

University of New Hampshire

University of New Hampshire Scholars' Repository

Doctoral Dissertations

Student Scholarship

Fall 2008

Biologically inspired composite image sensor for deep field target tracking

Pavlo Melnyk

University of New Hampshire, Durham

Follow this and additional works at: <https://scholars.unh.edu/dissertation>

Recommended Citation

Melnyk, Pavlo, "Biologically inspired composite image sensor for deep field target tracking" (2008).

Doctoral Dissertations. 446.

<https://scholars.unh.edu/dissertation/446>

This Dissertation is brought to you for free and open access by the Student Scholarship at University of New Hampshire Scholars' Repository. It has been accepted for inclusion in Doctoral Dissertations by an authorized administrator of University of New Hampshire Scholars' Repository. For more information, please contact Scholarly.Communication@unh.edu.

BIOLOGICALLY INSPIRED COMPOSITE IMAGE SENSOR FOR DEEP FIELD
TARGET TRACKING

BY

PAVLO MELNYK

BS in Electronics, Odessa State Polytechnic University, Ukraine, 1998

MS in Electronics, Odessa State Polytechnic University, Ukraine, 1999

MS in Electrical Engineering, University of New Hampshire, 2003

DISSERTATION

Submitted to the University of New Hampshire
in Partial Fulfillment of
the Requirements for the Degree of

Doctor of Philosophy
in
Electrical Engineering

September, 2008

UMI Number: 3333523

Copyright 2008 by
Melnyk, Pavlo

All rights reserved.

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI[®]

UMI Microform 3333523

Copyright 2008 by ProQuest LLC.

All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

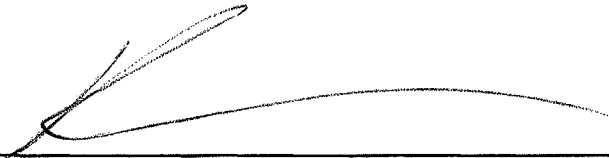
ProQuest LLC
789 E. Eisenhower Parkway
PO Box 1346
Ann Arbor, MI 48106-1346

ALL RIGHTS RESERVED


© 2008

Pavlo Melnyk

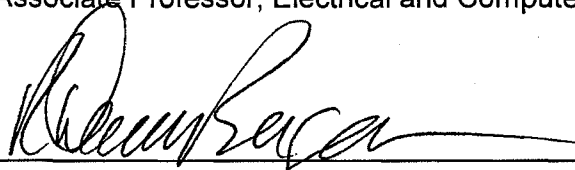
This dissertation has been examined and approved.




Dissertation Director, Richard A. Messner,
Associate Professor, Electrical and Computer Engineering



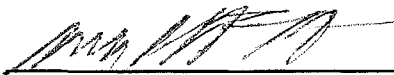
Andrew L. Kun,
Associate Professor, Electrical and Computer Engineering



R. Daniel Bergeron,
Professor, Computer Science



W. Thomas Miller, III,
Professor, Electrical and Computer Engineering



William H. Lenharth,
Associate Research Professor, Electrical and Computer
Engineering

27 MAY 2008
Date

DEDICATION

I would like to dedicate my dissertation to my family: my parents, Borys and Zinayda Melnyk, who always taught me the great values, my wife, Elena Vinokurova, for her love and endless energy she brings in my life, and my younger sister, Olga Melnyk, as a source of inspiration.

ACKNOWLEDGEMENTS

I am grateful to my advisor Dr. Richard Messner for wise mentoring during my graduate studies and research. He encouraged my inventiveness and gave me freedom in work, yet he was there to answer all my questions, and helped me to overcome professional, linguistic and cultural challenges.

I would like to express my appreciation to CATLAB Project54 for sponsoring most of my doctorate program through research assistantship. I would like to acknowledge UNH Graduate School for the Dissertation Year Fellowship. Also, I would like to thank the Department of Electrical and Computer Engineering for supporting me via teaching assistantship.

I would like to thank Dr. Tomas Miller, Dr. Andrew L. Kun, Dr. William H. Lenharth, and Dr. Daniel Bergeron for serving on my defense committee and reviewing my dissertation.

I am grateful to my parents and my wife Elena for their love and support.

Finally, I would like to thank all my friends and colleagues who helped me in my research.

TABLE OF CONTENTS

DEDICATION.....	iv
ACKNOWLEDGEMENTS.....	v
LIST OF FIGURES.....	viii
ABSTRACT.....	x
INTRODUCTION.....	1
I. BACKGROUND.....	8
<i>Biological influence on space-variant sensors</i>	8
Far seeing in vulture's vision.....	10
Image data reduction in the primate retina.....	13
Log-polar retina model.....	15
<i>Properties of log-polar transformation</i>	22
<i>Retina-like sensors</i>	24
II. A NOVEL COMPOSITE SENSOR FOR DEEP-FIELD VISION.....	27
<i>Composite image sensor</i>	27
<i>Composite sensor design</i>	36
<i>Calibration</i>	42
III. ADAPTATION FOR AUTOMOTIVE APPLICATIONS.....	47
<i>Composite log-polar space in automotive applications</i>	47
<i>Vanishing point fixation</i>	53
IV. VEHICLE TRACKING IN COMPOSITE LOG-POLAR SPACE.....	59
<i>On-road vehicle detection overview</i>	60

<i>Speed-based vehicle tracking in log-polar space</i>	61
<i>License plate scan</i>	71
CONCLUSIONS.....	77
APPENDICES.....	81
APPENDIX A. COMPOSITE IMAGE SENSOR PROTOTYPES.....	82
APPENDIX B. COMPOSITE SENSOR DESIGN THEORY.....	86
APPENDIX C. VANISHING POINT FIXATION IN LOG-POLAR SPACE.....	93
APPENDIX D. VANISHING POINT FIXATION IN CARTESIAN SPACE.....	102
REFERENCES.....	109

LIST OF FIGURES

<i>Fig. 1. Research flow.....</i>	<i>3</i>
<i>Fig. 2. Log-polar transformation as a 2D projection of "Log Funnel" 3D space warp.....</i>	<i>5</i>
<i>Fig. 3. Telephoto optics in falconiform eye.....</i>	<i>11</i>
<i>Fig. 4. Classical log-polar model of human eye.....</i>	<i>16</i>
<i>Fig. 5. "Central-blind-spot" log-polar mapping adapted for mobile vision.....</i>	<i>18</i>
<i>Fig. 6. Software log-polar mapper.....</i>	<i>21</i>
<i>Fig. 7. Properties of log-polar transformation.....</i>	<i>23</i>
<i>Fig. 8. Photo element distribution in retina-like sensors.....</i>	<i>25</i>
<i>Fig. 9. 3D log funnel and its 2D projection in the form of log-polar grid.....</i>	<i>28</i>
<i>Fig. 10. Warped space in front of the camera shows how infinite depth of the field is achieved...29</i>	<i>29</i>
<i>Fig. 11. Composite camera model</i>	<i>31</i>
<i>Fig. 12. Composite sensor transforms multiple camera views into a single log-polar image.....</i>	<i>32</i>
<i>Fig. 13. Composite log-polar output in two-camera deep-field sensor.....</i>	<i>34</i>
<i>Fig. 14. N-camera composite log-polar image.....</i>	<i>35</i>
<i>Fig. 15. Foveal and peripheral camera sensor spaces in logarithmic transformation.....</i>	<i>38</i>
<i>Fig. 16. Composite sensor design cycle.....</i>	<i>41</i>
<i>Fig. 17. "Log-polar Chessboard" calibration pattern.....</i>	<i>43</i>
<i>Fig. 18. Composite camera calibration setup</i>	<i>44</i>
<i>Fig. 19. Objectives of composite sensor calibration.....</i>	<i>45</i>
<i>Fig. 20. Two-camera composite sensor for vehicle tracking.....</i>	<i>48</i>
<i>Fig. 21. Road segmentation in log-polar space.....</i>	<i>49</i>
<i>Fig. 22. Composite log-polar image sequence for vehicle tracking.....</i>	<i>50</i>
<i>Fig. 23. Deviations from the vanishing point change road appearance in the log-polar image.....</i>	<i>52</i>
<i>Fig. 24. Vanishing point estimation in log-polar space.....</i>	<i>55</i>
<i>Fig. 25. Vanishing point estimation in Cartesian space.....</i>	<i>57</i>
<i>Fig. 26. Optical flow based vehicle tracking in log-polar space.....</i>	<i>63</i>
<i>Fig. 28. Optical flow based tracking of a single vehicle.....</i>	<i>67</i>
<i>Fig. 29. Optical flow based tracking of a chain of vehicles.....</i>	<i>68</i>

<i>Fig. 30. Optical flow based tracking in night conditions.....</i>	<i>69</i>
<i>Fig. 27. "Lost" and then "found" optical flow vector features.....</i>	<i>70</i>
<i>Fig. 31. License plate capture camera and the composite sensor installed in the test vehicle.....</i>	<i>72</i>
<i>Fig. 32. License plate scan application.....</i>	<i>73</i>
<i>Fig. 33. Deinterlacing via half imaging.....</i>	<i>74</i>
<i>Fig. 34. Third-party license plate recognition software.....</i>	<i>75</i>
<i>Fig. A.35. Composite sensor "Laboratory" prototype.....</i>	<i>83</i>
<i>Fig. A.36. The "field-test" composite sensor prototype and installation inside a vehicle.....</i>	<i>84</i>
<i>Fig. B.37. Decreasing the oversampled region with foveal camera.....</i>	<i>87</i>
<i>Fig. B.38. Primary and secondary oversampling under different input resolutions.....</i>	<i>88</i>
<i>Fig. B.39: Radius of secondary oversampled area as a function of log base.....</i>	<i>89</i>
<i>Fig. B.40. Composite sensor design charts.....</i>	<i>91</i>
<i>Fig. C.41. The vanishing point fixation algorithm.</i>	<i>94</i>
<i>Fig. C.42. Vanishing-point fixation applied to video stream taken in driving simulator.....</i>	<i>99</i>
<i>Fig. C.43. Field tests of the vanishing point fixation algorithm.....</i>	<i>100</i>
<i>Fig. D.44. Vanishing point fixation algorithm in Cartesian space.....</i>	<i>103</i>
<i>Fig. D.45. Vanishing point fixation in day conditions.....</i>	<i>107</i>
<i>Fig. D.46. Vanishing point fixation in night conditions.....</i>	<i>108</i>

ABSTRACT

BIOLOGICALLY INSPIRED COMPOSITE IMAGE SENSOR FOR DEEP FIELD TARGET TRACKING

by

Pavlo Melnyk

University of New Hampshire, September, 2008

The use of nonuniform image sensors in mobile based computer vision applications can be an effective solution when computational burden is problematic. Nonuniform image sensors are still in their infancy and as such have not been fully investigated for their unique qualities nor have they been extensively applied in practice. In this dissertation a system has been developed that can perform vision tasks in both the far field and the near field. In order to accomplish this, a new and novel image sensor system has been developed. Inspired by the biological aspects of the visual systems found in both falcons and primates, a composite multi-camera sensor was constructed. The sensor provides for expandable visual range, excellent depth of field, and produces a single compact output image based on the log-polar retinal-cortical mapping that occurs in primates. This mapping provides for scale and rotational tolerant

processing which, in turn, supports the mitigation of perspective distortion found in strict Cartesian based sensor systems. Furthermore, the scale-tolerant representation of objects moving on trajectories parallel to the sensor's optical axis allows for fast acquisition and tracking of objects moving at high rates of speed. In order to investigate how effective this combination would be for object detection and tracking at both near and far field, the system was tuned for the application of vehicle detection and tracking from a moving platform. Finally, it was shown that the capturing of license plate information in an autonomous fashion could easily be accomplished from the extraction of information contained in the mapped log-polar representation space.

The novel composite log-polar deep-field image sensor opens new horizons for computer vision. This current work demonstrates features that can benefit applications beyond the high-speed vehicle tracking for drivers assistance and license plate capture. Some of the future applications envisioned include obstacle detection for high-speed trains, computer assisted aircraft landing, and computer assisted spacecraft docking.

INTRODUCTION

The utilization of space-variant image sensors in mobile computer vision has not been sufficiently explored [1], [2], [3]. Although both mobile computer vision and space-variant visual sensors have been well developed for constrained machine vision domains, there are very few efforts that leverage such sensors for mobile applications [4]. It is the author's view that the potential of space-variant sensors is underestimated for mobile vision based applications.

Biologically motivated non-uniform image sampling was introduced in machine vision in the 1970's; it is a mathematical model resembling the primate retino-cortical projection [5]. Since this biologically motivated solution adopts the non-uniform distribution of photoreceptor cells found in the eyes of animals, the inherent non-uniform sampling that occurs in biologically based image acquisition offers a compromise between high resolution, wide field of view and a lighter demand for computing power. Because of this feature, retina-like sensors have gained some attention and development in the area of mobile robotic vision where limited computing resources are available and a significant amount of image processing needs to be performed [4].

Current investigation of the applicability of non-uniform sensors to mobile machine vision has been narrowly focused on mobile vision applications that

detect and track moving vehicles on the road from a mobile platform. This emphasizes a family of applications such as, driver assistance [2], autonomous vehicle navigation [6], and security applications (e.g., license plate scanning [7]). At the moment mobile computer vision applications are built around standard uniform visual sensors in several variations: 1) single video camera based, 2) stereo camera based [8], 3) multiple camera systems [9], and 4) data fusion of video and data from other sensors such as microwave radars [10] and laser radars (lidars) [11], [12].

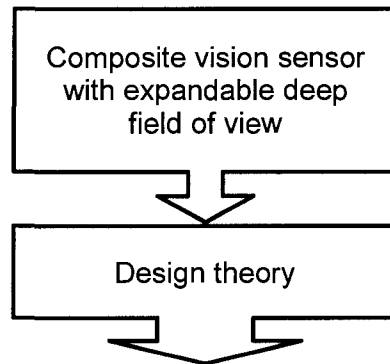
With regard to on-road vehicle detection, different sensors and algorithms are used for three distinct groups of vehicles [1] : 1) overtaking vehicles, 2) close-by vehicles, and 3) midrange/distant vehicles. Surprisingly, detection of approaching vehicles in incoming lanes received little attention. Perhaps the reason for this is the high relative speeds of vehicles in incoming traffic that require early detection and tracking at greater distances, and hence a different more complex kind of visual sensor. This fact became a departing point for the current research which led to the approaches developed. A diagram describing the research flow is shown in Figure 1.

Initial investigation discovered a niche in visual sensors for mobile machine vision: sensors that would have arbitrary deep field of view. The approach taken was the development of a non-uniform visual sensor motivated by studies of two biological based vision systems: 1) birds of prey (e.g., falcons, eagles and hawks [13], [14], [15]) and 2) the primate vision system [5], [16], [17], [18], [19].

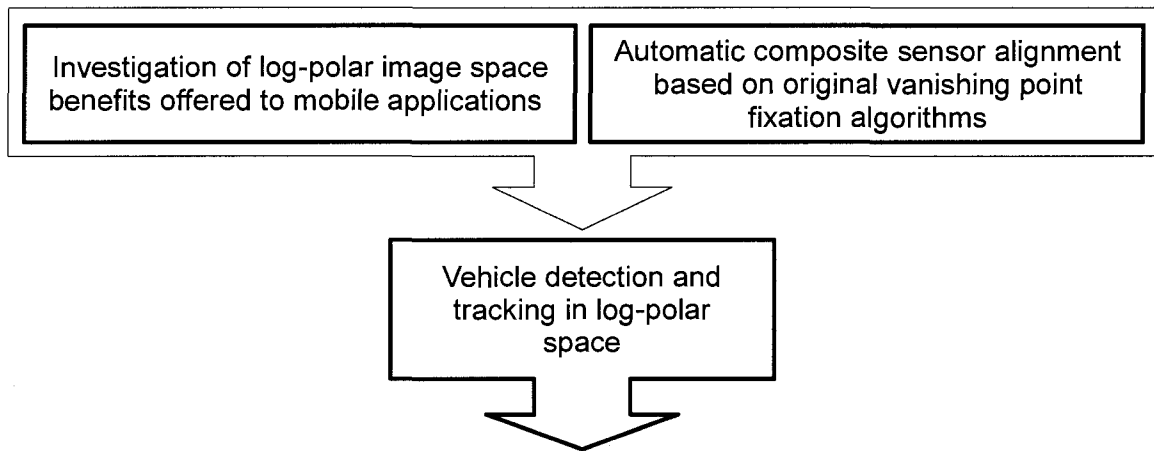
Literature research



Original ideas and research



Mobile application specific



Proof-of-concept application

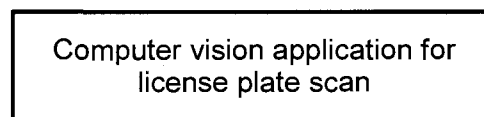
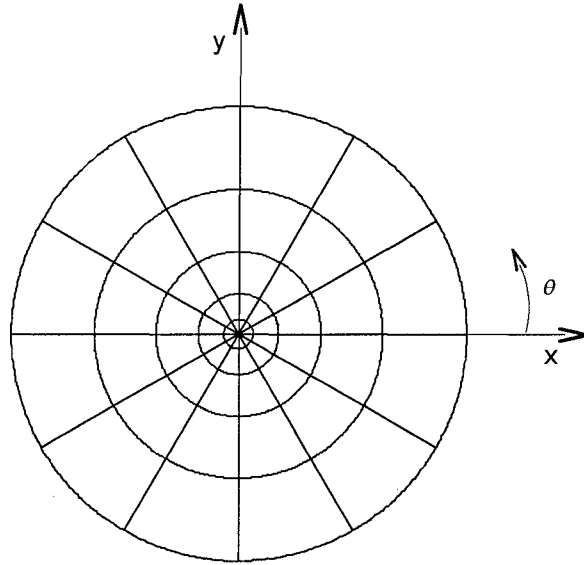


Fig. 1. Research flow

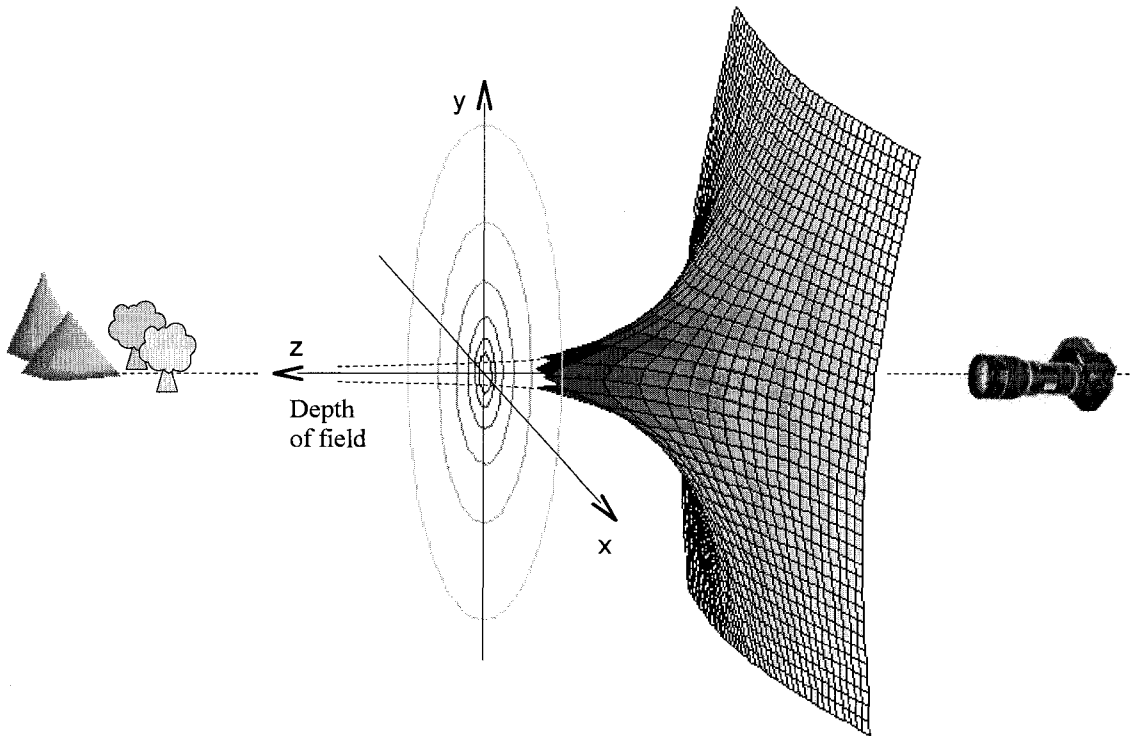
The key idea behind the new visual sensor is a novel paradigm based on a unique interpretation of the well known log-polar transformation where the log-polar grid is viewed as a two dimensional projection of a special three-dimensional surface. As depicted in Figures 2 and 10, this novel idea introduces a virtual three-dimensional surface in the form of a logarithmic funnel. The center axis is coincident with the direction of camera gaze. Now, if a 3D world in front of the observer camera is projected onto the surface of the funnel, this projection allows theoretically infinite depth of field around the center of the funnel (axis Z). Since the projection of the logarithmic funnel on the image sensor plane XY takes a form of a log-polar grid, this means that the log-polar image space mapped from the grid offers a scale invariant image of infinitely deep view. This concept is thoroughly explained in Chapter 2.

This proposed paradigm manifests itself in the creation of an expandable-deep-field-of-view composite visual sensor [20]. To briefly explain its operation, imagine making a panoramic picture with a sequence of smaller pictures captured at different view angles. The pictures are stitched together to form a continuous wide view. The concept of the deep-field-of-view composite visual sensor is similar, but instead of the wide view it creates a continuous deep view by stitching images taken by cameras with different magnification settings.

Moving from the concept to working prototypes, Chapter 2 continues with design theory and a description of the original sensor specific required calibration. Introduction of this novel sensor for deep-field object tracking in a



a) Classical log-polar transformation grid as a model for non-uniform sensor



b) A novel paradigm interprets log-polar grid as a 2D projection of 3D log funnel

Fig. 2. Log-polar transformation as a 2D projection of "Log Funnel" 3D space warp

mobile environment poses new challenges and gives rise to a chain of original design solutions. In general, the introduced approaches are aimed to make object tracking more efficient and also faster. The idea is to implement object tracking in log-polar space with much less computation load than it would take for the same task performed in a higher resolution Cartesian space. Having significantly less computations is important for the composite image sensor because of its expandable field of view. "Expandable" means that the amount of image data increases with every additional camera placed in the sensor. Therefore, computation efficiency and infinite field of view have to be present together or else the system's performance will be limited.

How to take advantage of log-polar image space in order to simplify vehicle tracking is described in Chapter 3. Also a very important camera alignment issue is solved on the preprocessing level. Two algorithms for aligning the sensor's optical axis to the host vehicle trajectory were developed in order to compensate for vibration and trajectory deviations (micro turns) [21]. Both of these algorithms were tested in the field and the one with better performance was chosen for the composite sensor alignment module of the final image processing pipeline.

The actual vehicle tracking algorithm was developed and successfully tested in the field. This is discussed in Chapter 4. It was hypothesized that utilization of a log-polar space-variant image sensor would open new possibilities for mobile computer vision. For instance, the proposed composite deep-field image sensor makes possible a more computationally simple method for the task of early

detection and tracking of high speed vehicles in incoming traffic. This has not been successfully accomplished with a conventional uniform image sensor. It was proposed that a good proof of concept application would be the capture and reading of license plates from incoming traffic. The log-polar based vehicle tracking engine provides a trigger for the acquisition and subsequent reading of the license plate information from approaching vehicles. Acquired license plate snapshots are stored on disk and then fed into the third party license plate recognition software [22]. This application was fully implemented and demonstrated reliable license plate acquisition.

The appendices present important implementation details. Appendix one and two provide complementary material for Chapter 2. They contain information about the composite sensor design theory and the construction of several prototypes: 1) a laboratory prototype, designed for tests in the driving simulator and 2) a more rigid and compact composite sensor for field tests. Appendices three and four provide complementary material for Chapter 3. They contain detailed descriptions of the log-polar based and Cartesian based algorithms for vanishing point fixation that are used for composite sensor alignment.

In terms of tailoring the proposed computer vision system to be implemented in an embedded mobile platform, this research is based on previous work in the field of video evidence recording for law enforcement [23], [24]. Additionally, leveraging of the log-polar model in the current research was inspired by the dissertation of Messner [25] and it is accompanied by a parallel work in the area of biologically inspired active vision conducted by Vidacic [26].

CHAPTER I

BACKGROUND

Biological influence on space-variant sensors

Nature has always been a source of inspiration for new ideas due to the efficient and elegant way that nature evolves biological systems. Many researchers thus contemplate nature and try and leverage and model biological systems to solve some interesting problems. This is especially true for the area of synthetic vision. There is a significant interest in creating computer vision systems that would “see” and interpret the environment in a manner similar to humans. This trend is natural. The ultimate goal is enabling machines to work without operator interaction or supervision for the benefit of mankind.

This “human factor” thought spawned an aggregation of biologically inspired machine vision and image processing research and development. Biologically inspired synthetic vision is multidisciplinary; it is based on parallel studies in the fields of biology, image processing and synthetic vision. A lot of attention and development in biologically inspired synthetic vision is based on studies of the human visual system. These studies involved studies on eye geometry and optics, retina structure on the cell level, and the retino-cortical projection [5], [17], [16], [18], [27], [28]. Some findings in biological research were interpreted into

mathematical and engineering models and applied into synthetic vision problems [5], [29].

Because of its complexity, the human vision system was not the only source of inspiration in biologically motivated machine vision. Other biological vision systems have also been investigated due, in some cases, to the relative simplicity and convenient replication that could be done in silicon. For instance, other visual systems being researched and modeled for robotic vision have included more primitive visual systems such as those possessed by insects [30] and crabs [31].

One of the paradigms adopted from biological vision systems is the non-uniform distribution of receptors in the eyes of animals. Non-uniform sampling in the image acquisition front end offers a compromise between high resolution, wide field of view and smaller amount of visual data. The latter is proportional to the computation power required for image processing tasks. These three components are essential requirements for many machine vision applications, yet they conflict with each other in cases which use an image sensor with uniform pixel density [29]. Indeed, in order to increase the field of view given a standard uniformly distributed camera sensor, it is necessary to tradeoff the effective resolution. However, if high resolution is to be preserved along with a wide field of view, the amount of visual data and affiliated computational resources grow often times exponentially.

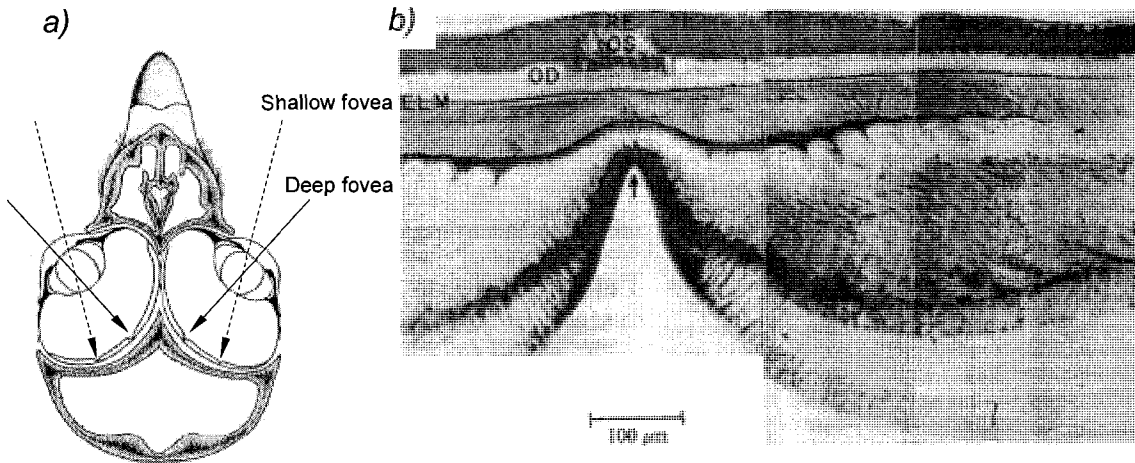
Far seeing in vulture's vision

Raptors, like falcons, hawks and eagles, can see their prey miles away and simultaneously use their vision for navigating in near field. These raptors have several times better resolution power than humans despite a smaller eye size and photoreceptor density [13], [14]. Their special eye structure allows for simultaneous perception of distant and near objects and scenes. If an image sensor with similar properties is constructed, it would be useful for synthetic vision applications where high acuity in the far field of view and wide view angle in the near field of view are of equal importance. Possible applications may include deep field object tracking and navigation for both airborne and ground based unmanned vehicles.

The study of a raptors' eye structure shows that unlike humans, they have two foveae in each eye. One is dedicated to the near visual field and the other is dedicated to the far visual field. The fovea for near field processing is the shallow fovea with straight forward line of sight that provides acute vision for navigation. The other fovea is the deep field fovea that has a line of sight that points approximately 45° to the side of the head axis (see Figure 3a). The deep fovea has better acuity and a higher photoreceptor density, and is used for locating remote targets. The interesting consequence of the deep fovea being located at 45° angle is that raptors move in spiral paths while approaching their prey [14].

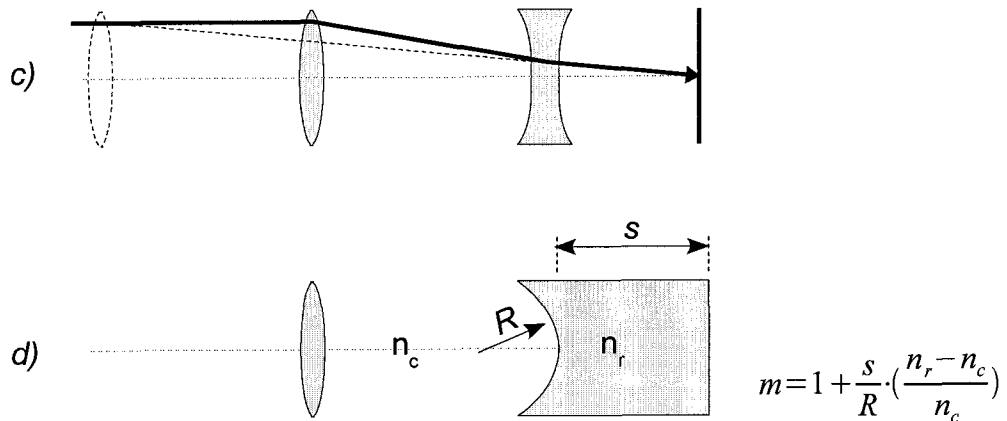
In the falcon's eye there are also two separate optical channels for each of the foveae. The pupil lens is the common optical element for both foveae.

Avian eye and cross-section of the retina through the deep fovea



- a) Section of head of a hawk showing line of sight of the deep and shallow foveae
 b) Thick section of the retina in region of deep fovea. Arrow shows the concave region of the foveal pit thought to act as diverging optical element to project magnified element on receptors at center of fovea. ELM – external limiting membrane; OD – oil droplets; OS – outer segments of cone photoreceptor cells; PE – pigment epithelium

Schematic of the telephoto lens



- c) Classical design
 d) Equivalent system from the avian eye
 m – magnification of the system
 n_r, n_c – refractive indexes of the medium
 s – distance from the apex of the spherical surface to the image plane.
 R – radius of the curvature of the concave spherical surface

Adapted by permission from Macmillan Publishers Ltd: [Nature 275, 127 - 129, "Telephoto lens system of falconiform eyes", ALLAN W. SNYDER & WILLIAM H. MILLER, 1978], copyright © 1978

Fig. 3. Telephoto optics in falconiform eye

However, in addition to the pupil lens, it has been shown that the spherical depression in the deep fovea (Figure 3b) acts as a negative lens component similar to that of the telephoto lens system found in binoculars [13]. The optics equivalent to the deep fovea structure [13], [15] is shown in Figure 3c. The purpose of a negative concave lens is to increase the focal length of the combination of lenses. This preserves the overall smaller dimensions of the system more so than an equivalent single lens system would have. The equivalent optical model of the foveal pit is shown on Figure 3d.

The idea inspired by the falconiform eye study is that two separate optical channels can be built into a single image sensor: one channel for far visual field and another for near visual field. It would be helpful to have some knowledge about the retino-cortical projection from the eye to the brain of these birds. However, there is no such information in the literature and it is impossible to tell how the images from two foveae are mapped in the bird brain and how they are treated. The alternative source of information about the retino-cortical projection found in biological systems comes from more comprehensively studied visual system of primates. It is referred next for mathematical model of retino-cortical projection.

