



Project no.: FP7-ICT-217077
Project full title: Heterogeneous 3-D Perception across Visual Fragments
Project Acronym: EYESHOTS
Deliverable no.: D2.2a
Title of the deliverable: Algorithm for 3D scene description through interactive visual stereopsis adaptation using a conventional binocular vision platform

Date of Delivery:	01 March 2010
Organization name of lead contractor for this deliverable:	K.U.Leuven
Author(s):	K. Pauwels, M. Chessa, N. Chumerin, F. Solari, S.P. Sabatini, M. Van Hulle
Participant(s):	K.U.Leuven, UG
Workpackage contributing to the deliverable:	WP2
Nature:	Other (software module)
Version:	1.0
Total number of pages:	20
Responsible person:	Marc Van Hulle
Revised by:	M. Chessa, F. Solari, S.P. Sabatini
Start date of project:	1 March 2008 Duration: 36 months

Project Co-funded by the European Commission within the Seventh Framework Programme		
Dissemination Level		
PU	Public	X
PP	Restricted to other program participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract:

An algorithm is presented for disparity estimation that does not require precise calibration information (related to the relative orientation of the cameras). This algorithm forms the basis for precise 3D depth estimation and for vergence control methods that are robust to imprecise motor feedback. Correspondence estimation and auto-calibration are performed simultaneously by this algorithm. The techniques involved can either operate directly on the camera (or retinal) input images, or be applied after a space-variant transformation (cortical mapping) of the input images. In both the retinal and cortical domain, greatly improved disparity estimates are obtained as compared to standard disparity or vector disparity (cf. optical flow) estimation algorithms. The algorithms are made available to the consortium in the form of a Matlab software package.

Contents

1	Executive Summary	2
2	Introduction	3
3	Auto-calibration in the Retinal Domain	4
3.1	Phase-based Disparity Estimation	4
3.2	Interpolating Phase Across Orientation	5
3.3	Epipolar Geometry Estimation	7
3.4	Combined Stereo and Auto-calibration	8
3.5	Results	10
4	Auto-calibration in the Cortical Domain	12
4.1	Central Blind-spot Model	12
4.2	Cortical Correspondence Estimation	13
4.3	Modified Algorithm	13
5	Results	14
6	Future Extensions	17
7	Package Description	17

1 Executive Summary

This deliverable describes the methods and algorithms developed by K.U.Leuven and University of Genoa regarding Task 2.2 (Interactive Depth Perception) of Work Package 2 (Active Stereopsis). The resulting algorithms are made available as a Matlab software package that is described in Section 7 of this document.

Task 2.2 is concerned with the extraction of depth (3D structure) by integrating disparity information across different eye movements. Transforming disparity from eye- to head-centric coordinates, but also estimating disparity (and controlling vergence) relies on accurate calibration information (in terms of the relative orientation of the eyes). In the active systems considered in Eyeshots, the motor feedback can not provide this information (due to the limited precision) and therefore vision is used to improve upon this. The procedures developed for this constitute the main part of this deliverable.

Two algorithms are presented. The first algorithm operates directly on images received from the cameras (the retinal domain). It exploits initial calibration estimates to improve disparity estimation using a variety of warping mechanisms

operating in the spatial and orientation domain. These improved disparity estimates are then in turn used to improve the calibration information. The second algorithm uses the same concepts but operates in the cortical domain, on images transformed using a space-variant log-polar mapping. The performance of these algorithms is evaluated in different configurations (retinal, cortical, close to vergence and far-away from vergence) and compared to standard algorithms.

The methods proposed here can operate together with the vergence mechanisms presented in Task 2.1 in various ways. Improved calibration estimates can feed directly in the convolutional network for vergence control presented in Deliverable 2.1, but can also modulate the weights of the mechanism (also reported there) that integrates the population responses into the vergence control.

The methods can also be configured to operate in the periphery only, which is useful for steering vergence to locations memorized in terms of motor information (WP1). In this situation, there is no need for small foveal disparities.

The corrected gaze information is returned in the form of a corrected fundamental matrix. This is used for the coordinate transforms in Task 2.2, but also for the fine motor control in WP 1 that is required for the interactive exploration of the fragment. This kind of feedback from the visual system is important for the actual control of the robot eyes in Task 1.4, the requirements of which have been discussed in Deliverable 1.1.

2 Introduction

In this report, we present methods that simultaneously solve the disparity estimation and auto-calibration problems. Unless images are rectified, both horizontal and vertical disparities are present (vector disparity). We rely on the measured vertical disparities (or more generally, deviations orthogonal to the hypothesized epipolar lines) to update the estimated geometry. This procedure can be closely integrated with the disparity estimation itself. Vertical disparity has been used for this purpose before, either directly [6] or indirectly [13]. It has not been considered together with correspondence estimation and using the more general error signal relied upon here.

The matching procedure used in most dense stereo methods cannot be efficiently adapted to changing camera geometry information. For this reason, nearly all proposed systems separate the calibration and dense disparity estimation stages. The calibration step is typically performed off-line, using feature-based techniques. The image pairs are then rectified, and a dense matching procedure is run for refinement [12]. Estimating the epipolar geometry from noisy correspondences, possibly including many outliers, is problematic. To improve calibration accuracy, either a special calibration object is used, or the information of multiple image pairs is combined [20]. Epipolar geometry estimation is also often stabilized by exploiting physical restrictions on the camera configuration. Björkman and Eklundh [3] for example present a system for externally calibrating a stereo pair by assuming fixation and no rotations around the line of sight. Papadimitriou and Dennis [10] propose a self-rectification method that focuses only on the removal of vertical disparities. It assumes a convergent

camera system where only rotations around an axis parallel to the vertical axis (pan) need to be compensated for. This reduces the problem and stabilizes the camera geometry estimation. A real-time embedded system combining disparity estimation and self-rectification is presented in Gao et al. [8]. The system corrects for vertical shifts only, but, as in the method we propose, both system components are tightly coupled since the rectification is refined based on the quality of the disparity measurements.

We first present the basic retinal domain algorithm in Section 3. All the ingredients required in the cortical domain algorithm are introduced there: disparity estimation, phase warping, epipolar geometry estimation, and finally combining disparity and calibration estimation. We also present results comparing the performance to a standard algorithm. The algorithm is then adapted for operation in the cortical domain (Section 4) and results are presented comparing to a standard vector disparity estimation procedure.

