Copyright Statement

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

This thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author's right to be identified as the author of this thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author's permission before publishing any material from their thesis.

To request permissions please use the Feedback form on our webpage.
http://researchspace.auckland.ac.nz/feedback

General copyright and disclaimer

In addition to the above conditions, authors give their consent for the digital copy of their work to be used subject to the conditions specified on the Library Thesis Consent Form and Deposit Licence.
Accuracy

of Stereo-Based Object Tracking

in a Driver Assistance Context

PhD Thesis

by

Waqr Khan

Submitted in Partial Fulfillment of the
Requirements of the
Degree of Doctor of Philosophy

Department of Computer Science
The University of Auckland
New Zealand
August 2013
This thesis is dedicated to my family. Each sentence I have written here is at the cost of time I spent away from them.
Abstract

Stereo vision is currently used in the car industry as a tool for designing driver-assistance systems. However, limitations inherently due to the discrete nature of disparities observed by a stereo-vision system have not yet been modelled and analysed so far; this thesis aims at closing this gap.

Stereo-vision results are used in driver-assistance systems for estimating trajectories or just speed. Besides accuracy limitations in stereo matching, the discrete nature of disparities also defines limitations to detected trajectories or speed.

This thesis proposes and discusses a novel tool for a safety engineer which permits the safety of these driver assistance systems to be estimated. It is based on a model which considers the true error in measured velocities of objects. Outputs from this tool show that the choice of stereo-system parameters, so as to optimally place the disparity change boundaries, is critical to the effectiveness of such a system. As soon as the possibly colliding object crosses one of these boundaries, the range of possible trajectories for a (possibly colliding) object reduces significantly.

This factor also means that larger objects (e.g. trucks) are slightly better tracked by stereo vision than smaller ones (e.g. signs or pedestrians). Completely safe stereo-based systems are also shown to issue many precautionary (and ultimately unnecessary) warnings if the stereo parameters are not chosen carefully.

Keywords: Stereo analysis, accuracy, driver assistance, object tracking, Kalman filter
Acknowledgments

First of all, I praise Allah, the almighty and the most merciful, who gave me this opportunity and granted me courage and determination to successfully complete this study.

I would like to express my gratitude to Professor Reinhard Klette, my main supervisor, in supporting me in my final stages of my research and thesis writing. I also like to thank him for introducing me to stereo-based driver assistance systems, and for letting me use HAKA1 for recording crucial experimental sequences.

I also like to thank Associate-Professor John Morris, my co-supervisor, for introducing me to the problem of accuracy in stereo measurements. His engineer’s perspective lead to a critical analysis of stereo accuracy in the driver assistance context.

I thank
- Simon Hermann for providing his iSGM executable,
- Veronica Suaste and Diego Caudillo for their collaboration in testing and evaluating stereo matchers linBP and iSGM on real-world data and joint work while preparing our publication at IEEE IV 2013,
- Junli Tao and Tariq Khan for assisting me when doing the HAKA1 sequence recording and choreographing collision scenario,
- Khurram Jawed, Tariq Khan, Ratheesh Kalarot, and Usman Butt for support when recording various sequences indoor and outdoor, and the quality time we spent together,
- YeoJin Jo, for assisting me with interdepartmental issues,
- my wife Mah Gul Riaz for supporting me in all ups and downs of life, my parents for their prayers and encouragement, my sister and brothers for their support, and
- my GP Brent Maxwell and vascular surgeon Murray Maccormick for keeping me healthy and fit during the course of this research.

A big thank-you also to the .enpeda. team; I enjoyed the time I spent with you all during hiking, joint lunches, BBQs, and all those various occasions at Tamaki campus and elsewhere.

Waqar Khan
Auckland
August 14, 2013
Contents

Abstract iii
Acknowledgements v

1 Stereo and Motion in Driver Assistance 3
  1.1 Motivation ......................................................... 3
  1.2 Common Terms .................................................... 6
  1.3 Structure of this Thesis ......................................... 6
  1.4 Main Contributions of this Thesis ............................. 7

2 Used Stereo Matchers and Test Data 11
  2.1 Introduction ...................................................... 11
  2.2 Stereo Matching Algorithms .................................... 13
    2.2.1 Dynamic Programming Stereo .............................. 14
    2.2.2 Semi-Global Matching .................................... 16
    2.2.3 Belief Propagation ......................................... 17
  2.3 Continuity Cost Functions ..................................... 19
  2.4 Data Cost Functions for Stereo Matching ...................... 20
    2.4.1 Pixel-based Cost Functions ................................ 20
    2.4.2 Window-Based Cost Function ................................ 21
  2.5 Results of Comparison and Discussion ........................ 22
    2.5.1 Experiments With the Middlebury Dataset ................. 23
    2.5.2 Experiments With the EISATS Dataset ........................ 25
    2.5.3 Experiments With the KITTI Dataset ........................ 26
    2.5.4 The Third-eye Approach .................................... 29
    2.5.5 Data Measures ............................................... 33
Chapter 1
Stereo and Motion in Driver Assistance

Driver assistance systems are designed to reduce the road users fatality risk. Already, laser-sensor based systems are providing very accurate distance and time to collision measurements. But, because of the poor performance in worst case environmental conditions like rain, snow etc., they are not a robust option. Another alternate is to use a pair of video cameras as a stereo system. In this system, due to discrete pixels, the depth is measured at discrete steps. The step size can be changed by changing the stereo equipment. However, due to a highly competitive automotive industry, the equipment should fulfil the budget too. The equipment should also fulfil the safety engineer criteria for designing a safe collision warning system. This chapter describes the motivation for developing a stereo driver assistance system and the structure of this thesis.

Parts of this chapter have been published in IEEE Image and Vision Computing New Zealand 2009. Paper \([3]\) was a joint work with John Morris and Reinhard Klette.

1.1 Motivation

According to the World Health Organization (WHO), around 1.3 million people die annually due to traffic accidents \([129]\), forty six percent of them are pedestrians, cyclist and motorcyclists. If no further safety measures are undertaken, then this annual rate would increase to 1.9 million people by the end of year 2020. There are several key factors that cause road accidents and fatalities: drunk driving, speeding, helmet-less cycling, lack of seat belts, and child restraints. WHO also reports that only fifteen percent of the countries enforce suitable laws to reduce these risk factors \([129]\).

There are various factors that could distract an active driver from safely avoiding a collision scenario. For example, the driver can be distracted due to in-car factors including mobile text messaging or attending to a call, tuning a radio, handling a GPS navigator, talking with passengers etc. Similarly, out of car incidents can also cause a distraction, for example, incident on the road, sun glare, driver blind spots etc. In case of a lazy driver the risk due to these factors are further heightened.
To mitigate these factors, some additional safety measures are already being taken into consideration. A few examples include: Antilock Braking Systems (ABS) which protects in cases of slippery surface and while cornering and braking; Adaptive Cruise Control (ACC), mainly applicable on highways, as it works out the distance to and the speed of the vehicle in front and adjusts the speed accordingly. These approaches decrease the time-to-collision (TTC), but do not guarantee that the driver is also paying sufficient attention to a possible collision scenario. With ABS installed on the majority of vehicles, in the United States alone, the Department of Transportation reported that in 2005, around 80-90% of traffic accidents were caused by driver errors [100].

Consequently, intelligent vehicle safety systems have gained more research interest in the area of intelligent transportation systems [32, 92, 93, 33]. There are two types of collision avoidance systems - systems that just warn the driver of a possible collision scenario, and more intelligent systems where the brakes are applied automatically without relying on the driver. Both types of systems have to measure the distance, direction and TTC.

Various systems accurately estimate these factors through range based sensors like laser [102, 52, 125] and radio detection and ranging (RADAR) [48]. However RADAR works only in a single direction, so has a limited extent of field of view. Whereas laser although has wider field of view, still fails in environments where signal is distorted due to rain, snow, or even distant objects with poor optical scattering.

A pair of video cameras capturing the same scene is another, stereo vision based alternative. Already, there have been numerous algorithms proposed for deriving depth maps from stereo pairs [111]. Stereo vision is also being actively studied for safety systems [21, 53, 65, 11, 12]. However, there have been relatively few studies of the depth accuracy of stereo vision systems in driver assistance context.

In this study, I have analysed accuracy of stereo vision in a driver assistance context as a collision warning system. The accuracy is determined as a whole for colliding objects appearing from anywhere in front of the system at any speed. Figure 1.1 shows an example of my typical camera set-up, as commonly used in stereo vision, and also in my models and experiments. Cameras are aligned in canonical stereo configuration geometrically defined by ideal central projection into both image planes, which are assumed to be coplanar with epipolar lines passing through the centres of pixels in the same image row in both images.

Figure 1.1 shows sets of rays, drawn through pixels of the virtual image planes. Scene points at which these rays intersect are imaged onto the centres of image pixels. The figure illustrates epipolar geometry as defined by the canonical stereo configuration. A pair of corresponding pixels combines a pixel in the left and a pixel in the right image, both on the same epipolar line, and both being a projection of the
1.1. Motivation

Figure 1.1: Canonical stereo configuration: The decreasing depth resolution (larger $\delta Z$) for larger $Z$ values is evident [98]. The figure only shows one epipolar plane. The third co-ordinate axis (the $Y$-axis) points towards the viewer.

