Large Displacement Optical Flow

Practical Course: GPU Programming in Computer Vision

O. Ganus[1] D. Korzh[1] Supervisor: M. Souiai[2]

[1]Chair of Scientific Computing
Technical University Munich

[2]Chair of Computer Vision Group
Technical University Munich

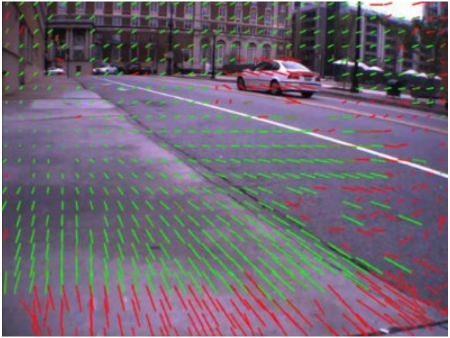
30.03.15 / GPU Programming

Outline

Outline

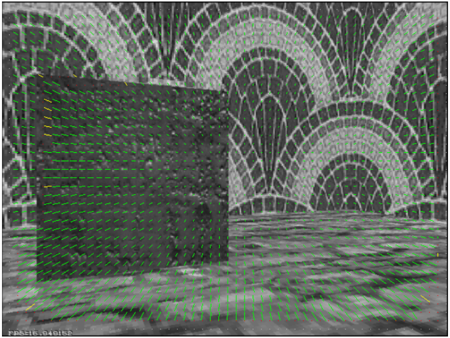
Optical Flow and it's Usage

- Optical flow is the pattern of motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene.



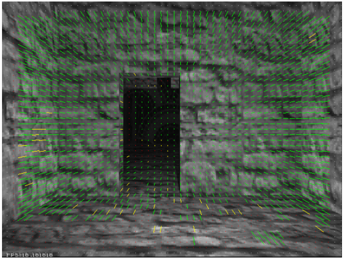
Robots — Targets

- Trailing a target: turn to the same side that shows more motion. It keeps the target in focus.
- Escaping from target: move backward, while keeping the object in focus.



Robots — Own Motion

- A scene from a simulated environment, doing one step forward from initial starting position.
- Compute an optical flow between each image.
- Information:layout of the surfaces, motion of the observation point.



Other Cases

- The registration of medical organs across different scans
- The matching of facial images for the purpose of recognition and the tracking of deformable objects
- Astronomy: motion of distant objects
- Biology: tracing the motion of the bacteria
- Computer games
- Video compression: saving memory

Why do we use GPU?

Speed (results from our project):

- CPU:
  1 iteration -> 4–5 min
- GPU:
  1 iteration -> 10 s

Short Description

- Our project based on the optical flow functional
- We will decouple it in two terms (a data term and regularizer)
- Then the problem can be solved by an alternation of two steps

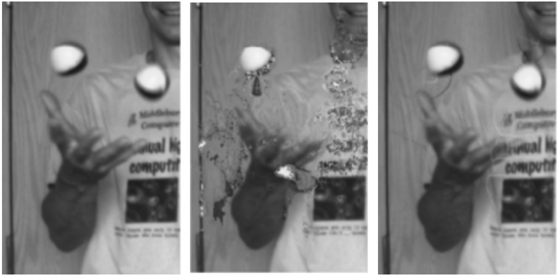
Variational Optical Flow Computation

- Determine for each point in one image an optimal corresponding point in the other image
- Use an energy minimization framework

$$E(v) = \underbrace{\int_\Omega \lambda\rho(v, x)\ d^2x}_{\text{Data Term}} + \overbrace{\int_\Omega R(v)\ d^2x}^{\text{Regularity Term}}$$

- Data term looks for points of similar intensity
- The regularity term (weighted by $\lambda > 0$) imposes spatial smoothness of the velocity field $v = (v1, v2)$

The Problem of Large Displacements



Original | Warping | Proposed

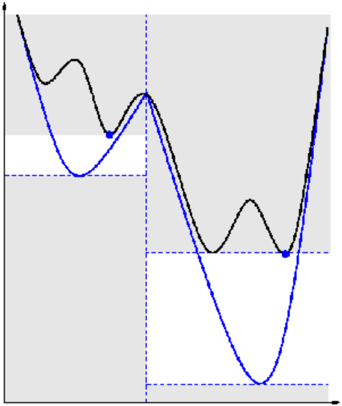
- Algorithm based on Warping do not recover large motions and motions of small scale structures.
- Proposed algorithm builds large-displacement optical flow even for small scale structures.

