Motion2D User Manual

Release 1.3.11

Fabien Spindler

Irisa/Inria, Campus universiatire de Beaulieu
35042 Rennes Cedex, France
e-mail: Fabien.spindler@irisa.fr

January 2005

Contents

Chapter 1

About Motion2D

1.1 Introduction

The Motion2D software estimates 2D parametric motion models between two successive images. It can handle several types of motion models, respectively, constant model (translation), affine, and quadratic models. Moreover, it integrates the possibility of taking into account the global variation of illumination. Motivations for the use of such models are, on one hand, their efficiency, which has been demonstrated in numerous contexts such as estimation, segmentation, tracking, and interpretation of motion, and on the other hand, their low computational cost compared to optical flow estimation. Moreover to have the best accuracy for the estimated parameters, and to take into account the problem of multiple motion, Motion2D exploit a robust, multiresolution and incremental estimation method exploiting only the spatio-temporal derivatives of the intensity function.

Motion2D was developped at `Irisa/INRIA Rennes` (`http://www.irisa.fr`) by the `Vista team` (`http://www.irisa.fr/vista`).

1.2 Description

The CMotion2D software implements a robust multiresolution estimation of parametric motion models.

We consider the displaced frame difference corresponding to the motion model:

$$DFD_{\Theta}(p_i) = I_{t+1}(p_i + \vec{w}_A(p_i)) - I_t(p_i) + \zeta$$

where I_t and I_{t+1} are two successive images, $\Theta^t = (A^t, \zeta)$ the motion model to estimate (including a global intensity shift ζ to account for global illumination change) and

$$\vec{w}_A(p_i) = \left[\begin{array}{c} u(p_i) \\ v(p_i) \end{array} \right] = B(p_i)A$$

denotes the flow vector modeled at the image point p_i.

To ensure the goal of robustness, we introduce a M-estimation criterion with a hard-redescending robust estimator. Thus, the motion model is given by:

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmin}} \sum_{p_i \in R_t} \rho\left(\mathrm{DFD}_{\Theta}(p_i), C\right) \qquad (1)$$

where $\rho(x)$ is the Tukey, Talwar, Cauchy or Welsh biweight function which is bounded for high values of x and C a scale parameter to be set.

The estimation support R_t usually consists of the whole image. If required, it can also be restricted to a specific area of the image. The minimization is embedded in a multiresolution and incremental scheme based on the Gauss-Newton method. At each incremental step k (at a given resolution level, or from a resolution level to a finer one), we have:

$$\Theta = \widehat{\Theta}_k + \Delta\Theta_k \quad \text{with} \quad \left\{ \begin{array}{l} A = \widehat{A_k} + \Delta A_k \\ \zeta = \widehat{\zeta}_k + \Delta\zeta \end{array} \right. \qquad (2)$$

where $\widehat{\Theta}_k$ is the current estimate of the parameter vector Θ. A linearization of $\mathrm{DFD}_{\Theta}(p_i)$ around $\widehat{\Theta}_k$ is performed, leading to the expression $r_{\Delta\Theta_k}(p_i)$ linear with respect to $\Delta\Theta_k$:

$$r_{\Delta\Theta_k}(p_i) = \vec{\nabla} I_{t+1}(p_i + \vec{w}_{\widehat{A_k}}(p_i)).\vec{w}_{\Delta A_k}(p_i) + \Delta\zeta_k + I_{t+1}(p_i + \vec{w}_{\widehat{A_k}}(p_i)) - I_t(p_i) + \widehat{\zeta_k}$$

where $\vec{\nabla} I_{t+1}(p_i)$ denotes the spatial gradient of the intensity function at location p_i and at time $t+1$. Then, we substitute for the minimization of $E(\Theta)$ the minimization of the expression given by:

$$E(\Delta\Theta_k) = \sum_{p_i \in R_t} \rho(r_{\Delta\Theta_k}(p_i), C)$$

which is equivalent to minimizing:

