Situational Awareness Using A Single Omnidirectional Camera

Master Project

of

Eugenio De Vito

Assistants : Dr. L. Jacques and PhD L. Bagnato
Director : Prof. Dr. P. Vandergheynst

Abstract

To retrieve scene informations using a single omnidirectional camera, we have based our work on a shape from texture method proposed by Lindeberg. To do so, we have adapted the method of Lindeberg, that was developed for planar images, in order to use it on the sphere $\mathbb{S}^2$. The mathematical tools we use are stereographic dilation to implement scale variations for the scale-space representation, and filter steerability on the sphere to decrease computational order. The texture distortions due to the projection from the real world to the image contain the informations that enable shape and orientation to be computed. A multi-scale texture descriptor, the windowed second moment matrix, that contains distorsions informations, is computed and analyzed, with some assumptions about the surface texture to retrieve surface orientation. We have used synthetic signals to evaluate the performances of the adapted method. The obtained results for the distance and shape estimations are good when the textures on the surfaces correspond to the assumptions, generally around the equatorial plane of $\mathbb{S}^2$, but when we move away from the equator, the precision of the estimations decreases significantly.

Contents

Introduction

The goal of the project is to implement a method that allows the retrieval of scene informations using a single omnidirectional camera. This problem comes down to a shape from texture problem.

Usually, the shape informations of a scene are obtained by stereo vision. The stereo vision is inspired by the human vision. It is based on the analysis of separated images, where features like edges are detected. The features from both images are then combined to retrieve 3-D informations.

The shape from texture is based on the analysis of one image only. The distortions of the textures present in the image enable the recovery of shape informations, because when the textures are projected from the real world onto an image, they are distorted, transformed. An image is not sufficient to determine surfaces shapes. Some *a priori* informations about the textures on the surfaces are also necessary.

The idea of applying shape from texture techniques to omnidirectional images is interesting, because an omnidirectional camera has a field of view that is much larger than a conventional camera. Domains where many cameras are needed to obtain a complete view of the situation can benefit from omnidirectional cameras, for example autonomous vehicles.

A technique developed by Lindeberg and Gårding [14] for planar images is adapted for omnidirectional images. To understand how we did, the first chapter introduces the mathematical tools we use. The second chapter describes the adaptation on the sphere of the mentioned technique. The third chapter shows the results of some experiments on omnidirectional images.

Chapter 1

Theoretical background

This first chapter is intended to define the theoretical notions needed to adapt on the sphere the work that Lindeberg has done on the plane. We first do a brief revue of what an omnidirectional camera is. Then some filtering notions on the sphere are necessary to implement a multi-scale analysis of an image on the sphere, who needs in its turn a certain dilation, and finally a geometrical and mathematical framework to handle the projection of the real world onto an image.

1.1 Omnidirectional cameras

The main purpose of omnidirectional cameras is to increase the field of view of planar cameras. Many different applications can benefit from wider fields of view, for example surveillance, tracking, teleconferencing, and so on. Omnidirectional sensors exist but they are still expensive. A cheaper way of building an omnidirectional camera is to combine a conventional camera with a mirror having a certain shape. We refer such combinations as catadioptric systems.
A constraint for the catadioptric system is that it must have a single effective viewpoint in order to obtain a perfect reconstruction and the camera must be an ideal perspective camera (see Figure 1.1). The effective viewpoint is the 3-D point where the catadioptric camera samples the plenoptic function and the ideal camera is the camera that can be modeled as the effective pinhole [2]. The plenoptic function describes the structure of the information in the light arriving on an observer, i.e. everything that can be seen.

The camera at our disposal is an orthographic camera with a parabolic mirror. This combination has a single effective viewpoint if the camera is ideal and satisfies the pinhole camera model and both, camera and mirror, are exactly aligned. Geyer and Daniilidis [10] have shown that this orthographic system is equivalent to a projection on the sphere with a subsequent stereographic projection. A nice property of the stereographic projection is that it is a conformal mapping, i.e. it preserves angles. The Figure 1.2 resumes these geometric properties of the catadioptric system. A point P is projected on a parabolic mirror on R, and R is orthographically projected on the image plane ℓ at point Q. If this point P is centrally projected on the sphere on R' and then R' is stereographically projected on the image plane ℓ on point Q', Q and Q' are the same point in the image plane ℓ.
The images coming from the camera are 2-D planar images. It is not possible to filter these images because they are in a sense "distorted". Since the stereographic projection is invertible,

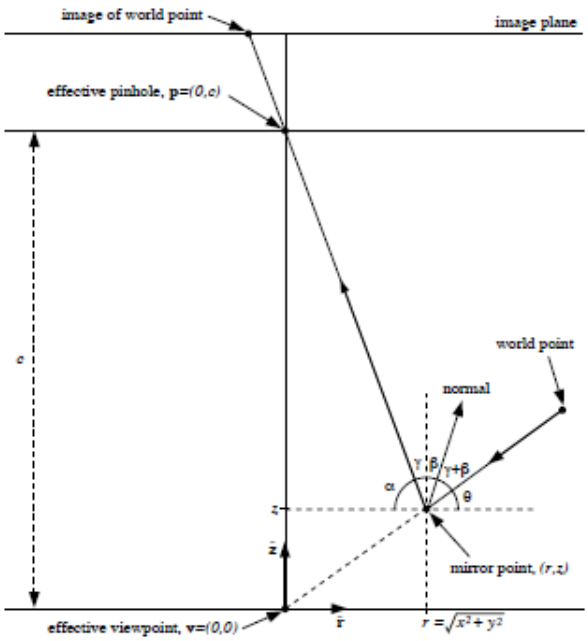


Figure 1.1: Effective viewpoint and pinhole. [2]

we first use the inverse stereographic projection to remap the image points from ℓ to the sphere and secondly we use the filtering on the sphere to process the remapped images.

1.2 Filters on the sphere

To filter a function on the plane, we compute the convolution between the function and the filter. The 2D convolution between two functions f and g on the plane is defined at point (x, y) by

$$h(x,y) = (f * g)(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x',y')g(x-x',y-y')dx'dy'.$$

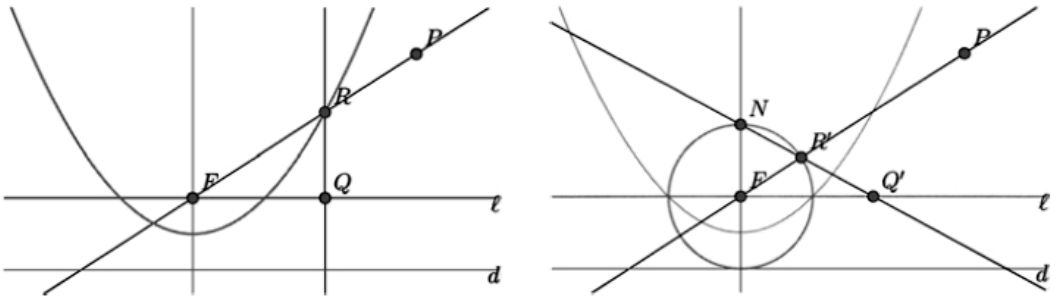


Figure 1.2: Left: orthographic projection. Right: Projection on the sphere and stereographic projection. [10]

This concept can be expanded to $\mathbb{S}^2$, but the correlation will be used instead of the convolution to filter a function.

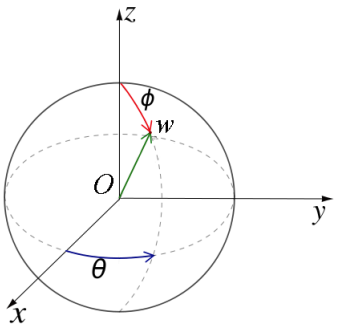


Figure 1.3: Spherical coordinate system

We define a Cartesian coordinate system (O, x, y, z) centered on the origin of the unit sphere (see Figure 1.3). A point ω has the spherical coordinates $\omega = (\theta, \varphi)$. If $f, g \in \mathbb{S}^2$, the correlation on $\mathbb{S}^2$, or more generally, the directional correlation (directional, because we can impose a direction to a function by a rotation χ around Oz) between f and g is defined as the scalar product between them, with f translated and eventually rotated to any point $\omega_0 = (\theta_0, \varphi_0)$

$$\langle Rf, g\rangle = \int_{\mathbb{S}^2} f^*(R^{-1}\omega)g(\omega)d\Omega,$$

where R is a 3-D rotation matrix and $d\Omega = \sin\theta d\theta d\varphi$ is the invariant measure on the sphere. R is used to displace and orient f on $\mathbb{S}^2$ with the help of three rotations: χ around Oz, θ_0 around Oy and φ_0 around Oz. (this is the Euler angles parametrization)
Like on the plane, filters can be isotropic (axisymmetric) or anisotropic (non-axisymmetric). A filter is isotropic when it is invariant under rotation on itself (rotation by $\chi \in [0, 2\pi[$); when not, the filter is said to be directional.
A filter f is steerable if any rotation by $\chi \in [0, 2\pi[$ of the filter around itself $R^{Oz}(\chi)f$ can be expressed by a linear combination of a finite number of basis filters f_m:

$$[R^{Oz}(\chi)f](\omega) = \sum_{m=1}^{M} k_m(\chi)f_m(\omega) \tag{1.1}$$

The weights $k_m(\chi)$ are called interpolation functions and the basis filters f_m can be specific rotations by χ_m of the original filter. [23] The steerability is interesting because it reduces the complexity order of the computation. It is not necessary to filter in any possible direction, but only a few directions must be computed. A linear combination of those directions gives the result of the filtering for any direction. [23] Something useful is that the steerability is still valid even after a stereographic projection because this projection is conformal.

[23] have shown that the inverse stereographic projection of radial functions (and their derivatives along the Ox direction) are steerable filters on $\mathbb{S}^2$. The filters we will use for the computation are first derivatives of a 2-D normalized Gaussian function in the tangent plane (the

mother wavelet, who is a radial function). The filters are steerable, because the derivative along the y axis is identical to the one along the x axis rotated by $\chi = \pi/2$:

$$g(x,y) = \frac{1}{2\pi} \quad e^{-\frac{x^2+y^2}{2}} \tag{1.2}$$

$$\partial_x g(x,y) = \frac{-x}{2\pi} \quad e^{-\frac{x^2+y^2}{2}} \tag{1.3}$$

$$\partial_y g(x,y) = \frac{-y}{2\pi} \quad e^{-\frac{x^2+y^2}{2}} \tag{1.4}$$

If a filter is a dilated version of a wavelet on the sphere, the directional correlation gives the same results as the wavelet analysis at the corresponding scale, then our analysis will be a wavelet analysis.

A fast computation method of the correlation on $\mathbb{S}^2$ is available via two toolboxes for Matlab®, the YAWTb [1] and SpharmonicKit[2]. It uses a scalar spherical harmonics transform (the spherical harmonics constitute an orthonormal basis on the sphere).

Remark. Even if we used second order derivatives, in a first time, to compute the Hessian matrix for the scale selection problem in section 2.4, we do not report the formulas of the second derivatives because we decide later to remain with the first order derivatives.

1.3 Stereographic dilation

An example is used to explain what is a stereographic dilation.
In $\mathbb{R}^3$, we define the 2-D sphere $\mathbb{S}^2$ (see Figure 1.4). We take a point

$$A = (\theta, \varphi) \in \mathbb{S}^2, \quad \theta \in [0, \pi], \quad \varphi \in [0, 2\pi[$$

and stereographically project it from the South Pole on the tangent plane at the North Pole and obtain the point B. If we apply a dilation by a factor a on the point B, we obtain the point B'. Now with the inverse stereographic projection, B' is projected on A' and the coordinates of A' are $A' = (\theta_a, \varphi)$ who are given by

$$\tan\frac{\theta_a}{2} = a \tan\frac{\theta}{2}.$$

All these steps define what is a stereographic dilation. We have stereographically dilated one point, but a function $f \in S^2$ can also be stereographically dilated. Indeed we can define a dilation by a factor a of a function f

$$[D_a f](\theta, \varphi) = \lambda(a, \theta)^{1/2} f(\theta_{1/a}, \varphi) \tag{1.5}$$

where $\lambda(a,\theta)^{1/2}$ is a normalization factor such that $\|D_a f\| = \|f\|$.[3]

Two important properties of the stereographic dilation are

1. it is a conformal projection, i.e. it preserves the angles,
2. it is radial, i.e. only the angle θ is transformed (φ remains unchanged).

If a function that has a certain orientation is stereographically dilated, its dilated version will still have the same orientation.

[1] YAWtb homepage : http://rhea.tele.ucl.ac.be/yawtb/
[2] SpharmonicKit homepage : http://www.cs.dartmouth.edu/~geelong/sphere/

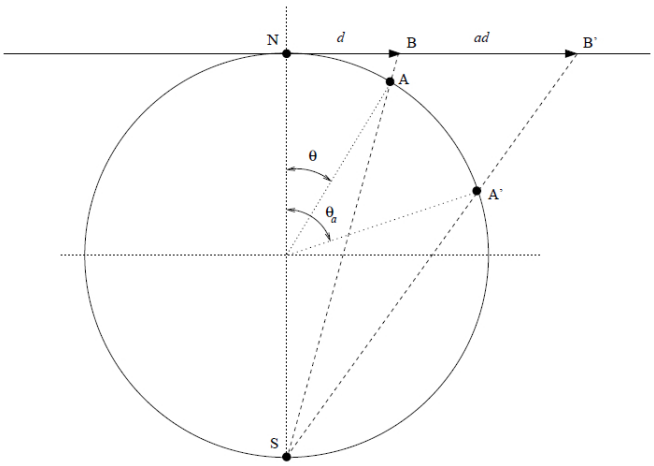


Figure 1.4: Stereographic dilation [3]

1.4 Image Geometry

Gårding [7] has defined a very useful mathematical framework for the estimation of the projection from the real world on an 2-D image. A summary of the important notions for a correct understanding follows.
When a surface texture from the real world is projected onto an image, the dimensions of the texture are changed in function of surface distance and surface orientation. These texture distortions are what allows the estimation of the shape and the orientation of the real surface.

