Cone of view camera model using conformal geometric algebra for classic and panoramic image sensors

Thibaud Debaecker, Ryad Benosman, Sio-Hoi Ieng

Cone of view camera model using conformal geometric algebra for classic and panoramic image sensors.

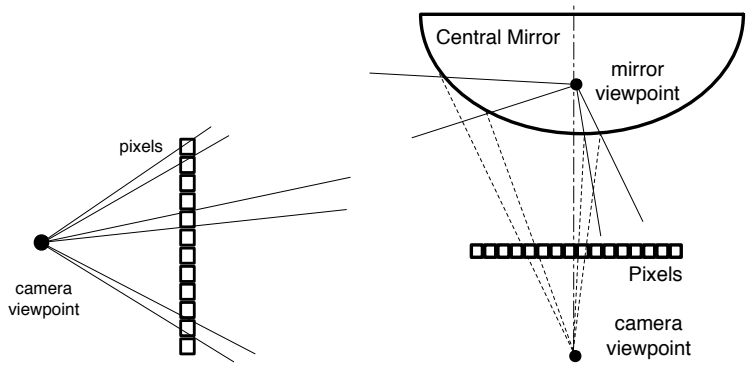
Thibaud Debaecker, Ryad Benosman and Sio-Hoi Ieng

Institut des Systèmes Intelligents et de Robotique
Université Pierre et Marie Curie,
4 place Jussieu, 75005 Paris
{debaecker, benosman, ieng}@isir.fr

Abstract. Camera calibration is a necessary step in 3D computer vision in order to extract metric information from 2D images. Calibration has been defined as the non parametric association of a projection ray in 3D to every pixel in an image. It is normally neglected that pixels have a finite surface that can be approximated by a cone of view that has the usual ray of view of the pixel as a directrix axis. If this pixels' physical topology can be easily neglected in the case of perspective cameras, it is an absolute necessity to consider it in the case of variant scale cameras such as foveolar or catadioptric omnidirectional sensors which are nowadays widely used in robotics. This paper presents a general model to geometrically describe cameras whether they have a constant or variant scale resolution by introducing the new idea of using pixel-cones to model the field of view of cameras rather than the usual line-rays. The paper presents the general formulation using twists of conformal geometric algebra to express cones and their intersections in an easy and elegant manner, and without which the use of cones would be too binding. The paper will also introduce an experimental method to determine pixels-cones for any geometric type of camera. Experimental results will be shown in the case of perspective and omnidirectional catadioptric cameras.

1 Introduction

A large amount of work has been carried out on perspective cameras introducing the pinhole model and the use of projective geometry. This model turns out to be very efficient in most cases and it is still widely used within the computer vision community. Several computation improvements have been introduced [1], nevertheless this model has limitations. It can only be applied to projective sensors. Slight fluctuations in the calibration process lead to the fact that two rays of view of pixels representing the same point in the scene, rarely exactly intersect across the same $3D$ point they should represent. Finally this model fails to introduce the reality of the sensor, as the approximation of the field of view of the pixel is restricted to a ray that can not match the real topology of the pixel



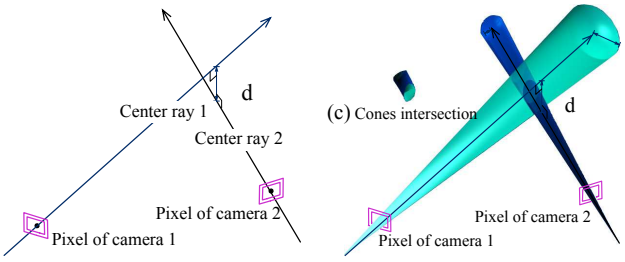
(a) Pixel-cones of a perspective camera.

(b) Catadioptric pixel-cones after reflection on a central mirror

Fig. 1. Pixel-cones in the case of perspective cameras and variant scale sensors (here a central catadioptric sensor). All cones are almost the same in (a), whereas in (b) the pixel-cones vary drastically according to the position of the pixel within the perspective camera observing the mirror.