3 Auto-calibration in the Retinal Domain

The method uses phase differences for the estimation of stereo correspondence in the absence of precise calibration information. To estimate possibly large 2D correspondences, it uses the responses from a multiscale and multi-orientation Gabor filterbank. Although this requires a computationally intense (but data-parallel) filtering step, the matching itself is effortless (*cf.* gradient-based methods in optical flow). To avoid having to re-filter the images while gradually improving the calibration, we propose to not actually rectify the images, but rather to adjust the read-out of the filter responses. For this purpose, we introduce a method to compensate for large orientation differences between left and right image features. Since we use noisy correspondences to update the epipolar geometry, we also present a simple alternation technique that increases the robustness.

3.1 Phase-based Disparity Estimation

The proposed method extends a coarse-to-fine multi-orientation phase-difference stereo disparity algorithm [15]. A closely-related optical flow algorithm has been shown to be suitable for real-time implementation on graphics hardware (GPUs) [11].

The stereo algorithm extracts phase using a bank of oriented Gabor filters. The different orientations, θ_q , are evenly distributed and equal to $\frac{q\pi}{K}$. We use a total of $K=8$ orientations in our implementation and let q range from 0 to $K-1$. For a specific orientation θ_q the spatial phase at pixel location $\mathbf{x} = (x, y)^T$ is extracted using 2D complex Gabor filters:

$$f_q(\mathbf{x}) = e^{-\frac{x^2+y^2}{2\sigma^2}} e^{j\omega_0(x \cos \theta_q + y \sin \theta_q)} , \quad (1)$$

with peak frequency ω_0 and spatial extension σ . The filterbank has been designed with efficiency in mind and relies on 11×11 separable spatial filter kernels that are applied to an image pyramid. For a more detailed description, see [14].

The filter responses, obtained by convolving the image, $I(\mathbf{x})$, with the oriented filter from Eq. (1) can be written as:

$$Q_q(\mathbf{x}) = (I * f_q)(\mathbf{x}) = \rho_q(\mathbf{x})e^{j\phi_q(\mathbf{x})} = C_q(\mathbf{x}) + j S_q(\mathbf{x}) . \quad (2)$$

Here $\rho_q(\mathbf{x}) = \sqrt{C_q(\mathbf{x})^2 + S_q(\mathbf{x})^2}$ and $\phi_q(\mathbf{x}) = \text{atan2}(S_q(\mathbf{x}), C_q(\mathbf{x}))$ are the amplitude and phase components, and $C_q(\mathbf{x})$ and $S_q(\mathbf{x})$ are the responses of the quadrature filter pair. The $*$ operator depicts convolution. This basic algorithm assumes rectified images. A stereo disparity estimate can thus be obtained from each oriented filter response (at orientation θ_q) by projecting the phase difference on the epipolar line (the horizontal) in the following manner:

$$\delta_q(\mathbf{x}) = \frac{[\phi_q^L(\mathbf{x}) - \phi_q^R(\mathbf{x})]_{2\pi}}{\omega_0 \cos \theta_q} = \frac{[\Delta\phi_q(\mathbf{x})]_{2\pi}}{\omega_0 \cos \theta_q} , \quad (3)$$

where the $[\]_{2\pi}$ operator depicts reduction to the $]-\pi; \pi]$ interval. In this way, multiple disparity estimates are obtained at each location. These estimates can be robustly combined using the median:

$$\delta(\mathbf{x}) = \text{median}_q \delta_q(\mathbf{x}) . \quad (4)$$

A coarse-to-fine control scheme is used to integrate the estimates over the different pyramid levels [2]. A disparity map $\delta^k(\mathbf{x})$ is first computed at the coarsest level k . To be compatible with the next level, it must be upsampled, using an expansion operator \mathcal{X} , and multiplied by two:

$$d^k(\mathbf{x}) = 2 \cdot \mathcal{X}(\delta^k(\mathbf{x})) . \quad (5)$$

This map is then used to reduce the disparity at level $k+1$, by warping the right filter responses before computing the phase difference:

$$\delta_q^{k+1}(\mathbf{x}) = \frac{[\phi_q^L(\mathbf{x}) - \phi_q^R(\mathbf{x}')]_{2\pi}}{\omega_0 \cos \theta_q} + d^k(\mathbf{x}) , \quad (6)$$

where $\mathbf{x}' = (x+d^k(\mathbf{x}), y)^T$. In this way, the remaining disparity is guaranteed to lie within the filter range. This procedure is repeated until the finest level is reached.

3.2 Interpolating Phase Across Orientation

Phase-based methods have been shown to be robust to changes in contrast, scale and orientation [7]. The robustness to orientation is very important in the context of disparity estimation since textures or features on slanted surfaces have a different orientation in the left and right images (orientation disparity). This robustness is not sufficient for the method we propose here, since we need to handle orientation differences that result from an arbitrary epipolar geometry (e.g. due to rotations around the line of sight).

We propose to compensate for these orientation differences by changing the read-out of the filter responses. Instead of computing phase differences from identically-oriented filters in the left and right images (such as in Eq. (3)) we

propose to *shift* the right filter responses across orientation. Since the filterbank only covers a fixed set of orientations, we also introduce a simple mechanism that allows for continuous shifts. For an arbitrary orientation disparity, we linearly interpolate the filter responses using the responses at the nearest filter orientations. Care must be taken here to correctly handle the sign of the sine-filter.

