

1 Introduction

What is computer vision?

Developing computational models to interpret images and to understand the visual world. CV is an **inverse problem** (most often highly ill-posed).

Basics

Bayes rule $p(\text{state}|\text{images}) = \frac{p(\text{images}|\text{state}) \cdot p(\text{state})}{p(\text{images})}$

Likelihood: $(p(\text{images}|\text{state}))$ is an observation model that describes how we obtained the image measurements, given a particular state of the world.

Prior: $(p(\text{state}))$ models our a-priori assumptions about the world, or the state of the world.

Normalization term: $(p(\text{images}))$ can be ignored in practice.

Gaussian $\mathcal{N}(\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right\}$

Laplace $\mathcal{L}(\mu, \sigma) = \frac{1}{\sigma} \exp\left\{-\frac{|x-\mu|}{\sigma}\right\}$

Chain rule

$p(x_n, \dots, x_1) = p(x_n|x_{n-1}, \dots, x_1) \cdot p(x_{n-1}, \dots, x_1)$

Independence $p(a, b) = p(a) \cdot p(b)$

2 Robust Statistics

A **robust** loss is a loss which is not dragged off by outliers.

Probability vs. Cost vs. Energy (Boltzmann distribution) $p(x) = \frac{1}{Z(T)} \exp\left\{-\frac{1}{T}E(x)\right\}$

$p(x)$ probability of a state, $Z(T)$ partition function, T temperature, $E(x)$ energy of a state

Modeling the Likelihood

General error distribution (likelihood)

$\log p(\mathcal{X}|\theta) - \sum_i \rho(f(x|\theta), \sigma) + \text{const}$, $p(x, \sigma)$ robust error function.

Squared Error $\frac{1}{2\sigma^2}x^2$ corresponds to the Gaussian distribution, influence is linear (loss gradient).

L1 Norm $\frac{|x|}{\sigma}$ corresponds to the Laplacian distribution, influence is constant.

Lorentzian $\log\left(1 + \frac{1}{2\sigma^2}x^2\right)$ corresponds to the Student-t distribution, influence is redescending to zero for large errors.

Barron Loss unifies multiple common loss functions

$\rho(x, \alpha, c) = \frac{|\alpha-2|}{\alpha} \left(\left(\frac{(x/c)^2}{|\alpha-1|} + 1 \right)^{\alpha/2} - 1 \right)$, parameters

(shape α & scale c) can be learned by using the>NNL of p corresponding to ρ

$(-\log(p(x|\alpha, c))) = \rho(x, \alpha, c) + \log(cZ(\alpha))$.

General remark Gaussian assumption is likely to be inappropriate in most cases.

Alternative Robust Approach (RANSAC)

RANdom SAMple Consensus (RANSAC) is an iterative technique for robust fitting that finds wide application in CV. Breakdown point 50%. Idea: sample (\mathcal{U}) small subset of data, fit subset, distinguish between signal (close to fit) and noise, refit (choose best fit).

Prior Modeling

Model a-priori assumptions about the world, such as assuming consistency in depth, disparity, or optical flow estimation.

Consistency assumption, however, should also be robust to discontinuities.

Markov Random Fields is a general prior for assuming consistency between neighboring pixels. $p((d)) \propto \prod_{i,j} f_H(d_{i,j}, d_{i+1,j}) \cdot f_V(d_{i,j}, d_{i,j+1})$, f_H, f_V compatibility functions. Can be seen as a undirected graphical model.

Potts Model

$f_H(d_{i,j} - d_{i+1,j}) = \frac{1}{Z(T)} \exp\left\{\frac{1}{T}\right\} \delta(d_{i,j} - d_{i+1,j})$ with

$$\delta(a - b) = \begin{cases} 1, & a = b \\ 0, & a \neq b \end{cases}$$
 Not differentiable!

3 Graphical Models

Main types: directed Graphical Models (GMs) (Bayesian networks) & undirected GMs (Markov random fields)

Main components: Nodes & Edges (directed or undirected)

Directed Graphical Models

Based on a **directed graph**, nodes correspond to **random variables**, directed edges correspond to (causal) **dependencies**.

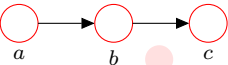


Nodes can be different random variables (binary events, discrete variables, continuous variables).

Simplest GM $p(b|a)$ conditional probability, $p(a)$ prior probability, $p(a, b) = p(b|a)p(a)$ joint probability, $p(a) = \sum_b p(a, b) = \sum_b p(b|a)p(a)$ marginalization, $p(a|b) = \frac{p(a,b)}{p(b)}$ conditional probability.

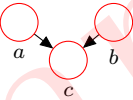


Chain of nodes $p(a, b, c) = p(c|b)p(b|a)p(a)$



Convergent connections

$p(a, b, c) = p(c|a, b)p(a)p(b)$



General Formulation

Set of nodes (random variables) $V = \{x_1, \dots, x_n\}$

Directed acyclic (no directed cycle) graph $G = (V, E)$

V : nodes, E directed edges

Conditional probability $p(x_i|\{x_j|j \in \text{Parents}(i)\})$

Joint probability

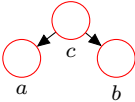
$p(x_1, \dots, x_n) = \prod_{i=1}^n p(x_i|\{x_j|j \in \text{Parents}(i)\})$

Complexity

GM reduce the complexity from $\mathcal{O}(2^n)$ (Joint prob. brute force) to $\mathcal{O}(n \cdot 2^k)$, with k as the max number of parents of a node.

