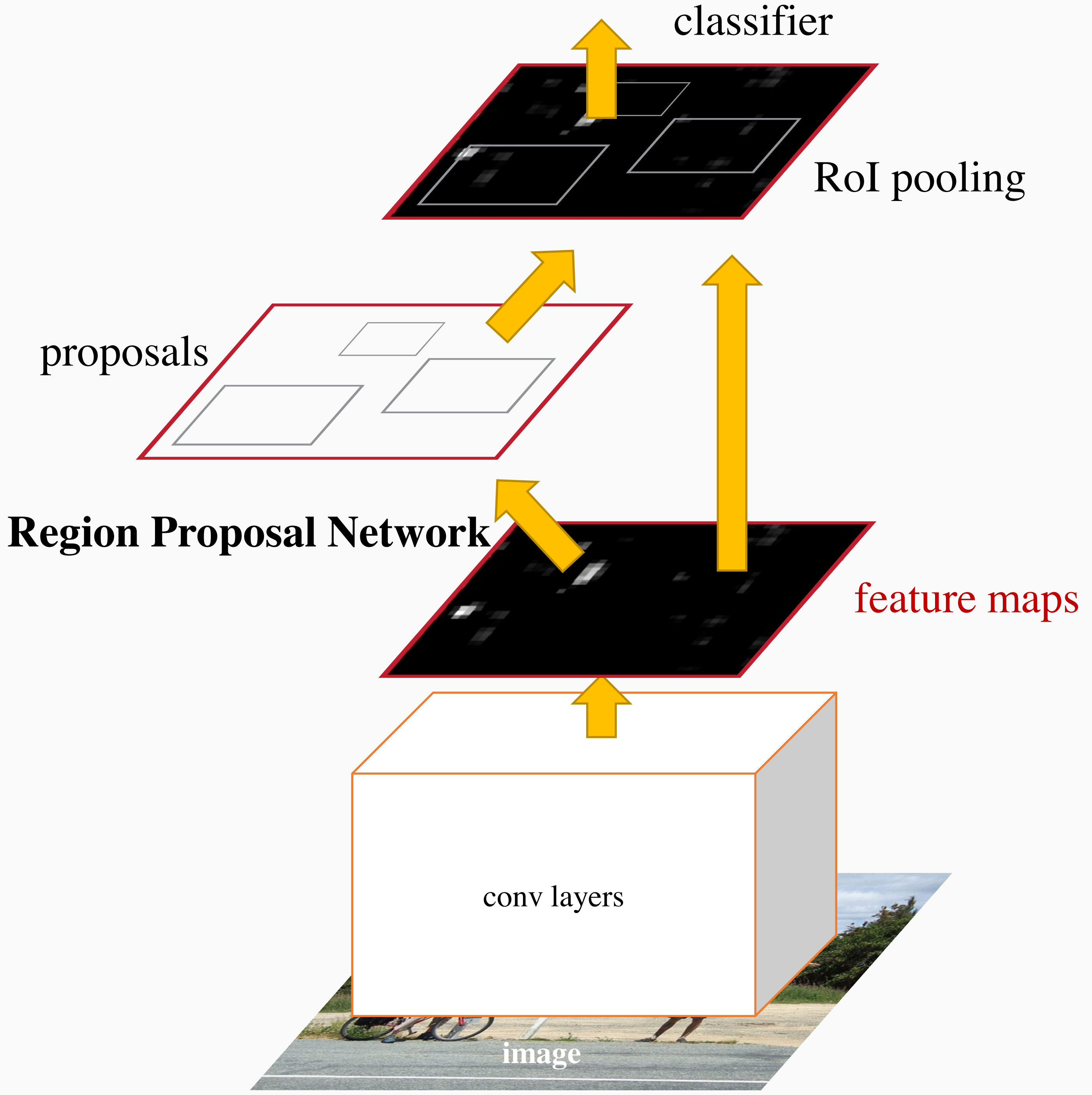
Hierarchical clustering

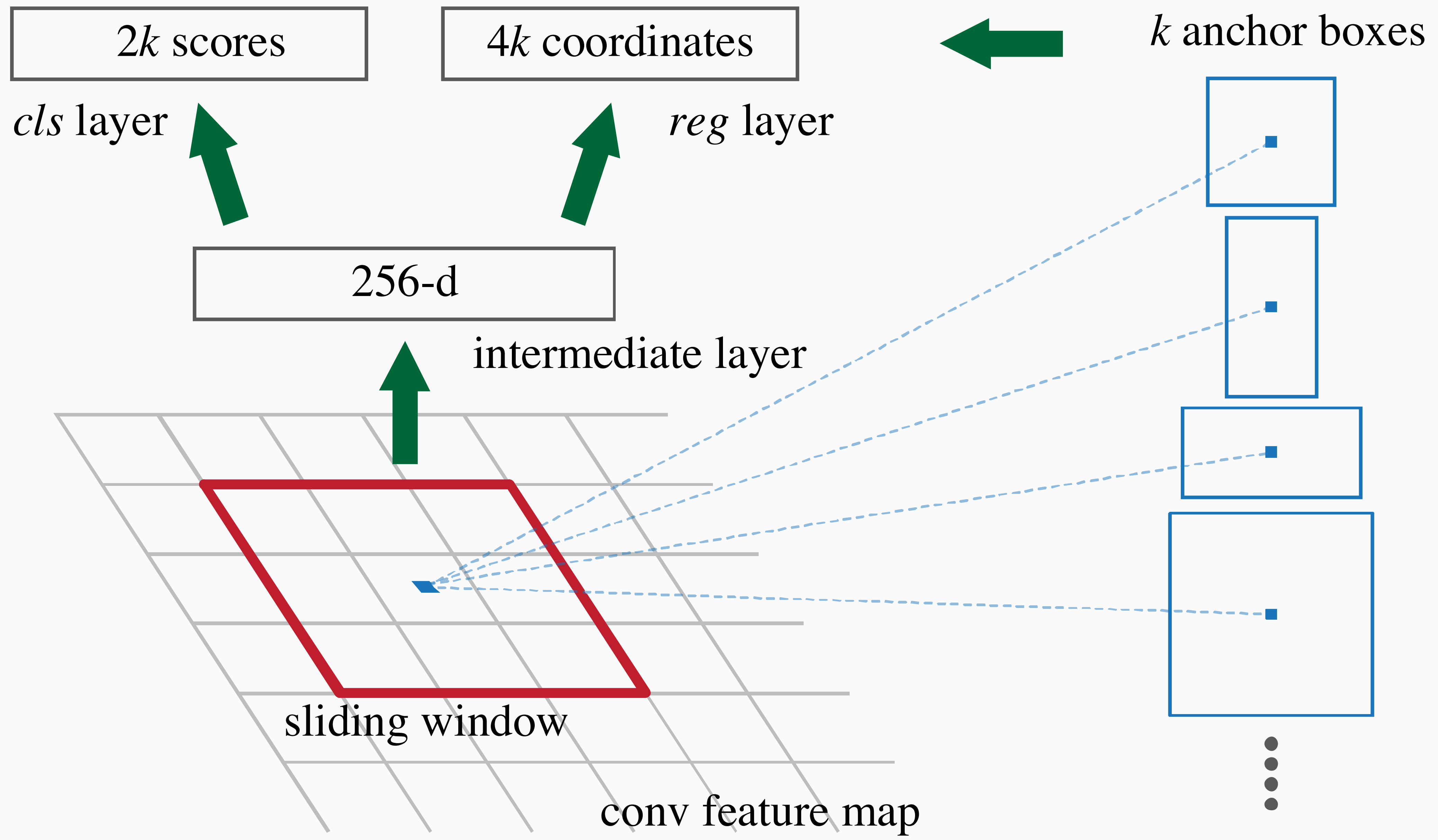
$$s(r_i, r_j) = a_1 s_{colour}(r_i, r_j) + a_2 s_{texture}(r_i, r_j) + a_3 s_{size}(r_i, r_j) + a_4 s_{fill}(r_i, r_j),$$