To get the response at an arbitrary angle θ^* , we first determine the corresponding and nearest neighbor filter indices. This implies discretizing the angle to a multiple of π/K and reducing it to the range of the filterbank $[0, \pi-\pi/K]$:

$$q^* = \left(\frac{\theta^*}{\pi/K} \right) \bmod K, \quad (7)$$

$$q^- = \left\lfloor \frac{\theta^*}{\pi/K} \right\rfloor \bmod K, \quad (8)$$

$$q^+ = \left(\left\lfloor \frac{\theta^*}{\pi/K} \right\rfloor + 1 \right) \bmod K. \quad (9)$$

A standard linear interpolation is then used to determine the responses. Note that the filterbank only covers half the unit circle. The cosine-filters are symmetrical, which means that responses at orientations θ and $\theta+\pi$ are identical. We can thus simply ignore the wrap-around when combining the responses:

$$C_{q^*} = (1 - q^f) C_{q^-} + q^f C_{q^+}, \quad (10)$$

where $q^f = q^* - q^-$. The sine-filters however are not symmetrical. When responses are requested at an angle θ outside the filterbank's range, the sign of the response at $\theta-\pi$ needs to be changed. Additionally, in case of wrap-around, q^+ is the index of the filter at angle 0 and the sign of its response has to be changed as well:

$$S_{q^*} = \text{sign}(\pi - \theta^*) \begin{cases} (1 - q^f) S_{q^-} + q^f S_{q^+} & (q^+ \neq 0) \\ (1 - q^f) S_{q^-} - q^f S_{q^+} & (q^+ = 0) \end{cases}. \quad (11)$$

The complex response then equals: $Q_{q^*} = C_{q^*} + j S_{q^*} = \rho_{q^*} e^{j \phi_{q^*}}$.

We next present a simple experiment to show the improvements that result from this interpolation. We compute phase differences, $\Delta\phi$, at corresponding locations of the *venus* image (see [16] and Fig. 3) and 2D rotated versions thereof. Since we use corresponding locations, the $\Delta\phi$'s are expected to be equal to zero. The standard deviation of the $\Delta\phi$'s for all locations and filter orientations are shown in Fig. 1 for different image rotations. For a uniform distribution between $-\pi$ and π (ignoring periodicity) this standard deviation equals $\sqrt{(2\pi)^2/12} \approx 1.8$. Without correction (dotted line), the standard deviation reaches this level at about 45° . It is clear that simply selecting the nearest filter response (dashed line) improves the results, but not as much as the proposed linear interpolation mechanism (solid line). Note in particular how small the differences are between the peaks and dips for the interpolation method. At the dips the image rotation is a multiple of a filterbank orientation and no interpolation is required. This can be considered as a baseline performance. The different baseline performances at different angles are due to the different precision that can be obtained with different filters (e.g. diagonal filters are less accurate than horizontal or vertical).

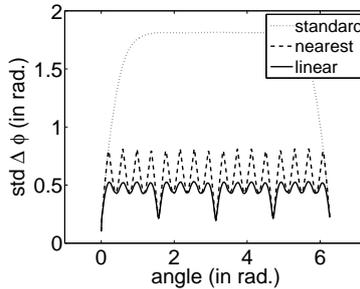


Figure 1: Standard deviation of the phase differences between the left *venus* image and 2D rotated versions thereof, considering all filter orientations and corresponding locations. The standard case (dotted) considers the same filter orientation whereas nearest neighbor (dashed) and linear (solid) use the known image rotation to select the corresponding filter orientation.

3.3 Epipolar Geometry Estimation

The estimation of the epipolar geometry is well-understood, and a large number of methods have been proposed [9]. Our goal is not to improve upon these methods, but rather to present a method to refine a rough geometry estimate on the basis of very noisy correspondences with a significant amount of outliers. This initial estimate can be a sensible guess (e.g. assuming a rectified situation), or can be derived from an inaccurate motor signal. The refinement should also not be restricted to particular camera configurations (e.g. assuming fixation, or only allowing for vergence). We only require that the intrinsic calibration of the cameras is known.

Conceptually, the method corrects an erroneous epipolar geometry estimate by applying 3D rotations to the hypothesized left and right camera orientations. This is achieved in practice by warping the images or, more precisely, the filter responses (this avoids the need for re-filtering). The required 3D rotations are derived from vector disparity deviations orthogonal to the currently hypothesized epipolar lines. Vector disparity is similar to optical flow and can be estimated from the phase differences at multiple orientations by allowing each phase difference to constrain the projection of the vector disparity on the filter's orientation:

$$\delta^*(\mathbf{x}) = \operatorname{argmin}_{\delta(\mathbf{x})} \sum_q \left(\frac{[\Delta\phi_q(\mathbf{x})]_{2\pi}}{\omega_0} - [\cos\theta_q, \sin\theta_q] \delta(\mathbf{x}) \right)^2, \quad (12)$$

Using these vector disparities, left and right correspondences \mathbf{x} and \mathbf{x}' can be hypothesized. The error vectors orthogonal to the right image epipolar line connect the right image correspondence to the epipolar line and are equal to:

$$\Delta\mathbf{e}(\mathbf{x}) = \left(\frac{\mathbf{x}'^T \mathbf{F} \mathbf{x}}{(\mathbf{F} \mathbf{x})_1^2 + (\mathbf{F} \mathbf{x})_2^2} \right) [(\mathbf{F} \mathbf{x})_1, (\mathbf{F} \mathbf{x})_2]^T, \quad (13)$$

where \mathbf{x} and \mathbf{x}' are now expressed in *normalized* homogeneous coordinates, and \mathbf{F} is the fundamental (or in this case the essential) matrix. These errors can

be reduced by updating the epipolar geometry estimate. We do this by least-squares fitting either a left ($\Delta\omega$) or right ($\Delta\omega'$) 3D camera rotation to these errors using an approximate linear rotation model [1] (angle-axis representation):

$$\Delta\omega' = \operatorname{argmin}_{\Delta\omega'} \sum_{\mathbf{x}} \left[|\Delta\mathbf{e}(\mathbf{x})| - \left(B(\mathbf{x}')\Delta\omega' \right)^{\text{T}} \frac{\Delta\mathbf{e}(\mathbf{x})}{|\Delta\mathbf{e}(\mathbf{x})|} \right]^2, \quad (14)$$

$$\Delta\omega = \operatorname{argmin}_{\Delta\omega} \sum_{\mathbf{x}} \left[|\Delta\mathbf{e}(\mathbf{x})| + \left(B(\mathbf{x})\Delta\omega \right)^{\text{T}} \frac{\Delta\mathbf{e}(\mathbf{x})}{|\Delta\mathbf{e}(\mathbf{x})|} \right]^2, \quad (15)$$

where

$$B(\mathbf{x}) = \begin{bmatrix} xy & -1 - x^2 & y \\ 1 + y^2 & -xy & -x \end{bmatrix}. \quad (16)$$