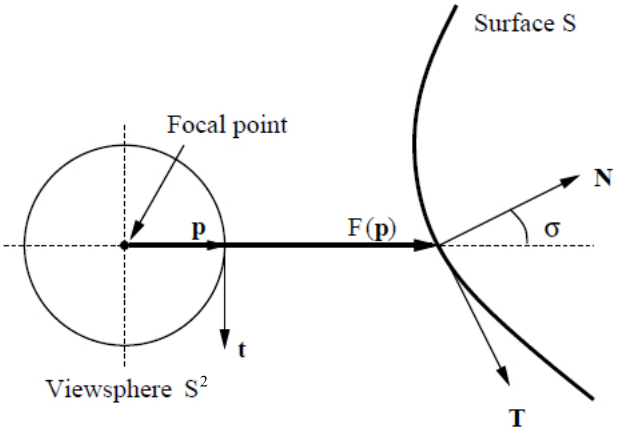


Figure 1.5: Basic geometry [7]

The Figure 1.5 shows the basic geometry used to define the mathematical framework. A smooth surface S is mapped by central projection onto a unit sphere $\mathbb{S}^2$ centered at the focal point. The backprojection map F from $\mathbb{S}^2$ to S is defined as

$$F(\vec{p}) = \vec{r}(\vec{p}) = r(\vec{p})\vec{p} \tag{1.6}$$

where $\vec{p}$ is a unit vector from the focal point to a point on the viewsphere $\mathbb{S}^2$ and $r(\vec{p})$ is

the distance along the visual ray from the focal point through $\vec{p}$ to the corresponding point $\vec{r} = F(\vec{p})$ on the surface S.
At any point $\vec{p}$ of the unit sphere $\mathbb{S}^2$, an orthonormal basis $(\vec{p}, \vec{t}, \vec{b})$ can be attached. $\vec{p}$ is collinear to the viewing direction, $\vec{t}$ gives the tilt direction (direction of the gradient of the distance from the focal point) and $\vec{b}$ is defined by $\vec{b} = \vec{p} \times \vec{t}$. With the perspective backprojection F we obtain the vectors $(\vec{B}, \vec{T})$ on S, that correspond to the vectors $(\vec{b}, \vec{t})$, and we retrieve also the vector $\vec{N} = \vec{T} \times \vec{B}$. $(\vec{N}, \vec{T}, \vec{B})$ is an orthonormal basis on S at the point $F(\vec{p})$.
Let $T_p(\mathbb{S}^2)$ be the tangent plane of $\mathbb{S}^2$ at point $\vec{p}$ and $T_{F(p)}(S)$ the tangent plane of S at the point $F(\vec{p})$. The derivative map F_* maps tangent vectors of $\mathbb{S}^2$ at point $\vec{p}$ to tangent vectors of S at point $F(\vec{p})$. F_* can be:

- a local linear approximation to F,
- an exact mapping from the tangent plane of $\mathbb{S}^2$ to the tangent plane of the surface S.

The backprojection map F_* can be expressed in the bases $(\vec{t}, \vec{b})$ and $(\vec{T}, \vec{B})$ by:

$$F_* = \begin{pmatrix} r/cos\sigma & 0 \\ 0 & r \end{pmatrix} = \begin{pmatrix} 1/m & 0 \\ 0 & 1/M \end{pmatrix} \tag{1.7}$$

σ is called the slant angle (angle between the viewing direction and the surface normal). m is the ratio of a projected length in the tilt direction in the image to the corresponding length in the surface and M is the analogous ratio computed in the direction perpendicular to the tilt. The local compression of length is inversely proportional to the distance in the unforeshortened direction and inversely proportional to the distance divided by the cosine of the slant in the foreshortened direction (the tilt direction). To resume, we have:

$$r = 1/M \tag{1.8}$$
$$cos\sigma = m/M \tag{1.9}$$

We see that F_* defines distance and slant completely, and can be imagined like an ellipse in $T_p(\mathbb{S}^2)$ that is the image of a unit circle in $T_{F(p)}(S)$ with m associated to the half-length of the minor axis in the tilt direction and M associated to the half-length of the major axis in the perpendicular direction. There is also a relation between areas in $T_p(\mathbb{S}^2)$ and $T_{F(p)}(S)$ that is given by m and M. A unit area element in $T_{F(p)}(S)$ has an area equal to mM in $T_p(\mathbb{S}^2)$. We define an area gradient $\nabla A = \xi_1 \xi_2 \nabla(mM)$ where ξ_1 and ξ_2 are unknown scale factors. These unknown factors can be eliminated if we consider normalized gradient, i.e. the rate of change of A relative to the value of A. We use these normalized gradients because we do not have a priori informations about the surface texture. The normalized area gradient is defined as:

$$\frac{\nabla A}{A} = \frac{\nabla m}{m} + \frac{\nabla M}{M} = -\tan\sigma \begin{pmatrix} 3 + r\kappa_t / \cos\sigma \\ r\tau \end{pmatrix} \tag{1.10}$$

where κ_t is the normal curvature of the surface in the tilt direction and τ the geodesic torsion of the surface in the tilt direction.

Chapter 2

Shape estimation method

Now that the basis have been explained, we can get to the heart of the matter. To retrieve informations about the shape of a scene (or an object) from the texture on its surface, we have adapted the technique developed by Lindeberg and Gårding for standard 2-D planar images in [14]. The basic idea of the technique is briefly summarized hereafter, but a more complete explanation will come later.

The scale-space theory, that tells how to manage size variations within images, is completed with a principle that enables the selection of the correct scale for further analysis. This principle is based on the research of local extrema over scales of normalized scale-invariant derivatives. These local extrema arise at scales that are related with the local size of the analyzed structure. Two different scales are necessary for the computation: a local scale, to compute the statistics of the image, and an integration scale, to accumulate these statistics. The differential entities in the scale-space are computed with Gaussian derivatives.

2.1 Scale-space

On the plane, a scale-space description of an image is a way of describing images from finer to coarser scales. The finer scales contain high-frequency information (details) and the coarser scales contain low-frequency information (smoothed data where details have been removed). This smoothing of details is called scale-space smoothing and it shall not introduce artifacts when going from a finer to a coarser scale. The scale-space produces a series of smoothed images that depend on a parameter t informing about the actual scale.

A scale-space representation L of a signal f can be obtained by convolving that signal with a Gaussian kernel g

$$L(\cdot;t) = g(\cdot;t) * f(\cdot). \tag{2.1}$$

The evolution of the scale-space L over scales is described by the diffusion equation

$$\partial_t L = \frac{1}{2}\nabla^2 L = \frac{1}{2}(\partial_{xx} + \partial_{yy})L. \tag{2.2}$$

Since convolution with the Gaussian kernel describes the solution of the diffusion equation, the Gaussian kernel is the unique kernel for generating a scale-space.

On the sphere $\mathbb{S}^2$, the scale-space representation L of a function is defined by its correlation with the Gaussian kernel. To compute the correlation, we define the Gaussian kernel g in the tangent plane at the North pole of the unit sphere by a normalized Gaussian

$$g(x, y; a) = \frac{1}{2\pi a^2} e^{-\frac{x^2+y^2}{2a^2}} \tag{2.3}$$

that is remapped by inverse stereographic projection on the unit sphere, and then we compute the correlation. The scale parameter of the scale-space representation on the sphere corresponds to the dilation a applied to the Gaussian. This dilation coefficient a informs about the actual scale (see Figure 2.1).

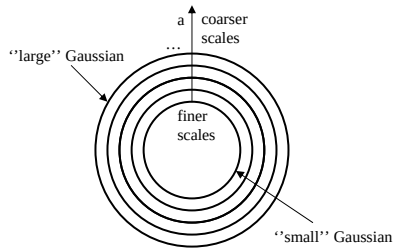


Figure 2.1: Scale-space vision on the sphere

Lindeberg uses a L^1 normalization to define its Gaussian kernel. Because of that, he introduced a normalization of his derivatives for the coherence of his analysis. We use a L^2 normalization for our derivatives, and we can prove that it is not necessary to normalize them.

Remark. In Equation 2.4, to obtain an averaging effect by the windowing, the window w is a Gaussian with a L^1 norm.

2.2 Texture descriptor

A texture descriptor is a mathematical entity that characterizes the texture informations contained in the images. For the shape-from-texture problem, the local linear distortions of the texture are what enables the recovery of shape informations. The texture descriptor that will be computed on the image has to reveal, to put in evidence these local linear distortions. The texture descriptor proposed by Lindeberg for this purpose is a second moment matrix of the scale-space representation of the image brightness. This descriptor on a point of $\mathbb{S}^2$ can be graphically represented by an ellipse painted on a planar surface (the ellipse in $T_p(\mathbb{S}^2)$ (cf. Chapter 1.4)). The descriptor is computed on each point of the image and the *windowed* second moment matrix is obtained by integrating over a surface the punctual descriptor. An explanation about the scales that must be used during the computation is available in the section 2.4.

In the plane, the texture descriptor used by Lindeberg is defined by

$$\mu_L(q) = \int_{\mathbb{R}^2} (\nabla L(x))(\nabla L(x))^T w(q - x) dx, \tag{2.4}$$

where q is the point of the plane where μ_L is computed, $L(x)$ is the image brightness and $\nabla L = (L_x, L_y)^T$ its gradient. μ_L is a 2×2 positive-definite matrix of rank 2 obtained by a convolution of the matrix $(\nabla L)(\nabla L)^T$ of rank 1 with a window w. The window w is a 2-D Gaussian function that weights the neighborhood of point q. Since the Gaussian function can generate a scale-space representation of a signal, μ_L will contain the scale-space representations of the components of $(\nabla L)(\nabla L)^T$. The punctual statistics given by $(\nabla L)(\nabla L)^T$ are integrated over a region weighted by w (we can compare it as an averaging). This means that on point q, μ_L gives a local, not punctual, information about the texture. Having a local information is more interesting than a punctual one because with the punctual behavior it is not really possible to retrieve a local shape information. So the windowing improves the computed statistics.
If we introduce E_q as the averaging operator describing the windowing operation, we can write

$$\mu_L(q) = \begin{pmatrix} \mu_{11} & \mu_{12} \\ \mu_{21} & \mu_{22} \end{pmatrix} = E_q \begin{pmatrix} L_x^2 & L_x L_y \\ L_x L_y & L_y^2 \end{pmatrix} = E_q((\nabla L)(\nabla L)^T) \tag{2.5}$$

For better understanding, we define the following quantities:

$$P = E_q(L_x^2 + L_y^2), \quad C = E_q(L_x^2 - L_y^2), \quad S = 2E_q(L_x L_y). \tag{2.6}$$

P is the average of the square of the gradient magnitude in a neighborhood of q. It's a measure of the strength of the operator. C and S contain directional informations that together give the anisotropy and the normalized anisotropy measures

$$Q = \sqrt{C^2 + S^2}, \quad \tilde{Q} = \frac{Q}{P}. \tag{2.7}$$

μ_L is coordinate independent, invariant to translations and rescaling can easily be applied by a uniform scaling factor. μ_L is not invariant to rotations but its eigenvalues

$$\lambda_{1,2} = \frac{1}{2}(P \pm Q) = \frac{1}{2}P(1 \pm \tilde{Q}) \tag{2.8}$$

are invariant to rotations and must be all positive since the matrix is positive-semidefinite.
Another property of μ_L is that if an invertible linear transformation B is applied to the image, and if a transformed image brightness R is defined by $L = RB$, then

$$\mu_L(q) = B^T \mu_R(p) B \tag{2.9}$$

where $\mu_R(p)$ is the second moment matrix of R at $p = Bq$.

On $\mathbb{S}^2$, μ_L is expressed differently. We do not compute the gradient of L. The entity that corresponds to L_x on the plane, is given by the correlation between f and the first derivative along the x direction of the Gaussian defined in the tangent plane at North pole. The other entity corresponding to L_y is the correlation between f and the derivative along the y direction of the same Gaussian. If we define Ψ_a as the invert stereographic projection of the first derivative along x of the Gaussian in the tangent plane, the two entities are expressed by

$$L_\theta = \langle R_0 \Psi_a, f \rangle = \int_{\mathbb{S}^2} \Psi_a^*(R_0^{-1}\omega) f(\omega) d\Omega \tag{2.10}$$

$$L_\varphi = \langle R_{\pi/2} \Psi_a, f \rangle = \int_{\mathbb{S}^2} \Psi_a^*(R_{\pi/2}^{-1}\omega) f(\omega) d\Omega \tag{2.11}$$

where $\omega = (\theta, \varphi)$ is a point on $\mathbb{S}^2$. The first entity responds on $\mathbb{S}^2$ in the θ direction and the second one in the φ direction, so we define $\vec{L} = (L_\theta, L_\varphi)^T$ as the entity on $\mathbb{S}^2$ corresponding to the gradient on the plane. We can redefine $\mu_L(\omega_0)$ on $\mathbb{S}^2$ by

$$\mu_L(\omega) = \begin{pmatrix} \mu_{11} & \mu_{12} \\ \mu_{21} & \mu_{22} \end{pmatrix} = E_\omega \begin{pmatrix} L_\theta^2 & L_\theta L_\varphi \\ L_\theta L_\varphi & L_\varphi^2 \end{pmatrix} = E_\omega(\vec{L}\vec{L}^T) \tag{2.12}$$

where the operator E_ω is also an averaging operator, who is defined by the correlation between the components of $(\vec{L}\vec{L}^T)$ and the inverse stereographic projection of the Gaussian in the tangent plane at the North pole (the window w). μ_L is then defined by

$$\mu_L(\omega_0) = \int_{\mathbb{S}^2} (\vec{L}\vec{L}^T) w(\omega_0 \cdot \omega) d\Omega \tag{2.13}$$

We suppose the property defined by the Equation (2.9) valid on $\mathbb{S}^2$, because it is also a linear transformation applied on μ_L who is defined on a tangent plane (where linearity is valid).