CNN based region proposal

Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, Ren et al., 2016

Nowadays: The same network does both the region proposal and the classification inside each region



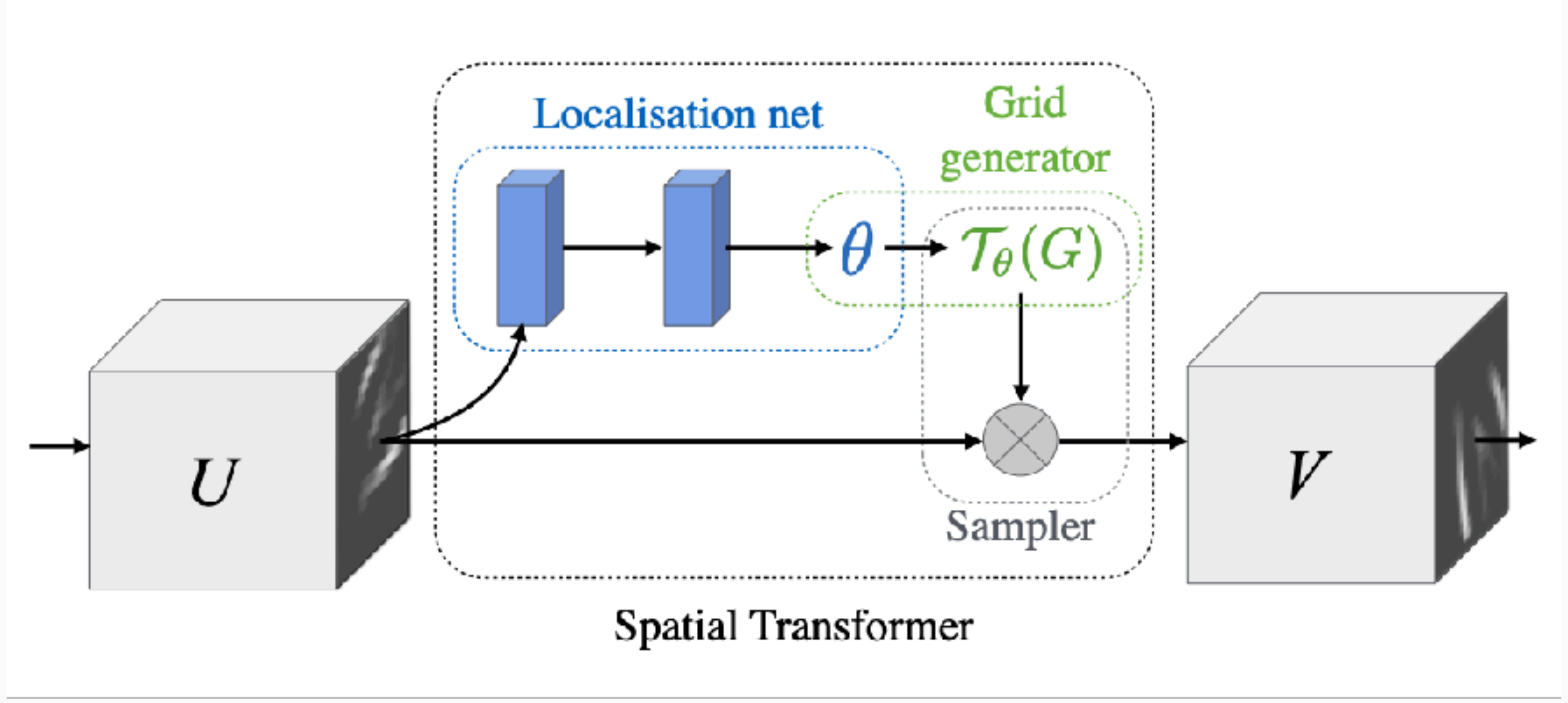


Spatial Transformer Network

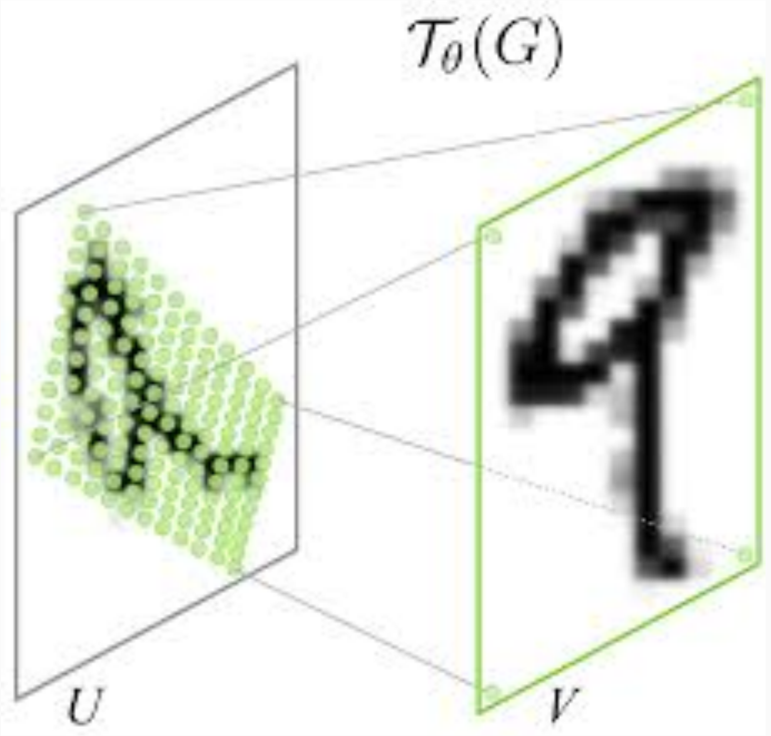
Learning to find and canonize interesting regions of the image

Can we do something more similar to saccades?

Localisation network selects a local reference frame in the image



Transformer resamples using that reference frame



When precision matters

The previous methods find a transformation that approximatively canonize an object. But what if we want a very accurate reference frame?

Images from *Oxford Buildings Dataset*

When precision matters

The previous methods find a transformation that approximatively canonize an object. But what if we want a very accurate reference frame?

Images from *Oxford Buildings Dataset*

When precision matters

The previous methods find a roximatively canonize an object. But what if we want a e frame?



Images from *Oxford Buildings Dataset*

Problems

Reference frame need to be unique and robust.

Due to occlusions, we can only trust local features and need redundancy

Need to be robust to all geometric transformations and small deformations.