Note that only the component of the rotation orthogonal to the epipolar lines is constrained. The effects of these two 3D rotations can be very similar and therefore estimating them simultaneously is very unstable. In that case only the differential effects are considered and this typically results in very large updates to the individual camera matrices. To overcome this we alternate between Eqs. (14) and (15). In this way, the updates are constrained by the actually measured error vectors $\Delta\mathbf{e}$. For consistency, we also only consider the errors in the right image in both cases. On the basis of the estimated 3D rotations, the left and right translation and rotation matrices are then updated as follows:

$$\mathbf{T}_{(i+1)} = \Delta\mathbf{R}^{\text{T}}\mathbf{T}_{(i)}, \quad (17)$$

$$\mathbf{R}_{(i+1)} = \Delta\mathbf{R}^{\text{T}}\mathbf{R}_{(i)}, \quad (18)$$

$$\mathbf{T}'_{(i+1)} = \Delta\mathbf{R}'^{\text{T}}\mathbf{T}'_{(i)}, \quad (19)$$

$$\mathbf{R}'_{(i+1)} = \Delta\mathbf{R}'^{\text{T}}\mathbf{R}'_{(i)}. \quad (20)$$

where $\Delta\mathbf{R} = e^{[\Delta\omega]\times}$ and $\Delta\mathbf{R}' = e^{[\Delta\omega']\times}$. These can then be combined to construct the updated fundamental matrix.

The algorithm proposed here is not guaranteed to converge to the global minimum. However, establishing dense correspondences is more important here than having accurate epipolar geometry estimates. To disambiguate the latter, it may be more sensible to exploit temporal information (dense correspondences from multiple image pairs).

3.4 Combined Stereo and Auto-calibration

In the previous sections we have discussed how large 2D orientation differences between left and right image features can be compensated for and how an initial epipolar geometry estimate can be refined using vector disparity estimates. We next explain how these two steps can be combined and assist each other.

In the following we still consider a one dimensional disparity along the epipolar line and refer to this measure as the *epipolar disparity* $\delta(\mathbf{x})$. It is the shift *along* the epipolar line necessary to establish the correspondence after compensating

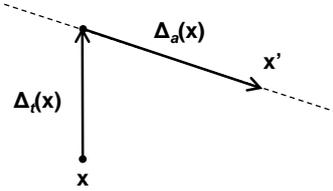


Figure 2: The epipolar disparity for pixel \mathbf{x} is the shift along the (dashed) epipolar line required to reach the corresponding pixel \mathbf{x}' , after compensating for the vertical shift $\Delta_t(\mathbf{x})$ towards the epipolar line. Its magnitude equals that of the vector $\Delta_a(\mathbf{x})$, but its sign is determined by Eq. (22)

for the vertical shift, $\Delta_t(\mathbf{x})$, *towards* the epipolar line. The correspondence is thus established as follows (see Fig. 2):

$$\mathbf{x}' = \mathbf{x} + \Delta_t(\mathbf{x}) + \Delta_a(\mathbf{x}) , \quad (21)$$

where

$$\Delta_a(\mathbf{x}) = \frac{\delta(\mathbf{x})}{\sqrt{(\mathbf{F}\mathbf{x})_1^2 + (\mathbf{F}\mathbf{x})_2^2}} \left[-(\mathbf{F}\mathbf{x})_2, (\mathbf{F}\mathbf{x})_1 \right]^T , \quad (22)$$

$$\Delta_t(\mathbf{x}) = \left[0, \frac{-x(\mathbf{F}\mathbf{x})_1 - (\mathbf{F}\mathbf{x})_3}{(\mathbf{F}\mathbf{x})_2} - y \right]^T . \quad (23)$$

We also compensate for the orientation disparity by shifting and interpolating the right image filter responses by the angular difference between the left and right epipolar lines:

$$\theta_R = \theta_L + \text{atan2}((\mathbf{F}^T \mathbf{x}')_2, (\mathbf{F}^T \mathbf{x}')_1) - \text{atan2}((\mathbf{F}\mathbf{x})_2, (\mathbf{F}\mathbf{x})_1) , \quad (24)$$

with θ_L and θ_R the left and right filter orientations. Algorithm 1 summarizes the proposed method.

The epipolar disparity is initialized to zero and the epipolar geometry to a sensible guess (e.g. rectified) or based on a motor feedback signal. The algorithm¹ then proceeds from coarse scales to fine scales. The previous scale estimate is first doubled and upsampled. An inner loop is then started to update the epipolar geometry on the basis of vector disparity estimates. All position and orientation shifts that result from the disparity and geometry estimates are compensated for by warping the right filter responses. The remaining errors are used to estimate the left and right camera 3D rotations in an alternating fashion. After a few iterations of this inner loop (± 5), the one-dimensional epipolar disparity is updated and the next finer scale is processed.

¹It is worth noting that the combined stereo and auto-calibration could be also achieved without an explicit decision on the direction along which to compute the disparity (epipolar disparity), but exploiting the information embedded in a reliable estimate of the vector disparity.

Algorithm 1 Combined Stereo and Auto-calibration

```
1: initialize epipolar disparity  $\delta^0(\mathbf{x}) = 0$ 
2: initialize T, R, T', R'
3: for k = 1 to n_scales do
4:   initialize epipolar disparity  $\delta^k(\mathbf{x}) = 2\delta^{k-1}(\mathbf{x})$ 
5:   compensate (warp) for epipolar and orientation disparity (Eqs. (21)–(24))
6:   for i = 1 to n_its do
7:     update vector disparity (Eq. (12))
8:     estimate incremental 3D rotation, alternating between the right
       (Eq. (14))
       and left camera (Eq. (15))
9:     update T, R, T', R' and F (Eqs. (17)–(20))
10:    compensate (warp) for epipolar and orientation disparity (Eqs. (21)–
      (24))
11:   end for
12:   update epipolar disparity (Eq. (4))
13: end for
14: return epipolar disparity and epipolar geometry
```

3.5 Results

In this section we evaluate the proposed method on a synthetic image pairs in terms of the quality of the epipolar disparity estimates. We compare the proposed method to the standard algorithm presented in Section 3.1.

In this example we have constructed an unrectified synthetic image pair by warping the left and right images of the *venus* stereo pair according to arbitrary 3D rotations of the left and right cameras. An anaglyph of the two images is shown in Fig. 3(A). The left image is in the red channel, and the right image in the green and blue channels. It can be seen that the images differ in terms of horizontal and vertical shifts, rotations and stereo disparity. The ground truth epipolar disparity is shown in Fig. 3(B).