2.3 Shape cues

The important thing is that when we apply a linearized perspective mapping $F_{*\omega}^T$ to μ_S (the second moment matrix defined in the tangent plane to the surface), we have the following relation

$$\mu_L(\omega) = F_{*\omega}^T \mu_S(F(\omega)) F_{*\omega} \tag{2.14}$$

Then by combining estimates of μ_L (on $T_\omega(\mathbb{S}^2)$) and some assumptions about the texture (on $T_{F(\omega)}(S)$), this enables us to deduce the structure of $F_{*\omega}$.

2.3.1 Weak isotropy

The first assumption is the weak isotropy. It means that the second moment matrix defined in $T_{F(\omega)}(S)$ is proportional to the unit matrix, $\mu_S = cI$, where $c > 0$ is a constant and I the identity matrix. It follows that the normalized anisotropy equals zero, $(\tilde{Q} = 0)$, meaning that there is no dominant direction in the surface texture. The result of the weak isotropy assumption is that the eigenvalues of $F_{*\omega}$ are proportional to the square roots of the eigenvalues of the second moment matrix on $T_\omega(\mathbb{S}^2)$. The slant is then given by equation 1.9

$$\cos\sigma = \frac{m}{M} = \sqrt{\frac{\lambda_2}{\lambda_1}} = \sqrt{\frac{1-\tilde{Q}}{1+\tilde{Q}}} \tag{2.15}$$

where λ_2 is the smallest eigenvalue and $\lambda 1$ the biggest one.

2.3.2 Constant area

This assumption means that the local size of the surface texture elements does not vary systematically. In that case, the area measure is given by $A = \frac{1}{\sqrt{det\mu}}$ and we can use equation 1.10

$$\frac{\nabla A}{A} = \frac{\nabla m}{m} + \frac{\nabla M}{M} = -\tan\sigma \begin{pmatrix} 3 + r\kappa_t / \cos\sigma \\ r\tau \end{pmatrix} \tag{2.16}$$

This assumption is promising because when we look at a small area of a texture, locally the texture seems isotropic (looks identical within the local area), what means that the normalized area of the texture is constant. This is a more common behavior than the weak isotropy. During the computation, we did not use this last equation in the given form, but we used its magnitude.

2.4 Scales

The method we are implementing is based on a multi-scale analysis. The scale-space of L we obtain can be seen as a 3-D entity. For each scale, we have a layer containing the corresponding analysis. But this representation contains too many information. The selection of the information at the right scale is of utmost importance. It is what will determine the accuracy of the method. We will explain how to select the most appropriate scale to analyze an image area.

The scale selection begins by the differentiation between two scales, a local scale t and an integration scale s. The local scale t corresponds to the scale at which the second moment matrix μ_L is computed, i.e. the size of the wavelet. It determines the amount of smoothing applied on the image. The integration scale s corresponds to the size of the window w that determines the size of the neighborhood where the punctual descriptor is integrated. This scale must not be too large, otherwise the local linear approximation of the perspective mapping is no more valid. We will see that the integration scale is related to the local scale. The next two subsections explain exactly how the scales are chosen.

2.4.1 Local scale selection

The proposed method for the local scale selection is the following:

When a combination of derivatives assumes a local maximum over scales, we can say that the scale at which the maximum occurs corresponds to a characteristic dimension of the local structure in the image.

Lindeberg proposes to select the local scale t where the normalized anisotropy $\tilde{Q}$ assumes a local maximum over scales. In the case of images without noise, the local maximum of $\tilde{Q}$ shall occur at the same scale where $\det \mu_L$ achieves its own maximum over scales. $\det \mu_L$ gives a good estimation of the local dimension only if significant variations along both θ and φ directions occur within the texture. In section 3.1, a short comparison of three different scale selection methods ($\tilde{Q}$, $\det \mu_L$ and trace μ_L) is proposed. The results show that, even if the images do not contain noise, the $\tilde{Q}$ method is not as reliable as the others, anyway on our images. Furthermore to estimate the slant by the constant area assumption $\det \mu_L$ is necessary, and since the scale selection seems to work quit well with it, we decide to use it.

2.4.2 Integration scale selection

Like we already mentioned in section 2.2, the window function used for weighting the neighborhood is a 2-D Gaussian function. Lindeberg proposes to use a slightly larger scale than the one indicated by the local maximum of $\det \mu_L$, who already gives an information about

the local dimension, more precisely the characteristic size of the 2-D pattern (the area). Multiplying this characteristic size by a constant equal to 2 or 3 in order to have an integration region big enough but not too big, will also ensure that the local linear approximation will still be valid.

Remark. Like mentioned on the other *Remark* on page 5, we used to compute the Hessian matrix. Lindeberg proposes this step as a pre-processing. We should search for the local maximum of $\det H_L$ over scales and space to find a set of "characteristic points" where to estimate the slant σ. We dropped this step because of the implementation we did (computation in the transformed domain).

2.5 Computation

The implementation of the above mentioned method is done in the following way:

1. define the limits of the local scale $t = [t_{min}, t_{max}]$,
2. define the multiplicative constant c applied to the local scale to obtain the integration scale, i.e $s = ct$,
3. compute μ_L for every local scale t and integration scale $s = ct$,
4. search for the local maximum over scales of $\det \mu_L$ to define the different scales within the image,
5. estimate shape cues (distance r and slant σ) at the selected scales within the image.

Chapter 3

Experimentation

After implementing the method in Matlab®, we have used five signals to test it: one synthetic signal, and four more interesting signals representing the projections on the sphere of a black and white chessboard painted on the inner surface of a cylinder and a cube. The omnidirectional camera was placed once in the center of the scenes, and once slightly displaced. The four images were rendered with Blender[1]. The synthetic signal is defined by the following equation:

$$f_1(\theta, \varphi) = \sin(\nu_0 \varphi) \sin(\nu_0 \cot \theta) \tag{3.1}$$

where ν_0 is the signal frequency. $\nu_0 \in \mathbb{N}^*$ to avoid any discontinuity of the signal on the sphere, i.e. have an integer number of periods. The synthetic signal f_1 was generated to test the weakly isotropic assumption, i.e. to have a texture that was rotationally invariant. f_1 can be compared to the centered cylinder case, but the cylinder would have an infinite height. The signal is shown in the Figure 3.1 for $\nu_0 = 10$.

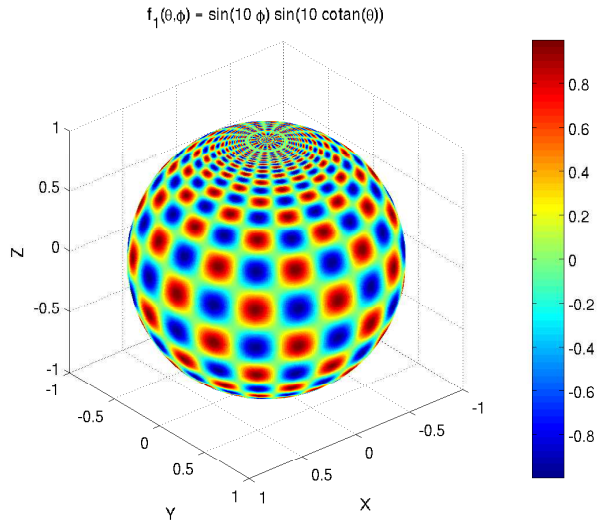


Figure 3.1: Synthetic signal

[1]Blender homepage : http://www.blender.org/

3.1 Scale selection

The first test we have done was to analyze the scale selection because it is an important step within the shape estimation method. The selection of the correct scale ensures that the computation is done at a scale that is related to the local size of the texture. To test the proposed scale selection methods, we have used the scene of the displaced cylinder (see Figures 3.2 and 3.3).

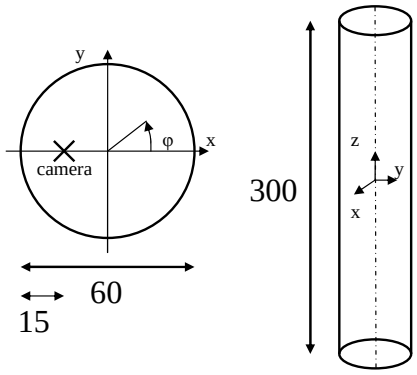
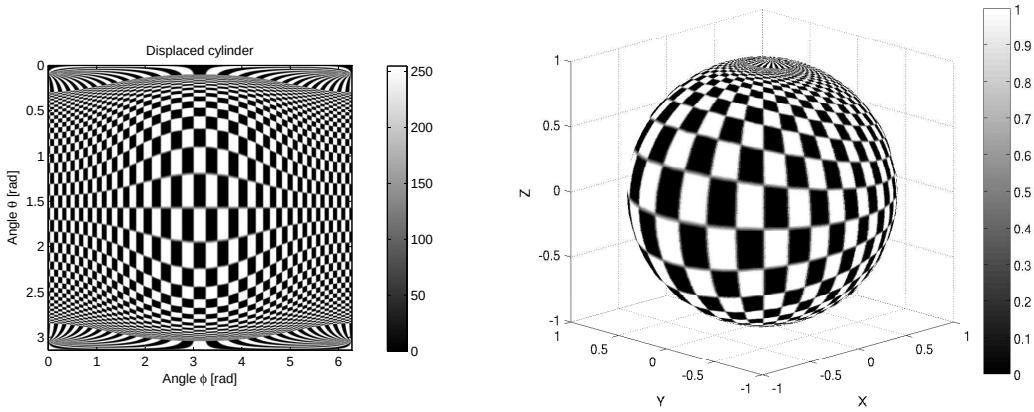


Figure 3.2: Schematic of the displaced cylinder scene.



(a) 2-D view

(b) 3-D view

Figure 3.3: Projected texture of the displaced cylinder on the unit sphere

We have compared the results of the scale selection for three differential entities: $\tilde{Q}, \det \mu_L$ and traceμ_L. We added traceμ_L because for unidirectional patterns, when $\tilde{Q} = 1$ or $\lambda_2 = 0$, $\det \mu_L = 0$ but trace$\mu_L \neq 0$ and so we could still have an information about scale. We have taken 6 points of the image that correspond to different sizes of the local texture and we have analyzed the behavior of each differential entity at each point. Two sets of points have been used and the results are available in the Figure 3.4. The values obtained for $\det \mu_L$ and

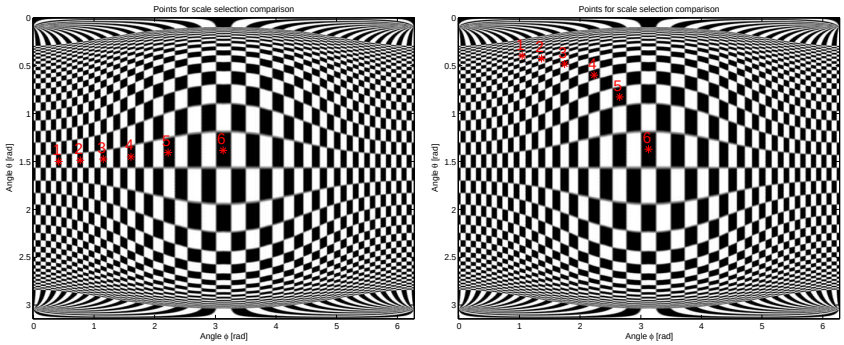
traceμ_L have been normalized to facilitate the comparison. Since $\tilde{Q} \in [0, 1]$ it is not necessary to normalize this entity. Like we can see, the use of the determinant or the trace for the scale selection is arbitrary, because for the two sets of points the results are identical between these two differential entities. The selected scales seem to behave exactly like expected, increasing texture pattern size corresponding to an increase of the scale. The behavior of $\tilde{Q}$, that should be the best suited differential entity for the scale selection according to Lindeberg, selects a different scale compared to the other two methods. When $\tilde{Q} \approx 1$, μ_L is of rank 1. This proves the role of the integration scale, i.e. by integrating μ_L over an area the rank can be equal to 2. (when $\tilde{Q} \approx 0, \mu_L \propto$ the identity matrix) Since the computation of the determinant is necessary for further analysis and not the trace, we decided to keep the determinant method.

