



# Une approche itérative rapide pour le calcul de flot optique par corrélation

Guy Le Besnerais, Frédéric Champagnat,  
Aurélien Plyer, Riadh Fezzani  
ONERA/DTIM

Images et Télédétection, estimation et analyse de champs vectoriels  
8 janvier 2009

**ONERA**  
THE FRENCH AEROSPACE LAB

retour sur innovation

## Plan

- Introduction : flot optique et recalage de fenêtres
- Un schéma itératif rapide
- Perspectives

## Problématique du flot optique

- Objectif
  - « Superposer » deux images  $\mathfrak{I}_1(s)$  et  $\mathfrak{I}_2(s)$ ,  $s \in S \subset \mathbb{R}^2$  par un champ de mouvement  $s \rightarrow s + \mathbf{u}(s)$  (flot optique)
- Modèle d'observation du flot optique
  - acquisition des images 
$$\begin{cases} I_1(\mathbf{k}) = \mathfrak{I}_1(\mathbf{k}\Delta) + b_1(\mathbf{k}) \\ I_2(\mathbf{k}) = \mathfrak{I}_2(\mathbf{k}\Delta) + b_2(\mathbf{k}) \end{cases}$$
  - conservation de la luminance 
$$\mathfrak{I}_1(s) = \mathfrak{I}_2(s + \mathbf{u}(s))$$
- NB : il s'agit d'un modèle non linéaire implicite
  - $y$  : images continues  $\mathfrak{I}_1, \mathfrak{I}_2$
  - $z$  : images observées  $I_1, I_2$
  - $f$  : échantillonnage et bruit
  - $A$  : contrainte issue de la conservation de luminance

### Deux approches principales

- Approche « variationnelle »
  1. critère fonctionnel
  2. schéma discret de résolution
    - majoritaire en flot optique depuis [Horn-Schunck81]

$$1. \quad J(\mathbf{u}; \mathfrak{I}_1, \mathfrak{I}_2)$$

$$2. \quad \nabla J_{\mathbf{u}} = 0 \rightarrow \{\mathbf{u}_t\}_t$$

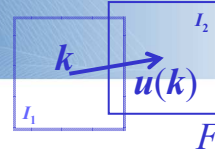
- Approche « directe »
  1. interpolation
  2. optimisation en dimension finie
    - majoritaire en paramétrique depuis [Lucas-Kanade81]

$$1. \quad \tilde{I}_2(s) = \sum_k I_2(k) \gamma(s - k)$$

$$2. \quad \min_{\mathbf{u}} J(\mathbf{u}; I_1, \tilde{I}_2)$$

- NB : cette dichotomie se retrouve dans les problèmes inverses non linéaires « classiques » (exemple : tomographie)

## Recalage de fenêtres locales

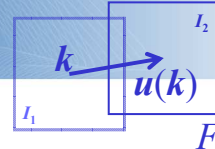


$$J(\mathbf{u}(\mathbf{k}), I_1, \tilde{I}_2) = \sum_m F(\mathbf{m} - \mathbf{k}) (\tilde{I}_2(\mathbf{m} - \mathbf{u}(\mathbf{k})) - I_1(\mathbf{m}))^2$$

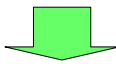
1. Le critère d'estimation d'un champ constant par morceaux  
 ⇒ Recalage local ou de fenêtre = *block matching*

- En général associé à une mise en œuvre non dense :
  - sélection ou sous-échantillonnage des positions  $\mathbf{k}$  sur lesquelles on calcule  $\mathbf{u}$
- Méthode « ubiquiste » : compression, métrologie, vision...
  - Algorithme Lucas-Kanade en vision (Simoncelli, Bouguet)

## Recalage de fenêtres locales



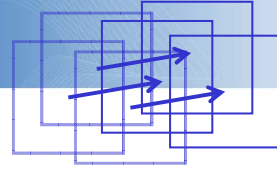
$$\mathfrak{I}_1(\mathbf{s}) = \mathfrak{I}_2(\mathbf{s} + \mathbf{u}(\mathbf{s}))$$



$$J(\mathbf{u}(\mathbf{k}), I_1, \tilde{I}_2) = \sum_m F(\mathbf{m} - \mathbf{k}) (\tilde{I}_2(\mathbf{m} - \mathbf{u}(\mathbf{k})) - I_1(\mathbf{m}))^2$$

