Feature points

D.A. Forsyth, UIUC

Image neighborhoods

- We want to find patches that are “worth representing”
  - to match from image to image
  - to represent textures
  - to represent objects
- Requirements
  - Covariant to translation, rotation, scale
    - i.e. if the image is translated, rotated, scaled, so are the neighborhoods
    - important to ensure that the representation of the patch is stable
  - Localizable in translation, rotation, scale
    - we can estimate the position, orientation and size of the patch
    - and get the answer about right
- Methods exist for richer sets of requirements

Recall: Edges

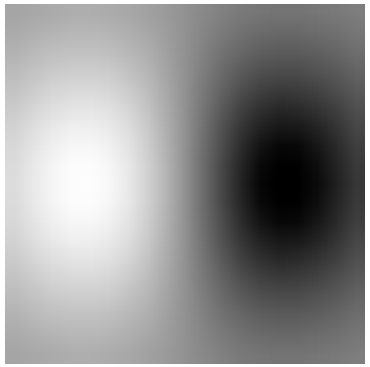
- Idea:
  - points where image value change very sharply are important
    - changes in surface reflectance
    - shadow boundaries
    - outlines
- Finding Edges:
  - Estimate gradient magnitude using appropriate smoothing
  - Mark points where gradient magnitude is
    - Locally biggest and
    - big

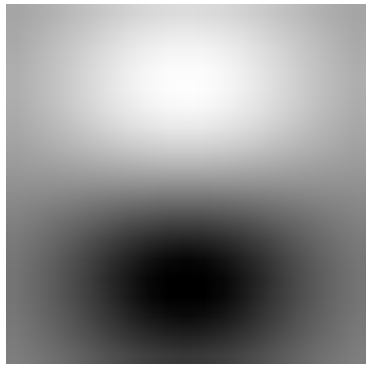
Recall: Smoothed gradients

- Fact: These two are the same
  - Smooth, then differentiate
  - Filter with derivative of Gaussian

$$\frac{\partial \left(G_\sigma * * I\right)}{\partial x} = \left(\frac{\partial G_\sigma}{\partial x}\right) * * I$$

- Exploit:
  - Filter image with derivative of Gaussian filters to get smoothed gradient





Edge Maps Depend on Shading

- If the image is brighter (resp. darker)
  - because the camera gain is higher (resp. lower)
  - because there is more (resp. less) light
  - because the pixel values got multiplied by a constant
- Then the gradient magnitude is bigger (resp. smaller)

- So scaling image brightness changes the edge map
  - because some magnitudes will go above (resp. below) the test threshold

- Edge maps differ for brighter/darker copies of a picture

Orientations - I

- Gradient magnitude is affected by illumination changes
  - but gradient direction isn't

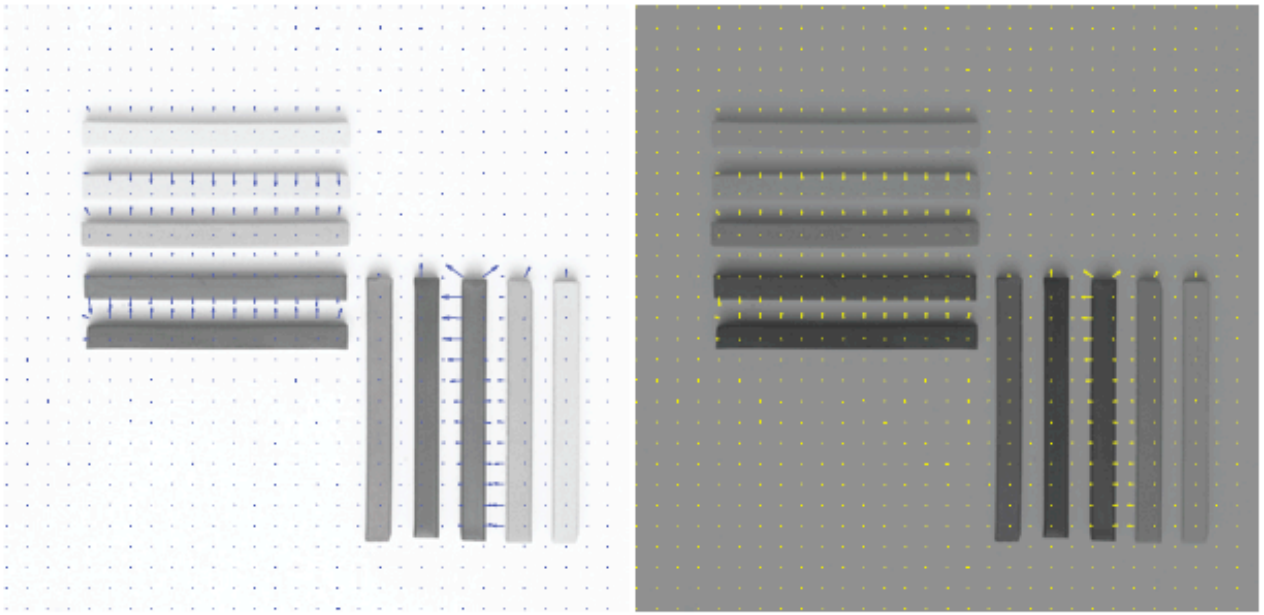


FIGURE 5.7: The magnitude of the image gradient changes when one increases or decreases the intensity. The orientation of the image gradient does not change; we have plotted every 10th orientation arrow, to make the figure easier to read. Note how the directions of the gradient arrows are fixed, whereas the size changes. *Philip Gatward © Dorling Kindersley, used with permission.*

Orientations - II

- Notice larger gradients are “better”
  - we know the orientation better; associated image points “more interesting”

