CENTER FOR
MACHINE PERCEPTION

CZECH TECHNICAL
UNIVERSITY

RESEARCH REPORT

ANNEX 2

Mirror Design for an Omnidirectional Camera with a Uniform Cylindrical Projection when Using the SVAVISCA Sensor

OMNIVIEWS – Omni-directional Visual System
FP5 RTD – FET Project No: IST–1999–29017
The technical description of the deliverable D2:
"Mirror implementation".

Stefan Gächter

stefan.gachter@ieee.org

№ CTU–CMP–2001–03

February 12, 2001

Available at
ftp://cmp.felk.cvut.cz/pub/cmp/articles/pajdla/Gaechter-TR-2001-03.pdf

Supervisor: Tomáš Pajdla

This research was supported by the EU Fifth Framework Programme, project OMNIVIEWS No. 1999–29017, by the Czech Ministry of Education under Research Programme J04/98:212300013, decision and control for industry, and by the Grant Agency of the Czech Republic under Project GACR 102/01/0971.

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Karlovo náměstí 13, 121 35 Prague 2, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

Contents

1. Introduction

This report describes the design of a specific omnidirectional camera. An omnidirectional camera is a combination between a conventional camera and an axially symmetric mirror. In the present case, the mirror projects to the conventional camera a cylinder by preserving its geometry. Therefore, a linear relation exists between the vertical position of a world point and the radial position of an image point. The design considers also the pixel density of the imager. One solution is presented for a uniform pixel density and an other for a non-uniform pixel density, especially for the log-polar density of the SVAVISCA sensor. For both solutions, the existence of a perspective projection is investigated.

The report is organized as follow, in Section 2 a general description of omnidirectional cameras is given. One approach for omnidirectional cameras, the catadioptric, is further investigated in Section 2.1. Different existing mirror shapes are presented in Section 2.1.1. The design in Section 3 treats first (Section 3.1) the problem of cylindrical projection without concerning the imager. Then, in Section 3.2 a solution for a uniform pixel density and in Section 3.3 a solution for a non-uniform pixel density is presented. Conclusions are given in Section 4.

2. Omnidirectional Camera

Cameras with a wide field of view are commonly called omnidirectional. The term omnidirectional is misleading. Strictly spoken, it means a field of view covering the whole space around the sensor but most omnidirectional cameras cover only a certain region. In Figure 1 there are depicted three possible field of views. Suppose that the camera is at the origin and points in the z direction and that the grey surface represents its field of view, when figure (a) depicts the small field of view of a conventional camera, and figure (b) and (c) depict the field of view as a hemisphere respectively as a panorama. The last two are the most commonly used for real omnidirectional cameras.

2.1. Catadioptric Sensor

There are roughly three different approaches used to obtain omnidirectional images; use of multiple standard cameras, use of a camera with special lenses, and use of a conventional camera along with a mirror[1]. The latter combination is also known as a catadioptric sensor[2]. Compared with the other two approaches, this one allows, by the law of geometric optics, a simple realization of demanded

1 *See* [24, 15] for a detailed survey.

2 *Dioptrics* is the science of refracting elements (lenses) whereas *catoptrics* is the optics of reflecting surfaces (mirrors). The combination of refracting and reflecting elements is therefore referred to as *catadioptrics* [11, 1].

field of view for competitive costs. This is mainly the reason for increased interest in research of catadioptric sensors and their applications[3]. In the following, the omnidirectional camera is based on the catadioptric approach; a combination of a conventional camera and an axially symmetric mirror where the optical axis of the camera and the symmetric axis of the mirror coincide. The omnidirectional image is then circular and must be unwrapped to obtain a panoramic image, as depicted in Figure 2.

2.1.1. Mirror Shapes

The conventional camera is supposed to be perspective and approximated by the pinhole camera model. The geometry of the pinhole camera is defined by the image plane, perpendicular to the optical axis, and the focal length. All points in the field of view are projected proportionally to the focal length to image points. Using the mirror to enlarge the field of view does change the projection characteristic and in general the pinhole model is not valid for omnidirectional cameras. A central perspective projection, however, is still valid for specific mirror shapes. Such mirrors are derived by Baker and Nayar in [1]. The shape of practical use is hyperboloidal[4]. Any image by such a camera is consistent with the way we are used to see images, which is desirable in surveillance and teleconferencing. Further, the central perspective projection is required by most techniques in machine vision.

The hyperboloidal shape is a specific solution of the family of polynomial mirror shapes derived by Chahl and Srinivasan in [4]. These mirrors do not provide a central perspective projection, except for the hyperbolic one, but still guarantee a linear mapping between the angle of elevation ϕ and the radial distance from the center of the image plane ρ, when referring to Figure 4. Another approach is to guarantee a uniform resolution for the panoramic image. The resolution in the omnidirectional image is increasing with growing eccentricity when using a camera with an imager of homogenous pixel density as it is visible in Figure 2.

The mirror shapes, that equalize the resolution, is derived by Conroy and Moore [5]. This family achieves solid angle pixel density invariance. A similar approach by Hicks and Bajcsy in [13] guarantees a uniform resolution in a plane that is parallel to the image plane. Further, this family preserves the geometry of the projected plane. They show in [12] that these mirrors approximate perspective projection. This is also achieved approximately with spherical mirrors as pointed out by Derrien and Konolige in [7].

3 *See* [6, 24, 9, 23, 16, 17, 10, 2, 3, 20] for more information about applications such as robot navigation, surveillance, model acquisition for virtual reality, and teleconferencing.

4 If instead of the perspective camera an orthographic is used, then the mirror shape is paraboloidal.

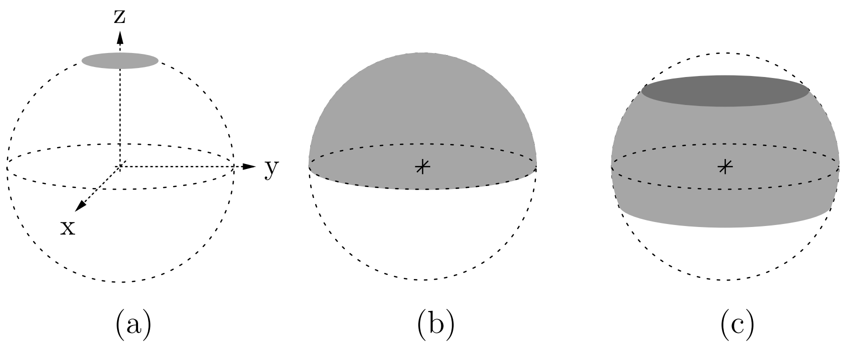


Figure 1: Different field of views. The camera is supposed to be at the origin and pointing into the z direction. The grey surface represents its field of view. Figure (a) is the small field of view of a conventional camera, figure (b) is a hemispheric field of view, and figure (c) is a panoramic field of view.