3.2 Distance estimation

According to the equations (1.8) to (1.10), the easiest information to retrieve seems to be the distance $r = 1/M$ that depends only of the minimal eigenvalue of μ_L. To verify if this is the case, we decided to begin with a simple signal. For this purpose, we generated the signal f_1, who represents a blob like structure that is supposed to be invariant to rotations in a neighborhood of the equator, i.e. for $\theta = \pi/2$. The equatorial distance is the first distance to be estimated.
We analyzed the behavior of the distance estimation by varying the value of the frequency $\nu_0 = 10, 20, \ldots, 90, 100$ because the distance is a function of the frequency. The normal behavior would be that the increase of ν_0 should correspond to the increase of the distance. The explanation is simple and can be illustrated with an example: imagine a wall with vertical and equally spaced stripes painted on it. The stripes represent a periodic signal having a certain spatial frequency. If we look at the wall from five meters away, and then from fifty meters away, we easily guess that the spatial frequency will be different in the two cases. At fifty meters, the stripes will look nearer of each other and so will generate a higher spatial frequency. With this simple example, we prove that a high spatial frequency corresponds to a big distance and that a low spatial frequency corresponds to a small distance.We took for each value of ν_0 the mean value of the estimated distance on the equator, and plot the behavior of the distance in function of the frequency. For the computation, we have used the following parameters:

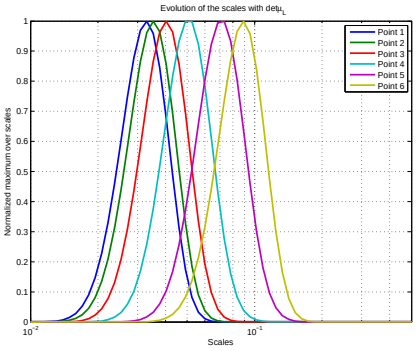
- local scales $t \in [0.01, 0.5]$ (logarithmic scale with 60 values)
- integration scales $s = 2t$ and $s = 3t$ (when not mentioned, the integration scale is $s = 2t$)
- the spherical grid has 512 points for the θ et φ directions

The result of this first experience is shown in the Figure 3.5(a). As we can see, it seems to be a slight problem! The evolution of the distance in function of the frequency is completely inverted compared to what is expected: the theory tells that closer surfaces would mean lower frequencies, and distant surfaces would mean higher frequencies, and here we obtain exactly the contrary. We have not managed to find the origin of this behavior, neither in the theory, nor within the Matlab® implementation.
So we decided to invert the obtained distance and see what would happen. The result is shown in Figure 3.5(b). This time, we obtain straight lines, what means that the estimation is perfect. The distance increases linearly with the frequency (distance). On the same Figure, we

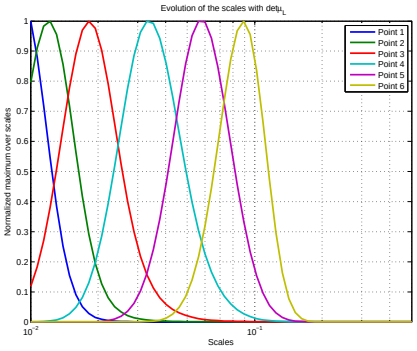


(a) 1st set

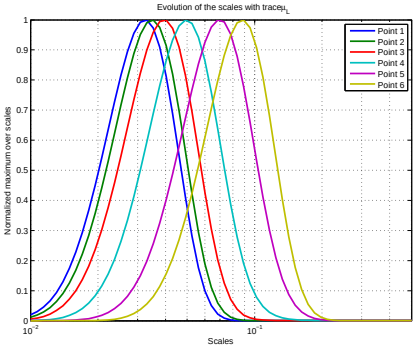
(b) 2nd set



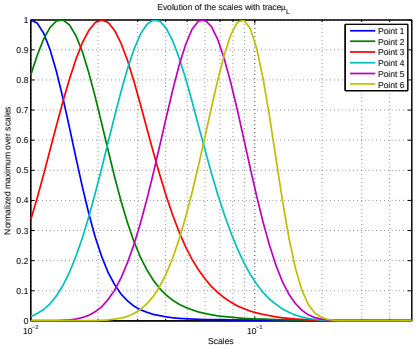
(c) $\det \mu_L$ for 1st set



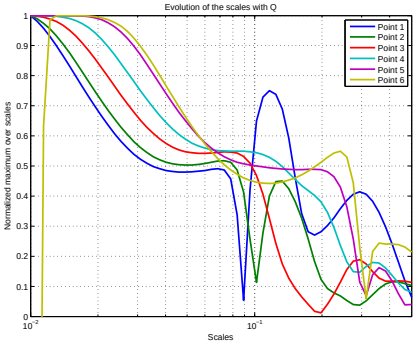
(d) $\det \mu_L$ for 2nd set



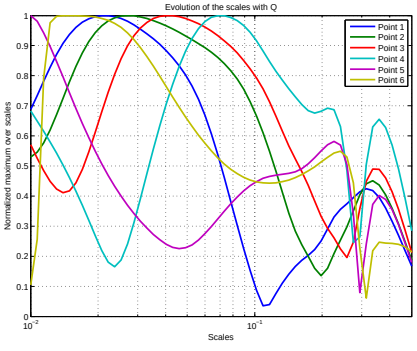
(e) traceμ_L for 1st set



(f) traceμ_L for 2nd set

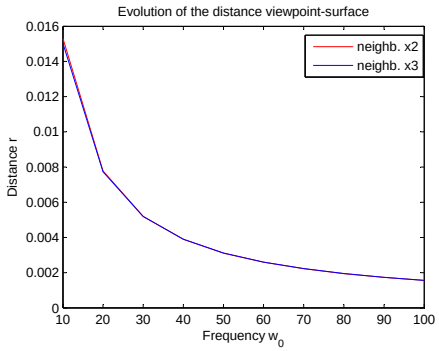


(g) $\tilde{Q}$ for 1st set

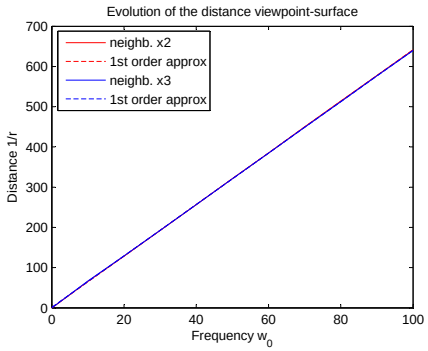


(h) $\tilde{Q}$ for 2nd set

Figure 3.4: Comparison of the scale selection between $\det \mu_L$, traceμ_L and $\tilde{Q}$.



(a) Distance r.



(b) Distance $1/r$

Figure 3.5: Evolution of the distance r and $1/r$ in function of the frequency ν_0.

have plotted the first order approximations for the two neighborhoods and they fit perfectly the experimental curves. We have also added on the same figure the zero frequency that would mean no distance between the viewing point and the surface.

A conclusion for this first experiment is that the distance estimation on the equator is very good apart the fact that we must take the inverse of what the theory tells us! The behavior of the distance estimation for the other values of the θ angle has been tested on the cube and the cylinder scenes. The results are available in the respective sections.

3.3 Slant estimation

For this second experiment, we tried to verify the slant estimation using the signal f_1. It corresponds to the case where the omnidirectional camera is placed in the center of a infinite cylinder. The situation is shown in Figure 3.6. The frequency of the texture (on the surface,

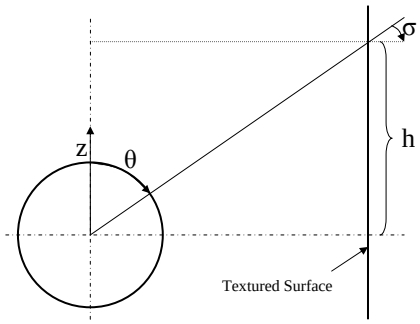


Figure 3.6: Situation for the signal f_1.

not its projection on the unit sphere) shall be constant along the Oz direction to verify the constant area assumption. To find the equation of f_1, we had to express the height z in terms

of θ, $z = h = \cot\theta$. In that way, the frequency of the texture could be constant along Oz. We varied the frequency ν_0 to see how the frequency change would influence the slant estimation. Something to keep in mind is that the values of ν_0 shall not be too small neither too big, because the maximum aloud frequency ν_0 is limited by the sampling frequency on the sphere, i.e. the resolution of the spherical grid. The computed values of the slant have sense only within different intervals that are function of ν_0 and N, the number of points of the spherical grid in direction θ. We use the Figure 3.6 to find the limits of the intervals.

$$f_1(\theta, \varphi) = \sin(\nu_0 \varphi) \sin(\nu_0 \cot\theta)$$

is the signal for which we would like to find the minimum angle $\theta \in [0, \pi/2]$ on the spherical grid under which the sampling frequency on the grid is no more sufficient. We can use the complex notation to get the argument and the frequency of the θ component of f_1

$$e^{\imath\nu_0 \cot\theta}, \qquad w(\theta) = \nu_0 \cot\theta, \qquad f(\theta) = \frac{\nu_0}{2\pi} \cot\theta.$$

The sampling period and the sampling frequency along θ are equal to

$$T_\theta = \frac{\pi}{N}, \qquad F_\theta = \frac{1}{T_\theta} = \frac{N}{\pi}.$$

The sampling theorem tells us that the sampling frequency has to be at least two times higher than the maximum frequency present in the signal, but in practice it should be ten to twenty times higher (we will take 20)

$$f(\theta) \leq \frac{1}{20} F_\theta = \frac{N}{20\pi}.$$

Now if we insert the expression of $f(\theta)$, we get to the solution

$$\frac{\nu_0}{2\pi} \cot\theta \leq \frac{N}{20\pi}, \quad \cot\theta \leq \frac{2N}{20\nu_0}, \quad \tan(\frac{\pi}{2} - \theta) \leq \frac{2N}{20\nu_0}$$

$$\theta \geq \frac{\pi}{2} - \arctan\frac{2N}{20\nu_0}, \qquad \theta \in [0, \frac{\pi}{2}]$$

For example, if the spherical grid has $N = 512$ points along θ and the frequency of the signal $\nu_0 = 40$, we obtain $\theta \geq 0.66[rad] = 38[deg]$. To know exactly the confidence interval is important in order to compare the various results obtained for different values of ν_0. A practical example, where the confidence interval limits are displayed by vertical lines, can be seen in the Figure 3.7.

For computing the slant angle σ, we have two formulas at our disposal, the first comes from the weakly isotropic assumption (Equation 2.15)

$$\cos\sigma = \frac{m}{M} = \sqrt{\frac{\lambda_2}{\lambda_1}} = \sqrt{\frac{1 - \tilde{Q}}{1 + \tilde{Q}}}$$

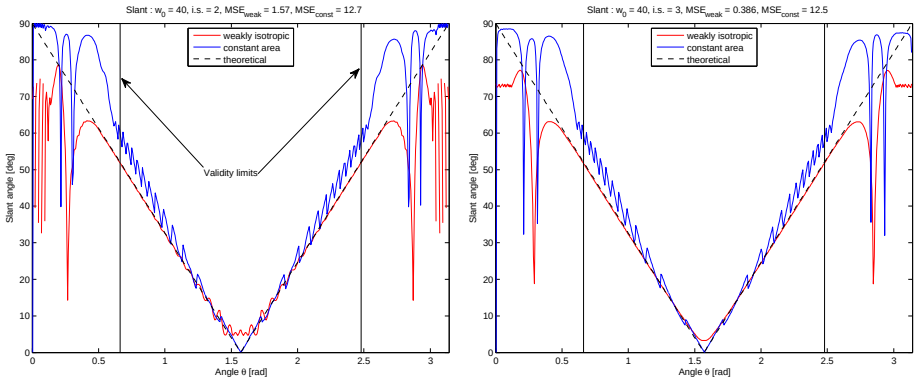
and the second from the constant area assumption (Equation 2.16)

$$\frac{\nabla A}{A} = \frac{\nabla m}{m} + \frac{\nabla M}{M} = -\tan\sigma \begin{pmatrix} 3 + r\kappa_t / \cos\sigma \\ r\tau \end{pmatrix}.$$

Like mentioned in subsection 2.3.2, we have not used exactly this last equation but we have taken its magnitude because $\kappa_t \ll 1$ and $\tau \ll 1$ in the scenes we use. So the slant σ is given by

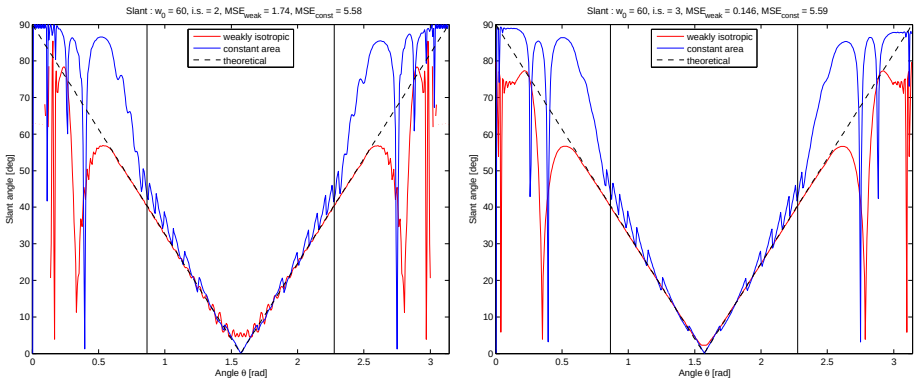
$$\left\| \frac{\nabla A}{A} \right\| = 3 \tan \sigma, \quad \sigma = \arctan\left(\frac{1}{3} \left\| \frac{\nabla A}{A} \right\|\right).$$

The results are shown in Figure 3.7. We obtained them by taking, for every angle θ, the mean value of the slant along the φ direction. The evolution of the slant along the θ direction can therefore be observed. Various frequencies ν_0 have been used (the same values as for the distance estimation in section 3.2), but only the results for $\nu_0 = 40$ and $\nu_0 = 60$ are displayed. The limits of the confidence intervals are indicated by black vertical lines and have been computed with a "practical" sampling frequency 20 times higher than the signal frequency ν_0.