that corresponds to a surface on the image plane and not only a point. The limitations pointed out become drastically problematic with the appearance of new kinds of non linear visual sensors like foveolar retinas [2], or panoramic sensors (see [3] for an overview). As show in Fig. 1(a) in the case of a perpective cameras, the real field of view of pixels is a cone, in the perspective case all pixels produce similar cones of view. Cones being barely the same, it is easily understandable why cones can be approximated using lines in this case. As shown in Fig. 1(b), for a catadioptric sensor (combination of a perspective camera and a hyperboloid mirror) it becomes obvious that the approximation using rays will lead to large imprecisions specially in the computation of intersections, cones become then an absolute necessity. Different methods have been developed to cope with the issue of non perfect intersection of rays in the case of perspective cameras. Bundle adjustment is one of the major techniques (a complete overview of existing methods can be found in [4]), it consists in refining a visual reconstruction to produce jointly optimal 3D structure and more accurate viewing parameters estimates. Optimal means that these methods are based on cost function minimizations. This processing can be seen as a consequence of the use of rays rather than cones, as we will show later intersections may happen even if the central rays of pixels do not intersect. Non intersections do not necessarily reflect an imprecise calibration result. Recently, cones have been introduced [5] not to consider the pixel field of view but to model the uncertainties of ray directions. Cone aperture is there set as an arbitrary error parameter of matched points. Other approaches which do not use non least square minimization methods have been used to correct these problem in mutliple view geometry problematics. It still remains mathematical instead of physical sens approaches [6, 7].

As shown in Fig. 2(a), most of the times, two rays do not have an exact intersection in 3D space. In most cases, they are separated by a distance d which can be computed easily if the pose information are known. The intersection is considered acceptable if it is under a certain threshold. The cones defined by the surface of the pixel encompass the rays of view of the pixels (Fig. 2(b)), each ray of view corresponds to the directrix of each cone. The rays still do not intersect, but as shown in Fig. 2(b) the cones do fully meet as the corresponding volume of intersection shown in Fig. 2(c) is not zero. Applying a bundle adjustment in this case will not lead to the best solution as the perfect intersection of the ray because it does not necessarily correspond to the optimal intersection.
The aim of the paper is not to compare the bundle adjustment versus projective pixel-cone of view, but just to show that introducing cones gives the opportunity to be closer to the real physics of the intersection of pixels and thus generate more accurate situations. The topic of comparing the bundle adjustment versus volume optimization will surely be the topic of following paper. After explaining the importance of cones, determining experimentally the cones of a sensor introduces the necessity of developing a calibration procedure that will be presented in section 3. Among the existing calibration methods, there has been recently an effort to developp new methods that can handle the most general sensors non necessarily central or relying on caustics ([8, 9]) but in its most most general form as a set of viewpoints and rays of view. The raxel approach [10] gave an interesting model for sensors different from the classic pinholes cameras. The association of a pixel and a direction enables a wider range of camera calibration, but does not consider the non linear resolution aspect of the sensors, and the variation of the solid angle of each pixel field of view. [11] provides also a general geometric model of a camera, but again the field of view of pixels is not taken into account.



(a) Two rays of light in a 3D space.

(b) Cones of light.

Fig. 2. Difference between ray of light and cone of light approach.

This paper is structured as follows. After describing in section 2 the mathematical formulation of the general pixel-cones model using twists [12], an exper-

imental protocol to find the pixel cones of light is presented in section 3. Section 4 shows the results got from this protocol applied on a pinhole camera and a catadioptric sensor. Conclusions and future works are included in section 5.

2 General model of a cone-pixels camera

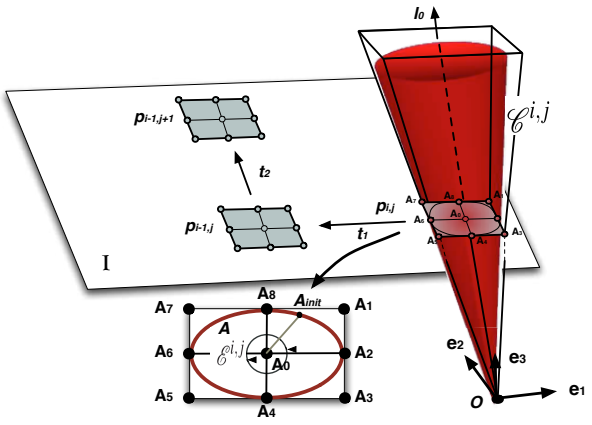


Fig. 3. Cones Geometric settings.

This section will present the general model of a camera using cone-pixels. There are several possible ways to write the equation of a cone. A single-sided cone with vertex V, axis ray with origin at V, unit-length direction A and cone angle $\theta \in (0, \pi/2)$ is defined by the set of points X such that vector $X-V$ forms an angle θ with A. The algebraic condition is $A \cdot (X - V) = |X - V|cos(\theta)$. The solid cone is the cone plus the region it bounds, specified as $A \cdot (X - V) \geq |X - V|cos(\theta)$. It is quite painfull to compute the intersection of two cones, and this tends to become even more complicated integrating rigid motions parameters between the two cones. Conformal geometric algebra through the use of twists allows to construct a wide variety of kinematics shapes. There exist many ways to define algebraic curves [13], among them twists can be used to generate various curves and shapes [12]. A kinematic shape is a shape that results from the orbit effect of a point under the action of a set of coupled operators. The nice idea is that the operators are what describes the curve (or shape), as introduced in [12] these operators are the motors which are the representation of $SE(3)$ in $\mathcal{R}_{4,1}$. The use of twists gives a compact representation of cones and brings the heavy computation of the intersection of two general cones to a simple intersection of lines. The reader unfamiliar with geometric algebra, should refer to [14, 15] for an overview of geometric algebra, examples of its use in computer vision can be found in [16, 17].