2. Version intégrée et discrète de la conservation de la luminance  
 ← prise en compte du *modèle implicite*

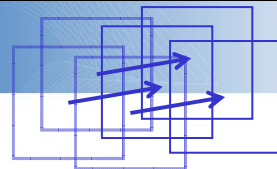
## Recalage de fenêtres locales



$$J(\mathbf{u}, I_1, \tilde{I}_2) = \sum_k \sum_m F(\mathbf{m} - \mathbf{k}) (\tilde{I}_2(\mathbf{m} - \mathbf{u}(\mathbf{k})) - I_1(\mathbf{m}))^2$$

3. Le modèle d'observation d'un estimateur de flot dense  
← prise en compte du *modèle implicite*
  - sous-jacent aux versions denses de LK (Simoncelli, Bouguet)

## Recalage de fenêtres locales



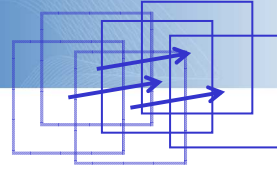
$$J(\mathbf{u}, I_1, \tilde{I}_2) = \sum_k \sum_m F(\mathbf{m} - \mathbf{k}) (\tilde{I}_2(\mathbf{m} - \mathbf{u}(\mathbf{k})) - I_1(\mathbf{m}))^2 + \lambda R(\mathbf{u})$$

2. Le modèle d'observation d'un estimateur de flot dense  
← prise en compte du *modèle implicite*
  - sous-jacent aux versions denses de LK (Simoncelli, Bouguet)

⇒ Coopération locale-globale

  - Proposée en approche variationnelle par [Bruhn *et al.* 05]

## Recalage de fenêtres locales



$$J(\mathbf{u}, I_1, \tilde{I}_2) = \sum_{\mathbf{k}} \sum_{\mathbf{m}} F(\mathbf{m} - \mathbf{k}) (\tilde{I}_2(\mathbf{m} - \mathbf{u}(\mathbf{k})) - I_1(\mathbf{m}))^2$$

### 2. Le modèle d'observation d'un estimateur de flot dense

← prise en compte du *modèle implicite*

- sous-jacent aux versions denses de LK (Simoncelli, Bouguet)

## Flot dense et recalage local

- Gauss-Newton 
$$\sum_{\mathbf{k}} \sum_{\mathbf{m}} F(\mathbf{m} - \mathbf{k}) (\tilde{I}_2(\mathbf{m} - \mathbf{u}(\mathbf{k})) - I_1(\mathbf{m}))^2$$

- DL du critère NL de corrélation en  $\mathbf{u}(\mathbf{k}) = \mathbf{u}_l(\mathbf{k}) + \delta\mathbf{u}(\mathbf{k})$

- 1ère itération

$$\mathbf{u}_0 = 0 \Rightarrow \sum_{\mathbf{k}} \delta\mathbf{w}(\mathbf{k})^t (F * \nabla_3 \tilde{I}_2 \nabla_3^t \tilde{I}_2)(\mathbf{k}) \delta\mathbf{w}(\mathbf{k})$$

- critère linéaire séparable
- coefficients ← convolutions
- ⇒ « forme convolutive » rapide

avec

$$\delta\mathbf{w}(\mathbf{k}) = \begin{bmatrix} \delta\mathbf{u}(\mathbf{k}) \\ 1 \end{bmatrix}$$

$$\nabla_3 \tilde{I}_2(\mathbf{m}) = \begin{bmatrix} \nabla \tilde{I}_2(\mathbf{m}) \\ \tilde{I}_2(\mathbf{m}) - I_1(\mathbf{m}) \end{bmatrix}$$