Conditional Independence

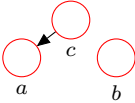
Example 1 are a and b independent?



Marginalize over c :

$p(a, b) = \sum_c p(a, b, c) = \sum_c p(a|c)p(b|c)p(c)$. a and b are not independent.

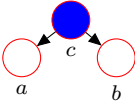
Example 2 are a and b independent?



Marginalize over c :

$p(a, b) = \sum_c p(a, b, c) = \sum_c p(a|c)p(b|c)p(c) = p(a)p(b)$. a and b are independent. If there is no undirected connection between two variables, then they are independent.

Example 3 are a and b independent, given c ?



Use conditional probability:

$p(a, b|c) = \frac{p(a, b, c)}{p(c)} = \frac{p(a|c)p(b|c)p(c)}{p(c)} = p(a|c)p(b|c)$. If c is given a and b become conditionally independent.

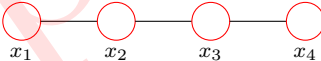
4 Undirected Graphical Models

Set of nodes (random variables) $V = \{x_1, \dots, x_n\}$

Undirected graph $G = (V, E)$, may include cycles V nodes, E undirected edges

Chain graph

$p(x_0, x_1, x_2, x_3) = \frac{1}{Z} f(x_0, x_1) f(x_1, x_2) f(x_2, x_3)$



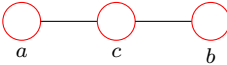
General Formulation

$p(x_1, \dots, x_n) = \frac{1}{Z} \prod_{c \in C} f_c(\{x_j|j \in c\})$, with $\frac{1}{Z}$ normalization factor, f_c unnormalized non-negative function, and C cliques of the graph.

Cliques are may ambiguous

Conditional Independence

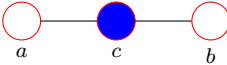
Example 1 are a and b independent?



Marginalize over c :

$p(a, b) = \sum_c p(a, b, c) = \frac{1}{Z} \sum_c f_1(a, c) f_2(b, c)$. a and b are not independent.

Example 1 are a and b independent?



Use conditional probability:

$p(a, b|c) = \frac{1}{p(c)} \frac{1}{Z} f_1(a, c) f_2(b, c) =$

$\frac{1}{Z} \hat{f}_a(a, c) \frac{1}{Z} \hat{f}_b(b, c)$, factors are proportional to $p(a|c)$ and $p(b|c)$. Hence conditional independence of the variables.

Generally two variables a and b are conditionally independent given a set other variables S , if you

cannot reach a from b in the graph without passing through S .

5 Probabilistic Inference

generally refers to:

- Computing the max of the posterior distribution (MAP)
- Computing expectations over the post distribution, such as the mean
- Computing marginal distributions

Continuous Optimization (MAP)

Is the most straightforward idea for MAP. Typically gradient ascent or similar first order approaches used. The posterior is most often multi-modal (resulting in non-convex optimization), thus optimization can end up in a local optimum.

Gradient Ascent

First order optimization approach. Gradient needed!

$\mathbf{x}^{(n+1)} \leftarrow \mathbf{x}^{(n)} + \eta \cdot \Delta f(\mathbf{x}^{(n)})$, η learning rate

Stereo with Continuous Optimization

Optimize the log-posterior

$\arg \min_{\mathbf{d}} (\log p(\mathbf{I}_1, \mathbf{I}_0, \mathbf{d}) + \alpha \log p(\mathbf{d}))$

Prior $\log p(\mathbf{d}) =$

$\log \left[\frac{1}{Z} \prod_{i,j} f_H(d_{i,j}, d_{i+1,j}) \cdot f_V(d_{i,j}, d_{i,j+1}) \right] = \sum_{i,j} \log f_H(d_{i,j}, d_{i+1,j}) + \log f_V(d_{i,j}, d_{i,j+1}) + c$

Gradient of Log-Prior

$\frac{\partial}{\partial d_{k,l}} \log p(\mathbf{d}) = \sum_{i,j} \frac{\partial}{\partial d_{k,l}} \log f_H(d_{i,j}, d_{i+1,j}) +$

$\frac{\partial}{\partial d_{k,l}} \log f_V(d_{i,j}, d_{i,j+1})$

$\frac{\partial}{\partial d_{k,l}} \log p(\mathbf{d}) =$

$\frac{\partial}{\partial d_{k,l}} \log f_H(d_{k,l}, d_{k+1,l}) + \frac{\partial}{\partial d_{k,l}} \log f_H(d_{k-1}, d_{k,l}) + \frac{\partial}{\partial d_{k,l}} \log f_V(d_{k,l}, d_{k,l+1}) + \frac{\partial}{\partial d_{k,l}} \log f_V(d_{k,l-1}, d_{k,l})$

$\frac{\partial}{\partial d_{k,l}} = \frac{\frac{\partial}{\partial d_{k,l}} f_H(d_{k,l}, d_{k+1,l})}{f_H(d_{k,l}, d_{k+1,l})}$

$\frac{\partial}{\partial d_{k,l}} f_H(d_{k,l}, d_{k+1,l})$ has to be differentiable. This is not the case for the Potts model!