2.1 Geometric settings

As shown in Fig. 3 in the case of a perspective camera, the image plane here represented by I contains several rectangular pixels $p(i,j)$, where i,j corresponds to the position of the pixel. Considering $p(i,j)$, its surface is represented by a rectangle defined by points $A_0...A_8$, with A_0 corresponding to the center of the rectangle.

Given a line $\underline{\mathbf{l}}$ (with unit direction) in space, the corresponding motor describing a general rotation around this line is given by : $\mathcal{M}(\theta,\underline{\mathbf{l}}) = exp(-\frac{\theta}{2}\underline{\mathbf{l}})$. The general rotation of a point $\underline{\mathbf{x}}_\theta$ around any arbitrary line $\underline{\mathbf{l}}$ is :

$$\underline{\mathbf{x}}_\theta^{'} = \mathcal{M}(\theta,\underline{\mathbf{l}})\underline{\mathbf{x}}_\theta\tilde{\mathcal{M}}(\theta,\underline{\mathbf{l}}) \tag{1}$$

The general form of *2twist* generated curve is the set of points defined $\underline{\mathbf{x}}_\theta$ such as :

$$\underline{\mathbf{x}}_\theta = \mathcal{M}^2(\lambda_2\theta,\underline{\mathbf{l_2}})\mathcal{M}^1(\lambda_1\theta,\underline{\mathbf{l_1}})\underline{\mathbf{x}}_0\tilde{\mathcal{M}}^1(\lambda_1\theta,\underline{\mathbf{l_1}})\tilde{\mathcal{M}}^2(\lambda_2\theta,\underline{\mathbf{l_2}}) \tag{2}$$

In what follows we are interested in generating ellipses that correspond to the values $\lambda_1 = -2$ and $\lambda_2 = 1$. $\underline{l_1}$ and $\underline{l_2}$ are the two rotation axis needed to define the ellipses [12].

Considering a single pixel $p_{i,j}$ (see Fig. 3), its surface can be approximated by the ellipse generated by a point A that rotates around point A_0 with a rotation axis corresponding to $\mathbf{e}_3$ normal to the plane I. The ellipse $\mathcal{E}^{i,j}(\theta)$ generated corresponding to the pixel $p_{i,j}$ is the set of all the positions of A :

$$\forall\theta \in [0,..,2\pi],\ \mathcal{E}^{i,j}(\theta) = \{\underline{\mathbf{A}}_\theta = \mathcal{M}^2(\theta,\underline{\mathbf{l_2}})\mathcal{M}^1(-2\theta,\underline{\mathbf{l_1}})\underline{\mathbf{A}}_0\tilde{\mathcal{M}}^1(-2\theta,\underline{\mathbf{l_1}})\tilde{\mathcal{M}}^2(\theta,\underline{\mathbf{l_2}}) \mid \}$$

The initial position of $\underline{\mathbf{A}}$ is to $\underline{\mathbf{A}}_{\text{init}}$. The elliptic curve is generated by setting the two connected twists so that to obtain an ellipse with principal axis $(\overline{A_8A_0},\overline{A_6A_0})$ in order to fit the rectangular surface of the projection of the pixel as shown in Fig. 3. It is now possible to generate the cone corresponding to the field of view of the pixel. We set the line $\underline{\mathbf{l}}^*_{\underline{\mathbf{0}}_{i,j}}$, cone axis corresponding to $p_{i,j}$ as :

$$\underline{\mathbf{l}}^*_{\underline{\mathbf{0}}_{i,j}} = e \wedge \underline{\mathbf{O}} \wedge \underline{\mathbf{A}}_0.$$