Need to be robust to changes of illuminations, shadows, …

SIFT: Scale Invariant Feature Transform

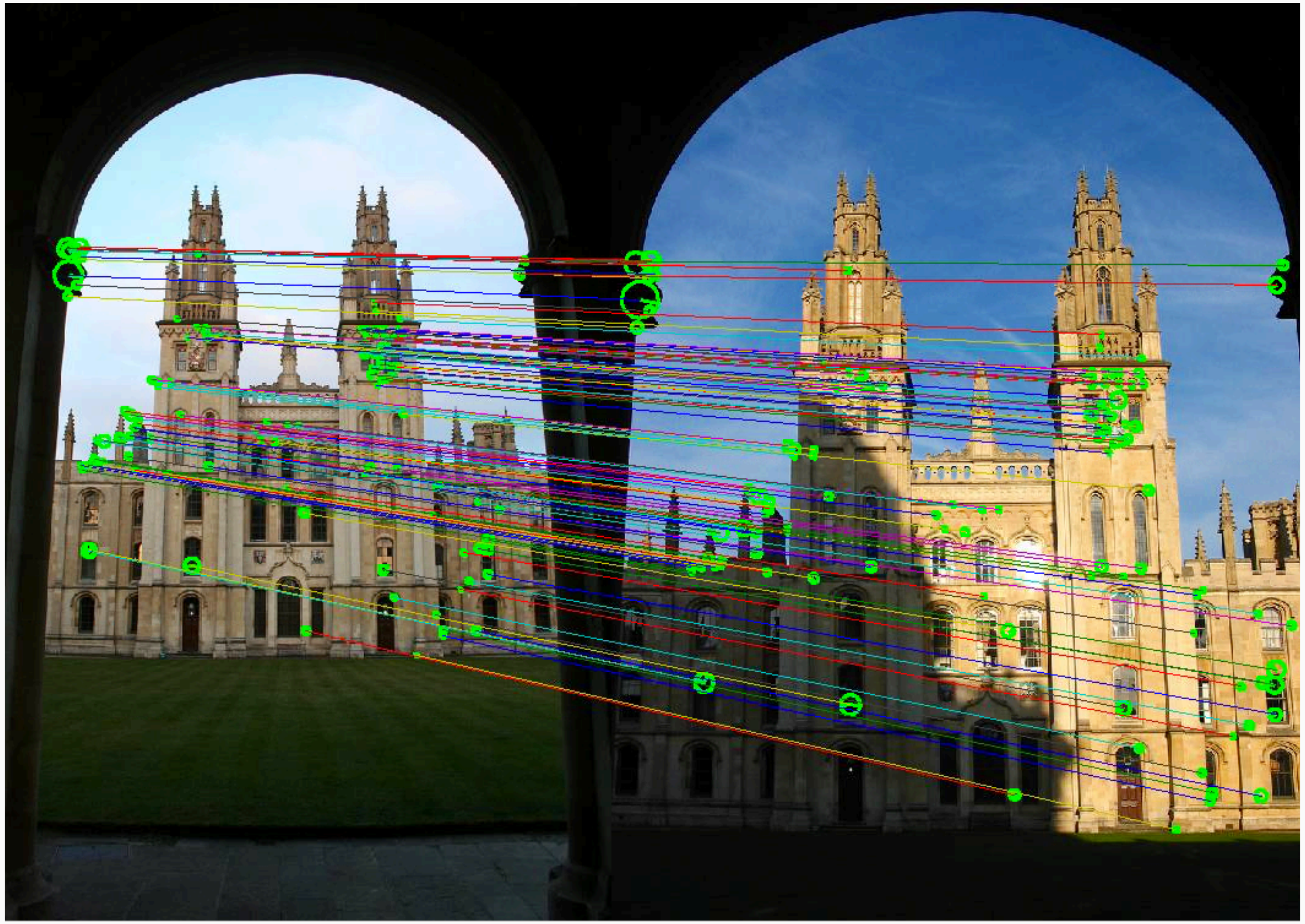
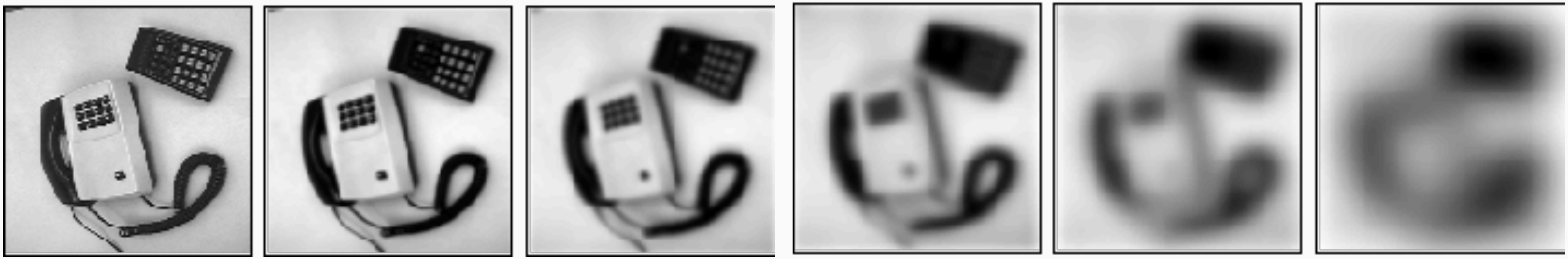


Image from http://www.robots.ox.ac.uk/~vgg/practicals/instance-recognition/index.html

SIFT: Finding the scale

Something for you

Find "interesting points" (*i.e.*, local maxima and minima) at all scales.