(a) $\nu_0 = 40$ i.s. $= 2$

(b) $\nu_0 = 40$ i.s. $= 3$



(c) $\nu_0 = 60$ i.s. $= 2$

(d) $\nu_0 = 60$ i.s. $= 3$

Figure 3.7: Slant estimation

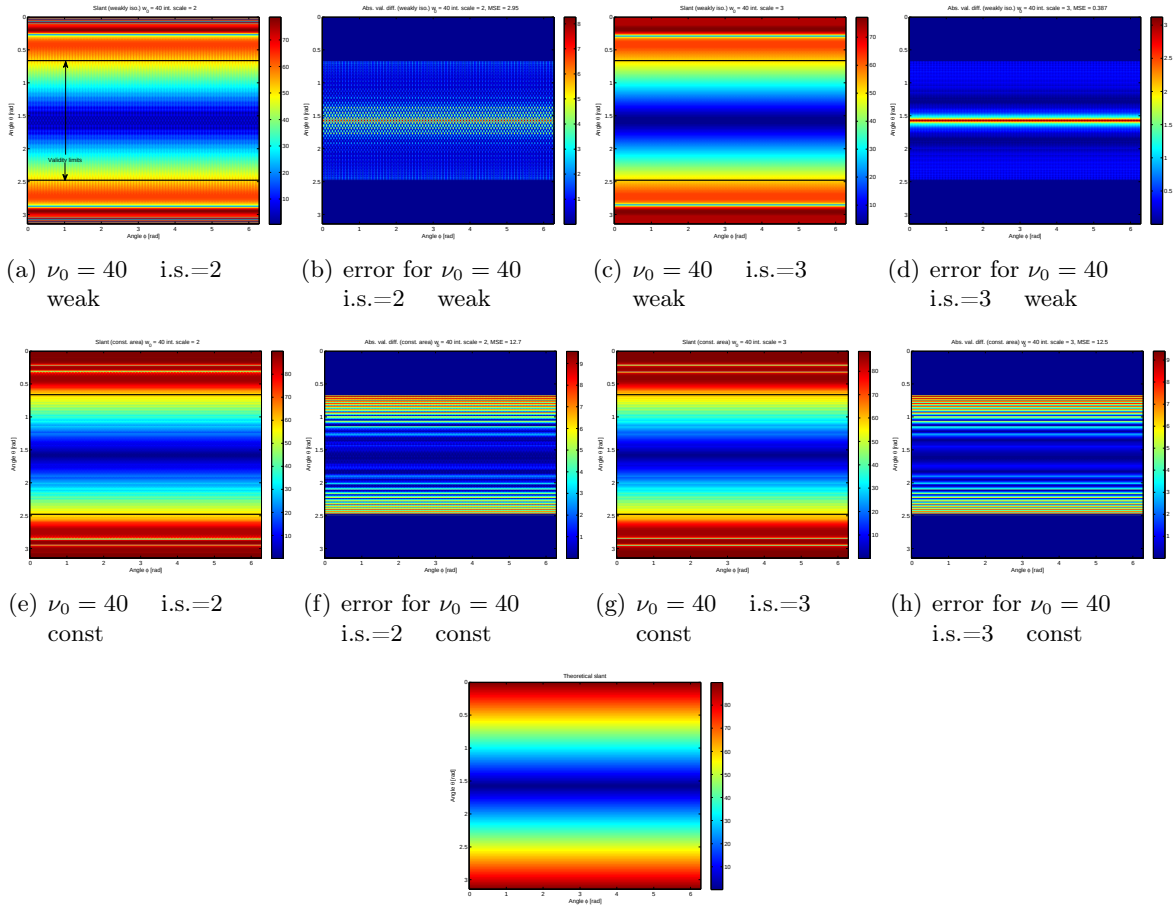
The estimation of the slant within the limits of validity works also quit well. The weakly isotropic assumption seems to give better results than the less restrictive constant area assumption when we move away from $\theta = \pi/2$ but near $\theta = \pi/2$, the constant area assumption estimates the slant better than the weakly isotropic assumption. To compare the two estimation of the slant, the Mean Square Error

$$MSE = \frac{1}{N} \sum_{i=1}^{N} |f_{theo}(i) - f_{estim}(i)|^2$$

has been computed for the two assumption within the validity interval, where f_{theo} is the theoretical value of the slant, f_{estim} the estimation obtained by one of the two methods and N the number of samples within the interval. The values are reported in the Table 3.1.

	weakly iso.		const. area	
	int. scale $s = 2t$	int. scale $s = 3t$	int. scale $s = 2t$	int. scale $s = 3t$
$\nu_0 = 40$	1.57	0.386	12.7	12.5
$\nu_0 = 60$	1.74	0.146	5.58	5.59

Table 3.1: MSE for the weakly iso. and const. area assumptions.



(a) $\nu_0 = 40$ i.s.=2 weak (b) error for $\nu_0 = 40$ i.s.=2 weak (c) $\nu_0 = 40$ i.s.=3 weak (d) error for $\nu_0 = 40$ i.s.=3 weak

(e) $\nu_0 = 40$ i.s.=2 const (f) error for $\nu_0 = 40$ i.s.=2 const (g) $\nu_0 = 40$ i.s.=3 const (h) error for $\nu_0 = 40$ i.s.=3 const

(i) Theoretical slant

Figure 3.8: Slant estimation for int. scale = 2 and 3

f_1 has been generated to verify the weakly isotropic assumption, and it is the weakly isotropic assumption who gives the best estimation for this signal. When the size of the window w used to accumulate the local statistics increases, a smoothing effect appears on the data. A bigger window stabilizes the mean value, its variations are smaller. The smoothing is more visible on the curves of the weakly isotropic assumption and the values of the table confirm it, but the overall behavior remains unchanged with the change of integration scale.
To check if the observed behavior was not due to the averaging we applied along the φ direction,

we computed the MSE on each point of the validity interval. For the frequency $\nu_0 = 40$, the Figure 3.8 shows the estimations of the slant on the complete grid for the two integration scales, and the absolute values of the estimation error within the validity intervals. The values of the MSE are in the Table 3.2.

	weakly iso.		const. area	
	int. scale $s = 2t$	int. scale $s = 3t$	int. scale $s = 2t$	int. scale $s = 3t$
$\nu_0 = 40$	2.95	0.387	12.7	12.5
$\nu_0 = 60$	3.07	0.147	5.6	5.59

Table 3.2: MSE for the weakly iso. and const. area assumptions.

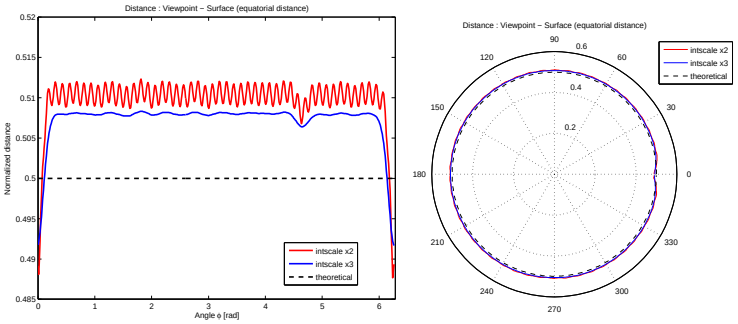
The only real difference between the two tables appears for the weakly isotropic assumption computed with an integration scale $s = 2t$. The averaging along the φ direction is surely responsible of the MSE decrease in Table 3.1. The other values are practically unchanged.
This second experiments shows that the slant estimation around the equator, $\theta = \pi/2$, is more accurate with the constant area assumption, but when we move away from $\theta = \pi/2$, the weakly isotropic assumption gives better results. All this has been done on a signal having a constant frequency. How the two methods will behave with more complex signals is reported in the next sections.

3.4 Cylinder analysis

Two different scenes with the cylinder have been generated. The cylinder and its texture are identical in both cases, but in the first case the omnidirectional camera is centered along the cylinder axis at half height (at the origin of the Cartesian basis), and in the second case the situation is identical to the case used in Section 3.1. The results are presented in the following subsections. We add the theoretical values of the slant only, because the representation of the distance is quite trivial for such shapes.

3.4.1 Centered cylinder

We start by estimating the distance in the equatorial plane for the two integration scales $s = 2t$ and $s = 3t$. To display the data, we normalize them as follows. The reference plane is the equatorial plane of the unit sphere. On the equator, we take the sum of the minimum and maximum value of the estimated distance and we use this value as the normalization coefficient. This coefficient corresponds to the diameter of the cylinder. The results of the distance estimation on the equatorial plane are shown in Figure 3.9. As we can see, the distance estimation is very good for both integration scales. The estimation computed with $s = 3t$ gives a smoother curve compared to the case $s = 2t$.
For the estimation of the distance on the entire spherical grid, the data are displayed in order to recreate the 3-D shape of the scene. For every point of the spherical grid, a ray having a length equal to the estimated distance is plotted for that direction. The data are normalized with the same normalization coefficient used on the equator.
The estimation of the distance for the top and bottom surfaces of the cylinder is more problematic, because the texture is "strangely" rendered on these two surfaces. The values obtained for these two surfaces are to large compared to what exactly should be. Since the dimensions

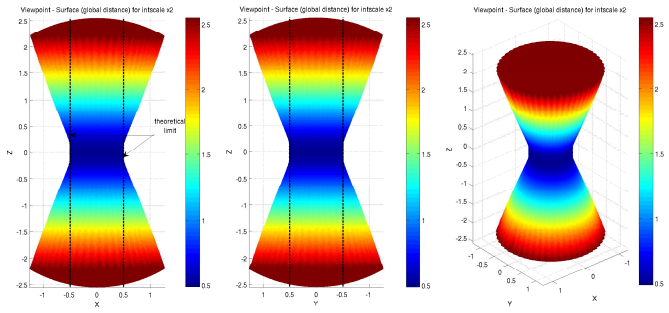


(a) Linear plot (b) Polar plot

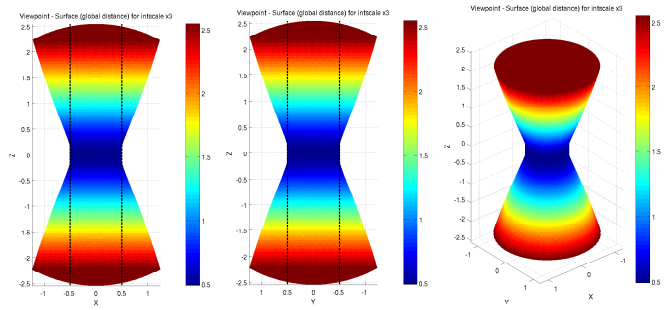
Figure 3.9: Equatorial distance for the centered cylinder.

of the scene are exactly known, diameter = 60, height = 300 and camera position at half height of the cylinder, we use them to find the maximal distance d_{max}, who occurs on the edge of the cylinder. We divide then d_{max} by the normalization coefficient, i.e. the diameter of the cylinder, and we obtain the maximal authorized value for the normalized distance.

$$\sqrt{(60/2)^2 + (300/2)^2} = d_{max}, \quad d_{max}/60 = 2.55.$$



(a) Projection Oxz, i.s.=2 (b) Projection Oyz, i.s.=2 (c) 3-D view, i.s.= 2



(d) Projection Oxz, i.s.=3 (e) Projection Oyz, i.s.=3 (f) 3-D view, i.s.=3

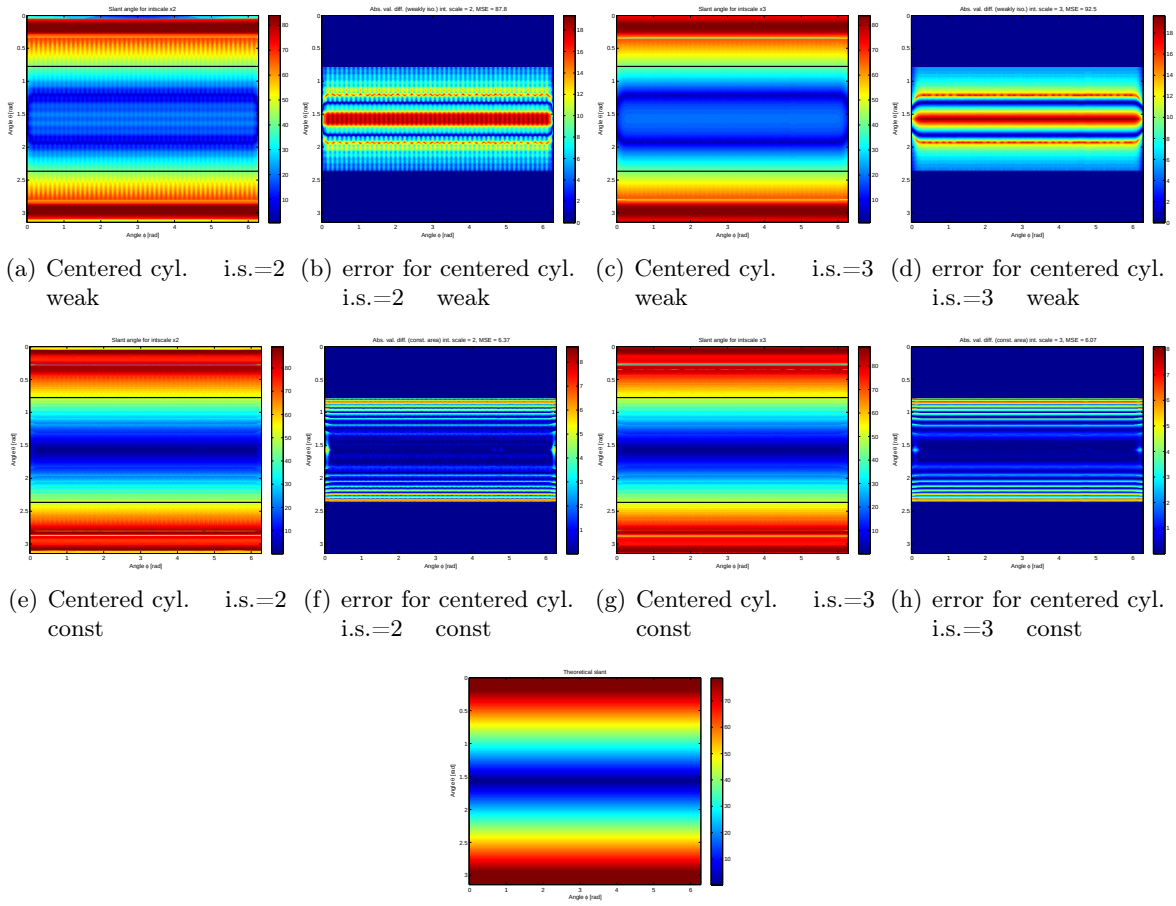
Figure 3.10: Distance estimation for the centered cylinder.