$$\sum_{p_i \in R_t} w_i . r^2_{\Delta\Theta_k}(p_i) \qquad (3)$$

with

$$w_i = \frac{\psi(r_{\Delta\Theta_k}(p_i))}{r_{\Delta\Theta_k}(p_i))} \qquad (4)$$

where ψ is the derivative of the ρ function. The error function $E(\Delta\Theta_k)$ is minimized using an Iterative-Reweighted-Least-Squares procedure, with 0 as an initial value for $\Delta\Theta_k$. For more details about the method and its performances, see the Motion-2D related references.

1.2.1 Motion2D supported platforms

The Motion2D software runs on Unix (Linux, SunOS, Solaris and Irix) and Windows 95/98/ME, Windows NT 4, Windows 2000, Windows XP operating systems.

Operating system	**Compiler**
Linux	g++
SunOS	g++
Solaris	g++, CC
Irix64	g++, CC
Windows	Visual C++, Cygwin g++

Motion2D does not require any special libraries.

1.3 Supported 2D polynomial motion models

Using matrix notation, supported 2D polynomial motion models can always be stated in the general way by:

$$\vec{w}_A(p_i) = \begin{bmatrix} u(p_i) \\ v(p_i) \end{bmatrix} = B(p_i)A$$

which is linear with respect to the n motion parameters $A^t = (c_1, c_2, a_1, ..., a_4, q_1, ..., q_6)$, and where $p_i = (x_i, y_i)$ denotes the spatial position of a point in $(C, \vec{x}, \vec{y})$, $\vec{w}_A(p_i)$ the flow vector modeled at point p_i; B is a matrix, the form of wich depends on the chosen model, but the coefficients of which depend only on the point coordinates.

1.3.1 Constant motion models

MDL_TX 2D translation motion model along the x axis with 1 parameter c_1:

$$\vec{w}_A(p_i) = \begin{bmatrix} u(p_i) \end{bmatrix} = \begin{pmatrix} c_1 \end{pmatrix}$$

MDL_TY 2D translation motion model along the y axis with 1 parameter c_2:

$$\vec{w}_A(p_i) = \begin{bmatrix} u(p_i) \end{bmatrix} = \begin{pmatrix} c_2 \end{pmatrix}$$

MDL_TR 2D translation motion model with 2 parameters (c_1, c_2) :

$$\vec{w}_A(p_i) = \begin{bmatrix} u(p_i) \\ v(p_i) \end{bmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$$

1.3.2 Affine motion models

MDL_AFF_TX_DIV 2D affine motion model with 2 parameters (c_1, a_1) taking into account the translation along the x axis and the 2D divergence:

$$\vec{w}_A(p_i) = \begin{bmatrix} u(p_i) \\ v(p_i) \end{bmatrix} = \begin{pmatrix} c_1 \\ \end{pmatrix} + \begin{pmatrix} a_1 & 0 \\ 0 & a_1 \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

MDL_AFF_TR_DIV 2D affine motion model with 3 parameters (c_1, c_2, a_1) taking into account the 2D translation and the 2D divergence:

$$\vec{w}_A(p_i) = \begin{bmatrix} u(p_i) \\ v(p_i) \end{bmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} a_1 & 0 \\ 0 & a_1 \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

MDL_AFF_TR_ROT 2D affine motion model with 3 parameters (c_1, c_2, a_1) taking into account the 2D translation and the 2D rotation:

$$\vec{w}_A(p_i) = \begin{bmatrix} u(p_i) \\ v(p_i) \end{bmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} 0 & a_1 \\ -a_1 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

MDL_AFF_TX_NULL 2D affine motion model with 5 parameters $(c_2, a_1, a_2, a_3, a_4)$ with:

$$\vec{w}_A(p_i) = \begin{bmatrix} u(p_i) \\ v(p_i) \end{bmatrix} = \begin{pmatrix} \\ c_2 \end{pmatrix} + \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