Gradient Log-Likelihood (Gaussian case)

$\log p(\mathbf{I}_1|\mathbf{I}_0, \mathbf{d}) = - \sum_{i,j} \frac{1}{2\sigma^2} (I_{i,j}^0 - I_{(i-d_{i,j}),j}^1)^2 + c$

See image as a continous function

$\frac{\partial}{\partial d_{k,l}} \log p(\mathbf{I}_1|\mathbf{I}_0, \mathbf{d}) =$

$\frac{1}{\sigma^2} (I^0(k, l) - I^1(k - d_{k,l}, l)) \frac{\partial}{\partial d_{k,l}} I^1(k - d_{k,l}, l)$

Second term is the horizontal image derivative

In practice this approach does not work too good, due to local many local optima.

Gaussian pyramid can be used to avoid local optima.

Stereo with Discrete Optimization

General is the discrete optimization of the posterior NP hard. For MRFs, there are polynomial time algorithms.

Disparity $d = 1$

Row of pixels

Disparity $d = 0$

Edges from disparity to pixels indicate the cost of pairing (cost for not assign $d = 1$ to the pixel). Horizontal edges is the cost for assigning the neighboring pixels to different disparities

Graph Cuts MAP estimation by using the min-cut/max-flow algorithm (e.g. Ford-Fulkerson) Only works for **submodular** functions $g(0,0) + g(1,1) \leq g(1,0) + g(0,1)$ (e.g. Potts model). **α -expansion** for multiple labels. Init disparity, repeatedly sweep through all disparities, Treat intermediate solution as one disparity level and the current proposed disparity α as the other, solve binary graph cut problem, repeat. Approximation guarantee, No worse than factor 2 from optimal (for Potts model).

Marginals

Useful to be able to compute marginal distributions. Needed for certain learning approaches, useful for assessing uncertainty, and allows for computing expecations over the results.

Marginalization with Messages

$$p(E) = \sum_A \sum_B \sum_C \sum_D p(A, B, C, D, E) = \sum_A \sum_B \sum_C \sum_D \frac{1}{Z} f_1(A, B) f_2(B, D) f_3(C, D) f_4(D, E)$$

$$m_{A \rightarrow B}(B) = \sum_A f_1(A, B)$$

$$m_{C \rightarrow D}(D) = \sum_C f_3(C, D)$$

$$m_{B \rightarrow D}(D) = \sum_B f_2(B, D) \cdot m_{A \rightarrow B}(B)$$

$$m_{D \rightarrow E}(E) = \sum_D f_4(D, E) \cdot m_{B \rightarrow D}(D) \cdot m_{C \rightarrow D}(D)$$

$$P(E) = \frac{1}{Z} m_{D \rightarrow E}(E)$$

Sum-Product Algorithm

Message rule:

$$m_{j \rightarrow i}(x_i) = \sum_{x_j} f_{ij}(x_i, x_j) \prod_{k \in \mathcal{N}(i) \setminus j} m_{k \rightarrow i}(x_i)$$

Marginals: $p(x_i) \propto \prod_{k \in \mathcal{N}(i) \setminus j} m_{k \rightarrow i}(x_i)$

Loopy Belief Propagation

Apply the sum-product algorithm even if the graph is not a tree. Iterative update messages until convergence. At convergence the so-called beliefs $b(x_i)$ are approximations of the marginals.

$$p(x_i) \approx b(x_i) \propto \prod_{k \in \mathcal{N}(i)} m_{k \rightarrow i}(x_i)$$

Max-product belief propagation is a slight variation of BP

6 Learning for Stereo

Since 2015 CNNs became popular for stereo. Two basic approaches: learning the matching cost and learning the entire predictor.

MC-CNN

Learn to compute the matching cost between to image patches with a CNN. Trained as a classification problem.

Direct Prediction of Disparity

Goal directly predict disparity and train end to end. Use a CNN with 2D encoder, cost volume, multi-scale 3D convolutions, and 3D transposed convolutions. Approaches also profit from ideas like Gaussian pyramids.

7 Image Restoration

Image restoration has multiple applications, such as image denoising, image inpainting, or super-resolution.

Image Denoising

More or less all images exhibit image noise. Image denoising aims to remove or reduce the amount of noise.

Image noises: shot noise (photons per-pixel noise), thermal noise, bias noise from amplifiers.

Classical approaches: linear filtering (e.g. Gauss filtering), meadian filtering, wiener filtering.

Modern approaches: PDE-based approaches, Wavelet approaches, MRF-based techniques, DNN.

Filtering

Linear Filtering and Median Filtering: both approaches remove noise, but the image gets blurred at edges.

Bilateral Filter: Make filter dependent on the image. $T(x) = \frac{1}{W_x} \sum_{y \in \Omega_x} N(y) \cdot \mathcal{N}(\|x - y\|; 0, \sigma) \mathcal{N}(\|N(x) - N(y)\|; 0, \sigma_I)$.

