Aalborg University Copenhagen

Department of Medialogy

Aalborg University Copenhagen
Lautrupvang 15, 2750 Ballerup, Denmark

Secretary: Dortea Koldborg Jepsen
Phone: 9635 2471
dkj@media.aau.dk
https://internal.media.aau.dk/

Semester: 10

Title: Signal-Symbol Feedback Process in a Hierarchical Vision Architecture Based on Model and Motion Predictions

Project Period:

From: 2007-09-01

To: 2008-03-30

Semester theme:

N/A

Supervisor(s):

Norbert Krüger,

Volker Krüger

Name: Shi Yan

Copies: 4

Pages: 66

Finished:

Abstract:

The studies of image processing and computer vision have been attempting to simulate human visual perception for decades to extract features and recognize objects from the images. Within these domains, the current work aims at improving feature extraction by employing feedback process from predictions of model-based features or motions to optimize signal-based features. The motion predictions utilize symbol-based features to employ feedback process. We represent 3D circles as model-based features in the current work. The 3D Circle models are computed with a sophisticated approach via 2D ellipses. An ellipticity-check function ensures that invalid 3D circle hypotheses are eliminated. A feedback mechanism is proposed that spatially propagates 3D symbol-based features or 3D model-based features to signal-based features via a three dimensional Gaussian function. Moreover, the updated signal triggers the re-creation of features. Such processes form a loop, which is called "signal-symbol loop". Both artificial and real sequences demonstrate the performance of the feedback mechanism.

Signal-Symbol Feedback Process in a Hierarchical Vision Architecture Based on Model and Motion Predictions

SHI YAN, M.SC. MEDIALOGY, AUK, SYAN06@IMI.AAU.DK

Supervised by: Norbert Krüger, Volker Krüger

Abstract:

The studies of image processing and computer vision have been attempting to simulate human visual perception for decades to extract features and recognize objects from the images. Within these domains, the current work aims at improving feature extraction by employing feedback process from predictions of model-based features or motions to optimize signal-based features. The motion predictions utilize symbol-based features to employ feedback process. We represent 3D circles as model-based features in the current work. The 3D Circle models are computed with a sophisticated approach via 2D ellipses. An ellipticity-check function ensures that invalid 3D circle hypotheses are eliminated. A feedback mechanism is proposed that spatially propagates 3D symbol-based features or 3D model-based features to signal-based features via a three dimensional Gaussian function. Moreover, the updated signal triggers the re-creation of features. Such processes form a loop, which is called "signal-symbol loop". Both artificial and real sequences demonstrate the performance of the feedback mechanism.

Acknowledgments

The master thesis was performed at Aalborg University Copenhagen, began in September 2007 and ended in March 2008.

Special thanks go to my supervisor, Norbert Krüger, who guided me through bachelor thesis and whole master studies. His support is always inspiration of my work, especially when I was frustrated and lost. It was simply a pleasure to discuss science with him on philosophical level. Having a chance to spend two years of my life in his research group was a remarkable experience.

I would also like to thank my cooperative supervisor Volker Krüger. Without him, my studies of Master of Medialogy could not have been done. Furthermore, I am grateful to Emre Baseski, Sinan Kalkan, and Florian Pilz for their persistent guidance, support, and friendship. Thanks also go to Dan Rydzewski and Jing Yang for their fruitful discussions and reviews.

Finally, I would like to thank my mother, Qingfeng Xing whose encouragement made my European studies achievable, and Mengyang Cui for her patience.

Definition and Abbreviations

Term	Description
2D/3D	Two dimensional/three dimensional
RBM	Rigid Body Motion
i_{0D} / i_{1D}/ i_{2D}	Intrinsic zero/one/two dimensionality
π	2D Primitive
Π	3D Primitive
ROC	Receiver Operating Characteristics
MoInS	Modality Integration Software
CoViS	Cognitive Vision Software

Table of Content

Signal-Symbol Feedback Process in a Hierarchical Vision Architecture Based on Model and Motion Predictions....2
Acknowledgments....3
Definition and Abbreviations....4
1 Introduction....11
1.1 Background....11
1.2 Objectives....13
1.3 Problem Analysis....15
1.4 Delimitations....17
1.5 Outline....18
1.6 Contribution....19
2 Hierarchical Visual Process....20
2.1 Introduction....20
2.2 Filtering layer....21
2.3 Intrinsic Dimensionality Layer....24
2.4 2D Primitive Layer....25
2.5 3D Primitive Layer....26
3 Circle Model....28
3.1 Introduction....28
3.2 3D Contours....28
3.3 Crude 2D Ellipse Models....28
3.4 Optimized 3D Circle Models....30
3.5 Ellipticity-Check Function....31
4 Feedback Model....34
4.1 Introduction....34
4.2 Feedback Mechanism....34
4.2.1 Feedback using Motion Information....36

4.2.2 Feedback by Model-Based Features 37
4.3 Feedback Model Generalization 39
5 Experimental results 41
5.1 Feedback from RBM 41
5.1.1 Artificial Scene 41
5.1.2 Real Scene 44
5.2 Feedback from Models 45
5.2.1 Artificial Scene 45
5.2.2 Real Scene 47
Circle with low-contrast parts 47
Complex Scene with Multiple Circles 48
5.3 Feedback from Both Motion and Models 51
5.3.1 Artificial Scene 51
5.3.2 Real Scene 52
6 Discussion 55
Appendix 57
A. Software Design 57
A.1 Introduction 57
A.2 Detailed Design Specification 57
A.2.1 KJet 58
A.2.2 CircleDetector 59
A.2.3 SignalSymbolLoop 59
A.2.4 SignalSymbolLoop_ Circle 60
A.2.5 SingleView 60
A.2.6 StereoView 61
B. More results for circle detections 62
B.1 Scene 1 62
B.2 Scene 2 63

References.....64

List of Figures

Fig. 1 Adjust sensitivity cannot solve the contradictories between completeness and spuriousness [11].....12

Fig. 2 The workflow diagram includes four states, indicated by blocks: input images, signal-based features, symbol-based features and models and four processes, indicated by arrows. The arrows above the blocks refer to a feed-forward process while the arrow beneath the blocks refers to a feedback process.....15

Fig. 3 The 3D Primitives and 2D Primitives attached to the 3D circle and 2D ellipse. The dashed lines indicate missing Primitives due to low contrast. In this example, the ellipse model is biased from the projected ellipse. Such problems are solved by the circularity-check function. 17

Fig. 4 Overview diagram presenting the visual process hierarchy modified based on [19]. To enhance the appearance, only three Gabor filter responses are shown. 20

Fig. 5 The Gabor wavelet kernels and the filter responses of a circle image. The top two rows denote the real and imaginary kernels (masks) in eight directions respectively. The bottom row denotes the magnitude of the resulting filtered images.....22

Fig. 6 Symbolic representation of an image with boxes. 26

Fig. 7 A center-positioned ellipse. 29

Fig. 8: Point o is in the center of the ellipse, p is a given point, and l is a line crossing o and p. q is the closing intersection point to p between l and the ellipse. p, q and o are located on line l. 32

Fig. 9 The elimination processes of hypothesized circles models. 33

Fig. 10 The projected ellipse is computed by the Primitives that are extracted from the input image. Feedback is also imposed by using Primitives.....38

Fig. 11 From left to right shows the left and right artificial input images and the

wireframe of right image respectively. The first row and second row indicate the first frame and the second frame correspondingly. 41

Fig. 12 Extracted Primitives and true-/false- positives without and with feedback based on motions 42

Fig. 13 ROC curve analysis in (c) which is made by the true-positives and false-positives shown in (a) and (b). 43

Fig. 14 Input images for testing feedback process with RBM in a real scene. 44

Fig. 15 Results for testing feedback process based on motions in areal scene. 45

Fig. 16 The left and right input images of a 3D circle in an artificial scene. 46

Fig. 17 The predicted 3D circle model is projected back to the input images in green for an artificial scene. 46

Fig. 18 a) and b) are the left and right input images for a driving scenario in a laboratory setting. The potential task is to understand the traffic sign as illustrated in c) and e). In d) and f), the extracted circle models are displayed. 47

Fig. 19 Primitives of the enlarged traffic sign area without and with feedback 48

Fig. 20 Results for a complex scene containing co-existing multiple circles. 50

Fig. 21 Transformed 3D circle models by RBM projected on the original frame in an artificial scene. 51

Fig. 22 Cropped Primitives extracted without and with feedback based on models 52

Fig. 23 Snapshots of feedback over four selected channels of the left image. 52

Fig. 24 Results for testing a circle model with RBM in a real scene. 54

Fig. 25: The abstract class diagram includes classes made for the signal symbol loop in the current work. The classes with shading are the core classes implemented in the current work. The classes marked with dash lines are the auxiliary classes supporting the core classes. The classes in the ordinary lines are the ones implemented in MoInS that are considered to be important

classes for the current work. 58

Fig. 26 Extra Results of Scene 1 for testing circle detection. 62

Fig. 27 Extra results of Scene 2 for testing circle detection. 63

1 Introduction

1.1 Background

Humans perceive visual information with binocular vision from which depth information can be reconstructed. Complex human brain structures allow for the performance of challenging visual tasks, giving humans the capabilities to recognize known object models rapidly and to quickly learn to recognize previously unknown object types. In the computer vision domain, inputs are typically images obtained from cameras. Unlike humans who have strong visual perception, the captured images from cameras are commonly used to solve the computational tasks related to a specific application.

Some approaches have been invented to extract features by using a local neighborhood operator, such as a Canny edge detector[1], Susan corner detector [2], Hough detector for line segment or circles [3][4], etc. Many algorithms deal with image sequences, where discrete images can be used to compute the motion of objects [5]. Motion analysis has been an interesting research area, as it provides correlation not only to the spatial domain, but also to the time domain [6][7]. Feature extraction can be also utilized in 3D, such as image based rendering, surface analysis, etc [8].

One can even use other sensory modalities besides vision to assist in perceiving and recognizing objects and behaviors. Ernst and Banks investigated various techniques for improving perception based on combining the haptical modality with the visual modality [9]. Bayerl and Neumann researched various phenomena in the bio-neural science discipline, and, discovered a modulatory excitatory feedback mechanism to retrieve motion using long-term propagations in an automated manner [10].

A common problem in feature extraction is how to determine thresholds for a specific application, in order to have optimum results with a minimal of flaws. However, adjusting sensitivity cannot solve the problem due to the contradictions between completeness and spuriousness. [11] described this problem in Fig. 1. The completeness increased by turning up sensitivity allows for greater spuriousness.

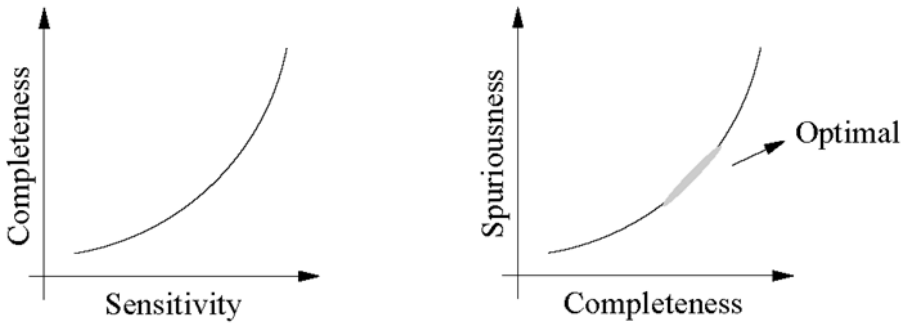


Fig. 1 Adjust sensitivity cannot solve the contradictories between completeness and spuriousness [11].

[12] raised three dilemmas between symbol-based [1] and signal-based visual representations. The dilemmas are induced by the opposing characteristics in unit size, accuracy, completeness, stability and entropy. Feedback from symbol-based visual representation to signal-based visual representation is the recommended method used to solve the dilemmas. Many applications use a bottom-up approach to extract features from visual processes. In visual processes, the incomplete features produced in an earlier process limit the latter processes' abilities to retrieve the lost features. The current work attempts to design a feedback process from model and motion predictions to enhance low-level image processing level. In [13], symbol-based features are recommended to resolve the ambiguities.

[1] The symbol-based representation in the paper is computed from signal-based representation by a bottom up approach.

1.2 Objectives

The definition of the term "features" varies depending on its application. Recalling the definition of feature in Computer Vision, a feature is information that is relevant for solving the computational task, related to a certain application, and can be referred to as:

- The result of a neighborhood operation applied to an image
- Specific structures in the image itself, ranging from simple structure such as points or edges to structures that are more complex such as objects.

Before explaining the goal of the current work, "features" in this context are categorized into five scopes. Features mentioned in the current work will correspond to these scopes rigorously.

- **2D signal-based features**: Pixel-wise information saved in the same size as the input image, which is computed by a feature detector or algorithm.
- **2D symbol-based features**: Encapsulated and condensed feature descriptor computed from 2D signal-based features, independent of the image size. In the current work, 2D symbol-based features include "2D Primitive" s and "2D contour" s. A "2D Primitive" can be regarded as a "fundamental visual element".
- **2D model-based features**: Object knowledge analyzed and reasoned from 2D symbol-based features and 2D signal-based features, independent of the image size. In the current work, we only consider an "ellipse" s as 2D model, which is the projected object model in 2D from a "3D circle".
- **3D symbol-based features**: Analogous to "2D Primitive" s, 3D symbol-based features are called "3D Primitive" s, which are computed from 2D signal-based and symbol-based features. "3D contour" s represent the 2D contours in three dimensions.

- **3D model-based features**: Object models in 3D. In the current work, we only consider a "3D circle" s as a 3D model. However, the theory of the current work can be easily migrated with other models, such as, squares, triangles, etc.

Within the computer vision domain, the current work aims to improve feature extraction by making use of predictions from circle models and motions via a feedback mechanism. 3D Circle models are computed by a sophisticated approach in which 2D ellipses first need to be hypothesized. To ensure that detected circle models precisely overlap the object presented in input images, an ellipticity-check function is made to remove unqualified circle models. A feedback mechanism is designed and implemented which feeds back the 3D circle models to Gabor wavelet responses by a 3D Gaussian function. Moreover, updated signals trigger the visual processing chain, leading to re-extraction of features including circle models.

The current work employs the visual process hierarchy from the "MoInS" Software package [14]. The inputs of the visual processes of the current work are 2D input images and the outputs are qualified 3D circle models. More specifically, 3D circle models are hypothesized from 2D contour pairs. 2D contours are comprised of 2D groups. 2D groups are computed by finding aligned Primitives. The term "Primitive" was first introduced in [15], and can be regarded as a "fundamental visual element". Primitives are extracted from multiple 2D signal-based features produced by neighborhood operations.

The purpose of the feedback process is to validate the predictions, to retrieve features from the predicted circle models or motions, and to improve the weak features that have been extracted. After the model knowledge has been fed back to the low-level image processing level, the visual processes can be re-started. In order to test the proposed feedback mechanism, both artificial and real scenes were experimented with where the

motions and the qualified circle models helped to retrieve the missing features.