If we express the parameter vector A^t in another basis of elementary motion sub-fields $\phi^t = (c_1, c_2, div, rot, hyp_1, hyp_2)$ with: $div = \frac{1}{2}(a_1 + a_4)$, $rot = \frac{1}{2}(a_3 - a_2)$, $hyp1 = \frac{1}{2}(a_1 - a_4)$ and $hyp2 = \frac{1}{2}(a_2 + a_3)$, this motion model has the particularity to consider the translation along the x axis as null (ie. $c_1 = 0$)

MDL_AFF_TY_NULL 2D affine motion model with 5 parameters $(c_1, a_1, a_2, a_3, a_4)$ with:

$$\vec{w}_A(p_i) = \begin{bmatrix} u(p_i) \\ v(p_i) \end{bmatrix} = \begin{pmatrix} c_1 \\ \end{pmatrix} + \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

If we express the parameter vector A^t in another basis of elementary motion sub-fields $\phi^t = (c_1, c_2, div, rot, hyp_1, hyp_2)$ with: $div = \frac{1}{2}(a_1 + a_4)$, $rot = \frac{1}{2}(a_3 - a_2)$, $hyp1 = \frac{1}{2}(a_1 - a_4)$ and $hyp2 = \frac{1}{2}(a_2 + a_3)$, this motion model has the particularity to consider the translation along the y axis as null (ie. $c_2 = 0$)

MDL_AFF_DIV_NULL 2D affine motion model with 5 parameters $(c_1, c_2, a_1, a_2, a_3)$ with:

$$\vec{w}_A(p_i) = \begin{bmatrix} u(p_i) \\ v(p_i) \end{bmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} a_1 & a_2 \\ a_3 & -a_1 \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

If we express the parameter vector A^t in another basis of elementary motion sub-fields $\phi^t = (c_1, c_2, div, rot, hyp_1, hyp_2)$ with: $div = \frac{1}{2}(a_1 + a_4)$, $rot = \frac{1}{2}(a_3 - a_2)$, $hyp1 = \frac{1}{2}(a_1 - a_4)$ and $hyp2 = \frac{1}{2}(a_2 + a_3)$, this motion model has the particularity to consider the divergence term at zero (ie. $div = 0$)

MDL_AFF_ROT_NULL 2D affine motion model with 5 parameters $(c_1, c_2, a_1, a_2, a_4)$ with:

$$\vec{w}_A(p_i) = \left[\begin{array}{c} u(p_i) \\ v(p_i) \end{array} \right] = \left(\begin{array}{c} c_1 \\ c_2 \end{array} \right) + \left(\begin{array}{cc} a_1 & a_2 \\ a_2 & a_4 \end{array} \right) \cdot \left(\begin{array}{c} x_i \\ y_i \end{array} \right)$$

If we express the parameter vector A^t in another basis of elementary motion sub-fields $\phi^t = (c_1, c_2, div, rot, hyp_1, hyp_2)$ with: $div = \frac{1}{2}(a_1 + a_4)$, $rot = \frac{1}{2}(a_3 - a_2)$, $hyp1 = \frac{1}{2}(a_1 - a_4)$ and $hyp2 = \frac{1}{2}(a_2 + a_3)$, this motion model has the particularity to consider the curl term at zero (ie. $rot = 0$)

MDL_AFF_HYP1_NULL 2D affine motion model with 5 parameters $(c_1, c_2, a_1, a_2, a_3)$ with:

$$\vec{w}_A(p_i) = \left[\begin{array}{c} u(p_i) \\ v(p_i) \end{array} \right] = \left(\begin{array}{c} c_1 \\ c_2 \end{array} \right) + \left(\begin{array}{cc} a_1 & a_2 \\ a_3 & a_1 \end{array} \right) \cdot \left(\begin{array}{c} x_i \\ y_i \end{array} \right)$$

If we express the parameter vector A^t in another basis of elementary motion sub-fields $\phi^t = (c_1, c_2, div, rot, hyp_1, hyp_2)$ with: $div = \frac{1}{2}(a_1 + a_4)$, $rot = \frac{1}{2}(a_3 - a_2)$, $hyp1 = \frac{1}{2}(a_1 - a_4)$ and $hyp2 = \frac{1}{2}(a_2 + a_3)$, this motion model has the particularity to consider the first hyperbolic term at zero (ie. $hyp_1 = 0$).