Disparity is the difference in corresponding pixel positions. Depth (or distance) is defined by the distance between a pixel and the projected scene point. Depth can easily be calculated from a known disparity, as known in stereo photogrammetry for more than one hundred years [46]. The challenge is still to obtain (very) accurate disparities automatically at all positions of corresponding pixels.

On the surface, stereo photogrammetry would appear to have the right charac-
teristics for an autonomous vehicle navigation system - the depth accuracy might be low at far distances but improves exponentially as an object gets closer, enabling more and more precise estimates of the likelihood of a collision to be made as the objects comes closer \([79,55]\). However, since measured disparity values are integral, measured depths belong to a discrete set \([88]\). This introduces a subtle trap: in the distance, when it would be desirable for the system to provide an early warning to allow more time to plan avoidance strategies, an object which is on a collision course might appear to be not moving at all with respect to the system. This problem is a general one for all vision guided autonomous navigation, but, because of its enormous social and economic costs, the examples in this study are based on a driver assistance system which is implemented in a vehicle. In particular, one function of such a system can be to warn a driver about an imminent collision \([21,45]\).

In this thesis I study the problem of rapidly decreasing accuracy of distance measurements derived from a stereo system, as distance to an 'object of interest' increases. Thus, I am concerned to determine the accuracy of the estimated trajectory of the object compared to its actual trajectory for a collision scenario. In this thesis it is assumed that images are rectified: distortions due to real cameras and lenses and practical alignment systems have been removed already.

### 1.2 Common Terms

The ego-vehicle refers to the vehicle on which the stereo vision based collision warning system is mounted. Typically, this would be a truck or a car.

The feature point is defined as any feature in the image that can be uniquely distinguished and located over the course of time. For example, it can be a corner of a silhouette, or a special distribution of intensity values around a pixel defining it to be a “landmark”.

The object is defined as any hazard in front of the ego-vehicle, such as another vehicle, a pedestrian, some part of the road furniture, or unexpected debris. Later in Chapter 3 and Chapter 4 I refer to an object by a set of feature points. The nearest of those feature points is the reference point for that object.

### 1.3 Structure of this Thesis

Determining the accuracy is not a trivial problem as the stereo system works in a noisy environment. The video camera sensor is composed of discrete pixels, so we are limited to using a discrete representation of a continuous scene. Furthermore, due to limited range of focus, areas in a dynamic scene are often blurred. There
are also other factors like electronic sensor noise and lens distortion. Current stereo systems face these common challenges, and then there are other challenges within object detection and tracking.

In the thesis, at first the accuracy of stereo matching algorithms is evaluated. Then it is assumed that the stereo system is working in a noise and distortion free environment without any image blur or stereo mismatches. The efficacy of a stereo based driver assistance system is modelled in such environment. Later, a real environment with the possibility of noise is considered. The thesis is structured as follows:

Chapter 2 covers the literature on various stereo matching algorithms. It also covers the robustness of matching algorithms with respect to difficult scenes.

Chapter 3 elaborates the problem statement by modelling a stereo system in a dynamic stereo environment. I provide the stereo model with a range of physically reasonable values and look for scenarios in which a stereo based object trajectory estimator could fail.

Chapter 4 generalizes the theoretical model covered in the previous chapter by considering more general collision trajectories.

In Chapter 5, stereo matching algorithms and object tracking are used to determine the object trajectory. On a preplanned sequence, I investigate whether the stereo based safety system can issue timely warnings.

The thesis ends with a chapter providing conclusions. See also an Appendix for additional figures, a list of symbols used in the thesis, and a generated index.

## 1.4 Main Contributions of this Thesis

This thesis reports about the following novel main contributions to the field of stereo vision and vision-based driver assistance:

1. I developed a belief-propagation-based stereo matcher, called linBP. linBP was ranked 13th on 5th Jan, 2013 on the KITTI Benchmark Suite, see [74]. Its performance was comparable in experiments to iSGM, as reported in the thesis; iSGM was ranked second on 5th Jan, 2013 on KITTI, see [75]. Using the third-eye approach, it was found out that linBP offered even better matching results compared to iSGM on challenging datasets (e.g. images with sun glare) using a normalized cross-correlation performance measure.

2. I developed a tool for the safety engineer to assess the effects of various stereo configuration parameters such as focal length, baseline length, and image resolution against other criteria like opposing object constant collision velocity.
3. I generalized the aforementioned tool to consider an opposing object with constant speed and variable velocity.

4. I developed a stereo-vision-based collision warning system. After recognizing an imminent head-on collision scenario, the system issues a timely braking warning for the driver to apply brakes and avoid collision.

Those contributions have been published in [3 4 5 6 7 8 10 11 12] at peer-reviewed international conferences; see also the list of my co-authored publications.
Chapter 2

Used Stereo Matchers and Test Data

This chapter briefly recalls (for later reference) three common stereo matchers, namely dynamic programming stereo, semi-global matching stereo, and belief propagation stereo. The objective of this chapter is also to contribute to the current discussion on how to evaluate the accuracy of various stereo matchers under varying conditions on “long” stereo sequences, not just on a few pairs of stereo images. This chapter considers various cost functions for each stereo matcher.

The evaluation was carried out on three different datasets: Middlebury dataset, EISATS dataset and KITTI dataset. The Middlebury dataset consists of small sets of images recorded in a controlled indoor environment. EISATS offers, for example, long synthetic sequences with ground truth or trinocular camera sequences where the third camera acts as ground truth (known as third-eye approach). KITTI data are recorded in real-world road traffic environment using a laser scanner for collecting ground truth.

Experiments have shown that matchers which have excellent performance in ideal conditions may fail in outdoor environments. In general, there is no “all-time best stereo matcher”. For example, using the third-eye approach, a belief-propagation matcher (with a linear smoothness function) provided often better matching results on some of the outdoor test sequences (e.g. images with sun glare) than a semi-global matcher, who was ranked high for other data.

Parts of this chapter have been published in IEEE Intelligent Vehicles Symposium 2013. Paper [?] was a joint work with Veronica Suaste, Diego Caudillo, and Reinhard Klette.

2.1 Introduction

The set-up for experiments in this study is a dynamic scene and a pair of cameras looking at the same scene in a canonical stereo configuration. The images are pre-rectified to be in this configuration, defined by parallel optical axes and row-aligned
2. Used Stereo Matchers and Test Data

Figure 2.1: Left: reference image. Middle: match image. Right: overlapping reference and match image. Disparity is the difference in pixel locations in reference and match images [75].

coplanar image planes. The left image is considered as the *reference image* and the right image as the *match image*.

As mentioned in the previous chapter, due to canonical configuration, the scanlines lie on the epipolar lines. So, the search region for locating the corresponding pixel between the reference and match images is limited to a scanline. The difference in corresponding pixel locations is denoted as *disparity*. For example Fig. 2.1 illustrates an example of disparity [75].

Stereo matching is a non-zero energy minimization problem. The defined *energy* (or *error*) is a combination of a *data term* and a *smoothness term*. The data term is the local matching cost determined as a similarity measure that can be pixel-based or window-based in a neighbourhood of tested pixels. The smoothness term uses a prior model to enforce the most likely case of disparity changes.

Stereo matching is a difficult problem. For example, there could always be intensity variations at corresponding pixels due to image blur, lighting variations, or occlusions (i.e. a scene point projected onto a pixel in the reference image may be occluded in the other view). To have a good estimate of the scene, some *matching constraints* [137] are necessary such as *similarity constraint* (i.e. identical intensities at corresponding pixels) [57], *smoothness constraint* (i.e. adjacent disparities vary smoothly) [57], a monotonic *ordering constraint* (i.e. the order of corresponding pixels does not change along one epipolar line), or a *uniqueness constraint* (i.e. for any pixel in the reference image, a corresponding pixel can be assigned in the match image) [57].

Stereo matching algorithms are generally classified into local, semi-global or global matching approaches [110, 118].

Local approaches have the advantage of lower processing time whereas the results relatively have more error [63, 135]. *Correlation methods* are examples of local
approaches which only consider sub-images, possibly at different levels of an image pyramid. Usually, these approaches are useful for real-time applications [64, 70]. Correlation based algorithms are simpler to implement but their accuracy is often the poorest. These methods locate the corresponding pixel by matching a correlation window in the stereo pair. They often also assume that pixels within the correlation window have constant disparities. However, if an object boundary lies in the window, this assumption becomes incorrect. Although there are techniques to reduce this effect [63, 110], it cannot be completely alleviated.

On the other hand, pixel-based stereo matching avoids this problem [23]. One such example is a dynamic programming (DP) stereo matching algorithm, which optimizes over an epipolar line. DP typically assumes ordering and uniqueness constraints and handles the matching process as a path finding problem [91, 20] enforced a piecewise-smoothness constraint and a simple relationship between occlusion and disparity for stereo matching by DP. Since DP works along one scanline only, the output usually suffers from a streaking effect.

DP algorithms along multiple scanlines (see [64, 110]) includes some semi-global techniques that reduce the negative effects of local methods. The semi-global matching (SGM) stereo approach [65], which, as the name suggests, is a local optimization approach that extends over large regions but only in a few directions. The accuracy is comparable to the accuracy of global methods such as belief propagation (BP) stereo [120] and graph cuts (GC) stereo [78, 27]. Thus, it has been used in a real-time environment like driver assistance systems [50].