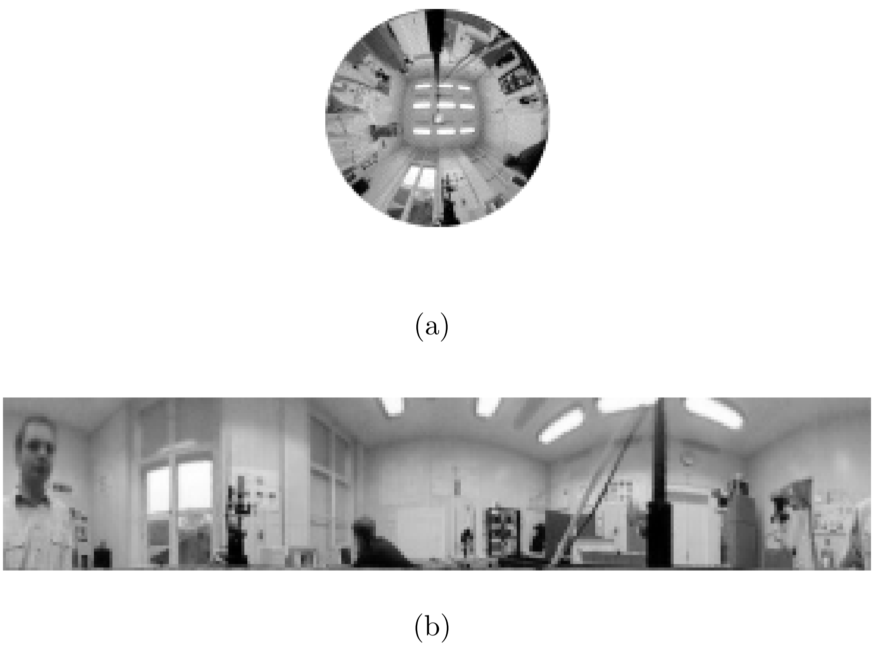


Figure 2: Images taken by an omnidirectional camera with an hyperboloidal mirror shape. Figure (a) depicts the omnidirectional image. Figure (b) depicts the corresponding panoramic image.

3. Design

In this project, the omnidirectional camera is used to detect and track a moving object. This is simplified, when the objects representation in the images rests equal. The aim of the system is, to detect and track moving objects, e.g. human faces, therefore it is suitable to have an omnidirectional camera providing images with a uniform representation of the object forefront. If only small objects are assumed, then the objects forefront lie approximately on a cylinder surface as depicted in Figure 3. Therefore the catadioptric sensor has to project the cylinder to preserve its geometry. Due to the axial symmetry, the problem is restricted to the ρz-plane and the mirror shape has to be designed such that at a given distance d from the origin, the vertical dimension h is linearly mapped to the radial distance from the center of the image plane ρ, when referring to Figure 4.

3.1. Derivation

The relation between a world point with coordinates $[d\ h]^T$, the cross section function $F(t)$ with $t(\rho)$, and an image point with coordinates $[\rho\ 0]^T$ is defined by the law of reflection, when referring to Figure 4 and Figure 5. Instead of deriving directly $F(t)$, first an expression for $h(t)$ is sought. This is not the most obvious way to find the cross section function but it results in some useful equations for the simulation of the sensor.

Referring to Figure 4, the world point at a distance d is given by the following relation

$$h(t) = F(t) - \cot(\phi)(d - t)\ . \tag{1}$$

This means that a ray emanating from an image point is reflected by the mirror and reaches a world point on the cylinder. The incident and coincident rays on the mirror as depicted in Figure 6 are described by their directional vectors $\vec{i}$ respectively $\vec{c}$, both of norm equal one. The law of reflection imposes that the angle between the incident ray and the normal to the surface is equal to the angle between the normal and the coincident ray. The normal vector $\vec{n}$ corresponds to a function of derivatives of the cross section function $F(t)$ at the point t. Expressed by the scalar product, the following condition must hold

$$-\vec{i} \cdot \vec{n} = \vec{n} \cdot \vec{c}\ . \tag{2}$$

The components of the vectors are as follows

$$\vec{i} = \frac{\vec{r}}{|\vec{r}|} = \begin{bmatrix} i_\rho \\ i_z \end{bmatrix}\ , \quad \vec{r} = \begin{bmatrix} t \\ F(t) - f \end{bmatrix}\ ,$$

$$\vec{n} = \begin{bmatrix} dF(t) \\ -dt \end{bmatrix}\ , \qquad \vec{c} = \begin{bmatrix} c_\rho \\ c_z \end{bmatrix}\ .$$

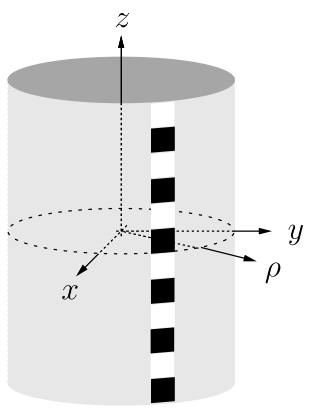


Figure 3: Cylinder surface for which an object forefront has to preserve its geometry.

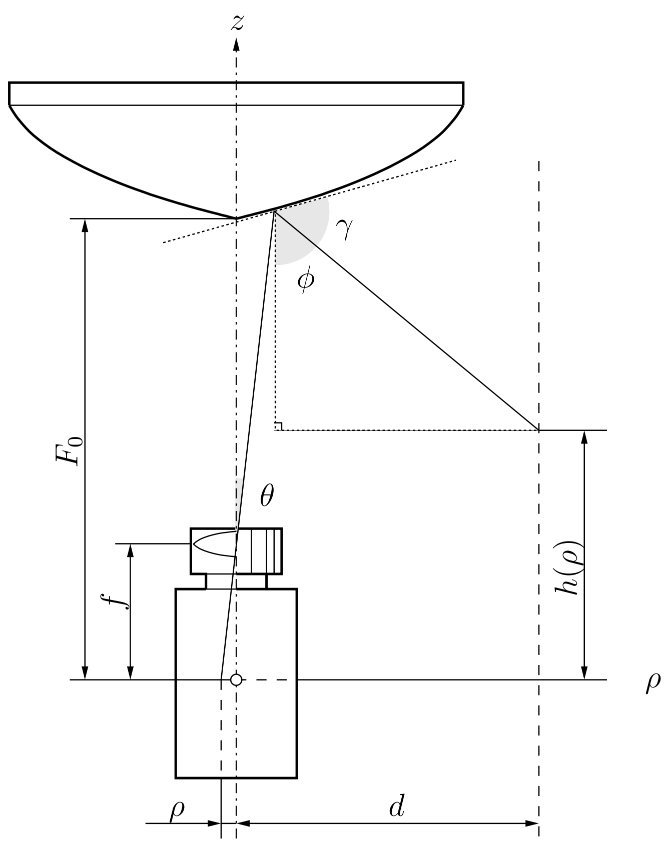


Figure 4: Schematic diagram of the catadioptric sensor.

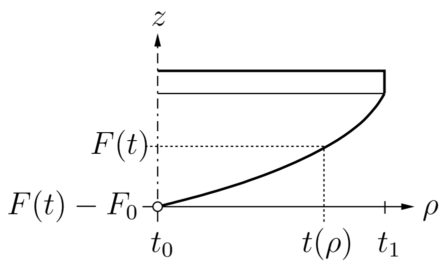


Figure 5: Cross section function.

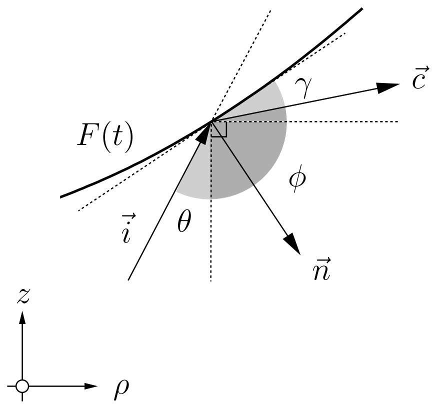


Figure 6: Reflection of a ray at the mirror surface.