MDL_AFF_HYP2_NULL 2D affine motion model with 5 parameters $(c_1, c_2, a_1, a_2, a_4)$ with:

$$\vec{w}_A(p_i) = \left[\begin{array}{c} u(p_i) \\ v(p_i) \end{array} \right] = \left(\begin{array}{c} c_1 \\ c_2 \end{array} \right) + \left(\begin{array}{cc} a_1 & a_2 \\ -a_2 & a_4 \end{array} \right) \cdot \left(\begin{array}{c} x_i \\ y_i \end{array} \right)$$

If we express the parameter vector A^t in another basis of elementary motion sub-fields $\phi^t = (c_1, c_2, div, rot, hyp_1, hyp_2)$ with: $div = \frac{1}{2}(a_1 + a_4)$, $rot = \frac{1}{2}(a_3 - a_2)$, $hyp1 = \frac{1}{2}(a_1 - a_4)$ and $hyp2 = \frac{1}{2}(a_2 + a_3)$, this motion model has the particularity to consider the second hyperbolic term at zero (ie. $hyp_2 = 0$).

MDL_AFF_TR_ROT_DIV 2D affine motion model with 4 parameters (c_1, c_2, a_1, a_2) able to consider motion like 2D translation with 2D rotation and divergence:

$$\vec{w}_A(p_i) = \left[\begin{array}{c} u(p_i) \\ v(p_i) \end{array} \right] = \left(\begin{array}{c} c_1 \\ c_2 \end{array} \right) + \left(\begin{array}{cc} a_1 & a_2 \\ -a_2 & a_1 \end{array} \right) \cdot \left(\begin{array}{c} x_i \\ y_i \end{array} \right)$$

If we express the parameter vector A^t in another basis of elementary motion sub-fields $\phi^t = (c_1, c_2, div, rot, hyp_1, hyp_2)$ with: $div = \frac{1}{2}(a_1 + a_4)$, $rot = \frac{1}{2}(a_3 - a_2)$, $hyp1 = \frac{1}{2}(a_1 - a_4)$ and $hyp2 = \frac{1}{2}(a_2 + a_3)$, this motion model has the particularity to consider the hyperbolic terms at zero (ie. $hyp_1 = hyp_2 = 0$).

MDL_AFF_COMPLET 2D affine motion model with 6 parameters $(c_1, c_2, a_1, ..., a_2)$ taking into account constant and affine parameters:

$$\vec{w}_A(p_i) = \left[\begin{array}{c} u(p_i) \\ v(p_i) \end{array} \right] = \left(\begin{array}{c} c_1 \\ c_2 \end{array} \right) + \left(\begin{array}{cc} a_1 & a_2 \\ a_3 & a_4 \end{array} \right) \cdot \left(\begin{array}{c} x_i \\ y_i \end{array} \right)$$

This model is a good tradeoff between complexity and representativeness. It can take into account many kind of motion (translation, rotation, scaling, deformation), and even if a rigid 3D motion gives rise to a quadratic model in the image plane, the affine flow recovers the essential part.