Both the standard and proposed stereo algorithm were applied to this image pair using five scales. The results obtained with the standard algorithm are shown in Fig. 3(C). A simple left/right consistency check has been performed here to remove unreliable estimates. If the difference is more than one pixel, the estimate is considered unreliable and removed. The same procedure can be applied to the proposed algorithm. In the right-to-left stage, the estimated geometry can be re-used. The results obtained with the proposed method are shown in Fig. 3(D). The method was initialized assuming a rectified situation. Within each scale, the epipolar geometry estimation typically converges after five updates. Therefore five internal iterations were performed here. Clearly, a much larger number of consistent estimates are found than in (C), and the estimates closely resemble the ground-truth epipolar disparity.

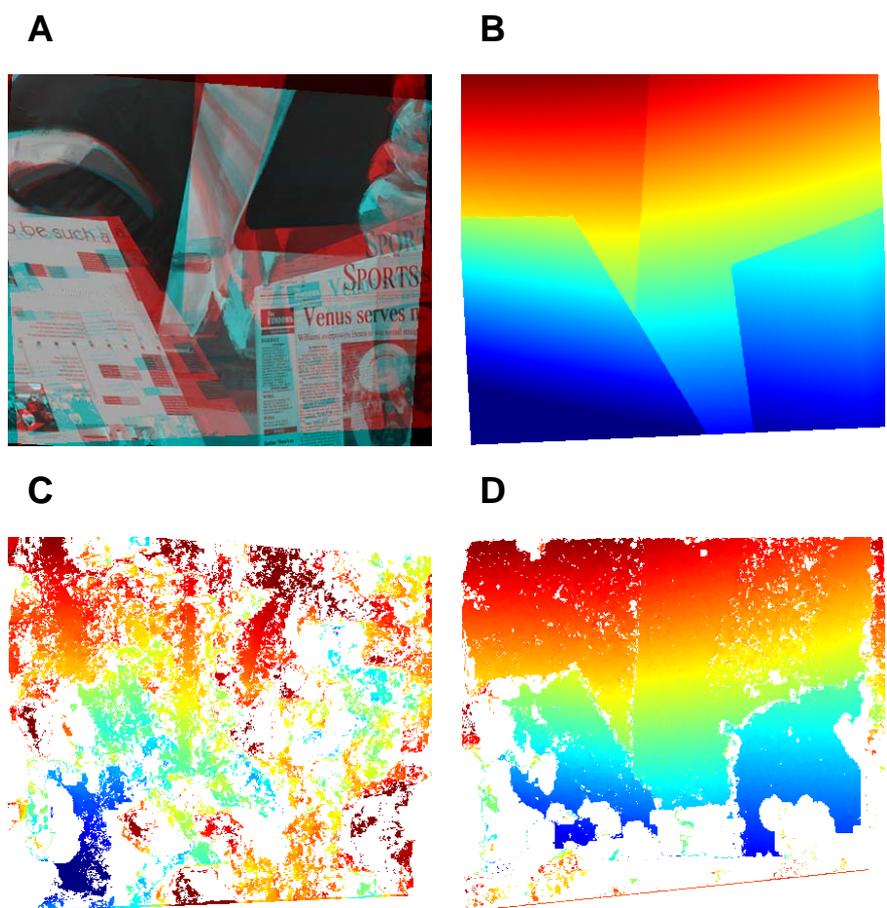


Figure 3: Ground truth epipolar disparity (B) and consistent estimates obtained with the standard (see Eq. 4) (C) and proposed (D) method on a synthetically generated image pair (A).

4 Auto-calibration in the Cortical Domain

We now present an extension of the algorithm that also includes a space-variant mapping from the retinal to the cortical domain using the log-polar transformation [17]. The filtering and correspondence estimation steps are now performed in the cortical rather than in the retinal domain [18]. However, the errors with respect to the (hypothesized) epipolar geometry, the geometry updates, and the corrections (from the estimated correspondences to the hypothesized epipolar lines) are still computed/performed in the retinal domain.

4.1 Central Blind-spot Model

Many mappings from retinal to cortical domain have been proposed, mainly differing in the way they deal with the singularity in the center. We use the Central blind-spot (CBS) model [4]. Design rules for this model can be found in [19].

According to the CBS-model, the forward mapping from cortical to retinal coordinates is defined as:

$$(\rho, \theta) \triangleq (\sqrt{x^2 + y^2}, \arctan -\frac{y}{x}) \quad (25)$$

$$(\xi, \eta) \triangleq (\log_a \left(\frac{\rho}{\rho_0} \right), \theta), \quad (26)$$

with ρ_0 the size of the blind spot. This results in the following inverse mapping:

$$x = a^\xi \rho_o \cos \eta \quad (27)$$

$$y = -a^\xi \rho_o \sin \eta. \quad (28)$$

The integer-valued coordinates (for discrete images) are $(u, v) = (\lfloor \xi \rfloor, \lfloor q \cdot \theta \rfloor)$, with $q = \frac{S}{2\pi}$ for an image consisting of R rings, $u \in [1, R]$, and S sectors, $v \in [1, S]$. The logarithmic factor, a , is determined on the basis of the number of rings, R , and the size of the blind spot: $a = \exp \left(\log \left(\frac{\rho_{\max}}{\rho_0} \right) / R \right)$. The number of sectors is chosen so that the receptive fields have an aspect ratio approximately equal to one: $S = \text{round} \left(\frac{2\pi}{a-1} \right)$. The mapping is thus completely determined on the basis of the size of the Cartesian image, the number of rings R and the blind spot radius ρ_0 .

An example transformation is shown in Fig. 4. The original (retinal) image is shown in Fig. 4(A), and the transformed (cortical) image in Fig. 4(B). The pixel size is the same in both images to show the compression factor achieved. Figure 4(C) contains the retinal image obtained by transforming the cortical image (B) back to the retinal domain using interpolation. This makes it clear how the precision is retained in the fovea (except of course in the blind spot), and gradually reduces towards the periphery.