## Flot dense et recalage local

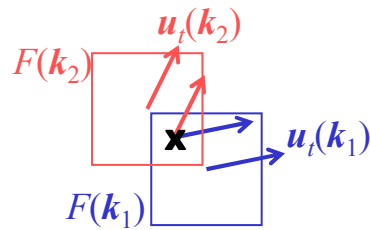
- Gauss-Newton
- 1ère itération rapide
- Itérations suivantes : difficultés

$$u(k) = u_t(k) + \delta u(k)$$

### 1. LK dense en chaque pixel

- un gros souci : interpolations multiples du même pixel

⇒ lourd



## Flot dense et recalage local

- Gauss-Newton
- 1ère itération rapide
- Itérations suivantes : difficultés

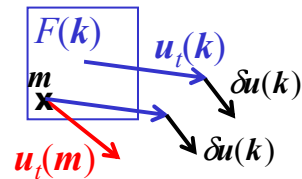
$$u(k) = u_t(k) + \delta u(k)$$

### 2. Schéma itératif rapide par déformation d'image

- réappliquer la 1ère itération rapide entre  $I_1$  et  $\tilde{I}_2^{(t)} = \tilde{I}_2(\bullet + u_t)$
- ⇒ IWS : Iterated Warping Scheme

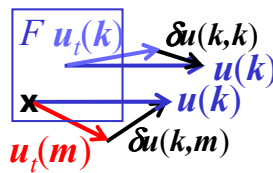
⇒ schéma divergent !

- contradiction modèle d'observation et déformation d'image



## Flot dense et recalage local

- Un schéma itératif rapide (*Modified IWS*)  $u(k) = u_i(k) + \delta u(k)$ 
  - comme *IWS* : recalculer  $I_1$  et  $\tilde{I}_2^{(t)} = \tilde{I}_2(\bullet + u_i)$
  - un nouveau développement en  $u(k) = u_i(m) + \delta u(k, m)$



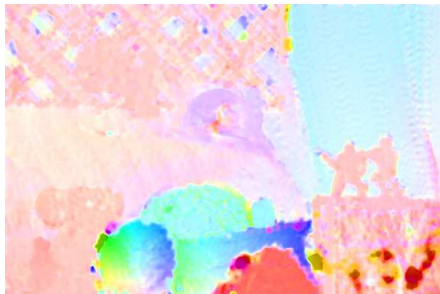
⇒ Algorithme FOLKI [LeBesnerais-Champagnat05]

## Convergence : illustration

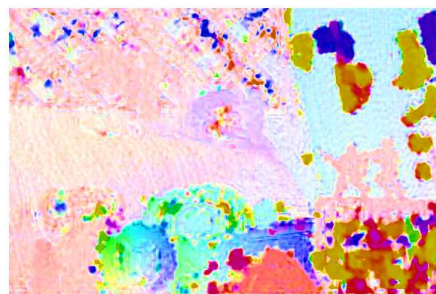
- FOLKI convergence comparée à IWS
  - Exemple Army, site d'éval. Middlebury/flow