Let the line $\underline{\mathbf{l}_{\mathbf{OA}}}^{i,j}$ be the generatrix of the cone. The pixel-cone of view of $p_{i,j}$ is the cone $\mathcal{C}^{i,j}$ defined by :

$$\mathcal{C}^{i,j}(\theta) = \mathcal{M}(\theta,\underline{\mathbf{l}_{\mathbf{0}}}_{i,j})\underline{\mathbf{l}_{\mathbf{OA}}}^{i,j}\tilde{\mathcal{M}}(\theta,\underline{\mathbf{l}_{\mathbf{0}}}_{i,j}) \tag{3}$$

with $\mathcal{M}(\theta,\underline{\mathbf{l}_{\mathbf{0}}}_{i,j}) = exp(-\frac{\theta}{2}\underline{\mathbf{l}_{\mathbf{0}}}_{i,j})$. Note the equivalent expression of the cone using the outer product generating a line after having generated an ellipse from the point A:

$$\mathcal{C}^{i,j}(\theta) = e \wedge \underline{\mathbf{O}} \wedge \left(\mathcal{M}(\theta,\underline{\mathbf{l}_{\mathbf{0}}}_{i,j})\underline{\mathbf{A}}^{i,j}\tilde{\mathcal{M}}(\theta,\underline{\mathbf{l}_{\mathbf{0}}}_{i,j})\right) \tag{4}$$

The same process is to be applied again after translating the pixel $p_{i,j}$ using $\mathbf{t_1}$ and $\mathbf{t_2}$, that corresponds to the translation to switch from one pixel to the

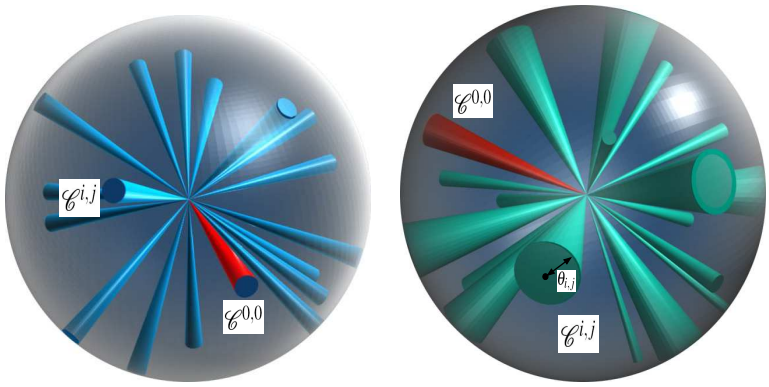
other. The projection $p_{i,j}$ of a pixel is moved to a next pixel :

$$p_{i+1,j+1} = \mathcal{T}^2(\mathbf{t}_2)\mathcal{T}^1(\mathbf{t}_1)p_{i,j}\tilde{\mathcal{T}}^1(\mathbf{t}_1)\tilde{\mathcal{T}}^2(\mathbf{t}_2)$$

where $\mathcal{T}(\mathbf{t})$ correspond to a translation operator in CGA.

2.2 The general model of a central cone-pixel camera

The general form of a central sensor whether it is linear scale (cones vary slightly) or variant scale is the expression of a bundle of cones. All cones $\mathcal{C}^{i,j}$ will be located using spherical coordinates and located according to an origin set as the cone $\mathcal{C}^{0,0}(\theta)$, that has $\mathbf{e_3}$ as a principal axis. The general form of a central linear scale



(a) A bundle of pixel-cones of a central linear sensor.

(b) A bundle of pixel-cones of a central variant linear sensor

Fig. 4. Different configurations of pixel-cones, in the case of linear and variant scale sensors. In red the principal cone according which every other is located.

camera (Fig. 4(a)) is then simply given by :

$$\mathcal{C}^{i,j}_{\phi,\psi}(\theta) = \mathcal{M}^2(\psi, \mathbf{e_{23}})\mathcal{M}^1(\phi, \mathbf{e_{13}})\mathcal{C}^{0,0}_{0,0}(\theta)\tilde{\mathcal{M}}^1(\phi, \mathbf{e_{13}})\tilde{\mathcal{M}}^2(\psi, \mathbf{e_{23}}) \tag{5}$$

with θ, ϕ denote the spherical coordinates of the cone.
The general form of a central variant scale sensor is slightly different. Each cone having a different size, cones need to be defined according to their position. A cone $\mathcal{C}^{i,j}_{\phi,\psi}$ is then defined by the angle between its vertex and generatrix. Due to the central constraint, all cones have the same apex. The rotation axis $\underline{\mathbf{l_0}}^{i,j}$ of $\mathcal{C}^{i,j}_{\psi,\phi}$ giving its location is computed from $\underline{\mathbf{l_0}}^{0,0}$ of $\mathcal{C}^{0,0}_{0,0}$ is then :

$$\underline{\mathbf{l_0}}^{i,j} = \mathcal{M}^2(\psi, \mathbf{e_{23}})\mathcal{M}^1(\phi, \mathbf{e_{13}})\underline{\mathbf{l_0}}^{0,0}\tilde{\mathcal{M}}^1(\phi, \mathbf{e_{13}})\tilde{\mathcal{M}}^2(\psi, \mathbf{e_{23}}) \tag{6}$$

Its generatrix $\underline{\mathbf{l}}^{i,j}$ is defined as:

$$\underline{\mathbf{l}}^{i,j} = \mathcal{M}(\theta, \mathbf{e_{12}})\underline{\mathbf{l_0}}^{i,j}\tilde{\mathcal{M}}(\theta, \mathbf{e_{12}}) \tag{7}$$

The expression of $\mathcal{C}^{i,j}_{\phi,\psi}$ can then be computed using equation(3).