Outline

1. Introduction
   - Optical Flow and it's Usage
   - Variational Optical Flow Computation
   - The Problem of Large Displacements
2. Alternating Optimization
   - Iterative Algorithms
   - Pros and Cons of the Approach
3. Summary
4. Questions

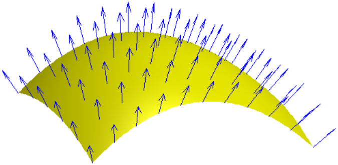
Data Term

- The major problem is that the data term is not convex in v. Hence, the quality of of the solution depends on the strategy of minimisation.



Additional Vector Field

- We introduce an additional vector field u: $\Omega \rightarrow R^2$ and decouple the problem making alternative minimisation possible.

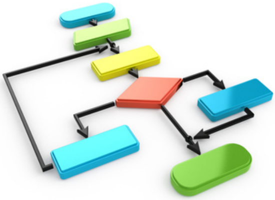


Alternating Optimization

- Alternatingly compute $\upsilon(\mathrm{x})$ with fixed $\mathrm{u}(\mathrm{x})$, and the same with respect to u for fixed υ.
- Decrement Θ forcing u and υ to converge at the end.

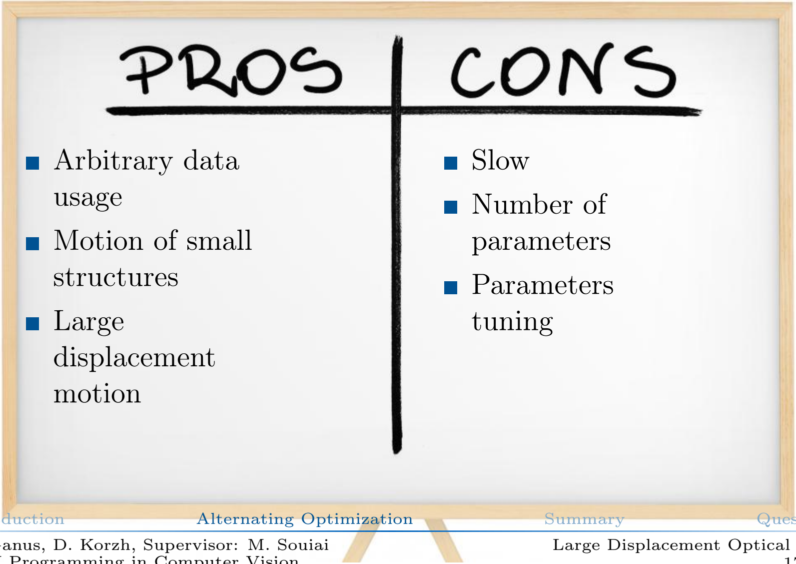
$$E(\upsilon, u) = \int_{\Omega} \underbrace{\lambda\rho(\upsilon, x) + \overbrace{\frac{1}{2\Theta} C(\upsilon - u)}^{\text{Regularity Term}}}_{\text{Data Term}} \overbrace{+ \psi(\nabla u)}\, d^2x$$

Iterative Algorithms



- Gradient descent: $C(v - u) = (v - u)^2$
- Primal-dual with quadratic decoupling: $C(v - u) = (v - u)^2$
- Primal-dual with L_1-penalization: $C(v - u) = |v - u|$

