Interest points by UNet

D.A. Forsyth

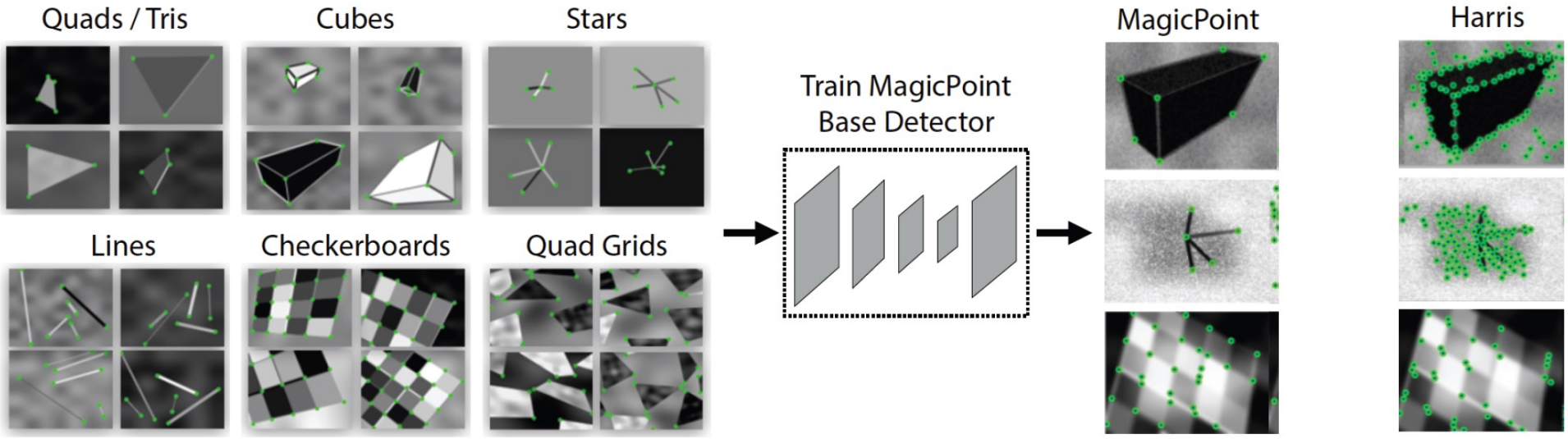
University of Illinois at Urbana Champaign

Interest points: Finding interest points

- Superpoint:
  - Break 8M X 8N image into 8x8 windows
  - Classify each window into 65 classes:
  - 64 locations and one “no-interest point in this window”
  - so decoder produces 65 X M X N block
  - [This can be refined…]

- Training:
  - First, make a lot of synthetic images with known corner locations
  - then train purely discriminatively (you know where the corners are!)

Yields MagicPoint



But you can do better...

Heatmaps

- Take the 65 x M x N block, compute the probabilities,
  - drop the “no interest point” score
  - get a 64 x M x N block of scores
  - don’t sum to one, cause one option is missing.

- Turn this into a 1 x (8 M) x (8 N) “image”
  - each pixel has a value that indicates the ”probability” of an interest point at that location

- This is a heatmap

Training with heatmaps

- Take a natural image $\mathcal{I}$
  - form its heatmap $\mathcal{H}[\mathcal{I}]$
  - Apply an affine transform A, to get A(I) and H(A(I))
  - Now $\mathcal{A}^{-1}[\mathcal{H}[\mathcal{A}[\mathcal{I}]]]$ and $\mathcal{H}[\mathcal{I}]$ should be the same
    - where they overlap