Non-local Means Filter: takes a mean of all pixels in the image, weighted by how similar these pixels are to the target pixel.

$$T(x) = \frac{1}{W_x} \sum_{y \in \Omega_x} N(y) \cdot \mathcal{N}(\|B(x) - B(y)\|; 0, \sigma_P)$$

Denoising as Probabilistic Inference

Model denoising as posterior $p(\text{true image}|\text{noisy image}) = p(\mathbf{T}|\mathbf{N})$.

Likelihood of noisy given true image

Prior for all true image

Likelihood

Assume conditional independence $p(\mathbf{N}, \mathbf{T}) = \prod_{i,j} p(N_{ij} | T_{ij})$

We can assume some distribution e.g. Gaussian $p(N_{ij} | T_{ij}) = \mathcal{N}(N_{ij} - T_{ij} | 0, \sigma^2)$

Typically works good if noise is really independent (not the case in some HD images). Also suboptimal when noise is non-adaptive.

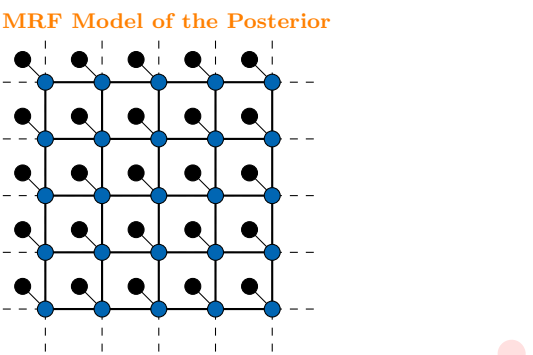
Prior

Statistics of natural images image derivative distribution sharp peak at zero and heavy tails (similar to Laplace or Student-t).

MRF Model of Image Prior

$$p(\mathbf{T}) = \frac{1}{Z} \prod_{i,j} f_H(T_{i,j}, T_{i+1,j}) \cdot f_V(T_{i,j}, T_{i,j+1})$$

Use Student-t $f_h = \left(1 + \frac{1}{2\sigma^2} (T_{i,j} - T_{i+1,j})^2\right)^{-\alpha}$



Blue nodes: pixels of the true image (hidden)

Black nodes: pixels of the noisy image (observed)

Likelihood: edges from black to blue nodes

Prior: edges between blue nodes

Natural images have many **complex** properties that are not modeled with the simple MRF. Natural images have scale-invariant statistics, that the model does not.

Inference

Typically gradient techniques work quite well for inference. Graph cuts or belief propagation can also be performed.

Image Inpainting

Goal fill in a missing part of the image.

General approach is similar to denoising (model $p(\mathbf{T}|\mathbf{C})$)

Image Super-Resolution

Goal increase the resolution of an image, whole making the high-resolution image seem sharp and natural. Simple solution is bicubic interpolation.

Yet again apply Bayes rule $p(\mathbf{H}|\mathbf{L}) = \frac{p(\mathbf{L}|\mathbf{H})p(\mathbf{H})}{P(\mathbf{L})}$.

Use the same **prior** as in denoising

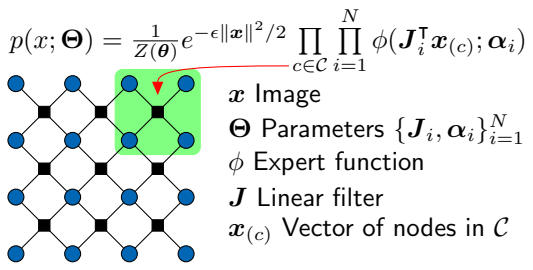
Likelihood one low resolution corresponds to a 2×2 patch in the high-resolution image. Conditional independence is assumed.

$$p(\mathbf{L}|\mathbf{H}) = \prod_{i,j} p(L_{i,j} | H_{2i,2j}, H_{2i+1,2j}, H_{2i,2j+1}, H_{2i+1,2j+1})$$

Gradient methods don't work very well, because the likelihood is more complex than i denoising or inpainting. Belief propagation typically used.

8 Higher-order MRFs

Idea relax strong conditional independence (Markov) assumption. Better model natural image statistics (responses to larger random linear filter are also heavy-tailed).



Problems intractable partition function $Z(\Theta)$, parameter learning necessary.

MRF Learning

Markov Network $p(x|\Theta) = \frac{1}{Z(\Theta)} \prod_c \phi_c(x_{(c)}|\Theta_c)$

Partition function $Z(\Theta) = \sum_x \prod_c \phi_c(x_{(c)}|\Theta_c)$

Training data $\mathbb{D} = \{x^1, \dots, x^N\}$

Log-Likelihood $\sum_n \sum_c \log \phi_c(x_{(c)}^n|\Theta_c) - N \log Z(\Theta)$

Log-Likelihood gradient can be somewhat derived. Problems the gradient depends on $p(x_{(d)}|\Theta)$, average needs to be computed over the model distribution $p(x|\Theta)$. **Learning requires inference.**

Generative Learning

Difficult due to intractable normalization. Image prior independent of application. Training only requires good natural images.