1.3.3 Quadratic motion models

MDL_QUA_PAN_DIV 2D quadratic motion model with 4 parameters (c_1, a_1, q_1, q_2) taking into account the translation along the x axis, some affine and quadratic parameters:

$$\vec{w}_A(p_i) = \begin{bmatrix} u(p_i) \\ v(p_i) \end{bmatrix} = \begin{pmatrix} c_1 \\ \end{pmatrix} + \begin{pmatrix} a_1 & 0 \\ 0 & a_1 \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} q_1 & q_2 & 0 \\ 0 & q_1 & q_2 \end{pmatrix} \cdot \begin{pmatrix} {x_i}^2 \\ x_i y_i \\ {y_i}^2 \end{pmatrix}$$

MDL_QUA_PAN_TILT 2D quadratic motion model with 4 parameters (c_1, c_2, q_1, q_2) taking into account the 2D translation and some quadratic parameters:

$$\vec{w}_A(p_i) = \begin{bmatrix} u(p_i) \\ v(p_i) \end{bmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} q_1 & q_2 & 0 \\ 0 & q_1 & q_2 \end{pmatrix} \cdot \begin{pmatrix} {x_i}^2 \\ x_i y_i \\ {y_i}^2 \end{pmatrix}$$

MDL_QUA_PAN_TILT_DIV Quadratic 2D motion model with 5 parameters $(c_1, c_2, a_1, q_1, q_2)$ taking into account the 2D translation, some affine and quadratic parameters:

$$\vec{w}_A(p_i) = \begin{bmatrix} u(p_i) \\ v(p_i) \end{bmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} a_1 & 0 \\ 0 & a_1 \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} q_1 & q_2 & 0 \\ 0 & q_1 & q_2 \end{pmatrix} \cdot \begin{pmatrix} {x_i}^2 \\ x_i y_i \\ {y_i}^2 \end{pmatrix}$$

MDL_QUA_2D 2D quadratic motion model with 8 parameters $(c_1, c_2, a_1, ..., a_4, q_1, q_2)$ taking into account the constants, the linear and some quadratic parameters:

$$\vec{w}_A(p_i) = \begin{bmatrix} u(p_i) \\ v(p_i) \end{bmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} q_1 & q_2 & 0 \\ 0 & q_1 & q_2 \end{pmatrix} \cdot \begin{pmatrix} {x_i}^2 \\ x_i y_i \\ {y_i}^2 \end{pmatrix}$$

MDL_QUA_COMPLET 2D quadratic motion model with 12 parameters $(c_1, c_2, a_1, ..., a_4, q_1, ..., q_6)$ taking into account the constants, the linear and all the quadratic parameters:

$$\vec{w}_A(p_i) = \begin{bmatrix} u(p_i) \\ v(p_i) \end{bmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} q_1 & q_2 & q_3 \\ q_4 & q_5 & q_6 \end{pmatrix} \cdot \begin{pmatrix} {x_i}^2 \\ x_i y_i \\ {y_i}^2 \end{pmatrix}$$

1.4 Contact information

For general information on the Vista team, please visit the Vista website: `http://www.irisa.fr/vista`.

For technical information on Motion2D, contact `Motion2D@irisa.fr.`

Chapter 2

Usage

2.1 Motion2D options

2.1.1 Synopsis

```
./Motion2D [-p image or video path] [-R rows number for RAW images]
[-C columns number for RAW images] [-f first frame]
[-s step] [-i iterations] [-m model id] [-g] [-x model col orig]
[-y model row orig] [-a support image or video path] [-z]
[-c support label] [-q nbsubsample] [-n number of pyramid levels]
[-l pyramid stop level] [-r dominant motion filename]
[-r multiplier factor] [-b back warped image path]
[-j back warped image nrows] [-k back warped image ncols]
[-t back warped image row origin] [-u back warped image col origin]
[-w weights image path] [-F optic flow field] [-I] [-v] [-h] [-?]
```

2.1.2 Description

The Motion2D software provides a method to estimate 2D parametric motion models between two successive images. It can handle several types of motion models, respectively, constant model (translation), affine, and quadratic models. Moreover, it integrates the possibility of taking into account the global variation of illumination. Motivations for the use of such models are, on one hand, their efficiency, which has been demonstrated in numerous contexts such as estimation, segmentation, tracking, and interpretation of motion, and on the other hand, their low conputational cost compared to optical flow estimation. Moreover to have the best accuracy for the estimated parameters, and to take into account the problem of multiple motion, Motion2D exploit a robust, multiresolution and incremental estimation method exploiting only the spatio-temporal derivatives of the intensity function.