Looking at (1), the term $\cot(\phi)$ has to be expressed as a function of the given geometry. Because $\tan(\phi)$ corresponds to the slope $\frac{c_z}{c_\rho}$ of the coincident ray[5] and because the slope of the incident ray can be expressed either by the vector $\vec{i}$ or $\vec{r}$ the following equation is obtained when solving (2).

$$\frac{c_z}{c_\rho} = F'(t) + \frac{i_\rho}{c_\rho}\left(F'(t) - \frac{F(t)-f}{t}\right) \tag{3}$$

The condition for two unity vectors, $\|\vec{i}\| = \|\vec{c}\| = 1$, gives the expression for the fraction $\frac{i_\rho}{c_\rho}$,

$$\left(\frac{i_\rho}{c_\rho}\right)^2 = \frac{1+\left(\frac{c_z}{c_\rho}\right)^2}{1+\left(\frac{F(t)-f}{t}\right)^2} \ . \tag{4}$$

Combining equation (3) and (4) results in a cubic expression

$$\begin{aligned}\left(\left(F'(t) - \frac{F(t)-f}{t}\right)^2 - \left(1+\left(\frac{F(t)-f}{t}\right)^2\right)\right)\left(\frac{c_z}{c_\rho}\right)^2 \\ +2F'(t)\left(1+\left(\frac{F(t)-f}{t}\right)^2\right)\frac{c_z}{c_\rho} \\ +\left(\left(F'(t)-\frac{F(t)-f}{t}\right)^2 - F'(t)^2\left(1+\left(\frac{F(t)-f}{t}\right)^2\right)\right) = 0 \ .\end{aligned} \tag{5}$$

One solution of this equation corresponds to the slope of the transmitted ray and the other to the slope of the reflected ray. Where the latter is the sought one.

$$\frac{c_z}{c_\rho} = \frac{F(t)-f}{t}$$

$$\frac{c_z}{c_\rho} = \frac{2tF'(t) - (F(t)-f)(1-F'(t)^2)}{2(F(t)-f)F'(t) + t(1-F'(t)^2)}$$

The expression for $h(t)$ can now be written as follows.

$$h(t) = F(t) + \frac{2tF'(t) - (F(t)-f)(1-F'(t)^2)}{2(F(t)-f)F'(t) + t(1-F'(t)^2)}(d-t) \tag{6}$$

Solving this equation for the derivative of the cross section function $F'(t)$ results in a cubic differential equation.

5 The law of reflection impose $2\gamma + \theta + \phi = \pi$ and therefore the slope of the coincident ray is given by $\frac{c_z}{c_\rho} = \cot(2\gamma + \theta) = -\cot(\phi)$.

$$F'(t)^2 + 2\frac{t(d-t) + (F(t)-f)(F(t)-h(t))}{(F(t)-f)(d-t) - t(F(t)-h(t))}F'(t) - 1 = 0$$

The differential equation, which defines the convex mirror shape, is consequently given by the following expression

$$\begin{aligned} F'(t) + \frac{t(d-t) + (F(t)-f)(F(t)-h(t))}{(F(t)-f)(d-t) - t(F(t)-h(t))} \\ -\sqrt{\left(\frac{t(d-t) + (F(t)-f)(F(t)-h(t))}{(F(t)-f)(d-t) - t(F(t)-h(t))}\right)^2 + 1} = 0 \ . \end{aligned} \tag{7}$$

Three parameters influence the mirror shape, the focal length f of the camera, the distance d, which corresponds to the perimeter of the projected cylinder, and the function $h(t)$, which corresponds to the vertical dimension of a world point. The parameter $h(t)$ defines also the characteristic of the catadioptric sensor. To have a linear relationship between the coordinate of a world point and the coordinate of an image point, the following condition must hold

$$h(\rho) = a\rho + b \ . \tag{8}$$

When substituting this expression in (7), the image coordinate ρ must be replaced by t. The relation between ρ and t results from the projection by a conventional camera.

$$\rho = \frac{ft}{F(t) - f} \tag{9}$$

A closed form solution for (7) seems not possible, when combined with (8) and (9). Therefore, the problem is solved numerically. The solution for the above differential equation is computed with the *MatLab* function `ode45.m`. This gets the best result compared with the other solvers of *MatLab* and that of *Mathematica*. The *MatLab* files for the numerical solution and the simulation are listed in Section B.

3.2. Solution for Uniform Pixel Density

What is the best mirror shape? As studied by Svoboda [21] for a hyperbolic mirror and a perspective camera, the mirror has to be designed such that a best combination between field of view and focal length is attained. The present case is different. A numeric solution for the cross section function is available and it is therefore not straight forward to find the optimal conditions between the different parameters. Nevertheless, the mirror shape has to guarantee a sufficient field of view and to be as flat as possible to avoid focusing problems, which are present for a real perspective camera and have not been considered in the design.

Four unknown parameters specify the mirror dimensions; the focal length f, the radius of the mirror rim t_1, the gain a, and the offset b of the function $h(\rho)$. The distance d is fixed and its influence is studied later. For the following simulation d is 2 m and f is 12.5 mm. To simplify the task, the radius of the mirror rim is chosen $t_1 = 3\,\text{cm}$ as used by Svoboda in [21, 22]. F_0 depends now on the focal length, the cross section function, the mirror rim, and the dimension of the imager. For the given camera[6], the maximal useful imager dimension is $\rho_1 = 2.4\,\text{mm}$. Because $F(t)$ is not known a priori, F_0 has to be estimated. F_0 is fixed in the present case to 15.5 cm. The remaining two parameters, the gain a and the offset b, define entirely the field of view. The minimal angle of elevation ϕ_0 is set by the offset b and the distance d. The maximal angle ϕ_1 is set by the gain a and the maximal imager dimension ρ_1. To have a large field of view, the minimal elevation angle must be as small as possible. Further the maximum elevation angle ϕ_1 should be as big as possible. Once b is fixed, the gain should be big enough to span the field of view over the whole considered vertical dimension. For a hemisphere, ϕ_0 is zero, but then b should be $-\infty$ and a consequently ∞. Hence technical realizable is only a panoramic field of view as depicted in Figure 1.

The cross section function for a mirror with a field of view of approximately $79.2°$ is depicted in Figure 7 and the distribution of the reflected rays in figure Figure 8. The technical drawing for the mirror is depicted in Figure 15. In figure Figure 9 is depicted the inverse projection of equidistant image points to the vertical dimension for different distances. It is not visible but the difference between the points is slightly varying. To have a feeling about that, the ratio between the numerical derivation of the function $h(\rho)$, $\frac{\Delta h}{\Delta \rho}$, given by (6) and the theoretical derivation of the function $h(\rho)$, $\frac{dh}{d\rho} = a$, given by (8) is computed. This ratio is normalized to the distance by dividing with the factor $\frac{d_i}{d}$, where $d_i \in \{1\,\text{m}, 1.5\,\text{m}, 2\,\text{m}, 3\,\text{m}, 4\,\text{m}\}$. The result is depicted in Figure 10. The gray surface is the hull of the result for the reference distance $d = 2\,\text{m}$. Because of the numeric derivation, the results are strongly oscillating, therefore only the averages are reproduced. The optimal ratio should be one and constant over whole range of ρ, this is even not the case for the reference distance. Thus, due to the numeric solution, a systematic error exists. The normalized ratios are somewhat linear and are decreasing with increasing ρ, hence the differences between the points are increasing in the top-to-bottom direction in Figure 9. The difference between the points is varying more for distances smaller than the reference, and less for distances greater than the reference. This is due to the fact, that for the first case a smaller and for the second case a bigger part of the cylinder is projected to the constant size of the imager. The slope of the graph is approximately inverse proportional to the distance and therefore the

6 *See* [19].

