INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

CamCal ***v1.0 Manual*** ***A complete software Solution for Camera Calibration***

Jean-Philippe Tarel, Jean-Marc Vezien

N˚ 0196

September 1996

THÈME 3

Rapport technique

ISSN 0249-0803

CamCal v1.0 Manual
A complete software Solution for Camera Calibration

Jean-Philippe Tarel[*], Jean-Marc Vezien

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet Syntim

Rapport technique n ° 0196 — September 1996 — 19 pages

Abstract: This technical report is the user manual of *CamCal* v1.0. *CamCal* is a software for geometric camera calibration from a 3D calibration set-up image.

Key-words: Geometric camera calibration, *CamCal* manual, 3D calibration set-up.

(Résumé : tsvp)

CamCal software is distributed by INRIA. For distribution, contact Jean-Philippe Tarel and/or Jean-Louis Bouchenez (E-mail : Jean-Louis.Bouchenez@inria.fr).

Le programme *CamCal* est distribué par l'INRIA. Pour obtenir le logiciel contacter Jean-Philippe Tarel et/ou Jean-Louis Bouchenez (E-mail : Jean-Louis.Bouchenez@inria.fr).

[*] E-mail : Jean-Philippe.Tarel@inria.fr

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : (33 1) 39 63 55 11 – Télécopie : (33 1) 39 63 53 30

Manuel du programme *Camcal* v1.0
Une solution logicielle complète pour la calibration de caméra

Résumé : Ce rapport technique est le manuel d'utilisation de *CamCal* v1.0. *CamCal* est un logiciel qui permet la calibration géométrique d'une caméra à partir de l'image d'une mire 3D.

Mots-clé : Calibration géométrique de caméra, Manuel de *CamCal*, Mire 3D.

Contents

1 Introduction

It is known for a long time that depth perception can be attained by merging the information captured from multiple images taken at different viewpoints, a process known as stereovision. For this, it is important to know, among other things, the relative positions of the sensors. Unfortunately, the precise camera locations are usually unknown at the time of acquisition. To solve this problem a range of methods exist, called CAMERA CALIBRATION. This manual briefly presents a specific implementation of such methods, the *CamCal* software. The resulting calibration can be applied to stereovision but also to a wide range of other machine vision algorithms.

Briefly put, our algorithm estimates, for a given set-up, the position of the camera (extrinsic parameters) and its internal viewing characteristics (intrinsic parameters), by using both 2D points and 3D ellipses drawn on a calibration device whose geometry is known with great accuracy. Let us now first briefly explain whose parameters we plan on actually recovering, and how the calibration procedure computes them. Then we will see how the program, named *CamCal*, actually works.

2 Camera model

2.1 extrinsic parameters: camera position

Although self-calibration methods have recently arisen, it is still customary to use the image of a special calibration pattern to recover the imaging process parameters. The first and most natural information one needs is the camera location with respect to a fixed coordinate frame. To describe the position of the camera we will use the translation T and the rotation R of an absolute coordinate system fixed on the calibration target, expressed in the camera coordinate sytem. This set of 6 numbers unambiguously defines the extrinsic parameters. We use the transformation **from** the camera frame **to** the world frame, so that object coordinates are transformed the other way, **from** the world **to** the camera, with the same transform.

The origin of the right-handed camera reference frame is positioned at the center of projection of the lens. So the camera looks from this point to the outside world. The z'-axis of the camera frame corresponds to the optical-axis. This is an imaginary axis passing through the middle of the lens (the lens -or system of lens- is supposed perpendicular to it). The x'-axis is parallel to the horizontal axis of the image, from left to right, and the y'-axis is parallel to the vertical axis in the down direction.

2.2 Imaging process: the pin-hole camera model and the intrinsic parameters

A camera transforms the real 3D space into a 2D image plane. Usually, the transformation is assumed to be a conic projection. The center of the projection is defined as the camera center (see above). This camera model uses the pin-hole assumption.