BP stereo and GC stereo [78, 27] are examples of potentially global approaches which may consider the complete image (e.g. if iterations in belief propagation run ‘long enough’). Since these global approaches use the matching constraints in both dimensions of images, they are considered less prone to the streaking effect.

Global approaches are reported to have potentially better accuracy but are also computationally more expensive. However, there have been various efficient implementations of BP [44, 106, 107].

The next section introduces stereo matching algorithms, namely DP, SGM, and BP and some of their variants, namely symmetric dynamic programming stereo (SDPS) [51], iterative SGM stereo (iSGM) [62], and linear BP stereo (linBP) [6]. A comparison between such variants is provided and discussed in Section 2.5.

2.2 Stereo Matching Algorithms

For a pair $I_L$ and $I_R$ of rectified stereo images of size $[w \times h]$ pixels, let
\[
\Omega = \left\{ (u, v) : \left[-\frac{w-1}{2}\right] \leq u \leq \left[\frac{w-1}{2}\right] \land \left[-\frac{h-1}{2}\right] \leq v \leq \left[\frac{h-1}{2}\right] \right\}
\]
be the set of all pixels on an image. A disparity $d \geq 0$ is the offset in an epipolar line $v$, comparing $I_L(u, v)$ with $I_R(u - d, v)$, for starting at $p = (u, v)$ in the left image. Let

$$L = \{ \Delta : \Delta \in \mathbb{Z} \land 0 \leq \Delta \leq d_{\text{max}} \}$$

be the set of disparity labels, with $d_{\text{max}}$ being the maximum disparity between base and match images, with $0 < d_{\text{max}} \leq w$.

A dense disparity field defines a labelling $f$ on $\Omega$, with $f : \Omega \to L$. Function $f$ is called a disparity function, or just labelling for short in the following.

### 2.2.1 Dynamic Programming Stereo

Dynamic programming is a general design strategy for algorithms, applied here for the particular task of stereo matching.

#### Assumptions

DP assumes the monotonic ordering constraint and uniqueness constraint (unless there is an occlusion). In conclusion, both constraints lead to the condition

$$\max \{0, f_u - 1\} \leq f_u \leq \min \{u - 1, d_{\text{max}}\}$$

for the calculated labelling $f$. See, for example, the notes on DP stereo on [77].

#### Approach

For matching corresponding pixels, a data cost function is used. A very simple example is a direct value comparison

$$D_u(\Delta) = |I_L(u, v) - I_R(u - \Delta, v)| \quad \text{for} \quad -\frac{w}{2} \leq u \leq \frac{w}{2} \text{ and } \Delta \in L$$

which would not lead to accurate matching results and is here only given for illustrating the basic idea; window-based mean-normalised sums of absolute differences, or window-based comparisons of census transform results are more robust with respect to intensity variations or local noise [59].

Assume a disparity function $f$, with $f_u = f(u)$ being a disparity for $(u, v)$ in $I_L$. Let $s \leq w$ denote the stage of matching along a scanline. At stage $s$, we already have the assignment of disparities $f_u$ at stages up to $s - 1$. We are matching in row $v$. At $s \leq w$ we would have the partial energy

$$E_s(f) = \sum_{1 \leq u \leq s} D_u(f_u)$$

---

1 The floor function $\lfloor \cdot \rfloor$ is used for the given fractions; note that the principal point is at $(u, v) = (0, 0)$ if $w$ and $h$ are odd.
2.2. Stereo Matching Algorithms

if only considering the data cost (and not a relation between adjacent pixels as, e.g., expressed in Equ. (2.1)). The recursion for minimizing the total energy along the scanline is then (i.e. in the case without additional relations) defined by

\[ E_s(f) = \min_{0 \leq \Delta \leq d_{\max}} \{ D_u(\Delta) + E_{s-1}(f) \} \]

(2.4)

where we stop at \( s = w \). More complex recursions (also taking labels at adjacent pixels into account) are considered in Sections 2.2.2 and 2.2.3. DP is globally minimizing along a scanline, but not taking assigned labels in adjacent rows into account.

Symmetric Dynamic Programming Stereo

A variant of DP is SDPS \[54\]. One of its main features is that it constructs a cyclopean disparity image, the one seen by a virtual camera on the centre of the baseline between the optical centres of stereo cameras. There are three possible system states for visibility: binocular (B), monocularly left (ML), or monocularly right (MR), i.e. \( S \in \{B, ML, MR\} \).

For each disparity \( d \) of a labelling \( f \), the best predecessor is stored in a so-called predecessor array \( \xi_{u,d,S} \)[54]. Let, \( dI(u,d) \) be the intensity dissimilarity between neighbouring pixels \( u - \frac{d}{2} \) and \( u + \frac{d}{2} \) on the scanline. For each pixel \( u \) on the scanline, SDPS considers all the visible states for each disparity \( d \) by accumulating the costs \( C_{u,d,S} \). The cost to reach the B visible state is:

\[ C_{u,d,B} = dI(u,d) + \min (C_{u-1,d-1,ML}, C_{u-2,d,B}, C_{u-2,d,MR}) \]  

(2.5)

The costs to reach the ML visible state is:

\[ C_{u,d,ML} = oc + \min (C_{u-1,d-1,ML}, C_{u-2,d,B}, C_{u-2,d,MR}) \]  

(2.6)

where \( oc \) is the occlusion cost that acts as a smoothness constraint.

The costs to reach the MR visible state is:

\[ C_{u,d,MR} = oc + \min (C_{u-1,d+1,MR}, C_{u-1,d-1,B}) \]  

(2.7)

For each of the above cases, the predecessor array \( \xi_{u,d,V} \) stores the cost. For B visible point \( \xi_{u,d,B} \) is:

\[ \xi_{u,d,B} = \arg \min (C_{u-1,d-1,ML}, C_{u-2,d,B}, C_{u-2,d,MR}) \]  

(2.8)

For ML visible state \( \xi_{u,d,ML} \) is:

\[ \xi_{u,d,ML} = \arg \min (C_{u-1,d-1,ML}, C_{u-2,d,B}, C_{u-2,d,MR}) \]  

(2.9)

For MR visible state \( \xi_{u,d,MR} \) is:

\[ \xi_{u,d,MR} = \arg \min (C_{u-1,d+1,MR}, C_{u-1,d+1,B}) \]  

(2.10)
2.2.2 Semi-Global Matching

The SGM method generalizes single scanline DP into a multiple scanlines method [103]. Various modifications have been published with respect to search strategy, cost functions or built-in smoothness constraint. As a significant change to the ‘standard’ SGM strategy, iterative SGM (iSGM) introduced iterations into the SGM scheme [62].

I briefly outline the basic SGM strategy first. The cost for pixel correspondences between $I_L$ and $I_R$ images is first derived for all possible disparities $0 \leq \Delta \leq d_{\text{max}}$. Using dynamic programming, the correspondence cost is computed along a set of (usually two to eight) scanlines in different directions. This leads to the computation of final disparities based on a the-winner-takes-all evaluation, defining a dense labelling function $f$ for all disparities.

For a partial energy function $E_s(f)$ at stage $s$ (now defined along one of the multiple scanlines), disparities need to be assigned at the previous stages first for continuing further. This is defined recursively as follows:

$$E_s(f) = D_s(\Delta) + M_s - \min_{0 \leq \Delta \leq d_{\text{max}}} E_{s-1}(f)$$  \hspace{1cm} (2.11)

where $D_s(\Delta)$ is the matching data cost between pixels in $I_L$ and $I_R$ defined by off-set $\Delta$. The data cost, can be found by using, for example, the census cost function on a $3 \times 9$ window as specified for iSGM [62]. Term $M_s$ is defined as follows:

$$M_s = \min \left\{ \begin{array}{l}
E_{s-1}(g) \\
E_{s-1}(g-1) + P_a \\
E_{s-1}(g-1) + P_a \\
\min_{0 \leq \Delta \leq d_{\text{max}}} E_{s-1}(g) + P_b
\end{array} \right\}$$  \hspace{1cm} (2.12)

where $P_a$ and $P_b$ are regularization penalties for enforcing disparity consistency along a scanline. The smaller value $P_a$ is for minor differences (here: just difference 1), and $P_b$ typically a larger penalty (but adaptive to intensity values) for larger differences.

This is basically a ‘two-step’ Potts model. Input parameters are penalties $P_a$ and $P_b'$, and adaptation of $P_b$ is derived by

$$P_b(u,v) = \max \left\{ \frac{P_b'}{|I_L(u-1,v) - I_L(u,v)|}, P_a + \kappa \right\}$$  \hspace{1cm} (2.13)

where $\kappa > 0$. As $P_b$ is derived from image data differences, and assuming that depth discontinuities ‘usually’ occur at intensity discontinuities, $P_b$ aims at improving the matching performance at such places. For further details, in particular for a description of the iSGM program used in my experiments, see [62]. The Potts model is also used to assign penalties for BP stereo (described later in Section 2.3).
2.2. Stereo Matching Algorithms

2.2.3 Belief Propagation

A relatively efficient version of BP stereo matching was published in [44]. Disparities between 0 and \(d_{\text{max}}\) are again the labels, and those labels are iteratively calculated by updating a function \(f\) defined on an array of size \(w \times h \times d_{\text{max}} + 1\). Each pixel has an assigned belief value after ending one iteration, defined for all the possible disparities. These belief values influence the decisions for the adjacent pixels in the next iteration. The spatial aspect of those iterative influences of belief values is illustrated in Figure 2.2. The figure illustrates the propagation of belief values assuming 4-adjacency for the underlying BP network.