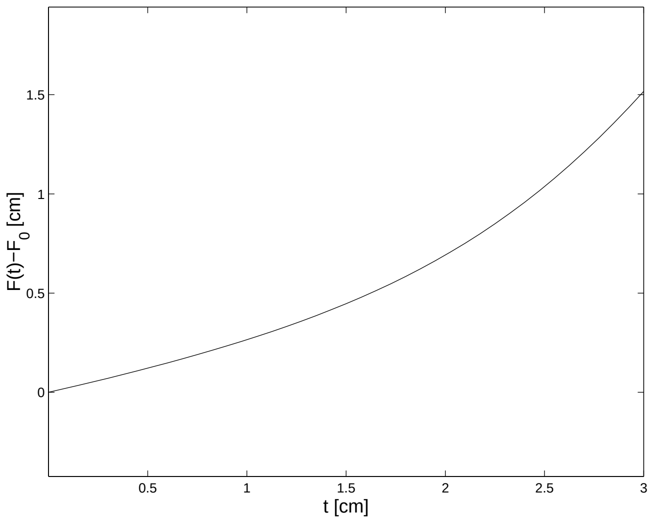


Figure 7: Cross section function for $a = 2000$ and $b = -400\,\mathrm{cm}$ results in a field of view of approximately 79.2° ($\phi_0 = 25.9°$ and $\phi_1 = 105.1°$).

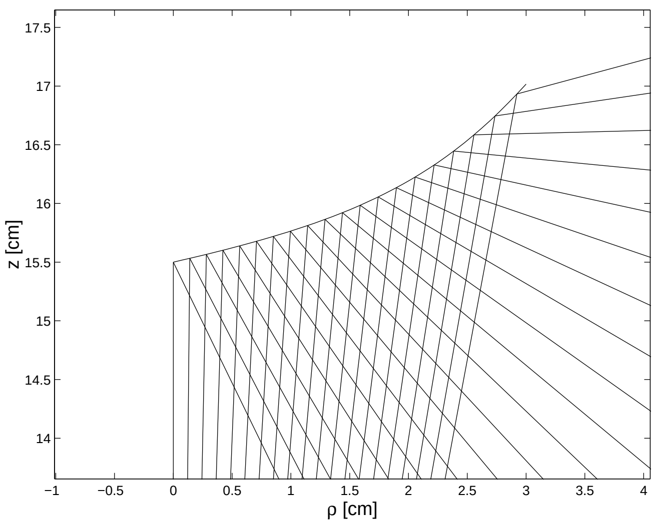


Figure 8: Rays reflected by the mirror which proportionally scales the vertical dimension.

projection is dependent on the distance.

3.3. Solution for Non-uniform Pixel Density (SVAVISCA)

For the precedent solution of the mirror shape, the pixel density of the imager is supposed to be uniform, otherwise there is no proportional relationship between the world and image dimension. When using a non-uniform pixel density, equation (8) must be modified. Using an imager with a log-polar pixel density[7], as described in [18, 8], the condition has to be of the following form

$$h(\rho) = a \log_k \left(\frac{\rho}{\rho_0} \right) + b \ , \tag{10}$$

when referring to Figure 11, where ρ is the radius, ρ_0 innermost circle of the log-polar layout and k the growth rate of the pixel size. The pixel size is linearly increasing from the foveal region towards the periphery. As before, a is the gain and b the offset. The foveal region of the SVAVISCA sensor has a uniform pixel density up to the radius $\rho_0 = 272.73\ \mu$m. For simplicity in the mirror design, this region is assumed to have a log-polar pixel density. The growth rate is $k = 1.02337$ for the discrete pixel density, when the condition is assumed to be the continuous approximation. Further specifications are given in [14].

The same parameter as before influence the mirror design; the focal length f, the radius of the mirror rim t_1, the gain a, and the offset b. For the following solution, d is 2 m and f is 25 mm. The mirror rim is $t_1 = 3$ cm, the maximal imager dimension ρ_1 is 3.6 mm and F_0 is fixed to 21.5 cm. The resulting cross section function is depicted in Figure 12 and the distribution of the reflected rays in Figure 13. The technical drawing for the mirror is depicted in Figure 16. The field of view is approximately 69.6° assuming that only log-polar part of the imager is used.

The mirror is designed for a fixed distance d. Therefore it is of interest how the mirror is behaving for objects outside the reference cylinder. As a criterion, the ratio between the numerical derivation of the function $h(\rho)$, $\frac{\Delta h}{\Delta \rho}$, given by (6), and the theoretical derivation of the function $h(\rho)$, $\frac{dh}{d\rho} = \frac{a}{\ln(k)} \frac{1}{\rho}$, given by (10) is computed. This ratio is normalized to the distance by dividing with the factor $\frac{d_i}{d}$, where $d_i \in \{1\text{ m}, 1.5\text{ m}, 2\text{ m}, 3\text{ m}, 4\text{ m}\}$. The result is depicted in Figure 14. The optimal ratio should be one and constant over the whole range of ρ. This is even not the case for the reference distance d. Thus, due to the numeric solution, a systematic error exists. The mapping between the world and image dimension is only approximately linear. The deviation of the ratios from the optimal ratio, which is equal one, is decreasing with increasing ρ, therefore the distortion in the mapping is increasing in the top-to-bottom

7 The advantage to use a log-polar pixel density is, that the pixel read-out results directly in the panoramic image and no additional computing for unwrapping is necessary.

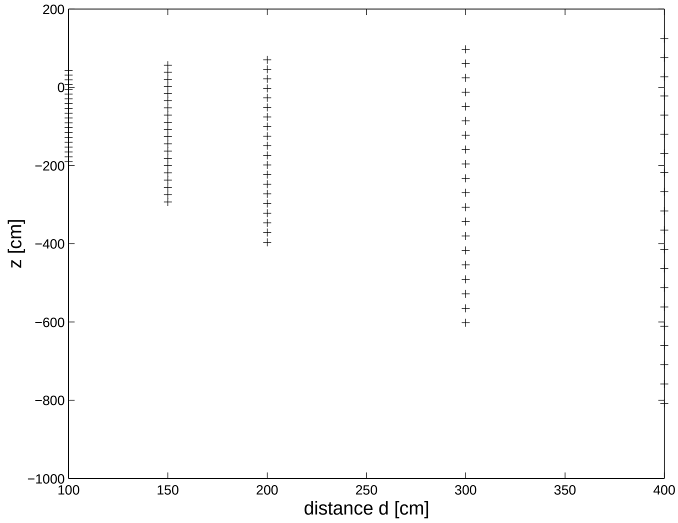


Figure 9: Inverse projection of equidistant image points to the vertical dimension for different distances. For each distance, the same mirror shape designed for a distance of $d = 200\,\text{cm}$ is used.