Image data reduction in the primate retina

From the standpoint of synthetic vision one of the most interesting biological vision mechanisms is the ability to reduce the amount of visual data that must be processed. For instance, in the human visual system data reduction occurs through the whole pathway traveling from the receptors of the retina to the striate cortex of the brain. This retina-cortical pathway is thought to contain three distinct stages. First, the non-uniform photoreceptor distribution of cones in the eye forms a small region of high acuity with a wide area of lower resolution surrounding it. The second stage of data reduction occurs inside the retina layers itself, which is evident from the fact that the number of photoreceptors is much larger than the number of ganglion cells axons (i.e. data channels) in the optic nerve. Finally, the third stage takes place in the visual area of the cortex (Area V1) [16], [17], [18].

Studies made on the primate visual system uncovered a mechanism of visual data reduction in the retina due to non-uniform photoreceptor distribution [17], [18]. This distribution consists of a small area of high acuity in the center of retina called the fovea and a wide periphery with a much smaller photoreceptor density (see Figure 4). This allows for the coexistence of a wide field of view with a high resolution at the center of view and a lower acuity moving away from the center of view reducing the amount of data for processing. This non-uniform organization provides a tradeoff between wide area of view and a high resolution at the central viewpoint. Naturally, this is only one of the mechanisms of data

reduction in the overall system. Biological systems are far more complex because both data reduction and reinterpretation of the data happens also in the visual pathway and visual area of the cortex [16], [17], [18]. The complete picture of how the human vision works is neither completely explored nor understood. However, having said this, the eye structure itself suggests possible ways of building non-uniform image sensors that maybe effective for real-time synthetic vision applications [29], [32].

Structurally, the retina has three layers built of six types of cells. Of interest here is the topmost layer consisting of photoreceptors, the cells that transform light into electrical impulses [16]. There are two types of photoreceptors: rods and cones; responsible for different light conditions. Rods are responsive to low intensity light; they are very sensitive but do not differentiate color. Cones are responsible for high illumination conditions, high acuity vision and color sensitivity. There are three different types of cones dedicated to red, green and blue spectra. The retina can be roughly divided into two regions characterized by photoreceptor density. In the center of retina, there is a small depression with high cone concentration. This depression, which covers about 5.2° of visual angle, is called fovea. The distribution diagram for cone photoreceptors as shown in Figure 4 suggests the highest density for cones within 1° - 2° of visual angle [18], [33]. The periphery, which occupies the rest of retina around the fovea, consists of a lower density of cones as one moves away from the fovea center.

As was already mentioned, the non-uniform photoreceptor distribution is the

very first mechanism for data reduction in the human visual system. Although the human visual system reduces data further along the visual pathway we are only concerned with the data reduction accomplished through the non-uniform sampling at the retina. For additional information on the human visual system the reader is referred to the bibliography related to this section [17], [18], [16]. This research focuses on a retina-like sensor design, and in particular a log-polar based model of non-uniform image acquisition [19], [29].

Log-polar retina model

Discovered as a mathematical model resembling the primate retino-cortical projection [5], [27] the log-polar image representation has been used for several decades in computer vision. The log-polar transformation is interesting because of its properties. It is a conformal mapping and under certain conditions is scale and rotation tolerant. Also, its non-uniform sampling can be used for data reduction. If a Cartesian sampled image is resampled by a log-polar mapping the resulting resampled image can be much smaller than the Cartesian sampled image. This property allows for the implementation of various geometry-specific and often real-time image and video processing. Some synthetic vision application areas where the log-polar transformation has been exploited include: optical flow [6], time-to-collision and depth from motion [34], [35]; active vision systems [36], [37]; target tracking [38]; spacecraft docking [39]; pattern

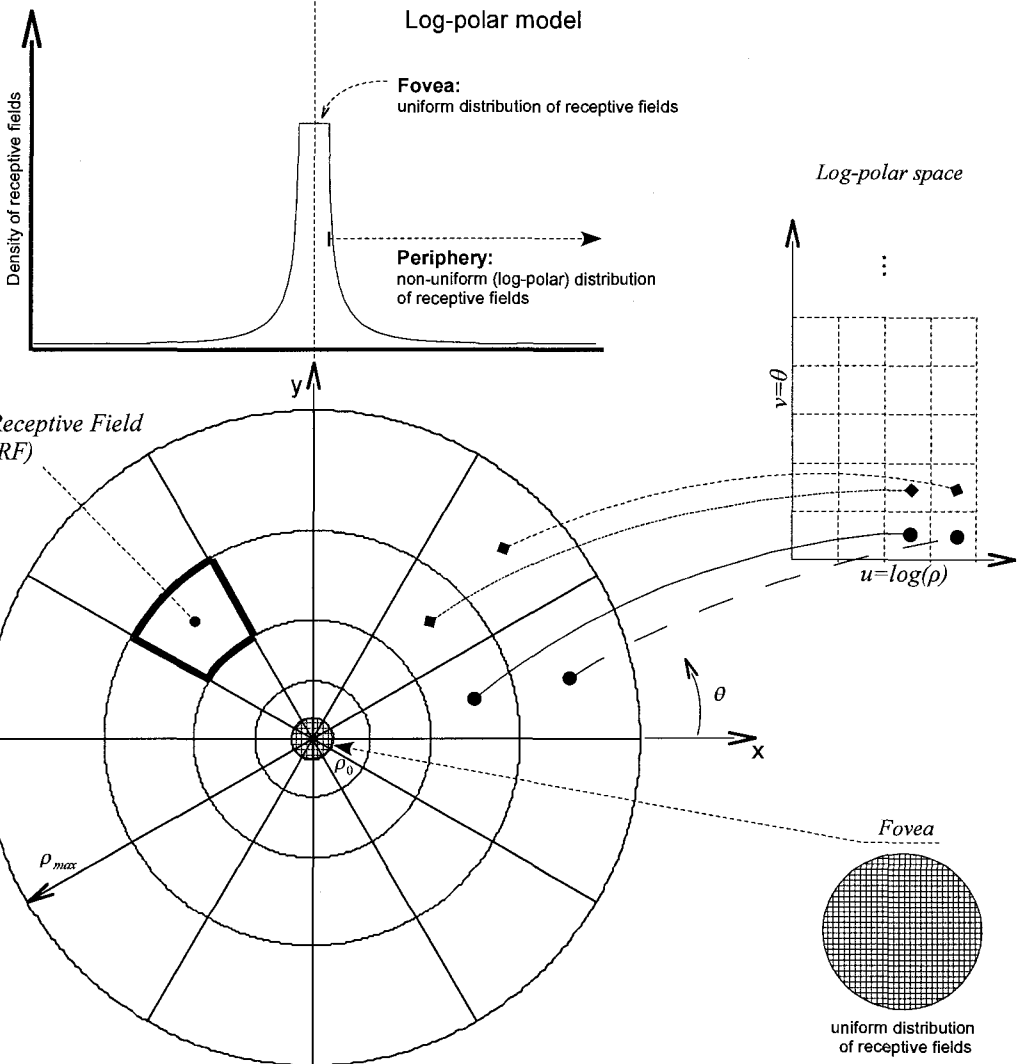
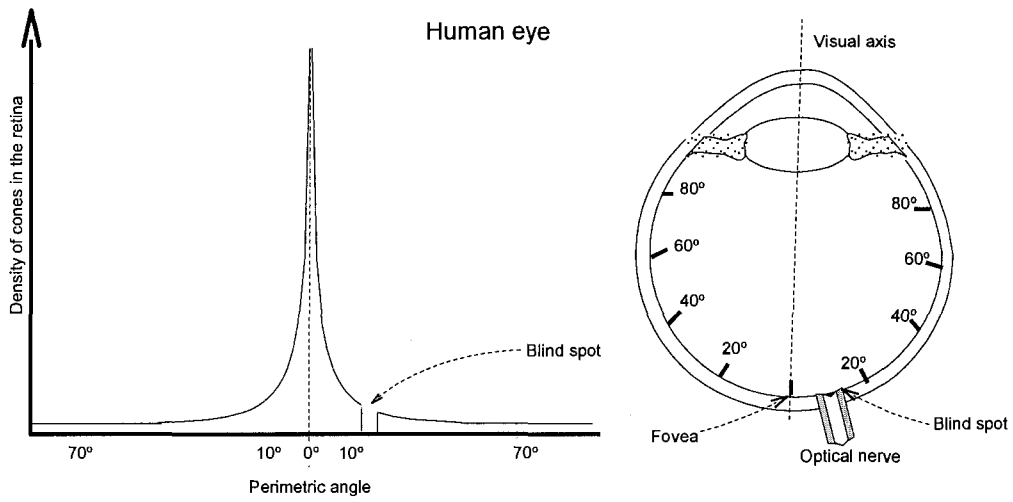


Fig. 4. Classical log-polar model of human eye

recognition [40]; image registration [41]; retina-like image sensors [42], [43], [44], [29]; and front-end for space-variant image processing [45], [46], [47].

The log-polar retina model has been well developed and has a number of variations [29]. The most common model used in synthetic vision is shown in Figure 4. The input domain mapping template (i.e., the log-polar grid), consists of a center part called the fovea and an outer part called the periphery. Because of the singularity of the logarithm at zero (i.e. in the center of log-polar transform), the inner part is often replaced with a Cartesian $x y$ image. This particular step is optional and often is ignored. Instead, a so-called central blind spot log-polar model is used (see Figure 5).

The log-polar space (u, v) is a modification of the polar coordinate space (ρ, θ) with radial logarithmic scaling. Polar coordinates (ρ, θ) are related to Cartesian coordinates (x, y) as $(\rho, \theta) = (\sqrt{x^2 + y^2}, \arctan(y/x))$ and the reverse relations is given as $(x, y) = (\rho \cos(\theta), \rho \sin(\theta))$. The log-polar transformation of the Cartesian space that is used in this work is shown in (1). It is called the central-blind-spot model commonly encountered in synthetic vision [19]. The term ρ_0 is the radius of innermost ring, that is actually cut out, and is the starting point of the transformation in radial direction (axis u in Figure 5).

$$\begin{aligned} u &= \log_a \frac{\rho}{\rho_0} = \log_a \left(\frac{\sqrt{x^2 + y^2}}{\rho_0} \right), & \rho_0 \leq \rho \leq \rho_{max} \\ v &= \theta = \arctan(y/x), & 0^\circ \leq \theta \leq 360^\circ \end{aligned} \quad (1)$$

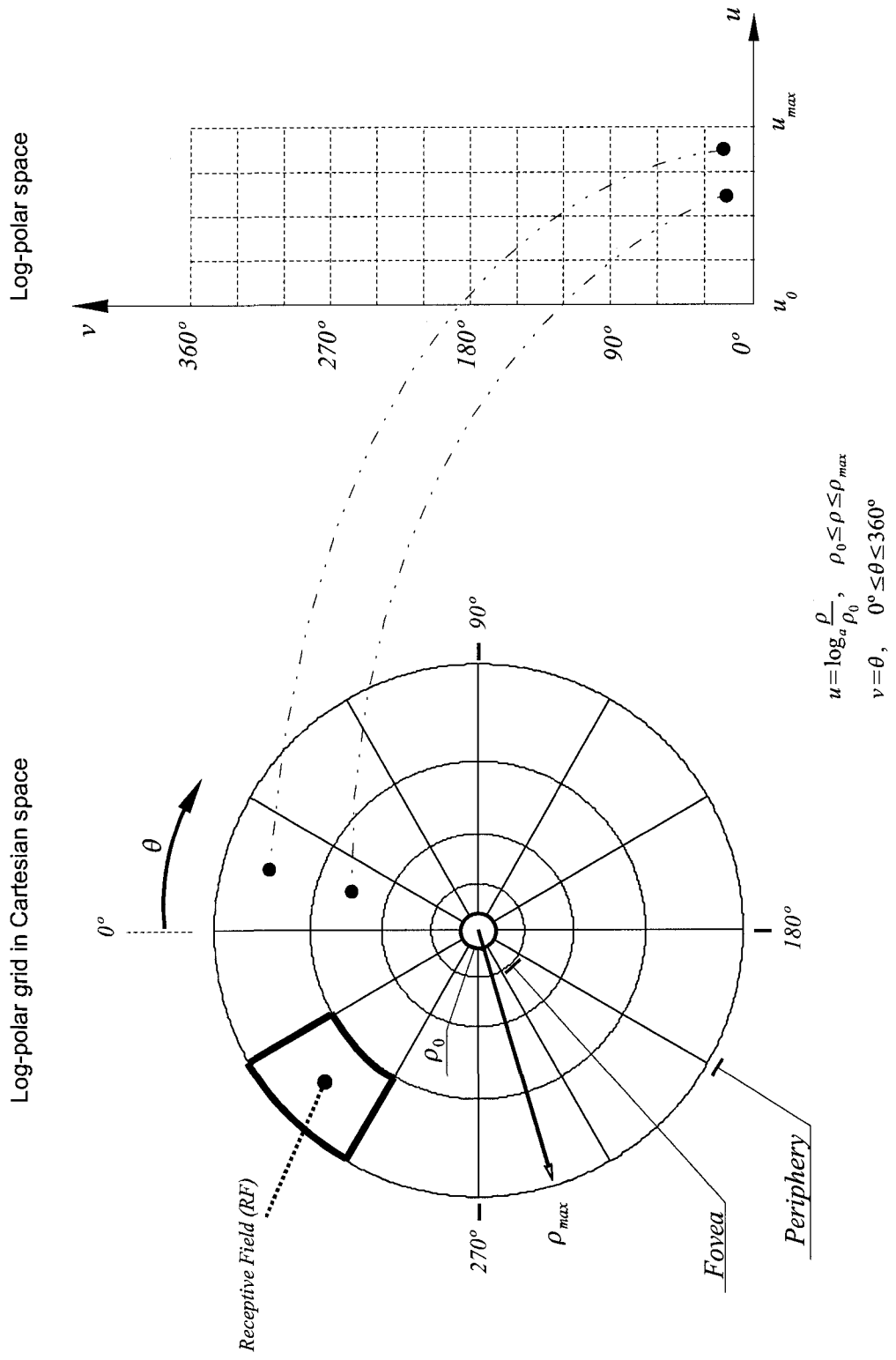


Fig. 5. "Central-blind-spot" log-polar mapping adapted for mobile vision

The reverse transformation from log-polar coordinates back to Cartesian space is given in (2).

$$\begin{aligned} x &= a^u \cos(v) \\ y &= a^u \sin(v) \end{aligned} \quad (2)$$

The log base is calculated according to (3):

$$a = e^{\frac{\ln(\frac{\rho_{max}}{\rho_0})}{u_{max}}}, \text{ since } u_{max} = \log_a \frac{\rho_{max}}{\rho_0} \quad (3)$$

Since images captured by a uniform camera index their pixels in positive Cartesian coordinates the log-polar mapper takes the form shown in equation set (4). Coordinates (x_c, y_c) define the center of log-polar transformation. These coordinates specify where the log-polar mapper grid is applied in the Cartesian image. Also, equation (4) implies the four-quadrant version of arctangent.

$$\begin{aligned} u &= \log_a \left(\frac{\sqrt{(x-x_c)^2 + (y-y_c)^2}}{\rho_0} \right) \\ v &= \arctan \frac{(y-y_c)}{(x-x_c)} \end{aligned} \quad (4)$$

The discrete log-polar transform is shown in equation set (5). The discrete form is used by synthetic vision applications, (e.g. software log-polar mapper). Typically discrete image warpers are implemented using look-up tables for real-time log-polar mapping [48], [49].

$$\begin{aligned}
u_d &= \text{floor}(\log_a \frac{\rho}{\rho_0}), \\
v_d &= \text{floor}(\frac{\theta \cdot v_{max}}{360^\circ}), \text{ where} \\
v_{max} &\text{ is the number of sectors, and } 0^\circ \leq \theta \leq 360^\circ
\end{aligned} \tag{5}$$

From (5) the size of output log-polar image is $N = u_{max} \cdot v_{max}$ pixels.

Oversampling is an essential property of discrete log-polar mapping when applied to Cartesian images, since the latter has finite resolution. The receptive fields (RF) near the center of the log-polar grid become smaller than Cartesian pixels (see Figure 6). Therefore, information near the center of Cartesian image results in a highly redundant area in the log-polar image. This effect is called *oversampling*, because the Cartesian image is oversampled close to the log-polar grid center. The oversampling effect manifests itself in the output log-polar image as a blocky area at small u (see Figure 6).

An opposite situation takes place in the periphery, where there is a many-to-one relation between Cartesian pixels and log-polar receptive fields. This is called *undersampling* and this is exactly the reason why the log-polar mapping is used for data reduction [19].

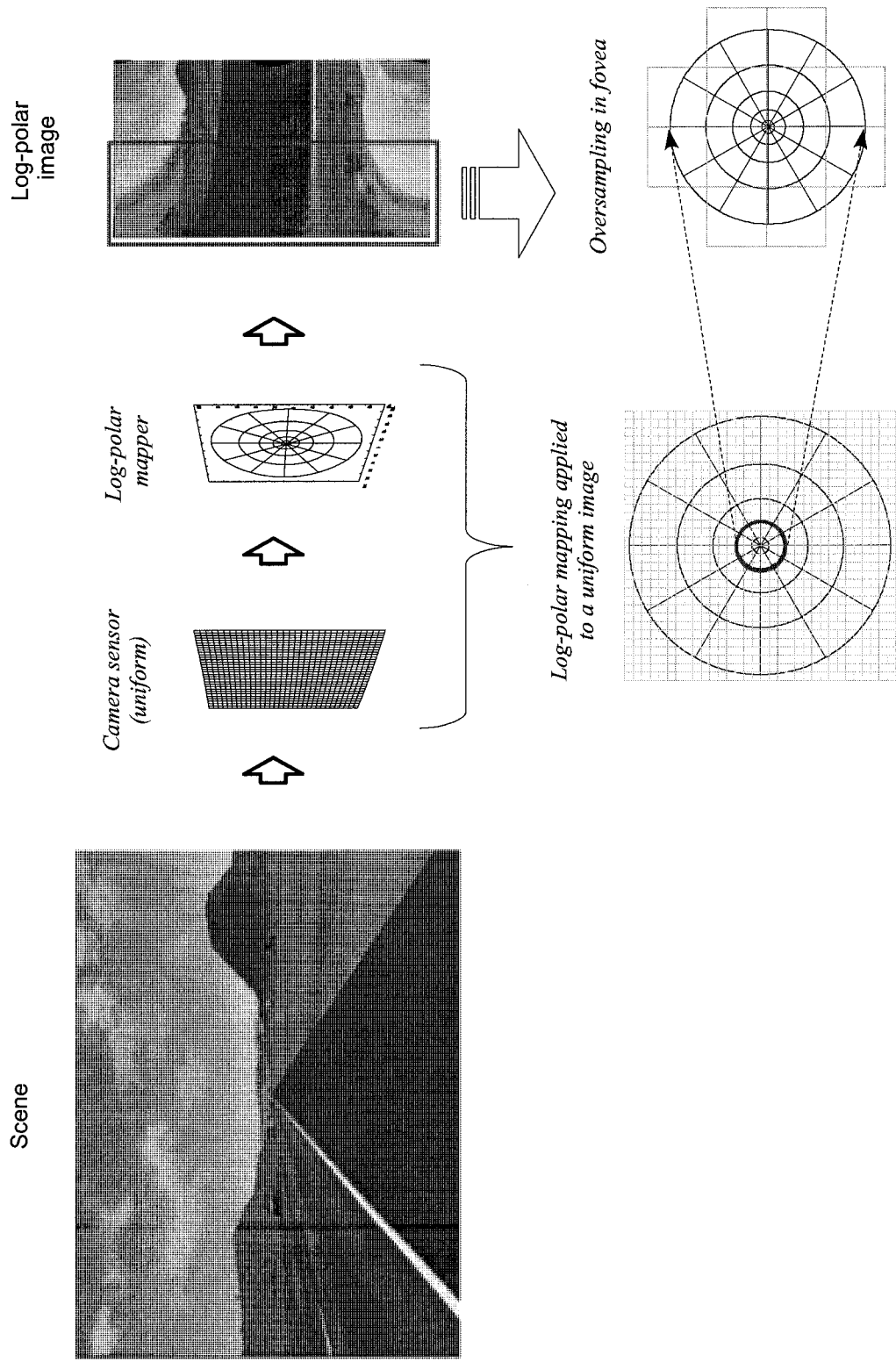


Fig. 6. Software log-polar mapper

Properties of log-polar transformation

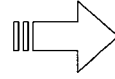
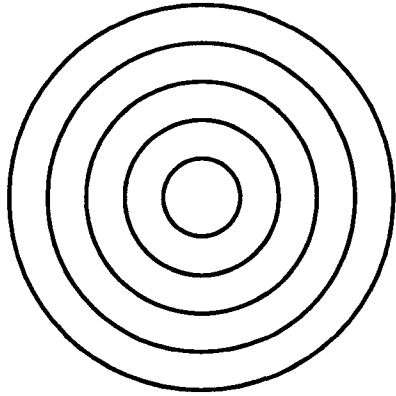
The log-polar transformation has been shown to possess scale and rotation “tolerance” properties and the ability to compress the input image through non-uniform sampling [29], [19]. Log-polar transformation is a conformal mapping, which maps a circular region in Cartesian space into a rectangular region. Its geometric properties are illustrated in Figure 7. The concentric circles in a Cartesian image become vertical lines in the cortical space because of the constant radius at all angles. Similarly a Cartesian image of radial lines, which have constant angle but variable radii, result in a map of horizontal lines.

This picture also illustrates the *rotation tolerance* property of cortical image. If the camera rotates around the center point the cortical image stays intact and only shifts in the vertical direction. This shift will be up or down depending on the rotation direction.

The scale tolerance property manifests itself in following. If the observing camera moves towards an object, its appearance does not change in the log-polar image and only shifts in the horizontal direction. This property is derived from the fact that the logarithmic function is applied in the radial dimension and the major utility of logarithms is that they reduce multiplication to addition (see Equation 6).

$$\log(\textit{scale} \cdot \textit{dimention}) = \log(\textit{scale}) + \log(\textit{dimention}) \quad (6)$$

Cartesian space



Log-polar space

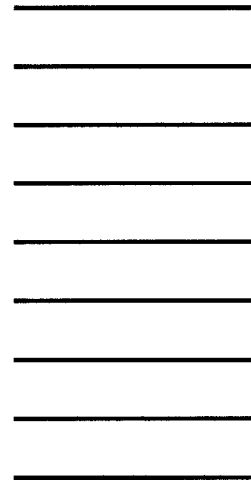
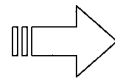
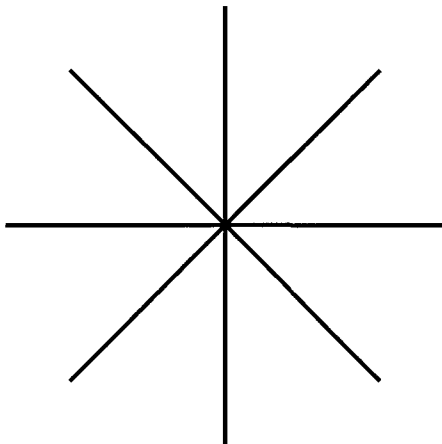
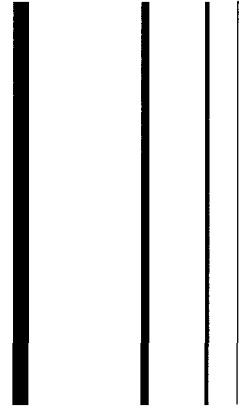


Fig. 7. Properties of log-polar transformation.

Retina-like sensors

Non-uniform image sampling has gained significant interest in the synthetic vision community. Researchers have created various retina-like visual sensors to explore their properties and application. Retina-like sensors can be divided into two categories: 1) log-polar model based [42], [44], [45], [48], [49] and 2) non-uniform but non-log-polar based sensors [43]. The first category assumes that photo-element distribution follows the log-polar grid pattern (i.e., the photo-element density grows exponentially towards the center (Figure 8a)). The second category of sensors utilizes a simplified version of the human eye which include two distinguishing structural features: 1) high photo receptor density in fovea and 2) low density in periphery. This results in two levels of photo element density as it is shown in Figure 8b.

There are three distinct types of log-polar retina-like visual sensors:

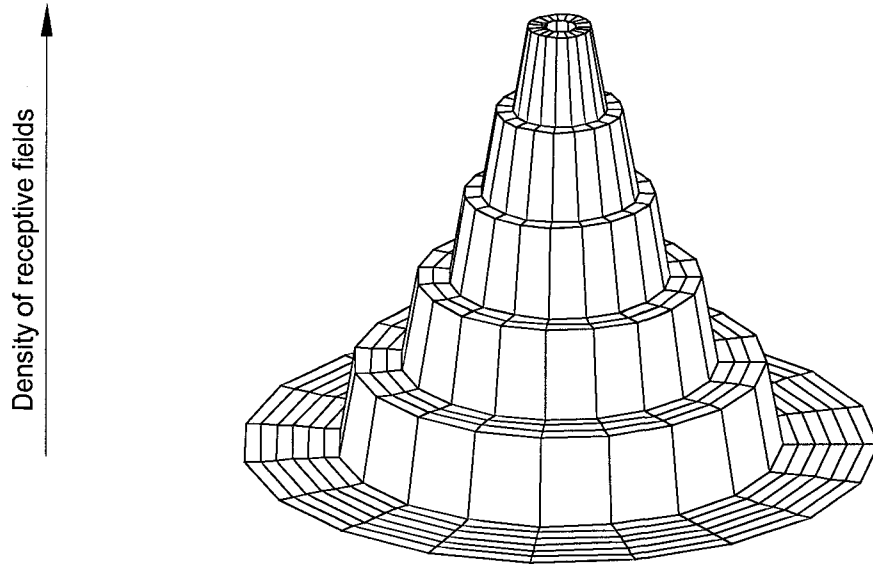
1) Solid state retina-like sensors. This is a group of specially designed and manufactured sensors that have photo element distributions directly yielding a log-polar grid arrangement [42], [44].

2) Software log-polar remappers use standard 2-D raster scan camera sensors and produce the log-polar transform by resampling the 2-D Cartesian images at the input to produce the log-polar output image [45].

3) Hardware remappers or virtual sensors use custom remapping hardware with standard 2-D raster scan camera sensors as input [48], [49].

It is easily noticeable that these three groups differ only by location of the

a) Photo element distribution follows log-polar grid style



b) Non log-polar photo element distribution

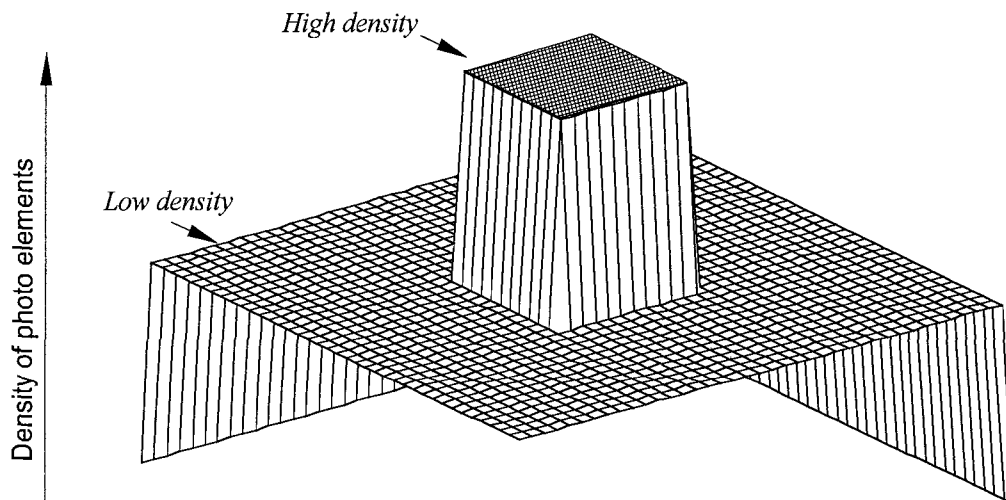


Fig. 8. Photo element distribution in retina-like sensors

log-polar data sampling or remapping in the visual data path. The first group of solid state retina-like sensors provides the most transparent image acquisition. No extra pixel mapping (and hence computing) is required because the sensor pixels already yield non-uniform distribution on the silicon surface. Their only drawback is in the manufacturing costs. Because these sensors are application specific their usage and acceptance has been limited. In contrast, the two other groups provide more cost effective solutions since they use low cost standard uniform image sensors for initial image acquisition. Hardware remappers are more expensive versus software ones, yet they are typically faster. As computers have increased in speed, and the fact that the log-polar remapping can be done through predefined look-up tables (LUT), software implementation can compete easily with dedicated hardware designs in many applications. In fact this is why the era of hardware implementation (both sensors and remappers) fell off during the late 1990s [42], [44], [48], [49], [50].

Similar classification of solid state versus software-based visual sensors can be made for non-log-polar retina-like sensors. Visual sensors with a photo element distribution shown in Figure 8b are known to be implemented on a single silicon die [43] and also by using two separate cameras with different magnification [51], [52]. In the latter case the software part of the sensor has to perform a relatively computationally expensive task of image registration or 3D reconstruction in order to locate the foveated (zoomed) image within a wide peripheral camera view.

CHAPTER II

A NOVEL COMPOSITE SENSOR FOR DEEP-FIELD VISION

Composite image sensor

In modern computer vision the log-polar transformation is a classical model for visual data reduction in primate vision. The log-polar grid outlines the spatially-variant photoreceptor density and hence defines the visual acuity. Here a novel interpretation of log-polar geometry is proposed. The novel image sensor is motivated by the vision system characteristics found in the falcon. Of particular interest is the far-seeing ability and the fact that there exist two foveae with dedicated optics for near and far vision.

To describe the idea behind the sensor, an imaginary three-dimensional surface in the form of a logarithmic funnel as depicted in Figure 9 is introduced. This surface is created by rotating a log curve around the vertical axis. This axis will be further referred to as the center axis of the log funnel. The surface of the funnel asymptotically approaches the axis at infinity, since $\log(x) \rightarrow -\infty$ when $x \rightarrow 0$.