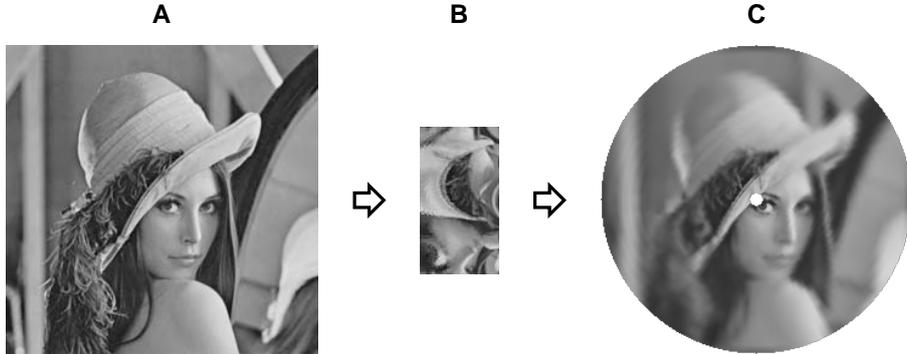


Figure 4: Example illustrating the transformation of an image from retinal (A) to cortical (B) coordinates using the central blind-spot model. The size of pixels is the same in both images. Image (C) contains the cortical image mapped back to the retinal domain.

4.2 Cortical Correspondence Estimation

As shown in [18], the same technique as presented in Section 3.1 can be applied to estimate correspondences in the cortical domain with good precision. This means applying the same filterbank directly to the cortical domain (which saves considerable computational resources), and estimating correspondences by integrating the oriented phase differences into a vector disparity estimate (Eq. 12).

The precision is improved by means of some minor modifications to deal with the periodicity of the η -domain during the filtering and warping operations.

Due to the much smaller image size, we use a smaller number of scales in the coarse-to-fine estimation (only three in the remainder).

4.3 Modified Algorithm

We now discuss the modifications required to apply Algorithm 1 to the cortical domain. The modified algorithm is summarized in Algorithm 2.

This algorithm only differs from the retinal domain algorithm in that the filtering and correspondence estimation are done in the cortical domain. The updates to the epipolar geometry, and the corrections of the estimated vector disparities to the current epipolar geometry estimates are still performed in the retinal domain. This does require intermediate coordinate transformations. To transform vectors between the two domains, the transformation equations from Section 4.1 are simply applied to the vectors' start- and endpoints.

One other difference between the two algorithms is that we do not (yet) compensate for orientation differences in the cortical domain.

Algorithm 2 Combined Stereo and Auto-calibration in the Cortical Domain

```
1: initialize cortical vector disparity to zero
2: initialize T, R, T', R'
3: for k = 1 to n_scales do
4:   double the previous scale cortical vector disparity
5:   compensate (warp) for cortical vector disparity
6:   update cortical vector disparity
7:   compute retinal orthogonal epipolar deviations
8:   for i = 1 to n_its do
9:     estimate incremental 3D rotation, alternating between right and left
       camera
10:    update T, R, T', R' and F
11:   end for
12:   project retinal vector disparity on updated epipolar lines
13:   transform to cortical vector disparity (using vector endpoints)
14: end for
15: return cortical vector disparity and epipolar geometry
```

5 Results

We again use the *venus* stereo pair to construct two different scenarios, and examine the improvements in disparity and geometry estimation with the proposed algorithm in the cortical domain. In both scenarios, we use three scales and initialize the geometry to a rectified situation (horizontal epipolar lines).

The first scenario is concerned with a camera setup close to vergence (in the image center), but disturbed by small inaccuracies in the hypothesized camera geometry (rectified). The anaglyph for this situation is shown in Fig. 5(A). Note that the disturbances result in epipolar lines that are clearly not horizontal (see e.g. how the ‘V’ of Venus has shifted upwards). The disturbances involve 3D rotations along all axes. The magnitude of the retinal ground truth vector disparity is shown in Fig. 5(B), which confirms that the center is close to vergence. The cortical image pair is shown in Fig. 5(C,D), with cortical ground truth vector disparity in Fig. 5(E). The results obtained with and without the proposed auto-calibration technique are shown in Fig. 5(F) and (G) respectively. Figure 5(G) was obtained by computing vector disparity directly in the cortical domain using a coarse-to-fine refinement procedure (similar to an optical flow algorithm). Note how the auto-calibration greatly improves the results and more closely resembles the ground-truth, particularly in the outer regions of the fovea (the central columns in Fig. 5C–G). The improved estimates there can then be used to further refine vergence. The mean and standard deviation of the errors (vector differences w.r.t. ground-truth) for the two methods can be found in Table 1 and confirm this improvement.

We also considered a very different scenario, far away from vergence and with large errors in the hypothesized geometry (again rectified). The anaglyph for this situation is shown in Fig. 6(A). It is clear from the magnitude of the retinal ground truth vector disparity in Fig. 6(B) and the cortical images in Fig. 6(C,D) that the fovea contains very different image parts and that the vector disparities