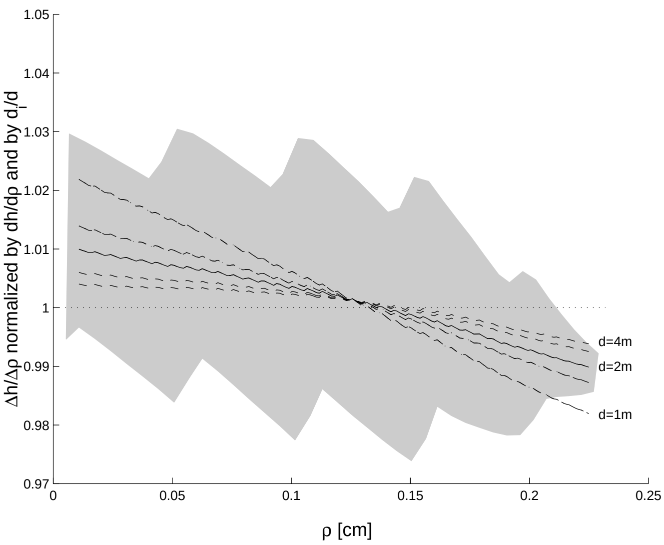
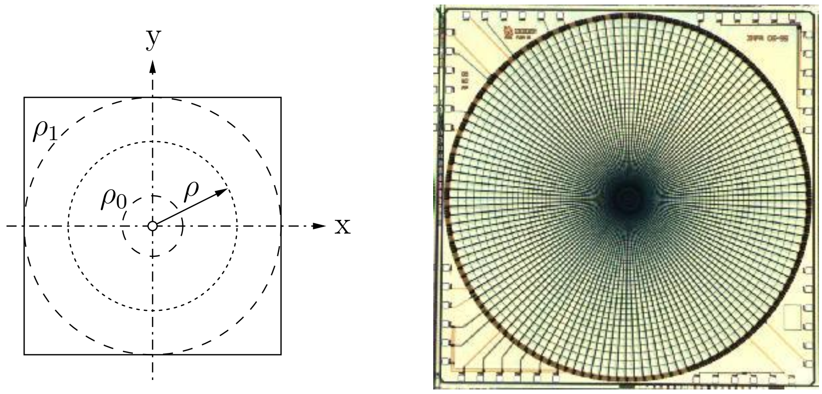


Figure 10: Ratio between numerical derivation of the function $h(\rho)$, $\frac{\Delta h}{\Delta \rho}$, and the theoretical derivation of the function $h(\rho)$, $\frac{dh}{d\rho} = a$. The ratio is normalized to the distance by dividing with the factor $\frac{d_i}{d}$, where $d_i \in \{1\text{ m}, 1.5\text{ m}, 2\text{ m}, 3\text{ m}, 4\text{ m}\}$. The gray surface is the hull of the result for the reference distance $d = 2\,\text{m}$. Due to the numeric derivation, the results are strongly oscillating, therefore only the averages are reproduced.



(a) (b)

Figure 11: Pixel density of the form log-polar for the camera imager. Figure (a) depicts the geometry and figure (b) an example imager similar to the SVAVISCA.

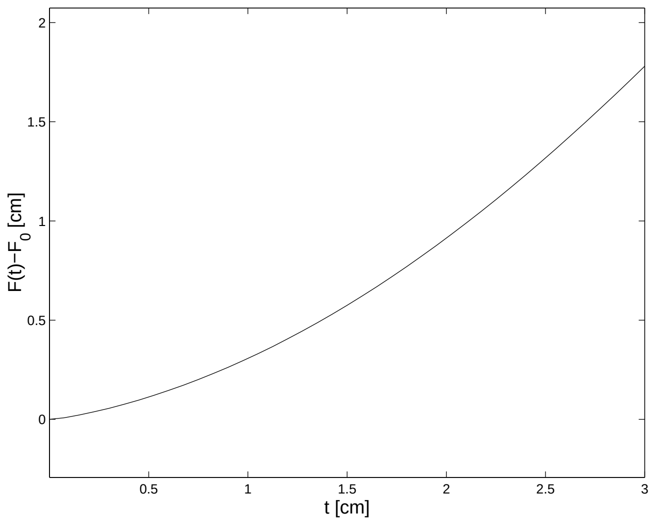


Figure 12: Cross section function for $a = 4$ and $b = -400$ cm results in a field of view of approximately 69.6° ($\phi_0 = 26.7°$ and $\phi_1 = 96.3°$).

direction. The ratios do not vary much with the distance. Therefore, the mirror has approximately a perspective projection.

4. Conclusion

This report has presented the design of two mirrors for a catadioptric sensor. These mirrors project a cylinder by considering the pixel density of the cameras imager. The geometry of the cylinder is only approximately preserved because of the limits of the numeric solutions. One mirror fits together with a conventional imager, the other with the SVAVISCA sensor. The design parameters for both mirrors yield in similar field of views, which are panoramic, and similar mirror dimensions. The mirrors are designed for a fixed cylinder radius. However as it is shown, the mirror for the SVAVISCA sensor is approximately independent of the radius.

The simulation with *MatLab* gave an idea about the sensor behaviour but only experimental results will show the usability of both mirror shapes. Further, only the influence of one parameter, the radius of the cylinder, has been investigated, it is therefore difficult to say, how critical the calibration of the sensor setup will be. Especially the calibration of the combination with SVAVISCA sensor could be difficult, because the mirror shape and the pixel density must match together to obtain a uniform projection.

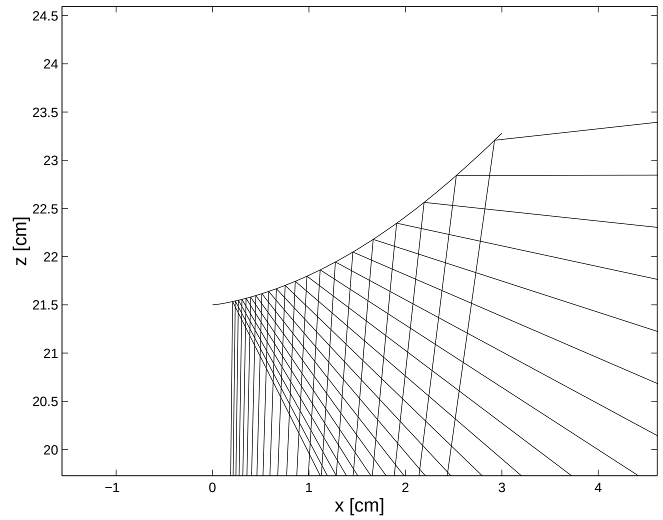


Figure 13: Rays reflected by the mirror which proportionally scales the vertical dimension. The imager pixel density is of the form log-polar.

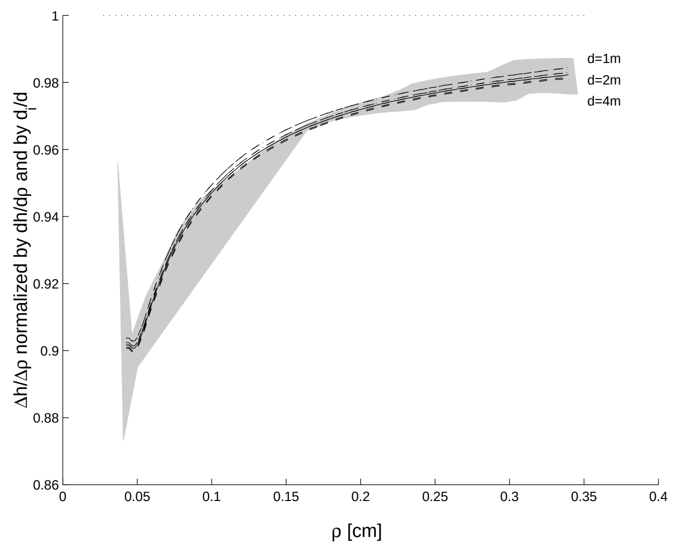


Figure 14: Ratio between numerical derivation of the function $h(\rho)$, $\frac{\Delta h}{\Delta \rho}$, and the theoretical derivation of the function $h(\rho)$, $\frac{dh}{d\rho} = \frac{a}{\ln(k)}\frac{1}{\rho}$. The ratio is normalized to the distance by dividing with the factor $\frac{d_i}{d}$, where $d_i \in \{1\,\mathrm{m}, 1.5\,\mathrm{m}, 2\,\mathrm{m}, 3\,\mathrm{m}, 4\,\mathrm{m}\}$. The gray surface is the hull of the result for the reference distance $d = 2$ m. Due to the numeric derivation, the results are strongly oscillating, therefore only the averages are reproduced.