Bayesian Decision Theory

Minimize expected loss to choose restored image $\hat{x} = \arg \min_{x'} \int_x \Delta(x', x) \cdot p(x|y) dx$

Loss function $\Delta(x', x)$

Posterior distribution $p(x|y) \propto p(y|x)p(x)$

Choice of loss function determines decision of the restored image.

MAP estimation implies 0 – 1 loss

$$x_{\text{MAP}} = \arg \max p(x|y), \Delta(x', x) \begin{cases} 0 & \text{if } x' = x \\ 1 & \text{if } x' \neq x \end{cases}$$

Minimum MSE estimation

$$x_{\text{MMSE}} = \arg \min_{x'} \int_x \|x' - x\|^2 p(x|y) dx = \arg \min_{x'} \left[\|x'\|^2 - 2 \left(\int_x x \cdot p(x|y) dx \right)^T x' \right] = \mathbb{E}(x|y)$$

Generative vs. Discriminative

Generative

- model $p(x)$
- train on natural images
- application neutral
- learning & inference is more difficult

Discriminative

- model $p(x|y)$
- train on input/output pairs
- application specific
- learning & inference are generally easier
- can lead to better performance

Discriminative Training Training

Two tpyes of discriminative training, probabilistic CRF training via conditional likelihood maximization, loss-based training (not probabilistic)

Probabilistic training has the problem of intractable normalization constant

Loss minimization nested optimization problem, often slow.

Image Restoration in Academia

Challenge work with very large images e.g. 8MP Shrinkage Fields is an approach with discriminative training with simplified learning. (use shrinkage functions instead of potentials, scales good)

9 Optical Flow

Optical flow field image $I(x, y, t)$, horizontal flow $u(x, y)$, and vertical flow $v(x, y)$.

Brightness constancy

$I(x + u(x, y), y + v(x, y), t + 1) = I(x, y, t)$ (assumption 1)

Optical flow is highly ill-posed. Aperture problem (only normal velocity perpendicular to edge can be measured).

Neighboring points in the scene typically belong to the same surface and hence typically have similar 3D motions (spatial coherence).

Lucas-Kanade method

$$\mathbf{u} = -\left(\sum_R \nabla I \nabla I^\top\right)^{-1} \left(\sum_R I_t \nabla I\right)$$

Applied using a small window R , allows for dense optical flow estimation (combined with coarse-to-fine estimation and iterative warping).

Problems window is too big \Rightarrow discontinuities in the flow are smoothed over. Window too small \Rightarrow bad flow estimation due to not enough image information. LK is a **local** optical flow method.

Horn & Schunk

Classical **global** optical flow technique developed by Horn & Schunk.

Optical flow estimation as a problem of energy minimization

$$E(u, v) = \int (I(x + u(x, y), y + v(x, y), t + 1) - I(x, y, t))^2 + \lambda (\|\nabla u(x, y)\|^2 + \|\nabla v(x, y)\|^2) dx dy$$

Combination of brightness difference between corresponding pixels with quadratic penalty and regularization of the gradient magnitudes of the flow.

Problem energy is non-convex. **Solution** linearize the brightness constancy assumption.

Problem linearization only works for small motions, hence coarse-to-fine refinement with warping.

Problems with H&S

Results are often a bit better than with LK.

General problem flow is very smooth. This is because of the quadratic penalty to penalize changes in the flow. This does not allow for discontinuities.

Probabilistic Formulation

Posterior for optical flow estimation (observation model and prior on the flow field)

$$p(u, v | I^0, I^1) \propto p(I^1 | u, v, I^0) \cdot p(u, v | I^0).$$

Assume conditional independence (Likelihood)

$$p(I^1 | u, v, I^0) = \prod_{i,j} p(I^1 | u_{ij}, v_{ij}, I^0_{ij}) = \prod_{i,j} p(I^1(i + u_{ij}, j + v_{ij}) | I^0_{ij}).$$

Use robust observation model.

Assume in prior that horizontal and vertical flow are independent and use MRFs

$$p(u, v) = p(u) \cdot p(v) \propto \prod_{i,j} f_H(u_{i,j} - u_{i+1,j}) \cdot$$

$$f_V(u_{i,j} - u_{i,j+1}) \cdot f_H(v_{i,j} - v_{i+1,j}) \cdot f_V(v_{i,j} - v_{i,j+1})$$

if potential are Gaussians, taking the negative log results in H&S.

Better potential function, allowing for discontinuities can be used, such as Student-t.

$$f_H(u_{i,j} - u_{i+1,j}) = \left(1 + \frac{1}{2\sigma^2}(u_{i,j} - u - i + 1, j)^2\right)^{-\alpha}$$

This can be motivated by looking at the statistics of the optical flow. Very similar to a Student-t distribution.

Optical Flow Datasets

Need for ground truth optical flow data. Flow cannot be measured directly.

Use set of natural scene geometries and natural camera motions.

Flow Estimation

For flow estimation with likelihood and prior probabilistic inference is needed.

Discrete optimization is difficult because representing 2D flow vectors discretely is difficult.

Continuous optimization is also difficult because of non-convexity.

Solutions deterministic annealing (start with a quadratic optimization problem and gradually increase the difficulty).