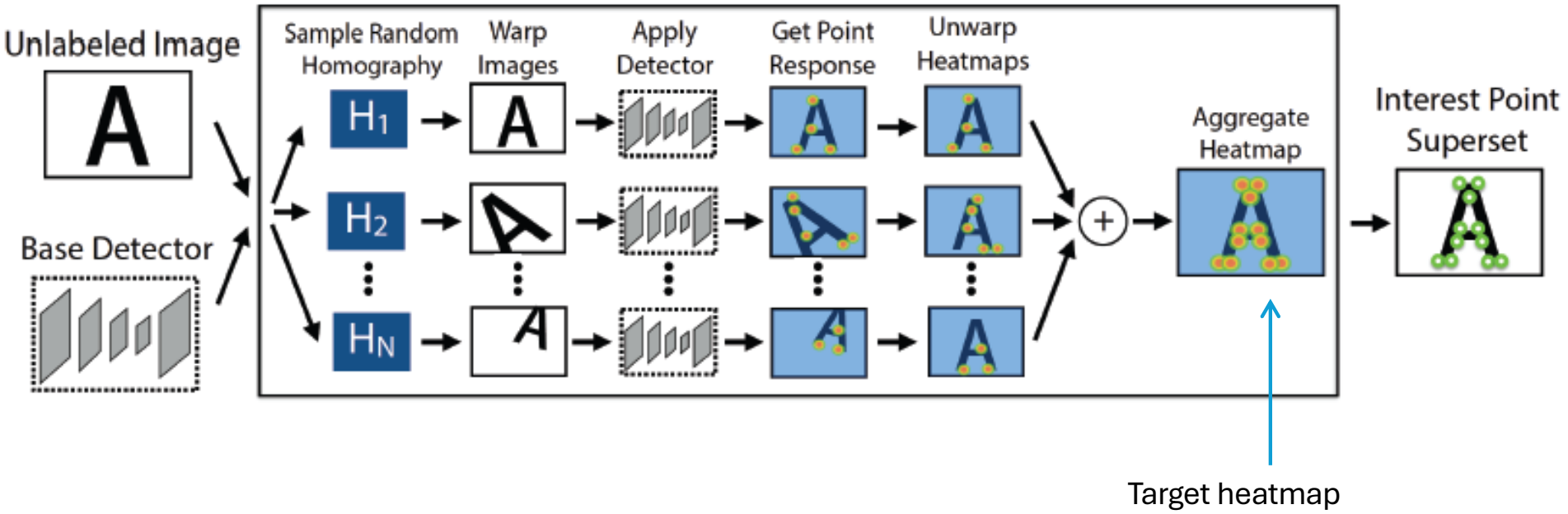
Training with heatmaps, II

- Idea:
  - get a collection of random affine transformations A_n
  - Now compute an average of

$$\frac{1}{N}\sum_i \mathcal{A}^{-1}\left[\mathcal{H}\left[\mathcal{A}\left[\mathcal{I}\right]\right]\right]$$

- use this as a target distribution
  - for training the unet that predicts the heatmap

SuperPoint


Unlabeled Image
Base Detector
Sample Random Homography
Warp Images
Apply Detector
Get Point Response
Unwarp Heatmaps
H_1
H_2
H_N
Aggregate Heatmap
Interest Point Superset
Target heatmap

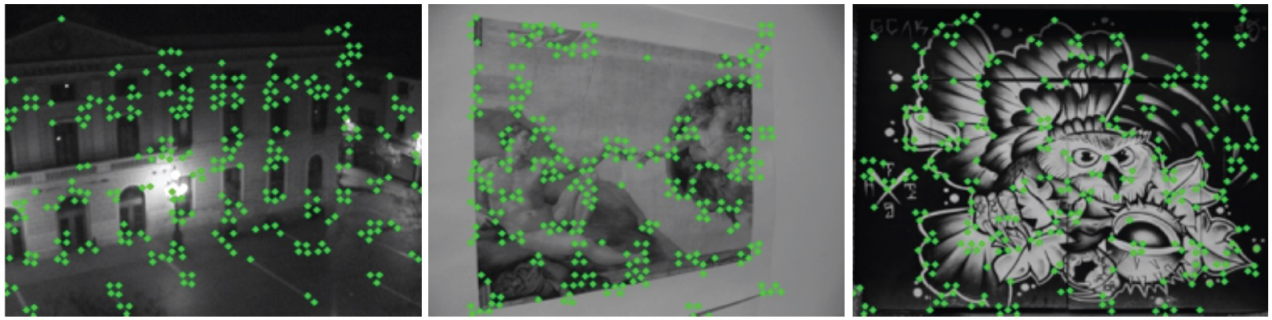
Locating the interest points

- Threshold scores from the trained decoder
  - interest points at locations where the score>threshold