References

[1] S. Baker and S.K. Nayar. A theory of catadioptric image formation. In *Proceedings of the International Conference on Computer Vision*, pages 35–42, January 1998.

[2] T. Boult, R. Micheals, X. Gao, P. Lewis, C. Power, W. Yin, and A. Erkan. Frame-rate omnidirectional surveillance and tracking of camouflaged and occluded targets. In *Second Workshop of Visual Surveillance at CVPR*, pages 48–58, 1999.

[3] T. Boult, C. Qian, W. Yin, A. Erkin, P. Lewis, C. Power, and R. Micheals. Application of omnidirectional imaging: Multi-body tracking and remote reality. In *Proceedings of the IEEE Workshop on Computer Vision Applications*, pages 242–3, October 1998.

[4] J. Chahl and M. Srinivasan. Reflective surfaces for panoramic imaging. *Applied Optics*, 36(31):8275–85, November 1997.

[5] T. Conroy and J. Moore. Resolution invariant surfaces for panoramic vision systems. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 1, pages 392–7, 1999.

[6] K. Danilidis. The page of omnidirectional vision. `http://www.cis.upenn.edu/~kostas/omni.html` (22 December 2000).

[7] S. Derrien and K. Konolige. Approximating a single viewpoint in panoramic imaging devices. In *Proceedings on the International Conference on Robotics and Automation*, pages 3931–8, April 2000.

[8] F. Ferrari, J. Nielsen, P. Questa, and G. Sandini. Space variant imaging. *Sensor Review*, 15(2):17–20, 1995.

[9] J. Gluckman and S. Nayar. Ego-motion and omnidirectional cameras. In *Proceedings of the International Conference on Computer Vision*, pages 999–1005, January 1998.

[10] D. Gutchess, A. Jain, and S. Chen. Automatic surveillance using omnidirectional and active camera. In *Proceedings of the Asian Conference on Computer Vision*, 2000.

[11] E. Hecht and A. Zajac. *Optics*. Addison-Wesley, 1974.

[12] A. Hicks and R. Bajcsy. Catadioptric sensor that approximate wide-angle perspective projections. In *IEEE Workshop on Omnidirectional Vision*, June 2000.

[13] A. Hicks and Bajcsy R. Reflective surfaces as computational sensors. In *CVPR-Workshop on Perception for Mobile Agents*, June 1999.

[14] A. Mannucci, P. Questa, and D. Scheffer. D1-document on specification, esprit project n. 31951 - svavisca, version 1.0. 1999.

[15] S. Nayar. Catadioptric omnidirectional camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 482–88. IEEE Computer Society Press, June 1997.

[16] K. Ng, H. Ishiguro, M. Trivedi, and T. Sogo. Monitoring dynamically changing environments by ubiquitous vision system. In *Proceedings of the Workshop on Visual Surveillance*, pages 67–73, June 1999.

[17] Y. Onoe, N. Yokoya, K. Yamazawa, and H. Takemura. Visual surveillance and monitoring system using an omnidirectional video camera. In *Proceedings of the International Conference on Pattern Recognition*, pages 588–93, 1998.

[18] F. Panerai, C. Capurro, and G. Sandini. Space variant vision for an active camera mount. *SPIE*, 2488, 1995.

[19] R. Sara. Settings of dt3152 for oscar and other ccir cameras. `http://cmp.felk.cvut.cz/cmp/hardware/oscar.html` (10 January 2001).

[20] R. Stiefelhagen, J. Yang, and A. Waibel. Modelling focus of attention for meeting indexing. In *Proceedings of ACM Multimedia'99*, pages 3–10, 1999.

[21] T. Svoboda. *Central Panoramic Cameras Design, Geometry, Egomotion.* PhD thesis, Czech Technical University, Centre for Machine Perception, 1999.

[22] T. Svoboda. Popis hyperbolického zrcadla, draft pro pracovníúčely. Technical report, Czech Technical University, Centre for Machine Perception, 1999.

[23] N. Winters and J. Santos-Victor. Omni-directional visual navigation. In *Proceedings of the 7^{th} International Symposium on Intelligent Robotic Systems*, pages 109–118, July 1999.

[24] Y. Yagi. Omnidirectional sensing and its applications. *IEICE Transactions on Information and Systems*, E82-D(3):568–79, March 1999.

A. Technical Drawings

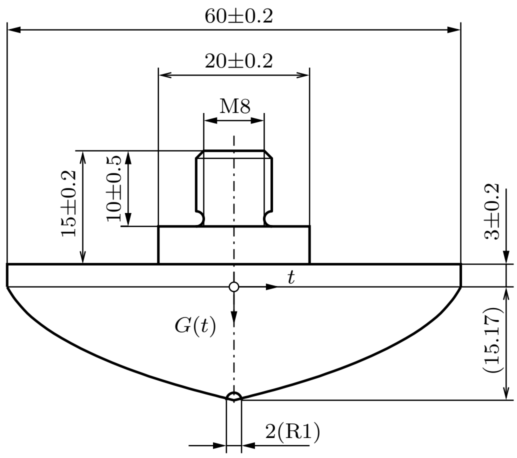


Figure 15: Technical drawing for the mirror matching with the uniform pixel density.

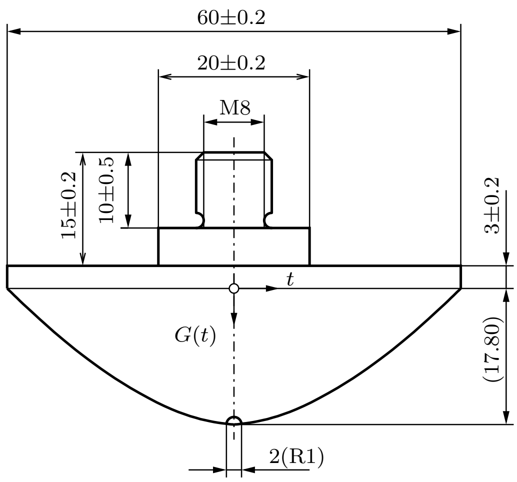


Figure 16: Technical drawing for the mirror matching with the non-uniform pixel density (SVAVISCA).