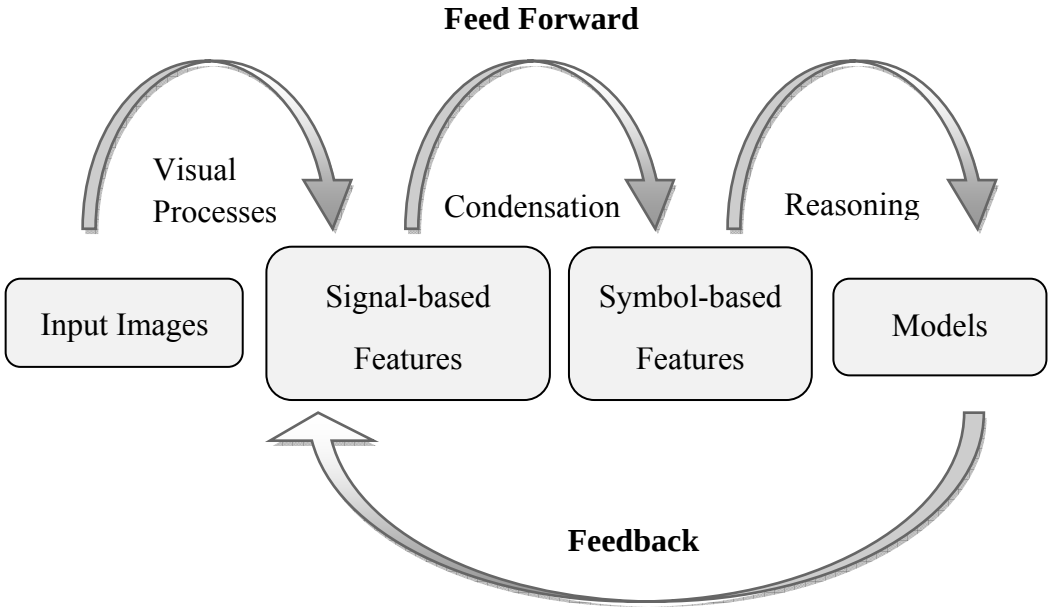


Fig. 2 The workflow diagram includes four states, indicated by blocks: input images, signal-based features, symbol-based features and models and four processes, indicated by arrows. The arrows above the blocks refer to a feed-forward process while the arrow beneath the blocks refers to a feedback process.

1.3 Problem Analysis

The goal of the current work is to design a feedback process that improves feature extraction in the computer vision and the image processing domains by using predictions. Many other applications attempt to use neighborhood operations in pixel domain to extract features or enhance feature extraction, such as convolution (filtering), differentiation, correlation, etc. The novel approach in the current work is to hypothesize predictions from model knowledge or motions and apply the predictions back to the low-level visual processes. The predictions are checked against signal-based features to ensure that the extracted models can be used to improve features. A feedback mechanism is designed to influence signal-based features with

model-based features using parameterized predictions. The difficulties encountered in the current work remain in designing feedback between model knowledge and pixel information because of the abstract status (from high-level processes) of model and the "continuous-like" (digitalized discrete grid from low-level processes) status of pixels.

In Fig. 2, the lifecycle of input images are presented and divided into the four states. Image inputs in the first stage typically are captured by cameras, where the raw data of intensity for each pixel is saved. Signal-based features are the outputs from visual processes where information is still conserved in a pixel manner in the next state. Then a condensation results in symbol-based features, information is compressed by dropping fine details in pixels in the third state. The refined symbol-based features from signal-based features become more meaningful and convenient for model-related processing. However, while the feed-forward processes provide refined features in an elegant way, the probability of having prejudices increases. To ensure the correctness of model knowledge, an approach that transfers data from the model-based or the symbol-based level to the signal-based level is proposed. Similar behavior also occurs in cognitive science, where selective attention assists object recognition [16] [17].

Fig. 3 illustrates the problem that needs to be solved. A 3D circle is projected as a 2D ellipse into the image sensor of a camera. The obtained images are often contaminated, by various issues such as shading problems, noises, reflections, etc. The 2D and 3D Primitives with dashed lines denotes the imperfect features that are not extractable. Our goal is to find the 3D circle model and retrieve those Primitives. To find the 3D circle model, we first calculate the 2D ellipse model to remove the redundant and erroneous candidates because 2D Primitives are more reliable than 3D Primitives are. Before the feedback mechanism is applied, an ellipticity-check function is used to ensure that the predicted 3D circle model has good correspondences to the signal-based features. The feedback process combines the predictions with the Gabor wavelet responses, which is

the output of the Gabor wavelet filtering. With this feedback process, the visual processes in Fig. 2 are re-started.

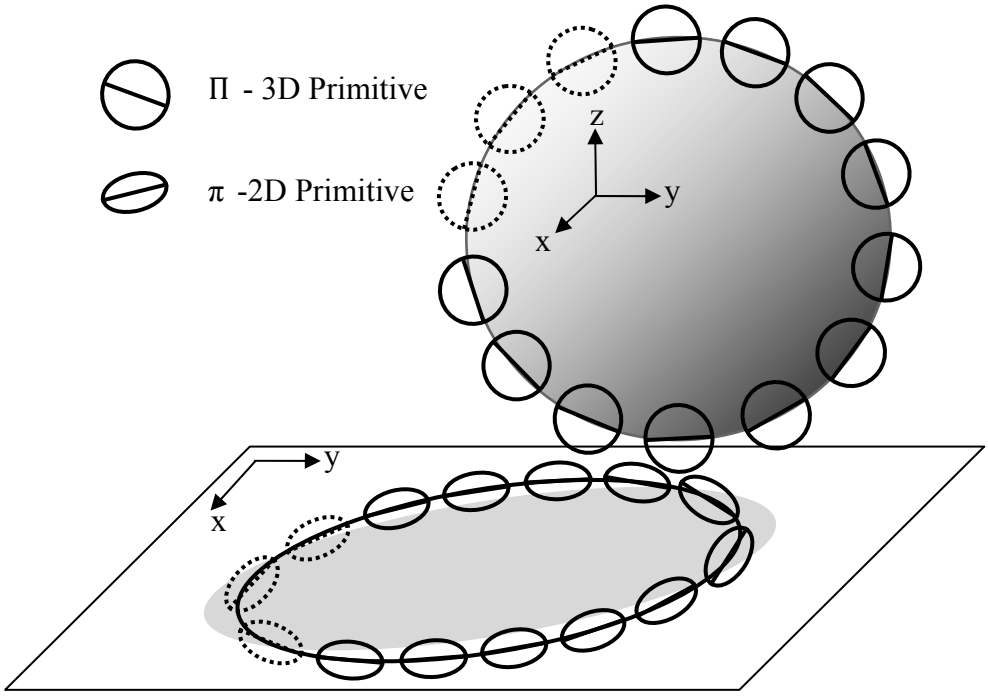


Fig. 3 The 3D Primitives and 2D Primitives attached to the 3D circle and 2D ellipse. The dashed lines indicate missing Primitives due to low contrast. In this example, the ellipse model is biased from the projected ellipse. Such problems are solved by the circularity-check function.

1.4 Delimitations

The focus of the current work is to demonstrate the proposed feedback mechanism with 3D circle models. Therefore, the input images are required to contain visible circles. However, the feedback process does not bind to circle models, and can be migrated with other models without difficulty. The current work utilizes the software framework from MoInS, so the implementation concentrates on extending the current structure as well as making it reusable and robust.

The thesis is planned to explain necessary relevant fields, so that people without

specialization within the domain can follow up on the content. However, fundamental knowledge of image processing, computer vision, neuron-science and mathematics is required. Due to the time constraints imposed upon this particular study, the author admits that certain tasks could have been refined further. However, the thesis presents an integrated workflow of the defining project, design solution, implementation, testing and reports.

1.5 Outline

Chapter 1 discusses the problem analysis and defines the scope of the current work.

Chapter 2 describes the necessary theoretical basis for extracting the fundamental visual elements. Primitives are used as the primary analysis tool in the thesis. The procedures for extracting Primitives are divided into different layers, named filtering layer, intrinsic dimensionality layer, 2D Primitive Layer and 3D Primitive Layer.

Chapter 3 describes the mechanisms and algorithms relating to circle detection. The circle detection makes use of 2D signal-based features. Primitives are used as the processing unit. To make the circle models reliable, the 2D ellipses projected from 3D circles are calculated and validated before the 3D circles are created. After eliminating unqualified models, circle models are passed to the feedback mechanism.

Chapter 4 explains the feedback mechanism. Having successful circle models in three dimensions, the task is to re-generate the lost features in the input images. An ellipticity-check function ensures that, only qualified circle models can feed the information back to the input images. The feedback is combined with the Gabor wavelet responses, which forms the bottom layer in the hierarchy. After the feedback, the features can be re-extracted to retrieve the lost ones.

Chapter 5 provides simulation results for different schemes. To show the reliability of the illustrated techniques, both artificial and real scenes are illustrated. To show the robustness of the techniques, both simple scene and complex scene are shown. To show the adaptability and extendibility of the techniques, image sequences are shown with known RBM.

Chapter 6 summarizes the overall work. Possible directions for future studies are also provided in this chapter.

Appendix A discusses the implementation from the viewpoint of software engineering. A class diagram and the relationships between different classes are outlined. This chapter can be used as a guideline for developing the project.

1.6 Contribution

The work has been previously studied by the author where the signal-symbol loop and feedback was initialized [18][19]. The paper has led to a conference paper [20] and a journal paper, which is being written.

The author's focus is to develop and extend the feedback mechanism. Either hypothesized circle models, RBM, or the combination of both can trigger the feedback process. The contribution from the author in the current work includes design, implementation and testing of the feedback mechanism, which is explained in details in Section 3.5, Chapter 4, Chapter 5 and Appendix A. The technologies and algorithms described in [6] [21][22][23][24] are utilized in the current work and considered to be important. Hence, they are explained briefly in Chapter 2 and Chapter 3 to keep the paper persistent and readable.

2 Hierarchical Visual Process

2.1 Introduction

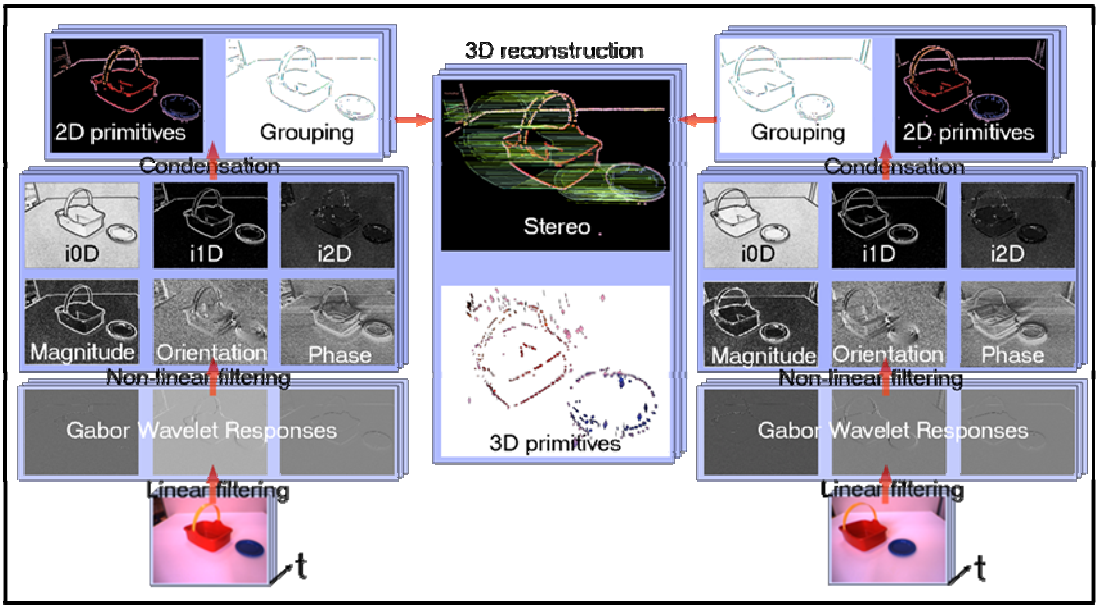


Fig. 4 Overview diagram presenting the visual process hierarchy modified based on [19]. To enhance the appearance, only three Gabor filter responses are shown.

The 3D circle models presented in the study are computed via different visual processes, which form a hierarchy of visual processes. Fig. 4 demonstrates these workflows. The input includes the left and right image with their corresponding projection matrices. The computation flow starts from the input images layer at the bottom of the hierarchy, feeding the pixel-wised information forward to the filtering layer, then further to the intrinsic Dimensionality layer. Afterwards, outputs from aforementioned layers are encapsulated to build the fundamental visual elements, so-called "Primitive" s. Primitives consist of a group of vectors representing multi-modal information within a local neighborhood. Then stereo matching is used to find correspondences of 3D data based on comparing 2D Primitives from different cameras. 3D Primitives are then

created from 2D Primitive pairs (one from left image and one from right image) which have the same 3D location. Hence, the hierarchy is a bottom-up approach driven by input images. As the process progresses, it becomes increasingly less detailed and more semantic. However, the information being "dropped" during the processes may contain conflicting votes against the output produced from the layers. This can cause prejudices and wrong decisions. This chapter depicts the workflow hierarchy in details. The later chapters will explain how this issue was addressed.

2.2 Filtering layer

In the filtering layer, input images are convoluted by the kernels, also known as masks. The kernels are normally designed to depict a pixel in respect of detecting a specific feature, such as an edge-structure, junction structure, etc. The output, or filter response assigns scores to a point that denotes the strength of each pixel for the required feature. Wavelets can be considered as a series of complex kernels containing real and imaginary kernels. Gabor wavelets are a series of Gaussian-modulated sinusoids, widely used in the bio-neural field because there is considerable evidence that images in the primary visual cortex (V1) are represented in terms of Gabor wavelets [25][26]. The Gabor wavelet response can be used to describe the gradient information of a specific frequency and orientation.[27] showed that the real part of complex Gabor functions map very well to the receptive field weighting function found in simple cells in a cat's striate cortex. In the current work, eight orientations are defined with one frequency level for general-purpose usage (other applications may use higher frequencies to detect texture or lower frequencies for image compression). The Gabor wavelets are illustrated in Fig. 5 with sample outputs.