Imagine that such a 3D surface is created around the gaze line of a camera as shown in Figure 10. Now consider a parallel projection of the 3D world in front

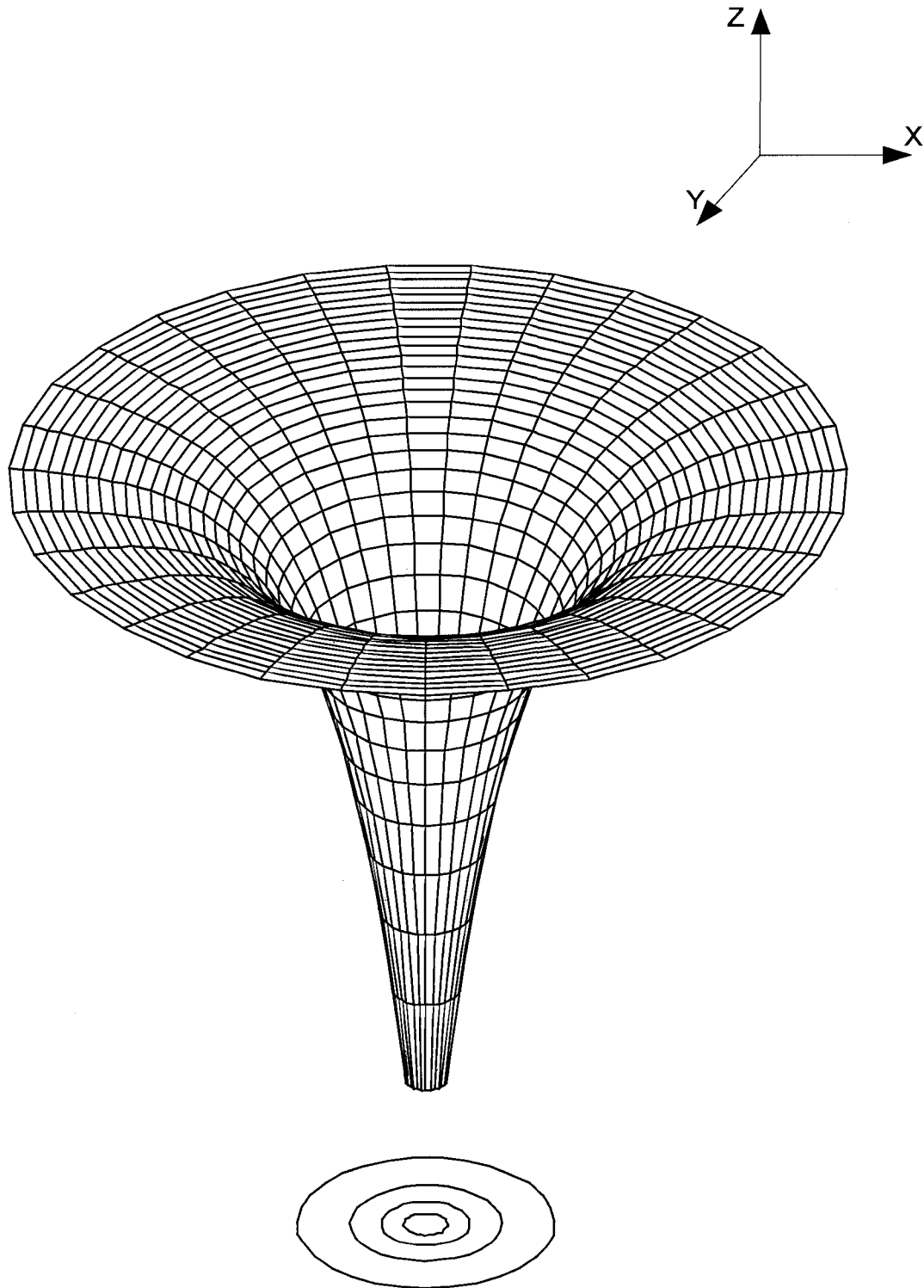


Fig. 9. 3D log funnel and its 2D projection in the form of log-polar grid

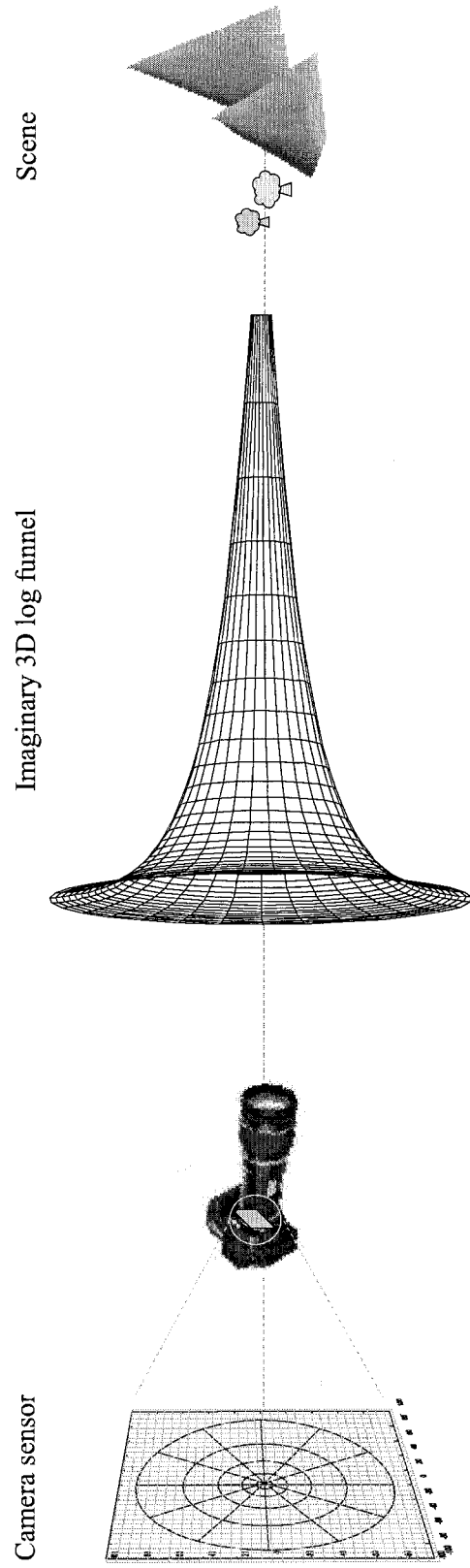


Fig. 10. Warped space in front of the camera shows how infinite depth of the field is achieved

of the camera onto the log funnel surface. Since the funnel goes to infinity when approaching the line of gaze, arbitrary distant fragments of the 3D world can be projected onto the log funnel surface. Consider now the projection of the same 3D log funnel upon a 2D plane of the camera sensor. The projection takes on the form of a log-polar grid. This suggests that in log-polar space it is possible to implement an infinitely deep field of view!

Naturally, an infinitely deep field of view can not be reached with a regular camera because of the inherent finite magnification and resolution. Instead, a set of cameras with different magnifications can be used together composing a composite visual sensor. Shown in Figure 11, the proposed composite sensor consists of two or more cameras that form a set of nested views fitting into the 3D log funnel model. The optical axes of all cameras comprising the composite sensor must be parallel and they should be packaged as close together as possible. Therefore, for sufficiently outlying scenes it can be approximated that the optical axes of all sensors' cameras rest on the same line that is now to be referred as the composite sensor's optical axis. Another condition is that the maximum radius, $\rho_{max(n+1)}$ in every nested image should be equal to the inner (minimum) radius $\rho_{0(n)}$ of the containing image. This implies that the nested images are seamlessly merged one inside the other. Under these conditions the log-polar images mapped from the nested cameras seamlessly merge into a single homogeneous composite log-polar space. This in turn is due to the scale tolerance property of log-polar transformation (Figure 12).

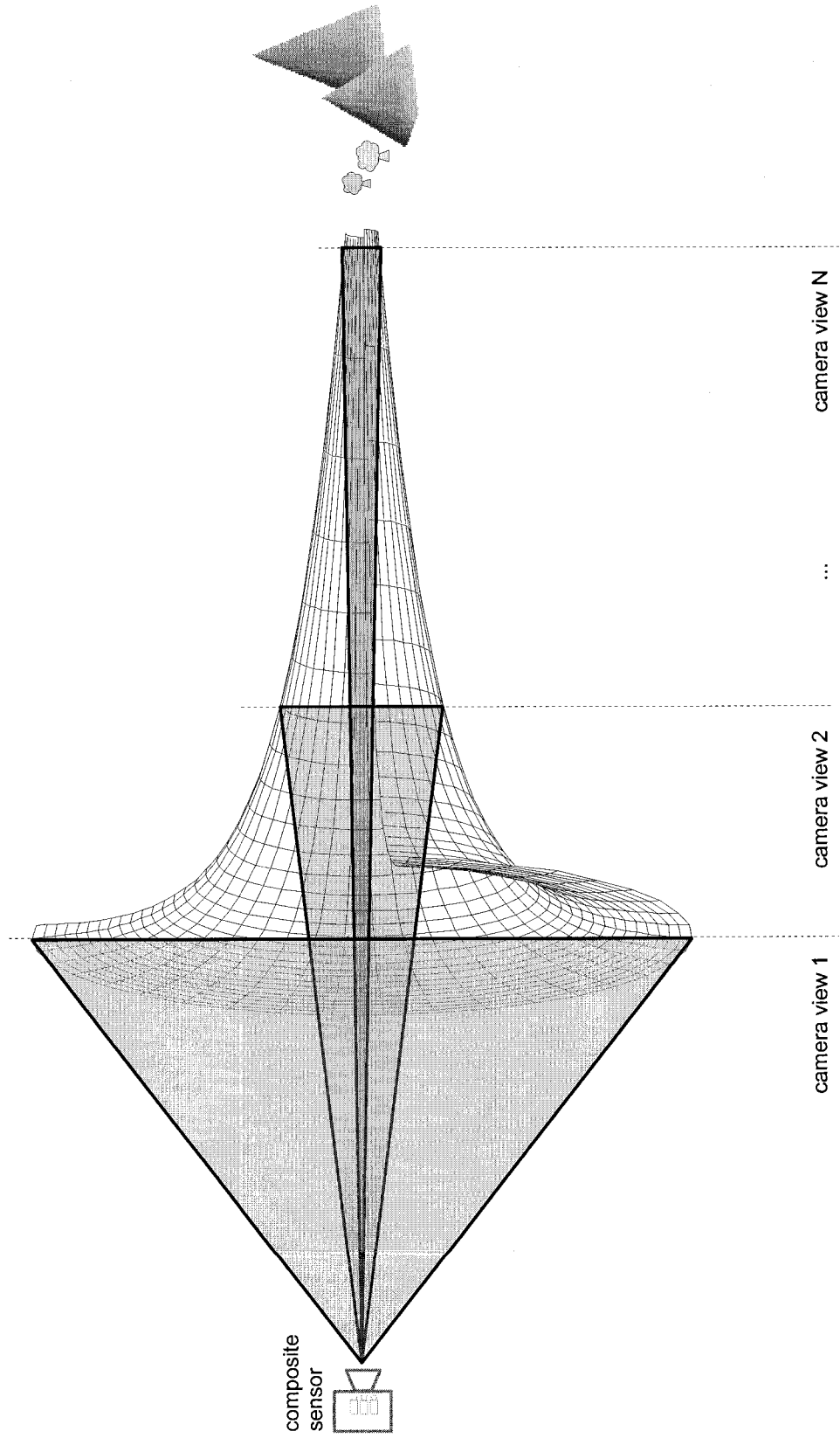


Fig. 11. Composite camera model

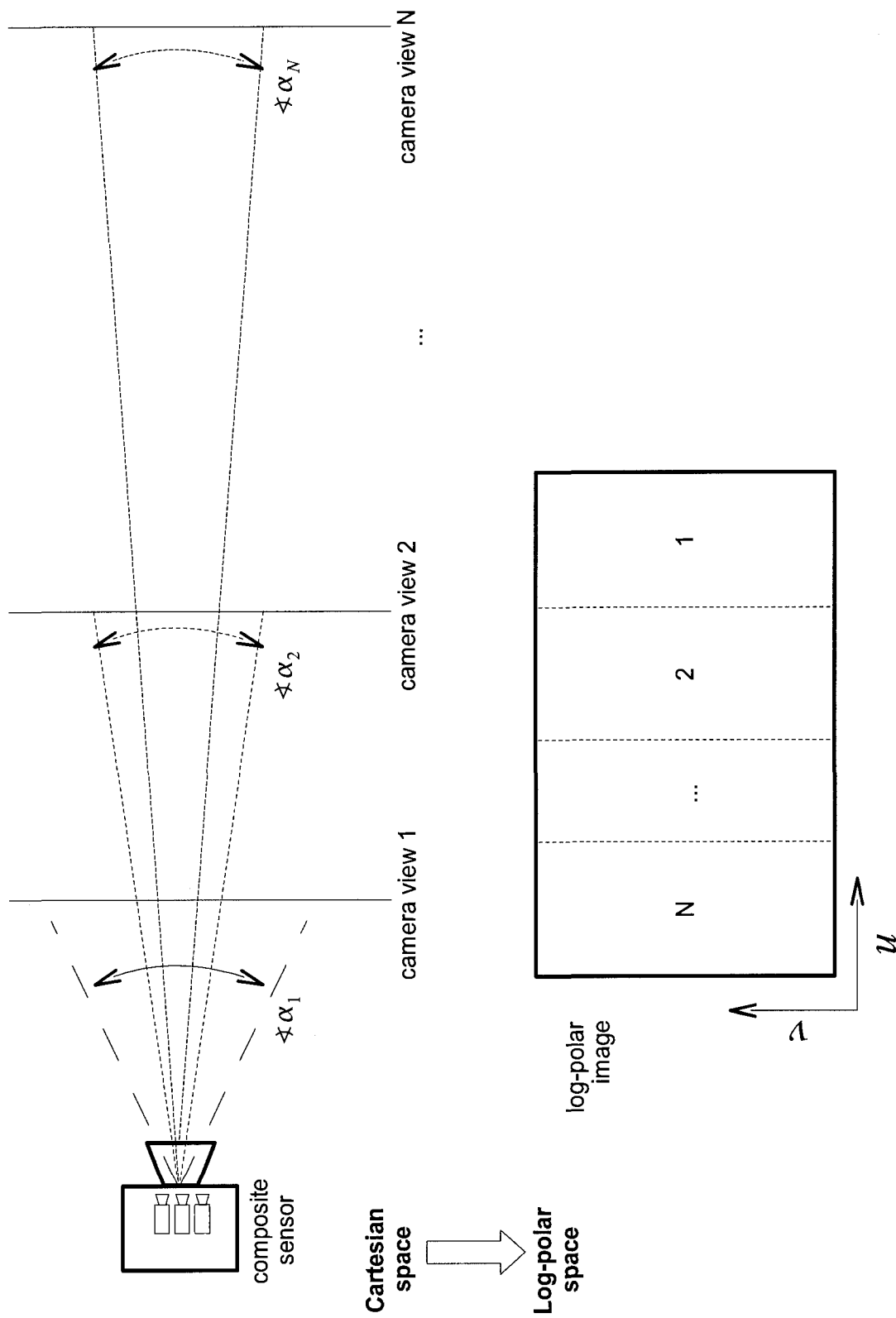
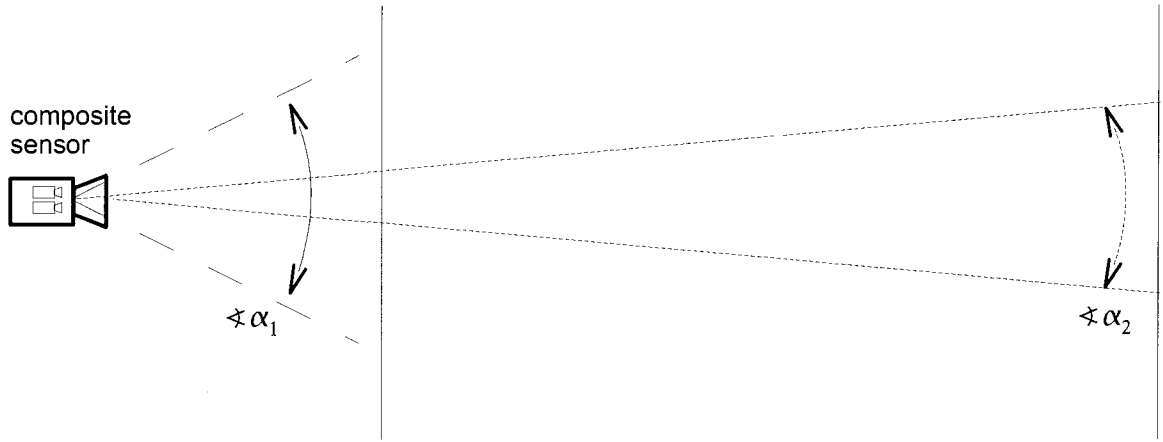


Fig. 12. Composite sensor transforms multiple camera views into a single log-polar image

A two-camera composite sensor prototype shown in Figure 13 demonstrates this spacial continuity of output in the log-polar image. Due to the properties of the log-polar transformation a continuous output image space can be created from a set of discrete image inputs taken at different magnification. Moreover, the depth of field is arbitrarily expandable. This proposed approach has not been considered before, but it extends classical log-polar sensor designs in a novel manner [29], [42], [44], [36], [37], [47].

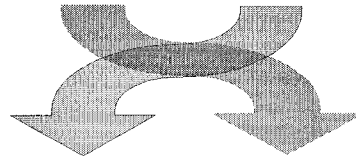
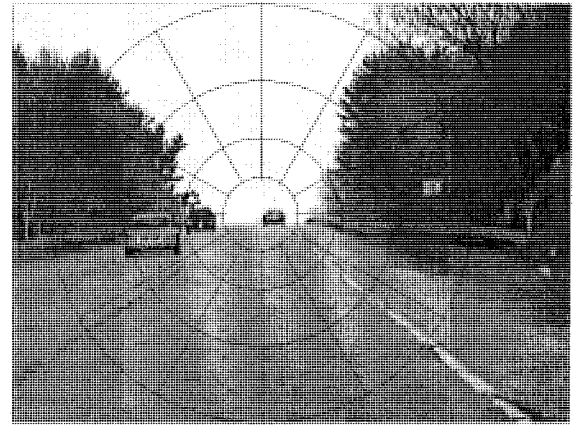
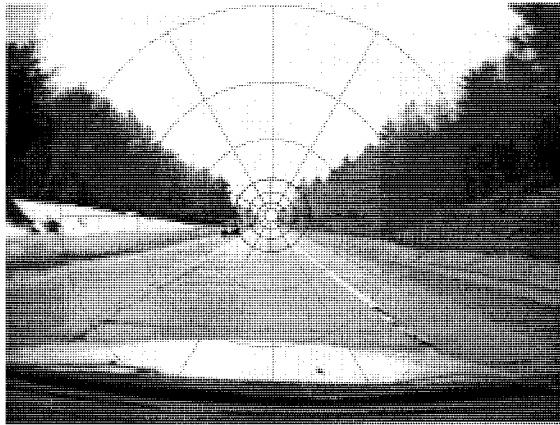
A positive collateral effect of the nested view design is that the issue of oversampling in log-polar images can be virtually eliminated. Cartesian images acquired from nested uniform cameras increase the effective resolution in the center of log-polar remapper. Therefore, the size of Cartesian pixels is reduced to keep up with decreasing receptive fields.

The scale tolerance property of log-polar transformation makes this proposed sensor uniquely advantageous for deep-field object tracking. When properly aligned the sensor preserves constant size of objects traveling at trajectories parallel to the sensor optical axis (Figure 13). It is an exceptional property for object tracking because the appearance of the objects remains relatively constant with distance. For the purposes of the mobile vision application to be revealed, this property will be referred to as a “perspective mitigation”, meaning that the objects and scenes whose appearance is changed in size due to visual perspective preserve their scale in the log-polar image space independent of distance to the observer camera.

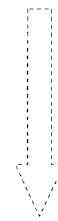


Peripheral camera view

Foveal camera view



Cartesian space



Log-polar space

Log-polar image

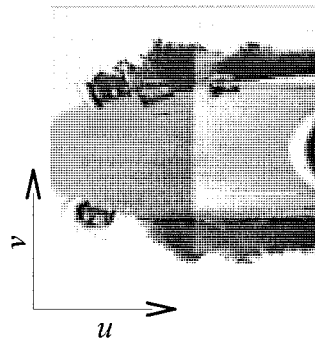


Fig. 13. Composite log-polar output in two-camera deep-field sensor

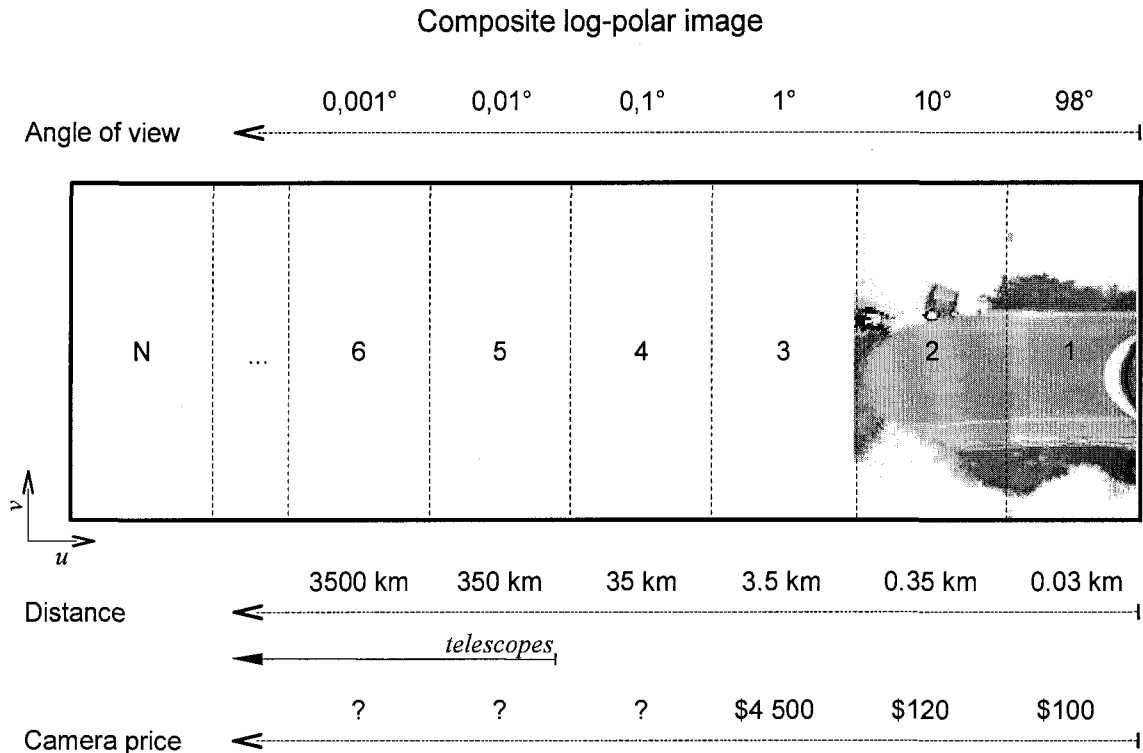


Fig. 14. N-camera composite log-polar image

To illustrate how deep is the theoretical depth of field in the composite log-polar sensor the sketch of an N-camera composite sensor output is provided in Figure 14. This picture gives an idea of how far the sensor can “see” having so many cameras. The distance grows and the view angles decrease exponentially with each camera increment. The complexity and price of required optics grow somewhat “exponentially” as well. For better visualization the first two log-polar images are taken from the two-camera field-test prototype (see Figures 20 and A.36).

This work is also focused on leveraging the proposed sensor for object tracking in the context of mobile computer vision. One of the reasons for this is that mobile computer vision is one of synthetic vision domains that would benefit

most from the proposed vision sensor. This conclusion comes directly from properties of the sensor. For example when a mobile observer and objects to be tracked move linearly in close parallel trajectories, the sensor offers 1) object tracking in long range and possibly with high relative speeds due to its deep field of view, and 2) additional computational simplicity benefit due to mitigation of perspective distortion. Therefore, synthetic vision applications that can benefit from the proposed sensor may include vehicle tracking from a mobile platform, computer vision for driver assistance, mobile computer vision applications for trains and aircraft, space based applications such as, computer assisted docking and navigation of unmanned vehicles.

Composite sensor design

The composite sensor design focuses mainly on methodology of how images from nested cameras should be stitched together and the way of choosing log-polar mapping parameters. The main goal of the guidelines is to achieve the best image quality in the output log-polar image. This section starts discussion of composite sensor design fundamentals that continues into Appendix B with more discussion and design charts. This discussion provides information on how to balance between reasonable log-polar image size, amount of oversampling given the resolutions of the input cameras and the lens choice for the cameras comprising the composite sensor.

Ideally, in order to maximize log-polar image quality, the oversampled area

in fovea should be omitted from mapping. It is recommended because the oversampled region in the log-polar image is nothing else but stretched pixels from the center of Cartesian image as shown in Figure 6. In other words it is a low resolution region in log-polar image that has limited or no use. Therefore it is preferable to choose the inner ring (ρ_0) of our center-blind-spot log-polar grid shown in Figure 5 as the border of oversampled region.

The transition between the oversampled region in the fovea and the undersampled periphery in the log-polar transformation is called the *border point*. This is the location where receptive fields in Cartesian and cortical images are equal, i.e. the receptive field in log-polar image is the size (in radial dimension) of 1 pixel in Cartesian space. In other words it is the point where the first derivative of the function $u=\log_a(\rho)$ is equal to one, i.e. the function growth is a single pixel. By solving this equation the border point radii values in the input and output spaces can be obtained. Formula (7) identifies the border circumference of radius ρ_b measured in Cartesian pixels on the camera sensor, and corresponding value of u_b in log-polar domain.

$$\begin{aligned}
 \frac{d}{d\rho} \log_a \rho &= \frac{1}{\rho \cdot \ln(a)} \\
 \frac{1}{\rho \cdot \ln(a)} &= 1 \\
 \rho_b &= \frac{1}{\ln(a)} \\
 u_b &= \log_a(\rho_b)
 \end{aligned} \tag{7}$$

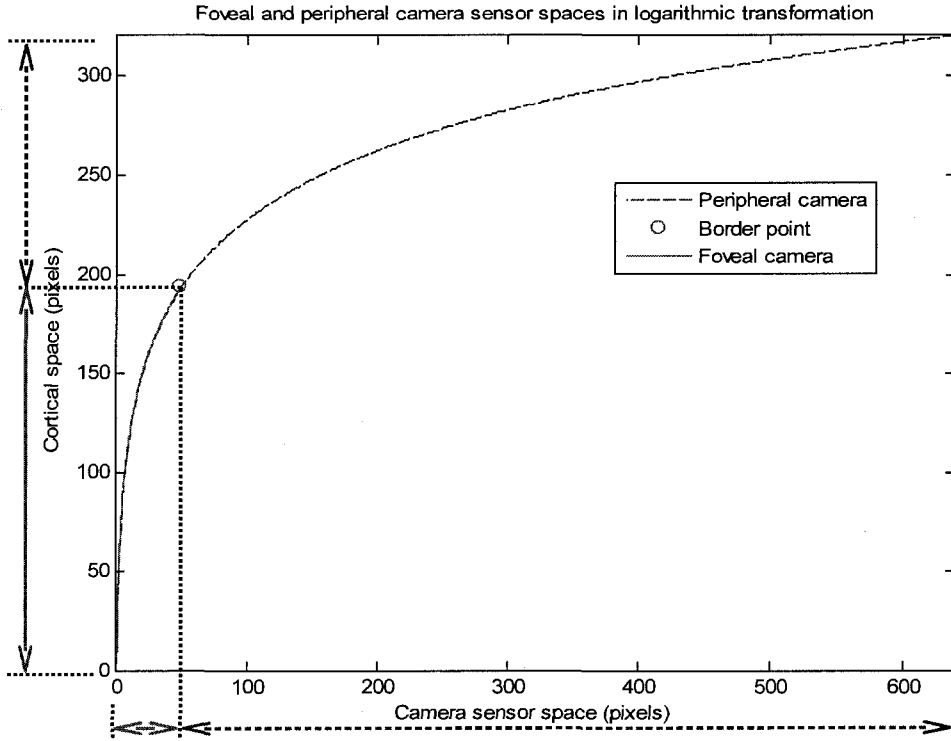


Fig. 15. Foveal and peripheral camera sensor spaces in logarithmic transformation.

The border point between oversampled and undersampled regions shows how the input images from the foveal and peripheral cameras should be fused into an output image (see Figure 15). The image from the foveal camera should be fused into the output image within the area of radius ρ_b defined by the border point. At the same time the circle area of radius ρ_b in the center of the peripheral camera input image does not participate in the fusion. In order to make a seamless insertion exactly at the border point a certain relation must be maintained between foveal and peripheral camera angles of view. This relation is called a *fovea overlay scale factor* and is described by equations (8) and (9).

$$K_b = \frac{\rho_b}{\rho_{max}} = \frac{1}{\rho_{max} \cdot \ln(a)} \quad (8)$$

Since the trigonometric function of the tangent can be approximated as a linear function near zero, the *fovea overlay scale factor* K can be reduced to foveal-to-peripheral angles ratio at small angles (i.e., $\theta/2 < 45^\circ$).

$$K_b = \frac{\rho_b}{\rho_{max}} = \frac{\tan(\theta_{fov}/2)}{\tan(\theta_{per}/2)} \quad (9)$$

$$K_b \approx \frac{\theta_{fov}}{\theta_{per}}, \text{ if } \theta_{fov} \rightarrow 0, \theta_{per} \rightarrow 0$$

The foveal camera view angle should be equal to the view angle of the peripheral camera times its scale factor as shown in equation (10). In practice it might be difficult to find cameras with matching view angles because usually manufacturers have certain lens parameters stepping while producing camera lenses. Therefore there are two ways to meet this design rule. One is to use a foveal camera with an adjustable zoom as is the case in the present prototype. Another is to adjust the log base a or ρ_{max} using formula (8).

$$\theta_{fov} = K_b \cdot \theta_{per} \quad (10)$$

In order to show both foveal and peripheral camera pixels on the same logarithmic curve (see Figure 15), equation (11) is used. Because of magnification in the foveal camera it can be said that it increases resolution in the center of peripheral camera image space. It can be observed from Figure 15 that the pixels from the foveal camera participate as sub-pixels in the foveal region of the log-polar software remapper. In order to use native indexing of the foveal camera pixels in the joint logarithmic mapping shown in Figure 15, the first

equation in equation set (1) changes into equation (11) that takes into account the scale factor. Notation fov in subindexes of ρ_{fov} and ρ_{fov0} shows that these are radii values for the foveal camera coordinates (in pixels). Practically, equation (11) shows that the foveal image is shifted left by the size of peripheral image, which otherwise means that both images are stitched together as shown in Figure 13.

$$u = \log_a \left(\left(\frac{\rho_{fov}}{\rho_{fov0}} \right) \cdot K_b \right) = \log_a \left(\frac{\rho_{fov}}{\rho_{fov0}} \right) - \log_a \left(\frac{\rho_{max}}{\rho_b} \right) \quad (11)$$

The composite sensor design theory continues in Appendix B.

The remainder of this section provides a clearer picture of the composite sensor design process. Figure 16 shows the design cycle. Typically, this process takes several iterations to match initial technical specification and available optics and equipment. It starts with determination of such log-polar parameters as log base, the upper and lower radii of the center-blind-spot log-polar grid, and the size of log-polar image (see Figure 5). The driving factor for choosing log-polar parameters is the size of target objects (here: vehicles). Then the same set of parameters should be found for the nested camera: 1) the log base remains constant, 2) the lower radius of peripheral log-polar grid becomes the upper radius of the nested (foveal) camera grid and the foveal camera view angle is determined using equation (10); 3) the joint log-polar image size is calculated. Now the foveal camera view angle drives the design process. Normally, the calculated view angle varies more or less from camera optics available from

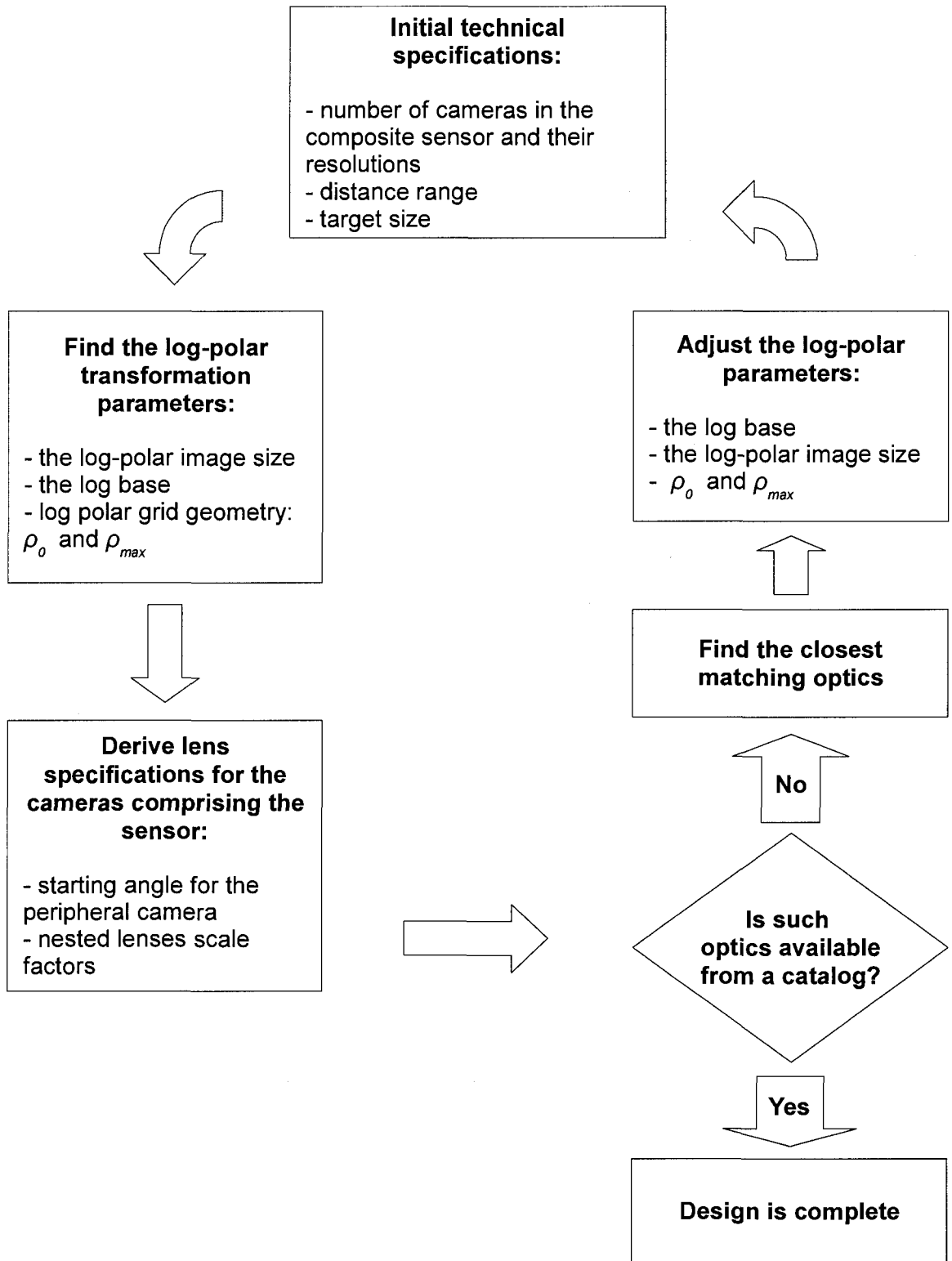


Fig. 16. Composite sensor design cycle

manufacturers. Therefore, a camera with closest view angle should be chosen and the same calculations should be repeated backwards. If the size of target objects in the new resized log-polar image is satisfactory, then the design may be considered finished.

