



FP6-513663

FLUID
FLUID Image Description and analysis

Instrument: Strep

Priority: IST FET OPEN

Deliverable 4.1

Demonstrator on Layer Motion Estimation Methods

Patrick Heas, Etienne Memin, Nicolas Papadakis

Due date of deliverable: 01/09/07
Actual submission date: 13/09/07

Start date of project: 01/12/04

Duration: 3 years

Organisation name of lead contractor for this deliverable: INRIA

Revision [1]

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

This document aims at describing codes compiled under a Linux environment which are used for estimating layered atmospheric motion with the methods presented in [1][2]. The first part briefly describes the files (compiled codes and data examples) which are provided. The second part constitutes the reference manual which explains how the programs can be used.

1- List of files

```
<root>.  
|-- Code  
|   |-- CodeDenseMotion  
|   |   |-- 2DLayeredMotion  
|   |   |-- 3DLayeredMotion  
|   |   |-- param_Meteo_V2  
|   |   |-- inImage_3D.txt  
|   |   |-- inDensity.txt  
|   |-- CreateImages  
|   |   |-- makeImages.sh  
|   |   |-- RawIR2pgmImages  
|   |   |-- Ir2transImages  
|   |   |-- Pres2transImages  
|   |   |-- inputRawIR  
|   |   |-- inputFiles_IR  
|   |   |-- inputFiles_P  
|-- DataExamples  
|   |-- Classification  
|   |   |-- Natl_CLA_scene_analysis.43  
|   |   |-- Natl_CLA_scene_analysis.44  
|   |   |-- Natl_CLA_scene_analysis.45  
|   |   |-- raw2pgmClass.sh  
|   |-- CorrelationVectorField  
|   |   |-- Natl_cst_c10.8.43_44  
|   |   |-- Natl_cst_c10.8.44_45  
|   |   |-- PIV00_01  
|   |   |-- PIV01_02  
|   |-- IR  
|   |   |-- Natl_c10.8.43  
|   |   |-- Natl_c10.8.44  
|   |   |-- Natl_c10.8.45  
|   |-- Transmit  
|   |   |-- Natl_CLA_cloud_top_press.43  
|   |   |-- Natl_CLA_cloud_top_press.44  
|   |   |-- Natl_CLA_cloud_top_press.45
```

2- User Manual

In this section we describe the program developed to estimate dense 2D and 3D wind fields in a layered atmosphere.

2-1 Creation of layer-related pressure difference images

Estimators `2DLayeredMotion` and `3DLayeredMotion` (described in the following) use a set of pressure difference images related to a stacks of layers. The pressure difference images are computed from cloud classification maps and top of clouds pressure maps or, if the latter are unavailable, the pressure difference images may be derived from cloud classification maps and infra-red images.

An example of infrared and layer-related pressure difference images computation can easily be created by simply launching the script `makeImages.sh`.

For other examples, this script can be modified by fixing the options of the sub-routine `RawIR2pgmImages` for creating infrared images in *pgm* format and of the sub-routine `Pres2transImages` for creating layer-related pressure difference images in *pgm* format from top of clouds images and cloud classification maps. In the case top of cloud pressure images are not available, an alternative is to perform a coarse approximation of pressure difference images using the sub-routine `Ir2transImages` which is based on infrared images and classification maps.

The option list of each of the following sub-routines can be displayed when they are launched with the option `-h`

Layer pressure difference images from top of cloud pressure: `Pres2transImages`
Layer related pressure difference image files in 2 bytes *pgm* format can be derived from 2 bytes binary images of top of cloud pressure by the routine named `Pres2transImages` which takes as inputs a text file `inputFiles_P` composed on each line of a path related to an input binary image and to its corresponding binary cloud classification map (option `-path`), the image dimensions (options `-dy` and `-dx`), the number of images in the sequences (option `-n`), the number of layer which should be equal to 3 or less (options `-nbLayer`), the maximum and minimum class index defining each layer (options `-cMinL`, `-cMaxL`, `-cMinM`, `-cMaxM`, `-cMinH`, `-cMaxH`), and inputs for an optional cropping procedure (options `-ofx`, `-ofy` and `-dx`). For example, one can launch the routine using the following command :

```
./Pres2transImages -h -path inputFiles_P -dy 512 -dx 512 -n 3 -cMinL 101 -cMaxL 103 -cMinM 104 -cMaxM 106 -cMinH 107 -cMaxH 109 -ofy 40 -ofx 80 -Dx 1024
```