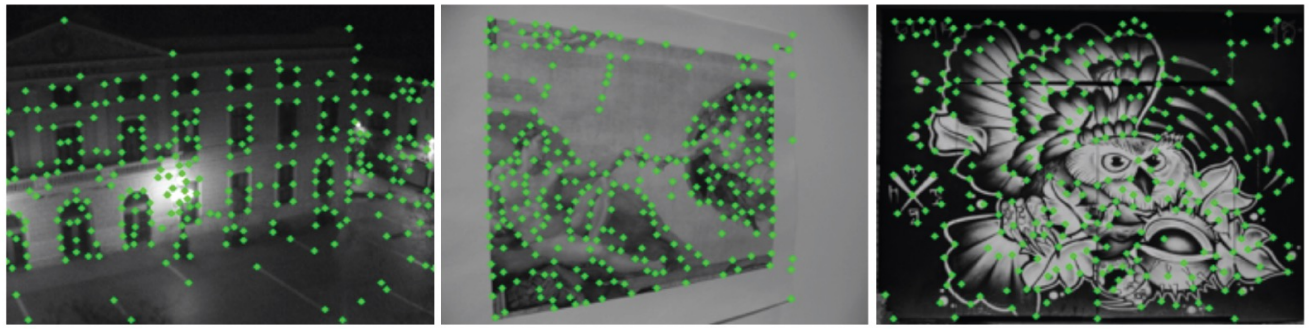
- These are on 8M x 8N grid

Real improvements from heatmap

MagicPoint



SuperPoint



Learning to describe the interest point

- Add a second decoder.
  - This decodes to a d x M x N block of descriptors
    - (but the interest point locations are on (8M x8N) grid – upsample and interpolate)

- Now use a version of the affine trick
  - For image I and image A(I) you know which pairs of tiles correspond

Tiles

One decoder produces the interest points, as above, and the other produces their descriptions. Because there is only one interest point per tile, it is enough to have one description per tile and upsample the descriptions by interpolation. Now take an image $\mathcal{I}$ and apply a transform $\mathcal{T}$ to the image to obtain $\mathcal{T}(\mathcal{I})$. This transform induces a correspondence between tiles, but remember the tiles are relatively coarse. For a point located at $\mathbf{x} = (i, j)^T$ in the image, write $\mathbf{u} = (u, v)^T = \mathcal{T}(\mathbf{x})$ for the

Using tile correspondences

- Idea:
  - Choose an image and an affine transformation
    - apply tx to image
    - know which tiles in txd image and original correspond
  - Loss:
    - corresponding tiles should have similar descriptors
    - non-corresponding tiles should have different descriptors
    - weight this, because there are more non-corresponding

Using tile correspondences

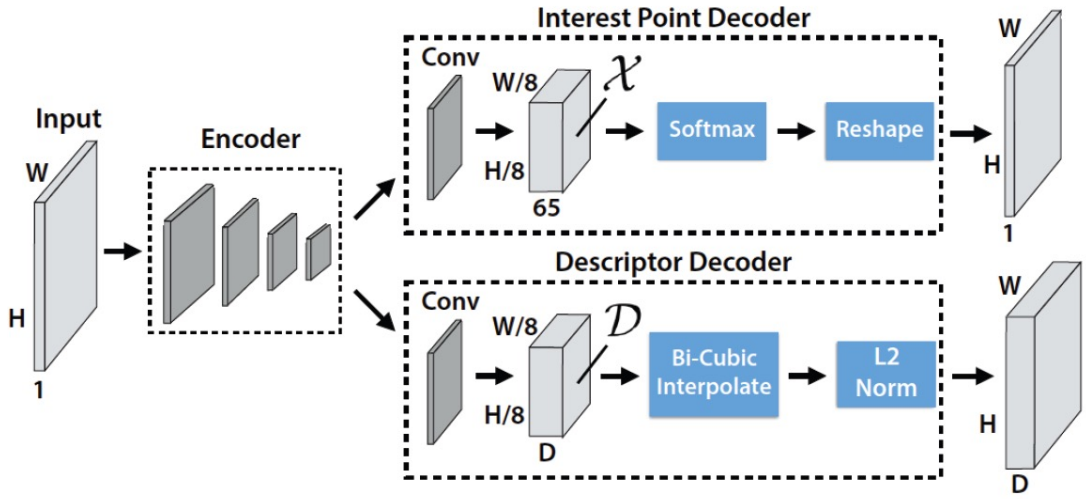
one description per tile and upsample the descriptions by interpolation. Now take an image $\mathcal{I}$ and apply a transform $\mathcal{T}$ to the image to obtain $\mathcal{T}(\mathcal{I})$. This transform induces a correspondence between tiles, but remember the tiles are relatively coarse. For a point located at $\mathbf{x} = (i, j)^T$ in the image, write $\mathbf{u} = (u, v)^T = \mathcal{T}(\mathbf{x})$ for the transformed coordinates. Now write