The center of the image (u_0, v_0), expressed in pixels, is defined as the intersection between the optical axis and the image on the CCD matrix (the sensor plane is also assumed to be perpendicular to the optical axis). This center is usually close to the ideal image center, but can wander significantly from it.

Sampling is realized with a rectangular pattern, whose dimensions are (a_u, a_v). Therefore, the vertical and horizontal sample sizes are $1/a_u$ and $1/a_v$, respectively. These ratios reflect the combined contribution of both the camera and the digitizer sampling processes.

The coordinates (u_0, v_0) of the image center as well as the dimensions (a_u, a_v) are the 4 intrinsic parameters of the pinhole camera model.

Thus, the relationship between the 3D coordinates (x', y', z') of a 3D point in the camera reference frame and its image coordinate (u, v) is:

$$u = u_0 + a_u \frac{x'}{z'}$$

$$v = v_0 + a_v \frac{y'}{z'}$$

Remember, it is assumed here that the final image has its origin $(0, 0)$ at the upper left corner.

2.3 From 3D to 2D

By combining the rigid displacement, the conic projection and the image sampling, we determine the relationship between the coordinates of a space point (x, y, z)

expressed in the absolute reference frame and its image (u, v). This equation has a simple expression with homogeneous notations ($X = (x, y, z, 1)^t$ for each point of the euclidian space).

$$u = \frac{L_1 X}{L_3 X}$$
$$v = \frac{L_2 X}{L_3 X}$$

or

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \begin{pmatrix} L_1 \\ L_2 \\ L_3 \end{pmatrix} X$$

Thus the complete transformation is represented by a 3×4 matrix, often named M:

$$M = \begin{pmatrix} L_1 \\ L_2 \\ L_3 \end{pmatrix} = \begin{pmatrix} a_u & 0 & u_0 & 0 \\ 0 & a_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix}$$

2.4 Model with distortions

The previous description is a very simple ideal model, usually refined by taking into account the optical distortions of the camera lens, for sake of realism. Optical aberrations account for the discrepancy between the real and the ideal image of a light ray on the camera sensor. As we deal here with geometric calibration, only geometric distortions are considered. Distortions are a special type of aberrations independent of the orientation of the incoming light ray. The modeling of distortion effects adds nonlinear terms to the projective relationship between a 3D space point and an image point. To get the complete 3D to 2D correspondence relations it is therefore necessary to combine the rigid displacement, the conic projection, the distortions, and the sampling, in this order. Usually, the three more important distortions are:

- The RADIAL DISTORTION produces generally the most important effect (model #7):

$$x_d = x + \alpha x(x^2 + y^2)$$

$$y_d = y + \alpha y(x^2 + y^2)$$

- The DECENTERING of a lens on the axe of view is described by the following equation (model #10):

$$x_d = x + \alpha(3x^2 + y^2) + 2\beta xy$$

$$y_d = y + \beta(x^2 + 3y^2) + 2\alpha xy$$

- The effect of an error in lens parallelism is the THIN PRISM distortion (model #11):

$$x_d = x + \alpha(x^2 + y^2)$$

$$y_d = y + \alpha(x^2 + y^2)$$

The last two distortion types can be made very low in good quality camera lenses. More exotic distortion types are added in *CamCal*, for instance:

- RADIAL DISTORTION effect with a gap between image and distortion centers (model #8):

$$x_d = x + \alpha x(x^2 + y^2) + \beta x$$

$$y_d = y + \alpha y(x^2 + y^2) + \gamma y$$

- SPECIAL RADIAL DISTORTION effect (model #9):

$$x_d = x + \alpha x(x^2 + y^2)$$

$$y_d = y + \beta y(x^2 + y^2)$$

- SECOND ORDER DISTORTIONS (model #12):

$$x_d = x + \alpha x^2 + \beta y^2$$

$$y_d = y + \gamma x^2 + \delta y^2$$

- SECOND ORDER + RADIAL DISTORTIONS (model #13):

$$x_d = x + \alpha x^2 + \beta y^2 + \epsilon x(x^2 + y^2)$$

$$y_d = y + \gamma x^2 + \delta y^2 + \epsilon y(x^2 + y^2)$$

- DECENTERING + THIN PRISM DISTORTIONS (model #14):