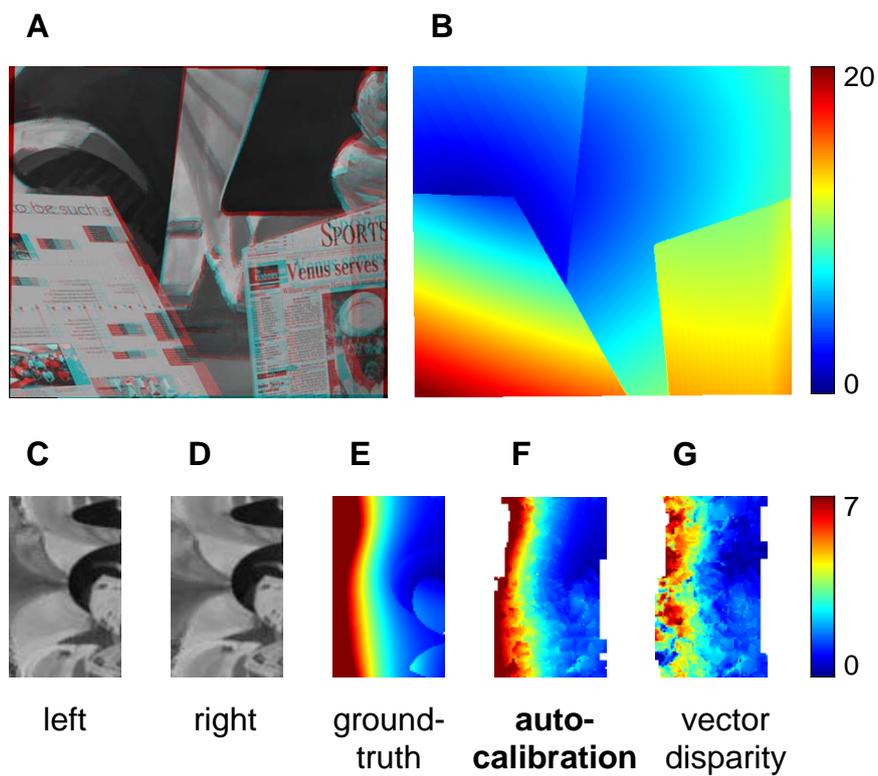


Figure 5: Improvements of auto-calibration over vector disparity in a scenario close to vergence with small errors in the geometry estimates. (A) Left and right retinal image anaglyph showing non-horizontal epipolar lines. (B) Magnitude of retinal ground truth vector disparity. (C) Left and (D) right cortical images. (E) Magnitude of cortical ground truth vector disparity. Magnitude of cortical vector disparity estimated with (F) and without (G) auto-calibration.

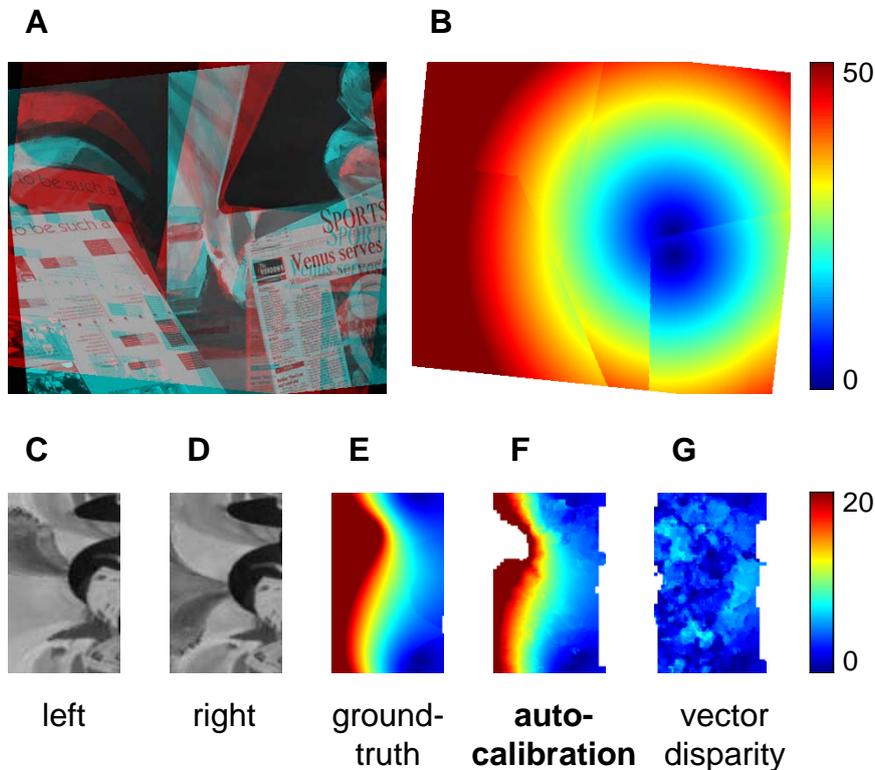


Figure 6: Improvements of auto-calibration over vector disparity in a scenario away from vergence with large errors in the geometry estimates. (A) Left and right retinal image anaglyph showing non-horizontal epipolar lines. (B) Magnitude of retinal ground truth vector disparity. (C) Left and (D) right cortical images. (E) Magnitude of cortical ground truth vector disparity. Magnitude of cortical vector disparity estimated with (F) and without (G) auto-calibration.

are much greater everywhere (and again clearly not horizontal as in a rectified situation) than in the previous scenario. As before, the results obtained with and without the proposed auto-calibration technique are shown in Fig. 6(F) and (G) respectively. Again we see that the proposed method greatly improves the results. This is also measured quantitatively in Table 1. We did use the prior knowledge about the lack of vergence in this situation, and did not use orthogonal disturbances in the fovea (step 7 in Algorithm 2) to update the geometry. Instead we only relied on peripheral errors here. Using foveal errors worsened the results but did not make the algorithm fail.

To demonstrate that the algorithm is capable of making large corrections to a hypothesized camera geometry, we also show the recovered epipolar geometry for the second scenario in Fig. 7. This figure shows corresponding epipolar lines for a few selected key points. The blue points in the left image (A) correspond to the epipolar lines in the right image (B) and vice versa. Note that very large changes have been made considering that the algorithm started with a rectified

scenario	auto-calibration		vector disparity	
	mean	std	mean	std
close to vergence	1.03	1.40	1.70	2.44
away from vergence	3.43	4.28	10.72	10.20

Table 1: Mean and standard deviation (in pixels) of the magnitude of the vector differences between ground truth and estimated vector disparity

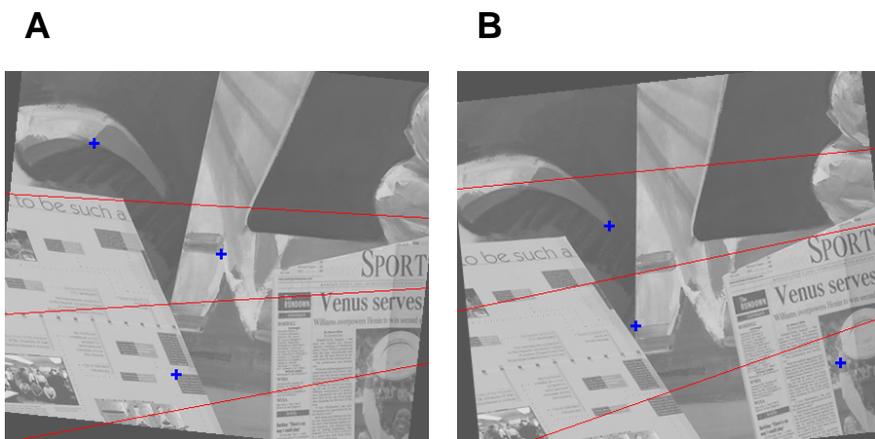


Figure 7: Recovered epipolar geometry for the scenario of Fig. 6. Red epipolar lines in the right image (B) correspond to the blue keypoints in the left image (A) and vice-versa.

situation, and that the epipolar lines are quite accurate (even in the fovea).