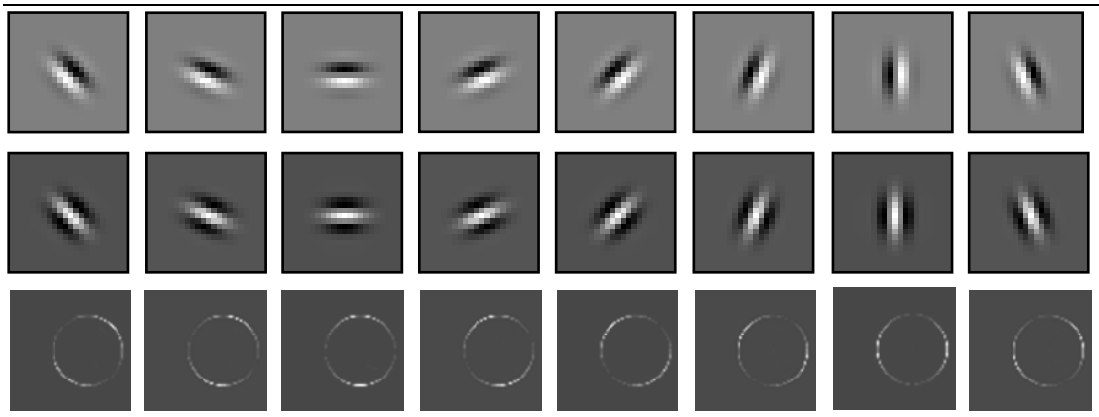


Fig. 5 The Gabor wavelet kernels and the filter responses of a circle image. The top two rows denote the real and imaginary kernels (masks) in eight directions respectively. The bottom row denotes the magnitude of the resulting filtered images.

In situations where there are Gabor wavelet kernels with one frequency level and eight directions as shown in Fig. 5, the original images are computed with standard convolution algorithm. As the Gabor wavelet kernels exist in complex domain, the resulting output can be denoted as $Re_c(x,y)$ and $Im_c(x,y)$ indicating the real and imaginary output with channel c (direction) and position(x,y). The convolution can be formulated as:

Eq. 1

$$Re_c(x,y) = g_c^{re} * image = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} g_c^{re}(m,n)image(x-m,y-n)$$

Eq. 2

$$Im_c(x,y) = g_c^{im} * image = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} g_c^{im}(m,n)image(x-m,y-n)$$

Where g_c^{re} and g_c^{im} are real and imaginary part of Gabor wavelet kernels with direction c, and $image$ is the input image in gray scale.

The Gabor wavelet responses are used to estimate magnitude, orientation and phase[28]. These results depict the overall estimation disregarding the specific

orientation from the Gabor responses:

For each channel, we have:

Eq. 3

$$m_c(x,y) = \sqrt{Re_c^2(x,y) + Im_c^2(x,y)}$$

Eq. 4

$$\vartheta_c(x,y) = \frac{c\pi}{K}$$

Eq. 5

$$\varphi_c(x,y) = arctan\left(\frac{Re_c(x,y)}{Im_c(x,y)}\right)$$

Where $m_c(x,y), \vartheta_c(x,y), \varphi_c(x,y)$ are the respective magnitude, orientation and phase for a given position (x,y) and direction c. For estimated magnitude $m(x,y)$, orientation $\theta(x,y)$ and phase $\phi(x,y)$ without specifying direction c, we have,

Eq. 6

$$m(x,y) = \sqrt{\widehat{Re}^2(x,y) + \widehat{Im}^2(x,y)}$$

Eq. 7

$$\theta(x,y) = arg\left[\frac{4}{3}\sum_{c=0}^{7} m_c(x,y) \cdot e^{2j\vartheta_c(x,y)}\right]$$

Eq. 8

$$\phi(x,y) = \frac{1}{2} arctan\left(\frac{\widehat{Re}(x,y)}{\widehat{Im}(x,y)}\right)$$

Where $\widehat{Re}(x)$ and $\widehat{Im}(x)$ are defined as a summation of channels,

Eq. 9

$$\widehat{Re}(x,y) = \sum_{c=0}^{7} Re_c(x,y)$$

Eq. 10

$$\widehat{Im}(x,y) = \sum_{c=0}^{7} Im_c(x,y)$$

The above processes are illustrated in the linear filtering block in Fig. 4.

2.3 Intrinsic Dimensionality Layer

The intrinsic dimensionality layer, which is taken from [23][21], is an extended filtering layer. The inputs for processing this layer are the estimated magnitude and orientation. The resulting outputs are three images that can be used as a descriptor for different structure types, intrinsic zero dimensionality (i0D), intrinsic one dimensionality (i1D) and intrinsic two dimensionality (i2D). i0D measures the likelihood of being classified into a homogeneous or textured region where no dominant structure is considered. i1D measures the likelihood of being classified into an edge-like structure of which the direction can be depicted by the largest eigenvalue. i2D measures the likelihood of being classified into a junction or complex structure, where two eigenvalues are needed to represent the structure. The intrinsic dimensionality is illustrated in the top of non-linear filtering block in Fig. 4.

The intrinsic dimensionalities are computed by observing line variance and origin variance and categorizing the result into three types. The origin variance depicts the likelihood of having any variance within a neighborhood. Therefore, origin variance is defined by normalized magnitude, because magnitude can be regarded as output of a gradient-based edge filter.

Eq. 11

$$\hat{\sigma}_R(x,y) = N\big(m(x,y)\big)$$

The line variance depicts how well the orientation information is persistently distributed within a neighborhood. The line variance can be computed by:

Eq. 12

$$\hat{\sigma}_L(x,y) = \frac{\sum_{x\prime=x-R}^{x+R}\sum_{y\prime=y-R}^{y+R} e^{2j\cdot\theta(x\prime,y\prime)\cdot N\left(m(x\prime,y\prime)\right)}}{\sum_{x\prime=x-R}^{x+R}\sum_{y\prime=y-R}^{y+R} \arg\left\{e^{2j\cdot\theta(x\prime,y\prime)\cdot N\left(m(x\prime,y\prime)\right)}\right\}}$$

To refine the line variance and origin variance so they produce evenly distributed

output, another normalization is applied,

Eq. 13

$$\sigma_R(x,y) = N_R\{\hat{\sigma}_R(x,y)\}$$

Eq. 14

$$\sigma_L(x,y) = N_L\{\hat{\sigma}_L(x,y)\}$$

Having defined the origin variance and line variance, intrinsic dimensionalities can be computed within a barycentric coordinate, meaning that for any given pixel (x,y), we have,

Eq. 15

$$i0D(x,y) = 1 - \sigma_R(x,y)$$

Eq. 16

$$i1D(x,y) = \sigma_R(x,y) - \sigma_L(x,y)$$

Eq. 17

$$i2D(x,y) = \sigma_L(x,y)$$

Eq. 18

$$i0D(x,y) + i1D(x,y) + i2D(x,y) = 1$$

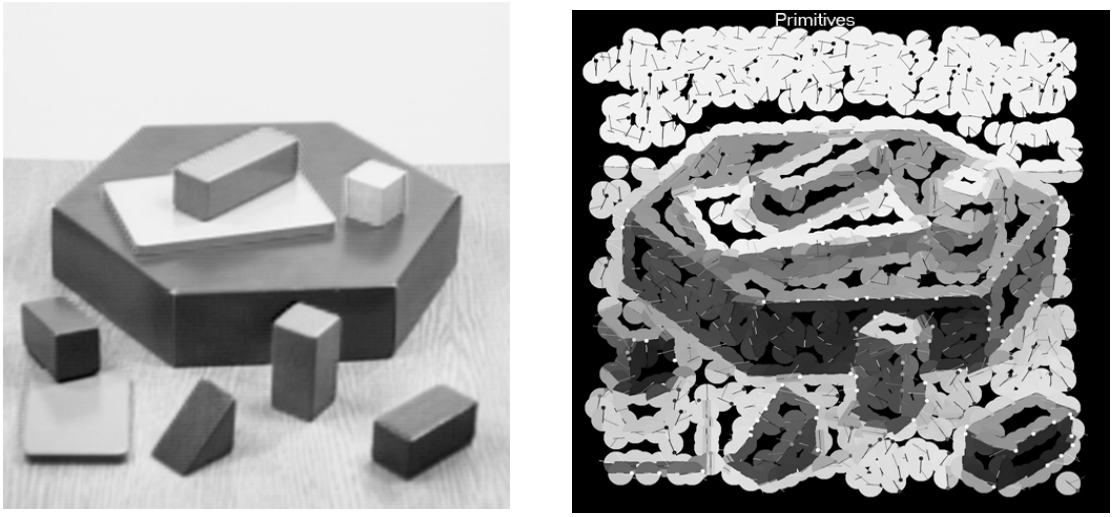
2.4 2D Primitive Layer

With the fore mentioned outputs from the filtering layer and the intrinsic dimensionality layer, the information is encapsulated in the 2D Primitive layer. Such encapsulation includes condensation and compression. Condensation simulates the human biological visual system. Condensation gains computational efficiency and, assembles the pixel or signal-based information into an encapsulated package, known as "Primitive" [15][24]. The term "Primitive" refers to a group of vectors representing multi-dimensional information within a local neighborhood area. In this thesis, the term, "Primitive" refers to descriptor "π". With position $\{x,y\}$, magnitude m, orientation θ, phase ϕ, intrinsic dimensionality iD, color c, Primitive can be defined as:

Eq. 19

$$\pi = (\{x, y\}, m, \theta, \phi, iD, c)^T$$

Thresholds can be set depending on the purpose of the applications, allowing for the creation of different types of Primitives. Typically, edge and junction structures are the regions of interests. In Fig. 6, Primitives are used to present a scene with boxes.



a) Input image b) Extracted Primitives from a)

Fig. 6 Symbolic representation of an image with boxes.

2.5 3D Primitive Layer

By using the approach in [15], 3D Primitives can be constructed by triangulation. Triangulation is 3D reconstruction using projection geometry with pairs of 2D Primitives. Each pair of 2D Primitives contains one Primitive from the left image and one from the right. Having the epipolar geometry and constraint, the work of finding correspondences is reduced by only searching the correspondences on the epipolar line in the right image for a specific Primitive in the left image. A 3D Primitive with descriptor Π can be defined as:

Eq. 20

$$\Pi = (\{x, y, z\}, \Theta, \Phi, C)^T$$

Where $\{x, y, z\}$ is the position in 3D, Θ is the 3D orientation, Φ is the 3D phase and C is the 3D color information.

3 Circle Model

3.1 Introduction

In chapter 2, the fundamental hierarchical visual processes are discussed. This workflow can be used similarly in other computer vision applications. In this chapter, the fundamental hierarchical visual processes are extended to an optional layer, in order to extract circle models. The optional layer is the specific task in the current work and can be substituted for other models [22].

3.2 3D Contours

The 3D grouped contours are linked with 2D grouped contours fulfilling two conditions [24]:

- The contour needs to have a minimum length implying a possible contribution for a circle model.
- If two contours are positioned within a minimum distance by measuring the starting point and end point, they are concatenated.

With the pre-processes mentioned above, the contributors are further grouped and able to predict better circle models.

3.3 Crude 2D Ellipse Models

The crude circle models are computed with 3D contours. Due to the uncertainties in the 3D data that forms 3D contours, various combinations of 3D contours are generated in the method described in Section 3.2, and part of the 3D contours may be incorrect. A 3D circle fitting for the full set of 3D contours may lead to too many candidates that are not computationally efficient. To limit the number of 3D circle candidates, a validation

process of 3D contours was designed. The 3D contours are first projected into 2D contours. The process should be noted that 3D circles become 2D ellipses when they are projected onto two dimensions. The difference between circles and ellipses is that circles have a radius measured by the circle center while ellipses have two foci points in addition to a "radius". Using the direct least square fitting of the ellipse [29], ellipse hypotheses are created from 2D projected contours. A center-positioned ellipse can be defined as a locus of points in a plane where the sum of the distances of two fixed points (foci) to the point is a constant (see Fig. 7).

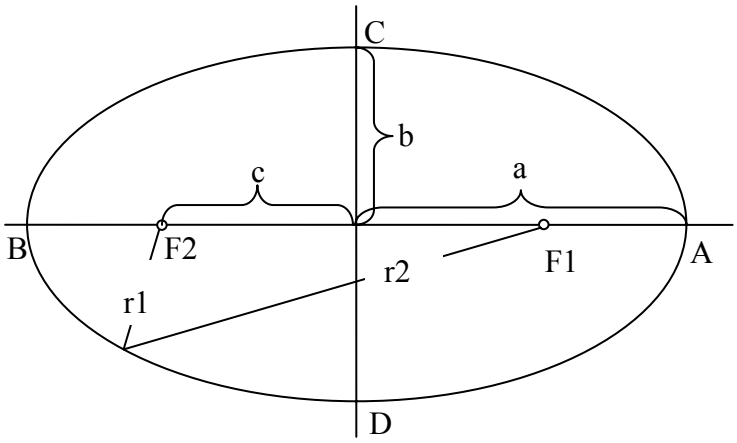


Fig. 7 A center-positioned ellipse.

The ellipse illustrated in Fig. 7 has such properties:

Eq. 21

$$r1 + r1 = 2a$$

The line segment between AB is called the major axis. The line segment of CD is called minor axis. To eliminate the flawed 2D projected contours, several methods are applied:

- Eccentricity is a parameter presenting the relationship between length of the major axis a and the minor axis b, expressed with e:

Eq. 22

$$e = \sqrt{1 - \frac{b^2}{a^2}}$$

e is required to be less than 0.942 in the study, so the ratio of b to a is greater than $\frac{1}{3}$.

- Coverage rate is used to measure the percentage of the 2D points projected from 3D Primitives (in the 3D contour) that map to the 2D ellipse. The threshold is set depending on the specific task requirement (50% is set to the minimum coverage rate in the current work).

After the eliminations, the ellipse models remain in 2D. However the 3D circle models are still not obtained. In the coming section, we discuss how to compute the 3D circle models.

3.4 Optimized 3D Circle Models

The 2D ellipse models from the previous section provide an implicit measurement for 3D circle models. The eliminated 2D ellipse models are then used to find the 3D circles [30]. A 3D circle model can be described as:

Eq. 23

$$C = (X, r, \vec{n})$$

where $(X, r, \vec{n})$ denotes the 3D circle center, radius and the 3D surface the circle is placed at. A 3D circle can be validated by fulfilling the following requirements:

- The computed quadratic curve fulfils the requirement of being an ellipse
- The 3D points which compute the 3D circle model are not biased far from the model.
- The radius data set has a low variance.

To reduce the duplicated circle with subtle differences, a second-round check is implemented to check the re-projected 2D ellipses in order to select the true positive 3D models and suppress the flaw ones.

The 3D circle detector can be regarded as an extended layer of the bottom up

approach described in Chapter 2. Such a "pyramid-structure" approach gives the top layer more semantic and compressed information while the uncertainties and unreliability increases. To address this issue, we introduce a check function in the next section that covers the signal level.

3.5 Ellipticity-Check Function

Based on the parametric forms of 2D ellipse models, a voting scheme is designed to compute the confidence of an ellipse model. A tolerance of projected area is allowed so that a small variance of the ellipse's location will also provide a satisfactory result (tolerance is set to two pixels from the projected points in the current work). Each pixel has its own weight according to its intrinsic dimensionality confidence. Note that only regions with high intrinsic one or two dimensionality are useful candidates for voting which indicates a high probability of non-homogenous areas. Another advantage of using intrinsic dimensionality is that the output is normalized. The result of circle confidence is in the range of zero and one. In an ideal condition, an ellipse has a confidence rating of 1.0 from the ellipticity-check function only if the orientation values of all pixels in the projected area are exactly equal to the expected orientation values. An ellipse has a confidence value of 0.0 from the ellipticity-check function only if the orientation values of all pixels in the projected area are exactly perpendicular to the expected orientation values.