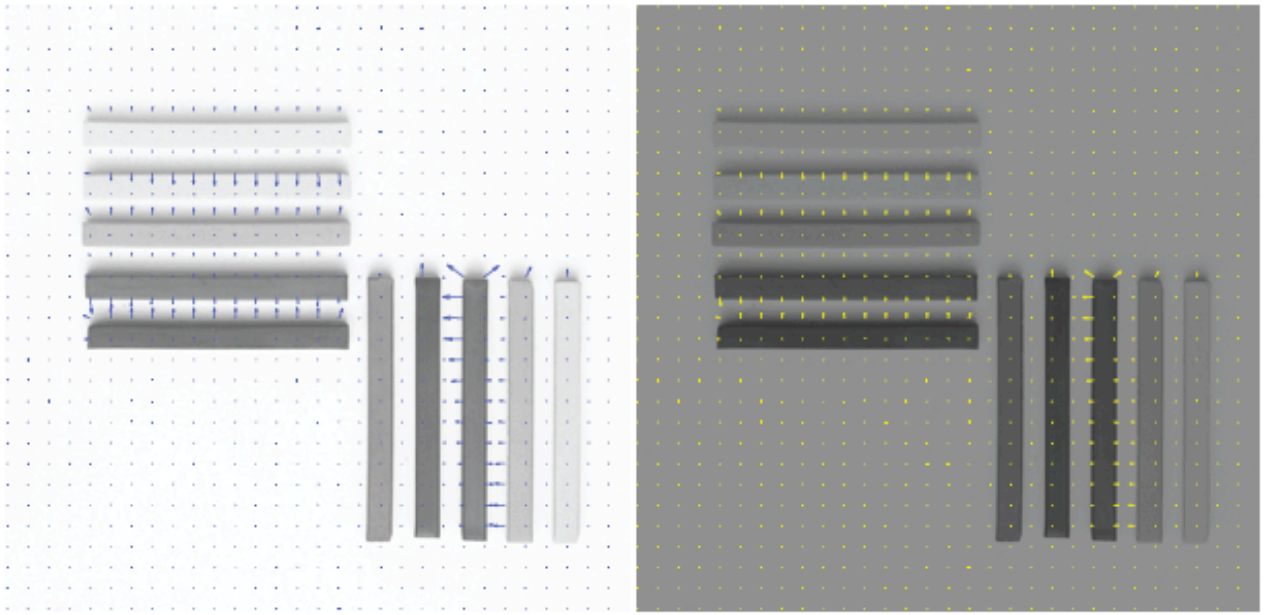


FIGURE 5.7: The magnitude of the image gradient changes when one increases or decreases the intensity. The orientation of the image gradient does not change; we have plotted every 10th orientation arrow, to make the figure easier to read. Note how the directions of the gradient arrows are fixed, whereas the size changes. *Philip Gatward © Dorling Kindersley, used with permission.*

Finding image neighborhoods - I

- Corner finding strategy
  - Find centers
  - At each center, estimate scale
  - Now from center, scale, estimate orientation

Harris corner detector - I

- Good corners
  - High contrast
  - Sharp change in edge orientation
- Image features at good corners
  - Large gradients
  - That change direction sharply
- Compute matrix H by summing over window

$$\mathcal{H} = \sum_{window} \{(\nabla I)(\nabla I)^T\}$$

$$\approx \sum_{window} \left\{ \begin{array}{ll} (\frac{\partial G_\sigma}{\partial x} * *\mathcal{I})(\frac{\partial G_\sigma}{\partial x} * *\mathcal{I}) & (\frac{\partial G_\sigma}{\partial x} * *\mathcal{I})(\frac{\partial G_\sigma}{\partial y} * *\mathcal{I}) \\ (\frac{\partial G_\sigma}{\partial x} * *\mathcal{I})(\frac{\partial G_\sigma}{\partial y} * *\mathcal{I}) & (\frac{\partial G_\sigma}{\partial y} * *\mathcal{I})(\frac{\partial G_\sigma}{\partial y} * *\mathcal{I}) \end{array} \right\}$$

Harris corner detector - II

- Matrix H

$$\mathcal{H} = \sum_{window} \{(\nabla I)(\nabla I)^T\}$$

$$\approx \sum_{window} \left\{ \begin{array}{ll} (\frac{\partial G_\sigma}{\partial x} * *\mathcal{I})(\frac{\partial G_\sigma}{\partial x} * *\mathcal{I}) & (\frac{\partial G_\sigma}{\partial x} * *\mathcal{I})(\frac{\partial G_\sigma}{\partial y} * *\mathcal{I}) \\ (\frac{\partial G_\sigma}{\partial x} * *\mathcal{I})(\frac{\partial G_\sigma}{\partial y} * *\mathcal{I}) & (\frac{\partial G_\sigma}{\partial y} * *\mathcal{I})(\frac{\partial G_\sigma}{\partial y} * *\mathcal{I}) \end{array} \right\}$$

  - Will have two large eigenvalues if the window has
    - Large gradients
    - That change direction sharply
- Look for big values of