Current Trends

Some trends non-local regularizers, piecewise parametric models (super pixels), incorporating semantics (use semantic segmentation), and deep learning.

Non-Local Regularizers

Idea use larger neighborhoods in the regularizer (i.e. non-local).

Solution use adaptive weights similar to bilateral filters.

Deep Learning for Optical Flow

In the recent years (starting 2015) deep learning approaches for optical flow estimation have emerged. These approaches are typically trained supervised on synthetic data and fine-tuned on real data (only very limited ground truth labels available).

Deep learning approaches took a long time to outperform conventional CV methods.

FlowNet

FlowNet is the first application of deep learning to optical flow (supervised).

Input: two RGB images, Output: optical flow field (4 times smaller than the input)

FlowNetS(imple) standard fully convolutional encoder-decoder network with skip connections and multi-scale loss.

FlowNetC(orrelation) more advanced CNN. Both images are encoded separately and correlated $c(\mathbf{x}_1, \mathbf{x}_2) = \sum_{o \in [-k, k] \times [-k, k]} \langle f_1(\mathbf{x}_1 + o), f_2(\mathbf{x}_2 + o) \rangle$ in the latent space. Encoding afterwards and decoding similar to FlowNetS. Flow prediction better but more compute needed compared to FlowNetS.

Problems bad generalization on other datasets due to mismatch between training and test data. Low resolution output (4 times lower).

Synthetic Data

Ground truth optical flow is very hard to obtain. Synthetic datasets have been produced to overcome this bottleneck.

Flying Chairs Real images as background, chair models as foreground. Ground truth flow over 2D

affine motion for background and foreground obtained.

FlowNet 2

Idea stack multiple networks to refine flow, use subnetwork for small displacements, optimized and complicated training schedule.

Architecture FlowNet2 stacks multiple FlowNetC/FlowNetS networks. Warping between second image with intermediate flow. Differentiable warping layer used.

Training is performed on multiple datasets (FlyingChairs, FlyingThings3D, order important). Optional fine-tuning.

Drawbacks long runtime (20ms (FlowNet) vs 120ms (FlowNet 2)), complicated training procedure

PWC-Net

Idea build domain knowledge into the network architecture. Use pyramids, warping, and cost volume (PWC-Net). Very light-weight network with fast runtime and strong performance.

Architecture Pyramid features, each stage composed of warping layer, cost volume layer, optical flow estimator and upsampling. Refines flow in each decoder stage.

PWC-Net+ use robust loss, changed data augmentation, disrupted learning rates for fine-tuning.

Iterative Residual Refinement

Idea iteratively refine residual flow by using the same network multiple times in a recurrent fashion. FlowNet used. Performance comparable to FlowNet 2 with less parameters due to weight sharing of iterative refinement.

Approach can also be applied to PWC-Net (share across scales).

Motivation based on scale invariant statistics of natural images

Occlusion estimation improves flow prediction further (use forward and backward flow).

UnFlow

Idea train a deep neural network without ground truth labels (unsupervised/self-supervised learning). Employ energy-based optical flow modeling applied for learning.

Training bidirectional flow estimation. Estimate occlusions. Use warping and data loss (only in non-occluded regions, census transform). Use consistency loss forward and backward flow. Use smoothness loss (second-order regularizer).

10 Tracking

Overview tracking is the task of finding the motion of an object in an image sequence.

Tracking cases tracking rigid objects, tracking articulated objects, and tracking fully non-rigid objects.

Challenges fast motions, changing appearance, changing object pose, dynamic backgrounds, and occlusions.

Simple solution tracking by detection of a single object. Simply find the object in the current frame.

Simple solution multiple objects data association is needed. Hungarian matching and detection possible simple solution.

Gradient-Based Tracking

Take matching function and compute the gradient w.r.t. the object motion between frames.

SSD-matching against a simple template (similar to LK)

$$E_{SSD}(\mathbf{x}, \mathbf{y}) = \sum_{r,c} (I(\mathbf{x} + c, \mathbf{y} + r) - T(c, r))^2$$

$\frac{\partial}{\partial \mathbf{x}} E_{SSD}(\mathbf{x}, \mathbf{y})$ and $\frac{\partial}{\partial \mathbf{y}} E_{SSD}(\mathbf{x}, \mathbf{y})$ can be computed by linearizing the brightness constancy assumption.

Assumptions brightness constancy constraint and small motions.

Bayesian Tracking

$\mathbf{x}_k \in \mathbb{R}^d$: internal state at the k -th frame.

$\mathbf{X}_k = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k]^\top$ history up to time step k

$\mathbf{z}_k \in \mathbb{R}^c$: measurement at the k -th frame.