![Belief Propagation Diagram](image)

Figure 2.2: Belief propagation: \(p\) influences pixels \(q\) in one iteration step, and thus (indirectly) also pixels \(r\) in the next iteration step [77].

Figure 2.3 illustrates the general BP message passing scheme between pixels in a 4-neighbourhood. Pixel \(q\) receives a message at time \(t\) from pixel \(p\). This message is also based on neighbouring pixels of \(p\) at time \(t - 1\), but excluding \(q\).

As before for DP or SGM, labels are assigned by energy (or error) minimization for the so-far generated labelling function \(f\). The energy is also again defined by data cost \(D_{\text{data}}(f)\) at pixels \(p\) in \(I_L\), and smoothness (or discontinuity) costs \(C_{p,q}(f)\) for adjacent pixels \(p\) and \(q\), aiming at spatial consistency of the generated labelling \(f\).

See Fig. 2.3 for an illustration of the following. Let \(m^t_{p \rightarrow q}\) be the message from

![Message Passing Diagram](image)

Figure 2.3: Belief propagation: \(pq\) message passing at time \(t\).
pixel \( p \) to its adjacent pixel \( q \) at iteration \( t \):

\[
m^t_{p \rightarrow q}(f_q) = \min_{0 \leq h \leq d_{\text{max}}} \{C(h, f_q) + H_{p, q}(h)\}
\] (2.14)

where

\[
H_{p, q}(h) = D_p(h) + \sum_{r \in A(p) \setminus q} m_{r \rightarrow p}^{t-1}(h)
\] (2.15)

and \( A(p) \setminus \{q\} \) denotes adjacent pixels of \( p \) excluding \( q \). Figure 2.4 illustrates the four directions of message passing.

![Figure 2.4: Belief propagation: Four directions of message passing in a 4-neighbourhood. Forward (f), backward (b), upward (u), and downward (d).](image)

An optimum labelling \( f \) would minimize

\[
\sum_{p \in \Omega_L} \left( D(f_p) + \sum_{(p,q) \in A} C(f_p, f_q) \right)
\] (2.16)

where \( A \) is the chosen adjacency relation (such as 4-adjacency). If BP would run such that every pixel communicates with every other pixel, the global minimum could be achieved. However, there are time limits, and practically one can only use a limited number of iterations, possibly combined with a hierarchical or alternating strategy to ensure better efficiency [44].
For example, [196] describes a hypertree for propagating messages in forward and backward directions for each scanline. The hypertree communicated with the neighbouring hypertrees with upward and downward message propagation. This way, BP was implemented for a very-large scale integration (VLSI) architecture using absolute difference as a data cost function. Although VLSI is very fast, still it has memory limitations. Due to this limitation the BP algorithm was only tested on low-resolution dataset for a maximum of 16 disparity labels.

To test their message passing scheme on bigger datasets, I re-implemented their message passing scheme on a general purpose (sequential) computer where there were no memory limitations. Due to which I was able to validate BP on various datasets using various data cost functions.

2.3 Continuity Cost Functions

In the Potts model, the cost is uniformly penalized as

\[
C(g_p, g_q) = \begin{cases} 
0 & \text{if } g_p = g_q \\
\ c & \text{otherwise}
\end{cases}
\]  

(2.17)

where \(c > 0\). The linear continuity cost function is defined based on steepness factor \(b > 0\):

\[
C(g_p, g_q) = b \cdot |g_p - g_q |
\]  

(2.18)

The truncated linear continuity cost function is defined based on truncation of the maximum cost at \(c > 0\):

\[
C(g_p, g_q) = \min \{b \cdot |g_p - g_q |, c\}
\]  

(2.19)

The truncated quadratic continuity cost function is defined as

\[
C(g_p, g_q) = \min \left\{ b \cdot (g_p - g_q)^2 , c \right\}
\]  

(2.20)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations (t)</td>
<td>24</td>
</tr>
<tr>
<td>Pyramid levels</td>
<td>1</td>
</tr>
<tr>
<td>Cost function window (width $\times$ height)</td>
<td>$5 \times 5$</td>
</tr>
<tr>
<td>Scaling factor (b) of discontinuity function</td>
<td>1.0</td>
</tr>
<tr>
<td>Truncation value (c)</td>
<td>(d_{max}/8)</td>
</tr>
</tbody>
</table>

Table 2.1: Configuration parameters for my belief propagation program. No hierarchical processing, and use of the census cost function.
For the continuity or smoothness term $C$, I only used a truncated linear function (defining a program called \textit{linBP}). For truncation, I used $c = d_{\text{max}} / 8$. Table 2.1 summarizes the BP parameters for program \textit{linBP}. I submitted my program to the KITTI benchmark suite [74], and it was at rank 13 at the time when submitted in January 2013.

2.4 Data Cost Functions for Stereo Matching

Stereo methods have been compared in several surveys [28][110][59], on the Middlebury site [111], and on the KITTI site [74]. The used matching strategy and the used cost functions are significant defining components of stereo matchers. In this section I briefly recall data cost functions.

For driver assistance systems, camera gain and exposure differences may often lead to radiometric variations, known as “lighting artefacts”, or “brightness differences” at corresponding pixels. There are still only a few studies about stereo matching behaviour in non-ideal environments also defined by varying lighting in recorded images, moving wipers, reflections on wet roads, and other challenges; see, for example, [60][66][76][113].

A matching cost (i.e. the data cost term $D$) is usually computed for all possible disparities at each pixel. Common matching cost measures are, for example, \textit{absolute differences} (AD), \textit{gradient} [60], or \textit{sampling insensitive absolute differences} [22][23], and by considering larger neighbourhoods also measures such as the \textit{zero-mean sum of absolute differences}, \textit{zero-mean sum of squared differences}, \textit{normalized cross correlation}, or the \textit{census transform} [136]. I briefly discuss such measures in this chapter.

2.4.1 Pixel-based Cost Functions

The \textit{absolute difference} $D_{\text{AD}}$ is a very simple data cost function, just mentioned for illustrating an extreme. It is defined by the difference between reference image pixel intensity $I_L(u, v)$ and the corresponding pixel intensity in the match image $I_R(u - \Delta, v)$ for disparity $\Delta$:

$$C_{\text{AD}}(u, v, \Delta) = |I_L(u, v) - I_R(u - \Delta, v)|$$  \hspace{1cm} (2.21)

It does not work well under varying signals whether disparity is integral or not, due to other noise sources. This is referred to as \textit{the problem of sampling} [23].

One of the possible solutions to this problem is to work at sub-pixel resolution [19][86]. Obviously, using more complex measures than $D_{\text{AD}}$ increases the computation time (e.g., [23] report an increase by 1100%).
Another alternate is to use the *linearly interpolated intensity functions* to estimate a sampling insensitive absolute differences cost function $D_{BT}$, known as Birchfield-Tomasi data cost. $D_{BT}$ is computed by linear interpolation over neighbouring pixels [22]. This increases computation time by only 10% compared to the use of $D_{AD}$; see [23].

For computing $C_{BT}$, not only intensities at pixel locations $(u, v)$ and $(u - \Delta, v)$ are used, but also interpolated intensities based on the intensities at $(u - 1, v)$, $(u, v)$, and $(u + 1, v)$ in $I_L$. Similarly, the interpolated intensities based on the $I_R$ intensities at $(u - \Delta - 1, v)$, $(u - \Delta, v)$, and $(u - \Delta + 1, v)$ are also used. The data cost $D_{BT}$ is defined as follows:

$$D_{BT}(u, v, \Delta) = \min\{\max\{a_1, a_2, 0\}, \max\{b_1, b_2, 0\}\}$$ (2.22)

with

$$a_1 = I_R(u - \Delta, v) - \max\{\frac{I_L(u, v) + I_L(u - 1, v)}{2}, \frac{I_L(u, v) + I_L(u + 1, v)}{2}\}$$

$$a_2 = \min\{\frac{I_L(u, v) + I_L(u - 1, v)}{2}, \frac{I_L(u, v) + I_L(u + 1, v)}{2}\} - I_R(u - \Delta, v)$$

$$b_1 = I_L(u, v) - \max\{\frac{I_R(u - \Delta, v) + I_R(u - \Delta - 1, v)}{2}, \frac{I_R(u - \Delta, v) + I_R(u - \Delta + 1, v)}{2}\}$$

$$b_2 = \min\{\frac{I_R(u - \Delta, v) + I_R(u - \Delta - 1, v)}{2}, \frac{I_R(u - \Delta, v) + I_R(u - \Delta + 1, v)}{2}\} - I_L(u, v)$$

### 2.4.2 Window-Based Cost Function

Another alternative for computing a data cost $D_{census}$ is using the census transform; see, for example, [68][69]. The census transform uses intensity differences within a window around the pixel $(u, v)$. It maps pixel intensities in the window into a bit string $\xi(u, v)$ as follows:

$$\xi(u, v) = \bigwedge_{i=-ww}^{ww} \bigwedge_{j=-hh}^{hh} \rho(I(u, v), I(u+i,v+j))$$ (2.23)

where $\bigwedge$ denotes bit concatenation. The auxiliary function $\rho$ denotes the difference in intensities between neighbouring pixels of an image $I$:

$$\rho(p_1, p_2) = \begin{cases} 1 & \text{if } p_1 > p_2 \\ 0 & \text{if } p_1 \leq p_2 \end{cases}$$ (2.24)