The ellipticity-check function computes w_E, for pixels p(x,y) located within the projected neighborhood of an ellipse model C and can be expressed as:

Eq. 24

$$w_E = \frac{1}{N \cdot \frac{\pi}{2}} \sum_{p_{(x,y)}}^{d(p,E)<t} (\mathrm{i1D} + \mathrm{i2D}) \cdot \Delta\theta$$

Eq. 25

$$\Delta\theta = \begin{cases} |\tilde{\theta}_p - \theta_p|, & |\tilde{\theta}_p - \theta_p| < \frac{\pi}{2} \\ \pi - |\tilde{\theta}_p - \theta_p|, & |\tilde{\theta}_p - \theta_p| \geq \frac{\pi}{2} \end{cases}$$

where $\tilde{\theta}_p$ is the expected orientation value of pixel p, θ_p is the estimated orientation value of pixel p from Gabor wavelet responses, N is the amount of pixels fulfilling the condition, $d(p,E) < t$ is the condition that the distance between p(x,y) and ellipse E should be less than the threshold distance t (t is set to 2.0 in the current work). (i1D+i2D) is the weight of the difference between expected and calculated orientation values. By using the above ellipticity-check function, an ellipse model value is given by a weight w_E between 0.0 and 1.0. A manual threshold can be set to eliminate the low-confidence circle hypotheses (0.8 is chosen in this study).

More specifically, $d(p,E)$ is computed by measuring the distance between point p and point q which is the closing intersection point between the circle C and line l made by point p and center of the circle model C, point o. $\tilde{\theta}_p$ is set to the orientation value of pixel q, $\tilde{\theta}_q$ (see Fig. 8).

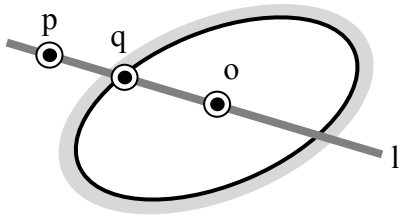
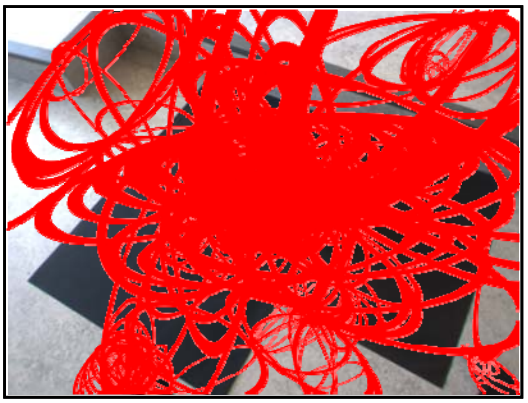


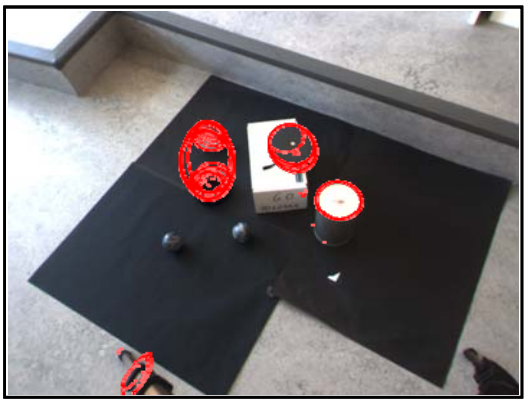
Fig. 8: Point o is in the center of the ellipse, p is a given point, and l is a line crossing o and p. q is the closing intersection point to p between l and the ellipse. p, q and o are located on line l.

Fig. 9 illustrates the elimination process described in this chapter. In (a), all the

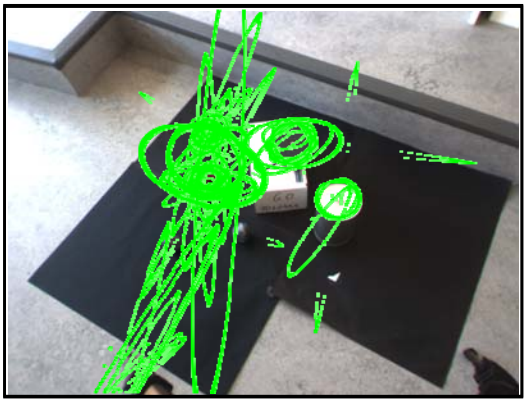
projected ellipses from hypothesized circles are shown. The results after eliminating the circles that do not meet the ellipse requirements are shown in (b). Further validation is made in 3D to remove the duplicated or erroneous circles and the results are shown in (c). By using the ellipticity-check function against signal-based features, the results become optimal as shown in (d). More results are illustrated in Appendix B, see Fig. 26 and Fig. 27.



a) Extracted circles from contours



b) After elimination by ellipses



c) Further elimination in 3D

d) Final yield circles by using ellipticity-check function

Fig. 9 The elimination processes of hypothesized circles models.

4 Feedback Model

4.1 Introduction

The previous chapter proposed a circle model detection presented in [22]. By using this algorithm, 3D circle models are hypothesized. The predicted 3D circle models may be erroneous due to threshold problems or redundant information from early processes. To remedy this problem, the ellipticity-check function is designed to compute the similarity between the 2D ellipse model and 2D input images in order to remove the 3D circle models with low confidence. The ellipticity-check function also ensures that only high quality circles are allowed to feed back to the circle models.

The visual process hierarchy's aforementioned feed-forward information forms primary sensory areas to the higher-level regions. Each layer within the hierarchy processes and encapsulates information depending on the role of the layer. The higher the layer is located, the more abstract the output will be. This chapter formalizes the feedback mechanism and further specifies the feedback configuration from 3D symbol-based features and 3D model-based features. The features that drive the feedback can be regarded as "dominant activities". The feedback mechanism amplifies the contrast and saliency of the local stimulus on pixel level by a Gaussian function.

Firstly, we discuss the feedback mechanism, which can be used by either motion information or 3D model-based features. Then we discuss the generalization of the feedback mechanism from a probability analysis point of view.

4.2 Feedback Mechanism

In the current work, we proposed a positive feedback mechanism meaning adding

positive values to the input to enhance the feature extraction. Gabor responses are used as the target of the feedback mechanism. We use the Gabor wavelet kernels with one frequency and eight directions. Similarly, the feedback is saved in an array of complex images. The updated Gabor response $\hat{G}$ can be defined with the original Gabor response G and 2D feedback F with normalizing parameter $G_{c_i}^{max}$ and w_τ:

Eq. 26

$$\hat{G}^{\mathrm{Re}}(x,y,c_i) = \mathrm{G}^{\mathrm{Re}}(x,y,c_i) + G_{c_i}^{max} \cdot F(x,y,c_i) \cdot \mathrm{w}_\tau$$

Eq. 27

$$\hat{G}^{\mathrm{Im}}(x,y,c_i) = \mathrm{G}^{\mathrm{Im}}(x,y,c_i) + G_{c_i}^{max} \cdot F(x,y,c_i) \cdot \mathrm{w}_\tau$$

Where (x,y) indicates the pixel location, c_i refers to the specific channel. w_τ is the dynamic weighting parameter to minimize the feedback from a homogenous-like or texture-like structure. w_τ can be determined depending on the specific feedback tasks. w_τ can be considered as a similarity measurement between the prediction and the 2D signal-based features. Usage of w_τ is discussed in detail in the subsections. As the Gabor response $G(x,y,c_i)$ is not normalized, and $F(x,y,c_i)$ is between 0 and 1, $G_{c_i}^{max}$ is employed to regulate feedback so the feedback is scaled according to the maximum of the specific Gabor response channel. 2D feedbacks are defined as being analogous to Gabor response channels, where the number of the channels corresponds to the number of orientations declared in the Gabor wavelet (eight channels are used in the current work). As each Gabor response channel comprises of a complex 2D map representing a complex energy map in a specific direction (orientation), it does not necessarily need a complex channel for feedback. Consider using a polar form to present the complex signal. The absolute value (argument) is sufficient to conserve the 2D geographical energy-map for well-predicted circle models. At the signal level, feedback response is defined by an un-normalized three-dimensional Gaussian function.

4.2.1 Feedback using Motion Information

In prior studies as described in [18], the feedback mechanism is applied in the scenario where the RBM between image sequences is known. The RBM provides positional implications of 3D Primitives in other frames. Such a scenario gives opportunities of applying the feedback mechanism to Primitives, which are the symbol-based features of the current work. The 3D Primitives computed in the current frame are transformed by RBM, predicting 3D Primitives at a later frame. A 3D Primitive Π with index i in time t can be transformed into a predicted 3D Primitive $\widehat{\Pi}$ in time $t + \Delta t$, as:

Eq. 28

$$\widehat{\Pi}_i^{t+\Delta t} = RBM_{t \to t+\Delta t}(\Pi_i^t)$$

Each predicted 3D Primitive is further projected into a predicted 2D Primitive, as the feedback process is conducted in two dimensions:

Eq. 29

$$\hat{\pi}_i^{t+\Delta t} = Proj(\widehat{\Pi}_i^{t+\Delta t})$$

In order to remove the erroneously predicted Primitives, we customize w_τ with $w_{\hat{\pi}}$ by revising the similarity function to verify the predicted 2D Primitive $\hat{\pi}$ with the signal-based features in time $t + \Delta t$. The resulting value $w_{\hat{\pi}}$ is also used as a threshold to eliminate the 3D Primitives. The threshold is set to 0.7 in the current work. $w_{\hat{\pi}}$ can be computed by using signal-based features for pixel p within neighborhood of Primitive $\mathcal{N}(\hat{\pi})$.

Eq. 30

$$w_{\hat{\pi}} = \frac{1}{N \cdot \frac{\pi}{2}} \sum_{p_{(x,y)}}^{p \in \mathcal{N}(\hat{\pi})} \left(\mathrm{i1D}_p + \mathrm{i2D}_p\right) \cdot \Delta\theta_p$$

Eq. 31

$$\Delta\theta_p = \begin{cases} |\theta_{\hat{\pi}} - \theta_p|, & |\theta_{\hat{\pi}} - \theta_p| < \frac{\pi}{2} \\ \pi - |\theta_{\hat{\pi}} - \theta_p|, & |\theta_{\hat{\pi}} - \theta_p| \geq \frac{\pi}{2} \end{cases}$$

Where $\mathrm{i1D}_p$, $\mathrm{i2D}_p$ and θ_p represent intrinsic one dimensionality, intrinsic two

dimensionality and orientation on pixel p respectively.

For qualified Primitives, feedback channels $F(x, y, c_i)$ of pixel (x, y) and channel c_i can be conserved as:

Eq. 32

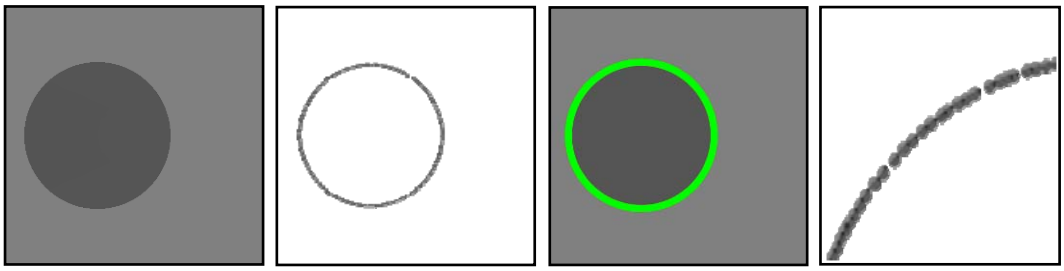
$$F(x, y, c_i) = \exp\left(-\frac{1}{2}\left\{\frac{[(x - x_{\hat{\pi}})\cos\theta_{\hat{\pi}} + (y - y_{\hat{\pi}})\sin\theta_{\hat{\pi}}]^2}{\sigma_x} + \frac{[-(x - x_{\hat{\pi}})\sin\theta_{\hat{\pi}} + (y - y_{\hat{\pi}})\cos\theta_{\hat{\pi}}]^2}{\sigma_y} + \frac{[c_i - c_{\hat{\pi}}]^2}{\sigma_c}\right\}\right)$$

Where $x_{\hat{\pi}}, y_{\hat{\pi}}$ and $\theta_{\hat{\pi}}$ denote the x-axis, y-axis (location) and the orientation of Primitive $\hat{\pi}$ respectively. The variances of the Gaussian function are set empirically as, $\sigma_x = 4, \sigma_y = 1, \sigma_c = 1$.

4.2.2 Feedback by Model-Based Features

The model-based features are exemplified with 3D circles in this section. The feedback mechanism can be configured for well-predicted circle models that have strong circularities according to the ellipticity-check function defined in Section 3.5. The target of the feedback is conserved on the Gabor response level, which is resulted from filtering a gray image with Gabor kernels. Gabor responses are comprised of a complex 2D image array. Therefore, the feedback mechanism uses circle models to influence and enhance the original Gabor responses. Using the proposed visual process hierarchy, circle models indirectly process on the Gabor responses, via Primitives. This also makes the approach of using the hierarchical processes more complete. As illustrated in Fig. 10, a circle model is spatially transformed into a vector of Primitives. In the current work, Primitives behave as a mediator between signal-based features and model-based features. More specifically, model-based features and circles are computed by Primitives without interacting with signal-based features. However, the feedback

process also employs Primitives as entities to propagate the model knowledge.



a) Input image b) Primitive c) Extracted ellipse d) Enlarged area

Fig. 10 The projected ellipse is computed by the Primitives that are extracted from the input image. Feedback is also imposed by using Primitives.

Analogous to Section 4.2.1, we use ellipticity w_E to replace $w_{\hat{\pi}}$ defined in Section 3.5 to remove erroneously predicted 3D circle models. For the qualified 3D circle models, the feedback manipulates the area close to the projected ellipse. The 2D Primitives with index i found in the neighborhood of the computed ellipse E can be defined as $\hat{\pi}_i^E$. For each Primitive found in the neighborhood of the extracted circle, a feedback process is made in the territory of the Primitive.

Eq. 33

$$F(x,y,c_i)_{\hat{\pi}_i^E} = \exp\left(-\frac{1}{2}\left\{\frac{[d(p,E)]^2}{\sigma_d} + \frac{[\Delta\theta]^2}{\sigma_\theta} + \frac{[c_i - c_E]^2}{\sigma_c}\right\}\right),$$

Where the variances of Gaussian function are set empirically as, $\sigma_d = 1, \sigma_\theta = 1, \sigma_c = 1$. $d(p,E)$ is the distance between the point $p(x,y)$ and the ellipse model E, $\Delta\theta$ is the angle difference between the orientation of the image and the ideal orientation of the found circle model on point $p(x,y)$. (See Eq. 25). $c_i - c_E$ is the channel distance where c_E is rounded from the falling orientation of pixel (x,y) into channel distribution (1~8). $d(p,E), \Delta\theta$ and $c_i - c_0$ are considered as three main aspects that determine the strength of the feedback on a specific pixel and channel. Eq. 33 is the core implementation of the signal symbol loops. Eq. 33 also joins the symbolic level and signal level together. After a circle model with high circularity confidence feeds back the circle model to the Gabor wavelet response level, the features of circles with

weak confidence will be enhanced.