Once the composite sensor is assembled it should be properly calibrated.

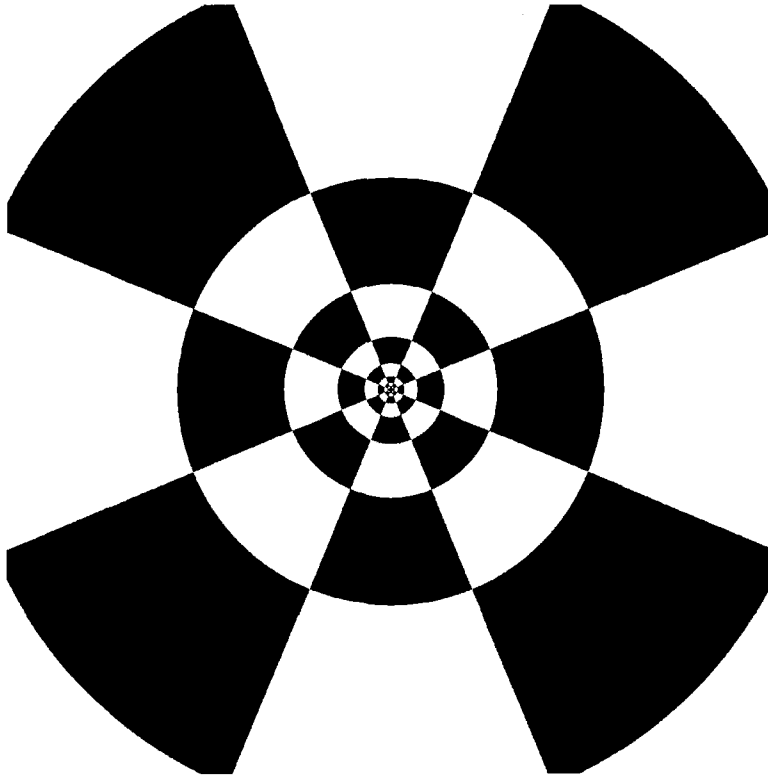
Calibration

Since the proposed composite log-polar image sensor is an original and novel idea, the developed calibration technique is also novel and original. No mention of camera calibration in log-polar space has been found in the literature. Therefore a popular “chessboard” camera calibration pattern has been adapted for log-polar space.

It is proposed to use the “Log-polar Chessboard” pattern depicted in Figure 17. The idea is to have the chessboard pattern in log-polar space. Therefore, in Cartesian space this pattern is transformed into a concentric form with exponentially scaled radial dimensions.

An example of calibration installation for a two-camera composite image sensor is shown in Figure 18. The Log-polar Chessboard pattern should be large enough to overlap from the foveal camera view into the peripheral camera view. An example of a calibrated log-polar image is also shown in Figure 18. The chessboard pattern in log-polar space makes it easy to detect angular and radial misalignments between cameras comprising the composite sensor. Examples of

Calibration pattern in Cartesian space



Log-polar space

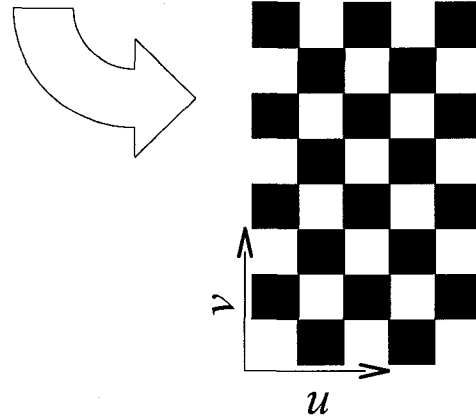


Fig. 17. "Log-polar Chessboard" calibration pattern

angular and radial misalignments are depicted in Figure 19 (a and b).

Angular (or rotational) misalignment is caused by differences in camera angles around their optical axis. It manifests itself in a vertical shift in calibration

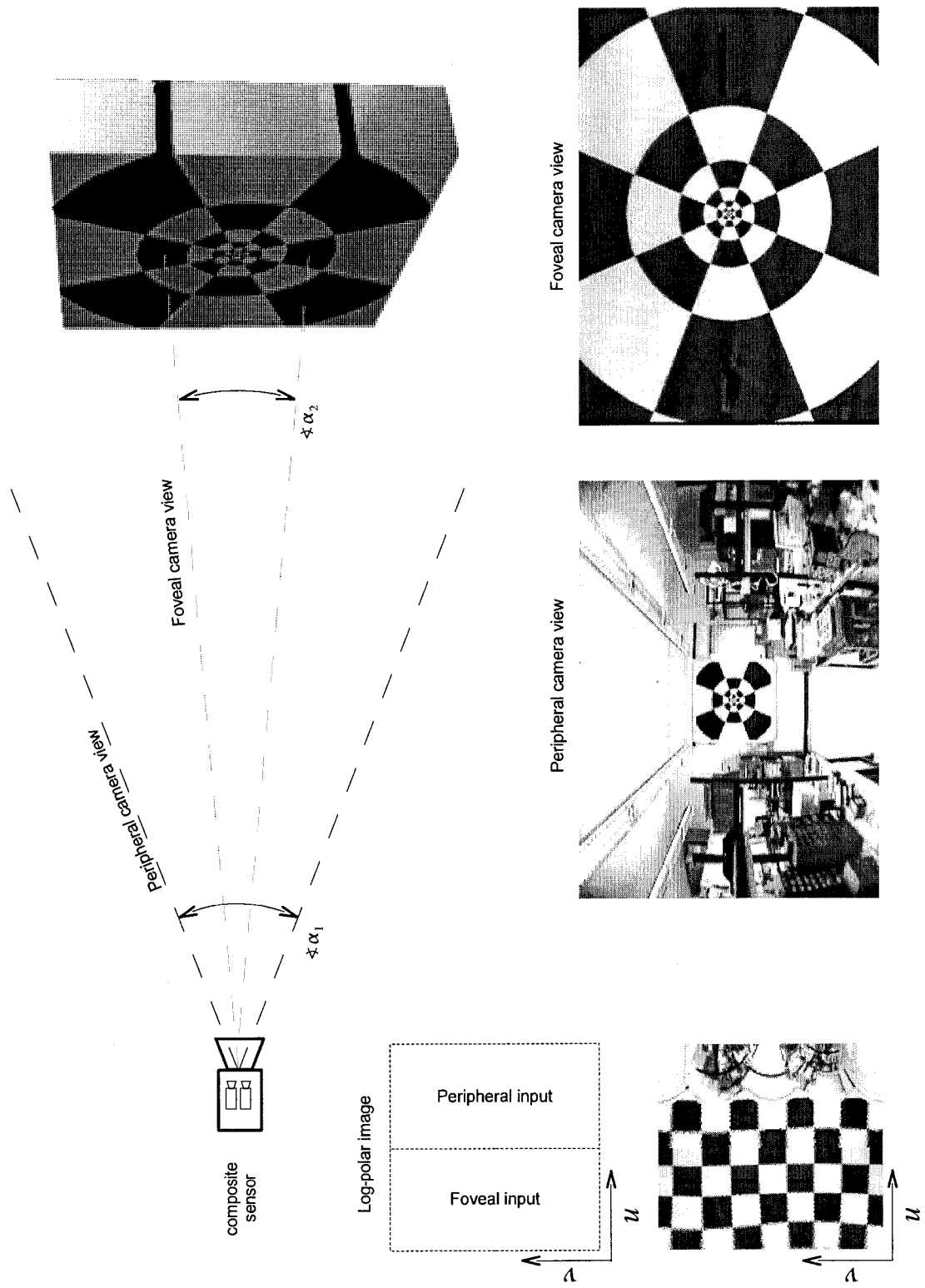
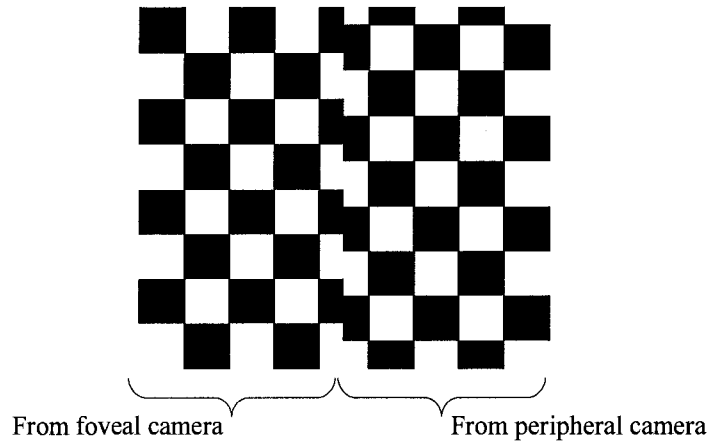
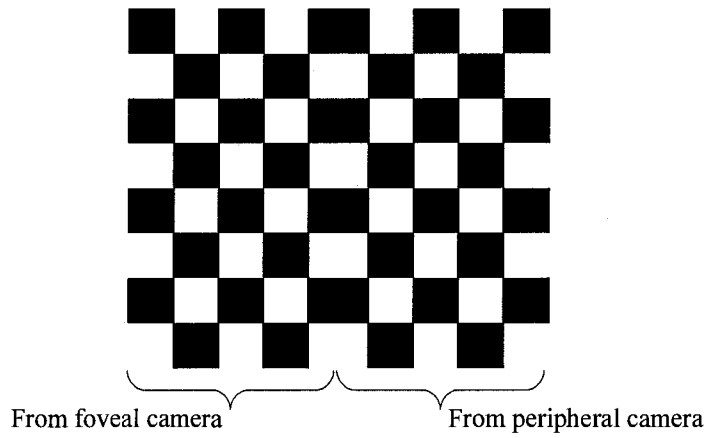


Fig. 18. Composite camera calibration setup

a) Angular misalignment



b) Radial misalignment



c) Optical axis misalignment

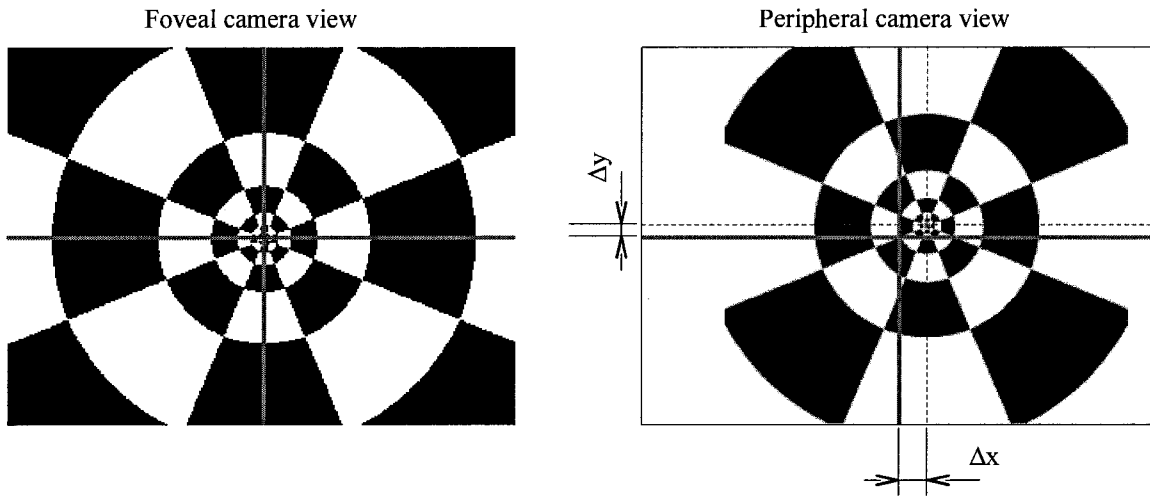


Fig. 19. Objectives of composite sensor calibration

pattern (see Figure 19a). It can be adjusted both mechanically or programmatically. Typically, rough adjustments are made by physically rotating the camera. However, fine-tuning on pixel-level is done by adjusting look-up tables of the software log-polar mapper.

Radial misalignment happens because of a mismatch between the inner radius of peripheral log-polar grid and the outer radius of foveal log-polar grid. It manifests itself in shrunk or stretched chessboard squares on the border of foveal and peripheral log-polar sections (see Figure 19b). The adjustments are made programmatically by changing radii values in the software log-polar mapper.

Optical axis misalignment means that the optical axis of the cameras that comprise the composite image sensor are not strictly parallel. Optical misalignment can be detected with the calibration pattern in the following way: if the foveal camera image center is pointed at the center of the Log-polar Chessboard pattern then peripheral camera image center is offset from the center of the Log-polar Chessboard pattern as shown in Figure 19c. In this case the calibration software reads the peripheral camera coordinates of the pattern center and uses them to adjust the position where the center of the log-polar mapper is applied.

CHAPTER III

ADAPTATION FOR AUTOMOTIVE APPLICATIONS

Composite log-polar space in automotive applications

As discussed in the prior chapter, the scale tolerance property that the log-polar transformation possesses offers a unique framework for merging multiple nested camera views into a single cortical image space. Since the initial motivation for this research had been focused on vehicle tracking of incoming traffic, the first question that arose was how many cameras in a composite image sensor would be enough to provide a reasonable depth of field for vehicle tracking. Field experiments showed that a two-camera composite image sensor is a reasonable number to work with. The “field-test” composite sensor prototype (see Figure A.36 in Appendix A) with a 98 degree peripheral view angle and a 10 degree foveal view angle provides vehicle appearance suitable for feature extraction in terms of size and detail at a distance of approximately 0.22 miles (see Figure 20). This provides approximately 6 to 7 seconds to detect a vehicle in incoming traffic moving with an average speed of 60 mph. Assuming that both vehicles are traveling at 60 mph in opposite directions the effective approach speed is 120 mph. An additional advantage is that the two-camera sensor does not require massive high-magnification optics. The constructed assembly is light

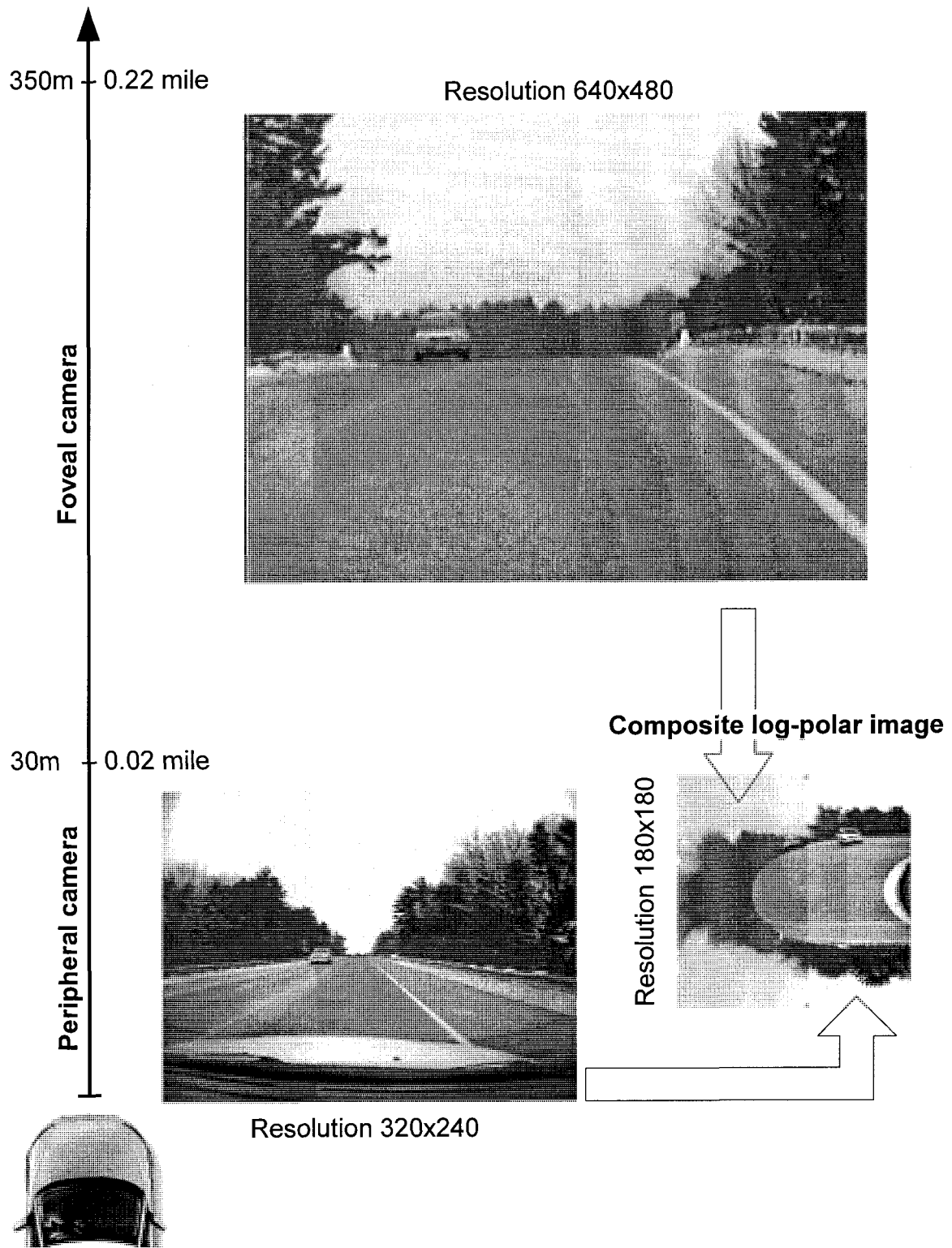


Fig. 20. Two-camera composite sensor for vehicle tracking

and compact and can be easily mounted behind the windshield of the vehicle.

For further information see Appendix A.

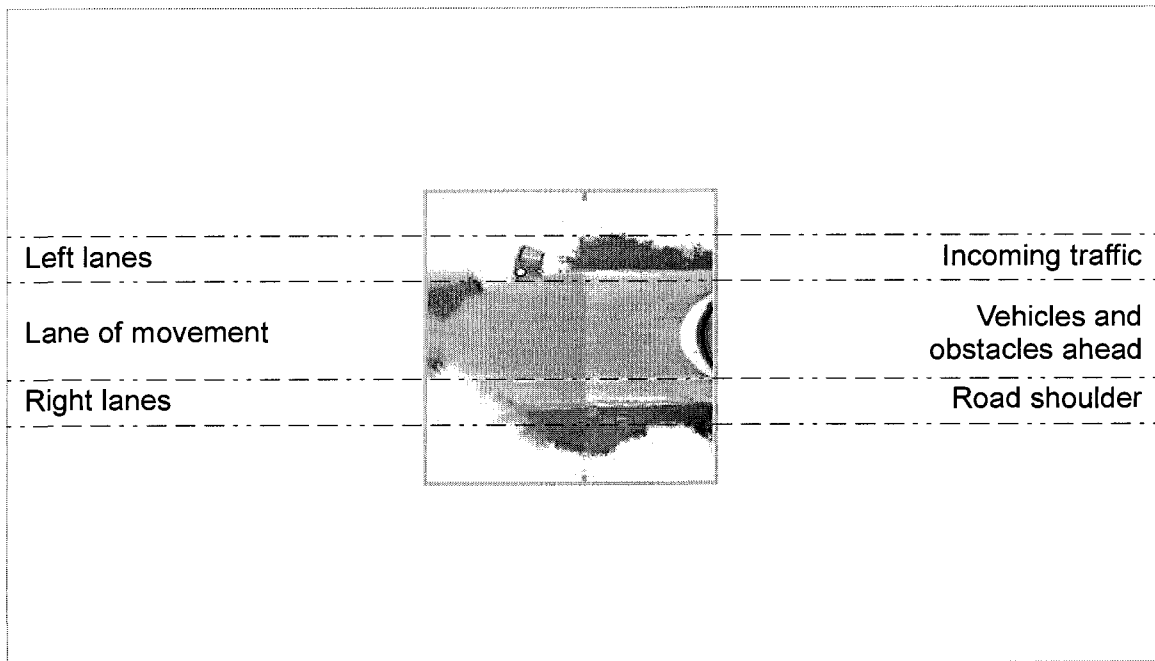
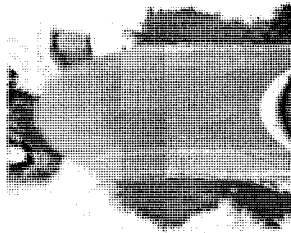


Fig. 21. Road segmentation in log-polar space

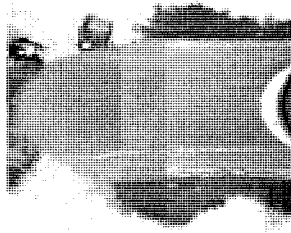
For automotive applications there are several benefits of using a composite deep-field image sensor (see Figures 21 and 22). First, the single joint image space can significantly simplify the task of detecting and tracking vehicles moving from far to near fields of view at high speed. Secondly, because of its scale and rotation tolerant properties, the log-polar mapping mitigates the perspective distortion that occurs in the image. Thus in log-polar space vehicles preserve their size at any distance from the observer (Figure 22). This reduces the complexity of the object caused by size changes due to the approach of the object from the far field to the near field. This, in turn, keeps the object feature set relatively constant. Thirdly, the log-polar image of the road provides for

Frame count

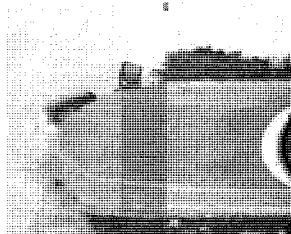
i



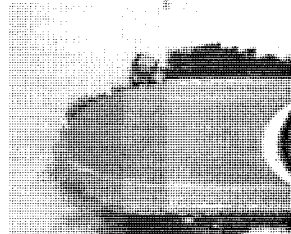
$i + 26$



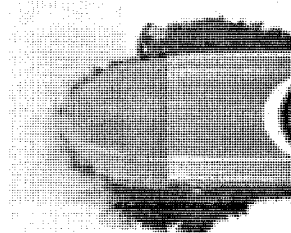
$i + 45$



$i + 89$

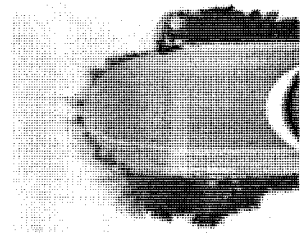


$i + 104$

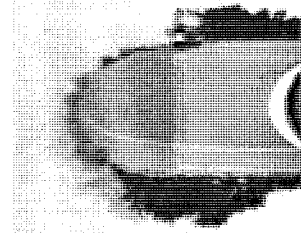


Frame count

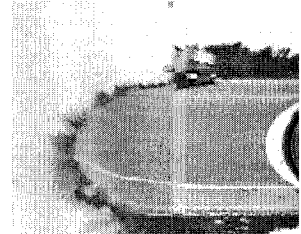
$i + 122$



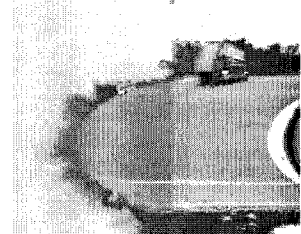
$i + 131$



$i + 138$



$i + 148$



$i + 156$

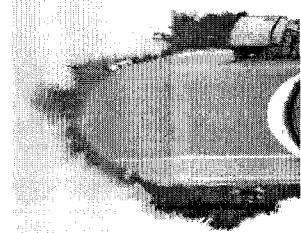


Fig. 22. Composite log-polar image sequence for vehicle tracking

excellent lane segmentation since the lanes appear in cortical space as rectangular horizontal sections (Figure 21). Fourthly, the log-polar transformation produces a much smaller image processing space as opposed to the high resolution Cartesian image. Consequently, operations in this smaller space is less computationally demanding.

It is important to note that the features of perspective mitigation and road segmentation are possible if and only if the optical axis of the camera is parallel to the road. In practice this means that the center of the log-polar transformation should be coincident with the vanishing point of the road perspective. Therefore, the advantages of log-polar representation are limited to linear road sections. Since the proposed far-vision sensor is most effective for high speed traffic, this limitation is negligible for high-speed highways due to the fact that they are mainly comprised of straight road sections.

The foveal side of a log-polar image is very sensitive to deviation from the vanishing point. For instance, Figure 23 shows possible log-polar image outcomes from a road scene acquired in the driving simulator. Figure 23a shows the ideal log-polar representation, and Figure 23b,c,d shows possible outcomes if the center of log-polar mapping is 5 pixels off the vanishing point in different directions. Thus, it is necessary to have a vanishing point fixation mechanism in order to keep the center of the log-polar grid on the vanishing point at all times..

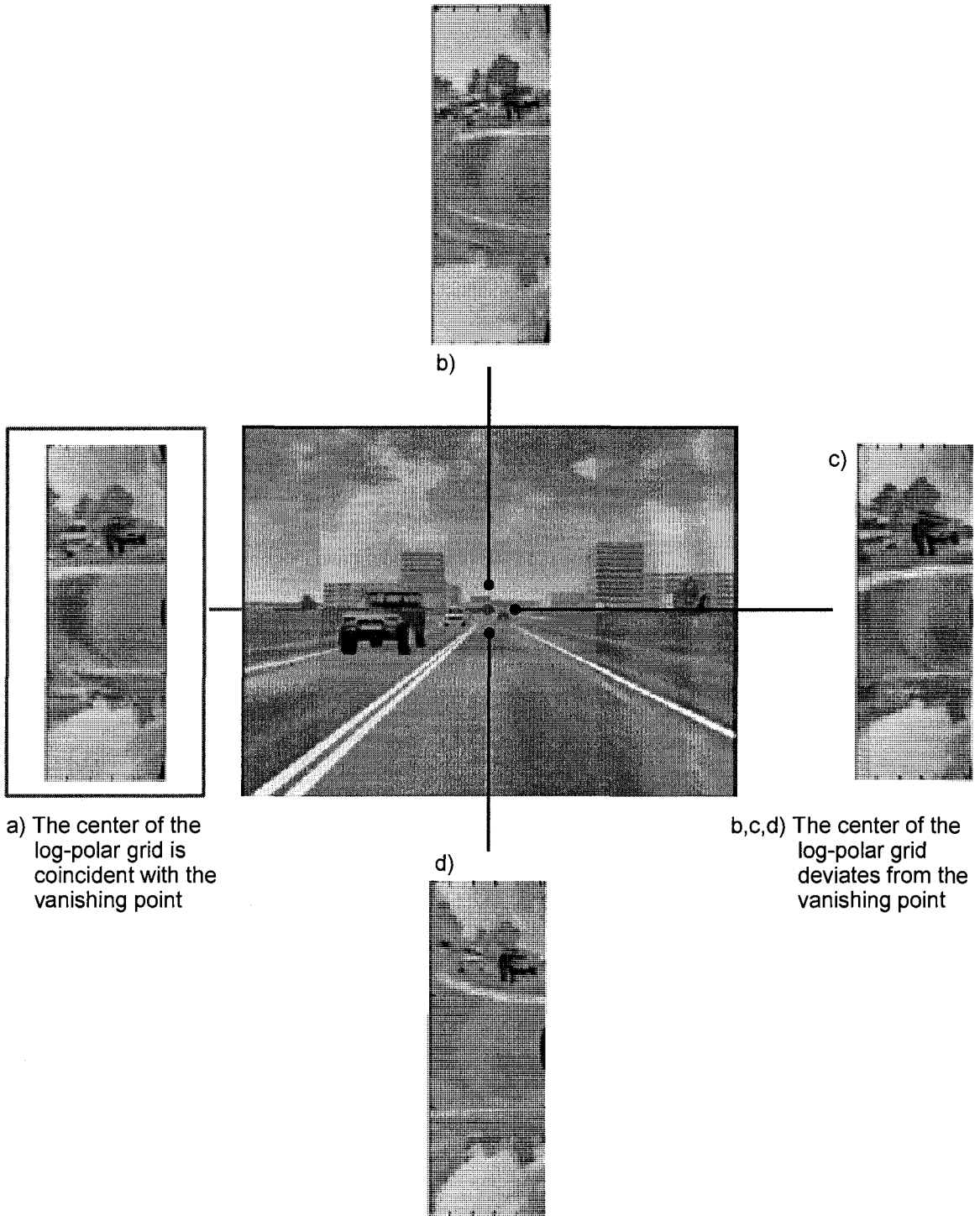


Fig. 23. Deviations from the vanishing point change road appearance in the log-polar image

Vanishing point fixation

In order to locate the road vanishing point in images (e.g., like the peripheral camera video frame depicted in Figure 20) it is possible to use some *a priori* knowledge about the environment. For instance it is possible to use such distinct visual features as road edges and painted lane separation lines to help the system properly center the transform. Since these road lines virtually converge at the vanishing point it is possible to estimate the vanishing point location as the crossing point of these lines.

A classical approach for line detection in an image is the Hough transform [53]. However, this can be computationally expensive especially when done in the high resolution Cartesian image space. Thus, it was decided to defer the Hough transform as a candidate for the vanishing point detection method and try to develop an alternative approach that would be application specific but more computationally effective. A new formulation was developed which conducts line detection directly in the log-polar space rather than in the high resolution Cartesian image. Performing the operations in the log-polar space provided two distinct benefits. The first benefit is that processing is done on a much smaller array of data and hence the computational burden for finding the lines can be significantly reduced. The second benefit is that finding lines in log-polar space reduces to merely searching a restricted area for only horizontal lines [21].

The concept of this approach is illustrated in Figure 24. The video stream acquired from the peripheral camera of the composite sensor is converted into a

log-polar image and then used for vanishing point detection. In order to find the road lines a simple horizontal edge detection is applied to the log-polar image. The longest horizontal lines are considered to be fragments of the road lines. The horizontal line length is calculated by counting adjacent pixels in the horizontal direction. By performing these running sums the result is also made tolerant to small breaks in the lines that can appear due to visual imperfections. Once lines are identified the algorithm takes the two longest horizontal lines in log-polar space, maps them back into Cartesian space and finds their crossing point in the image space. This crossing point is then the estimate of where the log-polar transform center is to be placed. For complete detail on this algorithm see Appendix C.

In the driving simulator this algorithm showed excellent precision since the computer generated environment provides clear road lines, sharp road edges, constant ambient lighting conditions, and ideal pavement conditions. However, when the vanishing point detector was tested in video sequences taken from an actual vehicle in a real road environment, it turned out that the algorithm is sensitive to poor visibility of the road markings (see Appendix C).

Visibility of the road lines greatly depends on weather conditions and natural illumination. Intuitive expectations of how weather would affect the functionality of the prototype computer vision system were quite different from the actual system response for the various environmental lighting conditions encountered during the field tests. For instance sunny weather is not ideal for road line detection.