$$x_d = x + \alpha(3x^2 + y^2) + 2\beta xy + \gamma(x^2 + y^2)$$

$$y_d = y + \beta(x^2 + 3y^2) + 2\alpha xy + \gamma(x^2 + y^2)$$

- DECENTERING + RADIAL DISTORTIONS (model #15):

$$x_d = x + \alpha(3x^2 + y^2) + 2\beta xy + \gamma x(x^2 + y^2)$$

$$y_d = y + \beta(x^2 + 3y^2) + 2\alpha xy + \gamma y(x^2 + y^2)$$

- THIN PRISM + RADIAL DISTORTIONS (model #16):

$$x_d = x + \alpha(x^2 + y^2) + \beta x(x^2 + y^2)$$

$$y_d = y + \alpha(x^2 + y^2) + \beta y(x^2 + y^2)$$

The default camera model for accurate estimation is the radial distortion model, but any of the previous models can be selected by using the following option ”`-m <camera_model_id>`” while running *CamCal*.

3 Camera calibration

Complete camera calibration consists in estimating both intrinsic and extrinsic parameters. Knowing the camera position would be easy if one could register object positions accurately (say, with a laser-based range finder). Unfortunately this is not possible, because we need to recover an imaginary position (the optical center, which cannot be observed directly) somewhere inside the camera.

Moreover, this point does not even have a fixed place in the camera. It is known that the center of projection changes when a zoom lens is being used, for example. In other words, one needs to recalibrate the camera everytime the focal length is changed.

As we have discussed, an image of a certain calibration set-up is needed for calibration (see test image included in this distribution). The 3D geometry of the template calibration object must be known accurately, so that in the operation

$$\text{Image} = M(\text{3D object})$$

both image and object are known and the projection M can be estimated.

The steps of the whole calibration process are the following:

a) Edge extraction from original image by Canny-Deriche's edge detector. For details see [1]:

b) From edge image, closed contour chains are linked using Giraudon's chain follower/linker. For details see [4] (in french) or [5]:

c) Extract image features from calibration target. These features are the center of gravity of image ellipses which are the images of planar ellipses drawn on two perpendicular grids (which form the calibration target).

d) Match 2D features with the set of known 3D patterns which are 3D ellipses on two perpendicular boards. Matching is made easy by the grid-like nature of the arrangement.

e) Initial approximate pin-hole camera calibration with Toscani's algorithm. For details see [9] , p33, 34, 35 and p21, 22, [2] (in french) and [3].

f) Accurate iterative camera calibration (with distortions) by minimizing of error function. You can look in the paper [8, 7, 6] for more details (in french) or `http://www-rocq.inria.fr/syntim/textes/calib-eng.html`.

Tsai's algorithm is not implemented in this version of *CamCal*. To download the Tsai's software, see `http://www.cs.cmu.edu/afs/cs.cmu.edu/user/rgw/www/TsaiCode.html` or the following adresse
`ftp://ftp.teleosresearch.com/VISION-LIST-ARCHIVE/SHAREWARE/CODE/CALIBRATION/`.

4 Input data

Basically, camera calibration algorithm inputs are:

- a gray-level 2D image of the calibration setup,
- the 3D geometric model of the calibration setup,

- some control parameters.

Optionally, other data inputs can be used:

- the edge image of the original image (if you want to use another edge detector than the one provided),
- the 2D points of center of gravity of the ellipses from the original image (if you want to provide pre-processed data). In this case, no image is needed.

4.1 Image acquisition

Digital 2D image frames are usually obtained via a CCD camera connected to a digitizer. An image frame is a name for all the pixels (PICture ELement) of an image, whereas an image contains more than only an image frame buffer. In particular it also contains the image size. The pixels are all delivered consecutively, from left to right and from top to bottom. The coding of image frame buffers depends a lot on your digitizing equipment. In our case, it is supposed to be a monochrome image frame buffer. Thus, each pixel is represented by an 8-bit value. Color images, where each pixel is coded as three values for the three basic colors: red, green and blue (also called an RBG-image), must be preprocessed and converted into a gray-level image.