6 Future Extensions

The next steps will first require modification of the procedures to the population-based methods for disparity estimation [5]. These are conceptually very similar to the phase-difference approach from Section 3.1, but allow the application of very specific (gain) modulation procedures. These gain modulation mechanisms will be used to transform the responses from eye- to head-centered coordinates. In addition, the population responses also allow for a closer interaction with the vergence mechanisms developed in Task 2.1.

7 Package Description

A Matlab package implementing the methods described in this report is available on request to the consortium. This package contains a wide variety of

supporting functions that perform Gabor filtering, correspondence estimation based on phase differences, the log polar mapping using the central blind-spot model (contributed by UGE), etc. The main functionality discussed in this report is provided through the following two main functions:

```
function [Dv,cal] = ret_autocalib(IL,Ir,cal,n_scales)
```

```
% [DV,CAL] = RET_AUTOCALIB(IL,IR,CAL,N_SCALES) performs
% simultaneous disparity estimation and auto-calibration using
% images in the retinal domain
%
% The function accepts as input:
%
% IL      : the left retinal image
% IR      : the right retinal image
% CAL     : structure containing the initial calibration
%           information
%           CAL.KL : left calibration matrix (referential)
%           CAL.KR : right calibration matrix
%           CAL.RL : left camera rotation
%           CAL.TL : left camera translation
%           CAL.RR : right camera rotation
%           CAL.TR : right camera translation
% N_SCALES : number of scales used in the image pyramid
%           (default = 4)
%
% and returns:
%
% DV      : the vector disparity (complex numbers)
% CAL     : the updated calibration structure
%
```

```
function [Dv,cal] = cort_autocalib(IL,Ir,cal,n_scales,bsm,fovea)
```

```
% [DV,CAL] = CORT_AUTOCALIB(IL,IR,CAL,N_SCALES,BSM,FOVEA) performs
% simultaneous disparity estimation and auto-calibration using
% images in the cortical domain
%
% The function accepts as input:
%
% IL      : the left cortical image
% IR      : the right cortical image
% CAL     : structure containing the initial calibration
%           information
%           CAL.KL : left calibration matrix (referential)
%           CAL.KR : right calibration matrix
%           CAL.RL : left camera rotation
%           CAL.TL : left camera translation
%           CAL.RR : right camera rotation
```

```

%           CAL.TR : right camera translation
% N_SCALES : number of scales used in the image pyramid
%           (default = 3)
% BSM      : structure defining the central blind-spot model
%           BSM.RR0 : retinal row coordinate of the
%                   image center
%           BSM.CC0 : retinal column coordinate of the
%                   image center
%           BSM.R0  : size of the blind spot
%           BSM.A   : logarithmic factor
%           BSM.S   : number of sectors
% FOVEA    : foveal region to exclude when updating geometry
%
% and returns:
%
% DV       : the vector disparity (complex numbers)
% CAL      : the updated calibration structure
%

```

In addition, a number of examples are provided to illustrate the intended use of the package.

References

- [1] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):384–401, 1985.
- [2] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *European Conference on Computer Vision*, pages 237–252, 1992.
- [3] M. Björkman and J.-O. Eklundh. A real-time system for epipolar geometry and ego-motion estimation. In *CVPR*, volume 2, pages 506–513, 2000.
- [4] C. Capurro, F. Panerai, and G. Sandini. Dynamic vergence using log-polar images. *International Journal of Computer Vision*, 24(1):79–94, 1997.
- [5] M. Chessa, S.P. Sabatini, and F. Solari. A fast joint bioinspired algorithm for optic flow and two-dimensional disparity estimation. In *ICVS*, pages 184–193, 2009.
- [6] C.J. Erkelens and R. van Ee. A computational model of depth perception based on headcentric disparity. *Vision Research*, 38(19):2999–3018, 1998.
- [7] D. J. Fleet and A. D. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5:77–104, 1990.
- [8] X. Gao, R. Kleihorst, P. Meijer, and B. Schueler. Self-rectification and depth estimation of stereo video in a real-time smart camera system.

- In *ACM/IEEE International Conference on Distributed Smart Cameras*, pages 426–433, 2008.
- [9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [10] D.V. Papadimitriou and T.J. Dennis. Epipolar line estimation and rectification for stereo image pairs. *IEEE Transactions on Image Processing*, 5(4):672–676, 1996.
- [11] K. Pauwels and M.M. Van Hulle. Realtime phase-based optical flow on the gpu. In *CVPR Workshop on Computer Vision on the GPU*, 2008.
- [12] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004.
- [13] J.C.A. Read and B.G. Cumming. Does depth perception require vertical-disparity detectors? *Journal of Vision*, 6(12):1323–1355, 2006.
- [14] S.P. Sabatini, G. Gastaldi, F. Solari, J. Diaz, E. Ros, K. Pauwels, M.M. Van Hulle, N. Pugeault, and N. Krüger. Compact and accurate early vision processing in the harmonic space. In *International Conference on Computer Vision Theory and Applications*, pages 213–220, Barcelona, 2007.
- [15] S.P. Sabatini, G. Gastaldi, F. Solari, K. Pauwels, M.M. Van Hulle, J. Diaz, E. Ros, N. Pugeault, and N. Krüger. A compact harmonic code for early vision based on anisotropic frequency channels. *Computer Vision and Image Understanding*, accepted for publication.
- [16] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.
- [17] E.L. Schwartz. Spatial mapping in the primate sensory projection: Analytic structure and relevance to perception. *Biological Cybernetics*, 25(4):181–194, 1977.
- [18] F. Solari, M. Chessa, and S.P. Sabatini. Design strategies for direct multi-scale and multi-orientation visual processing in the log-polar domain. *Pattern Recognition Letters*, submitted.
- [19] V. J. Traver and F. Pla. Log-polar mapping template design: From task-level requirements to geometry parameters. *Image and Vision Computing*, 26(10):1354–1370, 2008.
- [20] M. Vergauwen, M. Pollefeys, and L. Van Gool. A stereo-vision system for support of planetary surface exploration. *Machine Vision and Applications*, 14(1):5–14, 2003.