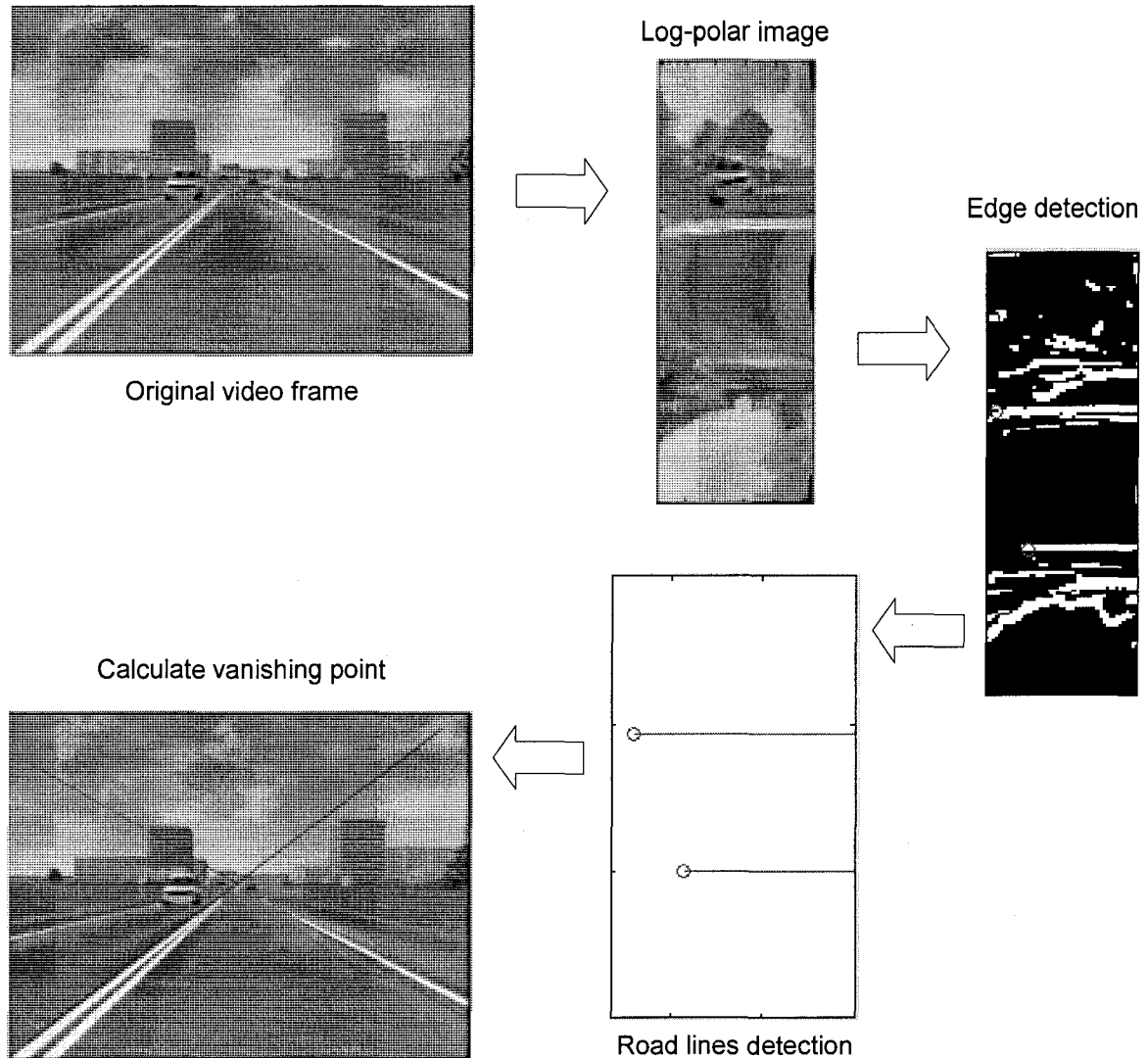


Fig. 24. Vanishing point estimation in log-polar space

This is due mainly to the presence of shadows which tend to cause false edges. Tree shadows were particularly troublesome. Another problem with both direct sunlight and sunlight reflected from the pavement is that it can simply “blind” the camera. One of the friendliest weather condition is an overcast sky with no rain. This condition provides a more even ambient light illumination. Another case where conditions were more favorable for success was during night driving. In this case the vehicle headlights illuminate the painted lane lines that are used for

estimating the vanishing point. Also, it can be difficult to detect painted lines in the rain because pavement becomes highly reflective and their visibility deteriorates; the road sides are thus a better feature for vanishing point estimation.

Because the proposed method showed rather unstable vanishing point detection in a real environment, another vanishing point detector was built using the Hough transform in Cartesian space (see Figure 25). In this case the location of the vanishing point is calculated inside the peripheral camera view. First, the Hough transform line detection algorithm is applied to a video frame. Following this the road lines are identified and retained from all lines detected in the image. Lines not deemed to be road lines are eliminated from consideration. The crossing point of the road lines is then calculated and the coordinates of the vanishing point in the peripheral view video frame are obtained. Using these coordinates, the vanishing point in the foveal camera video frame is found. This can be done knowing the composite sensor geometry determined during calibration. The log-polar transform is then applied to both video frames centered at their respective vanishing points. The details of this vanishing point detection algorithm and complete field test results are provided in Appendix D.

The Cartesian based method for finding the estimated vanishing point proved to be better than the log-polar based method because of the larger image space and hence higher resolution. This is a rather classical engineering trade-off: larger image space and higher precision versus demand for more

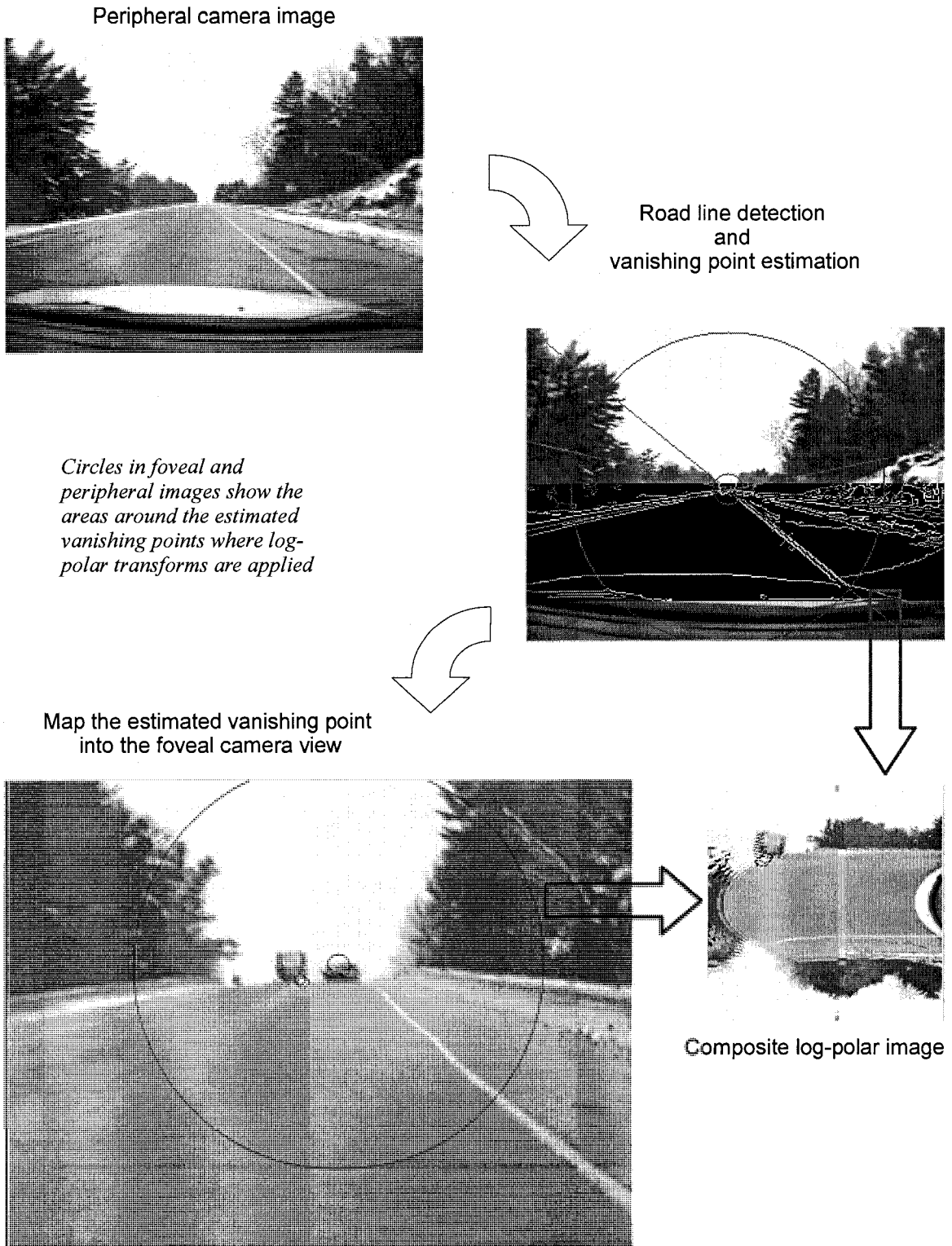


Fig. 25. Vanishing point estimation in Cartesian space

calculations resulting in larger CPU loads. The log-polar based algorithm trades on smaller CPU loads, and the Cartesian based algorithm provides for better estimation precision. Precision is more important at this “preprocessing” stage of our image processing pipeline because the vanishing point is critical for the camera alignment module, which forms the composite log-polar video stream for subsequent processing. The effectiveness of the vehicle tracking algorithm depends highly on the log-polar image quality. Therefore, the Cartesian based vanishing point detector is chosen for the camera alignment module in the image processing pipeline even though it has a higher computation burden.

CHAPTER IV

VEHICLE TRACKING IN COMPOSITE LOG-POLAR SPACE

“The utilization of space-variant image sensors in mobile computer vision is not sufficiently explored.” That was the initial observation and driving force behind this research. It was hypothesized that log-polar based space-variant image sensors may provide benefit for the implementation of certain tasks in mobile computer vision applications. The invention of the composite deep-field image sensor provides the ability to perform vehicle detection and tracking at greater distances. In conventional single-camera systems it is problematic to perform in-time vehicle detection for license plate capture because approaching vehicles are detected only in near proximity [6]. Before approaching vehicles grow to sufficient size to be detectable, they might already enter the license plate capture region and it might be too late to trigger the capturing camera.

The main benefits offered by the composite log-polar image sensor are: 1) the improved far visual range (see Figure 20), 2) the excellent road segmentation ability (Figure 21), 3) the scale invariant vehicle representation (Figure 22), and 4) the very compact image space. A system that possesses these features can perform the task of high speed vehicle detection and tracking in incoming traffic

by converting a complex set of methods that must be performed in Cartesian space into a relatively much simpler job of tracking an object of constant size moving along a linear (horizontal) trajectory contained in a much smaller rectangular image section, which ensures real-time execution.

It was proposed that a good proof of concept application would be license plate capture from incoming high speed traffic. In this application the log-polar based vehicle tracking engine triggers the license plate capture camera when approaching vehicles move into the capture range. While a vehicle moves through the capture region a number of license plate snapshots are made and stored on disk. The stored images are then fed into the third party license plate reading application that produces a text string with the license plate number [22].

On-road vehicle detection overview

The first step in vehicle tracking algorithm is to select reliable features for vehicle detection. There is a lot of material accumulated in this area during its several-decade history summarized in the best to date vehicle detection review by Zehang et al. [1]. This review classifies vehicle detection methods into three categories: 1) knowledge-based, 2) stereo vision based and 3) motion based. Only the first and the third categories are of our interest since they utilize a single camera input.

Knowledge-based vehicle detection employs *a priori* knowledge about spatial features introduced by the vehicle's presence in an image. Since vehicles

are man made objects, they can be distinguished by strong geometrical features such as symmetry, vertical and horizontal edges and corners. Other, less robust, vehicle detection methods are based on vehicle shadow, vehicle color, and texture based detection. All these methods do not work well for nighttime vehicle detection. In this case vehicle headlights are often used as the most distinct feature.

Motion-based methods are based on calculating optical flow. Giachetti et al. [6] published one of the best optical flow vehicle detection studies based on a dense optical flow vector field. They produced optical flow that can directionally and quantitatively distinguish approaching vehicles, and departing/overtaking vehicles from the background optical flow introduced by the host vehicle motion. However, using a dense optical flow vector field [6] is time consuming and impractical for a real-time system. In contrast, a sparse optical flow reduces the vector field to certain image features such as corners [54] or color blobs [55]. These methods lend themselves to a more real-time implementation in practice.

Speed-based vehicle tracking in log-polar space

The feature choice for the vehicle detection and tracking method to be adopted for the deep-field composite log-polar image sensor is based on the premise that a good vehicle tracking system should not be dependent on vehicle color, size, shape and lighting conditions at any time during day or night. The only unique geometric and light independent feature of vehicles in incoming

approaching traffic is their relative speed. When this feature is associated with optical flow it is unique in terms of direction and magnitude.

The proposed vehicle detection and tracking method is thus based on optical flow calculation. The compressed image space produced by the composite log-polar sensor offers an efficient platform for optical flow based tracking due to the limited pixel space processed (see Figure 26). The main benefit comes from the fact that the incoming lane stretching 0.22 miles ahead from the host vehicle (shown in Figure 20) can be segmented out (see Figure 21) and localized into a small image strip shown in Figure 26. Since optical flow calculation is a computationally expensive procedure, the reduced image calculation space ensures that there will be a smaller number of velocity vectors and thus faster execution. The scale-tolerance of log-polar space is also important because the vehicle shapes remain nearly constant along with geometrical features that will be used later as a base for the creation of the sparse optical flow vector field.

The next step is to decide how to generate a feature-based sparse optical flow. Since vehicles are man made objects they possess various salient features. A general paradigm of finding salient features in an image based on color, intensity and linear fragments orientation is described in [56]. Since color and intensity are not reliable features considering various light conditions and vehicle colors, a better method would be to rely on corners as salient features. Moreover, corners have been tested as features for vehicle tracking based on sparse optical flow [54].

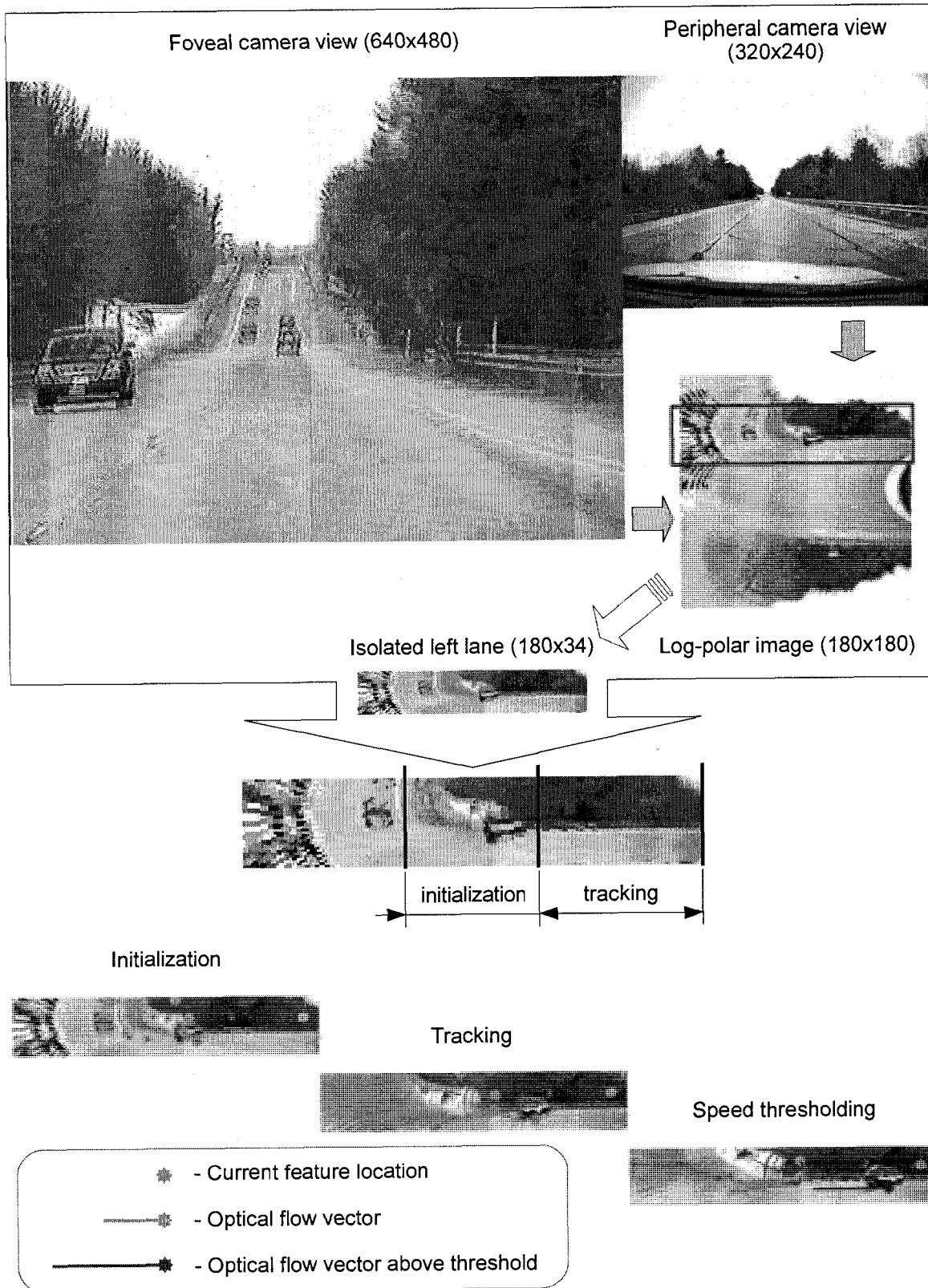


Fig. 26. Optical flow based vehicle tracking in log-polar space.

The principle of the proposed optical flow based vehicle tracking algorithm is shown in Figure 26. After preliminary image processing (i.e., the vanishing point fixation, the log-polar image composition and the road segmentation), the algorithm receives the isolated left lane image strip as an input. The resulting lane image is divided into “initialization” and “tracking” regions. The initialization region serves as vehicle detection ground: here the hypothesis generation regarding a vehicle presence takes place. The tracking region is for hypothesis verification: if an optical flow vector grows above the speed threshold, then it is identified as being generated by a vehicle.

The vehicle tracking algorithm works as a two-stage conveyor. The first stage is “hypothesis generation”, alternatively called “initialization”. It is set to detect the ten most salient features – corners with strongest eigenvalues. A small initialization region in the input image guarantees that some of those features will belong to vehicles. Obviously, some corners will be generated by the environment itself and not a vehicle. Once a feature is detected it is entered into the feature set for calculating the optical flow. Optical flow is calculated using the Lukas and Kanade algorithm [57]. Those features that propagate beyond the initialization region, (i.e. faster features), stay in the set. All others are dismissed before the next initialization. Hypothesis generation takes place once in 50 frames. This corresponds to 2 seconds given a 25 fps video frame rate giving enough time for vehicles to move out of the initialization region. The 50-frame interval between initializations was determined experimentally. It is a parameter

that can be changed to match other frame rates or tuned to different applications and target object speeds.

The second stage of the algorithm is “hypothesis verification”, alternatively called “tracking”. The features that reliably stay with vehicles within the tracking region have the largest speed and longest travel distance. A vector length threshold is introduced to determine if an optical flow vector belongs to a vehicle. This threshold is further referred as the “speed threshold”. In reality it is measured in pixels comprising an optical flow vector calculated over an interval of fifty video frames. All vectors with horizontal component (X coordinate) above the threshold are marked in red as shown in Figure 26. The coordinates of these features identified in red are now considered to be immediate coordinates of a vehicle. Thus a target vehicle is detected and its position in the input image is determined. This information can be used to determine the distance and trajectory of the approaching vehicle with respect to the host vehicle. One drawback to this algorithm is that it can not be reliably used for estimation of actual vehicle size. In practice the proposed vehicle tracking algorithm is an excellent candidate for driver assistance or warning systems, and for license plate scan applications. It can be used beyond the automotive environment and it is proposed that further work can be applied for applications related to high speed trains or aircraft.

As to graphical notation in Figure 26 and further in Figures 27, 28, 29 the features that are currently in the optical flow calculation set are shown with bold

dots. Lines connected to the dots show the distance vectors of propagating features. Green indicates features below the speed threshold and red indicates those that are above the threshold.

Figures 27, 28 and 29 illustrate the different cases of vehicle tracking. Figure 27 shows tracking of a single vehicle in a step-by-step fashion. In the right column there is a collection of images illustrating vehicle propagation between the feature initialization region and the last frame when the vehicle has left the scene. On the left there are two pairs of foveal and peripheral camera snapshots. The first pair is related to the second log-polar lane image in the sequence and the second pair corresponds to the second log-polar image from the end of the sequence. These snapshot pairs provide a better visualization and understanding of where the vehicles are situated in Cartesian space.

In Figures 27 and 28 it is interesting to see how salient features are detected and how they propagate with respect to the vehicle motion. As expected, most of the corners are detected in and around the vehicle as hypothesized. However, some of these features slip away from moving vehicles which can be problematic. For the “lost” features there are two further possibilities to consider. Firstly, they can remain on the background and thus be switched to the slow motion category (i.e., non vehicle). In this case if they were marked as “red”, their vectors are reset during the next initialization phase. Secondly, the slipped features can “stick” to the next incoming vehicle in which case they can essentially become “lost” then “found”. This situation is well illustrated by the

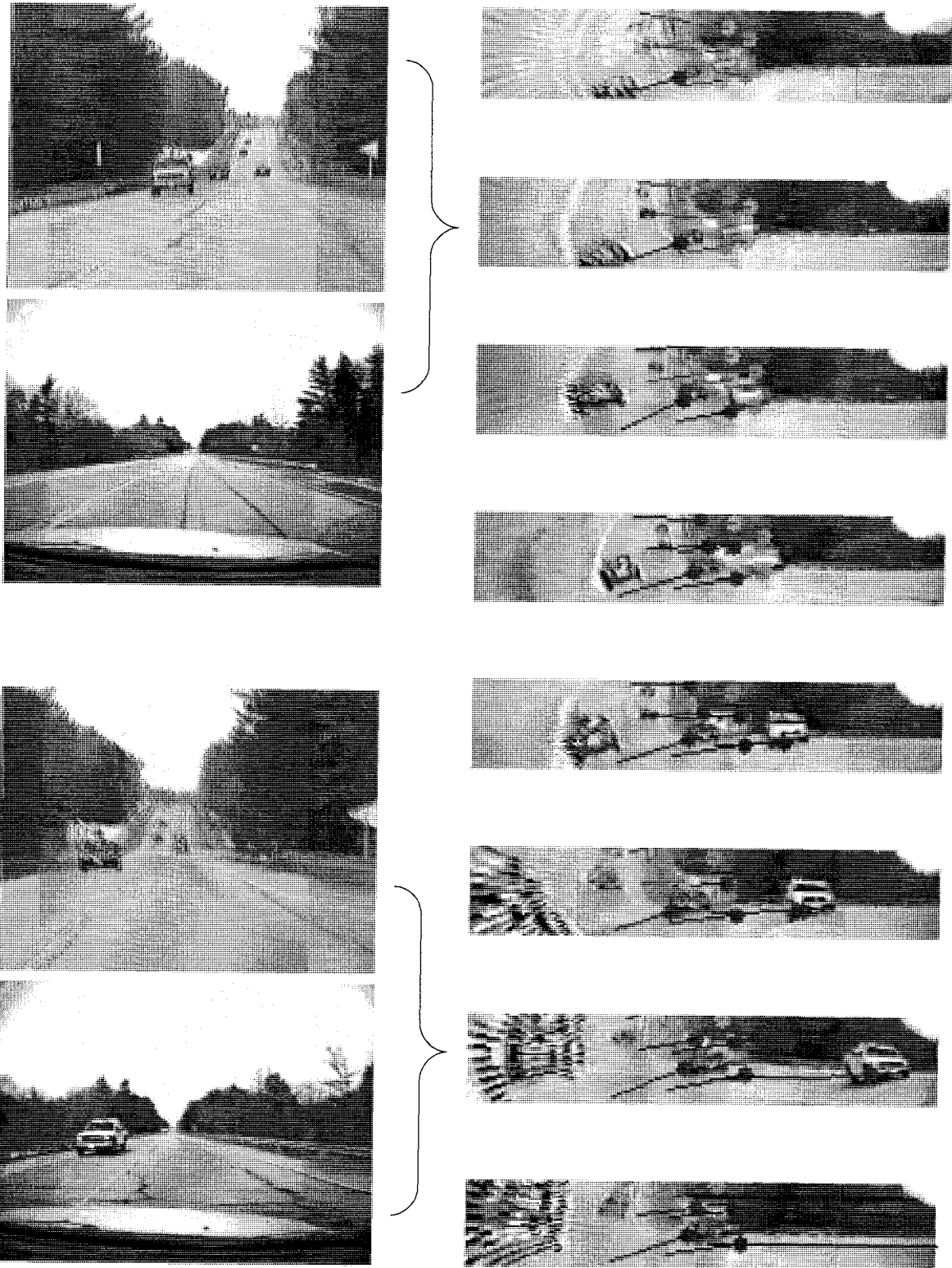


Fig. 27. Optical flow based tracking of a single vehicle.

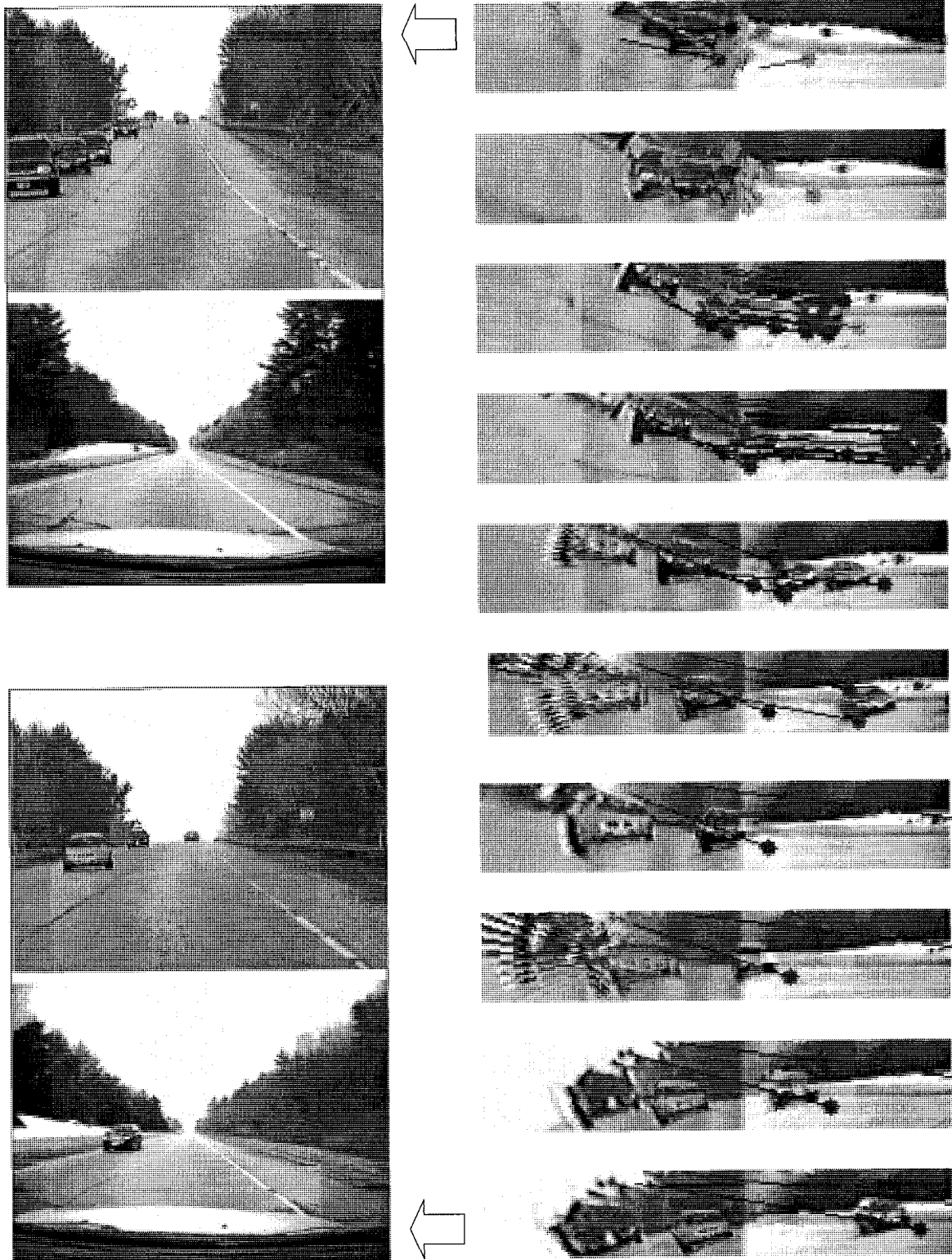


Fig. 28. Optical flow based tracking of a chain of vehicles.

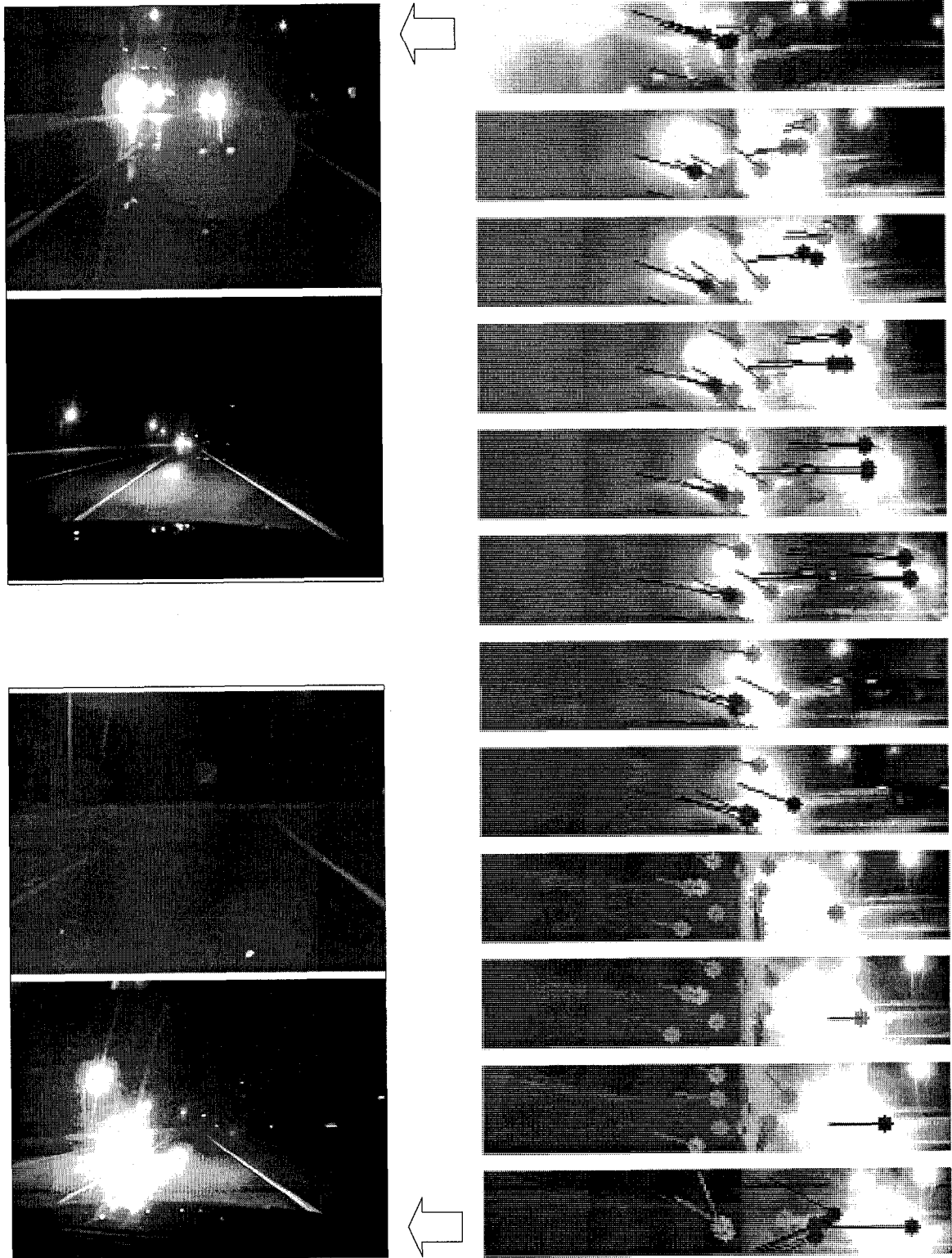


Fig. 29. Optical flow based tracking in night conditions

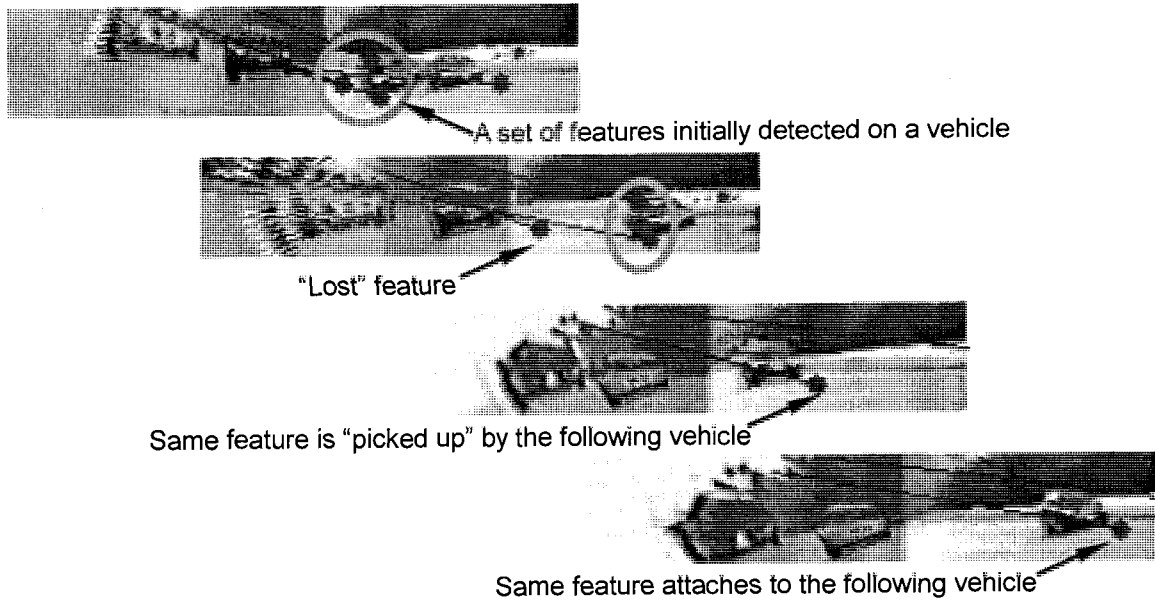


Fig. 30. "Lost" and then "found" optical flow vector features.

example of tracking a chain of vehicles as depicted in Figure 28 then shown in more detail in Figure 30. Similarly, the features originally detected in the background may "jump on" a passing vehicle and stay there. This can happen due to the original background texture being blocked by passing vehicles. When the vehicle leaves behind strong features in the background, they can attract the optical flow calculation algorithm and "steal" a feature vector from the vehicle. Because of these issues, the most stable features for a vehicle are located in the front part of the vehicle. This statement is supported by examples in Figures 27, 28 and 29.