2.1.3 Input video sequence options

```
-p [%s]: image or video path
   . Dealing with an image sequence:
       By image sequence, we mean one file per image.
       Specify the path and the generic name of the files
       containing the images to process. The following image
       file formats PNG, PNM (PGM P5, PPM P6) and RAW
       (RAW8, RAW16) are supported. The format is selected by
       analysing the filename extension.
       Example: -p rond-point%04d.png
   . Dealing with a video:
       By video, we mean one file for all images of the video.
       Specify the video file. Only Mpeg2 file are supported.
       Warning: in a Mpeg2 video stream the first frame
       has the number zero.
       Example: -p video.mpg -f 0

-R [%d]: Number of rows in RAW image.
   Specify the number of rows for RAW sequences.

-C [%d]: Number of columns in RAW image.
   Specify the number of columns for RAW sequences.

-f [%s]: first frame                                              1
   Specify the number of the first frame in the video
   sequence. If the image sequence numbering uses a fixed
   number of digits, complete whith 0 before the image
   number.
   Warning: the first frame of a Mpeg2 video stream has the
   number zero.

-s [%d]: step                                                     1
   Specify the step between two frames in the video sequence.
   If step > 0 images are processed forward. If step < 0
   images are processed backward. This parameter allow the
   video temporal subsampling.

-i [%lu]: iterations                                            33
   Specify the number of motion estimation iterations to
   process. The number of the last image computed is given by:
   first_frame + iterations * step.
```

2.1.4 Motion model options (input)

```
-m model_id [%s]                                                AC
   Specify the parametric motion model id to estimate. The
   table below gives the list of possible model_id strings:
   |-----------------------------------------------------|
   | model_id | type        | number of parameters       |
   |-----------------------------------------------------|
| TX       | constant    |  1 (c1)                    |
| TY       | constant    |  1 (c2)                    |
| TR       | constant    |  2 (c1,c2)                 |
| ----------------------------------------------------|
| AXD      | affine      |  2 (c1,a1)                 |
| ARD      | affine      |  3 (c1,c2,a1)              |
| ARR      | affine      |  3 (c1,c2,a1)              |
| AXN      | affine      |  5 (c2,a1...a4)            |
| AYN      | affine      |  5 (c1,a1...a4)            |
| ADN      | affine      |  5 (c1,c2,a1,a2,a3)        |
| ARN      | affine      |  5 (c1,c2,a1,a2,a4)        |
| AH1N     | affine      |  5 (c1,c2,a1,a2,a3)        |
| AH2N     | affine      |  5 (c1,c2,a1,a2,a4)        |
| ARRD     | affine      |  4 (c1,c2,a1,a2)           |
| AC       | affine      |  6 (c1,c2,a1...a4)         |
| ----------------------------------------------------|
| QPD      | quadratic   |  4 (c1,a1,q1,q2)           |
| QPT      | quadratic   |  4 (c1,c2,q1,q2)           |
| QPTD     | quadratic   |  5 (c1,c2,a1,q1,q2)        |
| Q2D      | quadratic   |  8 (c1,c2,a1...a4,q1,q2)   |
| QC       | quadratic   | 12 (c1,c2,a1...a4,q1...q6) |
------------------------------------------------------|
| MDL_TX               | same as TX                   |
| MDL_TY               | same as TX                   |
| MDL_TR               | same as TR                   |
| MDL_AFF_TX_DIV       | same as AXD                  |
| MDL_AFF_TR_DIV       | same as ARD                  |
| MDL_AFF_TR_ROT       | same as ARR                  |
| MDL_AFF_TX_NULL      | same as AXN                  |
| MDL_AFF_TY_NULL      | same as AYN                  |
| MDL_AFF_DIV_NULL     | same as ADN                  |
| MDL_AFF_ROT_NULL     | same as ARN                  |
| MDL_AFF_HYP1_NULL    | same as AH1N                 |
| MDL_AFF_HYP2_NULL    | same as AH2N                 |
| MDL_AFF_TR_ROT_DIV   | same as ARRD                 |
| MDL_AFF_COMPLET      | same as AC                   |
| MDL_QUA_PAN_DIV      | same as QPD                  |
| MDL_QUA_PAN_TILT     | same as QPT                  |
| MDL_QUA_PAN_TILT_DIV | same as QPTD                 |
| MDL_QUA_2D           | same as Q2D                  |
| MDL_QUA_COMPLET      | same as QC                   |
------------------------------------------------------|
```