4.2 Image

After the acquisition, all the pixels of an image frame are usually delivered in a file buffer. This file is the raw image data used by *CamCal*. *CalCam* only accept the specific raw image file format. But an image has a format whose only role (for us) is to describe the size of the image frame by two values. It just gives number of columns `<dimx>`, and number of lines `<dimy>`.

In summary, *CamCal* basically needs the following 2D informations:

- **Image frame buffer**: the values of all the pixels saved as unsigned chars `<image_file>`. This file is specified to *CamCal* by following option ”`-i <image_file>`” (by default `<image_file> = example`).

- **Format**: number of columns `<dimx>`, and number of lines `<dimy>`. These sizes are specified with following options "`-x <dimx>`" and "`-y <dimy>`" (`<dimx>=640` and `<dimy>=512` by default).

The first step of the camera calibration is to obtain an edge image from the original image in `<image_file>` with Canny-Deriche's algorithm. The edge image is stored in a separate file `<image_file>.edg`, in the same directory.

If you want to change the edge image file name, or use our own edge image, use *CamCal* option "`-e <edge_image_file>`" (by default `<edge_image_file> = example.edg`). Note that the edge image buffer is not recomputed if a file of the same name is present on the disk.

4.3 2D points

Correct 2D measurements are required to perform an accurate estimation of the camera model. A feature detection accuracy of $1/10^{th}$ pixel is typically necessary, in real situations, for a 5 mm. positionning of the calibration set-up.

After steps a) though d) of the camera calibration, 2D positions of the centers of gravity of the ellipses present in the image are obtained from the original image with high accuracy. Result is stored in a text file `<image file>.pt`, in the same directory.

If you want to change the 2D points filename, or use your own 2D points file, use *CamCal* option "`-p <2d_points_file>`" (by default `<2d_points_file> = example.pt`). Note that the 2D points file is not recomputed if it is already present on the disk.

4.4 3D calibration set-up

The intrinsic and extrinsic parameters are computed by estimating the orientation of all the circles of the calibration set-up on the image. To make the overall estimation as significant as possible many circles are drawn on the object. Usually, our calibration set-up is composed of two orthogonal planes where a regular grid of 7×6 disks are affixed at known locations. The accuracy of the calibration set-up should be high (in our case the target was machined with a $1/30^{th}$ mm. precision).

The reference object is built with two orthogonal planes, where elliptical shapes of an uniform color are painted on an uniform background (e.g. white shape on black background). Moreover, for accurate results, the calibration set-up must be illuminated by ambient light and the resulting image have good contrast. The calibration set-up surface must be as diffuse as possible.

In our camera calibration experiments, the calibration set-up is placed at a distance of about 1 or 2 meters, oriented in front of the camera. What is important is the apparent ellipse radius in the image, which should be more than 15 pixels. (the bigger the better, with the obvious trade-off that all the calibration target must be visible. Avoid using lenses with large field of view, which tend to distort images a lot).

The upper-left corner of the object is chosen as the origin of the reference frame to describe positions and orientations of the circular features. These circles are given from left to right and from top to bottom, for each face from left to right.

You must provide *CamCal* with the 3D geometry of the calibration setup, described in a text file in the following maner:

```
<type> <horizontal size of a grid> <vertical size of a grid> <number of grids>
<circle #1>
<x center circle #1> <y center circle #1> <z center circle #1>
<x first axis circle #1> <y first axis circle #1> <z first axis circle #1>
<x second axis circle #1> <y second axis circle #1> <z second axis circle #1>
.
.
.
<circle #n>
<x center circle #n> <y center circle #n> <z center circle #n>
<x first axis circle #n> <y first axis circle #n> <z first axis circle #n>
<x second axis circle #n> <y second axis circle #n> <z second axis circle #n>
```

When the calibration target is a set of ellipses, the `<Type>` value is always 1. But *CamCal* can also be used to perform camera calibration based on simple point-to-point correspondences (used point are then ellipse centers). For this, specify the `<type>` value to 0. In this case, the calibration set-up is described in the following manner:

```
<type> <horizontal size of a grid> <vertical size of a grid> <number of grids>
<point #1>
<x point #1> <y point #1> <z point #1>
.
.
.
<point #n>
<x point #n> <y point #n> <z point #n>
```

A default set-up file is provided (but you shall have to build your calibration target according to its specifications).