For vehicle tracking in the night time (Figure 29) there appears a different set of features from those that appear for day time. For night time it is seen that the corners are detected on the borders of a halo formed by the headlights. Although a bit unexpected, without any algorithmic or system modifications, headlight

halos provide good features to track and the tracking algorithm performs as well as in day time.

License plate scan

The proposed motion-based vehicle tracking algorithm can be used in a license plate capture application for vehicle detection in incoming traffic without further modifications. Screen shots from the proof-of-concept license capture application are displayed in Figure 32. For this application one more camera is added to the system. This camera is dedicated to capturing the high resolution imagery of the license plate. This third camera is installed on the dashboard in front of the steering wheel as shown in Figure 31.

The vehicle tracking engine software module triggers the license plate capture camera if an approaching vehicle enters into the capture range. The capture range is represented as “capture region” in the log-polar image of the left lane (see Figure 32). While a vehicle moves through the “capture region” the capture camera is active and takes snapshots with a frame rate of 25 fps. In this manner a number of license plate snapshots are made to ensure that a suitable image is stored for subsequent analysis by a character recognition module.

Before being stored onto disk the acquired snapshot image is deinterlaced (see Figure 33). The deinterlacing stage is present in the system since the capture camera acquires video in NTSC standard format [58]. The NTSC system used in North America and Japan captures frames every 25th of a second, but

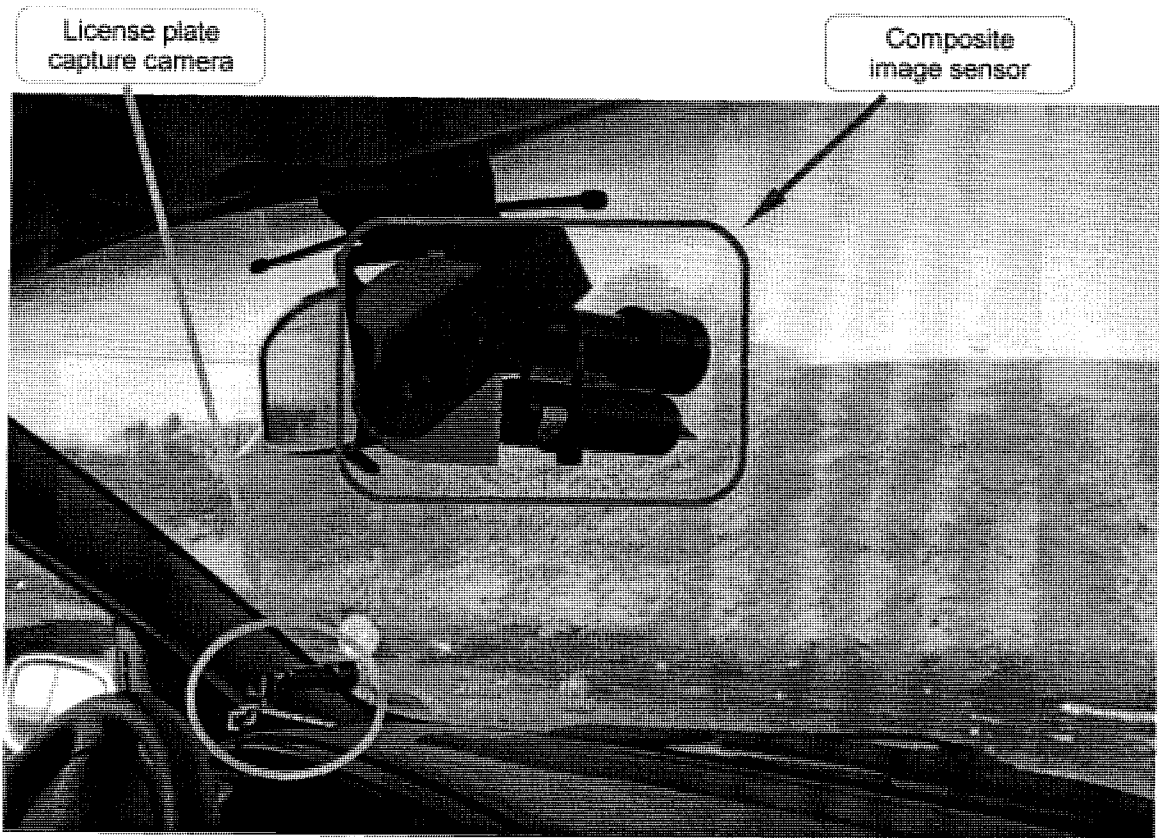


Fig. 31. License plate capture camera and the composite sensor installed in the test vehicle.

they are recorded as a series of 50th second fields, each containing only the odd or even lines of the display. Therefore, for every captured license plate image there are actually two images that are saved to disk. The two images are identified as the odd lines or the even lines and together form the whole image frame. Logically, these images are half the height of the original. This step is only necessary if one is to use interlaced format cameras for the system. Other cameras which are not interlaced can be purchased if desired and then this step is unnecessary. Interlaced cameras were chosen for this dissertation since they are easily obtained and are cost effective.

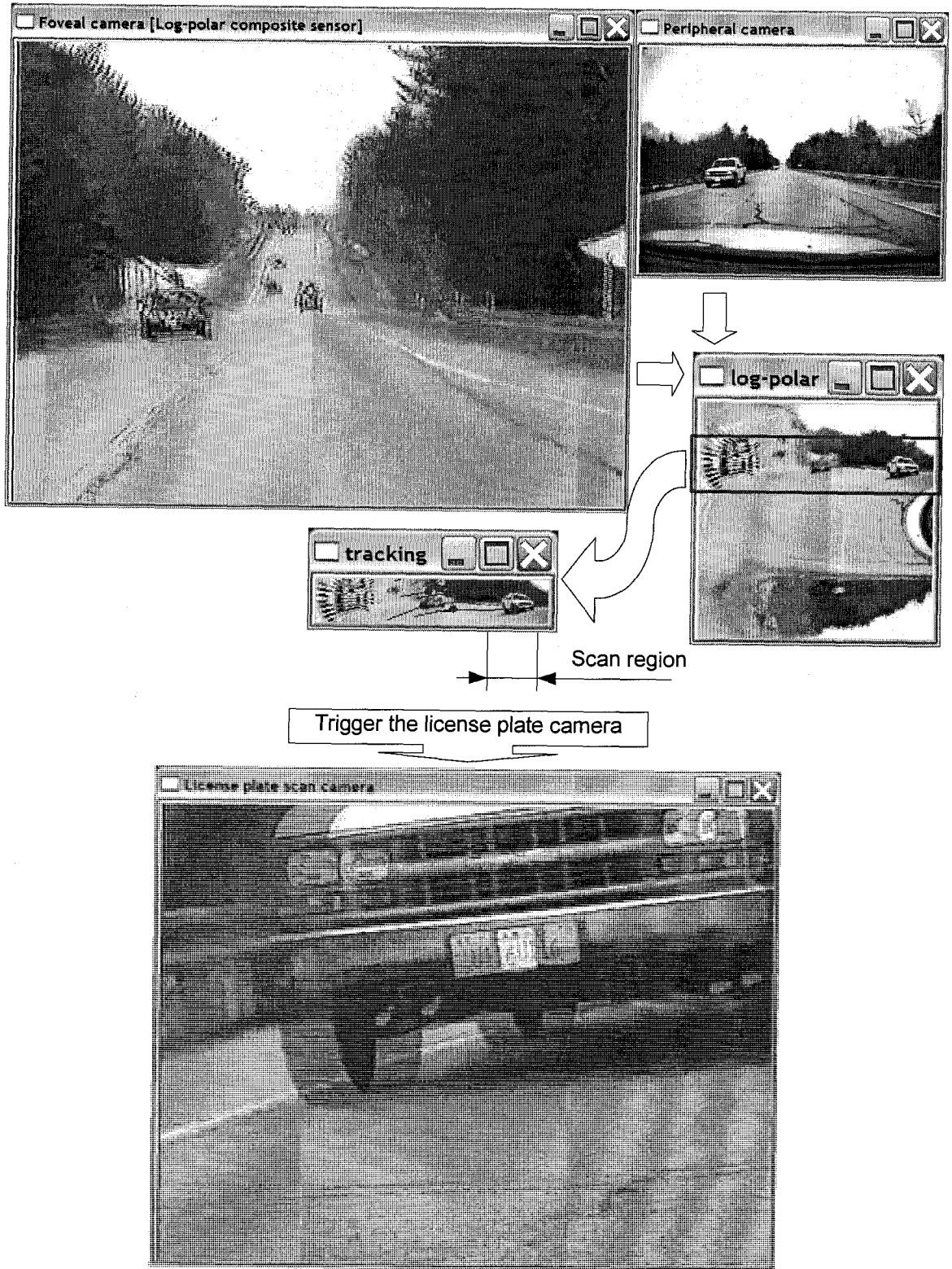
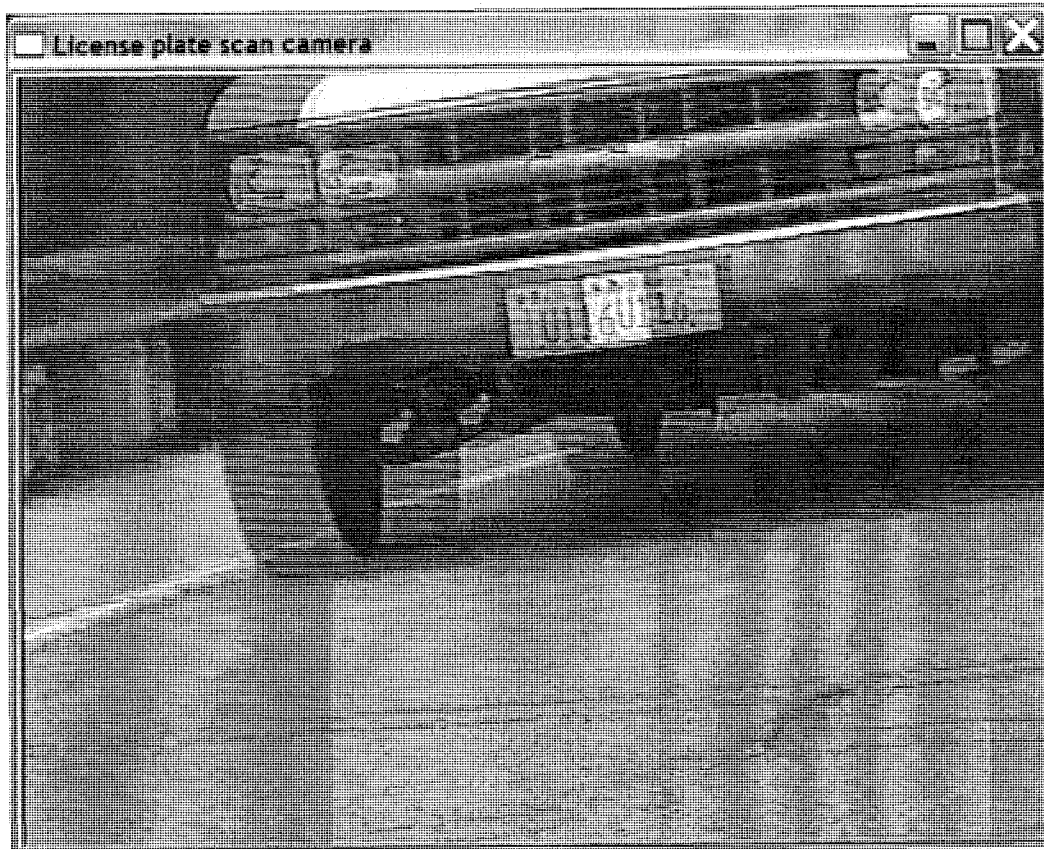
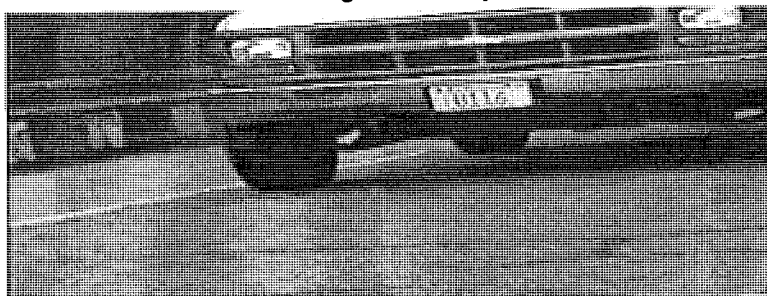


Fig. 32. License plate scan application



Half-sized image formed by odd lines



Half-sized image formed by even lines

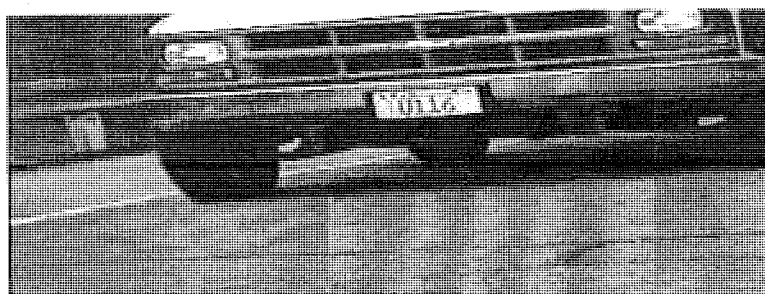


Fig. 33. Deinterlacing via half imaging

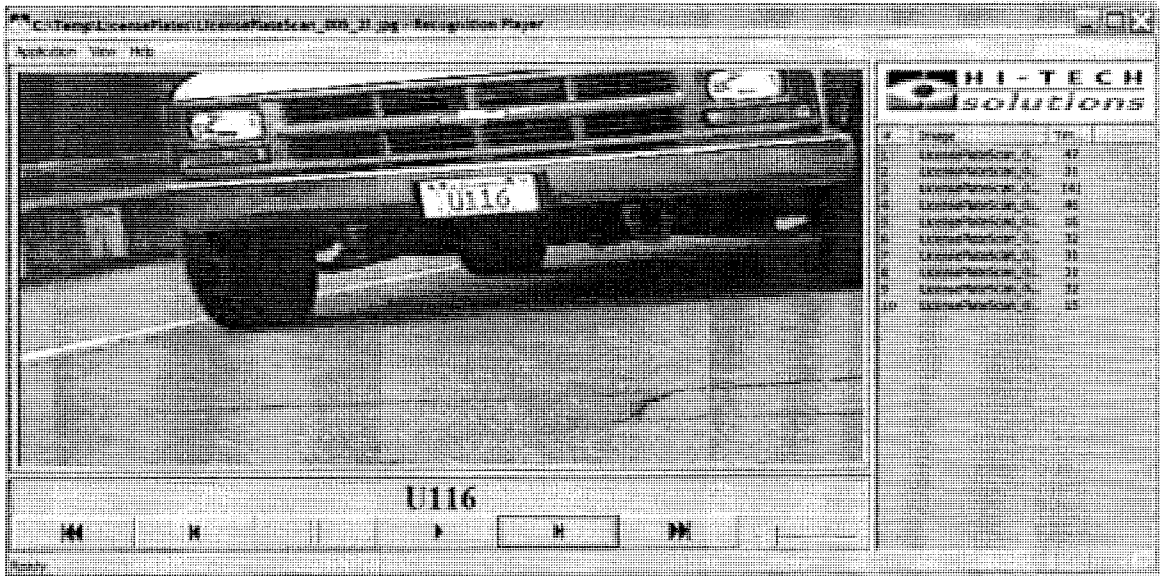


Fig. 34. Third-party license plate recognition software.

In order to demonstrate that the license scan system produces legitimate captures of license plates, the stored images are fed into a third party license plate recognition application that creates a text string out of a license plate number (see Figure 34). The third party software that was used for testing is a demonstration version of the license plate recognition application made by Hi-Tech Solutions [22]. This software is normally used for stop-and-go traffic at automatic gates. Using this as a module to the vehicle detection/tracking system it effectively became the license recognition portion of the proposed mobile application.

The demonstration license plate recognition software depicted in Figure 34 reads the image files previously stored by the license plate scan application and displays the recognized number on the screen. The test demonstrated that for every vehicle the capturing camera made two to four clear snapshots of the

license plates. The recognition software did not read all license plates with 100% accuracy. On average the software was capable to recognize all symbols in one of the snapshots made from the same vehicle. In other snapshots one or several figures or characters were identified erroneously. Although optical character recognition methods and robustness are beyond the scope of this dissertation, it was observed that such misreadings seemed to be directly related to the similarity of letters and/or numbers and the illumination and conditions of the environment at the time of image capture.

CONCLUSIONS

Two significant contributions to the field of synthetic vision are apparent from this dissertation. For the first contribution a new interpretation and view of the classical log-polar transformation is developed via expanding to a third dimension and viewing the z axis as being a mapping in the form of a log funnel. This novel paradigm shows how far visual fields can be seamlessly mapped into a single log-polar composite space. This visualization is new and offers researchers a method to combine far and near visual acuity into one computational space that can support simultaneous computation on both the near field and the far field in a unified way. The second contribution is that this work has demonstrated the benefits of log-polar image representation in a mobile computer vision based application with the ultimate test being of robust high frame rate performance in a “real world” automotive environment.

The novel composite log-polar deep-field image sensor opens new horizons for computer vision. Its theoretically unlimited expandability of depth of field paired with the ability to compress space in a scale tolerant manner makes this image sensor an efficient input device for computer vision applications imaging over large distances. The author has found no other single camera with standard

optics in the literature that can deliver better depth of field performance. In this dissertation the complete composite image sensor theory was developed followed by design guidelines, optics considerations, special calibration procedure and calibration tools which can now be used by other researchers for consideration in their applications.

It has been shown that under correct camera alignment the log-polar image space offers excellent image segmentation and perspective distortion mitigation for far outstretched scenes. In the case of vehicle tracking, this manifests itself in excellent road segmentation and scale tolerant vehicle representation. When the camera alignment is done properly, the center of log-polar transform (i.e., the optical axis of the composite sensor) is coincident with the road vanishing point. To ensure for correct camera alignment, appropriate vanishing point detection and fixation algorithms were developed. The downside for this strict camera alignment requirement is that failure of vanishing point fixation algorithm leads to road deformation in the log-polar space and the advantages in road segmentation and scale-tolerant vehicle representation are lost. The vanishing point fixation algorithm failure may happen on occasions when less than two road lines are detected. An example of such a situation is an unmarked road in the night time. This drawback may be mitigated by coupling other methods for alignment to the vanishing point. This is left as a topic for future work.

The proof-of-the-concept vehicle tracking application demonstrated the following advantages for the use of the composite log-polar image sensor: 1)

extended visual range that allows for vehicle detection at distances unattainable for contemporary computer vision systems [1], 2) convenient road segmentation in the form of horizontal rectangular sections, 3) scale tolerant vehicle representation, and 4) significantly smaller image space that reduces computational load. Furthermore, field tests demonstrated that motion-based vehicle detection and tracking is equally successful in both day time and night time. Preliminary test results showed near 100% vehicle detection and tracking, however extensive testing under different weather conditions in all seasons is recommended for future work. Another topic for further investigation would be looking at the phenomena of the “lost” and “found” features in the optical flow calculation. Such research might offer a method to prevent the tracked features from “slipping off” the vehicles under some observed conditions.

Not only does the composite sensor expand the depth of the field, but it also mitigates perspective distortion and significantly reduces the amount of visual data required to process. This in turn allows for real time tracking of vehicles from the far field to the near field. In the proof-of-the-concept application the input foveal camera image (with resolution of 640x480 pixels) and the input peripheral camera image (with resolution of 320x240 pixels) are collapsed into the joint composite log-polar image. In this log-polar space the incoming lane is extracted and has a resolution of 180x30 pixels. This represents a reduction in the data necessary to be processed of seventy-one. Thus the optical flow technique used for vehicle tracking is also on the order of seventy-one times faster in the log-polar space than if it is implemented in two separate Cartesian images.

The proof-of-the-concept vehicle tracking application developed in this dissertation is only scratching the surface of all possible applications for such a composite deep-field image sensor. From experience gained in this research, a general observation can be made about applications where such a sensor would be suitable. It is observed that this sensor is most effective for tracking objects with high relative speeds moving on linear trajectories parallel to the sensors' optical axis. In a similar manner to the automotive computer vision application presented in this research, the two-camera composite sensor can be used in obstacle detection for high-speed trains (e.g., magnetic levitation trains). Considering that the speeds of the fastest trains exceed 500 km/h the three-camera composite image sensors would be applicable to a system where an operator would want to be alerted to an obstacle in the tracks with sufficient time to slow the train. Such a scenario would necessitate a far field view to be successful.

This current work has not pushed the limits of the composite sensor. Any future work would require the exploration of three-, four-, and possibly greater multi-camera nested composite sensors. It is unlikely that ground-based applications such as vehicle tracking, or obstacle detection for high-speed trains will require more than three-camera composite sensors. It is the author's view that airborne and space borne applications may will benefit greatly from a vast depth of field. An example would be for computer assisted aircraft landing and computer assisted spacecraft docking [39].

APPENDICES

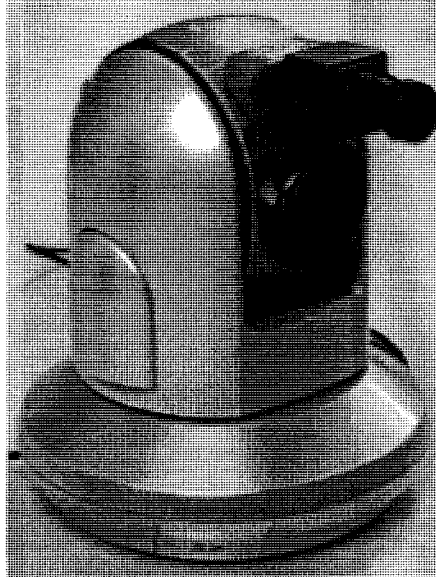
APPENDIX A. COMPOSITE IMAGE SENSOR PROTOTYPES

First “Laboratory” prototype of the composite image sensor.

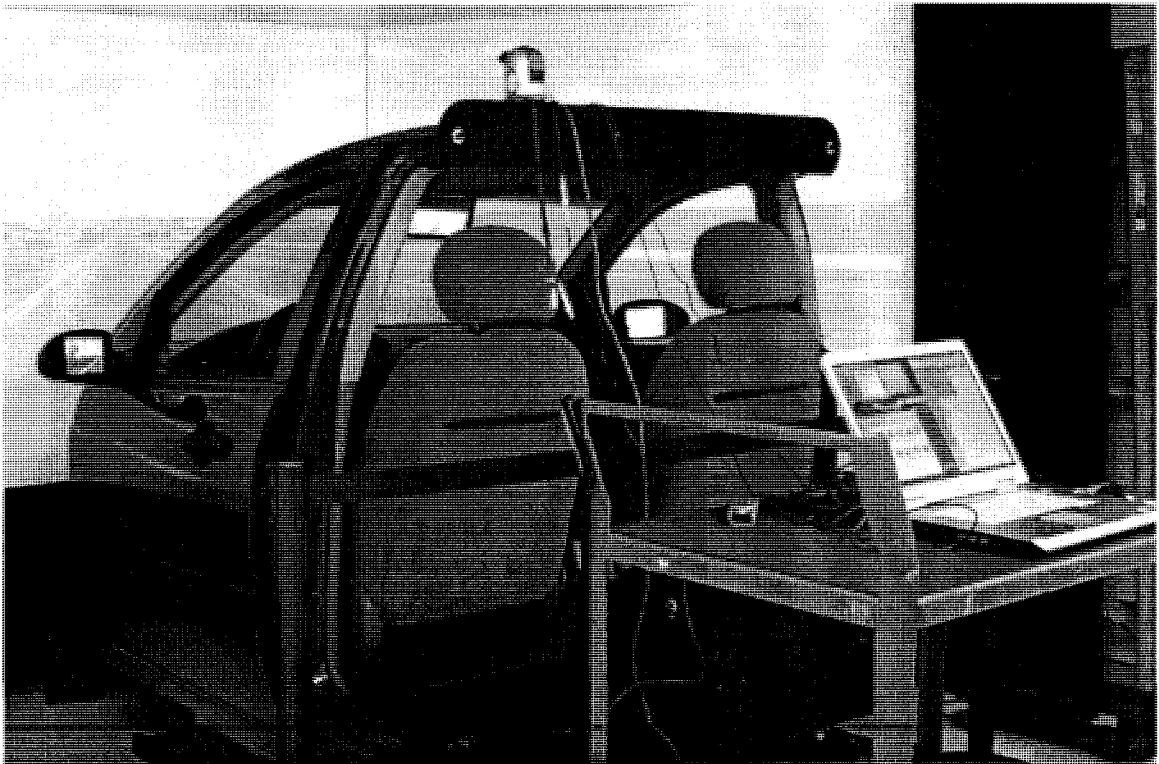
The laboratory prototype of the composite sensor is based on a tilt, pan and zoom (TPZ) network camera Panasonic BB-HCM381A. This camera serves as the foveal camera. It has variable zoom that provides optical magnification up to 21 times and the view angle as small as 2.6°. The peripheral camera is the 30° view angle ultra-compact Panasonic GP-CX161/53 camera. It is mounted above the TPZ camera lens as shown in Figure A.35. Such composite sensor prototype provides a very flexible laboratory platform. For instance the variable zoom was used to match the log base in experiments supporting the composite sensor design theory (see Figure B.38).

Second “Field-Test” prototype of the composite image sensor.

One of the requirement for the “field-test” prototype is rigid and compact design. The choice of cameras was made from rigid surveillance cameras with small “lipstick” form-factor. The peripheral camera has a 98 degrees view angle. Foveal camera has a 10 degrees view angle that provides visual range of approximately 0.22 miles for the vehicle tracking application. Also, both cameras have compact consumer-grade optics rather than bulky and heavy professional lenses. This makes the assembly light and compact and it can be easily mounted behind the windshield.



a) The "laboratory" prototype of 2-camera composite sensor



b) Test setup in Project54 driving simulator

Fig. A.35. Composite sensor "Laboratory" prototype

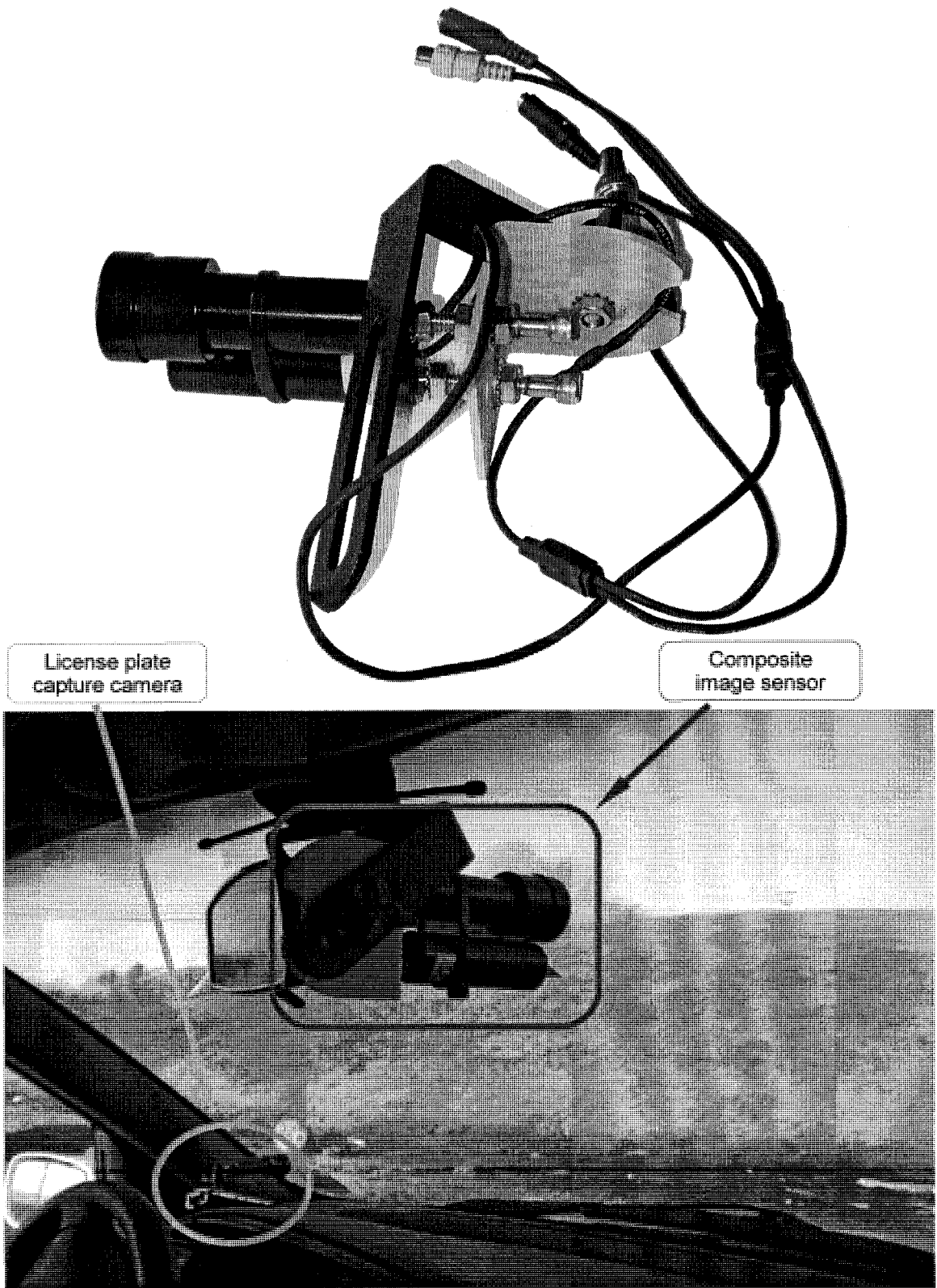


Fig. A.36. The "field-test" composite sensor prototype and installation inside a vehicle

The design of the composite image sensor does not pose any special requirement on cameras other than view angle scaling shown in equation (10). The “field-test” prototype shows that a regular surveillance camera retailer is capable to offer a satisfactory choice of mainstream cameras and optics to build a composite sensor. It is not relevant to mention what camera models were used in the “field-test” prototype because the industry has many alternatives and frequent upgrades. However, it is worth to mention that the total camera cost for the sensor was below \$250.

