Learning Filter-Based Motion Features for Dynamic Scene Analysis

Damien Teney, Martial Hebert

Carnegie Mellon University
The Robotics Institute

Overview

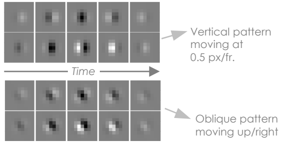
Motivation #1 Natural dynamic scenes include non-rigid objects, dynamic textures, (semi)transparencies, …
~~Optical~~ flow assumptions do not hold
→ **Motion analysis with spatiotemporal filters** [1,5]

Motivation #2 Success of video analysis with « 2-stream » CNNs: [3]
Spatial stream = appearance, fed with raw pixels
Temporal stream = motion, typically precomputed opt. flow
→ **This work: integrate motion/flow extraction into the convolutional framework**

Contribution Shallow CNN, building block for deeper architectures
Input = volume of raw pixels (stacked frames)
Output = optical flow
Intermediate layers capture more !
e.g. multiple transparent motions

Filter-based motion extraction

Classical method, applies 3D filters to the video volume of pixels: [2,4]



Typically hard-coded, for example: (left) Gaussian derivatives, 3D Gabors, …

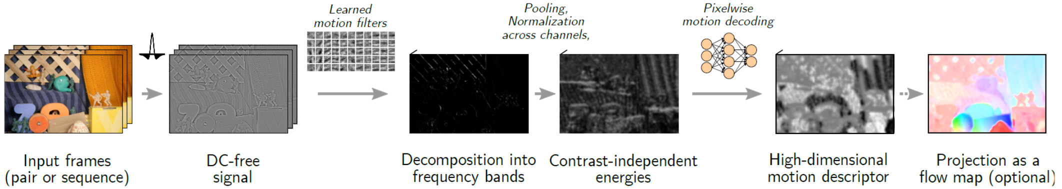
Bandpass filters decompose the signal in the freq. domain

This work: learn these filters

Translational motion in the image = energy along one plane in the frequency domain
Multiple transparent motions = energy along multiple planes

→ Recovery of motion(s) independent of appearance (contrast/texture/gradient ori.) possible through frequency analysis

Convolutional network architecture



Architecture similar to classical, biologically-inspired model of motion perception [2,4], mapping pixels → flow maps

Shallow network: single convolutional layer (spatiotemporal filters), then pixelwise decoding
Dense predictions: convolution/pooling stride of 1 pixel (overlapping pooling regions)

To recover motion **independent of appearance** (texture, contrast, …): provision for the required invariances
- brightness, contrast, gradient orientation : initial center-surround filter, normalization across filter responses
- spatial phase: pooling of filter responses
- translation, scale: pixelwise decoding, shared parameters when applied in multiscale manner

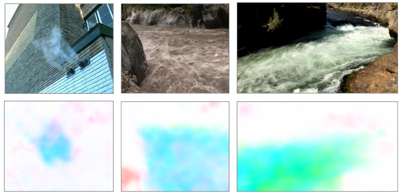
Training:
Trained with Middlebury optical flow dataset, using 5 frames as input
Relatively few parameters to train
Virtually unlimited augmentations: scalings, rotations, flips, …
Decoding initialized as if filters capture uniformly-distributed orientations

Test time:
Network applied on the input at multiple scales
Penultimate layer can capture multiple motions for each pixel

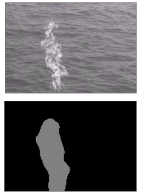
Experiments

Recovery of traditional optical flow: results comparable to classical techniques, even with purely local predictions: no smoothness/rigidness prior or regularizer !

Identification of motion in dynamic textures:
Transparent steam Rushing water (flicker, non-rigid)



Segmentation: k-means on features from penultimate layer



"ocean-fire" sequence

[1] K. G. Derpanis and R. P. Wildes. Spacetime texture representation and recognition based on a spatiotemporal orientation analysis. PAMI, 2012.
[2] D. J. Heeger. Model for the extraction of image flow. J. Opt. Soc. Am. A, 1987.
[3] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. NIPS Spotlight, 2014.
[4] F. Solari, M. Chessa, and P. Medathati, N. Kornprobst. What can we expect from a V1-MT feedforward architecture for optical flow estimation ?, Signal Processing: Image Communication, 2015.
[5] D. Teney and M. Brown. Segmentation of dynamic scenes with distributions of spatiotemporally oriented energies. In BMVC, 2014.

Work supported by the Belgian-American Educational Foundation (BAEF).