Optical Flow through Deep Flow

Introduction

Introduction of Optical Flow:

Optical flow computation is a key component in many computer vision systems designed for tasks such as action detection or activity recognition. It has many important applications in computer vision, including slow motion video generation, object tracking etc. Starting from the famous Horn-Schunck algorithm (B. K. Horn and B. G. Schunck. Determining optical flow. Proc. SPIE 0281, Techniques and Applications of Image Understanding, 1981.
) and Lucas-Kanade algorithm (B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In IJCAI, volume 81, pages 674–679, 1981.
), researchers have conducted decades of pioneering research, achieving great development in this area.

Optical flow algorithm aims at computing pattern of apparent motion of objects. Given two consecutive images, we would like to compute how each pixel moves from the first image to the second image. Imagine there are three-dimensional vectors describing the motion of the objects in 3-D space. Ideally, the optical flow is the 2-D projection of the three-dimensional vectors on the image. In computer vision, we call this projected two-dimensional vector as flow field. There are four important assumptions for optical flow algorithm:

1. Brightness constancy: The image grey value of the pixel does not change during the motion.
2. Gradient constancy: it is useful to allow some small variations in the brightness because it often appears in natural scenes. So we assume that the gradient of the image grey value does not vary due to the displacement.
3. Smoothness: The movement is a smooth movement.
4. Small displacement: Between each pair of consecutive images, the corresponding pixels do not move too far.
5. Spatial coherence: points move like their neighbors within patches.

The most of optical flow algorithms develop their ideas on the basis of these assumptions.

The Deep Flow Algorithm:

Variational methods are the state-of-the-art family of methods for optical flow estimation. (I can list some citations and examples if needed), each with its own focus of special optimization in some aspects. Energy minimization is performed by solving the Euler-Lagrange equations, then reducing the problem to solving a sequence of large and structured linear systems.

One of the state-of-the-art algorithm is the DeepFlow algorithm (Citation see the website of deepflow). This algorithm has its focus on solving Large displacement in optical flow estimation. This property makes it ideal for slow motion video applications: In the scenario of generating slow motion videos, the original videos are likely to have large displacement between consecutive images.

The deep flow algorithm mainly has two parts:

1. Deep matching: The matching algorithm is used to compute the matching of corresponding pixels in consecutive frames. The deep matching algorithm builds upon a multi-stage architecture with about 6 layers (depending on the image size), interleaving convolutions and max-pooling, a construction akin to deep convolutional nets (Use the citation of the deep matching paper). The matching stage prepares for the flow computation stage and is not the main work of our project. Therefore we skip further explanations on this part.
2. Deep flow: Deep flow is a variational refinement method that blends the result from deep matching algorithm into an energy minimization framework. The energy that the algorithm minimizes is a weighted sum of data term E_D, smoothness term E_S and matching term E_M.

$$E(\boldsymbol{w}) = \int_\Omega E_D + \alpha E_S + \beta E_M \boldsymbol{dx}$$

I. Data term. It is a combination of separate penalization of the color and gradient constancy assumptions with a normalization factor.

$$E_D = \delta \Psi \left(\sum_{i=1}^{c} \boldsymbol{w}^\top \bar{\boldsymbol{J}}_0^i \boldsymbol{w} \right) + \gamma \Psi \left(\sum_{i=1}^{c} \boldsymbol{w}^\top \bar{\boldsymbol{J}}_{xy}^i \boldsymbol{w} \right)$$

II. Smooth term. Based on the smoothness assumption. It is a robust penalization of the gradient flow norm:

$$E_S = \Psi \left(\|\nabla u\|^2 + \|\nabla v\|^2 \right)$$

III. Match term. This term penalizes the difference between the flow and the precomputed matching vector from deep matching.

$$E_M = c\phi\Psi(\|\boldsymbol{w} - \boldsymbol{w}'\|^2).$$

Where:

$$\phi(\boldsymbol{x}) = \sqrt{\tilde{\lambda}(\boldsymbol{x})}/(\sigma_M\sqrt{2\pi}) \ \exp(-\Delta(\boldsymbol{x})/2\sigma_M).$$

The minimization of the functional is non-convex and non-linear.

TODO: I don’t quite know how to minimize this functional T_T, especially how to get a linear system. It is a pure math problem. ZONGREN could you deal with this part? See the last part of chapter in the Deep Flow paper.