Infrared images: `RawIR2pgmImages`

The infrared image files in 2 bytes *pgm* format can be derived from MSG 2 bytes binary images by the routine named `RawIR2pgmImages` which takes as inputs a text file `inputRawIR` composed on each line of a path related to an input binary image (option `-path`), the image dimensions (options `-dy` and `-dx`), the number of images in the sequences (option `-n`) and inputs for an optional cropping procedure (options `-ofx`, `-ofy` and `-Dx`). For example, one can launch the routine using the following command:

```
./RawIR2pgmImages -h -path inputRawIR -dy 512 -dx 512 -n 3 -ofy 40 -  
ofx 80 -Dx 1024
```

Layer pressure difference images from infrared images: `Ir2transImages`

The infrared image files in 2 bytes *pgm* format can also be used with cloud classification maps in *pgm* format for the coarse approximation of layer-related pressure difference images. The routine used for computation is named `Ir2transImages`. It is based on the coarse retrieval of top of cloud pressure according to the procedure described in [3] with no cloud transparency corrections. The sub-routine takes as inputs a text file `inputFiles_IR` composed on each line of a path related to an input *pgm* image and to its corresponding *pgm* cloud classification map (option `-path`), the image dimensions (options `-dy` and `-dx`), the number of images in the sequences (option `-n`), the number of layer which should be equal to 3 or less (options `-nbLayer`), the maximum and minimum class index defining each layer (options `-cMinL`, `-cMaxL`, `-cMinM`, `-cMaxM`, `-cMinH`, `-cMaxH`), the black body calibration coefficient (options `-alpha`), the space count (options `-Ro`) and inputs for an optional cropping procedure (options `-ofx`, `-ofy`). Black body calibration coefficient and space count are values routinely provided by the EUMETSAT consortium. Binary Classification can be converted to *pgm* format launching (after modification for other examples) the script `raw2pgmClass.sh`. For example, one can launch the routine using the following commands :

```
cd ../../DataExamples/Classification  
raw2pgmClass.sh  
cd ../../Code/CreateImages/  
./ Ir2transImages -h -path inputFiles_IR -dy 512 -dx 512 -n 3 -cMinL  
101 -cMaxL 103 -cMinM 104 -cMaxM 106 -cMinH 107 -cMaxH 109 -ofy 40 -  
ofx 80-alpha 0.205034 -Ro -10.4568
```

2-2 Independent layered horizontal motion estimation

2-2-1 Description of the layered motion estimator

The proposed method solves the problem of estimating mesoscale dynamics of atmospheric layers from satellite top of cloud pressure image sequences. Due to the great deal of spatial and temporal distortions of cloud patterns and because of the sparse 3-dimensional nature of cloud observations, standard dense motion field estimation techniques used in computer vision are not well adapted to satellite images.

Relying on a physically sound vertical decomposition of the atmosphere into layers, we propose the dense motion estimator `2DLayeredMotion` dedicated to the extraction of multi-layer horizontal wind fields. This estimator is expressed as the minimization of a global function including a data term and a spatio-temporal smoothness term. A robust data term relying on the integrated continuity equation mass conservation model is proposed to fit sparse pressure difference observations related to each layer. Estimator `2DLayeredMotion` uses a div-curl regularizer en sus of a shallow water temporal regularizer enforcing in a first stage the time consistency of the flow vorticity and divergence according to simplified shallow waters equations. Moreover, the user can choose to incorporate correlation-based vector fields as external constraints in a variational scheme which yields in a first stage to the joint estimation of large scale motion based on correlation and optical flow. In a second stage both temporal and correlation-based constraints are removed and the estimation is refined at small scales based only on the ICE data model. The aim of this two-stage scheme is to propose an alternative approach overcoming multiresolution technique limitations and enforcing the time consistency of the estimates. For a detailed description please refer to [1].