$$\det(\mathcal{H}) - k(\frac{\text{trace}(\mathcal{H})}{2})^2$$

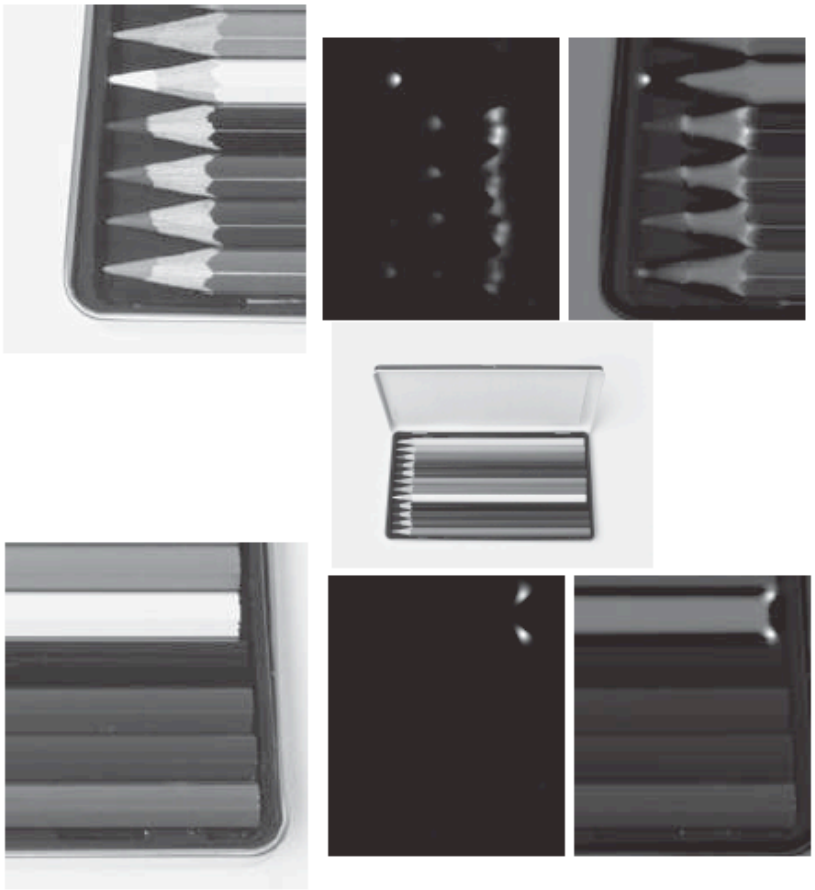


FIGURE 5.10: The response of the Harris corner detector visualized for two detail regions of an image of a box of colored pencils (**center**). **Top left**, a detail from the pencil points; **top center**, the response of the Harris corner detector, where more positive values are lighter. The **top right** shows these overlaid on the original image. To overlay this map, we added the images, so that areas where the overlap is notably dark come from places where the Harris statistic is negative (which means that one eigenvalue of $\mathcal{H}$ is large, the other small). Note that the detector is affected by contrast, so that, for example, the point of the mid-gray pencil at the top of this figure generates a very strong corner response, but the points of the darker pencils do not, because they have little contrast with the tray. For the darker pencils, the strong, contrasty corners occur where the lead of the pencil meets the wood. The **bottom** sequence shows corners for a detail of pencil ends. Notice that responses are quite local, and there are a relatively small number of very strong corners. *Steve Gorton © Dorling Kindersley, used with permission.*

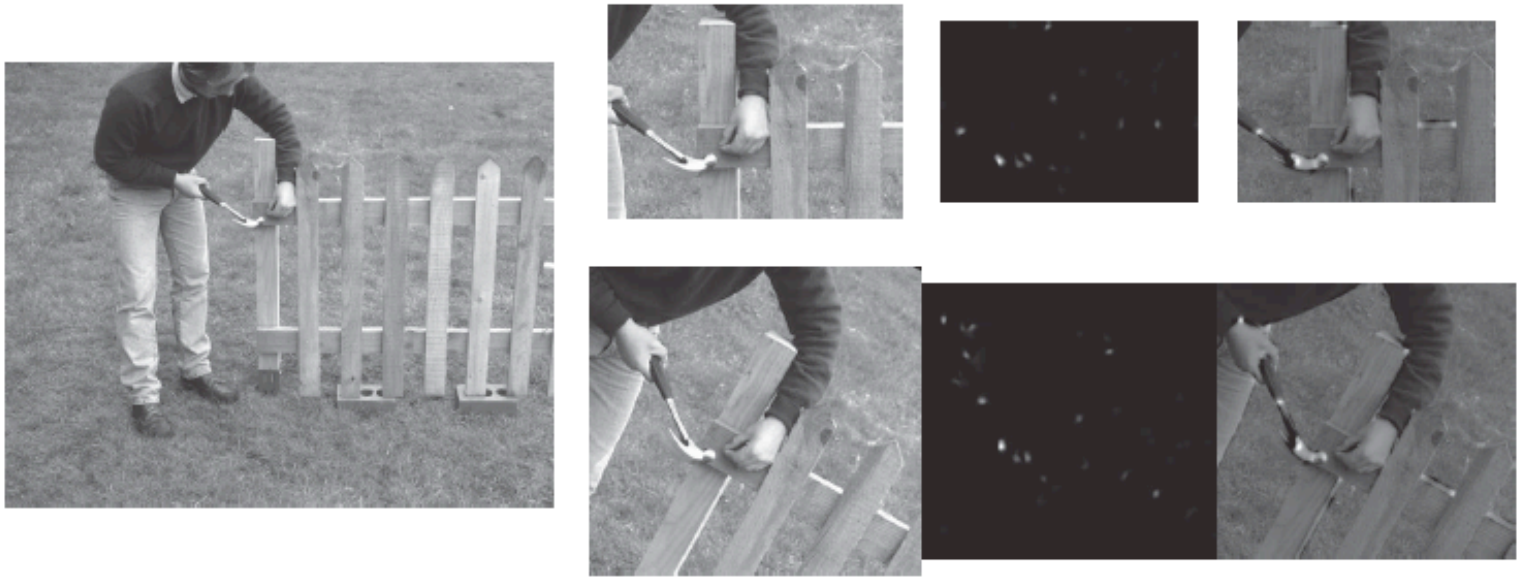


FIGURE 5.11: The response of the Harris corner detector is unaffected by rotation and translation. The **top row** shows the response of the detector on a detail of the image on the **far left**. The **bottom row** shows the response of the detector on a corresponding detail from a rotated version of the image. For each row, we show the detail window (**left**); the response of the Harris corner detector, where more positive values are lighter (**center**); and the responses overlaid on the image (**right**). Notice that responses are quite local, and there are a relatively small number of very strong corners. To overlay this map, we added the images, so that areas where the overlap is notably dark come from places where the Harris statistic is negative (which means that one eigenvalue of $\mathcal{H}$ is large, the other small). The arm and hammer in the top row match those in the bottom row; notice how well the maps of Harris corner detector responses match, too.

Estimating scale - I

- Assume we have detected a corner
- How big is the neighborhood?
- Use Laplacian of Gaussian filter
  - Details on next slide
  - Kernel looks like fuzzy dark blob on pale light foreground
  - Scale (sigma) of Gaussian gives size of dark, light blob
- Strategy
  - Apply Laplacian of Gaussian at different scales at corner
    - response is a function of scale
  - Choose the scale that gives the largest response
    - the scale at which the neighborhood looks “most like” a fuzzy blob
  - This is covariant (see text)

Estimating scale - II

- Laplacian of a function

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- Gaussian

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{\left(\frac{-x^2-y^2}{2\sigma^2}\right)}$$

- So Laplacian of Gaussian

$$\nabla^2 G_\sigma(x, y) = \left(\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4}\right) G_\sigma(x, y)$$

- Convolve with image

$$\nabla^2_\sigma \mathcal{I}(x, y) = \left(\nabla^2 G_\sigma(x, y)\right) * * \mathcal{I}(x, y)$$

