

MRF Stereo: Bayesian vs. CRF Approaches

Version 1.1

1 Introduction

This short paper attempts to motivate CRFs (conditional random fields) in the context of stereo.

CRFs in a nutshell: suppose observable (visible) variables are denoted V , and hidden variables by H . For instance, V represents all pixel data and/or image features, and H represents something you're trying to infer about the image.

The Bayesian approach says to make a prior model $P(H)$ and a likelihood model $P(V|H)$, and to learn each model separately from data.

By contrast, the CRF approach says that we only care about the posterior $P(H|V)$, so that is all that we explicitly model and learn. (Note: CRFs, which are conditional random fields, are usually formulated as graphical models, but I would argue that the basic idea of modeling and estimating just the posterior is more general than graphical models.)

2 Notation

The left and right images are L and R , with $L(i, j)$ representing the intensity value at row i and column j , etc.

D is the disparity field, with D_{ij} representing the disparity value at row i and column j .

Following the Point Grey convention, $L(i, j + d) \approx R(i, j)$ if $D_{ij} = d$ is the correct disparity at pixel (i, j) , and provided the camera gain is similar in the left and right images.

We denote disparity Cues by $C_k(i, j, d)$, where $k = 1, 2, \dots, K$ labels which cue, (i, j) is the pixel in question and d is a disparity value. The simplest cue is

$$C_1(i, j, d) = |L(i, j + d) - R(i, j)| \tag{1}$$

which is usually small for disparity values d close to the correct value.

Another cue compares the horizontal image derivative. Let $L_h = \partial L(i, j)/\partial j$ and $R_h = \partial R(i, j)/\partial j$. Then

$$C_2(i, j, d) = |L_h(i, j + d) - R_h(i, j)| \tag{2}$$

3 Bayesian Approach

We have a prior model $P(D)$, which can be thought of as a prior on surface geometry (recall that disparity d is given by $d = fB/Z$ where f is focal length, B is baseline separation of the two cameras and Z is the coordinate along the camera line of sight) since there is no mention of intensity image characteristics. The prior enforces smoothness and often has a form such as $P(D) = e^{\alpha E_s(D)}/Z(\alpha)$ where:

$$E_s(D) = \sum_{(i,j)} |D_{i+1,j} - D_{i,j}| + |D_{i,j+1} - D_{i,j}| \quad (3)$$

where we assume the sum is restricted so that the (row, column) coordinates on the RHS never go beyond the image bounds. (The subscript in $E_s(\cdot)$ stands for "smoothness.")

The likelihood model $P(C|D)$ models the evidence provided by the disparity cues. A simple form could be $P(C|D) = \prod_k P(C_k|D)$, where for instance $P(C_k|D) = \prod_{(i,j)} f_k(C_k(i,j), D_{ij})$ and $f_k(C_k) = e^{\beta_k C_k}/Z_k(\beta_k)$.

Notice that the likelihood model makes two kinds of conditional independence assumptions: (1) it assumes all cues are conditionally independent given the disparity field; and (2) it assumes the cue values are conditionally independent (given the disparity field) from pixel to pixel.

3.1 Bayesian Learning

We learn the prior and likelihood models *separately*. The prior $P(D)$ can be learned given samples of (correct) disparity maps, even without the accompanying left and right images they correspond to. In our example, the prior (smoothness) parameter α can be learned by maximum likelihood: given sample disparity images $D^{(1)}, D^{(2)}, \dots, D^{(M)}$, etc. we determine α as follows:

$$\alpha^* = \arg \max_{\alpha} P(D^{(1)})P(D^{(2)}) \dots P(D^{(M)}) \quad (4)$$

which is equivalent to:

$$\alpha^* = \arg \max_{\alpha} \sum_m \log P(D^{(m)}) = \arg \max_{\alpha} [\alpha \sum_m (E_s(D^{(m)}) - \log Z(\alpha))] \quad (5)$$

If we define the log likelihood $L(\alpha) = \sum_m \log P(D^{(m)})$, then (because the probabilities are defined as exponentials of energies scaled by the unknown parameter α) it turns out that $L(\cdot)$ is convex in α . This is good news, but it's still computationally difficult to estimate α^* , and there are lots of papers written on techniques for speeding up this process. See my tutorial "Maximum Entropy Distributions and Their Relationship to Maximum Likelihood" on the Tutorials section of the lab webpage for more about maximizing the $L(\cdot)$ function.