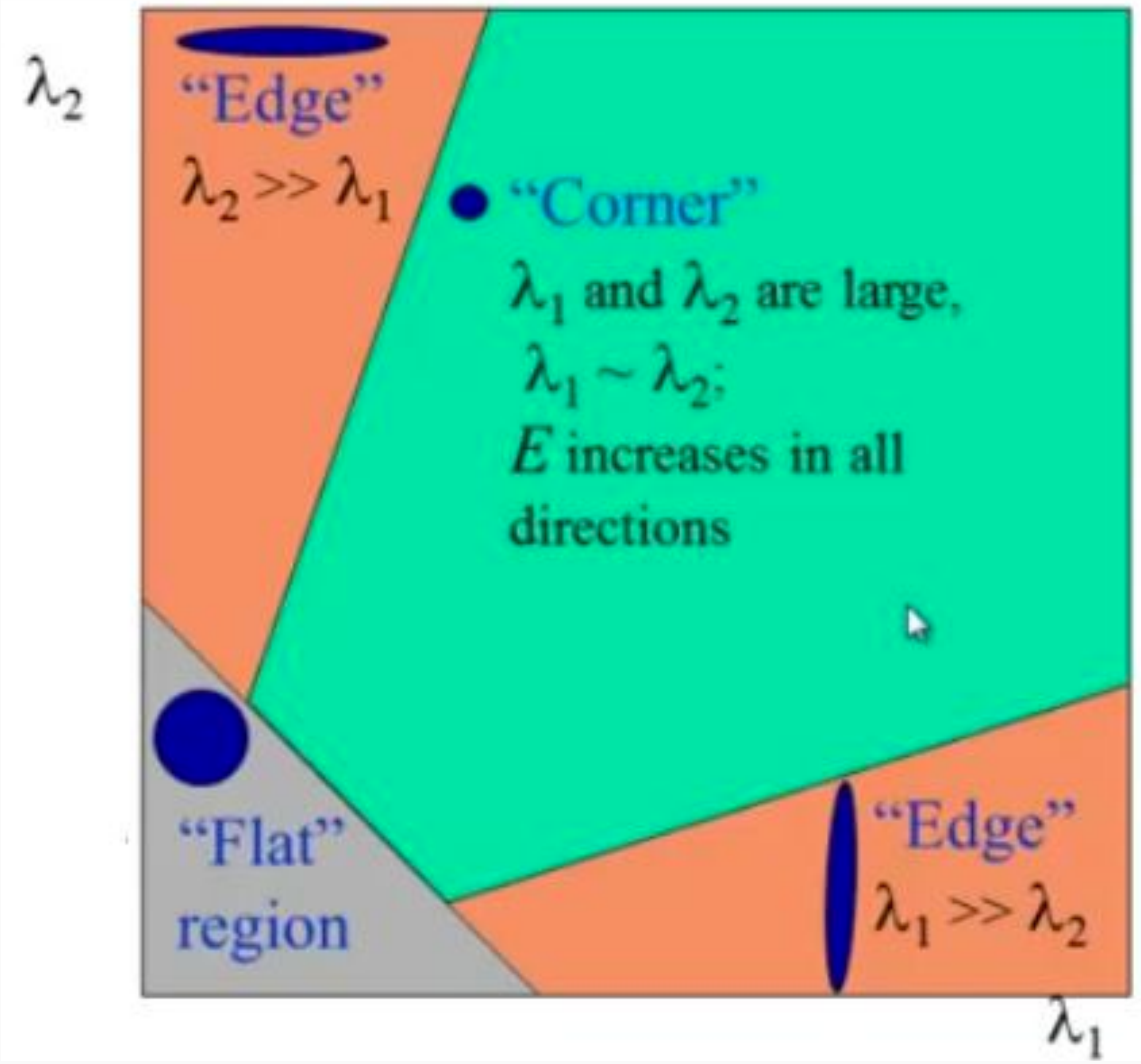


Done by constructing the scale space of the image and finding the first scale at which a local maximum (minimum) stops being a local maximum (minimum).

Harris corner detector

Points along edges are not useful keypoints, as they cannot be localized exactly.