$\mathbf{Z}_k = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k]^\top$ history up to time step k

Goal estimate the posterior $p(\mathbf{x}_k | \mathbf{Z}_k)$

Approach recursive estimation

$$p(\mathbf{x}_{k-1} | \mathbf{Z}_{k-1}) \Rightarrow p(\mathbf{x}_k | \mathbf{Z}_k)$$

$$p(\mathbf{x}_k | \mathbf{Z}_k)$$

$$= p(\mathbf{x}_k | \mathbf{z}_k, \mathbf{Z}_{k-1})$$

$$\propto p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{Z}_{k-1}) p(\mathbf{x}_k | \mathbf{Z}_{k-1}) \text{ Bayes rule}$$

$$\propto p(\mathbf{z}_k | \mathbf{x}_k) \cdot p(\mathbf{x}_k | \mathbf{Z}_{k-1}), p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{Z}_{k-1}) = p(\mathbf{z}_k | \mathbf{x}_k)$$

$$\propto p(\mathbf{z}_k | \mathbf{x}_k) \cdot \int p(\mathbf{x}_k, \mathbf{x}_{k-1} | \mathbf{Z}_{k-1}) d\mathbf{x}_{k-1} \text{ marginal.}$$

$$\propto p(\mathbf{z}_k | \mathbf{x}_k) \cdot \int p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{Z}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{Z}_{k-1}) d\mathbf{x}_{k-1}$$

$$\propto k p(\mathbf{z}_k | \mathbf{x}_k) \cdot \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{Z}_{k-1}) d\mathbf{x}_{k-1}$$

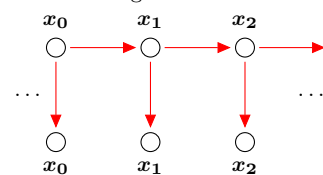
$p(\mathbf{x}_k | \mathbf{Z}_k)$ posterior probability at the current t. step

$p(\mathbf{z}_k | \mathbf{x}_k)$ likelihood

$p(\mathbf{x}_k | \mathbf{x}_{k-1})$ temporal prior

$p(\mathbf{x}_{k-1} | \mathbf{Z}_{k-1})$ posterior probability at the previous time step

k normalizing term



Assumptions: $p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{Z}_{k-1}) = p(\mathbf{z}_k | \mathbf{x}_k)$,

$$p(\mathbf{x}_k | \mathbf{x}_{k+1}, \mathbf{Z}_{k-1}) = p(\mathbf{x}_k | \mathbf{x}_{k-1}),$$

$$p(\mathbf{x}_k | \mathbf{X}_{k-1}) = p(\mathbf{x}_k | \mathbf{x}_{k-1})$$

Kalman Filter

If both likelihood and prior are Gaussian a closed form solution exists and the two estimators (posterior mean & MAP) are the same. Known as Kalman filter.

Likelihood

Assume a linear generative model of the observations.

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{q}_k$$

$$p(\mathbf{z}_k | \mathbf{x}_k) = \mathcal{N}(\mathbf{H}_k \mathbf{x}_k, \mathbf{Q}_k)$$

Prior

Assume linear model for the temporal evolution

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k+1} + \mathbf{w}_k$$

$$p(\mathbf{x}_k | \mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{A}_k \mathbf{x}_{k-1}, \mathbf{W}_k)$$

Update

Time update

Prior estimate

$$\mathbf{x}_k^- = \mathbf{A}_k \mathbf{x}_{k-1}$$

Error covariance

$$\mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^\top + \mathbf{W}_k$$

Measurement Update

Posterior estimate

$$\mathbf{x}_k = \mathbf{x}_k^- + \mathbf{K}_k \left(\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k^- \right)$$

Error covariance

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-$$

Kalman gain

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^\top \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^\top + \mathbf{Q}_k \right)^{-1}$$

Limitations

Can model fast motions and changing appearance.

Changing object pose is not really captured.

Kalman filter is restricted by assuming linear state and output transformations, as well as Gaussian noise.

Particle Filtering

Particle filtering uses multi-model posteriors.

Posterior is approximated by non-parametric approach. Using discrete set of samples (or particles) each of which has a weight

$$S = \{(\mathbf{x}^{(i)}, w^{(i)}); i = 1, \dots, N\}$$
 with $\sum_{i=1}^N w^{(i)} = 1$.

Can be converted back to a continuous representation by using Gaussian mixtures.

Compute integral of Bayesian tracking formula by

Monte-Carlo approximation.

Limitations & Properties

Infinite particle limit converges to the correct solution.

Many variants of the general procedure are available, some of which are more efficient.

Articulated Tracking

Goal track the position and the configuration of an articulated object, i.e. an object that consists of several parts (human tracking).

Mocap (marker-based) motion capturing of human motion by special markers fixed to a human and recorded by special cameras. Marker-based Mocap can be tedious and only works in lab settings.

Mocap (marker-less) use tracking techniques from CV to perform Mocap without the need of markers. Typically, not as robust as marker-based. Can be performed by using particle filtering.

11 Image Segmentation

Definition segmentation can roughly be described as the grouping of similar information in an image.

Application segmentation maps can be useful for other CV tasks like scene understanding, motion estimation etc. but also for applications like medical image analysis.

Figure-Ground Separation

Figure-ground separation describes the task of segmenting a foreground figure from the background.

Superpixels

Superpixels is a type of segmentation, in which the goal is to group pixels into small segments (superpixels).

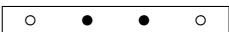
Efficient approach SLIC iteratively merging in CIELAB color space.

Gestalt Factors

Approach to answer the question, what belongs together?

 Not grouped

 Proximity

 Similarity

 Similarity

 Common Fate

 Common Region

Additional gestalt factors: parallelism, continuity, symmetry, closure.