$D_{census}$ is then the Hamming distance between the bit string $\xi_L(u, v)$ in $I_L$ and the bit string $\xi_R(u + \Delta, v)$ in $I_R$ defined by off-set $\Delta$. 
2. Used Stereo Matchers and Test Data

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Experiments</th>
<th>Stereo matchers</th>
<th>Data cost functions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SDPS</td>
<td>iSGM</td>
</tr>
<tr>
<td>Middlebury</td>
<td>Exp. 1</td>
<td>y</td>
<td>n</td>
</tr>
<tr>
<td>EISATS</td>
<td>Exp. 1 (set 2)</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td></td>
<td>Exp. 2 (set 2)</td>
<td>n</td>
<td>y</td>
</tr>
<tr>
<td></td>
<td>Exp. 3 (set 9)</td>
<td>n</td>
<td>y</td>
</tr>
<tr>
<td>KITTI</td>
<td>Exp. 1</td>
<td>n</td>
<td>y</td>
</tr>
</tbody>
</table>

Table 2.2: Summary of comparisons between stereo matchers on datasets. Entities used are represented by ‘y’, while the ones unused are represented by ‘n’.

2.5 Results of Comparison and Discussion

Assuming that a “reasonable” ground truth in disparity values (i.e. measured or calculated values being assumed to be close to the true disparity values) is available, the matching accuracy is determined as the percentage of all pixels (%) in the set \(\Omega\) of all \(w \times h\) pixels of the image whose ground truth disparity is greater than zero and the difference between the assigned disparity from the ground truth is below or equal to a threshold \(\tau\), such as 2 or 3.

I evaluated the matching accuracy of SDPS, linBP, and iSGM on a set of images with ground truth. Examples of available data sets with ground truth are those published at Middlebury College [110, 112], Set 2 of EISATS [42], and KITTI datasets [51, 74].

The summary of my experiments is highlighted in Table 2.2. For the Middlebury College dataset, SDPS was compared with linBP. iSGM is not taken into consideration because it is an extension of DP (and SDPS), and also because iSGM was designed to work on outdoor scenes. The comparison was carried out using only absolute difference \(D_{AD}\) as a data cost function. For EISATS datasets, I determined the accuracy of data cost functions on linBP alone. Later, the best cost function was chosen for a direct comparison between linBP and iSGM on EISATS (Set 2) and KITTI datasets where disparity ground truth is available.

I also evaluated the stereo matchers linBP and iSGM on real-world data from Set 9 of EISATS [42]. As disparity ground truth is unavailable for this data, so instead the third-eye approach proposed in [94] was used. For this evaluation method, it is required to have at least trinocular data. Data recorded with a third camera then take over the role of being the ground truth: calculated disparities are used to compare a virtual view obtained for the pose of the third camera with the actually recorded images.
2.5. Results of Comparison and Discussion

Figure 2.5: Reference images for used stereo pairs from the Middlebury College datasets. Top left: Tsukuba. Top right: Venus. Bottom left: Teddy. Bottom right: Cones.

2.5.1 Experiments With the Middlebury Dataset

This dataset consists of sets of images captured in controlled indoor environments. I considered four image sets from the Middlebury College datasets, namely Tsukuba, Venus, Teddy, and Cones with different $d_{\text{max}}$ values, being 15, 19, 59, or 59 pixels, respectively. Figure 2.5 shows reference images for those four datasets.

Since SDPS operates on a single scanline, it suffers from the streaking affect. In comparison, BP operates on neighborhoods growing in size depending on the number of iterations.

Using the simple data cost function $D_{AD}$, Fig. 2.6 shows the output of SDPS (middle column, with occlusion cost 18) and linBP (right column). Note that disparities of SDPS are generated after converting the cyclopean image to $I_L$ [70].

The SDPS streaking effect is visible particularly at sharp edges with sudden disparity changes along a scanline. For relatively smooth disparity changes such as in test image pair Venus, the SDPS output is better. Whereas for Teddy and Cones stereo pairs where there is a sudden disparity change with respect to the background, the streaking effect is more noticeable. Table 2.3 shows a direct comparison of matching accuracy for SDPS and linBP. The table shows the percentage of error pixels with an error greater than $\tau_t$.

Apparently, linBP outperforms SDPS on the considered data. However, this comes at the cost of more computation time. For example, the Venus image is of resolution $[w \times h] = [434 \times 383]$ pixels, and the stereo matcher has to compute up to $d_{\text{max}} = 59$ disparities on a single core of Intel i5-2500 CPU running at 3.3 GHz. linBP takes 915
Figure 2.6: Columns, left to right: Ground truth disparity maps, SDPS disparity maps, linBP disparity maps. Rows, top to bottom: Tsukuba, Venus, Teddy, and Cones. Note that all disparities are scaled at the same level as the ground truth images.

milliseconds to compute the disparity for a whole image, whereas SDPS only takes 273 milliseconds. Since the objective here is to illustrate the accuracy, linBP will be used in this thesis for future comparisons.
2.5. Results of Comparison and Discussion

2.5.2 Experiments With the EISATS Dataset

Set 2 of the EISATS benchmark datasets was made available by Daimler A.G. and the .enpeda. group [10] on an Auckland-based website [126]. It provides synthetic stereo sequences with accurate ground truth. For my experiments I used Set 2 from the EISATS data.

Since this is a synthetic sequence, the capturing environment is ideal, without any lighting variations between images in a stereo pair. The images have resolution \([w \times h] = [640 \times 480]\) pixels, with \(d_{\text{max}} = 58\) pixels. For an example of Sequence 2 of

<table>
<thead>
<tr>
<th></th>
<th>(\tau_t = 2)</th>
<th>(\tau_t = 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SDPS</td>
<td>linBP</td>
</tr>
<tr>
<td>Tsukuba</td>
<td>5.0</td>
<td>4.7</td>
</tr>
<tr>
<td>Venus</td>
<td>8.6</td>
<td>5.7</td>
</tr>
<tr>
<td>Teddy</td>
<td>21.6</td>
<td>16.6</td>
</tr>
<tr>
<td>Cones</td>
<td>23.6</td>
<td>18.8</td>
</tr>
</tbody>
</table>

Table 2.3: Stereo matching accuracy (error rate) of SDPS and linBP for the Middlebury dataset.

Figure 2.7: Illustration of EISATS Set 2, Sequence 2. Left to right, top row: Reference image and match image. Left to right, bottow row: Ground truth disparity map and linBP disparity map.
2. Used Stereo Matchers and Test Data

2.5.3 Experiments With the KITTI Dataset

These real-world sequences come with ground truth provided by a laser range-finder \cite{74, 51}. Shown scenes have high complexity and there are other challenges such as lighting variations among stereo pairs. The training dataset consists of 194 images, where ground truth ($d_{\text{max}} = 255$) is provided with or without occlusion. For an example of the KITTI dataset, see Fig. 2.9. It shows colour-coded disparities for linBP and the ground truth disparity map.

Data cost functions for linBP

I repeat the experiments for linBP as specified before for the synthetic sequence. I show results for the first 20 KITTI training stereo frames, compared with no-occlusion ground truth data and $\tau_t = 2$, as presented in Fig. 2.10. This again shows that the
2.5. Results of Comparison and Discussion

Figure 2.9: KITTI training data example. Top: Reference image. Middle: Ground truth. Bottom: linBP disparity map.

census data cost function provides the best matching accuracy compared to other two data cost functions, especially its robustness is worth to be noted with respect to brightness differences.

**linBP vs. iSGM**

For comparing linBP with iSGM, the census data cost function is used on the complete training dataset with non-occluded ground truth disparities.

Figure 2.11 shows this comparison between linBP and iSGM for $\tau_t = 2$. For this data set and $\tau_t = 2$, iSGM outperforms linBP “slightly” with respect to the mean error. Table 2.4 shows the mean error and standard deviation. However, this order changes for $\tau_t = 3$. Errors are computed using the KITTI stereo development kit.
To further analyse the matcher’s behaviour over the whole sequence, I computed the winning count where a matcher having the smaller error wins for that image. Surprisingly, for \( \tau_t = 2 \), the winning count is 99 for linBP, and 95 for iSGM. These numbers show that linBP is winning more often, even though its mean error is higher than that of iSGM. Because linBP also has a higher standard deviation than iSGM, this suggests that the iSGM error is more often closer to its mean error, whereas the linBP error is more often below the mean, but to nullify that, for a few images the error is much higher than the mean. This explanation makes sense to me, as occasionally, BP message passing fails to propagate in certain regions (e.g. across ‘strong’ intensity edges) leading to more errors in disparity matching.
Out-Noc (%) | Out-All (%)  
---|---|---|---
2 pixel | iSGM | linBP | iSGM | linBP |
8.04 | 11.78 | 10.09 | 13.87 |
3 pixel | 5.16 | 8.66 | 7.19 | 10.81 |
4 pixel | 3.87 | 7.06 | 5.84 | 9.22 |
5 pixel | 3.15 | 6.02 | 5.03 | 8.18 |

Table 2.5: Evaluation on KITTI test images for linBP and iSGM.