If you want to use your own 3D set-up file, use *CamCal* following option ”`-g <3d_setup_file>`” (by default `<3d_setup_file> = setup.ex`).

4.5 Control parameters

Some parameters can be changed in case the program fails to produce a calibration with the default values.

- ”`-b <size in pixel of the blur>`” size of the blurring zone around an edge (3 by default).
- ”`-l <distance in pixel between two ellipses>`” shortest distance between two ellipse edges (7 by default).
- ”`-c <distance close>`” if the gap between one end and the other end of a chain is bigger than this threshold, the chain is considered open (4 by default). Calibration retains only closed chains, according to this criterion.
- ”`-s <down value>`” down-hysteresis threshold between 0 and 255 (20 by default).
- ”`-S <up value>`” up-hysteresis threshold between 0 and 255 (40 by default).
- ”`-m <camera model id>`” between 0 and 16 (#7 by default):
  - # 0: point to point correspondance without distortion, and pixel error minimisation,
  - # 1: point to point correspondance without distortion, mm error,
  - # 2: point to point correspondance with a fixed center, mm error, without distortion,
  - # 3: point to point correspondance, mm error, radial distortion,
  - # 4: ellipse to point correspondance, pixel error, without distortion,
  - # 5: ellipse to point correspondance, mm error, without distortion,

  - # 6: ellipse to point correspondance with fixed center, mm error, without distortion,
  - # 7: ellipse to point correspondance, mm error, radial distortion,
  - # 8: ellipse to point correspondance, mm error, radial distortion and gap between distortion and image centers,
  - # 9: ellipse to point correspondance, mm error, special radial distortion,
  - # 10: ellipse to point correspondance, mm error, decentering distortion,
  - # 11: ellipse to point correspondance, mm error, thin prism distortion,
  - # 12: ellipse to point correspondance, mm error, second order distortions,
  - # 13: ellipse to point correspondance, mm error, second order and radial distortion,
  - # 14: ellipse to point correspondance, mm error, thin prism and decentering distortion,
  - # 15: ellipse to point correspondance, mm error, radial and decentering distortion,
  - # 16: ellipse to point correspondance, mm error, radial and thin prism distortion.

As a summary, others options are:

- ”`-g <3d_setup_file>`” filename of the calibration setup (by default we have `<3d_setup_file> = setup.ex`).
- ”`-i <image_file>`” filename of the original image (by default we have `<image_file> = example`).
- ”`-x <dimx>`” and ”`-y <dimy>`” image size of the original image (we have `<dimx>=640` and `<dimy>=512` by default).
- ”`-e <edge_image_file>`” filename of the edge raw format image extracted from the original image (`<edge_image_file> = example.edg` by default).

- ”`-p <2d_points_file>`” name of the 2D points file extracted from the original image (by default `<2d_points_file> = example.pt`).
- ”`-r <camera_filename>`” filename of the resulting camera (by default `<image_filename>.cam`).

All the filenames may be preceeded by a pathname.

5 Output data

Estimated parameters for the camera model and position are saved in the output file.

If you want to change the resulting camera filename, use the following option ”`-r <camera_filename>`” with *CamCal* (by default the camera name is `<image_filename>.cam` and `<image_filename>.cam.tos` contains the perspective matrix outputs). Note that camera files are not recomputed if they are already present on the disk.

6 Implementation aspects

6.1 Style

All Programs are written in the C language. Routines have been grouped according to their task type, so that edge detection, chain extraction, image point extraction and camera calibration each have their own set of files.