2.3 Discrete motion

In two generalized images, a correspondence between pixel (i_1, j_1) in the first image and pixel (i_2, j_2) in the second image has been generally defined by the epi-polar constraint. The generalized epi-polar constraint provides the extended Essential matrix [18], describing the intersection of two rays of view . For practical purposes, is never verified because of the uncertainties and especially because pixel fields of view are cones whose intersection can not be approximated by a simple line intersection.

In a cone-pixel approach, a correspondence between pixels means that the intersection between both cones of view $\mathcal{C}^{i_p,j_p}_{\phi_p,\psi_p}$ and $\mathcal{C}^{i_q,j_q}_{\phi_q,\psi_q}$ is not empty. To simplify the notations in the expression of this intersection, let $\mathcal{C}^m_k$ denote the cone expression of a pixel (i_m, j_m) in the frame relative to the image k.

To express a rigid body motion in $\mathcal{R}_{4,1}$ between both coordinate frames, the consecutive application of a rotor and translator can be written as their product. Such an operator is denoted as $\mathbf{M}$ as a combination of a rotor $\mathbf{R}$ and a translator $\mathbf{T}$:

$$\mathbf{M} = \mathbf{TR} \quad \text{with} \quad \mathbf{R} = exp\big(-\frac{\zeta}{2}\mathbf{l}\big) \quad \text{and } \mathbf{T} = exp\big(\frac{\mathbf{et}}{2}\big) \tag{8}$$

It comes that:

$$\mathcal{C}^m_1 = \mathbf{M}\mathcal{C}^m_2\tilde{\mathbf{M}} \tag{9}$$

Using the meet product in $\mathcal{R}_{4,1}$ to express the intersection $P_{p,q}$, we get:

$$P_{p,q} = \{\mathcal{C}^p_1 \vee \mathcal{C}^q_1\} = \{\mathcal{C}^p_1 \vee \mathbf{M}\mathcal{C}^q_2\tilde{\mathbf{M}}\} \neq \emptyset. \tag{10}$$

involving that:

$$(\mathcal{C}^p_1 \wedge \mathcal{C}^q_1)I^{-1} = (\mathcal{C}^p_1 \wedge \mathbf{M}\mathcal{C}^q_2\tilde{\mathbf{M}})I^{-1} = 0, \tag{11}$$

where I is the classic pseudo-scalar [19]. Then replacing with the equation (4) and changing the representation from the conformal space $\mathcal{R}_{4,1}$ to the projective space $\mathcal{R}_{3,1}$ thanks to the relation $\Theta_p = e_+ \cdot \Theta_c$ [16], it comes that:

$$0 = (e_+ \cdot \mathcal{C}^p_1) \wedge (e_+ \cdot \mathcal{C}^q_1)$$

$$0 = \big(\mathbf{O_1} \wedge \mathcal{M}(\theta^p, \underline{\mathbf{l}}^p_1)\underline{\mathbf{A}}^p_1\tilde{\mathcal{M}}(\theta^p, \underline{\mathbf{l}}^p_1)\big) \wedge \big(\mathbf{M}\underline{\mathbf{O}}_2\tilde{\mathbf{M}} \wedge \mathbf{M}\mathcal{M}(\theta^q, \underline{\mathbf{l}}^q_2)\underline{\mathbf{A}}^q_2\tilde{\mathcal{M}}(\theta^q, \underline{\mathbf{l}}^q_2)\tilde{\mathbf{M}}\big). \tag{12}$$

If the generatrix and the axis are merged, *ie* if $\mathbf{A}^m \wedge \underline{\mathbf{l_0}}^m = 0$, then this relation becomes:

$$\big(\mathbf{O_1} \wedge \mathbf{A}^p_1\big) \wedge \big(\mathbf{MO_2}\tilde{\mathbf{M}} \wedge \mathbf{MA}^q_2\tilde{\mathbf{M}}\big) = 0. \tag{13}$$

We find the general coplanarity constraint established by Perwass in [19] which leads to a conformal expression of the fundamental Matrix F and to the relation:

$$\alpha_1 \alpha_2 F = 0 \qquad \text{with} \quad \alpha_m = \begin{pmatrix} i_m \\ j_m \end{pmatrix} \tag{14}$$