Idea: Compute the Hessian at each interesting point. Consider only the points that have **large eigenvalues of the same magnitude**.



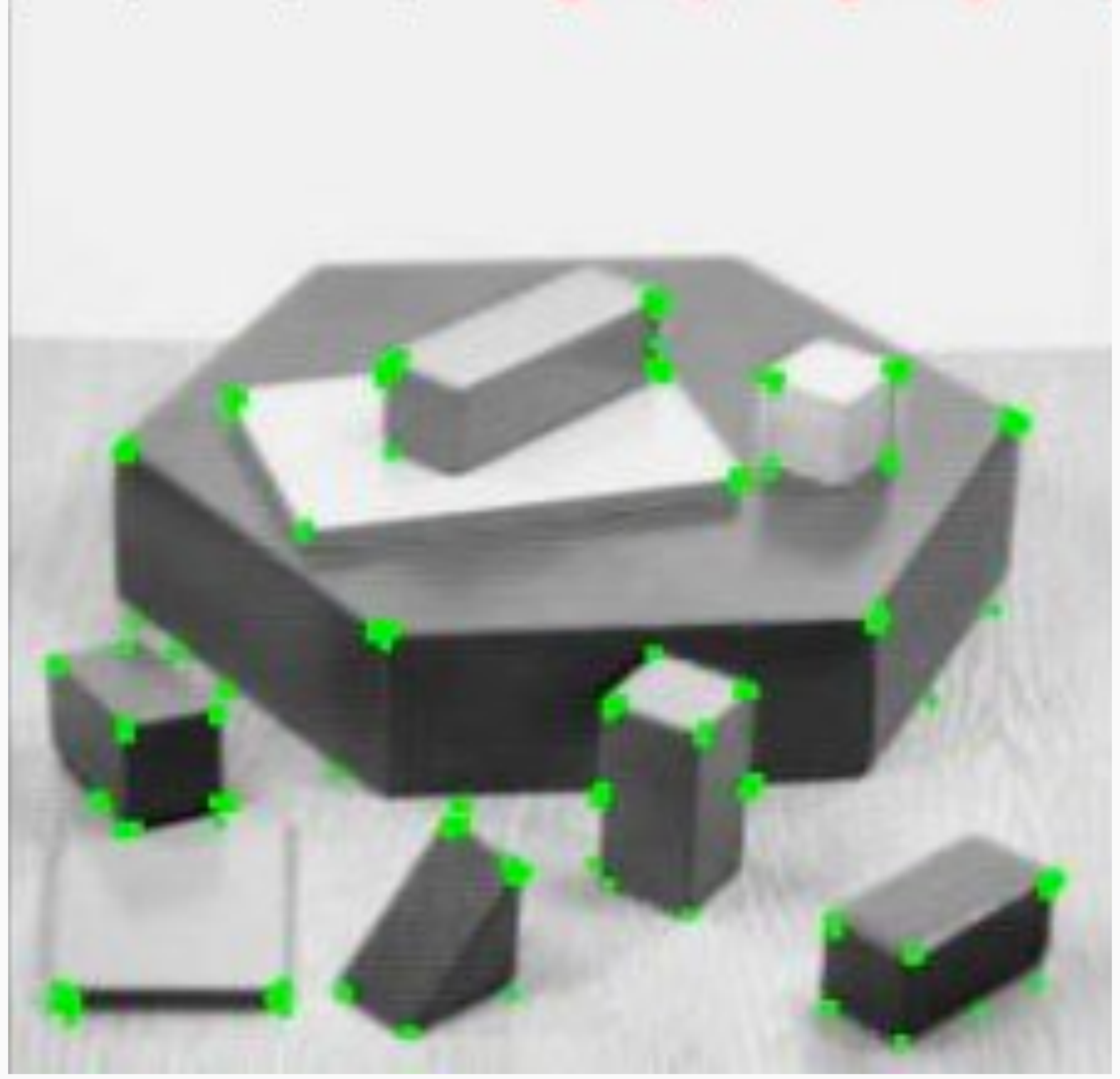
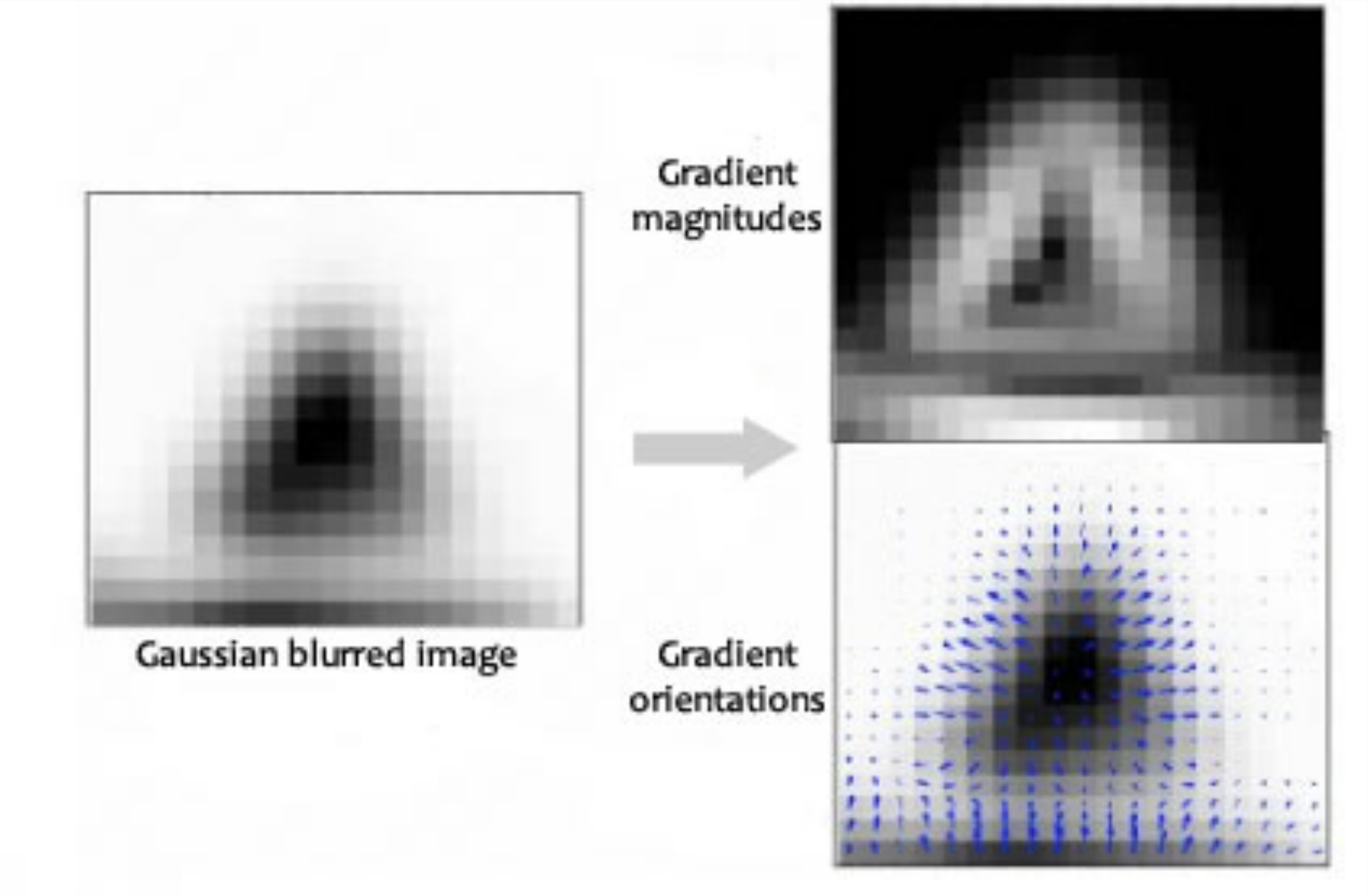
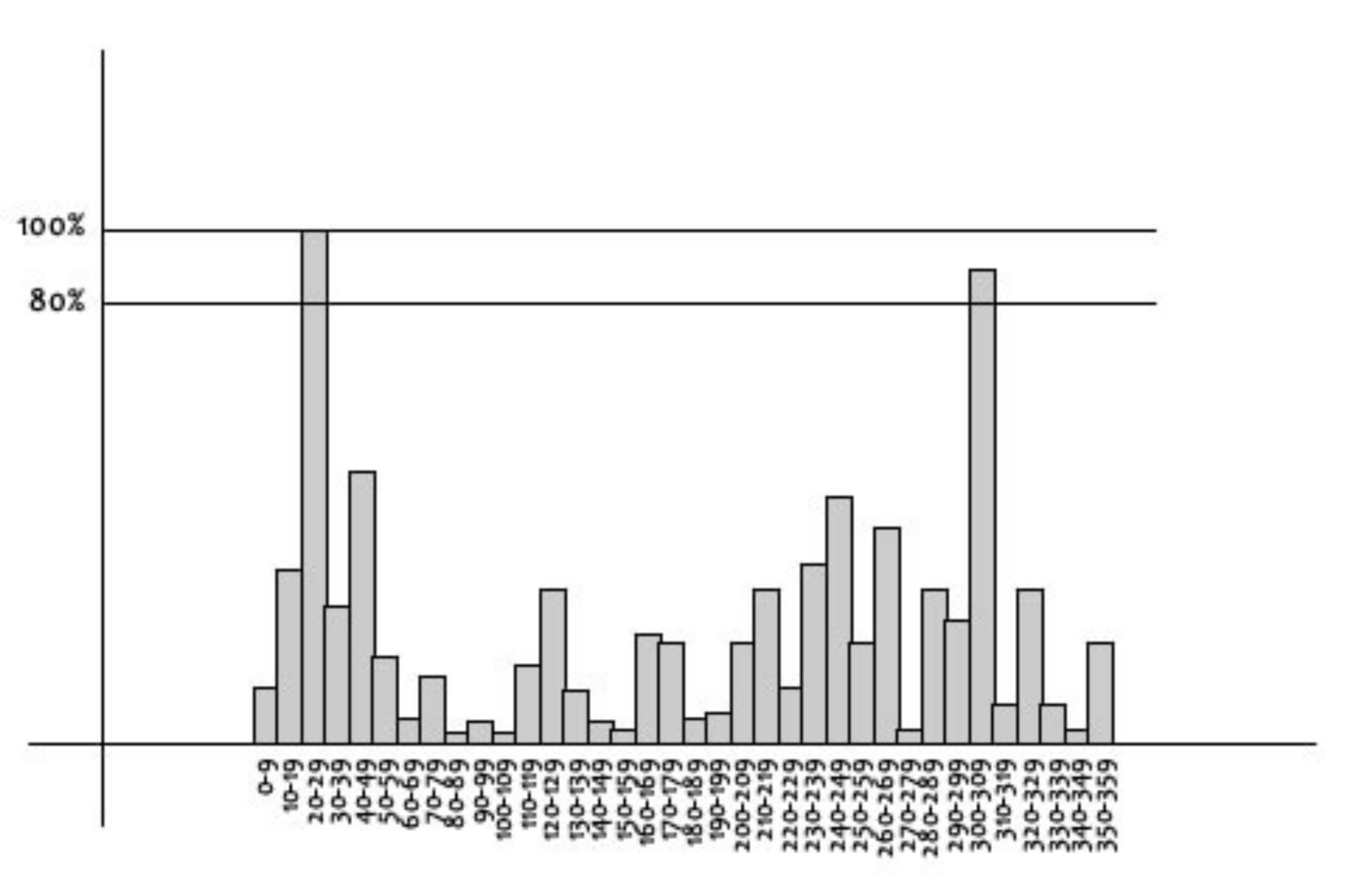