We can learn the likelihood parameters by using sample disparity maps and sample cue values (computed for real stereo image pairs). For example, we estimate β_k as follows:

$$\beta_k^* = \arg \max_{\beta_k} \sum_m \log P(C_k^{(m)} | D^{(m)}) = \arg \max_{\beta_k} \sum_m \sum_{(i,j)} [\beta_k C_k(i, j, D_{ij}^{(m)}) - \log Z_k(\beta_k)] \quad (6)$$

where $C_k^{(m)}$ is the m^{th} sample value of cue k (all over the lattice).

3.2 Bayesian Inference

Typically one estimates the MAP (maximum a posterior), i.e.

$$D^* = \arg \max_D P(D|C) = \arg \max_D P(D)P(C|D) \quad (7)$$

since $P(D|C) = P(C|D)P(D)/P(C)$ and $P(C)$ is independent of D . In other words, the MAP estimate is

$$D^* = \arg \max_D E_s(D) + \sum_k \sum_{(i,j)} \beta_k C_k(i, j, D_{ij}) \quad (8)$$

Notice that the partition functions $Z_k(\cdot)$, $Z(\cdot)$ don't appear in the MAP expression because they don't depend on D .

4 CRF Approach

Although we are not going to explicitly model a smoothness prior in the CRF approach, we can certainly consider smoothness of disparity to be a "cue" that happens to be a function only of disparities, not image properties. E.g. define a third cue in addition to the two others in Sec. 2:

$$C_3(i, j, d) = |D_{i+1,j} - D_{i,j}| + |D_{i,j+1} - D_{i,j}| \quad (9)$$

Then we can write the posterior in general as:

$$P_{crf}(D|C) = \frac{1}{Z_{crf}(C)} e^{E_{crf}(D,C)} \quad (10)$$

where $E_{crf}(D, C) = \sum_k \beta_k [\sum_{(i,j)} C_k(i, j, D_{ij})]$. If we define a vector c with K components as follows: $c_k = \sum_{(i,j)} C_k(i, j, D_{ij})$, then we have $E_{crf}(D, C) = \beta \cdot c$.

4.1 CRF Learning

Similar to before:

$$\beta^* = \arg \max_{\beta} \sum_m \log P_{crf}(D^{(m)} | C^{(m)}) = \arg \max_{\beta} \sum_m (\beta \cdot c^{(m)} - \log Z_{crf}(C^{(m)})) \quad (11)$$

4.2 CRF Inference

The MAP estimate is given by:

$$D^* = \arg \max_D \beta \cdot c \quad (12)$$

where c is a function of the disparity field D . The equation has the same form as for Bayesian inference (assuming that smoothness has been incorporated as one of the cues), but in general the coefficients in the equation are different.

5 Discussion

Main points:

(1.) The posterior model can be expressed as a factor graph. To prevent the misleading impression that the posterior models the distribution of cues (and/or image pixel values), the only variable nodes in the factor graph should be the hidden variables we are trying to infer. (The cues enter into the definitions of the factors, not as explicit variables in the factor graph.)

(2.) Since the posterior model doesn't attempt to model the distribution of cues, it has an advantage over the Bayesian approach, which often models the distribution of cues with various conditional independence assumptions, which are sometimes bad assumptions.

(3.) The factor graph you use to perform inference may have the same form in the Bayesian case and the CRF case: in both cases the variable nodes are unknown disparities, there are arity-1 factors to express evidence for various disparity values at each pixel, and there are arity-2 factors to enforce smoothness between neighboring pixels. The big difference is how you learn the factor graph parameters that define the factor graph potentials!

(4.) An extreme (maybe silly) example showing the advantage of the CRF over the Bayesian approach:

Suppose someone tells you there are two cues $C_1(i, j, d)$ and $C_2(i, j, d)$ but misleadingly fails to mention that these are in fact the exact same cues. Because of its independence assumption our naive Bayesian approach will learn separate coefficients β_1 and β_2 , which will turn out to be equal, but this will have the effect of exaggerating the amount of information given by the one real cue. (By contrast, if only one cue is used, the same coefficient value as before would be learned.) The CRF approach will also assign equal weights to β_1 and β_2 , but these weights will each be half as big as if only one cue (β_1) is used – in other words, it doesn't mistakenly assume the cues are independent.