The winning count for $\tau_t = 3$ further develops to 124 for linBP and 70 for iSGM. Table 2.5 shows the comparison between linBP and iSGM on the KITTI test dataset. Out-Noc is % of erroneous pixels in non-occluded areas, and Out-All is % of erroneous pixels in total. Avg-Noc is the ratio “average disparity / end-point error” in non-occluded areas (1.2 for iSGM and 1.7 for linBP), and Avg-All is the ratio “average disparity / end-point error” in total (2.1 for iSGM and 2.7 for linBP).

From these findings it appears that iSGM outperforms linBP on these real-world data. Since the ground truth for the KITTI test dataset is unavailable to the end-users, it is not possible to compute the winning count on this dataset.

### 2.5.4 The Third-eye Approach

For outdoor environments, a winning stereo matcher such as iSGM still faces challenges that are already summarized in the ECCV 2012 Robust Vision Challenge report. The KITTI data do not come with challenges such as sun flare, rain, wipers, or low light. I use the third-eye approach defined in [94] for evaluating the quality of stereo matchers on real-world video data for the common case when disparity ground truth is actually not available. This requires trinocular recording (at least). The ECCV 2012 Robust Vision Challenge did not provide trinocular stereo data. However, there are trinocular test sequences available, for example in Set 9 from EISATS [42].

Recording of those sequences was performed with three calibrated and time-synchronized cameras in a canonical stereo configuration. Two cameras provide the $I_L$ and $I_R$ images for stereo analysis, respectively, and the third camera provides image $I_T$ which is used for evaluation (and could also be used for trinocular stereo analysis).

Calculated disparities and calibration data allow us to map the $I_L$ into a virtual image $I_V$ at the pose of the third camera. Due to the geometric transform and due to occlusions, some pixel values in the virtual image $I_V$ remain undefined. As there might
be brightness differences between images $I_L$ and $I_T$, therefore I used the *normalised cross correlation* (NCC) for comparing defined image values in $I_V$ with those at the same pixel location in $I_T$, at any time $t$ (only specified in the following formula for the set $\Omega_t$ of pixel locations $(u,v)$ with defined values in $I_V$) of the recorded sequence:

$$S_{NCC}(t) = \frac{1}{|\Omega_t|} \sum_{(u,v) \in \Omega_t} \frac{[I_T(u,v) - \mu_T][I_V(u,v) - \mu_V]}{\sigma_T \sigma_V}$$

Symbols $\mu$ and $\sigma$ denote mean and standard deviation of the corresponding images, and $|\Omega_t|$ is the cardinality of this set. Fig. 2.12 illustrates the main steps involved for the evaluation of stereo matchers by the third-eye approach [94].

This *similarity measure* $S_{NCC}$ equals 1 in case of absolute identity, and decreases in magnitude with the ratio of differences between $V$ and $T$ at positions in $\Omega_t$. For reducing the effect of mismatches within homogeneously textured regions in $L$ (or $T$), it is also suggested in [61] to use a modified measure, where $\Omega_t$ only contains pixel locations which at a distance of 10 or less to an edge pixel in $L$. This defines similarity measure $S_{NCCmask}$. 

Figure 2.12: The third-eye approach uses three cameras. Left most one forms the third image, while right two are for computing disparity map by stereo matching [94].
2.5. Results of Comparison and Discussion

Set 9 of EISATS

In this experiment I tested iSGM and linBP on Set 9 of EISATS \[22\]. This set consists of eight trinocular sequences, each containing 400 stereo frames, grey-level images with resolution \([w \times h] = [640 \times 480]\) pixels, recorded at 10 bit per pixel \([61\) \[113\]. The test sequences are: ‘Barriers’ (crossing a bridge with road blocks on one side, very close to the ego-vehicle), ‘Dusk’ (a sun flare sequence), ‘Harbour bridge’ (same bridge but without road blocks), ‘Midday’ (hilly suburban road), ‘Night’ (with dense traffic), ‘People’ (pedestrians crossing in front of the ego-vehicle), ‘Queen street’ (driving along a CBD street), and ‘Wiper’ (same scene as in ‘Midday’, but now with running wipers).

Similarity Measures

I analysed the information provided by similarity measures \(S_{NCC}\) and \(S_{NCCmask}\), and also a simple sum-of-squared-differences (SSD) comparison between virtual and third images. The SSD measure appears to be of not much significance for understanding the quality of stereo matchers when using this third-eye approach. When comparing \(S_{NCC}\) with \(S_{NCCmask}\), it appears that confidence results obtained with both measures

Figure 2.13: Functions \(S_{NCC}\) and \(S_{NCCmask}\) for iSGM on sequence Midday, and for linBP on sequence Dusk.
2. Used Stereo Matchers and Test Data

Figure 2.14: Comparison of the performance of iSGM and linBP on four of the sequences from Set 9. iSGM performs better on Bridge, but linBP wins on Dusk, Midday, and Wiper.

are ‘well correlated’ with visually evaluated accuracy of stereo matching.

Measure $S_{\text{NCC}}$ provides a more detailed picture, with more ‘valleys’ in its curve indicating difficulties (where designers of a stereo matcher need to check for reasons and possible solutions). Figure 2.13 shows both $S_{\text{NCC}}$ and $S_{\text{NCC_mask}}$ for the sequence Midday when using iSGM, and for sequence Dusk when using linBP. The analysis prompts that only measure $S_{\text{NCC_mask}}$ should be used only in subsequent works.

**NCC_mask Comparison of iSGM with linBP**

Figure 2.14 shows four graphs that illustrate the performance of iSGM and linBP on four sequences from Set 9 of EISATS; functions $S_{\text{NCC_mask}}$ are only shown for the two stereo matchers.

Visual evaluations of many of the stereo frames in those eight sequences from Set 9 show that iSGM is the better performing algorithm than linBP due to its appealing accuracy at occlusion edges. However, when analysis is done by using the $S_{\text{NCC_mask}}$ measure, surprisingly the opposite appears to be the dominating event (i.e. linBP appears to win more frequently).
2.5. Results of Comparison and Discussion

Figure 2.15: Comparison of NCC_mask of iSGM on set 9 of EISATS with NCC_mask-normalized functions sigma_left, NCC_leftright, and sigma_Sobel

2.5.5 Data Measures

The third-eye approach can be possibly avoided by correlating the quality of stereo matchers directly with data measures on stereo videos. For example, let sigma_left be the standard deviation of the $I_L$, NCC_leftright be the NCC between $I_L$ and $I_R$, and sigma_Sobel be the standard deviation of the Sobel edge values of $I_L$. How do such measures relate to the performance of stereo matchers on challenging real-world data?

For comparing functions, I unify their means and variances, taking the mean and variance of function $S_{NCC\text{mask}}$ as uniform goal for all the other functions; see [119]. In general, let $f$ and $g$ be real-valued functions on the same domain with non-zero
variances. Let $\mu_f$ and $\sigma_f$ be the mean and standard deviation of function $f$. Using

$$\alpha = \frac{\sigma_g}{\sigma_f} \mu_f - \mu_g \quad \text{and} \quad \beta = \frac{\sigma_f}{\sigma_g}$$

$$g_{\text{new}}(x) = \beta(g(x) + \alpha) \quad (2.25)$$

we obtain a function $g_{\text{new}}$ which has the same mean and variance as function $f$.

Figure 2.15 shows plots of the functions $S_{\text{NCCmask}}$ and $S_{\text{NCCmask,normalized}}$ used for iSGM on the sequences in Set 9 of EISATS together with normalized plots of those three data measures discussed before. I also calculate $L_1$-distances $d_1(f, g)$ between normalized functions $f$ and $g$ by taking $\sum |f(k) - g(k)|$ for all frames $k$ in the sequence and dividing by the number of frames.

The question now is, for example, whether a low or high value of $d_1(S_{\text{NCCmask}}, g)$, where $g$ is one of the three data measures, has a particular meaning for analysing stereo matchers. A low value indicates that the measure might replace the third-eye approach on this particular sequence.

Table 2.6 shows the values for $d_1$ which is the distance between the three data measure functions and the NCC-mask function for those five sequences where values were “reasonably” small.

Indeed, the plots for sequences Dusk, Midday, and Queen in Figure 2.15 also indicate a good correlation between $S_{\text{NCCmask}}$ and data measures. I consider this a starting point for doing further in-depth analysis of the reasons of degrading behaviour of stereo matchers.

### 2.6 Summary

In this chapter, I highlighted the main aspects of some common stereo matchers. Depending on the processing requirements and the environment complexity, the performance, and thus the choice of a stereo matcher may vary. Adaptive solutions appear to be the logical consequence; they have been proposed in \cite{76}. Although SDPS \cite{71, 95, 96, 97} has been proved to be working well in real-time, due to the streaking effects it is not reliable. For example, iSGM and linBP do not suffer from such streaking effects.