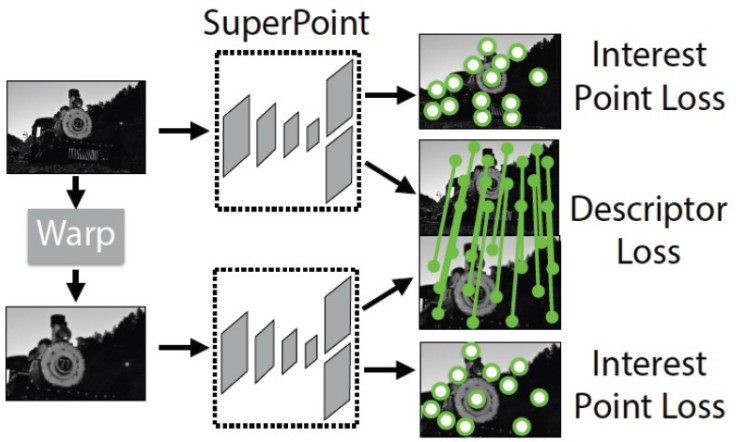
$$s_{\mathbf{x},\mathbf{u}} = \begin{cases} 1 & \text{if } \|\mathbf{x} - \mathbf{u}\|_2 \leq 8 \\ 0 & \text{otherwise} \end{cases}$$

Here $s_{\mathbf{x},\mathbf{u}}$ is one if the tiles correspond (roughly - the tiles are on a grid) and zero otherwise. As in the case of edge detection, there are more pairs that do not correspond than there are pairs that correspond. Deal with this using a weight λ. Then for a pair $\mathbf{x} \in \mathcal{I}$tiles and $\mathbf{u} \in \mathcal{T}(\mathcal{I})$tiles, the cost

$$C(\mathbf{x}, \mathbf{u}) = \lambda s_{\mathbf{x},\mathbf{u}} \max(0, c_1 - \mathbf{d}^T(\mathbf{x})\mathbf{d}(\mathbf{u})) + (1 - s_{\mathbf{x},\mathbf{u}}) \max(0, \mathbf{d}^T(\mathbf{x})\mathbf{d}(\mathbf{u}) - c_2)$$

forces $\mathbf{d}(\mathbf{x})$ and $\mathbf{d}(\mathbf{u})$ to be similar when $\mathbf{x}$ and $\mathbf{u}$ correspond and different when


Input
W
H
1
Encoder
Interest Point Decoder
Conv
W/8
H/8
65
$\mathcal{X}$
Softmax
Reshape
W
H
1
Descriptor Decoder
Conv
W/8
H/8
D
$\mathcal{D}$
Bi-Cubic Interpolate
L2 Norm
W
H
D


SuperPoint
Warp
Interest Point Loss
Descriptor Loss
Interest Point Loss

Think about this…

22.9. How would you modify Superpoint to get more or better interest points?

Lots of papers on this topic

ROUGH current SOTA: learned interest points don't produce notably better results than initial construction

Likely, this will change