4.3 Feedback Model Generalization

In previous sections, we discussed the feedback mechanism from the symbol-based or model-based features to the signal-based features. In this section, the feedback mechanism is generalized to a more abstract method. Defining model M and feedback F with weighting coefficients w_M and w_F, the updated model $\widehat{M}$ can be defined as:

Eq. 34

$$\widehat{M} = w_M \cdot M + w_F \cdot F$$

To have a bipolar feedback system meaning that outputs can be either increased or decreased, the weightings are normalized as:

Eq. 35

$$w_M + w_F = 1$$

Ideal weighting is reciprocal of its variance, meaning the less variance the dataset is, the higher its weight,

Eq. 36

$$weighting = \frac{1}{variance^2}$$

Assuming that the observations of model and feedback are both unbiased Gaussian probability density functions, the weightings of model and feedback can be defined as the normalized reciprocal variances respectively:

Eq. 37

$$w_M = \frac{\frac{1}{\sigma_M^2}}{\frac{1}{\sigma_M^2} + \frac{1}{\sigma_F^2}} = \frac{\sigma_F^2}{\sigma_M^2 + \sigma_F^2}$$

Eq. 38

$$w_F = \frac{\frac{1}{\sigma_F^2}}{\frac{1}{\sigma_M^2} + \frac{1}{\sigma_F^2}} = \frac{\sigma_M^2}{\sigma_M^2 + \sigma_F^2}$$

Based on the property of variance in the probability density function, variance of a weighted sum of variables can be computed as the summation of the sub-variances multiplied by their weightings. The variance of updated model can be computed as:

Eq. 39

$$\sigma_{\hat{M}}^2 = w_M^2 \cdot \sigma_M^2 + w_M^2 \cdot \sigma_F^2 = \frac{\sigma_M^2 \cdot \sigma_F^2}{\sigma_M^2 + \sigma_F^2}$$

Without difficulty, it can be proved that $\sigma_{\hat{M}} < \sigma_M$ and $\sigma_{\hat{M}} < \sigma_F$. The proof shows that the proposed feedback mechanism produces an updated model with less variance. One needs to ensure that w_M and w_F in Eq. 37 and Eq. 38 are correctly computed and represent the normalized weightings for model M and feedback F. Eq. 26 and Eq. 27 are customized based on Eq. 34, as our specific model remains in the computer vision domain where pixel-wised information tends to be strongly weighted. Moreover, feedback is hypothesized from the model. Therefore, only weightings (suppressions) are subject to feedback. The saturated magnitude is trimmed to ensure that the original extracted features are not suppressed in the normalization procedure.

5 Experimental results

In this chapter, a summary of the study's experimental results are presented and evaluated. Feedback from RBM, Models and the combination of both are shown in the following sections. For each section, results from both artificial and real scenes are illustrated.

5.1 Feedback from RBM

5.1.1 Artificial Scene

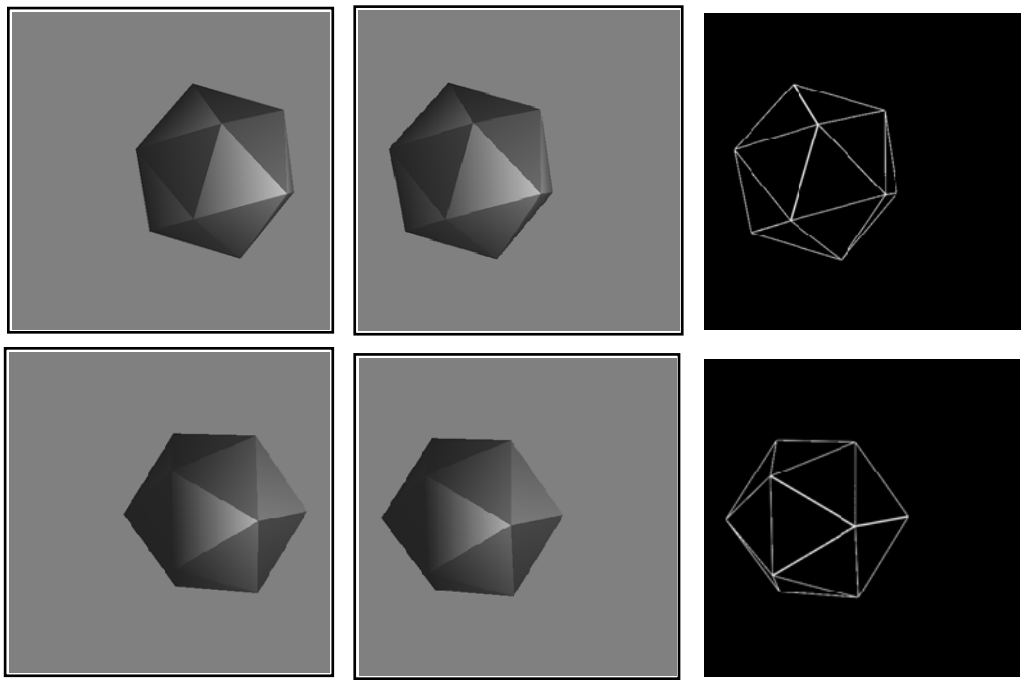
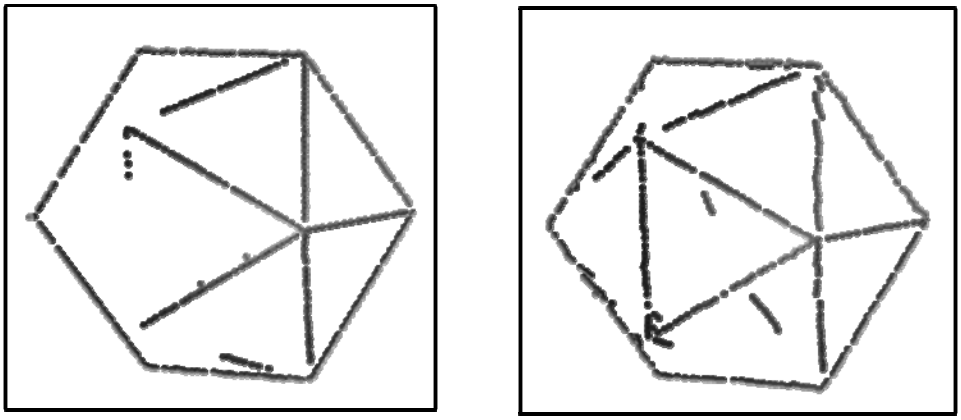


Fig. 11 From left to right shows the left and right artificial input images and the wireframe of right image respectively. The first row and second row indicate the first frame and the second frame correspondingly.



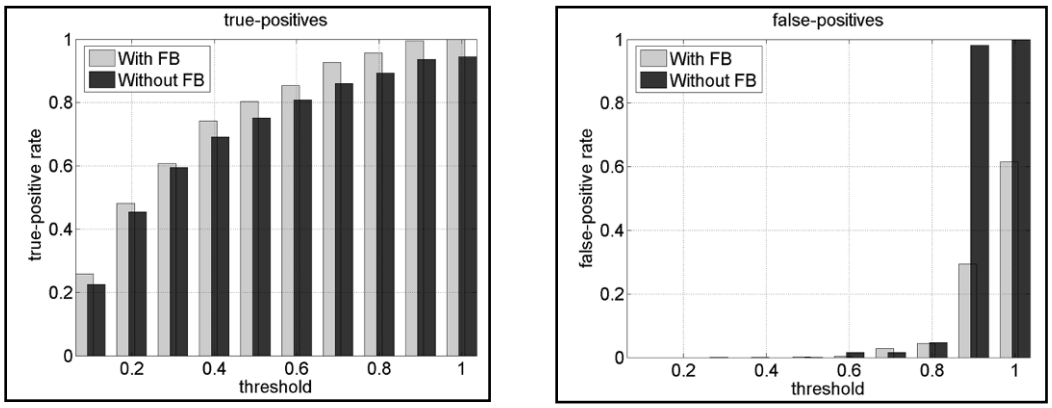
a) Primitives without feedback b) Primitives with feedback

c) Positives without feedback d) Positives with feedback

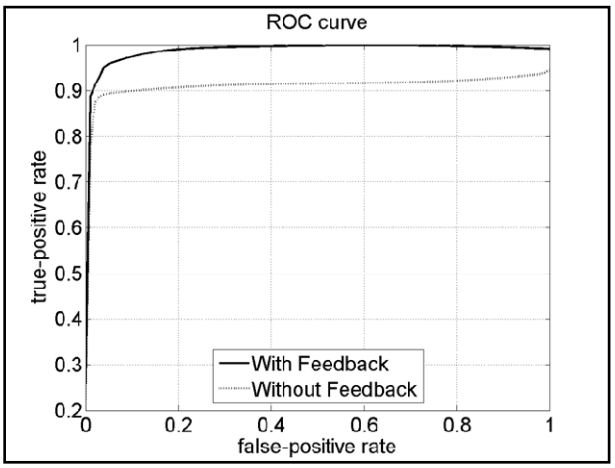
Fig. 12 Extracted Primitives and true-/false- positives without and with feedback based on motions.

An image sequence of icosahedrons (i.e., a polyhedron having 20 faces) is used for testing feedback from 3D symbol-based features. OpenGL generates the artificial image sequence, hence the RBM for the sequence can be obtained. The resulting wireframe data provides ground truth, which can be used to evaluate the performance of the proposed feedback mechanism. In Fig. 11, the input images and wireframes are shown. Fig. 12 (a) and (b) compares the extracted Primitives without and with feedback having the same threshold. Fig. 12 (c) and (d) mark the true-positives and false-positives for (a) and (b). True-positives and false-positives are defined by the distance between the extracted Primitive and the closest wireframe. We use the Receiver Operating Characteristics (ROC) curve to evaluate the performance of

proposed feedback process by quantifying threshold with 0.1 intervals. The ROC curve and its properties are shown in Fig. 13. From the results, we can see that the feedback mechanism improves the true-positives in addition to reducing the false-positives.



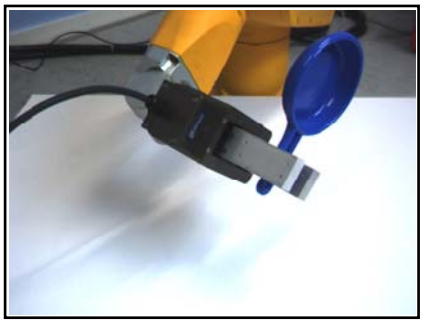
a) True-positives analysis b) False-positives analysis



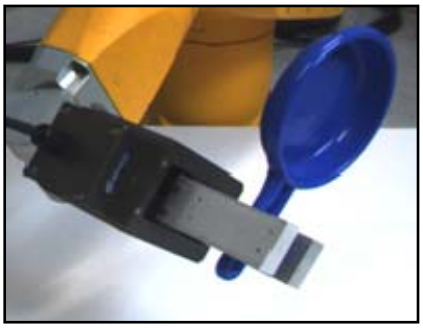
c) ROC curve

Fig. 13 ROC curve analysis in (c) which is made by the true-positives and false-positives shown in (a) and (b).

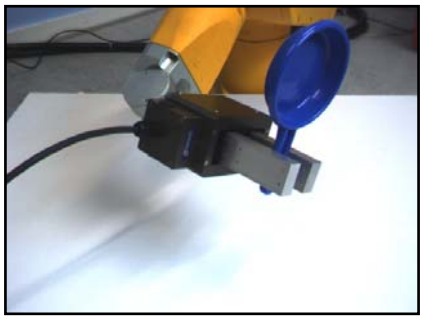
5.1.2 Real Scene



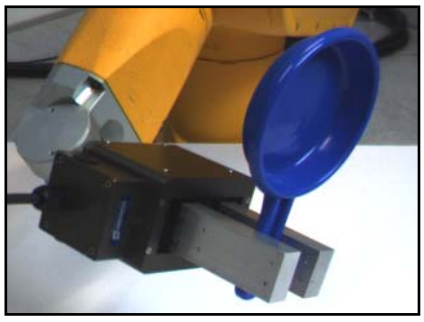
a) Left image in first frame



b) Enlarged a) to show details



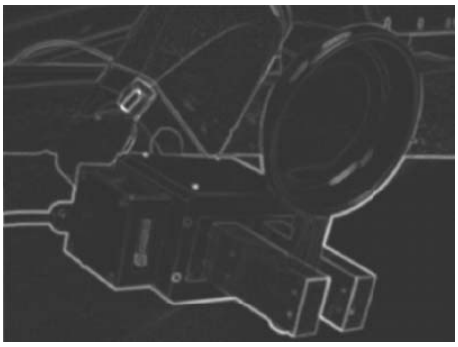
c) Left image in second frame



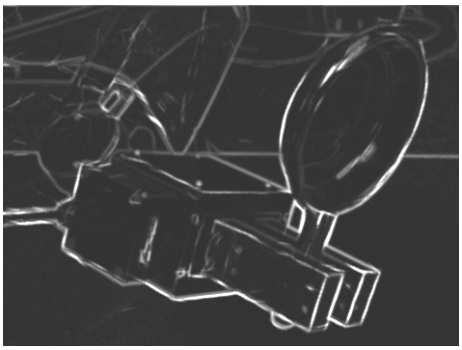
d) Enlarged c) to show details

Fig. 14 Input images for testing feedback process with RBM in a real scene.

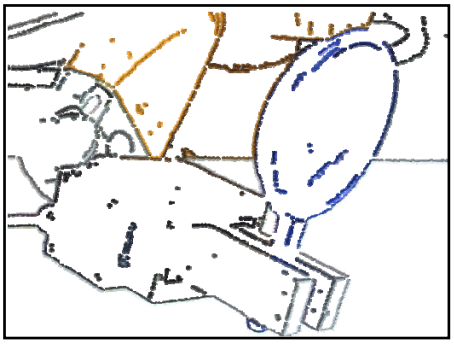
This section presents the performance of the proposed feedback mechanism on a sequence of scenes with known RBM. Such a scenario is often used in the robotic environment. Therefore, this study's findings may have an impact on robotic research efforts that focus on enhancing feature extraction. In Fig. 14, two frames are shown. The know RBM is valid for the blue pan and gripper. It can be seen that the contrast on the gripper is rather low in Fig. 14 (d) but more visible in Fig. 14(b). We apply the feedback mechanism from extracted 3D symbol-based features to enhance feature extraction of low-contrast edges. The result is shown in Fig. 15.