The values of the normalized estimated distance that exceed this limit are replaced by this

limit. The results are visible in the Figure 3.10. In the vicinity of the equator, the "reconstruction" of the cylinder is for both integration scales perfect. The distance is within the theoretical limits. When we move away from the equator, the situation is no more ideal and the distance is overestimated.
For the computation of the slant, the maximum vertical frequency of the texture has been estimated by counting the number of alternating black-white rectangles, and it is of 24. We have 24 periods over $2\arctan(150/30)$ radians, that corresponds to a frequency of approximately 55. Since it is an approximation, we round the frequency to 50 and we use this maximum frequency to limit the validity interval along the θ direction. The results for the slant estimation are shown in the Figure 3.11 and the values of the MSE are in the Table 3.3.

	weakly iso.	const. area
int. scale $s = 2t$	87.8	6.37
int. scale $s = 3t$	92.5	6.07

Table 3.3: MSE for the weakly iso. and const. area assumptions for the centered cylinder.



(a) Centered cyl. i.s.=2 weak (b) error for centered cyl. i.s.=2 weak (c) Centered cyl. i.s.=3 weak (d) error for centered cyl. i.s.=3 weak

(e) Centered cyl. i.s.=2 const (f) error for centered cyl. i.s.=2 const (g) Centered cyl. i.s.=3 const (h) error for centered cyl. i.s.=3 const

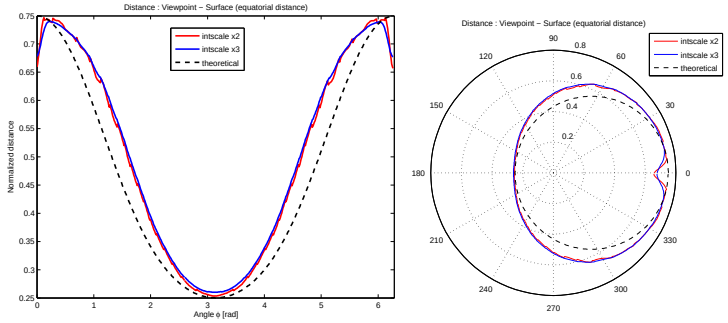
(i) Centered cyl. theoretical

Figure 3.11: Slant estimation of the centered cylinder for int. scale = 2 and 3

The results are a bit surprising. With the synthetic signal f_1, the weakly isotropic assumption gave the best results. For this situation, it is without a doubt the constant area assumption who gives the better results when referring to the obtained MSE values. It seems that for a "more realistic" signal, the less restrictive assumption estimates more precisely the slant because it is more frequent to come across a texture that, seen locally, seems to have a constant area.
Even if both cases are quit similar (the camera is placed in the center of a cylinder, with a finite height for this case and with an infinite height for the f_1 case), the signal f_1 has a pattern that changes in a continuous way (sine signal) and resembles to a blob texture, when the chessboard is composed of discontinuities between black and white rectangles. So the similarity stops here, because the textures are intrinsically different.

3.4.2 Displaced cylinder

We start again with the analysis of the distance on the equatorial plane. The normalization coefficient is obtained identically as for the centered cylinder, i.e. we sum the minimum and the maximum value of the estimated distance that gives the diameter of the cylinder. The results are available in the Figure 3.12.



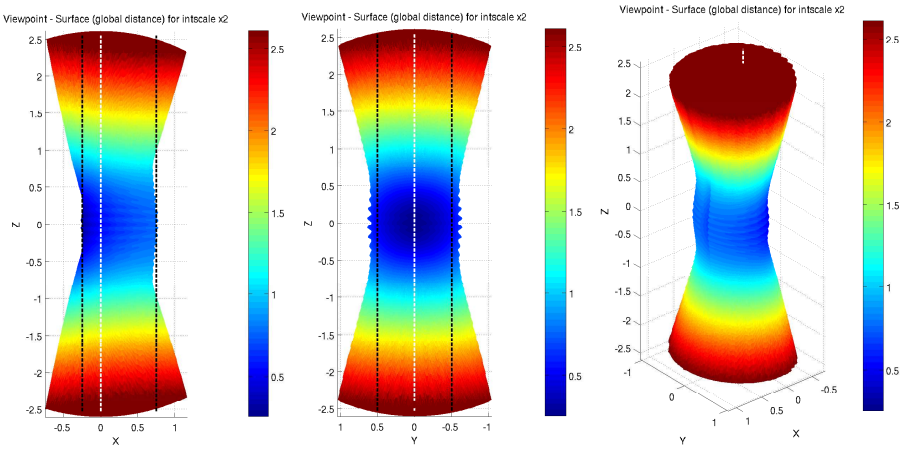
(a) Linear plot (b) Polar plot

Figure 3.12: Equatorial distance for the displaced cylinder.

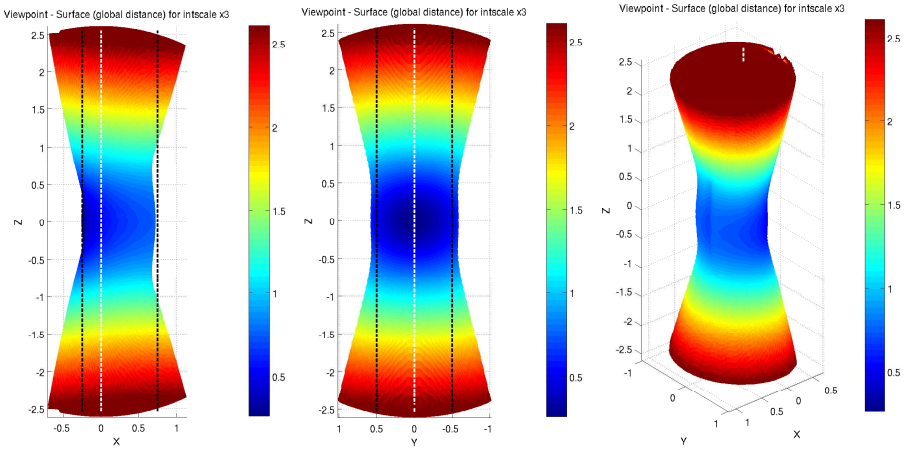
The estimated distance is not as good as for the previous case. With basic geometry, the theoretical distance r is given by

$$r = \frac{1}{2}(R\cos\varphi + R\sqrt{3+\cos^2\varphi}$$

where R is the cylinder radius and φ the longitudinal angle. The overestimation of the distance is surely due to the geometry of the problem, i.e we work on omnidirectional images and not planar ones. What happens in the vicinity of $\varphi = 0$ and $\varphi = 2\pi$ comes from the rendered texture. Blender is not the best software for the rendering of omnidirectional scenes. On the edges of the spherical grid, the texture should be continuous but it is not.
The change of integration scale, from $s = 2t$ to $s = 3t$, has a smoothing effect on the curve obtained with $s = 2t$. The extreme values of the curve $s = 3t$ are smaller than those from the curve $s = 2t$. It is the normal behavior of the smoothing.
On the entire grid, we obtain the results of the Figure 3.13. The results are equivalent to those obtained for the centered cylinder in the vicinity of the equator. The distance stays within the theoretical limits (black dashed lines) and the cylinder axis (the white dashed line)



(a) Projection Oxz, i.s.=2 (b) Projection Oyz, i.s.=2 (c) 3-D view, i.s.= 2



(d) Projection Oxz, i.s.=3 (e) Projection Oyz, i.s.=3 (f) 3-D view, i.s.=3

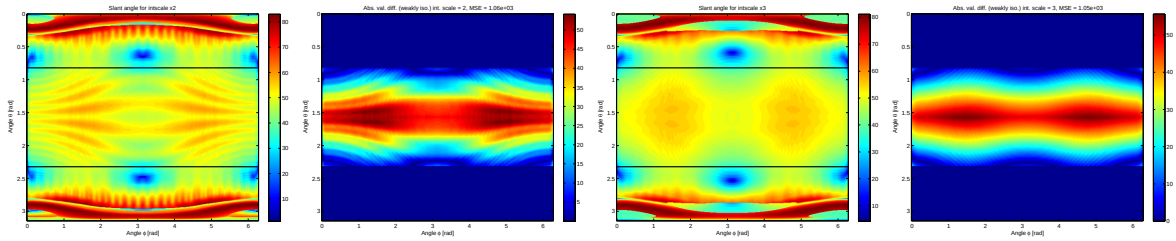
Figure 3.13: Distance estimation for the displaced cylinder.

proves that the displacement of the camera on the x axis is detected. The cylinder seems a bit too large when we look on the Oyz projection, and the surface is undulating. The same overestimation arises when we move away from the equator. With the integration scale $s = 3t$, the cylinder surface is smoothed, but the overall estimation is identical.

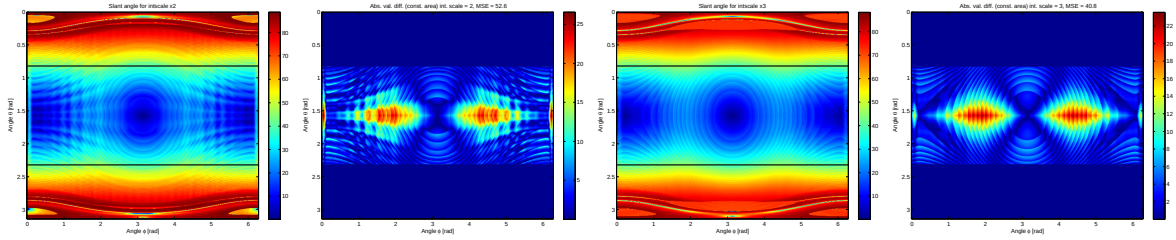
To estimate the slant, we need the maximum frequency of the signal to set a validity limit. The maximum vertical frequency is approximately of 59 in this case. To increase the validity limit, we lower the value to 55. The results are available in the Figure 3.14 and the MSE in the Table 3.4.

	weakly iso.	const. area
int. scale $s = 2t$	$1.06{\cdot}10^3$	52.6
int. scale $s = 3t$	$1.05{\cdot}10^3$	40.8

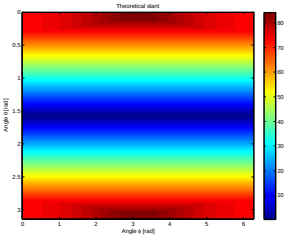
Table 3.4: MSE for the weakly iso. and const. area assumptions for the displaced cylinder.



(a) Displaced cyl. i.s.=2 weak (b) error for displaced cyl. i.s.=2 weak (c) Displaced cyl. i.s.=3 weak (d) error for displaced cyl. i.s.=3 weak



(e) Displaced cyl. i.s.=2 const (f) error for displaced cyl. i.s.=2 const (g) Displaced cyl. i.s.=3 const (h) error for displaced cyl. i.s.=3 const



(i) Displaced cyl. theoretical

Figure 3.14: Slant estimation of the displaced cylinder for int. scale = 2 and 3

Here again, the constant area assumption gives better results for the slant estimation than the weakly isotropic assumption. The same remark as for the preceding case is valid: the texture has a more "constant area" behavior.

3.5 Cube analysis

This scene is more complex than the cylinder one, because it has right angles that delimit different surfaces (there is only one surface for the cylinder). The cube has a black and white chessboard painted on all its interior faces. Two cases will be analyzed. In the first case, the omnidirectional camera is placed in the center of the cube, and in the second case, it is slightly displaced.

3.5.1 Centered cube

We will analyze this scene like we did for the cylinder. The first measure is the equatorial distance. We will use it to determine the normalization coefficient for the distance estimation. To obtain the normalization coefficient, we sum the minimum value of the estimated distance

on the equator with the value found at the opposite point of the equator (see Figure 3.15). This value gives the length of an edge of the cube.

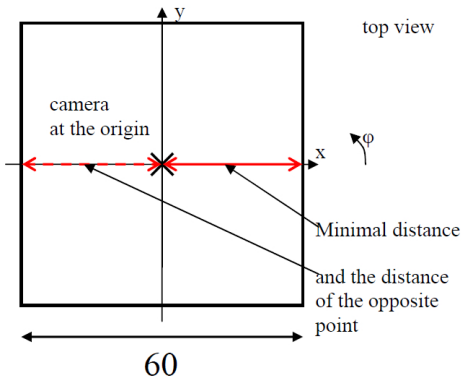
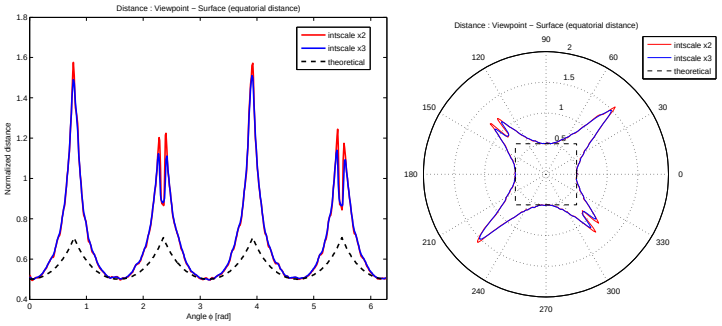


Figure 3.15: Normalization coefficient of the centered cube.