-g

Specify that the global illumination parameter will be estimated.

-I

```
   Uses the previous estimation to initialize the motion model
   to estimate.

-x model_col_orig [%f]
   Sets the origin (column coordinate) of the motion model.
   By default, this parameter is initialized to the mid image
   column number.

-y model_row_orig [%f]
   Sets the origin (row coordinate) of the motion model.
   By default, this parameter is initialized to the mid image
   line number.

-z
   Computes the covariance matrix of the motion model parameters.
```

2.1.5 Estimation support options (input)

```
-a [%s]: support image or video path
   Motion2D makes it possible to estimate motion either on
   all the image (by defect), or on a part of the image. In
   this case, it is necessary to indicate to the software the
   position of the area on which the motion estimation must be
   done. This is carried out while associating each image of
   the video sequence a file called estimation support, corres-
   ponding to an image with same size, numbering and extension
   as the images to be treated. Thus, this option specify the
   path and the generic name of the files containing the images
   corresponding to the estimation support. The images format
   depends on the filename extension. We support only Mpeg2, PNG,
   PNM (PGM P5, PPM P6) and RAW (RAW8, RAW16) image file formats.
   Be aware, images to proceed (those specified by option -p
   and option -a) must have the same size.
   For example, to estimate the motion between images
   "rond-point0001.png" and "rond-point0002.png", the
   support file name must be "support0001.png".

-c [%d]: support label
   This option is associated to the previous one. It fixes the
   value of the estimation support label where the estimation
   will be achieved.
```

2.1.6 Multi-resolution framework options (input)

```
-q [%d]: nbsubsample                                             0
   Number of subsampling to apply to the images before
```

starting the motion estimation process. If this parameter is equal to zero, images are not subsampled. If this parameter is greater than zero, images are resized at a lower resolution where the number of lines and columns is divided by 2^nbsubsample.
Be aware, the estimated parametric motion model is returned at the highest resolution.

-n [%u]: number of pyramid levels 4

Specify the number of levels in the Gaussian image pyramids and image gradients pyramids used in the multi-resolution framework. In general, when the size of the images to be treated is close to CIF format (352 X 288), it is advised to fix this parameter at 4. When the images are with QCIF format (176 X 144), this parameter can be fixed at 3.
This parameter implicitly fix the index of the initial level in the pyramids where the robust estimate begins. This initial level is than equal to the number of levels in the pyramids minus one. The selected level is then that of lower resolution.

-l [%u]: pyramid stop level 0

Specify the level of the image pyramids where the estimation process will be stopped. This parameter in general fixed at zero (corresponds then to the level of higher resolution) can vary in the range [0, number_of_pyramid_levels - 1].
The coarsest level is given by number_of_pyramid_levels - 1.
The finest level is 0.