<table>
<thead>
<tr>
<th></th>
<th>Dusk</th>
<th>Midday</th>
<th>Queen</th>
<th>Wiper</th>
<th>People</th>
</tr>
</thead>
<tbody>
<tr>
<td>sigma_left</td>
<td>3.69</td>
<td>1.52</td>
<td>3.78</td>
<td>5.16</td>
<td>6.90</td>
</tr>
<tr>
<td>NCC_leftright</td>
<td>4.18</td>
<td>1.26</td>
<td>3.33</td>
<td>4.59</td>
<td>5.21</td>
</tr>
<tr>
<td>sigma_Sobel</td>
<td>5.79</td>
<td>2.30</td>
<td>3.29</td>
<td>5.42</td>
<td>5.80</td>
</tr>
</tbody>
</table>

Table 2.6: Distance values of normalised data measures to $S_{\text{NCCmask}}$. 
In the trinocular camera set-up, the results came partially as a surprise to me, although, there seems to be good explanation for them. BP is propagating belief sequentially row-wise with feedback from adjacent rows, without paying particular attention to vertical propagations or accuracy along horizontal disparity discontinuities, as iSGM does. Thus, iSGM outperforms BP at such disparity discontinuities.

The accuracy within object regions, away from disparity discontinuities, appears to be better with linBP. For example, in the sequence Bridge, there are many structural details and not such a high ratio of homogeneous regions and accordingly, iSGM is winning here. In the Wiper sequence, the NCC\textsubscript{leftright} data measure indicates correctly the actual input data situation, and BP is able to cope better with those data (see also [113]).

For testing, it would be useful to have trinocular data with ground truth for challenging situations. This would allow to correlate ground truth, similarity measures, and data measures more accurately. I assume that these are useful tools for identifying critical events for stereo matchers which may need to be resolved in future.

Hypertree based BP with absolute difference data term has already been implemented on hardware but for a limited range of disparities; for example, [106] uses 16 disparity levels using $D_{AD}$. Whereas, SGM has been implemented already on FPGA for a wider range of disparities using $D_{census}$, and is already working in real-time (at 25 Hz) in driver assistance systems of Daimler A.G. In conclusion, in Chapter 5 (where I am testing theory in the real world), I will be using both $D_{census}$ based linBP and $D_{census}$ based iSGM for the evaluation of my theoretical models, proposed in Chapters 3 and 4. I will use the census based data cost function for both stereo matchers.
Chapter 3

Avoiding Constant Velocity Objects

In a stereo system, the measured disparity values are integral, therefore the measured depths are discrete. This can create a trap for a safety system whose purpose is to estimate the trajectory of a moving object and to issue an early warning for the driver. The change in measurable depths becomes obvious for closer regions, however the extent of the stereo common field of view also reduces in these regions. In this chapter, I propose a velocity estimation algorithm that takes into account the constraints of stereo, while determining an accurate estimate of the trajectory of an object moving at constant velocity. I provide a model that enables the safety engineer to design a “safe” (by some criterion of his/her choice) system. The model shows where it works and where it does not. The safety engineer may adjust the system parameters (e.g. focal length, baseline length, frame rate etc) until it is safe.

Parts of this chapter have been published in IEEE Intelligent Vehicles Symposium 2012. Paper [5] was a joint work with John Morris.

3.1 Introduction

A stereo system designer works in a complex design space considering the optical characteristics of the system such as lenses, baseline, image resolution, etc, as well as vehicle handling parameters, namely speed, braking distance and turning ability. To aid a safety engineer to determine the efficacy of a stereo based safety system, I propose here a model that demonstrates existing limitations for analysing potential hazards. The model assumes that a hazard can “appear out of nowhere”, i.e. pedestrians walk out from behind parked cars, or vehicles appear from behind other vehicles, so an object may first be observed at any position.

Figure 3.1 shows an example of a collision scenario where a car is moving with constant velocity towards the path of the ego-vehicle which is moving with constant velocity.

The model aims for the safety engineer to experiment with the optical and other vehicle parameters until he/she finds the best set of parameters (i.e. optimise them)
3. Avoiding Constant Velocity Objects

Figure 3.1: Example of a collision scene observed by the reference camera. Left image: The car on the right moves towards the path of the ego-vehicle for a potential collision. Right image: After few frames, this car is about to cause collision. Illustrated sequence is in Set 2 of EISATS.

and then implement the system accordingly.

Effectiveness is measured by the highest speed that an opposing colliding object (e.g. other car, pedestrian) can have and still be safely detected. “Safely detected” means that the object’s trajectory is correctly predicted in time to brake. A static hazard in the path of the ego-vehicle is also taken into consideration. My model outputs a contour map showing the maximum safe speed for objects in the vicinity of the ego-vehicle. This single map gives an overall view of the ability of the stereo system to warn the driver in time to avoid or mitigate hazards and highlights the

Figure 3.2: Collision speed $V_{\text{crit}}$ varies with scenarios and the amount of protection needed for the road users [130].
3.2. The Model

limitations of the stereo component. I assume that low speed collisions in which no one is injured, whilst undesirable, are “safe” and define a maximum tolerable collision speed, $V_{\text{crit}}^i$, for the model. $V_{\text{crit}}^i$ varies with scenarios as does the amount of protection necessary for road users [130]. Fig. 3.2 further illustrates this.

Here, I assume that the only collision avoidance strategy is braking. Thus the deceleration generated by the brakes is a key parameter in this safety model.

3.2 The Model

The model consists of a series of constraints each of which can further reduce the speed of an opposing object that the system can handle. Clearly, the first constraint is “unavoidable collision” when an object is travelling so fast that the system cannot take any action to avoid a collision; this sets an upper limit to the speed for my model. Note that even if a collision is inevitable, a timely warning can mitigate damage as the driver may be able to brake or turn the vehicle to avoid the opposing object.

3.2.1 The Ego-vehicle

The model represents the ego-vehicle’s starting location as

$$O^i(0) = [O_{ix}^i(0), O_{iy}^i(0), O_{iz}^i(0)]^T$$

and assumes that it travels at constant velocity $\overrightarrow{V^i}$ along the $Z$ axis so that $O_{ix}^i = V_{ix}^i = 0$.

The model has an exclusion zone of radius $r_{\text{exc}}$ around the ego-vehicle. The model uses an exclusion zone that is slightly larger than a typical vehicle to allow for the psychological impact of a near miss on a driver’s confidence as if there are too many near misses the system will certainly not be trusted.

The model, as presented in this chapter, is restricted to vehicles and objects moving on a flat surface, so all $Y$-components are set to 0.

3.2.2 Colliding Object

Assume that a rigid object of size $L \times H \times W$ is travelling at constant velocity

$$\overrightarrow{V} = [V_x, V_y, V_z]^T$$

\footnote{I exclude turning because it is not always a safe avoidance strategy, e.g. when driving is constrained by highway lanes.}
in the same direction as its $L$-dimension. There are $n$ equidistant feature points over the object surface. The point closest to the ego-vehicle is considered to be the object reference point. So, the reference point first appears at

$$\vec{O}(0) = [O_x(0), O_y(0), O_z(0)]^T.$$  

### 3.2.3 Model Basics

The warnings of the model are depicted in Fig. 3.3. The object’s position at time $t$ is $\vec{O}(t) = \vec{O}(0) + \vec{V} t$. The object’s real trajectory angle is $\eta = \tan^{-1}\left(\frac{V_x}{V_y}\right)$ relative to the $X$-axis.

![Figure 3.3: Collision scenario.](image)
3.2. The Model

The object enters the ego-vehicle’s path (i.e. crosses the $X = r_{exc}$) at time $t_{cross} = -O_x(0) + r_{exc} - W\sin \eta$ \[ (3.1) \]

at position $O(t_{cross}) = [r_{exc} - W\cos \eta, 0, Z_{cross}]^T$, where $Z_{cross} = O_z(0) - V_z t_{cross}$.

The object leaves the vehicle’s path after

$t_{leave} = -O_x(0) - r_{exc} + L\cos \eta$ \[ (3.2) \]

at $Z_{leave} = O_z(0) - V_z t_{leave}$.

At $t_{cross}$, the ego-vehicle’s exclusion zone is centred on $O(t_{cross}) = O(0) + V^t t_{cross}$.

The ego-vehicle will have a collision if the separation between the object and the ego-vehicle

$D_c = \sqrt{(O_x(t) - O_x(t))^2 + (O_z(t) - O_z(t))^2} \leq r_{exc}$.

The timing of warning $t_{warn}$ is very crucial \[13\]; if the system can generate a warning before $t_{warn} = t_{cross} - t_b - t_d$ where $t_b$ is the braking time and $t_d$ is the driver response time when the ego-vehicle is at $[0, 0, Z_{cross} - D_b]^T$, then the ego-vehicle will slow down to less than $V_{crit}$ in time.

An example of the significance of $t_{warn}$ is also shown in Fig. 3.4, where an object on a collision course is avoided only after a timely warning (Case 2 compared to Case 1).

3.2.4 Design Constraints

The stereo system works in a relative frame centred on the ego-vehicle; a superscript $r$ denotes quantities in this frame of reference, so $O^r(t) = O(t) - O^r(t)$ and collision trajectory

$\zeta = \tan^{-1}\left(\frac{-O_x(t)}{-O_z(t)}\right)$

The stereo system and the ego-vehicle’s capabilities have certain constraints explained below.