Pros and Cons of the Approach



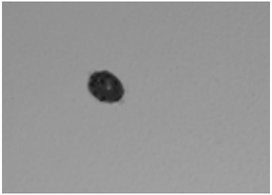
Pros

- Arbitrary data usage
- Motion of small structures
- Large displacement motion

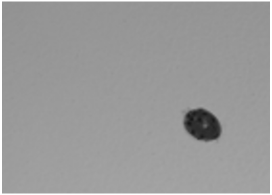
Cons

- Slow
- Number of parameters
- Parameters tuning

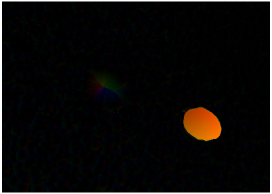
Results with Ladybug



Input 1



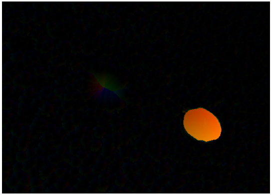
Input 2



Flow

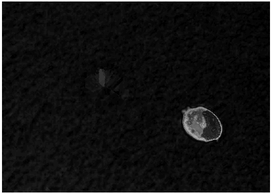
Result

Ladybug Before and After Smoothness

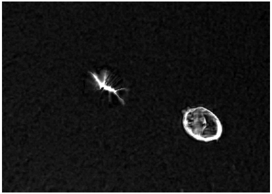


Before

After

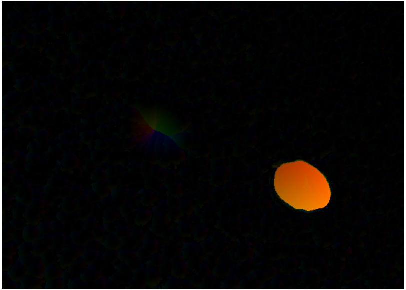


Error before

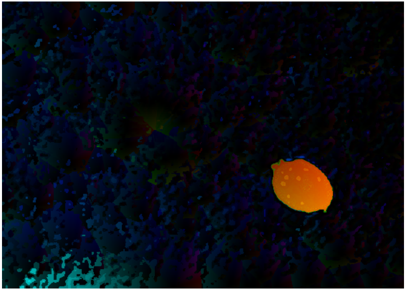


Error after

Primual Dual Algorithm with Quadratic Decoupling or with L_1-Penalization



Quadratic Decoupling



L_1-Penalization

Results with Walking Man

Input 1

Input 2

Flow

Result

Walking Man Before and After Smoothness

Before

After

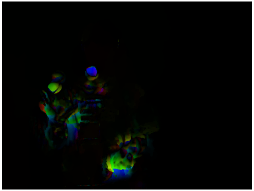
Error before

Error after

Results with Beanbags

Input 1

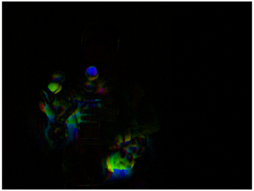
Input 2



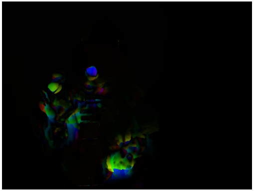
Flow

Result

Beanbags Before and After Smoothness



Before



After

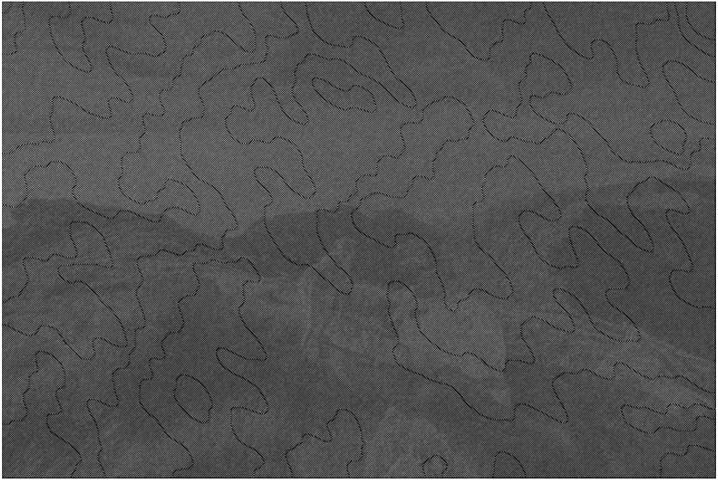
Error before

Error after

Playing with Parameters

- If the value of the theta parameter is not enough big, then we are getting such results.

Bugs in Code



Outline

Summary

- Primal-Dual algorithm is faster then Gradient Descent algorithm.
- L1-penalization algorithm showed no improvement compared with Quadratic Decoupling.
- Use of the array of random velocities gave unpredictable results, depending on the size of the array.
- Complete Search algorithm as itself gives proper optical flow, usage of the regularity term (second minimisation) makes flow looking better, but at the same time it increases errors.

Outline

Questions? Thank you for your Attention!