B. Files

B.1. Uniform Pixel Density

```
%mirror_uniform_pix_density.m  - Author: Stefan Gachter
%
%      batch file
%
%      See also: deq_csf_uniform_pixel_density.m, sqd_opt_fct_t_rel_r.m,
%              get_local_max.m, get_local_min.m

%      Author      : Stefan Gachter, stefan.gachter@epfl.ch
%                    00 Center for Machine Perception,
%                    Czech Technical University, Prague
%      Documentation: Gaechter-TR-2001-03.pdf
%      Language     : Matlab  5.3.1.29215a (R11.1), (c) MathWorks
%      Last change  : 13/01/2001
%      Status       : Ready
%

clear all;
close all;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% omnidirectional camera specifications

%% setup
d=200;      % cm  radius of the projected cylinder
           %     (distance)
F0=15.5;    % cm  distance image plane - mirror vertex
           %     (inital value of the mirror function at t0)

%% parameters function h(r)
a=2000;     %     gain
b=-400;     % cm  offset

%% camera
f=1.25;     % cm  focual length
%%% imager specification
r0=0;       % cm  minimal used dimension
r1=0.24;    % cm  maximal used dimension

%% mirror
t0=0.00001; % cm  minimal mirror radius
t1=3;       % cm  maximal mirror radius

% simulation specifications

N=20;       %     nb of rays for the visualization
h=0.001;    %     correctional factor for numerical computation
```

```
%% perspective projection investigation
Ndm=2;      %     nb of steps in direction to the mirror
sdm=50;     % cm  step size for distance variation
Ndp=2;      %     nb of steps in direction away from the mirror
sdp=100;    % cm  step size for distance variation

No=15;      %     order of smoothing filter

% MatLab parameters

gray=[0.8 0.8 0.8];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% cross section function

%% numerical solution
options=odeset('Refine',16,'RelTol',1e-12,'AbsTol',1e-24);
[t,F]=ode45('deq_csf_uniform_pixel_density',[t0 t1],F0,[],f,d,a,b);

figure(gcf+1);

plot(t,F-F0);

ylabel('F(t)-F_0 [cm]');
xlabel('t [cm]');
axis equal

dF=diff(F)./diff(t);
dt=t(1:length(dF));

% buffer
Fb=F;
tb=t;
dFb=dF;
dtb=dt;
clear F t dF dt;

% simulation

%% range for position of points on the mirror
%% (restricted to the corss section function)
t0a=(1+h)*dtb(1);
t1a=(1-h)*dtb(length(dtb));

%% range for position of image points
%% (restricted to the imager dimension)
r0a=f*t0a/(interp1(tb,Fb,t0a)-f);
if r0a<r0
  r0a=r0;
end
```

```
r1a=f*t1a/(interp1(tb,Fb,t1a)-f);
if r1a>r1
  r1a=r1;
end

%% equidistant image points
sr=(r1a-r0a)/(N-1);
R=[r0a:sr:r1a];

%% equidistant mirror points
%%% inital values for numerical solution
st=(t1a-t0a)/(N-1);
Ta=[t0a:st:t1a];
T=[];
%%% find mirror points
options=optimset('TolX',1e-12,'Display','off');
for i=1:length(R);
  t=fminsearch('sqd_opt_fct_t_rel_r',Ta(i),options,tb,Fb,f,R(i));
  T=[T t];
end
T=T(find(isfinite(T)));
R=R(find(isfinite(T)));

%% world points
F=interp1(tb,Fb,T);
dF=interp1(dtb,dFb,T);
%%% slope of coincident ray
C=(2.*T.*dF-(F-f).*(1-dF.^2))./(2.*(F-f).*dF+T.*(1-dF.^2));
%%% world points
H=F+C.*(d-T);

figure(gcf+1);
%%%% cross section function
plot(tb,Fb);
%%%% ray distribution
line([-R;T],[zeros(size(R));F],'Color',gray);
line([T;d*ones(size(T))],[F;H],'Color',gray);
%%%% world points
line([d*ones(size(T));d*ones(size(T))],[H;H],'Color','black','Marker','+');
axis equal
ylabel('z [cm]');
xlabel('\rho [cm]');

% buffer
Tb=T;
Cb=C;
clear F t dF R T H C;

% field of view
% (approximatly)

%% minimal elevation angle
```

```
if Cb(1)<0
  P0=-acot(Cb(1))*180/pi;
else
  P0=(pi-acot(Cb(1)))*180/pi;
end

%% maximal elevation angle
if Cb(length(Cb))<0
  P1=-acot(Cb(length(Cb)))*180/pi;
else
  P1=(pi-acot(Cb(length(Cb))))*180/pi;
end
fov=P1-P0;

disp_text=['field of view (approximately) : ' num2str(fov) ' degrees'];
disp(disp_text);

% perspective projection investigation

%% distances
D=[[d-Ndm*sdm:sdm:d] [d+sdp:sdp:d+Ndp*sdp]];
%% corresponding world points
F=interp1(tb,Fb,Tb);
H=[];
for i=1:length(D);
  H=[H;F+Cb.*(D(i)-Tb)];
end

figure(gcf+1);
plot(ones(length(H),1)*D,H','+');
xlabel('distance d [cm]');
ylabel('z [cm]');

clear F H;

%% continous image points
T=[Tb(1):st/10:Tb(length(Tb))];

F=interp1(tb,Fb,T);
dF=interp1(dtb,dFb,T);

%% slopes of coincident rays
C=(2.*T.*dF-(F-f).*(1-dF.^2))./(2.*(F-f).*dF+T.*(1-dF.^2));

%% world points
H=[];
for i=1:length(D);
  H=[H;F+C.*(D(i)-T)];
end

%% ratio between theoretical and numerical derivation
%%% numercial derivation
```

```
r=f.*T./(F-f);
R=ones(length(D),1)*r;
dH=diff(H,1,2)./diff(R,1,2);
%%% normailzation factor
%%% (theoretical derivation and distance)
Rn=a.*D'*ones(1,length(dH))./d;

Ra=dH./Rn;

%%% smoothing
B=fir1(No,0.01);
A=sum(B);
Ras=filter(B,A,Ra,[],2);
Ras=Ras(:,No+1:length(Ras));
Nc=ceil(No/2)+1;

figure(gcf+1);
%%% hull for result of reference distance
ind_max=get_local_max(Ra(Ndm+1,:),5);
ind_min=get_local_min(Ra(Ndm+1,:),5);
patch([R(Ndm+1,ind_min) fliplr(R(Ndm+1,ind_max))],...
        [Ra(Ndm+1,ind_min) fliplr(Ra(Ndm+1,ind_max))], ...
        gray,'EdgeColor',gray);
%%% ratios
line([R(1,1) R(Ndm+Ndp+1,length(Ra))],[1 1],[0 0],'Color','black', ...
      'LineStyle',':');
hold on
plot(R(1:Ndm,Nc:length(Ras)+Nc-1)',Ras(1:Ndm,:)','black-.', ...
      R(Ndm+1,Nc:length(Ras)+Nc-1)',Ras(Ndm+1,:)','black', ...
      R(1:Ndm+Ndp+1,Nc:length(Ras)+Nc-1)',Ras(1:Ndm+Ndp+1,:)','black--');
xlabel('\rho [cm]');
ylabel('\Deltah/\Delta\rho normalized by dh/d\rho and by d_i/d');
hold off

close(1);
```

```
function dF=deq_csf_uniform_pixel_density(t,F,flag,f,d,a,b);

r=f*t/(F-f);
h=a*r+b;
dF=-(t*(d-t)+(F-f)*(F-h))/((F-f)*(d-t)-t*(F-h))+. . .
    sqrt((t*(d-t)+(F-f)*(F-h))^2/((F-f)*(d-t)-t*(F-h))^2+1);
```

```
function y=sqd_opt_fct_t_rel_r(t,tb,Fb,f,r)

F=interp1(tb,Fb,t);
y=((F-f)*r-f*t)^2;