2.1.7 Result options (output)

-r [%s]: dominant motion filename

Specify the name of the file which will contain the estimated parametric dominant motion model results.
This result file contains the values of the estimated 2D parametric motion model given at the highest image resolution. The parameters of the estimated motion model are very small, especially affine and quadratic terms. To increase these parameters resolution, a multiplier factor can be applied to all the parameters (see option -e).

-e [%f]: multiplier factor for the motion parameter 1.0

This option is associated with the -r option.
It defines a multiplier factor applied to all the motion model parameter (c1,c2,a1,...a4,q1...q6) when those are saved in a result filename specified with option -r.

-b [%s]: back warped image path

Specify the path and the generic name of the files containing the back-warped images built using the estimated motion model. We support only PNG, PNM (PGM P5, PPM P6) and RAW (RAW8, RAW16) image file formats. The format is selected by analysing the filename extension.

Example: -b /tmp/B%04.png

-j [%d]: back warped image nrows

Set the number of rows of the back-warped image. By default, the back-warped image has the same rows number than the images to proceed. This option is taken into account only when option -b is used.

-k [%d]: back warped image ncols

Set the number of columns of the back-warped image. By default, the back-warped image has the same rows number than the images to proceed. This option is taken into account only when option -b is used.

-t [%d]: back warped image row origin 0

This option makes it possible to fix the co-ordinates of the row origin of the back-warped image compared to the origin of the images to treat. This option is taken into account only when option -b is used.

-u [%d]: back warped image col origin 0

This option makes it possible to fix the co-ordinates of the column origin of the back-warped image compared to the origin of the images to treat. This option is taken into account only when option -b is used.

-w [%s]: weights image path

Specify the path and the generic name of the files containing the M-estimator weights. These weights in [0,1] are rescaled in [0,255]. This map can be used to see if a pixel participate to the robust motion estimation (pixel in white) or is more considered as an outlier (pixel in black). We support only PNG, PNM (PGM P5, PPM P6) and RAW (RAW8, RAW16) image file formats. The format is selected by analysing the filename extension.

Example: -w /tmp/W%04.png

-F [%s]: field vector image path

Specify the path and the generic name of the files containing the displacement vectors in fieldshow format.

Example: -w /tmp/F%04.field

2.1.8 Other options

```
-v
   Activate the verbose mode.

-h
   Print the help.

-?
   Print the help.
```

2.2 Example of 2D parametric motion models estimation.

Here are given some examples of Motion2D usage to estimate 2D parametric motion models between two successive images of a video sequence. It can handle several types of motion models, respectively, constant model (translation), affine, and quadratic models. Moreover, it integrates the possibility of taking into account the global variation of illumination.

To get help or to see the available options try:

```
./Motion2D -?
```

To estimate the dominant 2D affine motion model between each couple of images of the round-about sequence in Mpeg2 format and put the results in the /tmp/result.txt file try:

```
./Motion2D -m AC -p ../../test/sequence/mpeg2/rond-point.mpg -f 0 -i 31
-r /tmp/result.txt -v
```

To estimate the dominant 2D affine motion model between each couple of images of the round-about sequence in PNG format and put the results in the /tmp/result.txt file try:

```
./Motion2D -m AC -p ../../test/sequence/png/rond-point%04d.png -f 1 -i 33
-r /tmp/result.txt -v
```

To estimate the dominant motion using a quadratic motion model between each couple of images of the round-about sequence in PPM P6 format and generate a back-warped sequence in PNG format try:

```
./Motion2D -m QPTD -p ../../test/sequence/ppm/rond-point%04d.ppm -f 1 -i 33
-b /tmp/bwarp%04d.png -j 264 -k 450 -t -30 -u -20 -v
```

To estimate the motion of the mobile vehicle in the round-about sequence in PNG format using a quadratic motion model and an input estimation support sequence in PNG format too, try:

```
./Motion2D -m QC -p ../../test/sequence/png/rond-point%04d.png -f 1 -i 33
-a ../../test/sequence/png/support%04d.png -c 255 -v
```