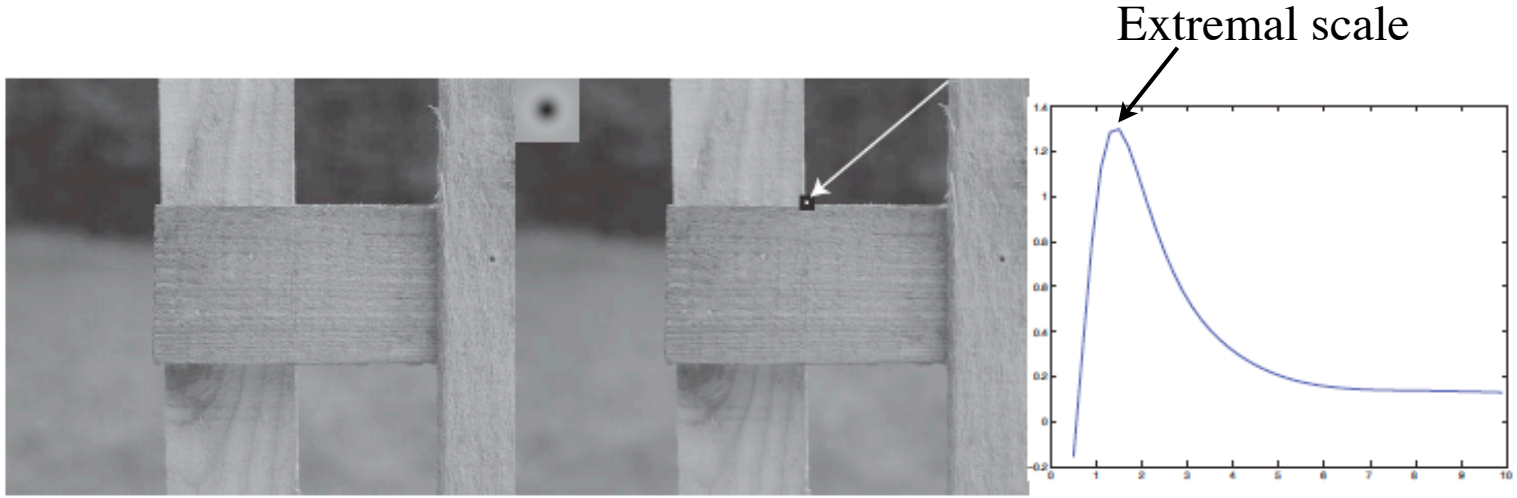


FIGURE 5.12: The scale of a neighborhood around a corner can be estimated by finding a local extremum, in *scale* of the response at that point to a smoothed Laplacian of Gaussian kernel. On the **left**, a detail of a piece of fencing. In the center, a corner identified by an arrow (which points to the corner, given by a white spot surrounded by a black ring). *Overlaid* on this image is a Laplacian of Gaussian kernel, in the **top right** corner; dark values are negative, mid gray is zero, and light values are positive. Notice that, using the reasoning of Section 4.5, this filter will give a strong positive response for a dark blob on a light background, and a strong negative response for a light blob on a dark background, so by searching for the strongest response at this point as a function of scale, we are looking for the size of the best-fitting blob. On the **right**, the response of a Laplacian of Gaussian *at the location of the corner*, as a function of the smoothing parameter (which is plotted in pixels). There is one extremal scale, at approximately 2 pixels. This means that there is one scale at which the image neighborhood looks most like a blob (some corners have more than one scale). © *Dorling Kindersley, used with permission.*

Estimating orientation of neighborhood

- Within neighborhood, estimate image orientations
  - Trick: use a smoothing scale that is a function of neighborhood size
- Form a histogram of orientations
  - Weighting by distance to center, or unweighted
- Choose the orientation with the maximum count
  - if there are two or more
    - make copies of the neighborhood
    - each has one of the peak count orientations

Assume a fixed scale parameter k
Apply a corner detector to the image $\mathcal{I}$
Initialize a list of patches
For each corner detected
  Write (x_c, y_c) for the location of the corner
  Compute the radius r for the patch at (x_c, y_c) as

$$r(x_c, y_c) = \underset{\sigma}{\text{argmax}} \nabla^2_\sigma \mathcal{I}(x_c, y_c)$$

  by computing $\nabla^2_\sigma \mathcal{I}(x_c, y_c)$ for a variety of values of σ, interpolating these values, and maximizing
  Compute an orientation histogram $H(\theta)$ for gradient orientations within a radius kr of (x_c, y_c).
  Compute the orientation of the patch θ_p as

$$\theta_p = \underset{\theta}{\text{argmax}} H(\theta).$$

  If there is more than one theta that maximizes this histogram, make one copy of the patch for each.
  Attach (x_c, y_c, r, θ_p) to the list of patches for each copy

Algorithm 5.2: Obtaining Location, Radius and Orientation of Pattern Elements Using a Corner Detector.

Describing Neighborhoods

- Alternative: SIFT features
  - SIFT=Scale Invariant Feature Transform
  - Very strong record of effectiveness in matching applications
    - use orientations to suppress intensity change effects
    - use histograms so neighborhood need not be exactly localized
    - weight large gradients higher than small gradients
  - Weighting processes are different
  - SIFT features behave very well using nearest neighbors matching
    - i.e. the nearest neighbor to a query patch is usually a matching patch