```

B.2. Non-Uniform Pixel Density

```
%mirror_non_uniform_pix_density.m  - Author: Stefan Gachter
%
%       batch file
%
%       See also: deq_csf_non_uniform_pixel_density.m, sqd_opt_fct_t_rel_r.m,
%                 get_local_max.m, get_local_min.m

%       Author       : Stefan Gachter, stefan.gachter@epfl.ch
%                      OO Center for Machine Perception,
%                      Czech Technical University, Prague
%       Documentation: Gaechter-TR-2001-03.pdf
%       Language     : Matlab  5.3.1.29215a (R11.1), (c) MathWorks
%       Last change  : 13/01/2001
%       Status       : Ready
%

clear all;
close all;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% omnidirectional camera specifications

%% setup
d=200;      % cm  radius of the projected cylinder
            %     (distance)
F0=21.5;    % cm  distance image plane - mirror vertex
            %     (inital value of the mirror function at t0)

%% parameters function h(r)
a=4;        %     gain
b=-400;     % cm  offset

%% camera
f=2.5;      % cm  focal length
%%% imager specification (SVAVISCA)
pw=0.00068;          % cm minimal pixel witdh
Na=252;              %    nb of cells per circle
r0=Na*pw/(2*pi);     % cm minimal used dimension
r1=0.356772;         % cm maximal used dimension
k=1.02337;           %    pixel growth rate (approximative k=(Na+2*pi)/Na)
```

```
%% mirror
t0=0.00001; % cm  minimal mirror radius
t1=3;       % cm  maximal mirror radius

% simulation specifications

N=20;       %     nb of rays for the visualization
h=0.001;    %     correctional factor for numerical computation

%% perspective projection investigation
Ndm=2;      %     nb of steps in direction to the mirror
sdm=50;     % cm  step size for distance variation
Ndp=2;      %     nb of steps in direction away from the mirror
sdp=100;    % cm  step size for distance variation

No=15;      %     order of smoothing filter

% MatLab parameters

gray=[0.8 0.8 0.8];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% cross section function

%% numerical solution
options=odeset('Refine',16,'RelTol',1e-12,'AbsTol',1e-24);
[t,F]=ode45('deq_csf_non_uniform_pixel_density',[t0 t1],F0,[],f,d,a,b,r0,k);

figure(gcf+1);

plot(t,F-F0);

ylabel('F(t)-F_0 [cm]');
xlabel('t [cm]');
axis equal

dF=diff(F)./diff(t);
dt=t(1:length(dF));

% buffer
Fb=F;
tb=t;
dFb=dF;
dtb=dt;
clear F t dF dt;

% simulation

%% range for position of points on the mirror
%% (restricted to the corss section function)
t0a=(1+h)*dtb(1);
```

```
t1a=(1-h)*dtb(length(dtb));

%% range for position of image points
%% (restricted to the imager dimension)
r0a=f*t0a/(interp1(tb,Fb,t0a)-f);
if r0a<r0
  r0a=r0;
end
r1a=f*t1a/(interp1(tb,Fb,t1a)-f);
if r1a>r1
  r1a=r1;
end

%% log distributed  image points
sr=(log(r1a)-log(r0a))/(log(k)*N);
R=r0*k.^[0:sr:N*sr];

%% equidistant mirror points
%%% inital values for numerical solution
st=(t1a-t0a)/N;
Ta=[t0a:st:t1a];
T=[];
%%% find mirror points
options=optimset('TolX',1e-12,'Display','off');
for i=1:length(R);
  t=fminsearch('sqd_opt_fct_t_rel_r',Ta(i),options,tb,Fb,f,R(i));
  T=[T t];
end
T=T(find(isfinite(T)));
R=R(find(isfinite(T)));

%% world points
F=interp1(tb,Fb,T);
dF=interp1(dtb,dFb,T);
%%% slope of coincident ray
C=(2.*T.*dF-(F-f).*(1-dF.^2))./(2.*(F-f).*dF+T.*(1-dF.^2));
%%% world points
H=F+C.*(d-T);

figure(gcf+1);
%%%% cross section function
plot(tb,Fb);
%%%% ray distribution
line([-R;T],[zeros(size(R));F],'Color',gray);
line([T;d*ones(size(T))],[F;H],'Color',gray);
%%%% world points
line([d*ones(size(T));d*ones(size(T))],[H;H],'Color','black','Marker','+');
axis equal
ylabel('z [cm]');
xlabel('\rho [cm]');

% buffer
```

```
Tb=T;
Cb=C;
clear F t dF R T H C;

% field of view
% (approximatly)

%% minimal elevation angle
if Cb(1)<0
  P0=-acot(Cb(1))*180/pi;
else
  P0=(pi-acot(Cb(1)))*180/pi;
end

%% maximal elevation angle
if Cb(length(Cb))<0
  P1=-acot(Cb(length(Cb)))*180/pi;
else
  P1=(pi-acot(Cb(length(Cb))))*180/pi;
end
fov=P1-P0;

disp_text=['field of view (approximately) : ' num2str(fov) ' degrees'];
disp(disp_text);

% perspective projection investigation

%% distances
D=[[d-Ndm*sdm:sdm:d] [d+sdp:sdp:d+Ndp*sdp]];
%% corresponding world points
F=interp1(tb,Fb,Tb);
H=[];
for i=1:length(D);
  H=[H;F+Cb.*(D(i)-Tb)];
end

figure(gcf+1);
plot(ones(length(H),1)*D,H','+');
xlabel('distance d [cm]');
ylabel('z [cm]');

clear F H;

%% continous image points
T=[Tb(1):st/10:Tb(length(Tb))];

F=interp1(tb,Fb,T);
dF=interp1(dtb,dFb,T);

%% slopes of coincident rays
C=(2.*T.*dF-(F-f).*(1-dF.^2))./(2.*(F-f).*dF+T.*(1-dF.^2));
```

```
%% world points
H=[];
for i=1:length(D);
  H=[H;F+C.*(D(i)-T)];
end

%% ratio between theoretical and numerical derivation
%%% numercial derivation
r=f.*T./(F-f);
R=ones(length(D),1)*r;
dH=diff(H,1,2)./diff(R,1,2);
%%% normailzation factor
%%% (theoretical derivation and distance)
Rn=a./log(k)./R(:,1:length(R)-1).*(D'*ones(1,length(dH))./d);

Ra=dH./Rn;

%%% smoothing
B=fir1(No,0.01);
A=sum(B);
Ras=filter(B,A,Ra,[],2);
Ras=Ras(:,No+1:length(Ras));
Nc=ceil(No/2)+1;

figure(gcf+1);
%%% hull for result of reference distance
ind_max=get_local_max(Ra(Ndm+1,:),5);
ind_min=get_local_min(Ra(Ndm+1,:),5);
patch([R(Ndm+1,ind_min) fliplr(R(Ndm+1,ind_max))],...
         [Ra(Ndm+1,ind_min) fliplr(Ra(Ndm+1,ind_max))], ...
         gray,'EdgeColor',gray);
%%% ratios
line([R(1,1) R(Ndm+Ndp+1,length(Ra))],[1 1],[0 0],'Color','black', ...
       'LineStyle',':');
hold on
plot(R(1:Ndm,Nc:length(Ras)+Nc-1)',Ras(1:Ndm,:)','black-.', ...
       R(Ndm+1,Nc:length(Ras)+Nc-1)',Ras(Ndm+1,:)','black', ...
       R(1:Ndm+Ndp+1,Nc:length(Ras)+Nc-1)',Ras(1:Ndm+Ndp+1,:)','black--');
xlabel('\rho [cm]');
ylabel('\Deltah/\Delta\rho normalized by dh/d\rho and by d_i/d');
hold off

close(1);
```

```
function dF=deq_csf_non_uniform_pixel_density(t,F,flag,f,d,a,b,r0,k);

r=f*t/(F-f);
h=a*(log(r)-log(r0))/log(k)+b;
```

```
dF=-(t*(d-t)+(F-f)*(F-h))/((F-f)*(d-t)-t*(F-h))+...
       sqrt((t*(d-t)+(F-f)*(F-h))^2/((F-f)*(d-t)-t*(F-h))^2+1);
```

```
function y=sqd_opt_fct_t_rel_r(t,tb,Fb,f,r)

F=interp1(tb,Fb,t);
y=((F-f)*r-f*t)^2;
```