Image from https://docs.opencv.org/3.4.2/dc/d0d/tutorial_py_features_harris.html

Find corner orientation

Decide the orientation of the corner by plotting the histogram of the gradients orientation and find the most frequent orientation.





If multiple orientations are very frequent (> 0.8 * max), select all.

Image from http://aishack.in/tutorials/sift-scale-invariant-feature-transform-keypoint-orientation/

Corner descriptor

Gradient orientation is the only invariant to contrast changes.

Idea: Describe local patch around corner using orientations of the gradients.

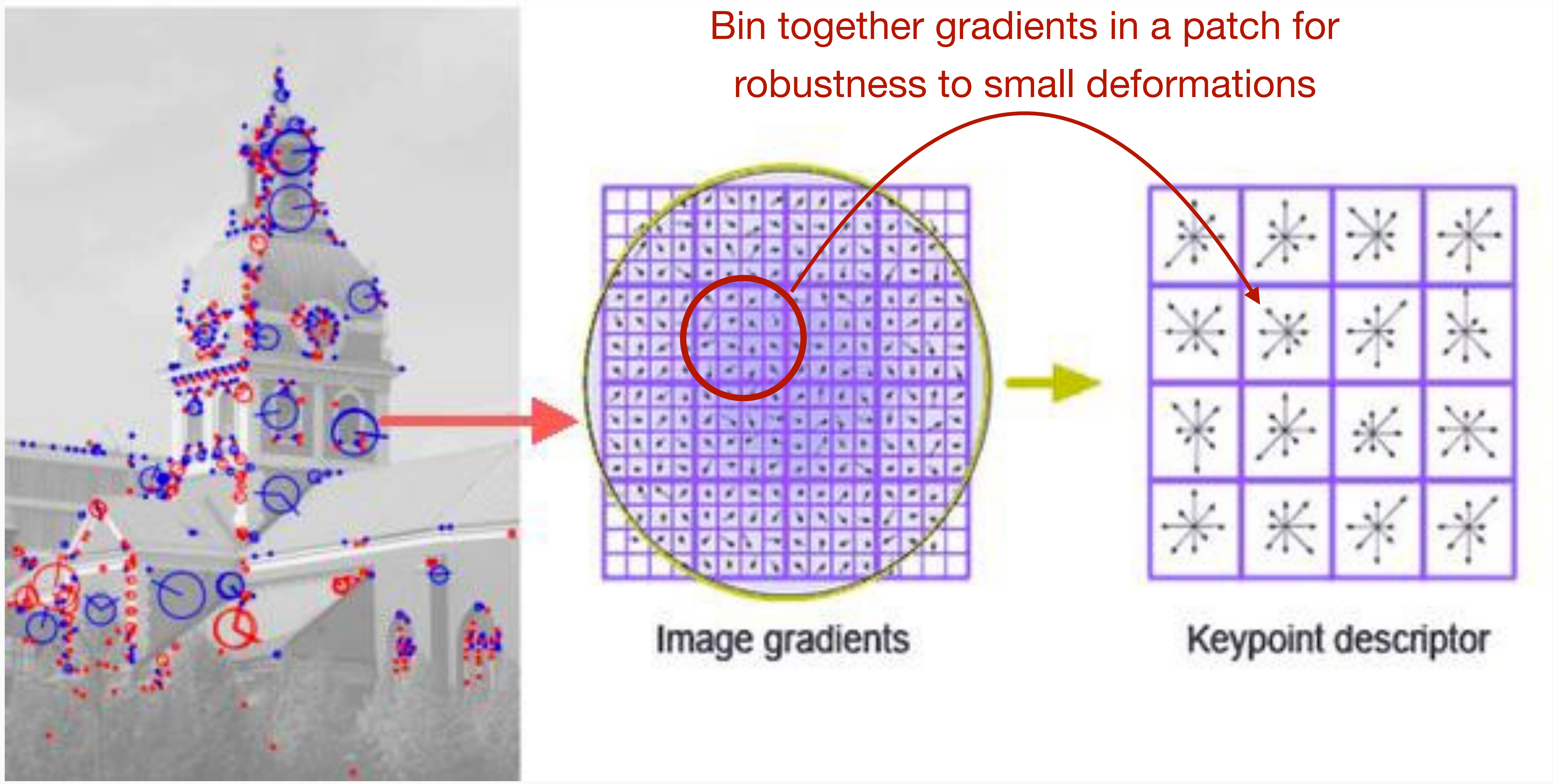
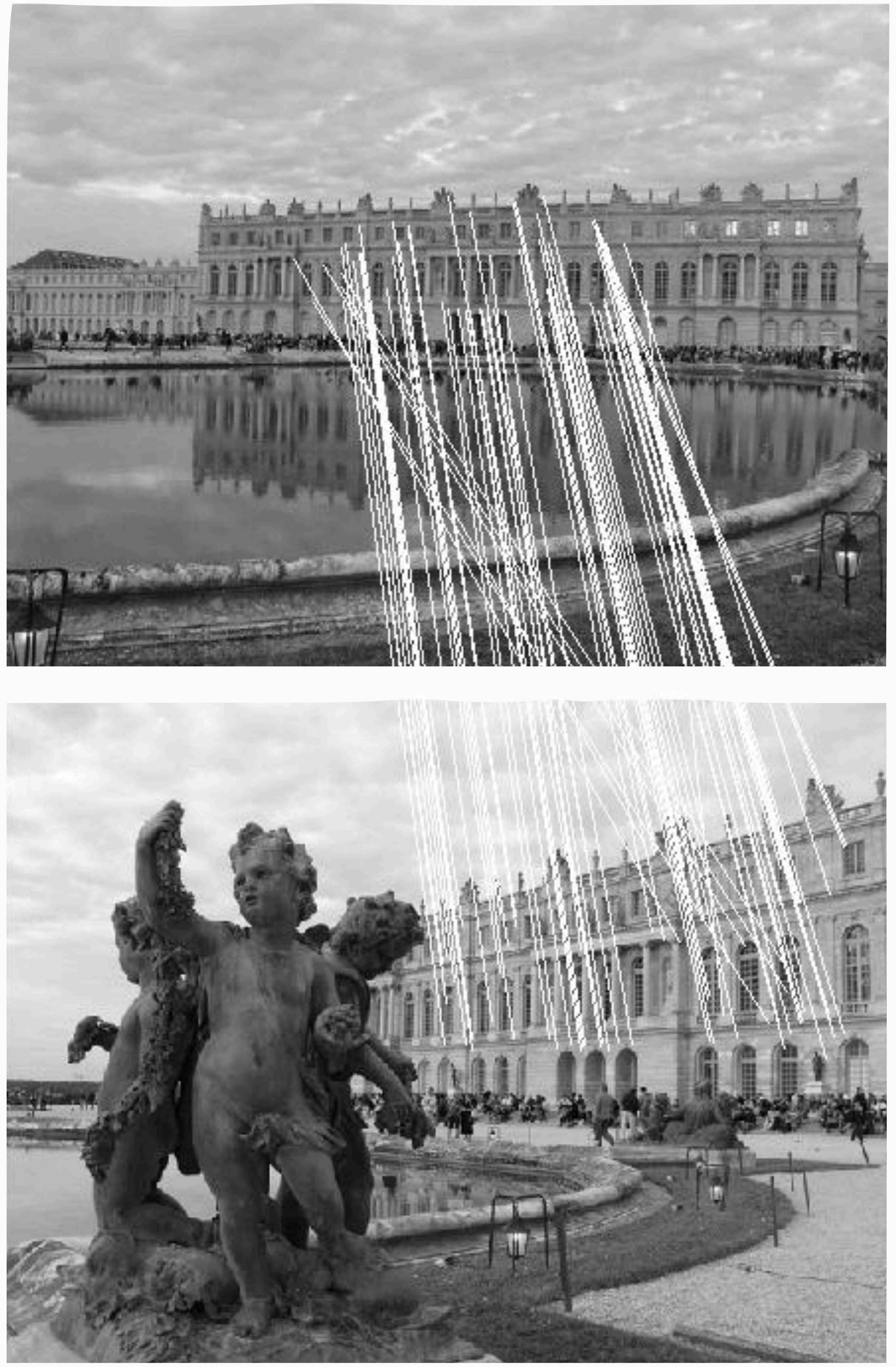


Image from http://aishack.in/tutorials/sift-scale-invariant-feature-transform-keypoint-orientation/

The final algorithm (with refinements)