The intersection of a cone with the image plane define in theory the pixel area. Because the experimental protocol which is presented in the following section provides the solid angle of view, it is possible to know the exact uncertainties δ_m such as:

$$(\alpha_1 + \delta_1)(\alpha_2 + \delta_2) F = 0 \qquad \text{with} \quad \delta_m = \begin{pmatrix} \delta_{i_m} \\ \delta_{j_m} \end{pmatrix} \tag{15}$$

In other words, this last relation means that, if two cones intersect, then two rays encompassed in these cones verify the coplanarity constraint. We get then:

$$\alpha_1 \alpha_2 F + \alpha_1 \delta_2 F + \alpha_2 \delta_1 F + \delta_1 \delta_2 F = 0 \tag{16}$$

If the intersection exists $P_{p,q}$ is not empty, the set of points then form a convex hull $H(P_{p,q})$ which volume can be computed using [20].

3 Experimental protocol

Cones being at the heart of the model, we will now give the experimental set up of the calibration procedure to provide an estimation of the cone of view of each pixel of a camera. The method is not restricted to a specific camera geometry, it relies on the use of multiple planes calibration ([10, 21]). As shown in Fig. 5, the camera to be calibrated is observing a calibration plane (in our case a computer screen), the aim is to estimate the cone of view of a pixel $p_{i,j}$, by computing for each position of the screen its projection surface $SP^k(i,j)$, k being the index of the calibration plane. The metric is provided using a reference high resolution calibrated camera (RC)[1] that observes the calibration planes. The reference camera uses the calibration planes to determine its parameters, the position of each plane is then known in the RC reference coordinates, and implicitly the metric on each calibration plane too. The impact surfaces $SP^k i,j)$ once determined on each screen lead normally as shown in Fig. 5 to the determination of all pixel-cones parameters.

Fig. 6 shows the experimental set up carried out for the experiments. The keypoint of the calibration protocol relies then on the determination of pixels' impact $SP^k(i,j)$. The idea is then to track the activity of each pixel $p(i,j)$ while they are (see Fig. 8) observing the screens calibration planes. At each position the screen is showing a scrolling white bar translating on a uniform black background (see Fig. 6). The bar will cause a change in the grey level values of pixels when

[1] 6 Megapixel digital single-lens Nikon D70 reflex camera fitted with 18-70-mm Nikkor micro lens. The micro lens and the focus are fixed during the whole experiment.

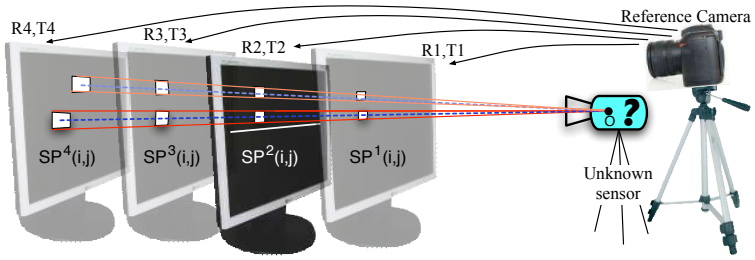
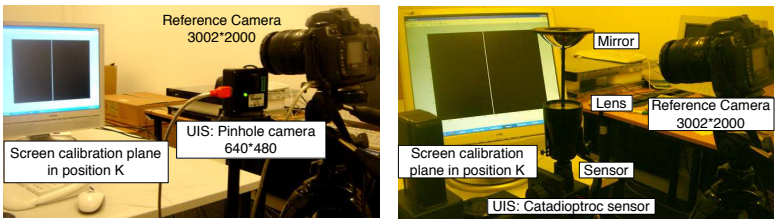


Fig. 5. Experimental protocol : Cone construction and determination of the center of projection of the sensor. The R_i, T_i represent the rigid motion between the reference camera and the calibration planes coordinates system.



(a) Case of a pinhole camera as UIS.

(b) Case of a catadioptric sensor as UIS.

Fig. 6. Experimental protocol for two kinds of UIS (Unknown Image Sensor) .

it is in their cone of vision. The pixels' gray-level increases from zero (when the bar is outside $SP^k(i,j)$) to a maximum value (when the bar is completely inside $SP^k(i,j)$), and decreases down to zero when the bar is again outside $SP^k(i,j)$. Fig. 8 gives a visual explanation of the process.

A sensitivity threshold can be chosen to decide pixels' activation. Using the reference camera calibration results, it is then possible once $SP^k(i,j)$ is determined to compute its edges as the positions of y_{in} and y_{out}, in the RC coordinate system. The bar is scrolled in two orthogonal directions providing two other edges x_{in} and x_{out} (Fig. 7), the envelope of $SP^k(i,j)$ is then completely known. The location and size of pixel-cones can then in a second stage be estimated once all $SP^k(i,j)$ are known. Cones are computed using the center of $SP^k(i,j)$ that give the rotation axis, the envelope is given by computing rays that pass through all the intersection points of the vertex of each $SP^k(i,j)$ corresponding to each pixel (Fig. 5).