encodage couleur



FOLKI fenêtre gaussienne sigma=2



IWS fenêtre gaussienne sigma=2

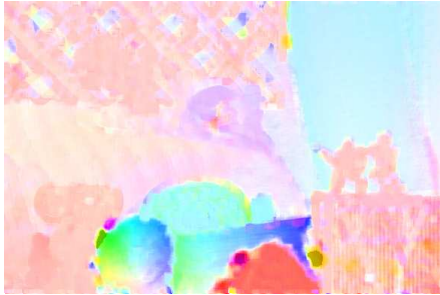
## Convergence : illustration

- FOLKI  
convergence comparée à IWS



encodage couleur

- Exemple Army, site d'éval. Middlebury/flow



FOLKI fenêtre  
porte largeur 9



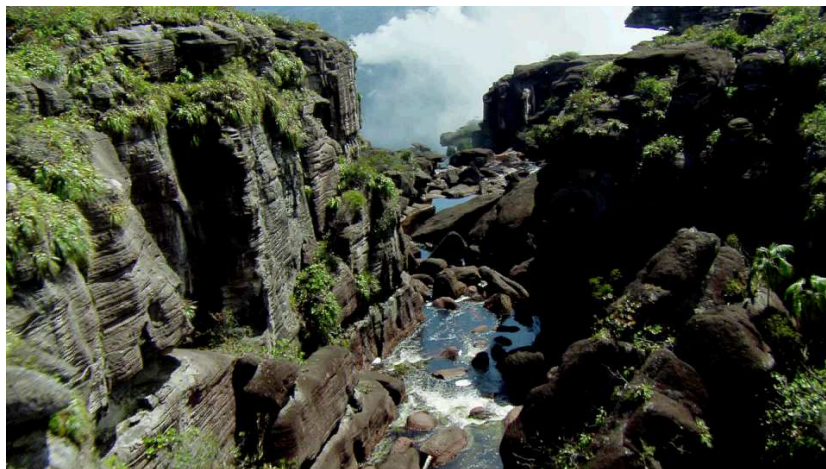
IWS fenêtre  
porte largeur 9

G. Le Besnerais *et al.* présentation GdR 8/1/9 – T15

ONERA  
THE FRENCH AEROSPACE LAB

## Exemples de résultats

- Données test : cf. site Middlebury
- « Yosemite 2008 » : plus grand, plus beau et séquence réelle!



G. Le Besnerais *et al.* présentation GdR 8/1/9 – T16

ONERA  
THE FRENCH AEROSPACE LAB

## Exemples de résultats



- Données test : cf. site Middlebury
- « Yosemite 2008 »

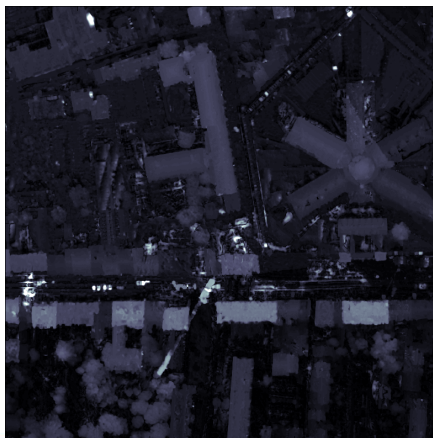


G. Le Besnerais *et al.* présentation GdR 8/1/9 – T17

ONERA  
THE FRENCH AEROSPACE LAB

## Exemples de résultats

- Corrélation en télédétection (image de simulation 2000x2000)
  - ! cadence vidéo sur GPU
  - Fenêtre gaussienne écart-type 4, pas de contrainte épipolaire

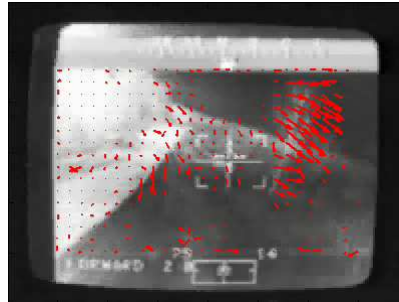


G. Le Besnerais *et al.* présentation GdR 8/1/9 – T18

ONERA  
THE FRENCH AEROSPACE LAB

## Exemples de résultats

- Séquences aériennes
  - Séquence IR, vol hélicoptère de nuit en rase-mottes
  - Applications
    - estimation de l'egomotion
    - aide au pilotage
    - évitement d'obstacle



G. Le Besnerais *et al.* présentation GdR 8/1/9 – T19

ONERA  
THE FRENCH AEROSPACE LAB

## FOLKI : aspects algorithmiques

- Forme convolutive obtenue

$$\sum_k \mathbf{w}(k)^t (F * \nabla_3 \tilde{I}_2^{(t)} \nabla_3^t \tilde{I}_2^{(t)})(k) \mathbf{w}(k) \quad \mathbf{w}(k) = \begin{bmatrix} \mathbf{u}(k) \\ 1 \end{bmatrix}$$

$$\nabla_3 \tilde{I}_2^{(t)}(\mathbf{m}) = \begin{bmatrix} \nabla \tilde{I}_2^{(t)}(\mathbf{m}) \\ \tilde{I}_2^{(t)}(\mathbf{m}) - I_1(\mathbf{m}) - \nabla \tilde{I}_2^{(t)}(\mathbf{m})^t \mathbf{u}_t(\mathbf{m}) \end{bmatrix}$$

Redonne forme classique pour itération initiale

$$\sum_k \delta \mathbf{w}(k)^t (F * \nabla_3 \tilde{I}_2 \nabla_3^t \tilde{I}_2)(k) \delta \mathbf{w}(k) \quad \delta \mathbf{w}(k) = \begin{bmatrix} \delta \mathbf{u}(k) \\ 1 \end{bmatrix}$$

$$\nabla_3 \tilde{I}_2(\mathbf{m}) = \begin{bmatrix} \nabla \tilde{I}_2(\mathbf{m}) \\ \tilde{I}_2(\mathbf{m}) - I_1(\mathbf{m}) \end{bmatrix}$$