Task	Files
edge detection	Deriche.c Deriche.h
chain extraction	Chain.c ChainIma.c ChainExam.c Chain.h
chain approximation	AppChain.c AppChain.h
image point extraction	ExtPoint.c ExtPoint.h ExtBary.c ExtBary.h
camera calibration	Toscani.c Toscani.h CalIter.c CalIter.h
main, I/O	CamCal.c CamCal.h
linear algebra	Matrix.c Matrix.h
memory management	Mem.c Mem.h

Besides, for each task type there is a special header file with all the information required to make use of the functions.

6.2 Compiling

In order to compile the programs, a Makefile is included. This makefile should work fine on a Unix system.

To compile, you can just send the command in the source directory to obtain the *CamCal* binary:

```
>make
```

6.3 Running

An example of data is given to check if your *CamCal* executable runs properly:

- a raw image file named `example`,
- a 3d set-up file named `setup.ex`.

To run the calibration, just run in the source directory:

```
>CamCal
```

You should obtain results identical to the references, i.e. :

- `exedg.ref`: raw image of the extracted edges,
- `expt.ref`: 2d points extracted from original and edge files,
- `excamtos.ref`: perspective matrix obtained by Toscani's algorithm,
- `excam.ref`: final camera result.

7 Contact and copyright

CamCal has been designed by Jean-Philippe Tarel
INRIA
Domaine de Voluceau, Rocquencourt,
BP 105, 78153 LE CHESNAY Cedex
France

Tel: 39.63.54.79
Fax: 39.63.57.74

E-mail: Jean-Philippe.Tarel@inria.fr
WWW: http://www-syntim.inria.fr/∼tarel/

The Inter deposit Digital Number of *CamCal* software version 1.0 is:

IDDN.FR.001.290004.00.R.P.1996.000.21000.

so the *CamCal* software is under INRIA copyright 1996.

Permission to use, copy, modify and distribute this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of authors and the Institut National de Recherches en Informatique et Automatique (INRIA) not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

CamCal software is distributed by INRIA. For distribution, contact Jean-Philippe Tarel and/or Jean-Louis Bouchenez (Jean-Louis.Bouchenez@inria.fr).

References

[1] Deriche (R.). – Using canny's criteria to derive an optimal edge detector recursively implemented. *International Journal of Computer Vision*, vol. 1, n° 2, 1987.

[2] Faugeras (O.), Lustman (F.) et Toscani (G.). – Calcul du mouvement et de la structure à partir de points et de droites. *In : 6ème congrès AFCET, Reconnaissance des Formes et Intelligence Artificielle*, pp. 75–89. – Antibes, November 1987.

[3] Faugeras (O.D.) et Toscani (G.). – The calibration problem for stereo. *In : Proceedings, CVPR '86 (IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Miami Beach, FL, June 22–26, 1986)*. pp. 15–20. – IEEE.

[4] Giraudon (G.). – *Chainage efficace de contours*. – Rapport de recherche 605, INRIA, 1987.

[5] Giraudon (G.). – A real time parallel edge following in single pass. *In : Workshop on Computer Vision (Proceedings published by IEEE Computer Society Press, Washington, DC.)*, pp. 228–230. – Miami Beach, FL, 1987.

[6] Tarel (Jean-Philippe). – *Calibration de caméra fondée sur les ellipses*. – Rapport de recherche n° 2200, INRIA, 1994. http://www-rocq.inria.fr/syntim/textes/calib94-eng.html.

[7] Tarel (Jean-Philippe) et Gagalowicz (André). – Calibration de caméra à base d'ellipses. *In : 9ème congrès AFCET, Reconnaissance des Formes et Intelligence Artificielle*. – Paris, France, 1994.

[8] Tarel (Jean-Philippe) et Gagalowicz (André). – Calibration de caméra à base d'ellipses. *Traitement du Signal*, vol. 12, n° 2, 1995, pp. 177–187. – http://www-rocq.inria.fr/syntim/textes/calib-eng.html.

[9] Toscani (G.). – *Systèmes de calibration et perception du mouvement en Vision Artificielle*. – Thèse de PhD, Université Paris-Sud, 1987.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399