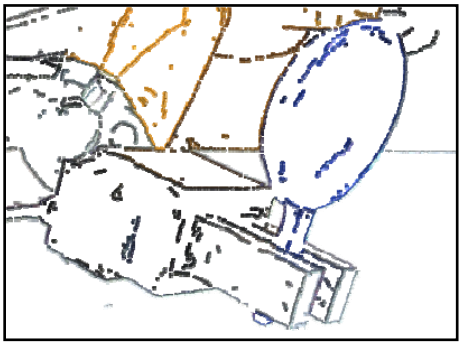
a) Magnitude before feedback



b) Magnitude after feedback



c) Primitives before feedback



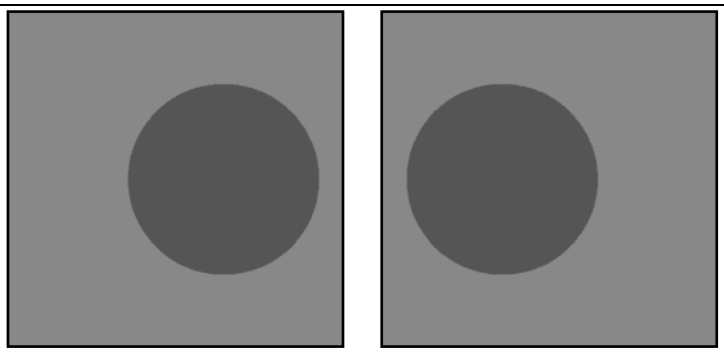
d) Primitives after feedback

Fig. 15 Results for testing feedback process based on motions in areal scene.

5.2 Feedback from Models

5.2.1 Artificial Scene

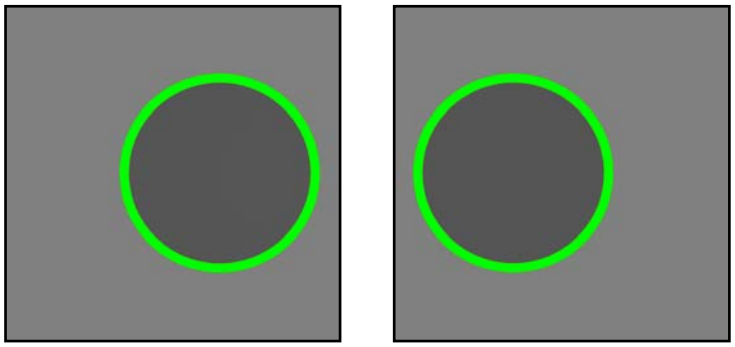
In this section, a virtual 3D circle is made by using OpenGL. Two cameras are configured to capture the object located in the virtual world. The input images are shown in Fig. 16.



a) Left image b) Right image

Fig. 16 The left and right input images of a 3D circle in an artificial scene.

To verify that the predicted 3D circle model is correct, we projected the 3D circle back to the input images as shown in Fig. 17:



a) Left image b) Right image

Fig. 17 The predicted 3D circle model is projected back to the input images in green for an artificial scene.

The feedback mechanism does not influence much in this scene, as the features are well-extracted. Hence, the results of feedback are not shown.

5.2.2 Real Scene

Circle with low-contrast parts

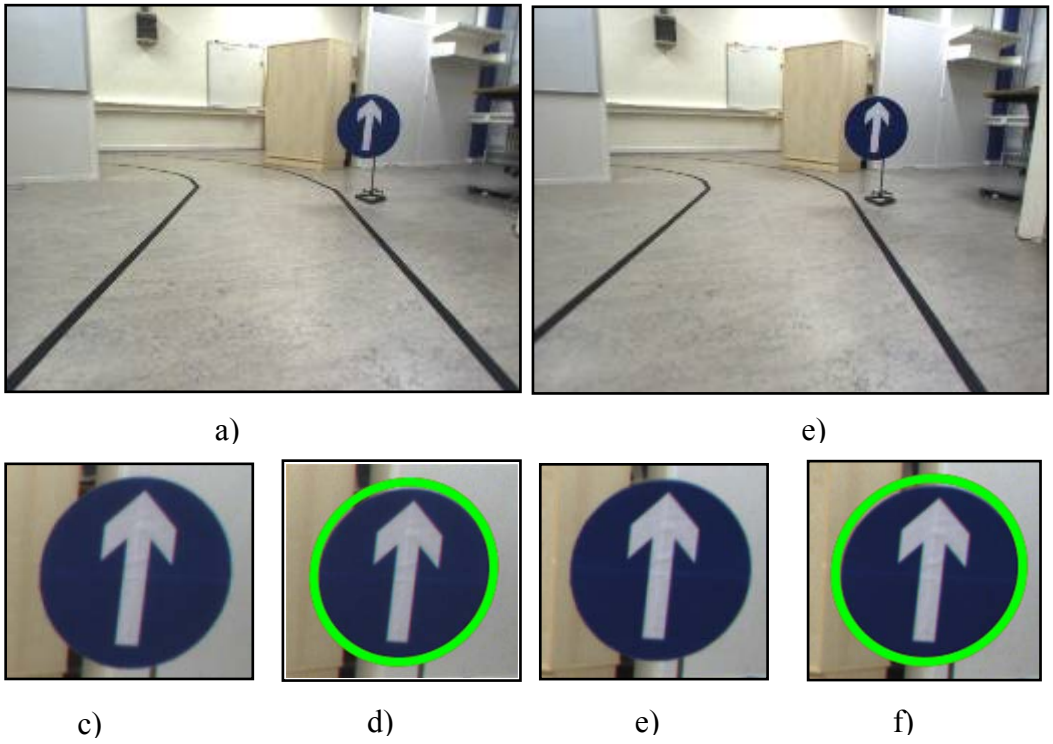
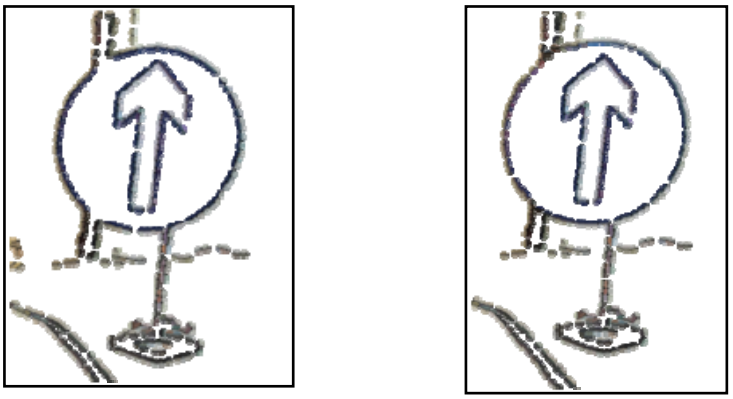


Fig. 18 a) and b) are the left and right input images for a driving scenario in a laboratory setting. The potential task is to understand the traffic sign as illustrated in c) and e). In d) and f), the extracted circle models are displayed.

Previously we have illustrated how the proposed feedback mechanism propagates the model knowledge to input images and enhance the feature extraction. This section focuses on how to influence a driving-scenario where circles are frequently shown and used as indicators. Inside the traffic sign circles, the area can be directly used as input for pattern recognition regarding traffic-related semantics. In Fig. 18, it shows a traffic scenario where the traffic sign is a "Proceed Straight Only" sign. With such a specific

scenario, traffic signs become the region of interest. Knowing the boundaries of the region of interest allows us for focusing on this specific area. Many other pattern recognition methods can benefit from such yielded "segmentation". The feedback mechanism improves the extraction of edge-structures where the contrast is low, as shown in Fig. 19. By checking at the signal level, it also assists efforts to define the region of interest in a sequence of driving scenario scenes with known or predicted RBM.



a) Without feedack b) With feedback

Fig. 19 Primitives of the enlarged traffic sign area without and with feedback

Complex Scene with Multiple Circles

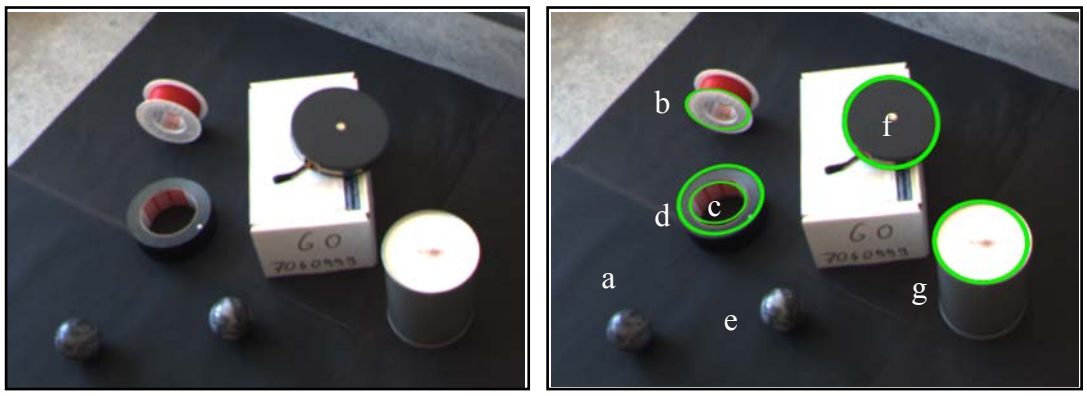
This section illustrates the performance of the proposed feedback mechanism on complex scenes where multiple circles co-exist. The results are shown in Fig. 20. To apply the feedback mechanism, our application requires a minimum of 60% of the path found by Primitives, although hypotheses can be made with a smaller coverage percentage. The reason is that it is not sufficient and persistent to predict parameterized ellipses with many invisible continuous arcs. Note that epipolar geometry provides discontinuity on the 2D contours. It is not possible to determine the stereo correspondences when the orientation is parallel to the epipolar line.

The detected circle models are marked from a) to g) in Fig. 20 (b). Five out of Eight circles are successfully detected. In Table 1, some comments are given for each circle.

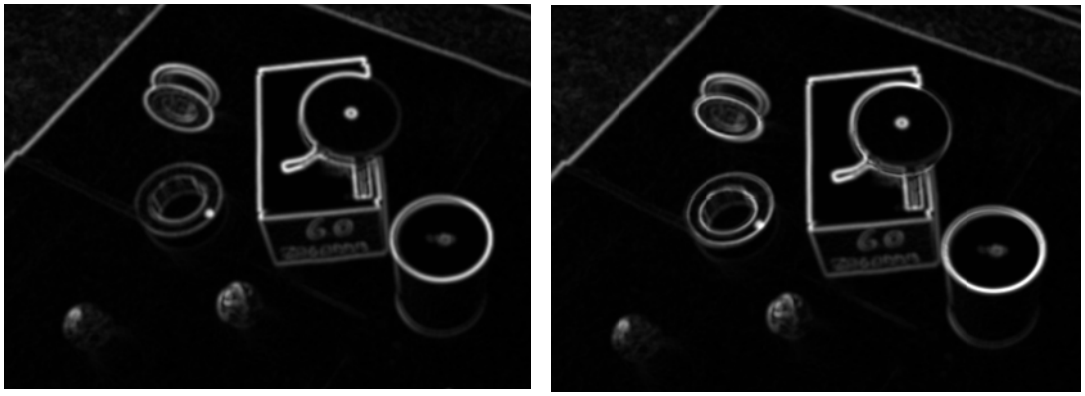
No	Description	Status
a)	Texture overwhelms the contrast of the boundary. The boundary itself cannot be detected. Not a good sample.	false
b)	Possible to detect even though many outliers exist nearby.	true
c)	Low contrast and partly interrupted by the overlapping structures.	true
d)	Low contrast on the left side and disturbance from the inner circle.	true
e)	Same as a).	false
f)	Partly no contrast, disturbance from the bottom structure.	true
g)	Good condition to extract circle.	true

Table 1 Description of detected circles in Fig. 20

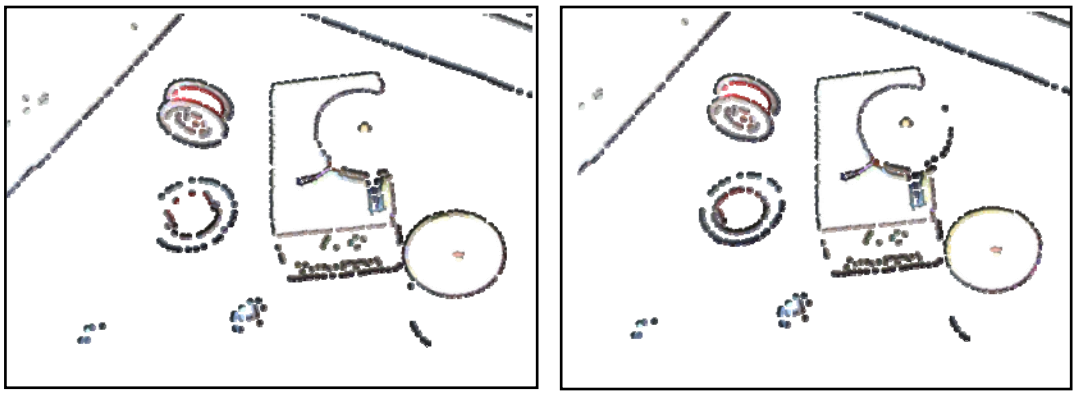
Fig. 20 gives a comparison in magnitude before and after applying the feedback mechanism in (c) and (d) and Primitives in (e) and (f). It demonstrates that the features are enhanced on the magnitude level, which leads to an improvement in Primitive extraction. Be aware that the increment of the magnitude relies on the original input, which is demonstrated in circle f).



a) Cropped right image b) Projected detected circle



c) Magnitude without feedback d) Magnitude with feedback



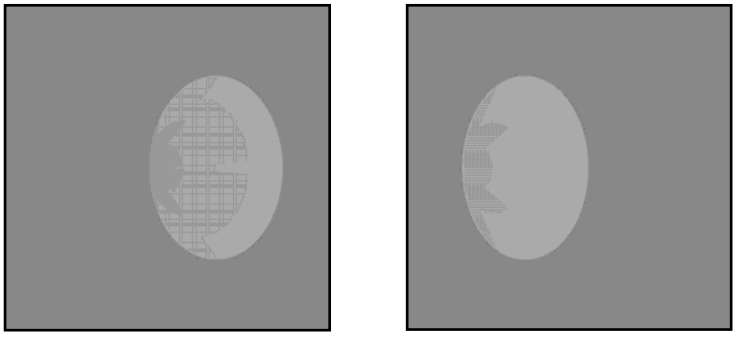
e) Primitives without feedback f) Primitives with feedback

Fig. 20 Results for a complex scene containing co-existing multiple circles.

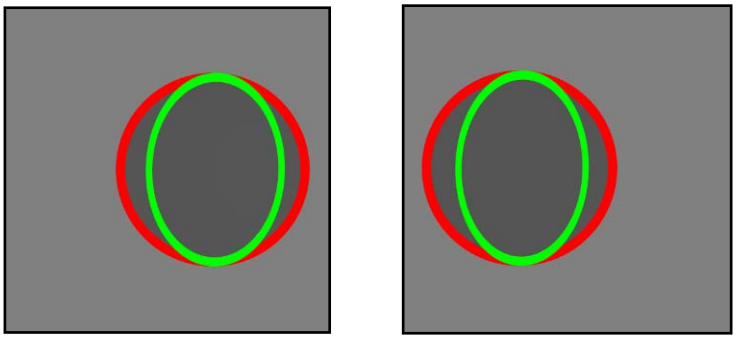
5.3 Feedback from Both Motion and Models

5.3.1 Artificial Scene

To demonstrate the feedback mechanism, we rotated the 3D circle and captured the input images in an "under-exposure" condition, so the contrast between the object and the background was relatively low; see Fig. 21 (a). As the contrast is weakened, the extracted Primitives are increasingly incomplete, see Fig. 22 (a). It can be seen that the amount of Primitives extracted after applying the feedback mechanism is increased as shown in Fig. 22 (b).