(a) Linear plot (b) Polar plot

Figure 3.16: Equatorial distance (not normalized) for the centered cube.

The distance along the equator is shown in Figure 3.16. A behavior that did not happen with the cylinder, because there was only one continuous surface, excepted at the limit of the cylinder, can be seen at the edges, who are problematic areas for the analysis, and the computed values are wrong. To understand this, we have looked at the scales that were selected by the $\det \mu_L$ method, and unfortunately only a very small range of scales (from the first to the sixth scale, the smallest ones) were selected for the computation. The selected scales are represented in the Figure 3.17. There is a large area that is associated to the smallest scale used for the analysis, 0.01. This area covers the most problematic points of the image, i.e. the edges and the corners. This figure proves then to be useful. We can use this scales map to create a mask that will define the validity interval to be used for the comparison of the slant estimation. We will not take into account the area associated to the minimum scale.

The fact that the texture is, from the beginning, composed of small rectangles is not helpful and a chessboard composed of bigger squares could give a different (and better) result. Because of this, we decided to limit the estimated values obtained from the computation to the maximum theoretical value of the normalized distance on the equator, i.e. $\sqrt{0.5}$. The result of this limitation is shown on Figure 3.18. The distance seems to be correctly estimated only in the

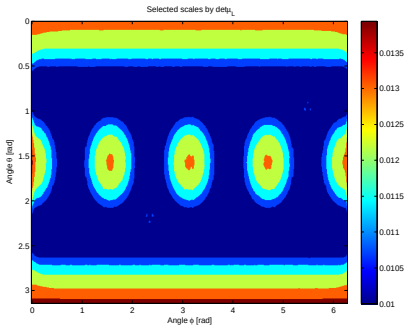
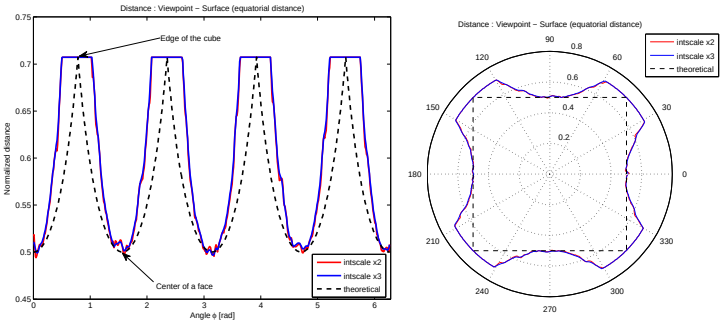


Figure 3.17: Selected scales for the centered cube.

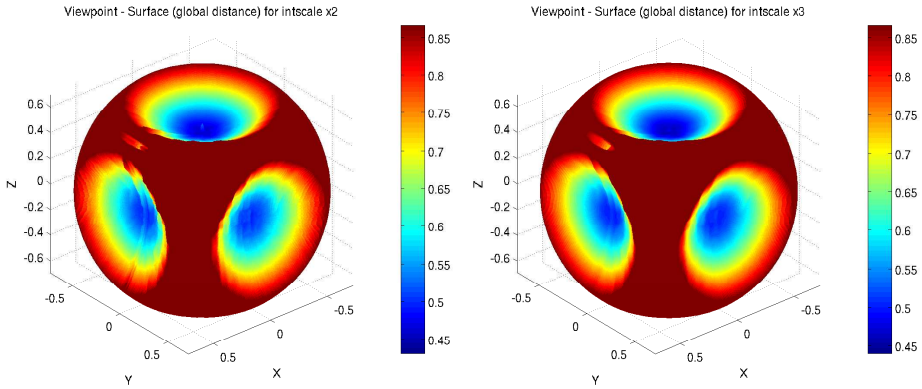


(a) Linear plot (b) Polar plot

Figure 3.18: Equatorial distance for the centered cube.

middle of the surfaces, elsewhere it is overestimated. The only difference between the curves is the smoothing due to the bigger integration scale $s = 3t$.

The distance estimation on the entire grid gives the results shown in the Figure 3.19. The maximum normalized value equals $\sqrt{3}/2$ for the 3-D case (normalized distance from the cube center to a corner). It is possible to imagine that the shape is a cube thanks to the applied

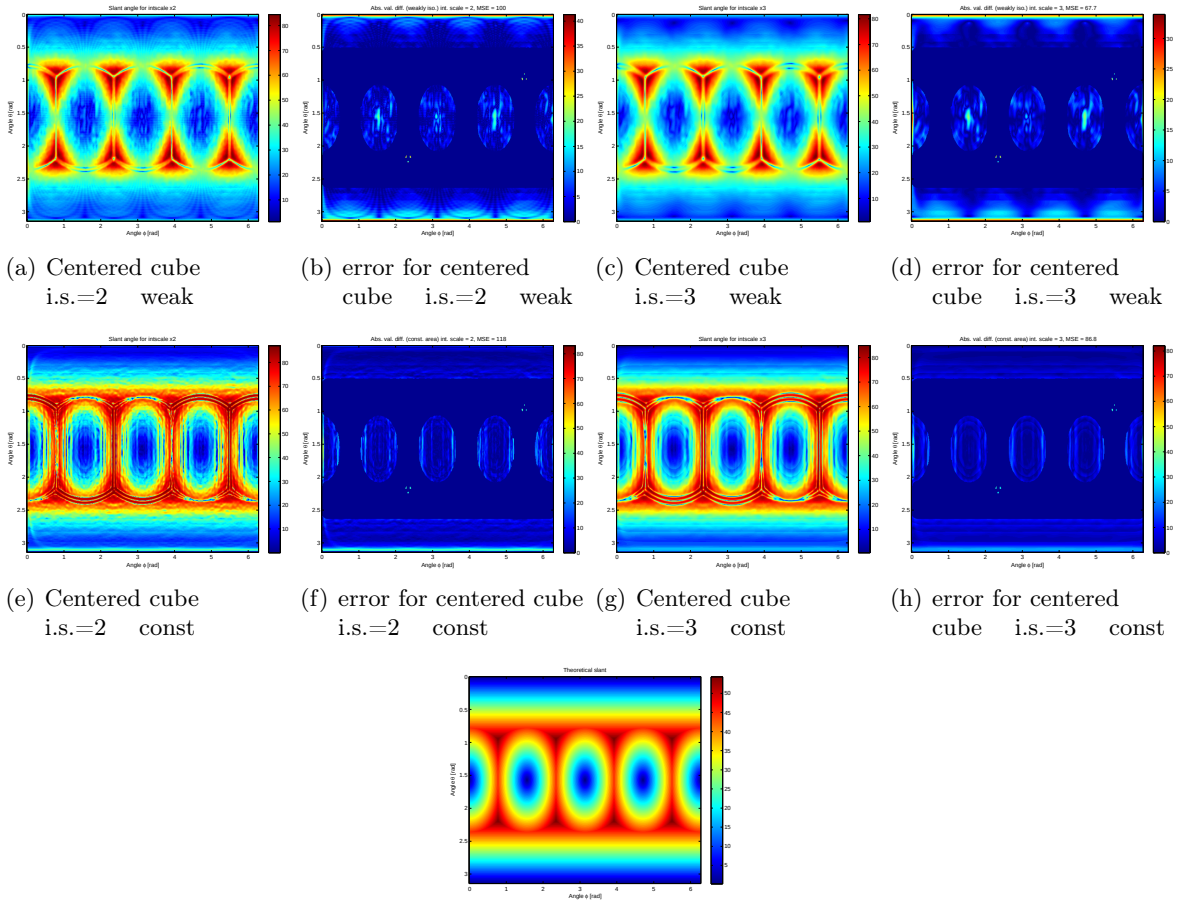


(a) 3-D distance estimation, i.s.=2 (b) 3-D distance estimation, i.s.=3

Figure 3.19: Estimated distance on the entire grid for the centered cube.

limit. Without the limit, the shape does not really resemble to a cube, because the distance values near the edges increase when we move away from the equator, and they are even bigger in the corners.

The slant estimation was not easier. Like mentioned above, we used the scales map to create a mask defining the validity interval for the comparison of the two methods. The results are shown in the Figure 3.20 and the MSE is found in the Table 3.5. The best results are obtained



(a) Centered cube i.s.=2 weak
(b) error for centered cube i.s.=2 weak
(c) Centered cube i.s.=3 weak
(d) error for centered cube i.s.=3 weak
(e) Centered cube i.s.=2 const
(f) error for centered cube i.s.=2 const
(g) Centered cube i.s.=3 const
(h) error for centered cube i.s.=3 const
(i) Centered cube theoretical

Figure 3.20: Slant estimation of the centered cube for int. scale = 2 and 3

	weakly iso.	const. area
int. scale $s = 2t$	100	118
int. scale $s = 3t$	67.7	86.8

Table 3.5: MSE for the weakly iso. and const. area assumptions for the centered cube.

with the weakly isotropic assumption this time. But we do not really think that this measure can be taken seriously, because the validity interval has not been defined carefully, only by chance after the observation of the scales map.

3.5.2 Displaced cube

The Figure 3.21 represents the scene. The cube is identical to the previous case, only the

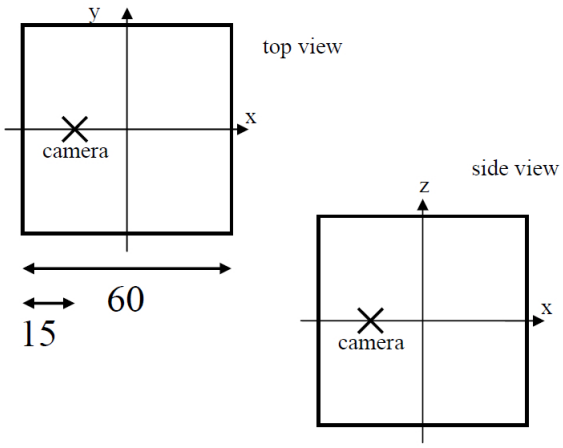
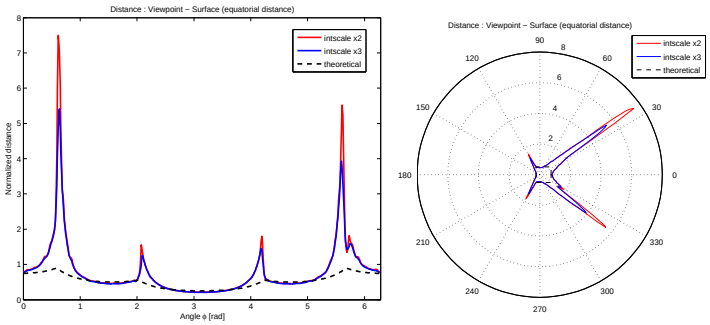


Figure 3.21: Scene of the displaced cube.

position of the camera has been changed. We begin by analyzing the equatorial distance in order to retrieve the normalization coefficient. The coefficient is obtained in the same way as for the centered case. The equatorial distance is shown in Figure 3.22. The edge effect here

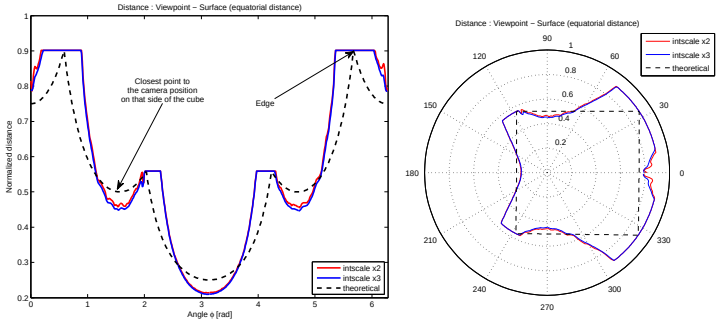


(a) Linear plot (b) Polar plot

Figure 3.22: Equatorial distance (not normalized) for the displaced cube.

is more important than in the centered case. A limit is therefore necessary to constrain the values within the maximum value of the normalized distance. Since the cube is displaced, we define two different values depending on which side the distance is evaluated. The side that is nearer to the camera will be limited to $\sqrt{5}/4$ and the other side to $\sqrt{13}/4$. The limited distance is shown in Figure 3.23. The estimation is not very good. The only places where the estimated value is not too different from the theoretical are the closest points of each surface to the point where the camera is placed.

We looked also at the scales that were selected by $\det \mu_L$ for the analysis (see Figure 3.24). A few more scales than in the centered case were selected because the size of the texture varies more in this situation. Like for the centered case, the smallest scale is assigned to the problematic regions (edges, corners) but also to complete sides of the cube (the most distant).



(a) Linear plot (b) Polar plot

Figure 3.23: Equatorial distance for the displaced cube.