G. Le Besnerais *et al.* présentation GdR 8/1/9 – T20

ONERA  
THE FRENCH AEROSPACE LAB

## FOLKI : aspects algorithmiques

- Forme convolutive obtenue

$$\sum_k \mathbf{w}(\mathbf{k})^t \left( F * \nabla_3 \tilde{I}_2^{(t)} \nabla_3^t \tilde{I}_2^{(t)} \right) (\mathbf{k}) \mathbf{w}(\mathbf{k}) \quad \mathbf{w}(\mathbf{k}) = \begin{bmatrix} \mathbf{u}(\mathbf{k}) \\ 1 \end{bmatrix}$$

$$\nabla_3 \tilde{I}_2^{(t)}(\mathbf{m}) = \begin{bmatrix} \nabla \tilde{I}_2^{(t)}(\mathbf{m}) \\ \tilde{I}_2^{(t)}(\mathbf{m}) - I_1(\mathbf{m}) - \nabla \tilde{I}_2^{(t)}(\mathbf{m})^t \mathbf{u}_t(\mathbf{m}) \end{bmatrix}$$

- En chaque pixel, un système local 2x2

$$\mathbf{H}(\mathbf{k}) \mathbf{u}(\mathbf{k}) = \mathbf{c}(\mathbf{k}) \quad \mathbf{H}(\mathbf{k}) = F * \left( \nabla \tilde{I}_2^{(t)} (\nabla \tilde{I}_2^{(t)})^t \right) (\mathbf{k})$$

$$\mathbf{c}(\mathbf{k}) = F * \left( \nabla \tilde{I}_2^{(t)} \right) \boldsymbol{\varepsilon}(\mathbf{k})$$

$$\boldsymbol{\varepsilon}(\mathbf{k}) = \tilde{I}_2^{(t)}(\mathbf{k}) - I_1(\mathbf{k}) - \nabla \tilde{I}_2^{(t)}(\mathbf{k})^t \mathbf{u}_t(\mathbf{k})$$

## FOLKI : aspects algorithmiques

- Pseudo-code d'une itération

- Utilisable dans tout schéma itératif

- ex : multi-résolution

- Version « inverse »

- matrices  $\mathbf{H}$  invariantes précalculées

- Code « facilement »

parallélisable et performances élevées

- Version GPU [A. Plyer, 2008] (bientôt) disponible en ligne
- démo

- compute interpolated gradient image  $\nabla \tilde{I}_0$
- compute error image  $\boldsymbol{\varepsilon} = T - \tilde{I}_0 + \nabla \tilde{I}_0 \odot \mathbf{u}_0$
- compute RHS images  $c_x = \nabla_x \tilde{I}_0 \otimes \boldsymbol{\varepsilon}$  and  $c_y = \nabla_y \tilde{I}_0 \otimes \boldsymbol{\varepsilon}$ .
- compute pseudo-Hessian matrix components  $H_{1,1}, H_{1,2}, H_{2,2}$  by pixelwise multiplication and separable convolution.
- solve  $KL$  local 2x2 systems by pixelwise ops:
  - compute  $D = H_{1,1} \otimes H_{2,2} - H_{1,2} \otimes H_{1,2}$
  - get the 2 updated flow update
 
$$\mathbf{u} = (H_{2,2} \otimes c_x - H_{1,2} \otimes c_y) \oslash D$$

$$\mathbf{v} = (H_{1,1} \otimes c_y - H_{1,2} \otimes c_x) \oslash D$$

## Perspectives

- Application à l'analyse de scène en vidéo aérienne
  - Orientations
    - intégration capteur-traitement
    - intégration dans boucle de commande (drones)
- Approche directe de la coopération locale-globale
  - CLG-FOLKI + multi-grille [R. Fezzani 08]
- Applications à la métrologie (projet ONERA MEMFIS)
  - Mesure de déformations
  - PIV (*imagerie de fluides par vélocimétrie de particules*)
    - Exemple : FOLKI-3C (3 composantes) [B. Jaubert, 2008]