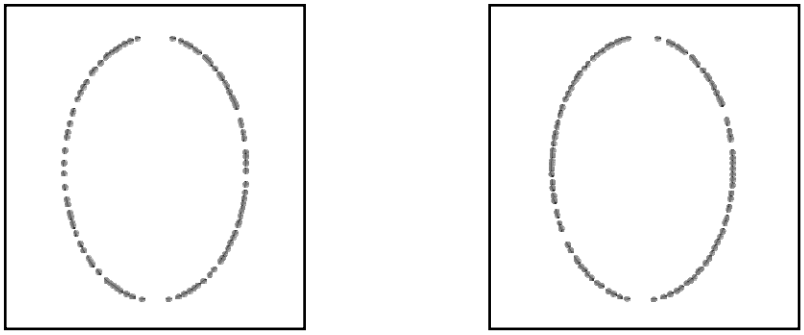
a) Left and right images with RBM from the circle



b) Predicted circle (red) and transformed circle with RBM (green) in original image.

Fig. 21 Transformed 3D circle models by RBM projected on the original frame in an artificial scene.

As discussed in Section 4.2, feedback is saved in a channel of images, which will be combined with the Gabor wavelet responses. In an artificial scenario where the RBM is already known, we can transform the detected circle defined by RBM, and enhance the feature extraction in the transformed frame. This is shown in Fig. 21 (b). Fig. 23 shows the snapshot of the feedback for the transformed circle.



a) Primitives Without feedback b) Primitives With feedback

Fig. 22 Cropped Primitives extracted without and with feedback based on models

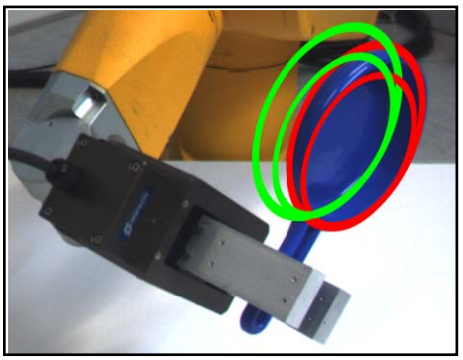
Fig. 23 Snapshots of feedback over four selected channels of the left image.

5.3.2 Real Scene

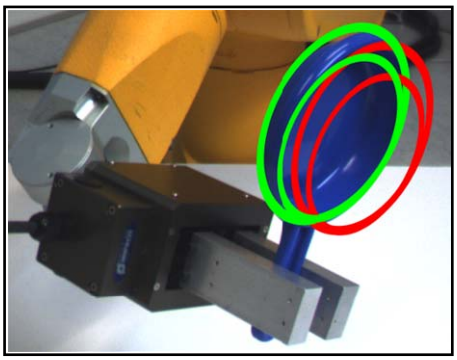
This section presents the performance of the proposed feedback mechanism from the combination of extracted 3D circles and 3D Primitives. The testing input images are identical to those described in Section 5.1.2, as shown in Fig. 14. In the 3D circle detection, two circles are hypothesized which are the inner part and the outer part of the

blue pan. From the Primitive images, it is clearly shown that some edge-structures of the pan are not found due to the reflection. With the feedback mechanism, the Primitives are retrieved which indicates that feature extraction is enhanced. Fig. 24 (a) and (b) present the stability of applying RBM with predicted 3D circle models. Here we emphasize that it is important to have the correct circle models detected in order to obtain good results from applying RBM. The check function in Section 3.5 gives circle detection the flexibility to pass more circle hypotheses by lowering the thresholds. The results here imply that the possibilities of extending the feedback model into a recurrent loop where the input frame can be used to assist in the feature extraction of the coming frame.

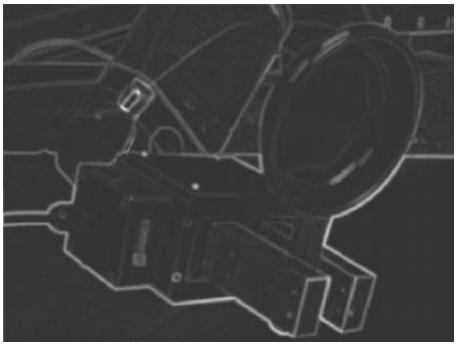
The feedback mechanism can directly operate on 3D Primitives. In addition, the feedback mechanism can indirectly operate on 3D models via circle models. It can be seen in Fig. 24 (d) and (f) that many edge structures are retrieved by applying the feedback mechanism based on Primitives without model knowledge. This solves the common problem in RBM scenarios where the features are occluded or lost during the object's motion. There are several advantages of using model knowledge as the source of the feedback. One of the advantages to feed back 3D models instead of 3D Primitives is the potential to enhance feature extraction in the current frame. As the predicted models are hypothesized from Primitives, the predicted models have more uncertainties. Analogously, Primitives are condensed sparse symbolic-based representations computed from signal-based representations and tend to be more stable. The feedback mechanism itself also constrains the propagation area close to the extracted Primitives. Using model knowledge to feed back also gains stabilities due to the flaws in visual processing, such as stereo matching, noises, shading, etc.



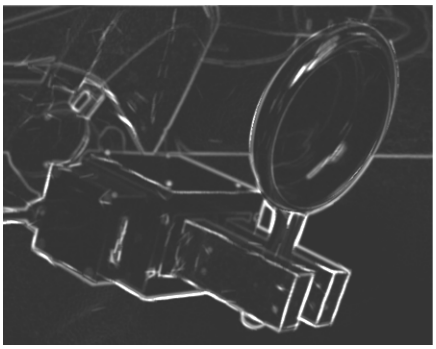
a) Circles on the first frame, red for predicted circles, and green for transformed circles



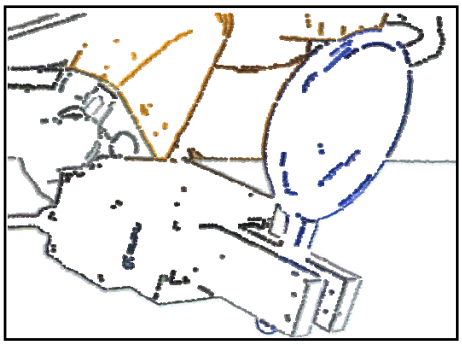
b) Circles on the first frame, red for predicted circles, and green for transformed circles



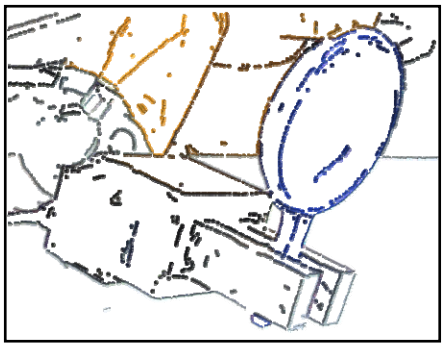
c) Magnitude before feedback



d) Magnitude after feedback



e) Primitives before feedback



f) Primitives after feedback

Fig. 24 Results for testing a circle model with RBM in a real scene.

6 Discussion

The current work proposed a feedback mechanism between symbolic-based representation and signal-based representation, to retrieve lost features due to low contrast. The feedback is used to enhance feature extraction where input images are often contaminated. Prior studies have been further investigated and extended so that the feedback mechanism can not only handle motion-based stimuli but also model-based stimuli, as described in [19][18]. We studied the 3D circle model and the methods of detecting 3D circle models and their projected 2D ellipse models. The novelty of the current work includes the establishment of a feedback protocol between different layers, i.e., signal-based feature layer, symbol-based feature layer and model-based feature layer. To ensure the reliability of the feedback source, we eliminated redundant predicted models, by using a signal-based ellipticity check function. We also presented how the model propagates the model knowledge with Gabor wavelet responses using a multi-dimensional Gaussian function. This current work took 3D circles to exemplify the feedback mechanism of model knowledge and demonstrated that 3D circles can be migrated to other types of model knowledge without difficulty.

In the experiments, we illustrated the performance of the proposed mechanism on both artificial scenes and real scenes. To demonstrate the extendibility of the feedback mechanism, we used both artificial-scene and real-scene sequences from which the RBM between the frames is known. Moreover, the application shows a strong adaptability in cooperation with RBM in the real scene sequence, as it not only influences the frame-independent knowledge in the time domain, but also eliminates the crude models by checking against signals in both the current working frame and prediction frame. The concept of the current work is to re-check the model knowledge generated from bottom-up approaches by validating the models against low-level

image processes. This innovative methodology is enabled by the incomplete status of Primitives, which are encapsulated from signal-based representation. The feedback mechanism requires additional computational overhead, which may become an issue when using it in real-time applications. However, due to the rapid growth of the processing abilities of computers, the author is optimistic that other applications could benefit from the current work.

The current work benefits from the existing software infrastructure in MoInS, in which the symbolic-based representation is suitable to investigate the feedback mechanism. To make the application reusable and robust, we also designed and implemented some additional software components, which extended MoInS from a software engineering point of view. These extensions allows users to apply other models in addition to circle models in the similar circumstances.

Potential future works include attempting feedback with other signal-based visual process layers. It would also be interesting to alter the feedback loop so it functions in a recurrent manner and then test it on an image sequence.

Appendix

A. Software Design

A.1 Introduction

This project was conducted using the C++ programming language under the Linux operation system. The project was based on a fundamental software system, called Modality Integration Software (MoInS) which has been currently being merged into Cognitive Vision Software (CoViS) [14]. MoInS was a joint work among several Ph.D. students in Kiel University in the 1990s. Norbert Krüger and his research group has further developed MoInS since 1998. MoInS itself covers a wide variety of different domains within the field of computer vision, image process and cognitive studies. Many developers working on various ongoing projects continue to develop and evolve the MoInS software structure with the goal of making it faster, more robust, and increasingly user friendly. This chapter outlines the project's software engineering design, and the implementation. In addition, this chapter discusses how the project customized and extended MoInS.[1]

A.2 Detailed Design Specification

The software design as described in the class diagram in Fig. 25 makes the application more robust and maintainable. Each class is discussed in detail.

[1] For more details about MoInS, please visit http://www.mip.sdu.dk/covig/

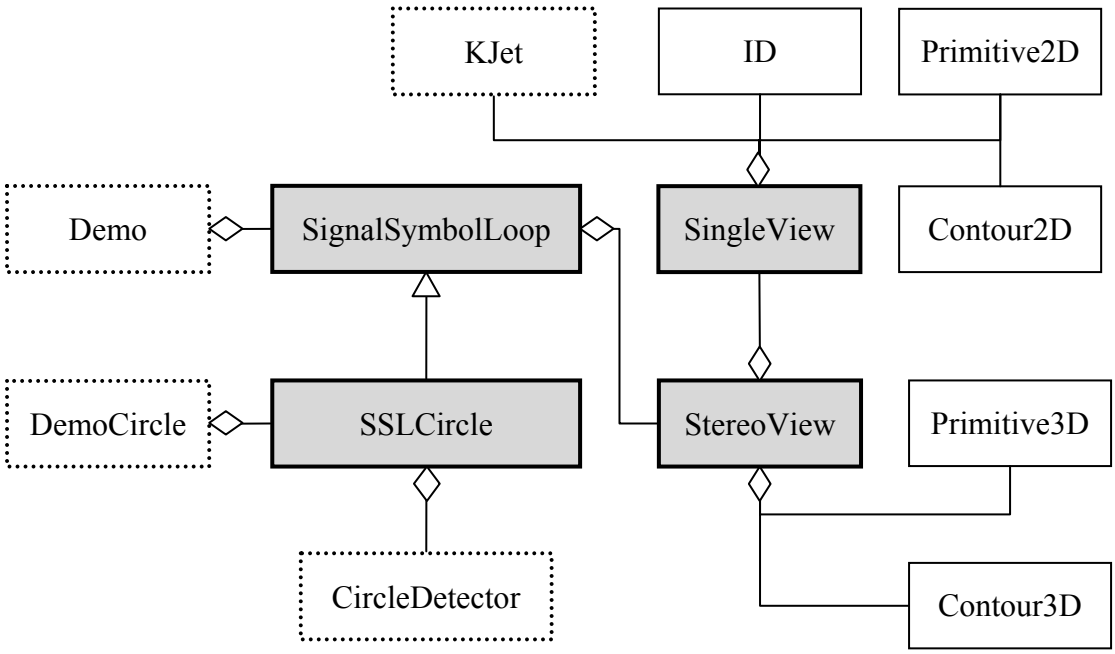


Fig. 25: The abstract class diagram includes classes made for the signal symbol loop in the current work. The classes with shading are the core classes implemented in the current work. The classes marked with dash lines are the auxiliary classes supporting the core classes. The classes in the ordinary lines are the ones implemented in MoInS that are considered to be important classes for the current work.

A.2.1 KJet

Description: The KJet class contains process units for filtering a gray scale image with Gabor wavelets into Gabor responses. Gabor responses are complex image channels where the number of channels is the number of orientations. The number of orientation was set to eight in the current work. Gabor responses can be considered to be a piece of middleware from which magnitude, orientation and phase images are generated. A SingleView class normally contains one instance of KJet. Computations are started after the image is fed into the instance. In the context of the signal symbol loop, after multichannel feedback has been merged with the Gabor responses, the KJet instance

updates the result images.

Input: A gray scale image.

Output: Magnitude, orientation and phase images

A.2.2 CircleDetector

Description: The CircleDetector class is implemented to find the hypotheses of 3D ellipse models. The method of circle extraction consists of a series of algorithms, including:

- Finding 3D corresponding contours
- Combing contours
- Fitting ellipses to combined contours
- Eliminating false ellipses in 3D
- Projecting 2D ellipses
- Eliminating false ellipses in 2D

Input: A color image, 3D contour, 2D contour, 2D Primitives and thresholds, which are used in the above processes.

Output: An array of predicted circle models parameterized in both 2D and 3D.

A.2.3 SignalSymbolLoop

Description: The SignalSymbolLoop class includes the algorithms relating to the feedback mechanism. It is used as a base class that can be inherited by other classes with specific feedback mechanisms or object models. The class and its derived classes contain the core framework of this project. To impose the feedback mechanism in a time domain, a source StereoView object and a target StereoView object are instantiated with a RBM matrix presenting their position relationship (each StereoView object can be regarded as a frame in the time domain). The loop starts by processing the source frame and creating feedback according to the similarities between source frame

and target frame. Finally, the loop merges the feedback with the target frame. By replacing the target frame as source frame and introducing a new target frame, this class can be further utilized as a tool that is repetitively invoked so one frame can implicitly influence more than just the next frame.

Input: Two color images and a RBM matrix for the transformation between them, parameters for configuring the feedback processes.