SIFT Features

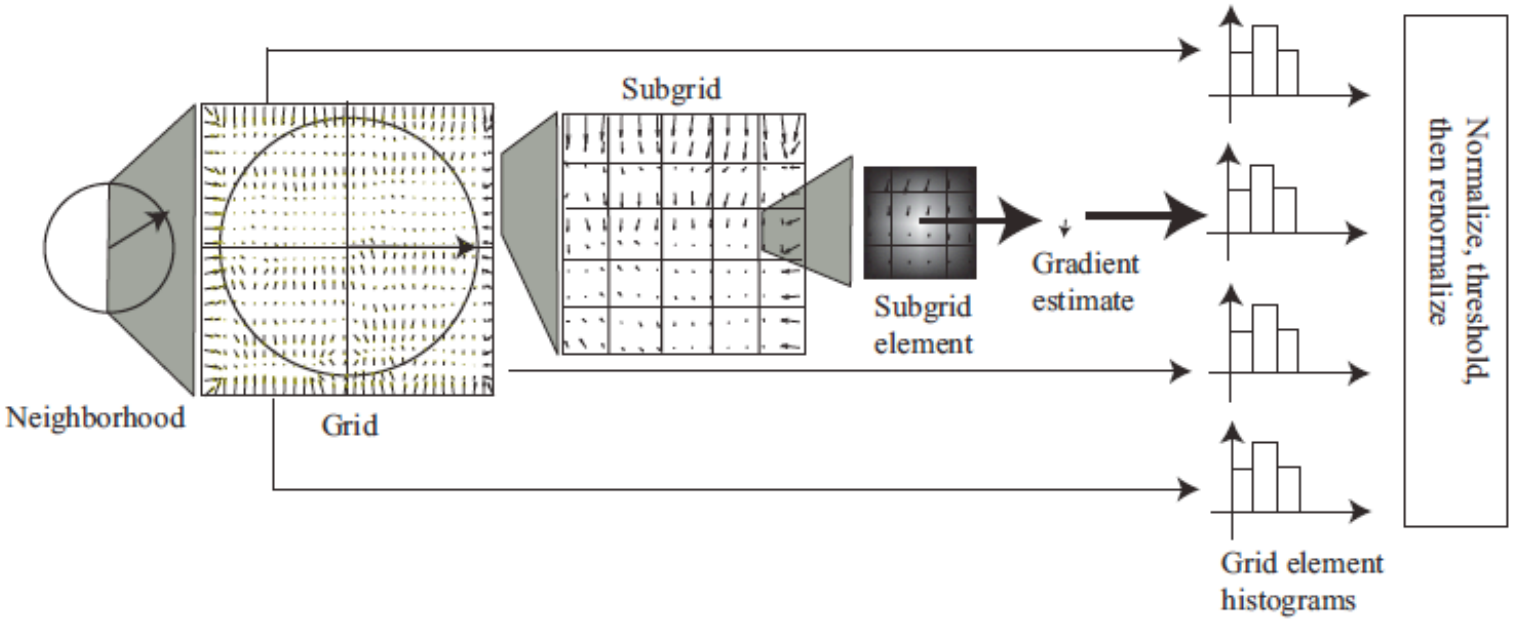


FIGURE 5.14: To construct a SIFT descriptor for a neighborhood, we place a grid over the rectified neighborhood. Each grid is divided into a subgrid, and a gradient estimate is computed at the center of each subgrid element. This gradient estimate is a weighted average of nearby gradients, with weights chosen so that gradients outside the subgrid cell contribute. The gradient estimates in each subgrid element are accumulated into an orientation histogram. Each gradient votes for its orientation, with a vote weighted by its magnitude and by its distance to the center of the neighborhood. The resulting orientation histograms are stacked to give a single feature vector. This is normalized to have unit norm; then terms in the normalized feature vector are thresholded, and the vector is normalized again.

Neighborhoods and SIFT - Key Points

- Algorithms to find neighborhoods
  - Represented by location, scale and orientation
  - Neighborhood is covariant
    - If image is translated, scaled, rotated
    - Neighborhood is translated, scaled, rotated
    - Important property for matching
  - Affine covariant constructions are available
- Once found, describe with SIFT features
  - A representation of local orientation histograms, comparable to HOG
  - Normalized differently

Learning to detect and describe keypoints

- You should be able to learn all this
  - keypoints are stable under rotation, translation, scale (homographies)
  - descriptions are stable under rotation, translation, scale (homographies)

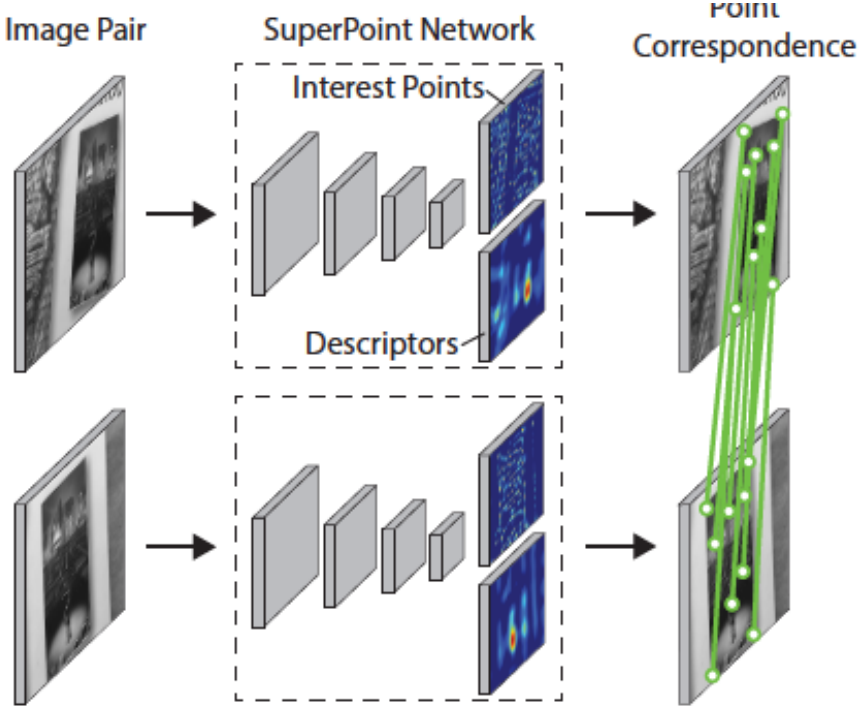


Figure 1. **SuperPoint for Geometric Correspondences.** We present a fully-convolutional neural network that computes SIFT-like 2D interest point locations and descriptors in a single forward pass and runs at 70 FPS on 480×640 images with a Titan X GPU.