The proposed two-camera composite image sensor has several look-a-likes. However, they come from areas of computer vision other than vehicle tracking [51], [52]. There are several differences that set aside the proposed image sensor from these similar in appearance non-log-polar foveated two-camera sets.

- 1) They do not target the same goals, i.e. single image output, expandable range and scale invariance.
- 2) These cameras were designed for near field. Their magnification ratio between the foveal and peripheral cameras is about 2x to 4x. In contrast, the magnification ration in the log-polar composite sensor is as much as 10x.
- 3) A user of the composite log-polar sensor is abstracted from the number of cameras in the composite sensor. From the user's standpoint it is a single image sensor with a single output image. In contrast, the look-a-like camera sets are treated as sets of separate cameras where the foveal-to-peripheral view relation is established via relatively computationally expensive tasks of image registration [51] and 3D reconstruction [52].

APPENDIX B. COMPOSITE SENSOR DESIGN THEORY

Secondary oversampling in nested cameras.

As it was shown the composite sensor design section of Chapter 2 it is a good design practice to adjust the foveal camera view to overlap and eliminate the oversampled region of peripheral log-polar image. Naturally, a similar oversampled region also exists in the foveal camera log-polar image. This oversampled region will be further referenced as a *secondary oversampling*.

Similarly to equation set number (7) the equations (12) identify the border circumference of radius ρ_{bFov} measured in Cartesian pixels on the foveal camera sensor, and corresponding value of u_{bFov} in log-polar domain. The last formula in equations (12) shows how foveal border point is positioned in joint log-polar space. The effect of using the foveal camera is shown in Figure B.37. Higher resolution provided by the foveal sensor has reduced the oversampled region but it was not eliminated completely.

$$\begin{aligned}
 \frac{d}{d\rho} \log_a(K_b \cdot \rho_{fov}) &= \frac{1}{\rho_{fov} \cdot \ln(a)} \\
 \frac{1}{\rho_{bFov} \cdot \ln(a)} &= 1 \\
 \rho_{bFov} &= \frac{1}{\ln(a)} \\
 u_{bFov} = \log_a(K_b \cdot \rho_{bFov}) &= \log_a\left(\frac{1}{\rho_{max} \cdot \ln^2 a}\right)
 \end{aligned} \tag{12}$$

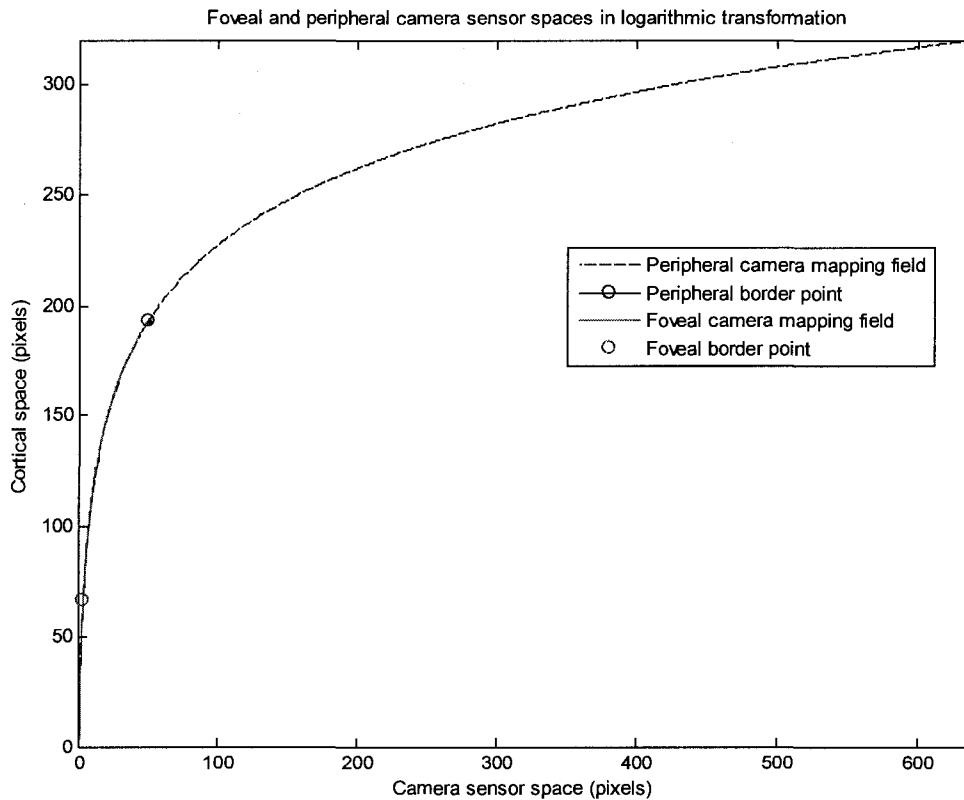


Fig. B.37. Decreasing the oversampled region with foveal camera.

In fact it is possible to influence the size of secondary oversampling by choosing appropriate dimensions for the input and output spaces or, more precisely, by choosing the relation between them. Figure B.38 shows different log-polar mapping schemes and corresponding peripheral and foveal border points. From these diagrams an interesting observation can be made: as the higher resolution input space is collapsed into a smaller resolution output space secondary oversampling is lessened. Actually, with certain parameters the secondary oversampling can be eliminated completely. Given the camera sensor resolution as a design constant, the secondary border point in the output space

Image sensor resolution in foveal and peripheral cameras

640 x 480

320 x 240

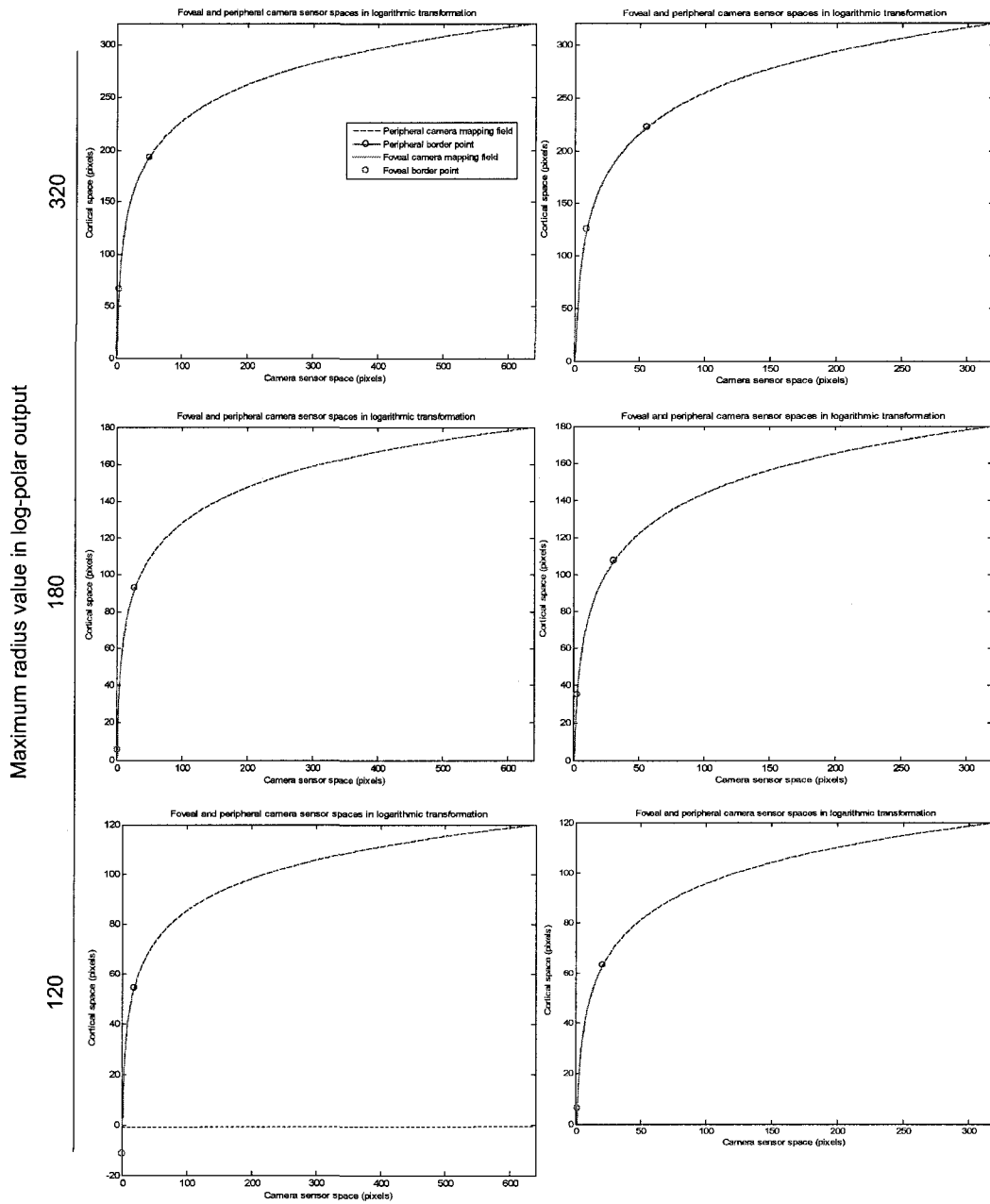


Fig. B.38. Primary and secondary oversampling under different input resolutions

(u_{bFov}) can be made as small as desired by changing the base of logarithmic transformation.

This leads to another design consideration:

The size of the secondary oversampled area can be tuned using formula (12), but now as a function of logarithm base: $u_{bFov}(a) = \log_a\left(\frac{1}{\rho_{max} \cdot \ln^2 a}\right)$

Figure B.39 shows this function for the two input space resolution used in current composite sensor design.

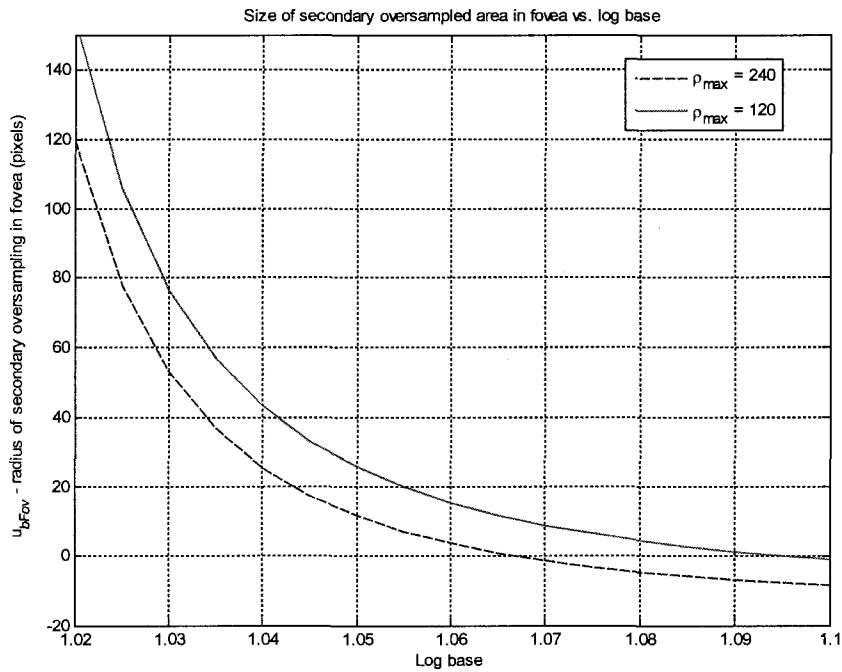
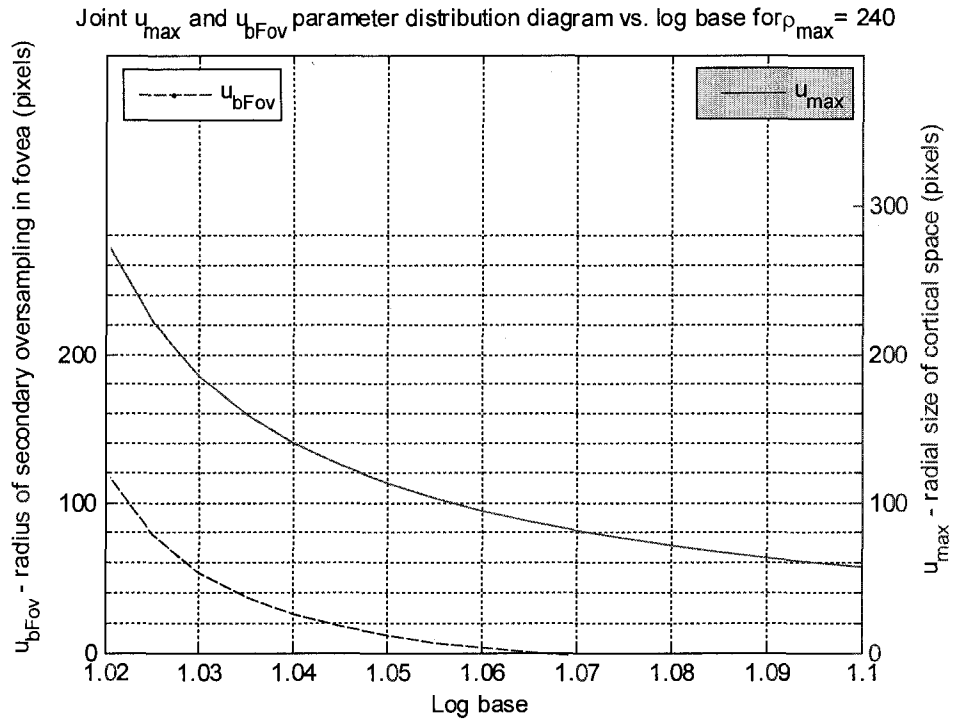


Fig. B.39: Radius of secondary oversampled area as a function of log base

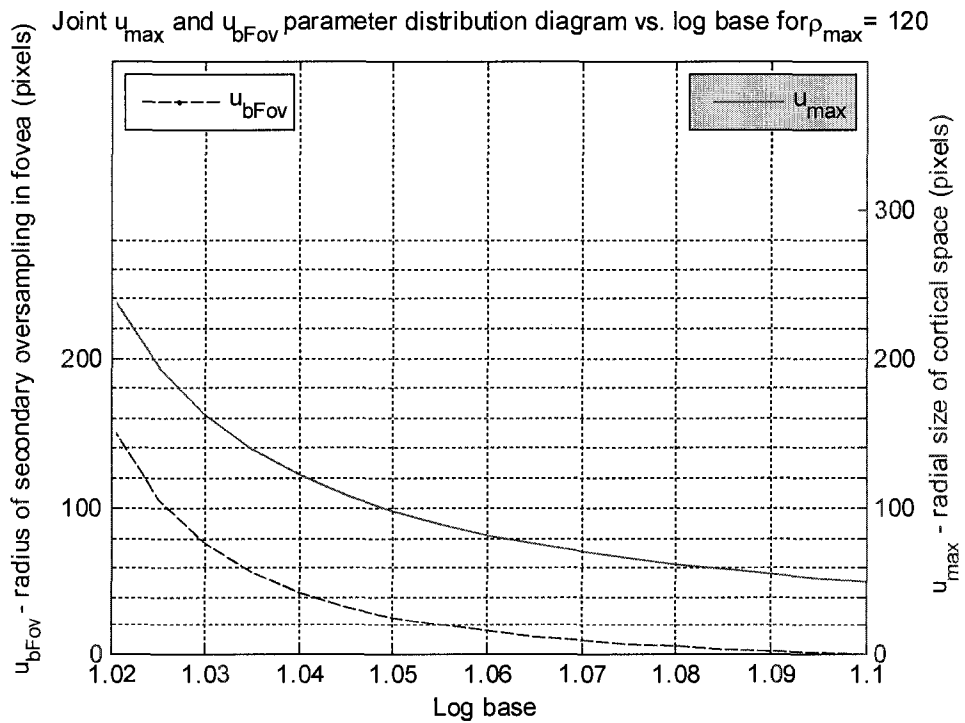
The change of log base will also influence the size of output log-polar space in radial dimension, since $u_{max} = \log_a(\rho_{max})$. In fact there might be a tradeoff between the output space size and the size of secondary oversampled area. To make a design decision the output space must be assessed as a function of the input space size, oversampled area size, and log base value (equation (13)).

$$u_{max} = \frac{u_{bFov} \cdot \ln(\rho_{max})}{\ln\left(\frac{1}{\rho_{max} \cdot \ln^2 a}\right)} \quad (13)$$

Composite Sensor Design Chart is a helpful tool for choosing sensor design parameters (see Figure B.40). This chart offers a quick graphical tool for choosing a combination of key parameters (a , u_{max} , u_{bFov}) based on formulas (12) and (13). Joint-parameter diagrams given in Figure B.40 reflect two popular camera resolutions: (a) 640x480 pixels ($\rho_{max} = 240$) and (b) 320x240 pixels ($\rho_{max} = 120$). These camera resolutions are used in the current composite sensor prototype. Once the parameter combination choice is made, the *foveal camera view angle* should be calculated using formula (10).



a) Design chart for input camera resolution of 640x480



b) Design chart for input camera resolution of 320x240

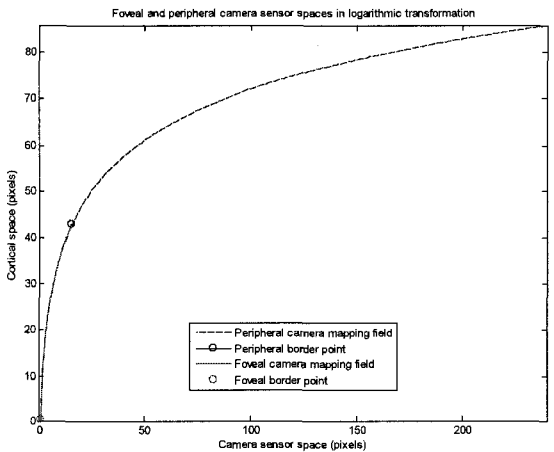
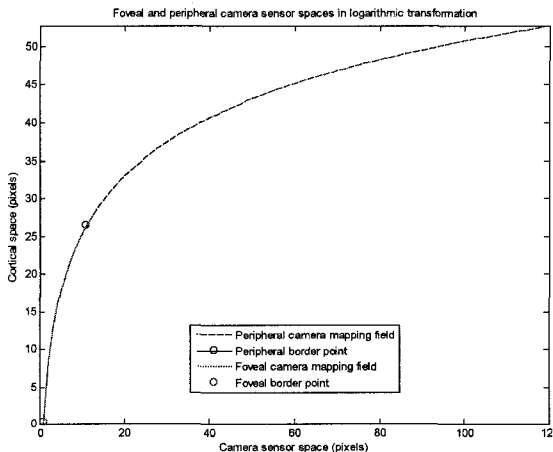
Fig. B.40. Composite sensor design charts.

Custom design charts for arbitrary resolutions can be generated using MATLAB code inside “CompositeSensorDesign.m” file that can be found on the CD/DVD accompanying this dissertation.

Example. Design of a composite sensor with no secondary oversampling in output image. Peripheral camera angle is static and equal to 53.4 degrees. The initial design constrains applied on foveal and peripheral cameras are following:

Case 1: camera resolution is 640x480 pixels.

Case 2: camera resolution is 320x240 pixels.

<i>Case 1: 640x480 input ($\rho_{max} = 240$)</i>	<i>Case 2: 320x240 input ($\rho_{max} = 120$)</i>
<p>From Figure B.40a the log base at $u_{bFov}=0$ is a ~ 1.066</p> <p>Output log-polar space radial dimension: $u_{max} = 86$ pixels</p> <p>Camera angles: $\Theta_{per} = 53.4^\circ$ $\Theta_{fov} = 3.5^\circ$</p>	<p>From Figure B.40b the log base at $u_{bFov}=0$ is a ~ 1.095</p> <p>Output log-polar space radial dimension: $u_{max} = 53$ pixels</p> <p>Camera angles: $\Theta_{per} = 53.4^\circ$ $\Theta_{fov} = 4.9^\circ$</p>
	

APPENDIX C. VANISHING POINT FIXATION IN LOG-POLAR SPACE

The proposed vanishing point fixation method is based on some a priori knowledge about the environment. The road edges and painted lane separation lines are used for estimating the vanishing point location. Since the aforementioned road lines virtually converge at the vanishing point, the log-polar mapping centered on the vanishing point transforms the road edges and lane separation lines into strictly parallel lines (see Figure 23a). If the center of log-polar mapping drifts from the vanishing point, the parallel lines start bending on the log-polar image (see Figure 23b,c,d). This bending has an important property: because the foveal (left) region of the log-polar image is more sensitive to the center point shift in opposition to the peripheral (right) region, the road lines in the log-polar image bend mostly in the foveal region while their peripheral section stay parallel withstanding larger deviations from vanishing point. This property is immediately evident from the geometry of the log-polar transformation: 1) the angular shifts corresponding to the center point displacement are larger with regards to a point in the foveal region opposed to a point in the periphery because the latter is more distant from the log-polar mapping central point; and also 2) because the size of receptive fields in the fovea are smaller than in the periphery (see Figure 5).

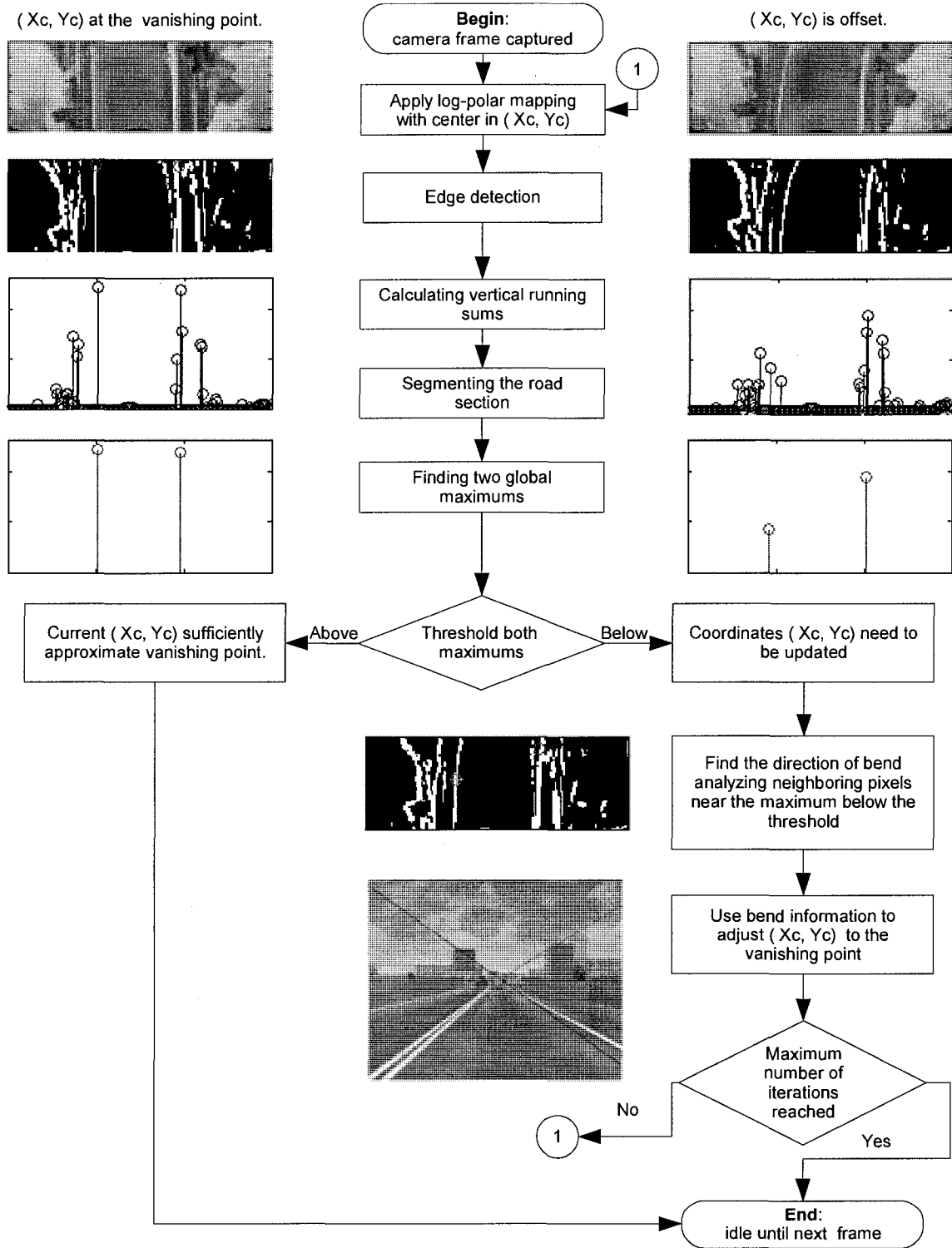


Fig. C.41. The vanishing point fixation algorithm.

The algorithm for vanishing point fixation is described in Figure C.41. Images on the sides of the blocks display actual data at every stage of the image processing pipeline. The notation (X_c, Y_c) is used to describe the center of log-polar mapping applied to the incoming Cartesian images – the video frames captured with a uniform sensor camera. Initial position of the log-polar mapping center is the middle point of the video frame: $X_c = \text{FrameWidth}/2$ and $Y_c = \text{FrameHeight}/2$. It is assumed that the host vehicle is moving straight along the road and the optical axis of the camera is parallel to the road. In the ideal case these conditions are sufficient for the center point of the acquired image be coincident with the vanishing point of road perspective. In practice, the deviations that cause their misalignment are inevitable and persistent. They are caused by vehicle vibration and micro-turns peculiar to the human way of driving via adjusting to the road. These two causes were discovered during our laboratory and field tests. Those tests also showed that the vanishing point deviations have a relatively small amplitude around the center of the image. However, it is impossible to give a definite quantitative evaluation of the vanishing point deviations amplitude, because they entirely depend on particular camera fixture, properties of the pavement, vehicle suspension and possibly other parameters. For the purposes of this paper it is assumed that the vanishing point remains in the central part of the image within a circular region of diameter equal to $1/6$ of the image width.

The algorithm starts with log-polar image acquisition by applying the log-

polar mapping with center at (X_c, Y_c) as shown in Figure C.41. The horizontal edge detection using a Sobel filter followed by gray level thresholding is then applied to the acquired image. This is the first step in extraction of the road lines. The peripheral sections on the road lines remain parallel to each other and perpendicular to the axis v of the log polar grid . This property allows the use of running sums (projection) in the vertical direction of the image for the purpose of detecting straight lines perpendicular to the axis v . In order to identify the best candidates for the road lines, the running sums not belonging to the road section on the log-polar image are segmented out. Then, two global maximums have to be found between the remaining running sums, which correspond to two most significant straight lines perpendicular to the axis v . They are treated then as parallel sections of the two most recognizable road lines. Later in the algorithm the intersection of two road lines in the Cartesian space will estimate the location of the vanishing point.

When the two most prominent parallel segments of road lines are extracted, they are used to assess the deviation of the log-polar mapping center (X_c, Y_c) from the vanishing point. The shorter road lines' parallel segments are, the earlier they bend in the log-polar image, the larger is the deviation from the vanishing point. Therefore, the running sum maximums are thresholded against a reference value. This threshold essentially states that if the parallel road segments are above it, then the center of the log-polar mapping (X_c, Y_c) is in sufficient proximity to the vanishing point. The higher the threshold is set implies the higher the

precision. Here, the threshold is defined as a fraction of the log-polar image size in u dimension, or otherwise a percentage of u_{\max} (u_{\max} is shown in Figure 5). In our experiments the threshold values from 60% to 80% were used.

If thresholding detects a significant deviation from the vanishing point, the algorithm branches into calculation of its better approximation. For every set of two running sum maximums the corresponding pixels in the edge-detected log-polar image are found. Neighborhoods are then searched for the bending fragments in upper left and right directions. Once such “bending neighbor” pixels are acquired, their log-polar coordinates are converted back to Cartesian space, $(X1_{RunMax1}, Y1_{RunMax1})$. Now we need another point on the road line corresponding to the same running sum maximum to derive the (road) line equation in the Cartesian space. For this purposes the $(v_{RunMax1}, u_{\max})$ is taken and its Cartesian coordinates are calculated $(X2_{RunMax1}, Y2_{RunMax1})$. Then these points are used to calculate the line equation in Cartesian space in form of $y = kx + b$ (shown in (14)).

$$\begin{aligned} k_1 &= k_{RunMax1} = \frac{(Y2_{RunMax1} - Y1_{RunMax1})}{(X2_{RunMax1} - X1_{RunMax1})}; \\ b_1 &= b_{RunMax1} = \frac{(X1_{RunMax1} * Y2_{RunMax1} - X2_{RunMax1} * Y1_{RunMax1})}{(X1_{RunMax1} - X2_{RunMax1})}; \end{aligned} \quad (14)$$

Finally, two road line equations in Cartesian space are obtained for each of the parallel road segments (or running sum maximums) found earlier. We will denote them as $y = k_1x + b_1$ and $y = k_2x + b_2$. These line equations describe the two most distinguishable road lines, which may belong to road edges or painted

lane markings. Naturally, the vanishing point would be the intersection of these lines. Hence, the new estimation for the vanishing point, i.e. its coordinates (x_{vp}, y_{vp}) , would be calculated as shown in (3).

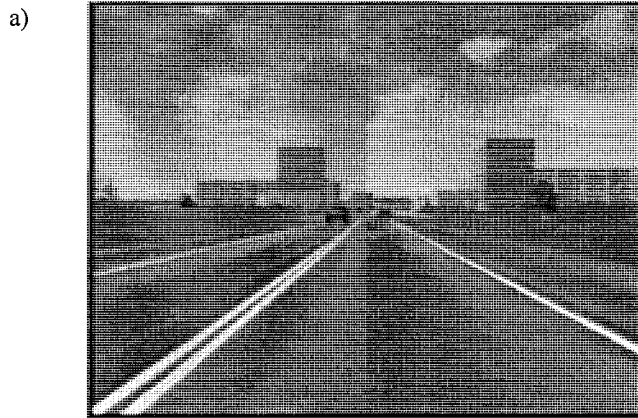
$$x_{vp} = \frac{(b_2 - b_1)}{(k_1 - k_2)}; \quad y_{vp} = \frac{(k_1 \cdot b_2 - k_2 \cdot b_1)}{(k_1 - k_2)} \quad (15)$$

Once the new estimation for the vanishing point is made, the log-polar mapping is applied to the same video frame. The parallelism of road lines on the log-polar image is verified again until a sufficient threshold is reached, or the algorithm is stopped because the maximum number of iterations has been reached. The latter may occur if the input image is of bad quality and the road lines do not produce distinctive edges.