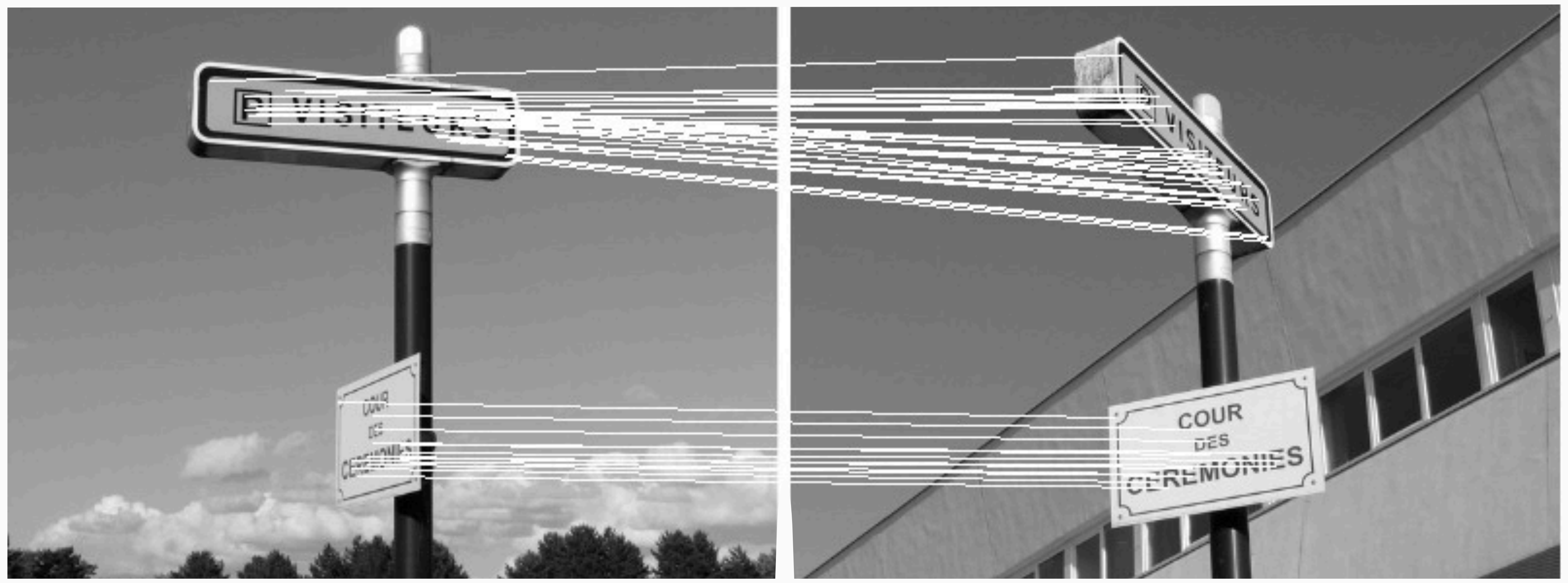


Image from http://www.cmap.polytechnique.fr/~yu/research/ASIFT/demo.html

Feature matching in Visual-Inertial SLAM system

Robust Inference for Visual-Inertial Sensor Fusion, K. Tsotsos et al., 2015

Demo video from https://sites.google.com/site/ktsotsos/visual-inertial-sensor-fusion

Feature matching in Visual-Inertial SLAM system

Robust Inference for Visual-Inertial Sensor Fusion, K. Tsotsos et al., 2015

Demo video from https://sites.google.com/site/ktsotsos/visual-inertial-sensor-fusion

Summary

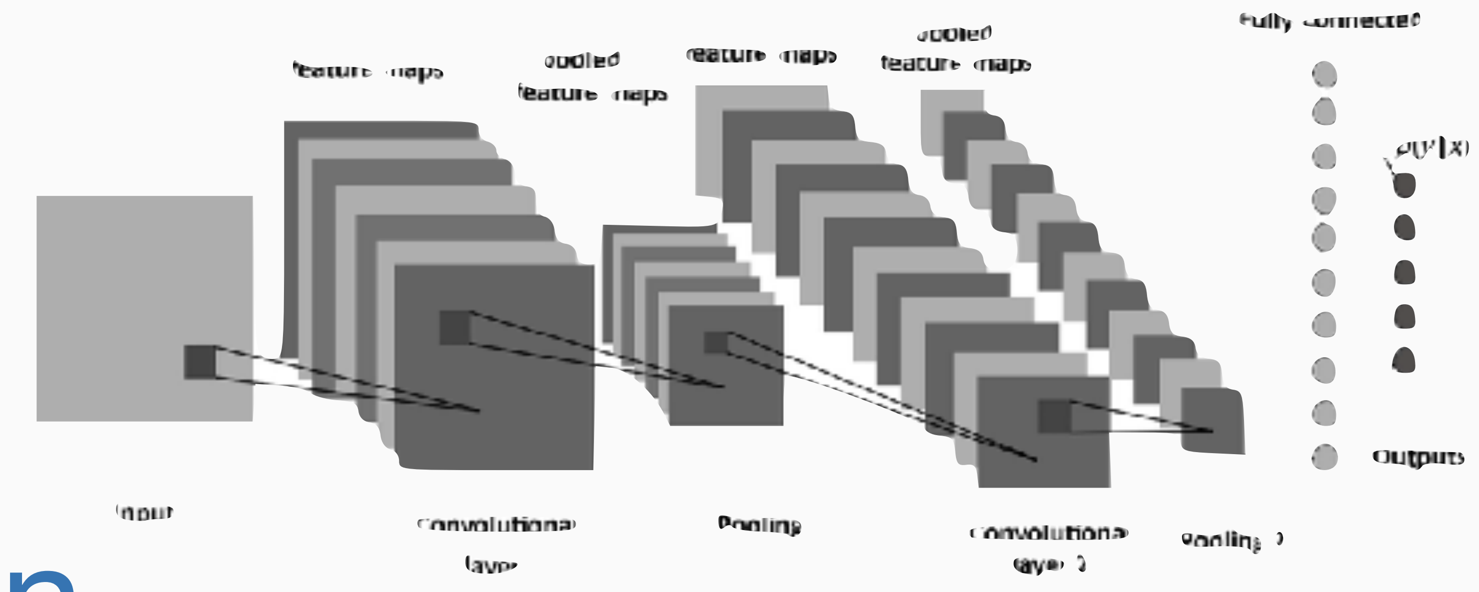
We want something:

- Equivariant to change of scale: search over scale space
- Equivariant to translations: find corners (points in edges and flat region are not localizable exactly)
- Equivariant to rotations: find most frequent gradient orientation
- Invariant to contrast changes: Use gradient orientation to describe patch

Put all this requirements together to get the SIFT descriptor (or one of the many variants: SIFT, ASIFT, DSP-SIFT, SURF, KAZE, AKAZE, ORB, …)

Take-away: a set of corners with an associated description vector is a surprisingly powerful representation for many complex tasks.