Output: The class has a standard output function, which results a complete set of properties including images from different layers, middleware, Primitives in 3D, parameterized models, etc.

A.2.4 SignalSymbolLoop_ Circle

Description: The SignalSymbolLoop_Circle class is a derived class of the SignalSymbolLoop class meaning that most variables (except private members) and methods are inherited. The purpose of extending the SignalSymbolLoop class is to expand the functionality of the base class while retaining the code infrastructure of the base class. It ensures that common features are synchronized and redundant code is minimized. The SignalSymbolLoop class focuses on handling Primitives by implementing the top layer models. The SignalSymbolLoop_Circle class treats the circle as the "model" and decomposes the circle model into an array of Primitives. Some methods are overridden, such as checking validity, feeding back models and transforming models, etc.

Input: Identical as input in Appendix A.2.3

Output: Circle models, remains are identical as outputs in Appendix A.2.3

A.2.5 SingleView

Description: The SingleView class is a wrapper class for resultant images, attributes and properties in a one-camera environment. It contains the outputs from the following

processes:

- Filtering a gray image with Gabor wavelets and resulting Gabor responses, which are comprised of complex image channels.
- Computing magnitude, orientation and phase by using the complex image channels.
- Computing intrinsic dimensionalities as normalized feature-based indicator.
- Condensing pixel-wised information into "Primitive" s, which focuses on the coarse features of the local neighborhood using symbolic representation.
- Linking the individual and standalone "Primitive" s into grouped contours.

Input: A gray scale image and related configuration settings for the afore mentioned processes.

Output: Gabor responses, magnitude, orientation, phase images, an array of Primitives and an array of grouped contours.

A.2.6 StereoView

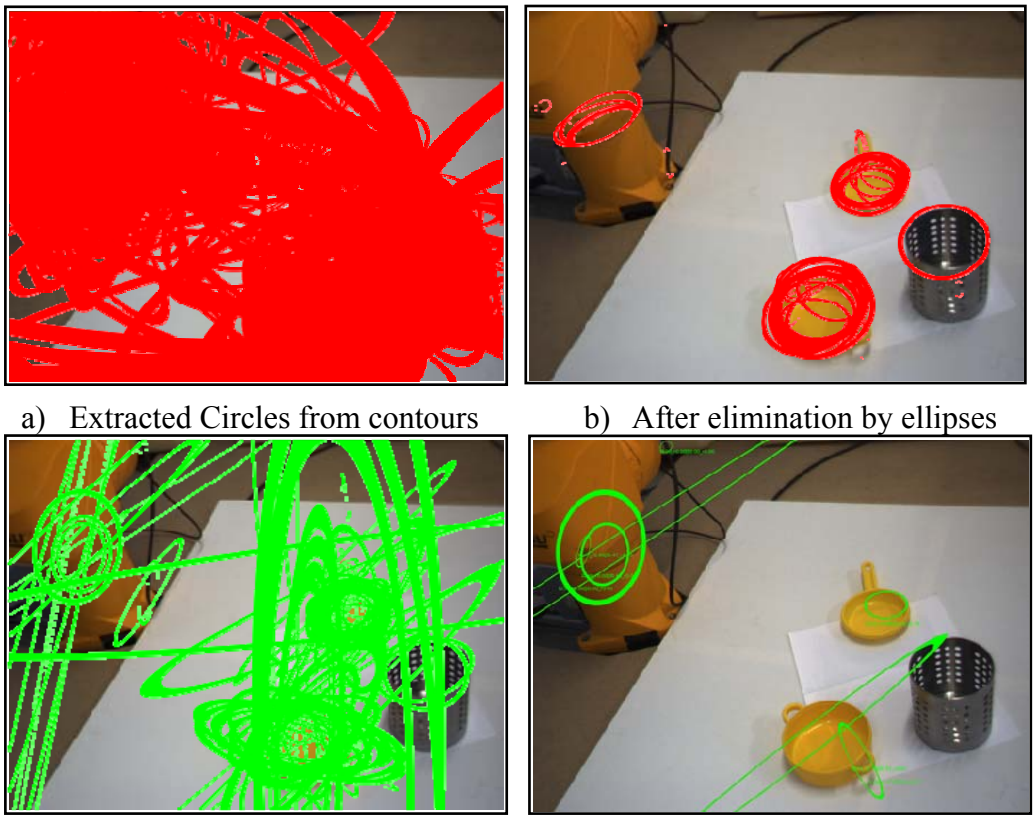
Description: The StereoView class is a wrapper class gathering the output and processes for computing 3D information using multiple SingleViews (the current work only considers situations involving of two cameras). Feature-based or symbol-based stereopsis allows the users to run the application in real time. By matching 2D Primitives in left and right cameras with their respective projection matrices using epipolar constraints, an array of 3D Primitives is computed. 3D grouped contours are linked to these 3D Primitives. Each StereoView instance represents a snapshot of synchronized left and right SingleViews, greatly reducing the complexity of relationships between the different frames. It also enhances the coherence and weakens the coupling among the different frames.

Input: Left and right color images with their respective SingleViews and configuration setups for the relevant processes.

Output: An array of 3D Primitives and grouped 3D contours.

B. More results for circle detections

B.1 Scene 1



a) Extracted Circles from contours

b) After elimination by ellipses

c) Further elimination in 3D

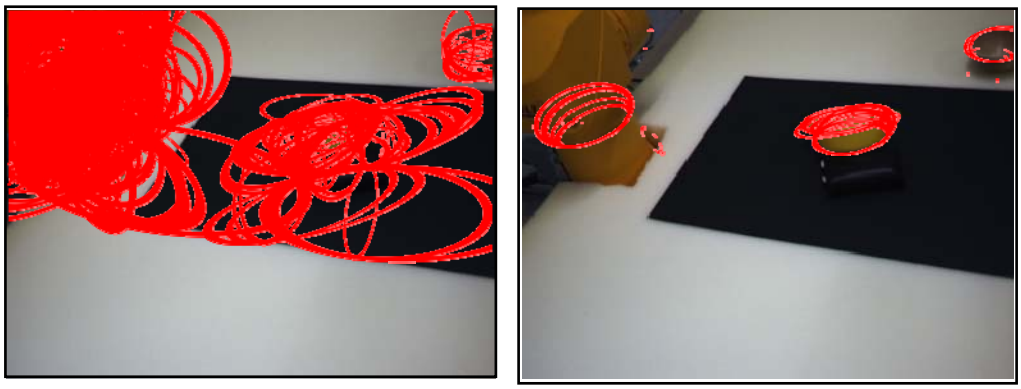
d) By ellipticity-check function with threshold 0.0-0.5

e) By ellipticity-check function with threshold 0.7-0.8

f) By ellipticity-check function with threshold 0.8-1.0.

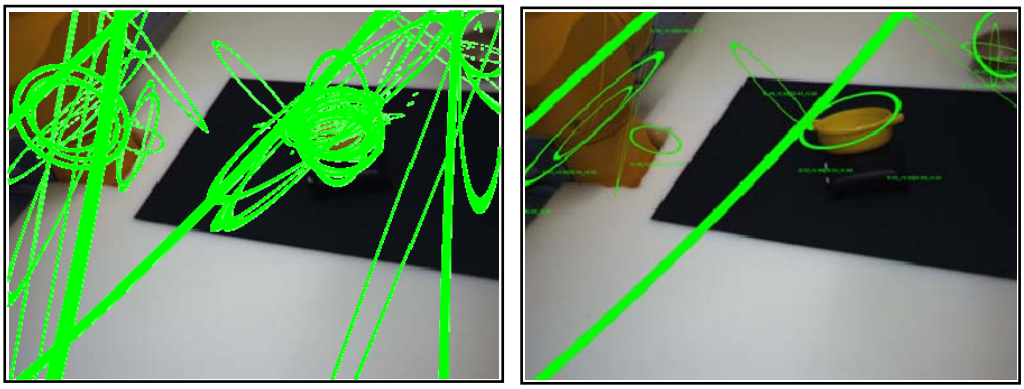
Fig. 26 Extra Results of Scene 1 for testing circle detection.

B.2 Scene 2



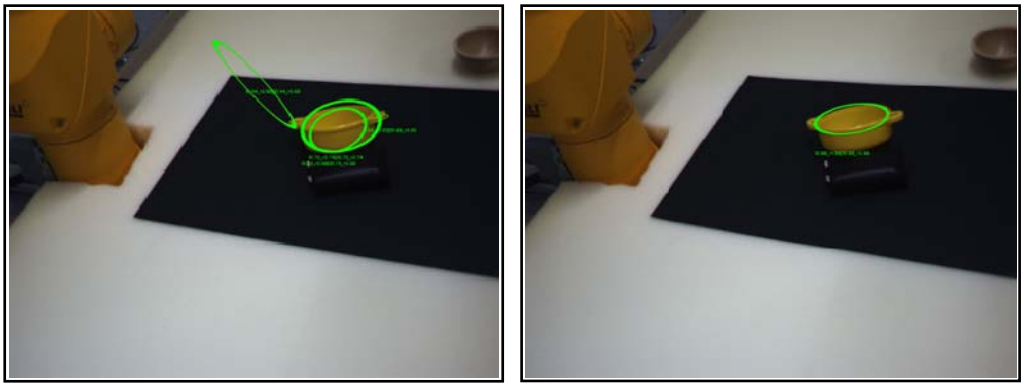
a) Extracted Circles from contours

b) After elimination by ellipses



c) Further elimination in 3D

d) By ellipticity-check function with threshold 0.0-0.5



e) By ellipticity-check function with threshold 0.7-0.8

f) By ellipticity-check function with threshold 0.8-1.0.

Fig. 27 Extra results of Scene 2 for testing circle detection.

References

[1] J. Canny., "A Computational Approach to Edge Detection." *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 6, 1986, Vol. 8.

[2] Stephen M. Smith and J. Michael Brady., "SUSAN - A New Approach to Low Level Image Processing." *Int. J. Comput. Vision.* 1, 1997, Vol. 23, 0920-5691, pp. 45--78.

[3] B. Jähne., *Digital Image Processing. Springer.* 2002. pp. 436-439. 3-540-67754-2.

[4] P.V.C. Hough., "Methods and Means for Recognizing Complex Patterns." *U.S. Patent 3,069,654, Dec. 18.* 1962.

[5] O.D. Faugeras., *Three--Dimensional Computer Vision. MIT Press.* 1993.

[6] David J Fleet, Michael J Black, Yaser Yacoob, Allan D Jepson., "Design and Use of Linear Models for Image Motion Analysis." *International Journal of Computer Vision.* 2000, Vol. 36.

[7] M. Irani and S. Peleg., "Motion analysis for image enhancement: Resolution, occlusion, and transparency." *JVCIP.* 1993, Vol. 4.

[8] Nicholas D. Molton and Andrew J. Davison and Ian D. Reid., "Locally Planar Patch Features for Real-Time Structure from Motion." *Proc British Machine Vision Conference 2004.*

[9] M.O. Ernst, M.S. Banks,., "Humans integrate visual and haptic information in a statistically optimal fashion." Nature, 2002, Vol. 415.

[10] P. Bayerl and H. Neumann., "Disambiguating visual motion by form--motion interaction -- a computational model." *International Journal of Computer Vision (Special Issue).* 2006.

[11] S. Kalkan., S. Yan, F. Pilz, N. Krüger., "Improving Junction Detection by Semantic Interpretation." *Int. Conf. on Computer Vision Theory and Applications (VISAPP'07).*

[12] N. Krüger., "Three Dilemmas of Signal- and Symbol-based Representations in

Computer Vision." *Proceedings of the 1st International Symposium on Brain, Vision and Artificial Intelligence 19-21 October, 2005, Naples, Italy, Lecture Notes in Computer Science, Springer, LNCS 3704.* 2005.

[13] P. König and N. Krüger., "Perspectives: Symbols as self-emergent entities in an optimization process of feature extraction and predictions." *Biological Cybernetics.* 2006.

[14] E. Baseski, S. Kalkan, L. B. W. Jensen, A. K.-Nielsen, D. Kraft, N. Krüger, F. Pilz, N. Pugeault and M. Skov., "User Manual for Cognitive Vision Software." [Online] http://www.mip.sdu.dk/covig/publications/Covis_User_ManualNK.pdf.

[15] N. Krüger, M. Lappe and F. Wörgötter., "Biologically Motivated Multi-modal Processing of Visual Primitives." *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour.* 2004.

[16] Dirk Walther., "Interactions of visual attention and object recognition : computational modeling, algorithms, and psychophysics." *PhD thesis.*

[17] R Fay, U Kaufmann, A Knoblauch, H Markert, G Palm., "Combining Visual Attention, Object Recognition and Associative Information Processing in a NeuroBotic System." *LECTURE NOTES IN COMPUTER SCIENCE.* 2005.

[18] Shi Yan., *Feature Extraction by Interaction of Visual Cues and Motion Cues.* 2007.

[19] —. "Feature extraction enhancement based on signal-symbol loop exemplified by a vision based robotic system." *2nd 7SEMCON.* 2007.

[20] S. Kalkan ; S. Yan ; V. Krüger ; F. Wörgötter ; N. Krüger., "A Signal-Symbol Loop Mechanism For Enhanced Edge Extraction." *Int. Conf. on Computer Vision Theory and Applications (VISAPP'08).*

[21] M. Felsberg and N. Krüger., "A probablistic Definition of Intrinsic Dimensionality for Images." *Pattern Recognition, 24th DAGM Symposium.* 2003.

[22] E. Baseski, D. Kraft, N. Krüger., "A hierarchical 3d circle detection algorithm applied in a grasping scenario." *(submitted to) 6th Int. Conference on Computer Vision*

Systems (ICVS).

[23] M. Felsberg, N. Krüger., "A continuous Formulation of intrinsic Dimension." *Proceedings of the British Machine Vision Conference.* 2003.

[24] N. Pugeault and F. Wörgötter and and N. Krüger., "Multi-modal Scene reconstruction using Perceptual Grouping Constraints." *Proceedings of the 5th IEEE Computer Society Workshop on Perceptual Organization in Computer Vision, New York City June 22, 2006 (in conjunction with IEEE CVPR 2006).*

[25] B. MacLennan., "Gabor Representations of Spatiotemporal Visual Images." *Computer Science Department technical report CS-91-144.* 1991.

[26] Tai Sing Lee., "Image Representation Using 2D Gabor Wavelets." *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 1996.

[27] T. Kasamatsu, U. Polat, M.W. Pettet and A.M. Norcia., "Colinear facilitation promotes reliability of single-cell responses in cat striate cortex." *Experimental Brain Research.* 2004.

[28] L. Haglund and D. J. Fleet., "Stable Estimation of Image Orientation." *Proceedings of the IEEE-Proceedings of IEEE International Conference on Image.* 1994.

[29] M. Pilu, A. Fitzgibbon, R.Fisher., "Direct least-squares fitting of ellipses." *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 1999.

[30] P. L. Rosin., "Further Five Point Fit Ellipse Fitting." *Graphical Models and Image Processing.* 1999.