2-2-2 Running the layered motion estimator

Image file (in `.pgm` format) paths and parameters needed for running `2DLayeredMotion` are grouped in a text file named `param_Meteo_V2` which has necessary to be on the same directory. It comprises on each of its line a character string describing the input. Default inputs are saved in `param_Meteo_V2` so that the program can be directly launched without fixing any options and using predefined images, classifications and correlation-based vector fields (provided by *LMD*). Note that the program must be run independently for each of the *K* layer.

The program inputs can be changed preserving the following file structure:

```
Number_of_images_in_the_sequence_(>=2): X
layer_t1_path: X
image_t2_path: X
layer_t2_path: X
image_t3_path: X
.
.
Regularisation_Coefficient: X
Coarsest_Resolution: X
Finest_Resolution: X
Coarsest_Grid: X
Finest_Grid: X
Robust_threshold_data_model: X
Regularization_coefficient_for_Div_and_Curl_smoothing: X
Constrain_velocity_estimates_with_correlation_yes(1)_no(0): X
Weighting_factor_for_correlation_constraints: X
Image_of_latitudes: X
Temporal_regularization_Coefficient: X
Standard_deviation_of_gaussian_extending_the_correlation_influence: X
Offset_in_x_for_reading_correlation_files: X
Offset_in_y_for_reading_correlation_files: X
image_t1_path: X
correlVectors_t1_t2: X
correlVectors_t2_t3: X
.
.
```

The correlation vector fields and the latitude image are text files which have the format produced by the *LMD software*. For a description of the file structure, please refer to [3].

The program produces horizontal motion files for each image couple in the binary format `'x.xaf'` and in the ascii format `'x.motion'` where each pixel of coordinate (i, j) is coded on a line with the following syntax: `"i j u v"`

2-2 Estimation of 3D winds of an atmosphere with interacting layers

2-2-1 Description of the interconnected layered 3D motion estimator

We propose now a motion estimator `3DLayeredMotion` solving for the first time the complex problem of 3D winds field recovery from satellite image sequences. In order to manage incomplete observations, physical knowledge on 3D mass exchanges between atmospheric layers have been introduced within an optical flow scheme.

The estimator `3DLayeredMotion` is based on a functional minimization. The data term relies on the 3D-ICE model which describes the dynamics of an interacting stack of atmospheric layers. The 3D-ICE model applies on a set of sparse pressure difference images related to the different atmospheric layers. To overcome the problem of sparse observations, a robust estimator is introduced in the data term. The data term is combined with a smoother that preserves the bi-dimensional divergent and vorticity structures of the three-dimensional flow and reinforces regions of homogeneous vertical winds. For a detailed description please refer to [2].

2-2-2 Running the interconnected layered motion 3D estimator

Default parameters which have been fixed for running `3DLayeredMotion` can all be interactively tuned using options in the command line. All options are visible by executing (in the directory which has necessary to be the one where is located the executable `3DLayeredMotion`) the command `-h`. Indeed executing the command :

```
./3DLayeredMotion -h
```

may be used to see all the current (here default) parameters :

```
3DLayeredMotion : 3D layered atmospheric motion estimation (Sep 13 2007, 14:07:12)

-inImage = ./inImage_3D.txt : text file with input image paths
-inDensity = ./inDensity.txt : text file with input average (in time and space) layer densities
-nbTimeSample = 3 : number of time samples in the image sequences
-nbLayers = 3 : number of layers
-crop = 0 : image cropping (yes=1/no=0)
-ofy = 0 : offset line number for cropping
-ofx = 0 : offset column number for cropping
-dy = 1024 : line number in cropped image
-dx = 1024 : column number in cropped image
-coarseResol = 0 : coarsest scale for multiresolution (no 2 stage scheme initialization)
-coarseGrid = 5 : coarsest grid of multigrid optimization
-alpha = 700 : global div-curl regularization coefficient
-lambda = 150 : regularization coefficient within the div-curl functional
-beta = 5000 : coefficient regularization vertical winds
-sigmaData = 9 : coefficient Phi-functions for robust data term
-SigmaReg = 1000 : coefficient Phi-functions for robust div-curl regularization term
-SigmaRegW = 1000 : coefficient Phi-functions for robust regularization vertical wind
-init = 0 : init. with (cropped) 2D motion estimates (name syntax
'inputImageFileNames.xaf') in 'xaf' format (yes=1/no=0)
```

