Distribution Fields

A Unifying Representation for Low-Level Vision Problems

Erik Learned-Miller
with Laura Sevilla Lara, Manju Narayana, Ben Mears
Basin of attraction studies
GIVEN A RANDOM PATCH...
Basin of attraction studies

AND A RANDOM DISPLACEMENT...
Basin of attraction studies

CAN WE FIND OUR WAY HOME?
Basin of attraction studies
Basin of attraction results
Question

- How can we get the benefits of congealing without lots of images, and without a massive computational burden?
How do we line up a new image? *Funneling*...

Sequence of successively “sharper” models

Take one gradient step with respect to each model.
How to align a new image after congealing?

- More efficient to save sequence of distribution fields from congealing
  - High entropy to low entropy sequence → “Image Funnel”
- Funneling: increase likelihood of new image at each iteration according to corresponding distribution field
Aligning two images using the funneling concept

- Given image I and image J
- Generate many perturbed versions of image I, including the original image.
- Generate image funnel for set of I images.
Perturbed versions of an image
As an image stack.
Summing the perturbed stack.

\[ \text{Sum}(\text{stack}) = \text{result} \]
Distribution of perturbed stack.

\[ \text{Dist.}() = \]
Distribution fields

- Is there a simpler way to generate the idea of the distributions in a perturbed stack than to randomly make the images and then compute the distributions?
- Yes, distribution fields.
Exploding an image
Spatial Blur: 3d convolution with 2d Gaussian
Spatial Blur: 3d convolution with 2d Gaussian

KEY PROPERTY: doesn't destroy information through averaging
Feature space blur

- Delta function at one pixel
- Spatial blur
- Spatial and feature-space blur
How to compare?
How to compare?

- L1 distance?
- L2 distance?
- KL divergence?
The likelihood match

- Recall image I and patch J.
- Make a distribution field out of I and evaluate the likelihood of J under the field.
The likelihood match

Given distribution field \( D = D(I; \sigma) \) and image \( J \).

\[
Prob(J) = \prod_{i=1}^{N} p_{x,y}(J_{x,y})
\]
Sharpening match

$$\max_{\sigma} \text{Prob}(J; \sigma) = \prod_{i=1}^{N} p_{x,y}^{\sigma}(J_{x,y})$$
What standard deviation maximizes the likelihood of a given point under a zero-mean Gaussian?
Intuition behind sharpening match

- Increase standard deviation until it matches “average distance” to matching points.
Properties of the sharpening match

- A patch has probability of 1.0 under its own distribution field.
- Probability of an image patch degrades gracefully as it is translated away from best position.
- Optimum “sigma” value gives a very intuitive notion of the quality of the image match.
Tracking results

- State of the art results on tracking with standard sequences
  - Very simple code
  - Trivial motion model
  - Simple memory model
Distribution Fields
It’s not perfect...
Closely Related work

- Mixture of Gaussian backgrounding (Stauffer...)
- Shape contexts (Belongie and Malik)
- Congealing (me)
- Bilateral filter
- SIFT (Lowe), HOG (Dalal and Triggs)
- Geometric Blur (Berg)
- Rectified flow techniques (Efros, Mori)
- Mean-shift tracking
- Kernel tracking
- and many others...
- Lots more applications
  - Backgrounding
  - Image matching
  - Pixel unmixing
  - Superresolution
Motivations

- A distance between images:
  - Many metrics "broken" by slight misalignments.
    - Measure of distance or similarity should degrade gracefully with transformation.
  - "Invariant metrics" throw away a lot of information.
    - Integrating over regions
      - “max pooling”
      - Averaging over regions
  - Lose fine-grained spatial info:
    - Face recognition
Spatial Blur: Compare to regular image blur