4 Experimental results

The following experiments were carried out using PointGrey DragonFly®2, with a 640×480 resolution and a parabolic catadioptric sensor with a telecentric

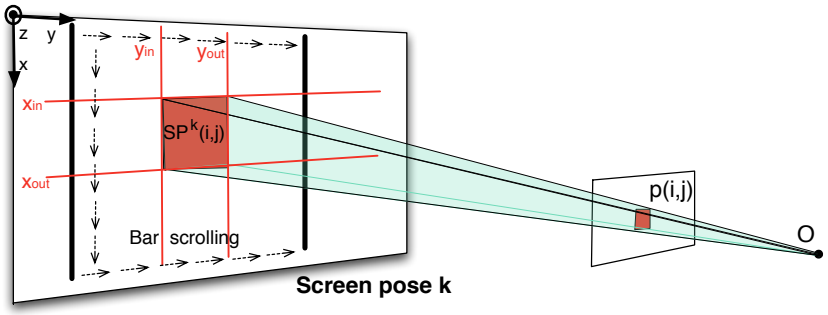


Fig. 7. Intersection surface $SP^k(i,j)$ between the calibration plane k and the cone $\mathcal{C}(i,j)$

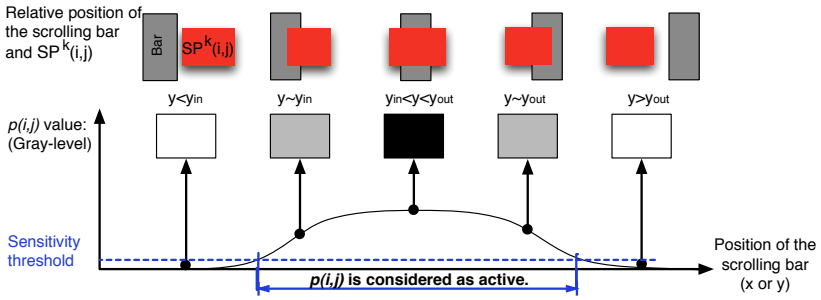


Fig. 8. Pixel response according to the scroll bar position: pixel activity.

lens, both are central sensors. (Fig. 6(a)). Fig. 9 shows cones reconstruction on $SP^1(i,j)$ and $SP^2(i,j)$ in the case of a pinhole camera. Only few cones were drawn, it is then a expected result to see repetitive pattern corresponding fields of squares of estimated pixels impact.

In the case of the catadioptric camera, it is a geometric truth that the aperture angle of each cone will increase as pixels are far from the optical axis of the camera. This phenomenon is experimentally shown in Fig. 10 where the evolution of the solid angle of pixel-cones are presented. In principle the solid angle should not depend on the position of the calibration plane that was used to compute it, the curves are then very close even if a small bias appears for very large cone-pixels at the periphery of the mirror where the uncertainties of the measure on the surface due to the non linearity of the mirror are the highest.

The method allows the estimation of the position of the central point of the calibrated sensor. in the case of the pinhole camera the calibration screens were located between $800 - 1050$ cm from the reference camera, while the camera to be calibrated was set few cm far (see Fig. 6). In order to obtain a ground truth data the pinhole camera was calibrated using classic ray method. Three

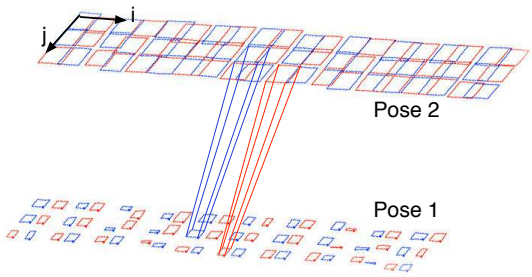


Fig. 9. Cones of view in the case of a pinhole camera.

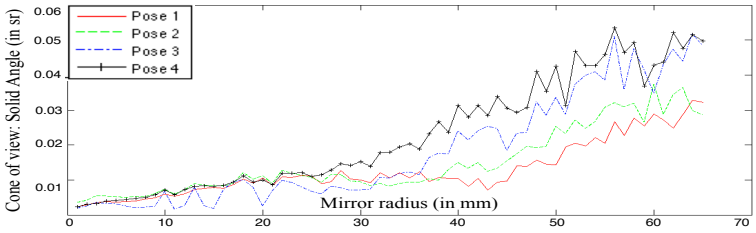


Fig. 10. Solid angle of view according to the mirror radius.

position of the optic center are then computed for comparison. the first one is given by the classic calibration, the second by the intersection of the rotation axis of estimated cones and the last one by the intersection of all rays (traced as shown in Fig. 9) representing estimated cones. The results are shown in Table.1.