Default inputs can be used and the program can be directly launched without fixing any options and using predefined images.

The inputs parameters can be changed using the following options :

`-inImage` may be used to give the path of the text file with the input image paths. The structure of this file should be similar to the example provided in `./inImage_3D.txt`. That is to say the structure alternating a masked image a time 't' and a non-masked image at time 't+1' :

```
layer_t1_path: X
image_t2_path: X
layer_t2_path: X
image_t3_path: X
```

`-inDensity` may be used to give the path of the text file with the input density values. The structure of this file should be similar to the example provided in `./inDensity.txt`. That is to say this file should have on K (default is 3) consecutive lines the values of the averaged in time and space densities characterizing the surface of the K different layers

`-nbTimeSample` may be used to fix the number of time samples in the image sequence

`-nbLayers` may be used to fix the number of layers

`-crop`, may be used to crop a region of interest in big images. Note that if this option is set to '1', one should fix parameters `-ofy`, `-ofx`, `-dy` and `-dx`.

`-coarseResol` may be used to fix the coarsest scale (exponent of 2) resolution procedure (maximum is 3). Note that if an initialization is provided, this option is automatically set to '0'

`-coarseGrid` may be used to fix the coarsest grid size (exponent of 2) used in the multigrid optimization procedure.

`-alpha` may be used to fix the global div-curl regularization coefficient.

`-lambda` may be used to fix the regularization coefficient within the div-curl functional

`-beta` may be used to fix the coefficient for regularization of vertical wind

`-sigmaData` may be used to fix the coefficient Phi-functions for robust data term

`-sigmaReg` may be used to fix the coefficient Phi-functions for robust div-curl regularization term

`-SigmaRegW` may be used to fix coefficient Phi-functions for robust regularization vertical wind

`-init` may be used to initialize with 2D motion estimates previously computed the 3D layered estimation procedure. The name syntax of the input motion files should be ' `x.xaf` '. Such files should be in 'xaf' format (produced by the previous 2D layered estimation method for the below 3D estimation) and such file must be located in the same directory as the input images ' `x.pgm` ' used for the 3D estimation. Using this option, one can take advantage of the 2-stage estimation scheme proposed in [1]. In a first stage, the estimation focuses in independent 2D motion estimation able to tackle large displacements of fine cloud structures and providing a physical sound time-consistency, while in a second stage incremental 2D motion fields together with vertical components are estimated.

The program produces horizontal motion files for each image couple in the ascii format `'x_3D.motion'` where each pixel of coordinate (i, j) is coded on a line with the following syntax: `"i j u v"`

The program produces vertical motion files for each image couple in the ascii format `'x_3D.motionUpMatlab'` which can be read with *Matlab* using simply the function `load('x_3D.motionUpMatlab')` and in the binary format `'x_3D.motionUp'` where vertical wind for each pixel location is coded in float.

References

- [1] P. Heas, E. Memin, N. Papadakis, A. Szantai, **Layered estimation of atmospheric mesoscale dynamics from satellite imagery**, IEEE Transactions on Geosciences and Remote Sensing (in press.).
- [2] P. Heas, E. Memin, **3D motion estimation of atmospheric layers from image sequences**, IEEE Transactions on Geosciences and Remote Sensing (in revision).
- [3] A. Szantai and F. Desalmand. **Basic information on msg images**. Fluid project deliverable 1.1, 2005.