DeTone et al, 18

SuperPoint

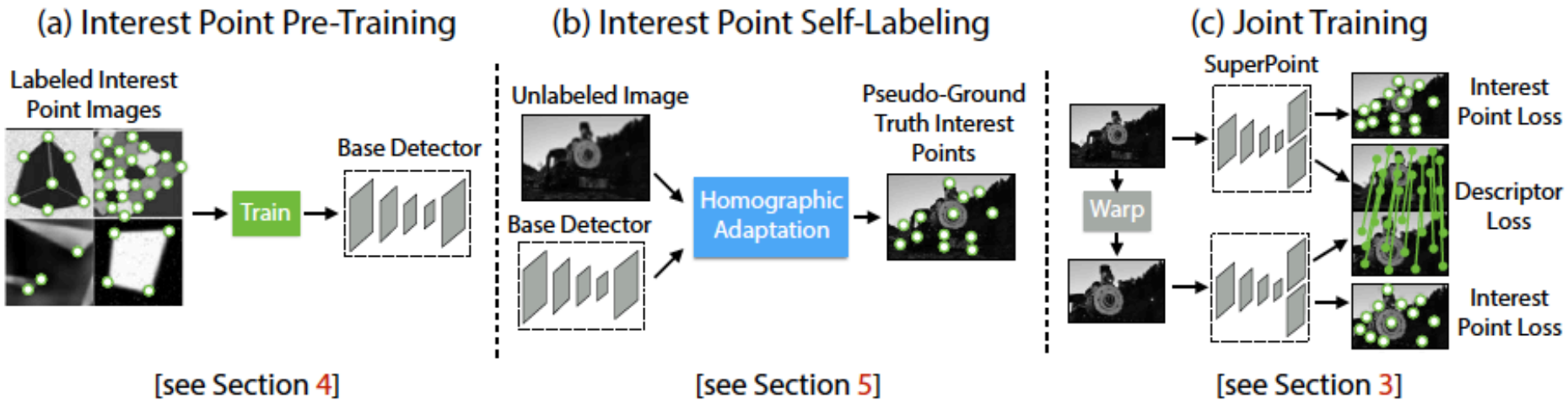


Figure 2. **Self-Supervised Training Overview.** In our self-supervised approach, we (a) pre-train an initial interest point detector on synthetic data and (b) apply a novel Homographic Adaptation procedure to automatically label images from a target, unlabeled domain. The generated labels are used to (c) train a fully-convolutional network that jointly extracts interest points and descriptors from an image.

DeTone et al, 18

SuperPoint

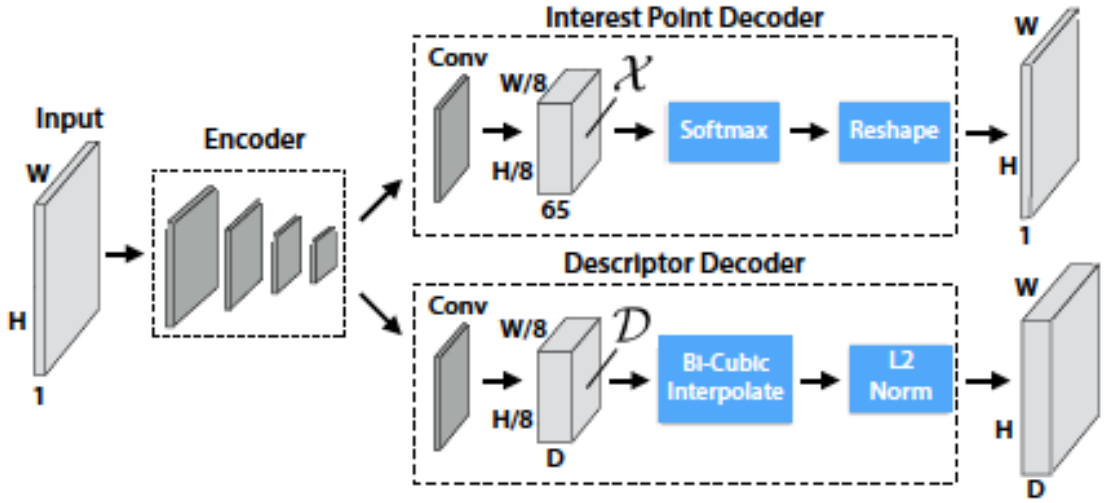


Figure 3. **SuperPoint Decoders.** Both decoders operate on a shared and spatially reduced representation of the input. To keep the model fast and easy to train, both decoders use non-learned upsampling to bring the representation back to $\mathbb{R}^{H \times W}$.

DeTone et al, 18

SuperPoint

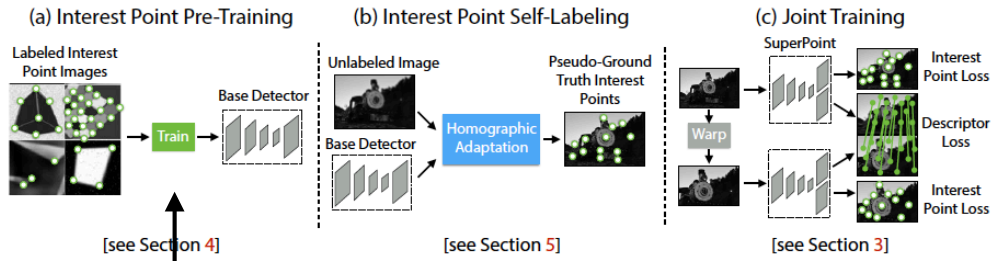


Figure 2. **Self-Supervised Training Overview.** In our self-supervised approach, we (a) pre-train an initial interest point detector on synthetic data and (b) apply a novel Homographic Adaptation procedure to automatically label images from a target, unlabeled domain. The generated labels are used to (c) train a fully-convolutional network that jointly extracts interest points and descriptors from an image.