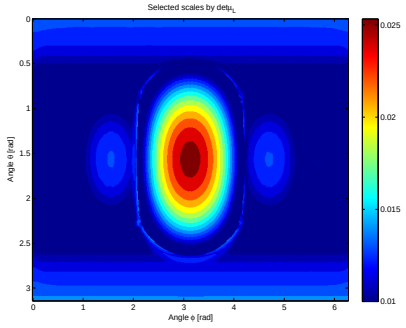
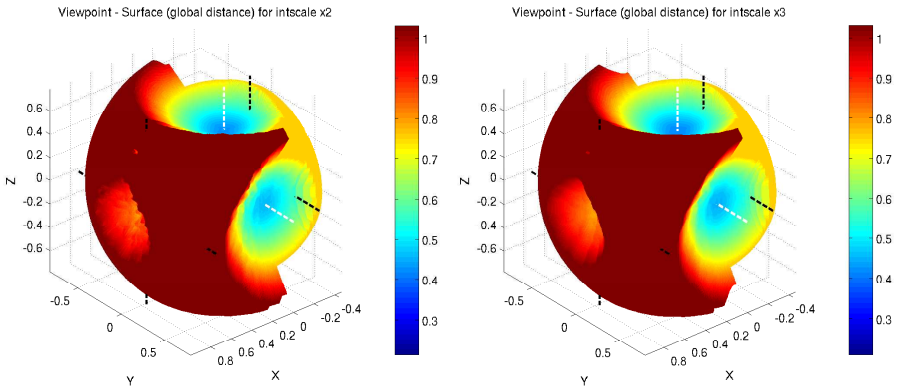


Figure 3.24: Selected scales for the displaced cube.

We decided to do the same as for the other case, i.e. use this scales map to create a mask to define the validity limits used for the computation of the MSE.

Before proceeding to the comparison of the slant estimation, the distance on the entire grid is shown on the Figure 3.25. The white dashed lines represent the Cartesian basis. We can



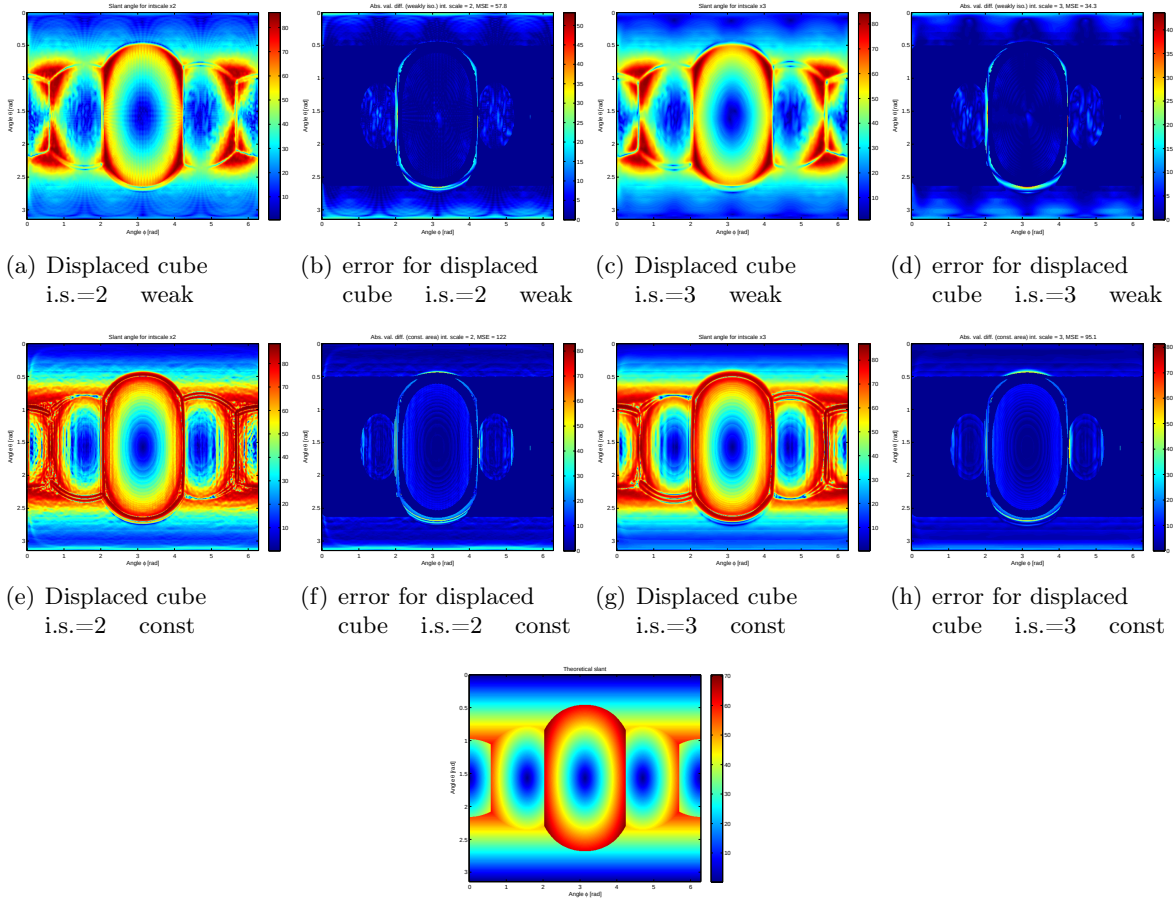
(a) 3-D distance estimation, i.s.=2 (b) 3-D distance estimation, i.s.=3

Figure 3.25: Estimated distance on the entire grid for the displaced cube.

see that the "reconstructed cube" is effectively displaced. The black dashed lines show the

theoretical limits of the cube, but we can easily see that when we move away from the nearest points to the position of the camera the distance estimation is poor and the values of the distance are only limited by the limits we have set. All we can say is that for this particular scene, the points where the distance can "correctly" be estimated are the nearest points to the position of the camera.

Now, for the estimation of the slant, we used the mask generated from the scales map. The results are shown in the Figure 3.26 and the MSE is found in the Table 3.6. From these



(a) Displaced cube i.s.=2 weak
(b) error for displaced cube i.s.=2 weak
(c) Displaced cube i.s.=3 weak
(d) error for displaced cube i.s.=3 weak
(e) Displaced cube i.s.=2 const
(f) error for displaced cube i.s.=2 const
(g) Displaced cube i.s.=3 const
(h) error for displaced cube i.s.=3 const
(i) Displaced cube theoretical

Figure 3.26: Slant estimation of the displaced cube for int. scale = 2 and 3

	weakly iso.	const. area
int. scale $s = 2t$	57.8	122
int. scale $s = 3t$	34.3	95.1

Table 3.6: MSE for the weakly iso. and const. area assumptions for the displaced cube.

results, again the weakly isotropic assumption achieves a better estimation. But looking at the figures of the error between the theoretical and the estimated slant, we see that the mask

does not completely remove the edges (where the error is very large). So we can not trust blindly the results of the comparison.

Conclusion

We have adapted a shape from texture method developed initially for planar images on the sphere, in order to compute the shape informations contained in omnidirectional images. From the obtained results, the method gives satisfactory results only for limited situations.

The selection of the right scales is not as simple as it seems. There are different methods for selecting the right scales. We used the $\det \mu_L$ because it seemed to give good results. When the analyzed texture becomes "complicated", the scale selection is not accurate. It leads to an estimation not as precise as it should be, despite the fact that an averaging is performed to limit these phenomena. The size of the window used for the averaging has above all a smoothing effect.

The distance estimation in the vicinity of the equatorial plane gives good results. Only if discontinuities arise in the texture, then the estimation looses its accuracy. In general, when the texture satisfies the *a priori* assumptions that are made, the results are good (this is always near the equator). But when we move away from the equator, the accuracy decreases.

The slant estimation can give also good results, but only within certain confidence intervals. The comparisons done with the MSE show mixed results. Sometimes the slant estimation by the weakly isotropic assumption performs better than the constant area assumption, who is a less restrictive assumption. For the cylinder for example, within a certain interval of the angle θ, one assumption gives the best estimation, but outside this interval the other assumption is better. There is no winner we can say. The best could be to use both in a complementary way.

Some future works that could be done:

Use real images of "simple" textures to see if the adapted method gives some good results for the distance and slant estimation.

Implement a function that compares the estimations of the slant from both assumptions with a new estimation method based on the 3-D shape obtained by the computed distance:
we take 3 neighbor points on $\mathbb{S}^2$ and their respective estimated distances d to obtain the projected vectors on the surface S

$$\vec{u_1} = \overrightarrow{(\theta_1, \varphi_1)} d(\theta_1, \varphi_1) \quad \vec{u_2} = \overrightarrow{(\theta_2, \varphi_2)} d(\theta_2, \varphi_2) \quad \vec{u_3} = \overrightarrow{(\theta_3, \varphi_3)} d(\theta_3, \varphi_3)$$

now, we define two new vectors $\vec{w_1}$ and $\vec{w_2}$ in the tangent plane of the surface at the point $\vec{u_1}$

$$\vec{w_{12}} = \vec{u_2} - \vec{u_1} \quad \vec{w_{13}} = \vec{u_3} - \vec{u_1}$$

we compute the vector product between $\vec{w_{12}}$ and $\vec{w_{13}}$ to obtain the vector $\vec{n_1}$ who is normal to the tangent plane at the point $\vec{u_1}$ and we normalize it

$$\vec{n_1} = \frac{\vec{w_{12}} \times \vec{w_{13}}}{\|\vec{w_{12}} \times \vec{w_{13}}\|}$$

it is now possible to have an estimation of the slant at point $\vec{u_1}$ if we do the scalar product between $\vec{n_1}$ and the normalized vector $\vec{u_1}$

$$\sigma_d = \vec{n_1} \cdot \frac{\vec{u_1}}{\|\vec{u_1}\|}$$

we can analyze the absolute value of the difference of the slant estimations $\sigma - \sigma_d$ where σ is the estimation given by one of the two assumption. If the difference is contained within an interval of acceptance that defines if the two estimations are compatible between them, the point $\vec{u_1}$ and its associated point on $\mathbb{S}^2$ can be kept. Otherwise, if the difference is outside the acceptance interval, the point $\vec{u_1}$ and its associated point on $\mathbb{S}^2$ must not be kept.

Another possible implementation would be the definition of a correction function for the distance. As an example we can use the behavior of the estimated distance with the cylinder case. Near the equator, the distance is well estimated. But when we move away from the equator, the estimation is less precise. We could analyze the behavior of the distance estimation function in the direction θ and try to find some corrective function in order to obtain a better estimation for values of θ distant from the equator.

Lausanne, 21 December 2007 Eugenio De Vito

List of Figures

List of Tables

Bibliography

[1] J.P. Antoine, L. Demanet, L. Jacques, and P. Vandergheynst. Wavelets on the sphere: implementation and approximations. *Applied and Computational Harmonic Analysis*, 13(3):177–200, 2002.

[2] S. Baker and S. Nayar. Single viewpoint catadioptric cameras. 2001.

[3] I. Bogdanova, P. Vandergheynst, JP Antoine, L. Jacques, and M. Morvidone. Stereographic wavelet frames on the sphere. *Applied and Computational Harmonic Analysis*, 19:223–252, 2005.

[4] K. Daniilidis. Omnidirectional image processing, iccv 2003 course.

[5] K. Daniilidis, A. Makadia, and T. Bülow. Image processing in catadioptric planes: Spatiotemporal derivatives and optical flow computation. In *3th Workshop on Omnidirectional Vision*, 2002.

[6] L.M.J. Florack, B.M. ter Haar Romeny, J.J. Koenderink, and M. Viergever. Scale and the differential structure of images. *Image and Vision Computing*, 10(6):376–388, 1992.

[7] J. Garding. Shape from texture for smooth curved surfaces. In *Computer Vision - ECCV '92. Second European Conference on Computer Vision Proceedings*, pages 630–8. Springer-Verlag Berlin, Germany, 1992.

[8] J. Garding and T. Lindeberg. Direct estimation of local surface shape in a fixating binocular vision system. *Proc. 3rd European Conference on Computer Vision*, 800:365–376, 1994.

[9] J. Gårding and T. Lindeberg. Direct computation of shape cues using scale-adapted spatial derivative operators. *International Journal of Computer Vision*, 17(2):163–191, 1996.

[10] C. Geyer and K. Daniilidis. Catadioptric projective geometry, 2001.

[11] J. Krumm and S. Shafer. Local spatial frequency analysis of image texture. 1990.

[12] T. Lindeberg. Discrete derivative approximations with scale-space properties: A basis for low-level feature extraction. *Journal of Mathematical Imaging and Vision*, 3(4):349–376, 1993.

[13] T. Lindeberg. Principles for automatic scale selection. *Handbook on Computer Vision and Applications*, 2:239–274, 1999.

[14] T. Lindeberg and J. Garding. Shape from texture from a multi-scale perspective. In *Computer Vision, 1993. Proceedings., Fourth International Conference on*, pages 683–691, 1993.

[15] T. Lindeberg and J. Garding. Shape-adapted smoothing in estimation of 3-d depth cues from affine distortions of local 2-d brightness structure. *Proc. ECCV*, pages 389–400, 1994.

[16] Tony Lindeberg. Edge detection and ridge detection with automatic scale selection. *Int. J. Comput. Vision*, 30(2):117–156, 1998.

[17] Tony Lindeberg. Feature detection with automatic scale selection. *Int. J. Comput. Vision*, 30(2):79–116, 1998.

[18] J. Malik and R. Rosenholtz. Computing Local Surface Orientation and Shape from Texture for Curved Surfaces. *International Journal of Computer Vision*, 23(2):149–168, 1997.

[19] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, 2005.

[20] K.A. Stevens. The information content of texture gradients. *Biological Cybernetics*, 42(2):95–105, 1981.

[21] A.C. Super, B.J.; Bovik. Shape-from-texture by wavelet-based measurement of local spectral moments. *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, pages 296–301, 15-18 Jun 1992.

[22] BJ Super and AC Bovik. Shape from texture using local spectral moments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(4):333–343, 1995.

[23] Y. Wiaux, L. Jacques, P. Vielva, and P. Vandergheynst. Fast directional correlation on the sphere with steerable filters. *The Astrophysical Journal, Volume 652, Issue 1, pp. 820-832.*, 11 2006.