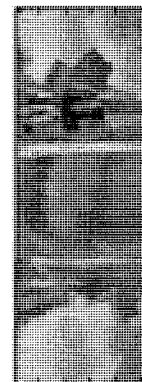
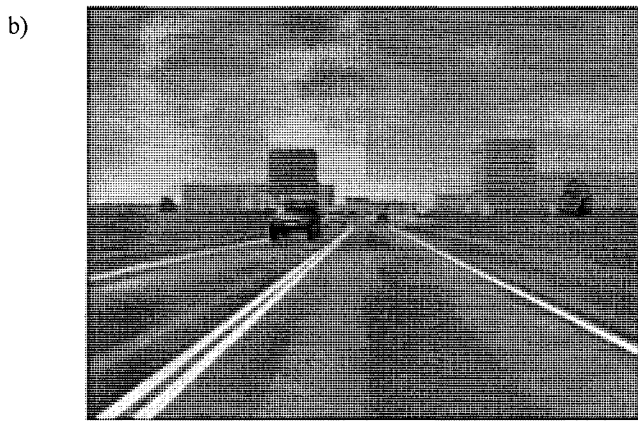
The demonstration of the algorithm is presented in Figure C.42. The input video was acquired in the driving simulator while moving along a two-lane road. The left column shows several time-interval successive video frames from the sequence. The corresponding “motion compensated” log-polar video frames are shown on the right. Since the input video quality is very good and the simulated environment is free from the pavement imperfections, the algorithm performed with high accuracy (near 100%) to achieve the desired vanishing point approximation defined by 60% threshold.

Captured video stream

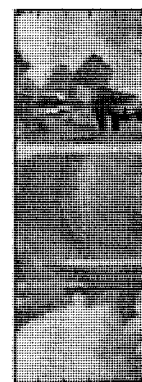
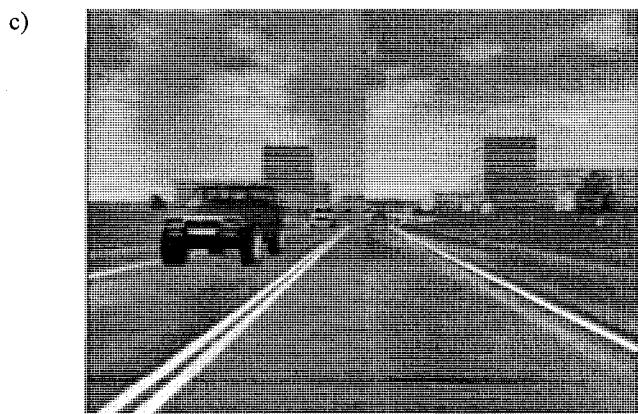
Log-polar video stream



Frame i



Frame $i+5$



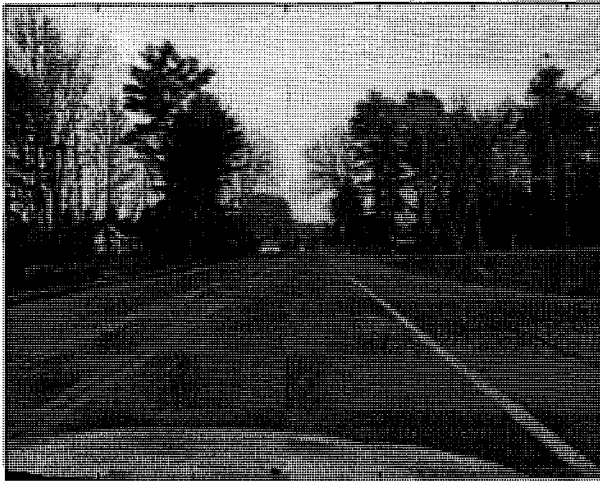
Frame $i+10$

Fig. C.42. Vanishing-point fixation applied to video stream taken in driving simulator.

Captured video stream

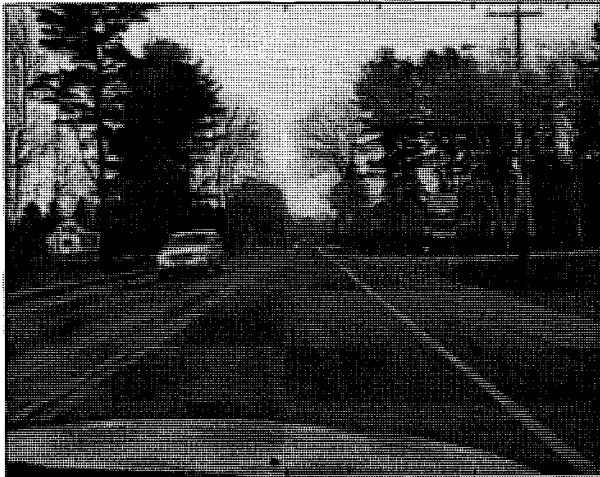
Log-polar video stream

a)



Frame i

b)



Frame $i+5$

c)



Frame $i+10$

Fig. C.43. Field tests of the vanishing point fixation algorithm

In order to complete evaluation of the road follower, it was tested in a video sequences taken in real environment on the road (see Figure C.43). These tests have shown that the algorithm greatly depends on visibility of the road markings and it is sensitive to the lane markings damage and their reflective properties. Weather conditions, quality of ambient illumination, and road marking imperfections may cause problems which diminish algorithm performance. On “good” road sections the algorithm is able to detect the vanishing point as shown in Figure C.43.

APPENDIX D. VANISHING POINT FIXATION IN CARTESIAN SPACE

Like the vanishing point fixation method in log-polar space described in Appendix C the Cartesian method also relies on painted lane separation lines and road edges to detect the vanishing point. The road lines virtually converge at the vanishing point. Therefore, the major stages of detection algorithm are 1) to find the road lines in peripheral camera video frames, 2) find their line equations and 3) calculate the crossing point coordinates.

The block diagram of vanishing point detection algorithm in Cartesian space is shown in Figure D.44. The principal element of this algorithm is line detection based on Hough transform [53]. Before the Hough transform is applied the video frame has to be preprocessed. First, the region of interest in the image is set to the lower part of the video frame which contains the road. Second, the Canny edge detection algorithm [59] is applied to this region. The output of this preprocessing stage is a binary image of the road containing edge information. Typically, the road sides and painted lane dividers produce long linear edges that are well detected by Hough transform. When Hough transform is applied to the road image there are many more lines detected in addition to the road lines and lane dividers. At this point it is necessary to filter the road lines from rest of the lot.

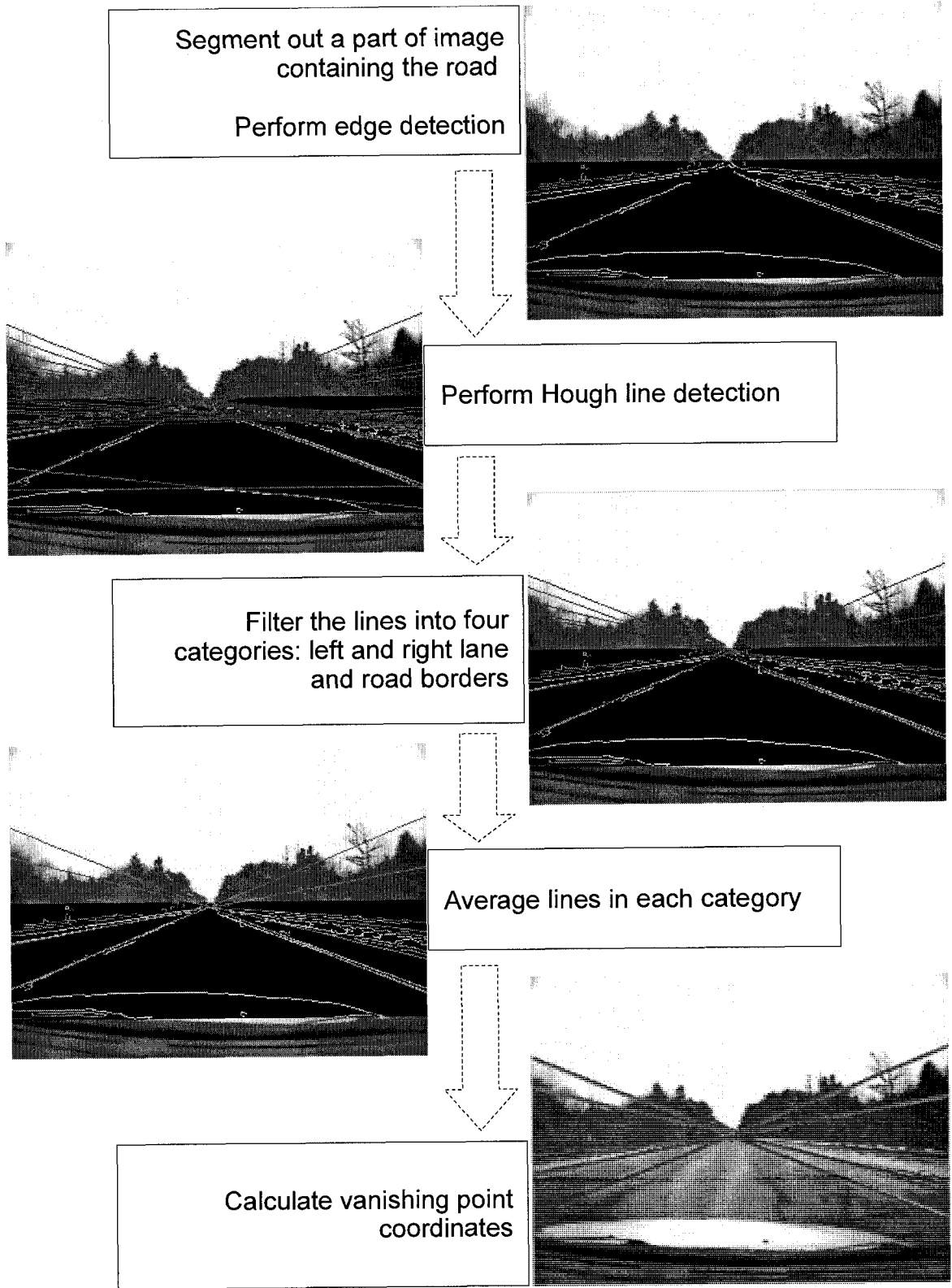


Fig. D.44. Vanishing point fixation algorithm in Cartesian space

The results of Hough transform are presented as a table of line parameters in polar coordinate system, i.e. (ρ, θ) . The elements of this table are sorted into four bins: 1) left road border, 2) left lane divider, 3) right lane divider, and 4) right road border. Each bin has a filter of polar coordinates $(\rho \pm \Delta\rho, \theta \pm \Delta\theta)$ that corresponds to expected location of each aforementioned road line. Those lines in the Hough detection result table which parameters do not match any of the bin's filters are thrown away. Now, each bin may contain zero, one or many lines. No lines in a bin means that it was impossible to detect any line in this location. It often happens to lane dividers because of poor visibility due to physical damage or if those are dashed road dividers. One line in a bin is usually generated by single lane border lines. Most often there are several lines in a bin. They are usually generated by double lined lane dividers and lines and linear textures on the road sides. If there are several lines in a bin their (ρ, θ) parameters are averaged to produce a single representative of this road line as it is shown in the fourth block down on Figure D.44.

$$\theta_x = \text{atan} \left(\frac{\rho_2 \cos \theta_1 - \rho_1 \cos \theta_2}{\rho_1 \sin \theta_2 - \rho_2 \sin \theta_1} \right) \quad (16)$$

$$\rho_x = \frac{\rho_2}{\cos(\theta_x - \theta_2)}$$

The last algorithm stage is to calculate crossing point of the detected road lines. General equation for calculating a crossing point of two lines in polar coordinates is given in (16). Each pair of the road lines generates a candidate for vanishing point. Therefore, if total number of N road lines (where $0 \leq N \leq 4$) is

detected in a video frame, the number of vanishing point candidates is: $(N - 1)!$. Finally, in order to obtain a single vanishing point estimate the coordinates of all candidates are averaged.

Once the vanishing point polar coordinates are calculated they are translated into Cartesian coordinates of the peripheral camera video frame. The log-polar software mapper is then applied with the center at this point as it is shown in Figure 25. Also, the Cartesian vanishing point coordinates are translated into video frame coordinates of the foveal camera. It is made with respect to scale factor K from equation (8) and optical axis coordinate adjustments stored during calibration of the composite sensor. Again, the center of log-polar mapper is applied at the vanishing point in foveal video frame. Together with the peripheral log-polar image they compose the joint log-polar video frame as it is shown in Figure 25.

The field tests of the Cartesian vanishing point detection algorithm have shown excellent results (see Figures D.45 and D.46). The vanishing point estimates were stable and accurate. The quality of vanishing point detection can be judged by appearance of the road lines in composite log-polar image: on straight road they should be parallel in horizontal direction. The video sequences in Figures D.45 and D.46 show peripheral and foveal camera inputs along with corresponding composite log-polar video output. The circles in foveal and peripheral images show the areas around the vanishing points where the log-polar mappers are applied. Also the peripheral camera images show 1) the result

of edge detection applied on the image section containing the road and 2) the road lines detected from these “edge” images. The detected road lines are marked with cyan lines.

The algorithm demonstrated nearly 100% precise vanishing point estimation. It is equally reliable in both day time and night time road tests despite different lighting conditions (see Figures D.45 and D.46). The simple explanation is that the algorithm shall work as long as two of the road lines are visible. In night time the headlights usually provide illumination to make visible at least two reflective lines bounding the immediate lane of movement. Often the headlights provide enough illumination to pick up as many lines as in day light.

The algorithm is rather robust because it is designed to detect four road lines, but under poor visibility it is sufficient to detect only two road lines to estimate the vanishing point location. However there are situations when no road lines are visible. For instance on freshly paved and unmarked road in the night when neither lane dividers nor edges of the road are visible. Same situation may happen if it rains in the night. Rain makes pavement highly reflective, so lane dividers have poor visibility. Possible solution is to use a night vision infrared camera. Such camera should provide reliable visibility in night time and possibly in other harsh weather conditions like fog and heavy snow.

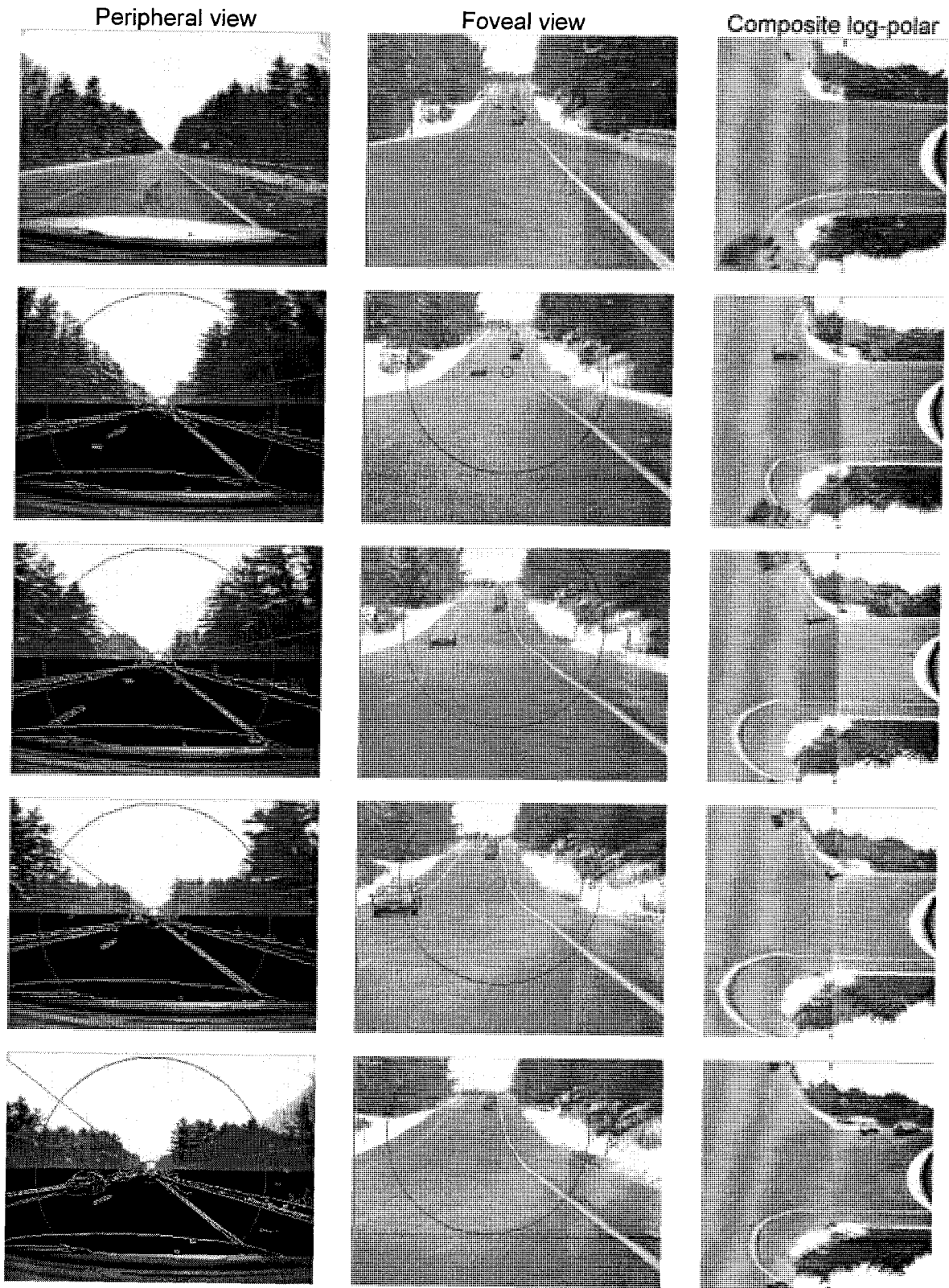


Fig. D.45. Vanishing point fixation in day conditions

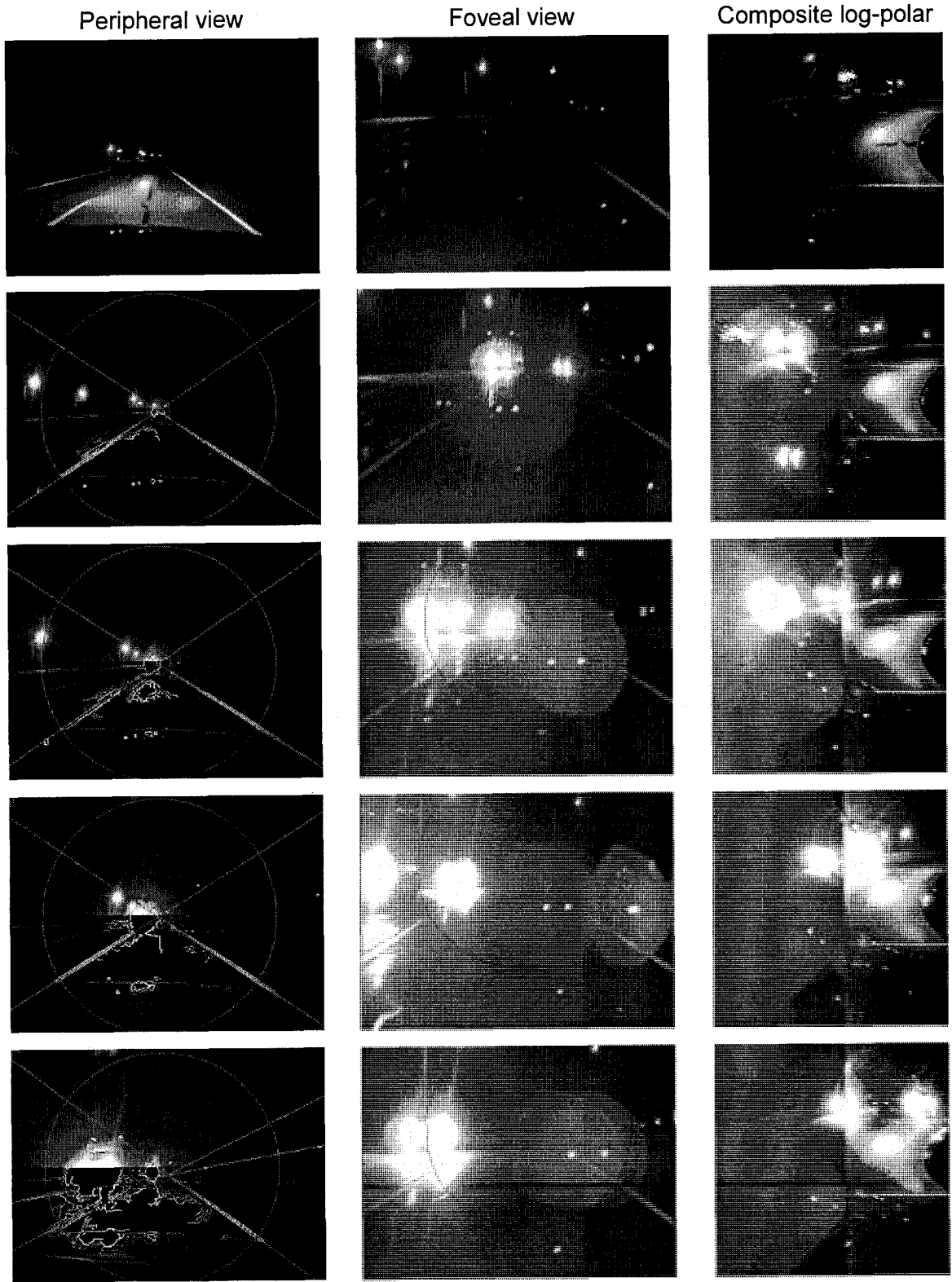


Fig. D.46. Vanishing point fixation in night conditions

REFERENCES

- [1] Zehang Sun; Bebis, G.; Miller, R., "On-road vehicle detection: a review", IEEE Transactions on Pattern Analysis and Machine Intelligence, 5 vol. 28, pp. 694-711, May 2006
- [2] Handmann, U.; Kalinke, T.; Tzomakas, C.; Werner, M.; von Seelen, W. "Computer vision for driver assistance systems", Proceedings of SPIE - Enhanced and Synthetic Vision, vol. 3364, pp. 136-147, July 1998
- [3] McCall, J.C.; Trivedi, M.M. "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation", IEEE Transactions on Intelligent Transportation Systems, 1 vol. 7, pp. 20-37, March 2006
- [4] Pardo, F.; Boluda, J.A.; Coma, I. "High speed log-polar time to crash calculation of mobile vehicles", SERVICEROB 2001. Santorini, Greece, June 2001
- [5] Schwartz E. L. "Spatial mapping in the primate sensory projection: Analytic structure and relevance to perception", Biological Cybernetics, vol. 25, pp. 181-194, 1977
- [6] Giachetti, A. and Campani, M. and Torre, V.; "The use of optical flow for road navigation", IEEE Transactions on Robotics and Automation, 1 vol. 14, pp. 34-48, February 1998
- [7] Anagnostopoulos, C.N.E.; Anagnostopoulos, I.E.; Loumos, V.; Kayafas, E. "A License Plate-Recognition Algorithm for Intelligent Transportation System Applications", IEEE Transactions on Intelligent Transportation Systems, 3 vol. 7, pp. 377-392, September 2006
- [8] Bertozzi M., Broggi A., Fascioli A., Nichele S. "Stereo vision-based vehicle detection", Proceedings of the IEEE Intelligent Vehicles Symposium, Dearborn, MI, USA, pp. 39-44, October 3-5, 2000
- [9] "Automatic license plate recognition", Civica Software, www.platescan.com
- [10] Park, S.; Kim, T.; Kang, S.; Heon, K. "A novel signal processing technique for vehicle detection radar", Microwave Symposium Digest, 2003 IEEE MTT-S International, vol. 1, pp. 607-610, June 2003
- [11] Wang, C.; Thorpe, C.; Suppe, A. "LADAR-based detection and tracking of moving objects from a ground vehicle at high speeds", Intelligent Vehicles Symposium, 2003. Proceedings. IEEE, vol. 1, pp. 416-421, June 2003
- [12] Hancock, J.; Hoffman, E.; Sullivan, R.; Ingimarsen, D.; Langer, D.; Hebert M. "High-Performance Laser Ranger Scanner", Proceedings of SPIE on Intelligent Transportation Systems, vol. 3207, pp. 40-49, January 1998
- [13] Snyder A.W., Miller W.H. "Telephoto lens system of falconiform eyes",

- Nature, 5676 vol. 275, pp. 127 -129, 1978
- [14] Tucker V.A. "The deep fovea, sideways vision and spiral flight paths in raptors", JOURNAL OF EXPERIMENTAL BIOLOGY, 24 vol. 203, pp. 3745-3754, Dec. 2000
- [15] Land M.F., Nilsson D.E. Animal Eyes. Oxford University Press, 2002
- [16] Marr, D. "The computation of lightness by the primate retina", Vision Res., vol. 14, pp. 1377-88, December 1974
- [17] Pirenne, M.H. Vision of the Eye. London Chapman and Hall, 1967
- [18] Levine M.D. Vision in Man and Machine. Addison-Wesley, 1984
- [19] Traver, V. J. and Pla, F. "Designing the lattice for log-polar images", DISCRETE GEOMETRY FOR COMPUTER IMAGERY, PROCEEDINGS, vol. 2886, pp. 164-173, 2003
- [20] Pavlo B. Melnyk and Richard A. Messner "Biologically motivated composite image sensor for deep-field target tracking", Proceedings of SPIE, The international society for optical engineering, pp. 649905, January, 2007
- [21] Pavlo B. Melnyk and Richard A. Messner "Log-polar-based framework for mobile vehicle tracking with road follower", Proceedings of SPIE, The international society for optical engineering, pp. 65790N, May, 2007
- [22] "Optical character recognition solutions.", Hi-Tech Solutions, <http://www.htsol.com/>
- [23] Melnyk, P.B. "*Mobile digital video system for law enforcement*", Master's Thesis, Department of Electrical and Computer Engineering, University of New Hampshire, 2003
- [24] Messner, R.A. and Melnyk, P.B. "Mobile digital video system for law enforcement", Vehicular Technology Conference, 2002
- [25] Messner, R.A. "*Smart visual sensors for real-time image processing and pattern recognition based upon human visual system characteristics*", Dissertation, Clarkson University, 1984
- [26] Vidacic, D. "*Biologically Inspired Computer Vision Model for Pattern Analysis*", Ph.D. Proposal, Department of Electrical and Computer Engineering, University of New Hampshire, 2005
- [27] Shah, S. and Levine, M.D. "Visual information processing in primate cone pathways. I. A model", IEEE Transactions on Systems, Man and Cybernetics, Part B, 2 vol. 26, pp. 259-274, April 1996
- [28] Shah, S. and Levine, M.D. "Visual information processing in primate cone pathways. II. Experiments", IEEE Transactions on Systems, Man and Cybernetics, Part B, 2 vol. 26, pp. 275-289, April 1996

- [29] Bolduc, M. and Levine, M.D. "A review of biologically motivated space-variant data reduction models for robotic vision.", *Computer Vision and Image Understanding*, vol. 69, pp. 170-184, February 1998
- [30] Franceschini F., Pichon J.M., Blanes C. "From insect vision to robot vision", *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* [0962-8436], 1281 vol. 337, pp. 283-294, 1992
- [31] Barnes W.J.P., Johnson A.P., Horseman G.B. "Computer-aided studies of vision in crabs", *Marine behaviour and physiology*, vol. 35, pp. 37-56, 2002
- [32] Vincent, B. T. and Baddeley, R. J. and Troscianko, T. and Gilchrist, I. D. "Is the early visual system optimised to be energy efficient?", *NETWORK-COMPUTATION IN NEURAL SYSTEMS*, 38751 vol. 16, pp. 175-190, 2005
- [33] Gonzalez R.C., Woods R.E. *Digital Image Processing*. Prentice Hall, 2002
- [34] Tistarelli, M., Sandini, G. "On the advantages of polar and log-polar mapping for direct estimation of time-to-impact from optical flow", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4 vol. 15, pp. 401-410, April 1993
- [35] Bartlett, Sandra L., Jain, Ramesh C "Depth determination using Complex Logarithmic Mapping", *Proceedings of SPIE*, vol. 1382, pp. 3 - 13, February 1991
- [36] Stasse, O. and Kuniyoshi, Y. and Cheng, G. "Development of a biologically inspired real-time visual attention system", *Proceedings in Biologically Motivated Computer Vision*, vol. 1811, pp. 150-159, 2000
- [37] Bernardino, A. and Santos-Victor, J. "A binocular stereo algorithm for log-polar foveated systems", *Proceedings in Biologically Motivated Computer Vision*, vol. 2525, pp. 127-136, 2002
- [38] Metta G., Gasteratos A., Sandini G. "Learning to track colored objects with log-polar vision", *Mechatronics*, 9 vol. 14, pp. 989-1006, November 2004
- [39] Bailey J.G., Messner R. A. "Docking target design and spacecraft tracking system stability", *SPIE - Intelligent Robots and Computer Vision VIII: Algorithms and Techniques*, vol. 1192, pp. 820-831, March 1990
- [40] Traver, V. J. and Pla, F. "The log-polar image representation in pattern recognition tasks", *Pattern Recognition and Image Analysis, Proceedings*, vol. 2652, pp. 1032-1040, 2003
- [41] Zokai, S., Wolberg, G. "Image registration using log-polar mappings for recovery of large-scale similarity and projective transformations", *IEEE Transactions on Image Processing*, 10 vol. 14, pp. 1422-1434, October 2005
- [42] G. Sandini and P. Questa and D. Scheffer and B. Dierickx and A. Mannucci. "A retina-like CMOS sensor and its applications", *Proceedings of: 1st IEEE SAM Workshop*, 2000

- [43] Etienne-Cummings R., Van der Spiegel J., Mueller P., Zhang MZ. "A foveated silicon retina for two-dimensional tracking", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II-ANALOG AND DIGITAL SIGNAL PROCESSING, 6 vol. 47, pp. , June 2000
- [44] Kim, W. C. and Kim, J. H. and Lee, M. and Shin, J. K. and Yang, H. S. and Yonezu, H. "Smooth pursuit eye movement system using artificial retina chip and shape memory alloy actuator", IEEE Sensors Journal, 3 vol. 5, pp. 501-509, June 2005
- [45] Szu, H.H. and Messner, R.A. "Image Processing Architecture for Real Time Generation of Scale and Rotation Invariant Patterns", Computer Vision, Graphics, and Image Processing, vol. 31, pp. 50-66, 1985
- [46] Szu, H.H. and Messner, R.A. "Adaptive invariant novelty filters", Proceedings of the IEEE, 3 vol. 74, pp. 518-519, March 1986
- [47] Luengo-Oroz, M. A. and Angulo, J. and Flandrin, G. and Klossa, J. "Mathematical morphology in polar-logarithmic coordinates. Application to erythrocyte shape analysis", Pattern Recognition and Image Analysis, pt 2, Proceedings, vol. 3523, pp. 199-206, 2005
- [48] Cobos, P. and Monasterio, F. "FPGA implementation of a Log-polar Algorithm for real time Applications", XIV Design of Circuits and Integrated Systems, pp. 63-68, November, 1999
- [49] Blasco, F. and Pardo, F. and Boluda, J.A. "A FPGA based PCI bus interface for a real-time log-polar image processing system", XIV Design of Circuits and Integrated Systems, pp. 379-384, November, 1999
- [50] "Retina-like sensors", Laboratory for Integrated Advanced Robotics, <http://www.liralab.it/projects/logpolar/retina.html>
- [51] Scassellati, B. "A binocular, foveated active vision system", MIT, Artificial Intelligence Laboratory, Tech. Rep. A.I. Memo No. 1628, 1999
- [52] Ude, A.; Gaskett, C.; Cheng, G.; "Foveated vision systems with two cameras per eye", Proceedings of IEEE International Conference on Robotics and Automation ICRA 2006, pp. 3457-3462, May 15-19, 2006
- [53] Richard O. Duda and Peter E. Hart "Use of the Hough transformation to detect lines and curves in pictures", Commun. ACM, pp. 11-15, January, 1972
- [54] Smith, S. "ASSET-2: Real-Time Motion Segmentation and Object Tracking", Real-Time Imaging, 1 vol. 4, pp. 21-40, February 1998
- [55] Heisele, B.; Ritter, W. "Obstacle detection based on color blob flow", Proceedings of the IEEE Intelligent Vehicle Symposium, vol. 1, pp. 282-286, September 1995

- [56] Itti, L.; Koch C.; Niebur E. "A Model of Saliency-Based Visual Attention for Rapid Scene Analysis", IEEE Transactions on Pattern Analysis and Machine Intelligence, 11 vol. 20, pp. 1254-1259, November 1998
- [57] Lucas, B. D. and Kanade T. "An Iterative Image Registration Technique with an Application to Stereo Vision", International Joint Conference on Artificial Intelligence, 1981
- [58] "Color television, NTSC tutorials", National Television System Committee, <http://www.ntsc-tv.com>
- [59] Canny, J.F. "A computational approach to edge detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, 6 vol. 8, pp. 679-698, November 1986