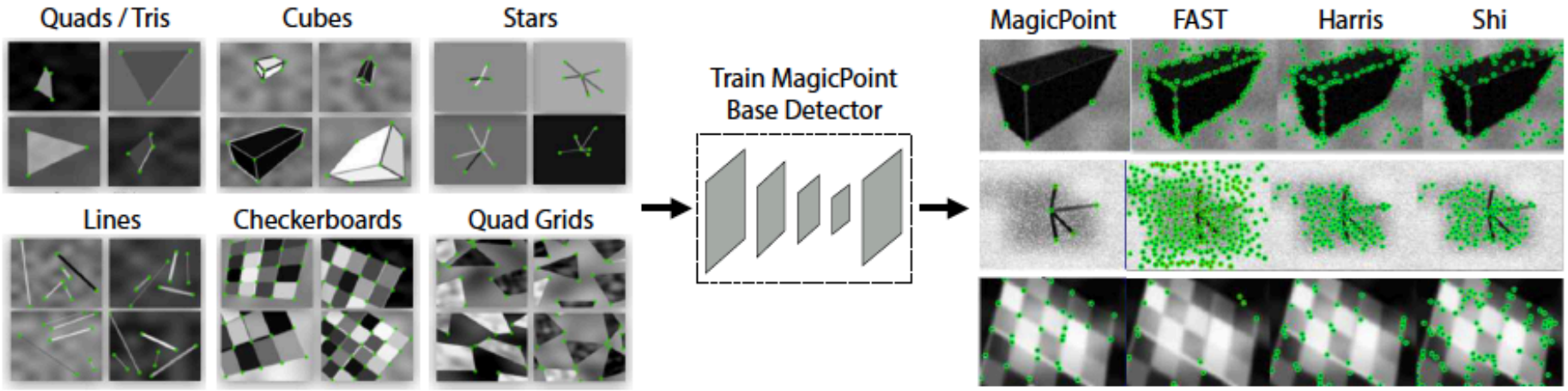


Figure 4. **Synthetic Pre-Training.** We use our Synthetic Shapes dataset consisting of rendered triangles, quadrilaterals, lines, cubes, checkerboards, and stars each with ground truth corner locations. The dataset is used to train the MagicPoint convolutional neural network, which is more robust to noise when compared to classical detectors.

DeTone et al, 18

SuperPoint

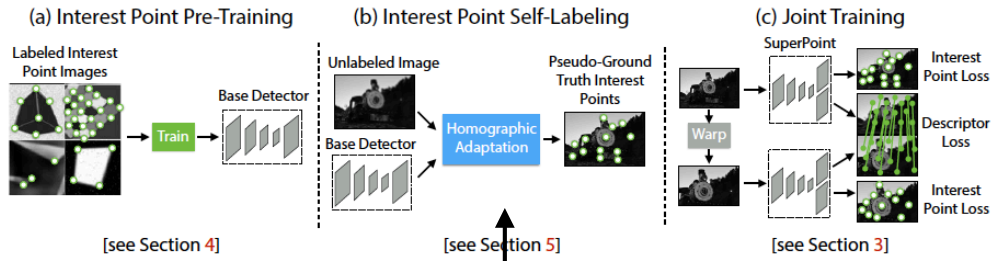


Figure 2. **Self-Supervised Training Overview.** In our self-supervised approach, we (a) pre-train an initial interest point detector on synthetic data and (b) apply a novel Homographic Adaptation procedure to automatically label images from a target, unlabeled domain. The generated labels are used to (c) train a fully-convolutional network that jointly extracts interest points and descriptors from an image.

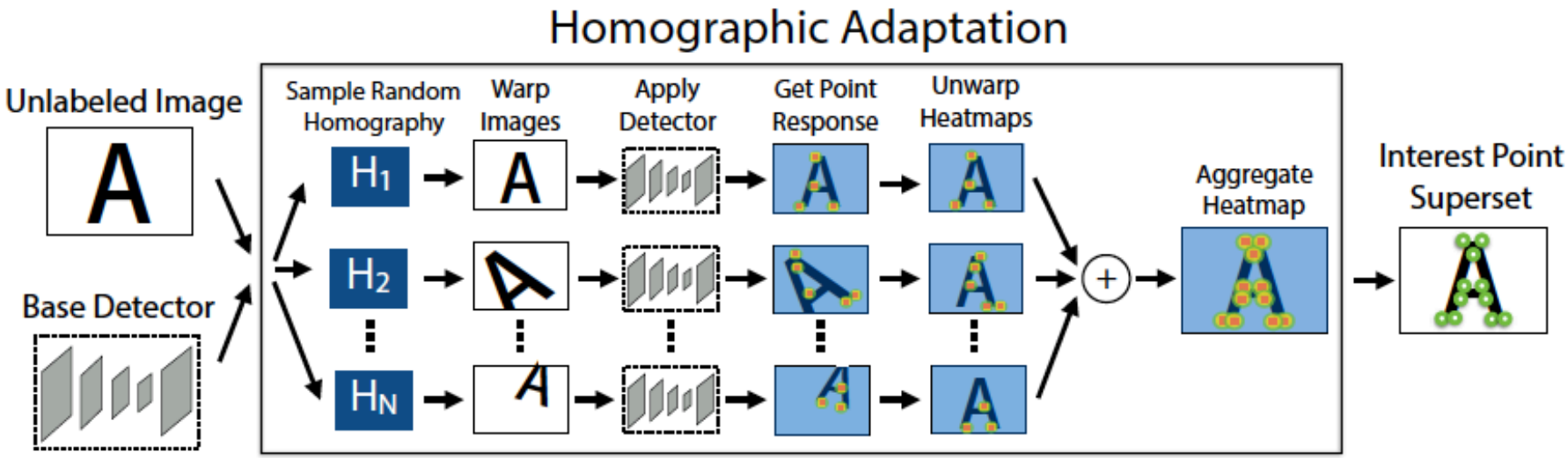


Figure 5. **Homographic Adaptation.** Homographic Adaptation is a form of self-supervision for boosting the geometric consistency of an interest point detector trained with convolutional neural networks. The entire procedure is mathematically defined in Equation 10.

DeTone et al, 18

SuperPoint

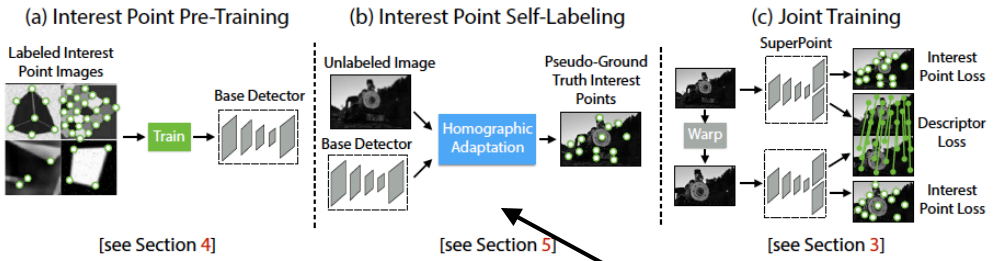


Figure 2. **Self-Supervised Training Overview.** In our self-supervised approach, we (a) pre-train an initial interest point detector on synthetic data and (b) apply a novel Homographic Adaptation procedure to automatically label images from a target, unlabeled domain. The generated labels are used to (c) train a fully-convolutional network that jointly extracts interest points and descriptors from an image.