Segmentation by Clustering

Simple approach for segmentation. Represent each pixel as a feature vector and cluster these.

Feature vector may include: position, pixel intensity, pixel color, or a description of the local texture.

Agglomerative Clustering

Each pixel is a separate cluster. Merge two clusters with the smallest inter-cluster distance until the clustering is satisfied.

Divisive Clustering

All pixels are in one cluster. Split clusters that yields two clusters with largest inter-cluster distance until the clustering is satisfied.

Other Clustering Methods

k-Means clustering define number of clusters and iterative update cluster assignments.

Mean-Shift clustering mean shift procedure estimates a density. Estimates clusters based on local maximums. Number of clusters have not to be set by hand.

Graph-based Clustering

Interpret clustering as cutting a graph in which each node represents a pixel into pieces. Node weights are affinities (similarities) between pixels.

Affinities: distance between, intensity difference, color distance, or texture (much more approaches available).

Affinity Matrix: includes all pairwise affinities.

Graph-Cut based Segmentation by finding the min-cut on the graph.

Normalized Cuts normalize the cut to remove bias. (NP hard)

Correct Segmentation

There is no general right (ground truth) segmentation. A segmentation can only be right for a certain purpose.

Different people may segment images differently. This makes evaluation hard.

Interactive Segmentation

Idea let the user annotate some examples of foreground and background.

Energy-Based Segmentation

Mumford-Shah function, approximates an image f with a smooth function u and explicit discontinuities C .

$$E(u, C) = \int_{\Omega} (f - u)^2 dx + \alpha^2 \int_{\Omega - C} |\Delta u|^2 dx + v \|C\|$$

$\|C\|$ denotes the boundary length. Trade-off between describing the image well and having a short boundary.

Properties: without discontinuities similar to denoising approach. Regularizer is spatially continuous. To implement it, it has to be spatially discretized. Level-set methods are used for approximation. Hard to optimize.

Level-Set Methods

Idea Introduce an additional variable (function) and model the segment boundaries as the zero crossings of that function. Similar to contour lines.

Shape Knowledge

Energy-based formulations can profit from shape-based prior knowledge.

Probabilistic Segmentation

Energy-based approaches can often be interpreted as probabilistic approaches.

Basic approach for (interactive) segmentation is a discrete MRF.

Energy formulation

$$E(S) = E_d(S; I) + \lambda \cdot E_s(S), \quad S \in \{0, 1\}^{m \times n}$$

Simple solution Potts model for $E_s(S)$.

Intensity/color histogram for

$$E_d(S; I) = \sum_k -\log h_{S_k}(I_k). \quad h_0 \text{ background, } h_1 \text{ foreground histogram.}$$

Inference use graph cuts.

Extension prior is agnostic of the image, use contrast-sensitive (Potts) model. Particular instance of a CRF.

Problems only binary segmentation, lots of interaction often needed, color model too unspecific.

Advanced approach iterated graph cuts. Iterate between determine histogram from current foreground and background. Segmenting the image with current likelihood.

Semantic Segmentation

Semantic Segmentation = Segmentation + Recognition
Segment the image and determine the category at every pixel.

Pixel Classification

Apply hand crafted filters and cluster the feature map (or use different classifier).

TextonBoost uses CRF and joint boosting for multi-class classification. Inference using graph cuts + alpha expansion.

CNNs for Semantic Segmentation

Simple approach, upsample latent space of classification network.

Current approach use decoder including upsampling (transposed convolution or upsampling), standard convolution, and skip connections. CNNs for semantic segmentation can profit from using CRFs.

12 Recent Research

Cityscapes Dataset

Overview the Cityscapes dataset is a huge dataset for segmentation (instance and semantic).

Scenarios Urban scenarios, complex scenes and stereo video.

Other datasets KITTI (stereo video, no official labeling), CamVid (monocular video), Daimler Urban Scenes (stereo video, limited labels and classes).

Objectives Multimodal input (HDR, LDR, 2MP cameras), stereo setup, 30 frame video snippets, long videos, GPS tracks.

Annotations 5000 images (2975 train, 500 val, 1525 test) with dense labels, 20000 images with weak labeling, many complex classes.

Playing for Data

Overview generate labeled dataset through computer games (GTA 4). Easy to get fast and accurate dense labels.

Single-stage Semantic Segmentation

Idea train a single stage segmentation model (CNN) only with image labels (weak supervision).

Key components score aggregation function for classification, self-supervised training on segmentation, regulariser for self-supervised training.

Self-Supervised Monocular Scene Flow Estimation

Idea train a network in a self-supervised fashion to predict scene flow from a monocular image, at inference time. Use stereo images for self-supervising depth. Use 3D point reconstruction loss. Apply losses only on visible pixels by using disocclusion cue.

Approach use an inverse problem view, use the property that optical flow is the projection of scene flow. Train variation of PWC-Net with adopted encoder. Use bi-directional estimation.

Neural Nearest Neighbor Networks

Idea model self-similarity and non-local processing in neural networks. Enlarge receptive field of CNNs.

Approach use a differentiable continuous nearest neighbors building block.

Results very effective in tasks like image denoising and correspondence classification.