Contents

12 Texture

Key issue Representing texture
1. Texture analysis/segmentation ⟶ representing texture
2. Texture synthesis: Useful, also gives some insight into quality of representation
3. Shape from texture Computer graphics: texture mapping

General
- Textures are made up of quite stylised subelements, repeated in meaningful ways.
- Representation: Find the subelements, and represent their statistics.
- Subelements: Normalized correlation, apply filters.
- Filters: Spots and oriented bars at a variety of different scales (by experience), details probably don't matter
- Statistics: Within reason, the more the merrier. At least, mean and standard deviation. Better, various conditional histograms.

Oriented pyramids Laplacian pyramid is orientation independent. Apply an oriented filter to determine orientations at each leayer. By clever filter design, we can simplify synthesis. This represents image information at a particular scale and orientation.

Final texture representation
1. Form an oriented pyramid (or equivalent set of responses to filters at different scales and orientations)
2. Square the output
3. Take statistics of responses
   a. e.g. mean of each filter output (are there lots of spots)
   b. std of each fiter output
   c. mean of one scale conditioned on other scale having a particular range of values (e.g. are the spots in straight rows?)

Texture synthesis
1. Use image as a source of probability model
2. Choose pixel values by matching neighbourhood, then filling in
3. Matching process: Look at pixel differences, count only sythesized pixels

12.1 Histogram

Principle
1. Intensity probability distribution
2. Captures global brightness information in a compact, but incomplete way
3. Doesn't capture spatial relationships

13 Questions

- $$I_{\text{comp}} = I_\alpha I_a + (1 - I_\alpha) I_b$$
- MAP, Maximum a posteriori detector.
- graph cuts
- Solve MRFs with graph cuts
- impulse response $t(-x, -y)$
- Canny nonmaxima suppression
- Entropy Coding (Huffman code)
- Aperture problem: normal flow
- Lucas-Kanade: Iterative refinement/local gradient method
- Coarse-to-fine-estimation
- SNR scalability EI, EP frame
- MPEG Structure
- Radon transform: Just FT with first Fourier coefficient?
- Sparse representations for image restoration: Inpainting, demosaickiking

A Big equations

$$\begin{aligned} \mathcal{F}[h](u,v) &= \frac{1}{2\ell}\int_{-\ell}^{\ell} \mathrm{d}x_1\ \exp(-i2\pi u x_1) \cdot \underbrace{\int_{-\infty}^{\infty} \mathrm{d}x_2\, \delta(x_2) \exp(-i2\pi v x_2)}_{=1} \\ &= \mathrm{sinc}(2\pi u \ell) \end{aligned} \tag{A.1}$$

$$\begin{aligned} E = \iint \mathrm{d}x\mathrm{d}y \Bigg[\left(\frac{\partial I}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial I}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial I}{\partial t} \right)^2 \\ + \alpha^2 (\|\nabla \dot{x}\| + \|\nabla \dot{y}\|)^2 \Bigg] \end{aligned} \tag{A.2}$$

$$\begin{aligned} \mathbf{v} = &\begin{pmatrix} \sum_i w_i I_x(q_i)^2 & \sum_i w_i I_x(q_i) I_y(q_i) \\ \sum_i w_i I_x(q_i) I_y(q_i) & \sum_i w_i I_y(q_i)^2 \end{pmatrix}^{-1} \\ &\cdot \begin{pmatrix} -\sum_i w_i I_x(q_i) I_t(q_i) \\ -\sum_i w_i I_y(q_i) I_t(q_i) \end{pmatrix} \end{aligned} \tag{A.3}$$