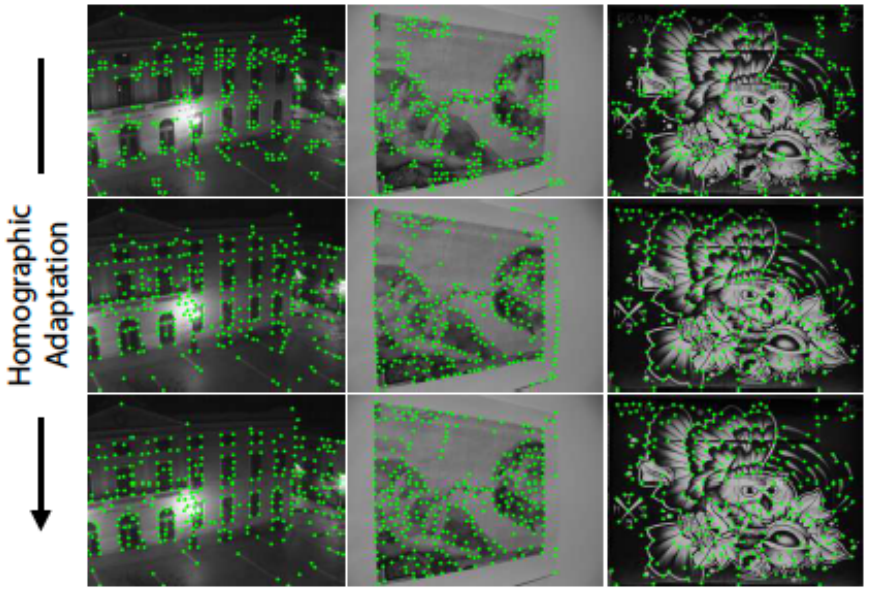


Figure 7. **Iterative Homographic Adaptation.** Top row: initial base detector (MagicPoint) struggles to find repeatable detections. Middle and bottom rows: further training with Homographic Adaption improves detector performance.

DeTone et al, 18

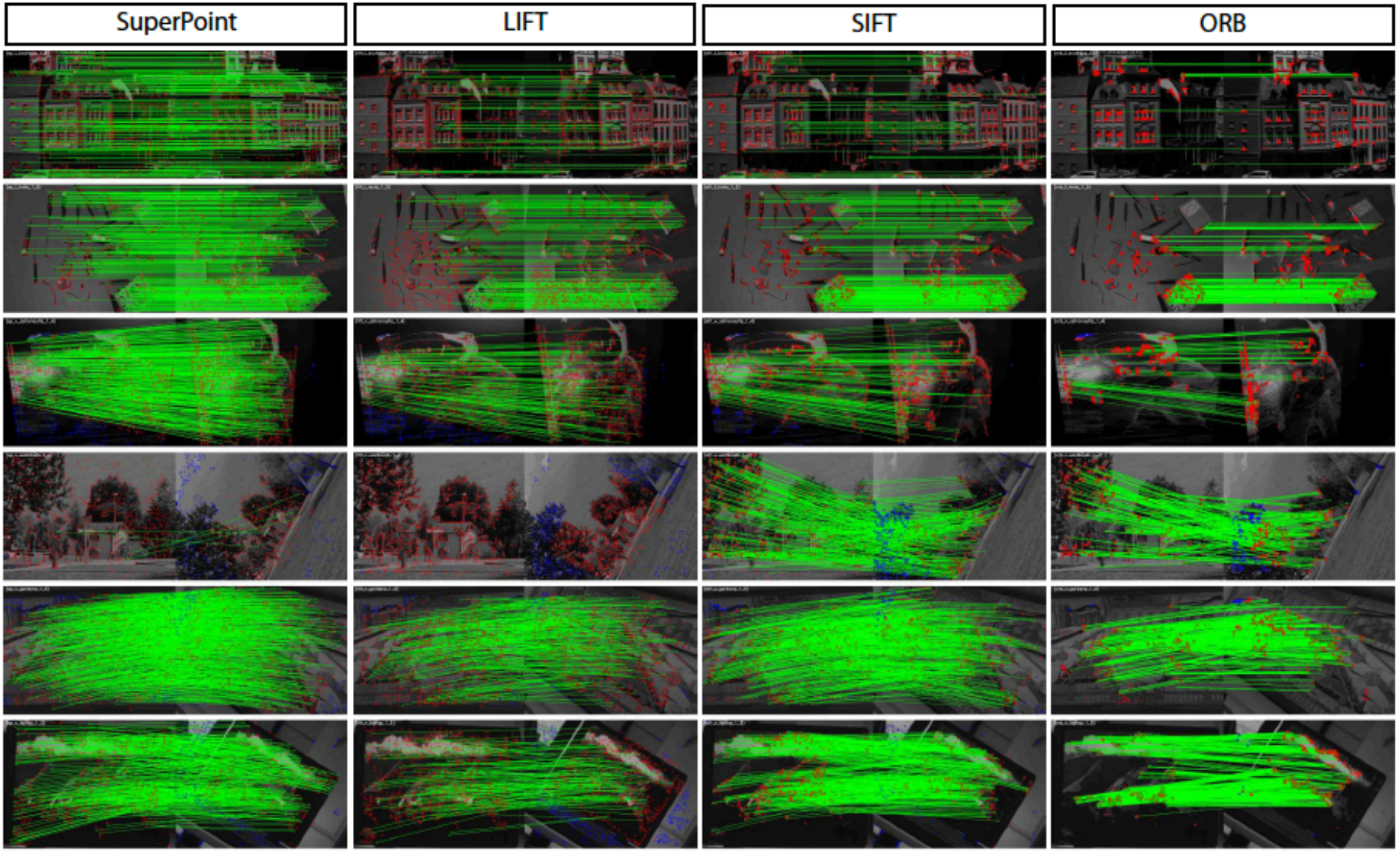


Figure 8. **Qualitative Results on HPatches.** The green lines show correct correspondences. SuperPoint tends to produce more dense and correct matches compared to LIFT, SIFT and ORB. While ORB has the highest average repeatability, the detections cluster together and generally do not result in more matches or more accurate homography estimates (see 4). Row 4: Failure case of SuperPoint and LIFT due to extreme in-plane rotation not seen in the training examples. See Appendix D for additional homography estimation example pairs.

DeTone et al, 18

SuperPoint

DeTone et al, 18