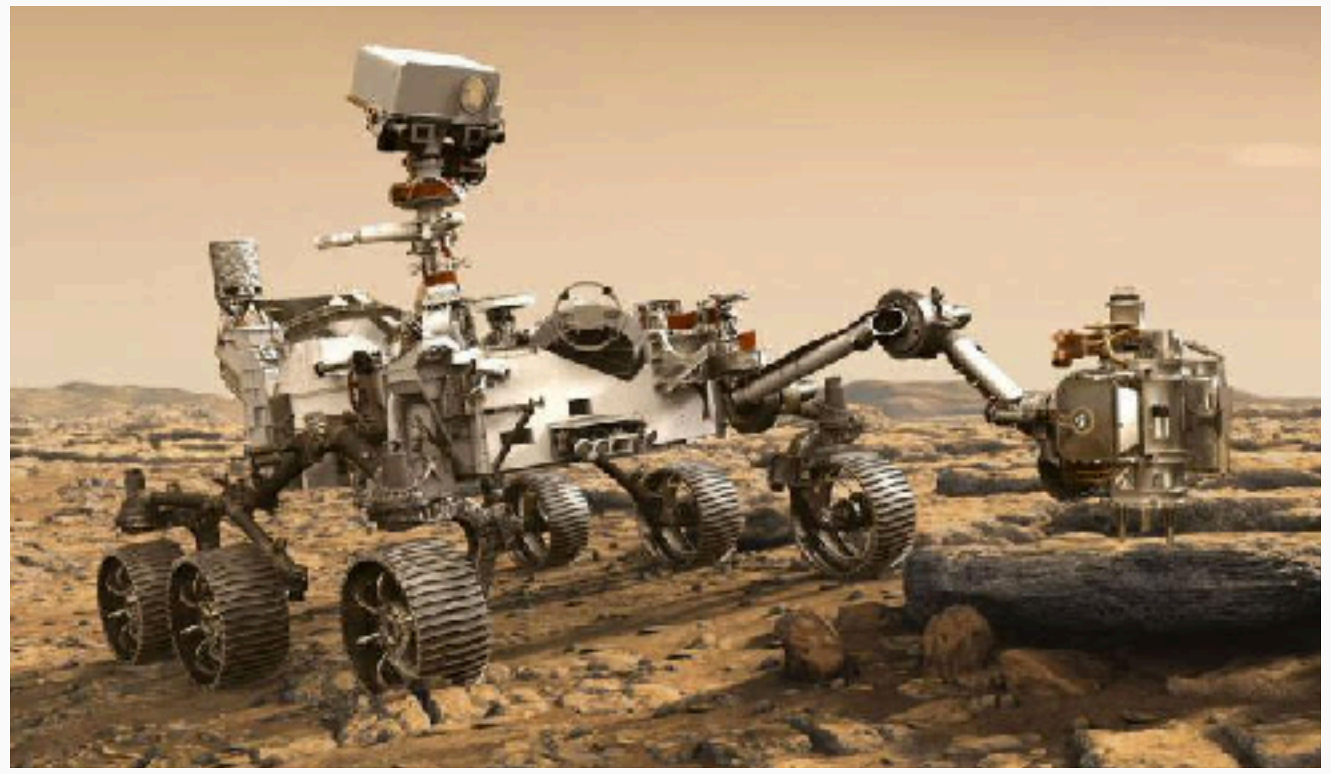
Where are we now



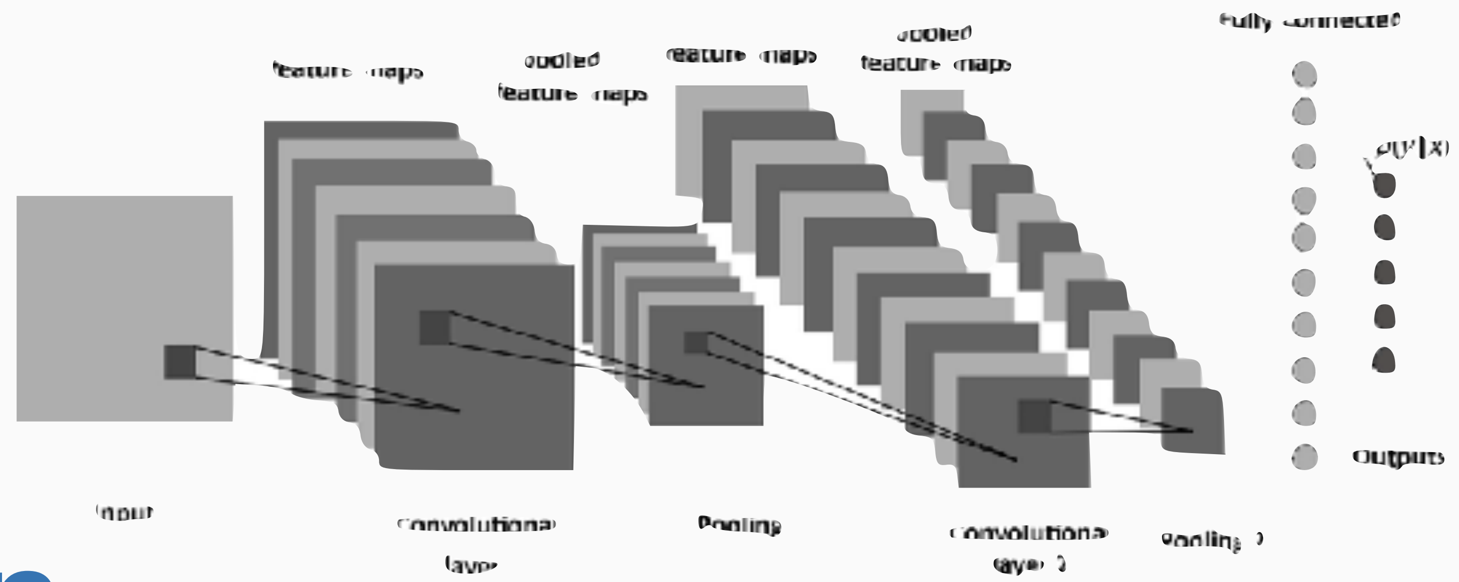
Cognition

Sensing

Action



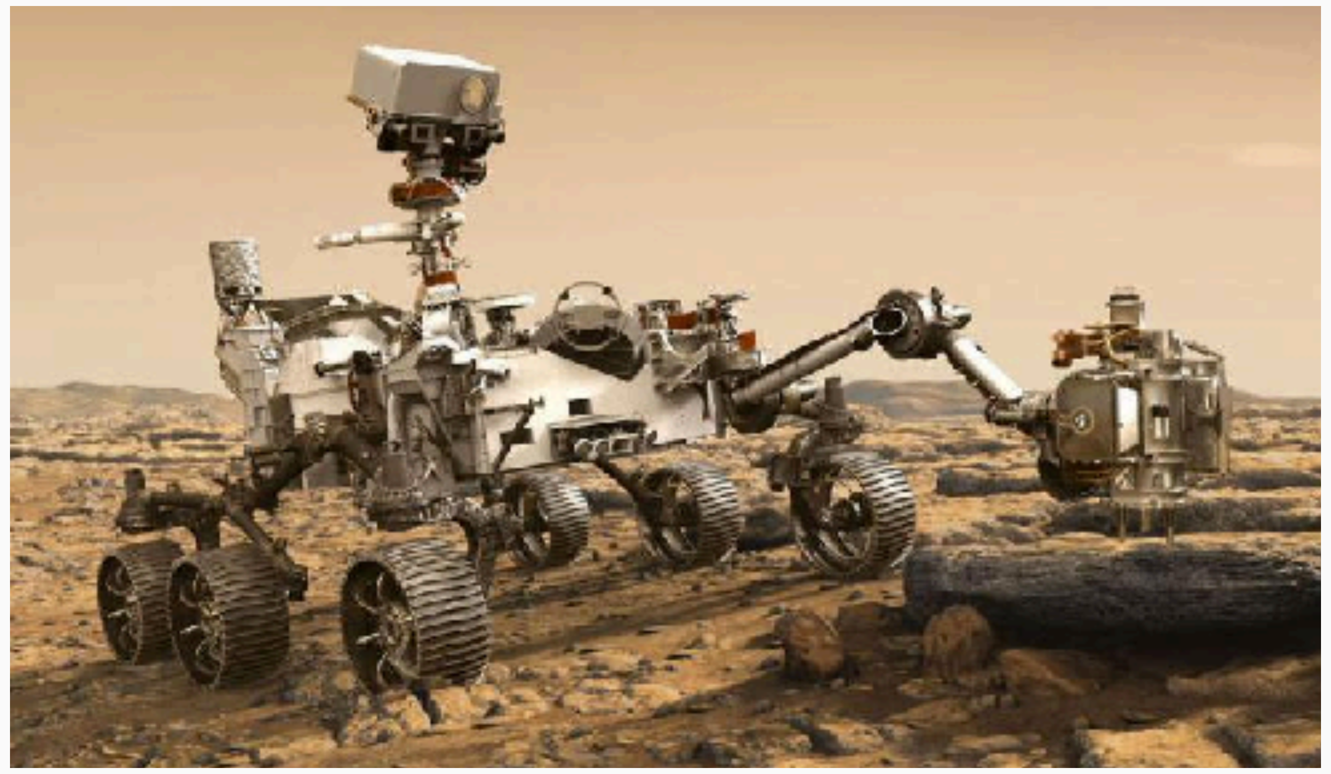
Where are we now



Cognition

Sensing

Action



Invariance to simple geometric nuisances, corner detectors, …