Table 1. Central projection point estimation coordinates: case of a pinhole camera

mm	Ground Truth	Axis estimation	Error	Apex estimation	Error
x	-78,33	-78,97	0,65	-78,97	0,65
y	45,36	44,07	1,28	44,07	1,29
z	45,74	57,89	12,16	57,90	12,16

We notice that a single viewpoint has been found for each one, the classic ray method and the use of the rotation axis produce very similar results. There are slight variations in the position of the center using the third approach that can be explained by the fact that the calibrated portion of the sensor used to estimate cones was limited (55×60 pixels located around the center of the image). Concerning the catadioptric sensor, the results show that the cones intersect at

a single point. The a combination of a parabola and a telecentric lens can only produce a central sensor, so far the method proved to be efficient. The estimation of the position of the viewpoint using the principal axis of the estimated cones and all the rays that form the estimated cones produce similar results (in mm: $x = -23.38, y = 147.55, z = 384.79$ and $x = -23.32, y = 147.33, z = 385.57$). The mean distance between the rotation axis and their estimated single point is 3.71 mm. The mean distance between the apex and their estimated single point is 2.97 mm.

5 Conclusion

This paper presented a general method to modelize cameras introducing the use of cones to give a better approximation of the pixels' field of view (rather than the usual use of lines). We also introduced an experimental protocol to estimate cones that is not restricted to any geometry of cameras. The presented model used conformal geometric algebra that allowed to handle cones in a simple manner using twists. Geometric algebra allows natural and simplified representations of geometric entities without which the formulation of the problem would have been much more difficult. We are extending the use of cones to non central cameras that are variant scale and for which the use of lines is inadequate. Non central sensors introduce major improvements in the field of view of robots as they allow a better and a more adapted distribution of rays that eases the tasks to be performed.

References

1. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press (2003)
2. Debaecker, T., Benosman, R.: Bio-inspired model of visual information codification for localization: from retina to the lateral geniculate nucleus. Journal of Integrative Neuroscience **6** (2007) 1–33
3. Benosman, R., Kang, S.: Panoramic Vision: Sensors, Theory, Applications. Springer (2001)
4. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment – A modern synthesis. (2000) 298–375
5. Kim, J.H., Li, H., Hartley, R.: Motion estimation for multi-camera systems using global optimization. (2008)
6. Ke, Q., Kanade, T.: Quasiconvex optimization for robust geometric reconstruction. In: ICCV. (2005)
7. Kahl, F., Hartley, R.: Multiple view geometry under the l-infinity norm. In: PAMI. (2008)
8. Swaminathan, R., Grossberg, M., Nayar, S.: Caustics of catadioptric cameras. ICCV01 (2001) II: 2–9
9. Ieng, S., Benosman, R.: Geometric Construction of the Caustic Curves for Catadioptric Sensors. Kluwer Academic Publisher (2006)
10. Grossberg, M.D., Nayar, S.K.: A general imaging model and a method for finding its parameters. ICCV (2001) 108–115

11. Ramalingam, S., Sturm, P., Lodha, S.: Towards complete generic camera calibration. (2005) I: 1093–1098
12. Sommer, G., Rosenhahn, B., Perwass, C.: Twists - an operational representation of shape. In: IWMM GIAE. (2004) 278–297
13. Campbell, R., Flynn, P.: A survey of free-form object representation and recognition techniques. **81** (2001) 166–210
14. Hestenes, D.: The design of linear algebra and geometry. Acta Applicandae Mathematicae: An International Survey Journal on Applying Mathematics and Mathematical Applications **23** (1991) 65–93
15. D.Hestenes, Sobczyk., G.: Clifford Algebra to Geometric Calculus. D. Reidel Publ. Comp. (1984)
16. Rosenhahn, B.: Pose estimation revisited. PhD thesis, Christian-Albrechts-Universitat zu Kiel, Institut fur Informatik und Praktische Mathematik (2003)
17. Rosenhahn, B., Perwass, C., Sommer, G.: Free-form pose estimation by using twist representations. Algorithmica **38** (2003) 91–113
18. Pless, R.: Discrete and differential two-view constraints for general imaging systems. (2002) 53–59
19. C.Perwass: Applications of Geometric Algebra in Computer Vision. PhD thesis, Cambridge University (2000)
20. Barber, C.B., Dobkin, D.P., Huhdanpaa, H.: The quickhull algorithm for convex hulls. ACM Transactions on Mathematical Software **22** (1996) 469–483
21. Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 1330–1334