Extent of the Stereo Common Field of View (CFoV)

I assume that the safety system has two cameras, each with $w \times h$ square pixels of size $\tau = \tau_u = \tau_v$ in a canonical stereo configuration with baseline $b$, focal length $f$, and vergence angle $\phi = 0$. 
3. Avoiding Constant Velocity Objects

The angular extent of the CFoV is $2\theta$, where

$$\theta = \tan^{-1}\left(\frac{w\tau}{2f}\right)$$

(3.3)

The maximum disparity that the stereo system can process, $d_{\text{max}}$, determines the closest distance at which depth can be measured: $Z_{\text{min}} = \frac{fb}{td_{\text{max}}}$. A reference point at $O(t)$ is observable in the stereo CFoV if $-\theta \leq \tan^{-1}\left(\frac{O_r(t) - O_r(t) - b/(2\tan \theta)}{O_r(t) - b/(2\tan \theta)}\right) \leq \theta$ and its distance may be measured if $O_r(t) \geq Z_{\text{min}}$. Similarly, for other feature points this constraint is also applicable.
3.2. The Model

Depth Resolution

The depth resolution \( \delta Z(d) \), is the smallest change in distance that can be measured, and it increases with distance. The depth resolution at \( d \) is the difference between the depth corresponding to sequential disparity values \( d \) and \( d + 1 \):

\[
\delta Z(d) = \frac{fb}{\tau} \left( \frac{1}{d} - \frac{1}{d+1} \right)
\]

and the uncertainty in depth for a feature point appearing to be at depth \( \hat{Z} \) is

\[
\Delta Z(d) = \frac{\delta Z(d) + \delta Z(d-1)}{2} = \frac{fb}{2\tau} \left( \frac{1}{d-1} - \frac{1}{d+1} \right). \tag{3.5}
\]

Clearly, a system with better depth resolution will be able to estimate velocity faster and more accurately.

Vehicle Braking Performance

Latency and inertia in the braking system must also be considered. First, the driver takes time \( t_d \) to respond to a warning and push the brake pedal \([39]\). Second, the ego-vehicle slows down to \( V_{crit} \) in time \( t_b \):

\[
t_b = \frac{V_i - V_{crit}}{2\mu g} \tag{3.6}
\]

where \( \mu \) is a coefficient of friction appropriate for the road conditions modelled \([124]\), while \( g \) is the gravitational constant. Thus, after a warning is issued, the ego-vehicle travels a distance \( D_b \):

\[
D_b = V_i t_d + \frac{(V_i)^2 - (V_{crit})^2}{2\mu g} \tag{3.7}
\]

while the ego-vehicle slows down to speed \( V_{crit} \).

3.2.5 Measurements at Each Sample for Each Feature Point

Combining disparity \( d \) and reference image gives \( I_L \) the location of each feature point in image plane as \( I_L(u, v, d) \). By using the stereo configuration parameters feature point location in real-world is measured as

\[
\vec{O} \rightarrow \begin{bmatrix} \hat{X}^r \\ \hat{Z}^r \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \begin{bmatrix} u \\ \frac{v}{f/\tau} \end{bmatrix} - \begin{bmatrix} b \\ 0 \end{bmatrix} \tag{3.8}
\]

where it is assumed that \( \hat{X}^r(u+\tau, v, d) > \hat{X}^r(u, v, d) \) and the optical axes crosses the stereo camera sensors \( L|R \) at \( I_{L|R}(0, 0, d) \).
In a safe system, one must consider all measurement errors.

This is particularly important here because, due to the discrete sensor pixels, disparities are integral and measured Z values lie in a discrete set, separated by disparity change boundaries (shown as horizontal dotted lines in Fig. 3.8). This causes significant errors in trajectory estimates. However, as shown in [3], this error decreases significantly as the object crosses disparity change boundaries. Discrete pixels introduce an error in \( X \) measurements too.

The measurement error for \( X \) arises from discrete pixels also but is magnified by the distance, \( \hat{Z} \), to the feature point. So, the actual position for a point observed at \( u \) with disparity \( d \), lies in the uncertainty area bounded by the four points:

\[
\vec{O}_q = \left[ \begin{array}{c} Z(d) \pm \left( \frac{Z(d)-Z(d+1)}{2} \right) \frac{\tau}{f} (u \mp \frac{\tau f}{2} - \frac{b}{2}) \\ 0 \\ Z(d) \pm \left( \frac{Z(d)-Z(d+1)}{2} \right) \end{array} \right]
\]

(3.9)

where the four values for \( q = \{0, 1, 2, 3\} \) are obtained by taking all combinations of +,- for the \( \pm \) and \( \mp \) operators:

\[
q = 0 : \quad \pm = +, \mp = + \\
q = 1 : \quad +, - \\
q = 2 : \quad -, - \\
q = 3 : \quad -, +
\]

Note that \( q = 2 \) represents the nearest point to the ego-vehicle in the region of uncertainty.

Object Velocity Estimation

I denote the observed position of the \( j \)th point in Frame \( k \), as \( \vec{O}_{q,k,j} \) (extending the notation of Equation 3.9).

For Frame \( k \), the relative velocity \( \vec{V}_j \) for a feature point \( j \) ranges between \( \min_j \vec{V}^f \) and \( \max_j \vec{V}^f \), where

\[
\min_j \vec{V}^f = \frac{1}{l} \left[ \begin{array}{cc} \hat{X}_{2,k,j} - \hat{X}_{0,0,j} \\ 0 \\ \hat{Z}_{3,k,j} - \hat{Z}_{1,0,j} \end{array} \right]
\]

(3.10)

and

\[
\max_j \vec{V}^f = \frac{1}{l} \left[ \begin{array}{cc} \hat{X}_{1,k,j} - \hat{X}_{0,0,j} \\ 0 \\ \hat{Z}_{1,k,j} - \hat{Z}_{2,0,j} \end{array} \right]
\]

(3.11)
where $t$ is the time between Frame $k$ and 0; $X_{q,k,j}$ is the $X$ component of $\vec{O}_q^r$ in the $k^{th}$ frame for $j^{th}$ feature point.

After the first change in disparity, the assumption that the object moves at constant velocity allows the velocity extremes to be truncated to the maximum values consistent with all previous observations. After further changes in disparity, the velocity uncertainty reduces further, but probably it would not be wise to continue to assume constant velocity as objects will often change velocity; therefore, I only apply this constraint over one disparity change so as to model a more realistic scenario in which some change in object velocity may be expected. Thus the 0 in Equations 3.10 and 3.11 should be replaced by the index of the frame at which the previous disparity change was observed.

Each feature point has its own range of velocities. Since I assume a rigid object, in which all points move at the same velocity, the system chooses the largest minimum and the smallest maximum as the range of velocities is consistent with all feature point observations; for the whole object:

$$\begin{align*}
\min V^r = & \begin{bmatrix}
\max (\min_1 V_{x}^r, \min_2 V_{x}^r, \ldots, \min_n V_{x}^r) \\
\max (\min_1 V_{z}^r, \min_2 V_{z}^r, \ldots, \min_n V_{z}^r)
\end{bmatrix} \\
\max V^r = & \begin{bmatrix}
\min (\max_1 V_{x}^r, \max_2 V_{x}^r, \ldots, \max_n V_{x}^r) \\
\min (\max_1 V_{z}^r, \max_2 V_{z}^r, \ldots, \max_n V_{z}^r)
\end{bmatrix}
\end{align*}$$

(3.12) (3.13)

After each observation, the velocity range narrows, but it never reaches zero. After Frame 2, the extremes of the velocity range often represent speeds which are unrealistically high for the current traffic scenario, e.g. a speed of over 100 km/h is rare in dense urban traffic. These high possible speeds mean that the system must warn of a possible collision after the second frame when, in fact, a collision is extremely unlikely. To avoid excessive false warnings, the model assumes that the highest speed of an object cannot be more than a speeding factor $s$ times the legal limit: $V_{\text{max}} = sV_{\text{limit}}$.

The range of trajectory angles $(\rho_L, \rho_R)$ is then computed from the truncated $\min V^r$ and $\max V^r$.

Figure 3.5 shows a typical narrowing of the range of trajectory angles. The range of possible trajectories is shown after each observation. Affect of threshold of speeds is shown for Observation 2. After Observation 4, the system is at state $S_3$ as it issues a precautionary warning - where an object could be anywhere within the range of possible trajectories but as its worse case represents a collision. While after Observation 5 the system could be at state $S_4$ as it would have issued a necessary warning for a definite collision.
3. Avoiding Constant Velocity Objects

Figure 3.5: Range of estimated paths of an object represented by the reference point on the object, initially at $\hat{O}_r^{p,q,k,1}(0)$ and the ego-vehicle first at $\hat{O}_i^{i}(0)$ in world co-ordinates.

Tangent Angles

I convert the $j^{th}$ feature point’s nearest measured position to polar co-ordinates in (X,Z) plane: $\vec{O}_{z,k,j}^{2} \rightarrow (D,\zeta)$. The tangents to the ego-vehicle’s exclusion zone are:

$$\zeta_L = \zeta - \sin^{-1}\left(\frac{r_{exc}}{D}\right) \quad \zeta_R = \zeta + \sin^{-1}\left(\frac{r_{exc}}{D}\right)$$

(3.14)
System States

For an object observed in the scene, the system associates with it one of the following states $S = \{S_0, S_1, S_2, S_3, S_4\}$ as presented in Table 3.1.

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>First frame, only distance is known</td>
</tr>
<tr>
<td>S1</td>
<td>Object will not collide with the ego-vehicle</td>
</tr>
<tr>
<td>S2</td>
<td>Object may collide, but safe to make further observations</td>
</tr>
<tr>
<td>S3</td>
<td>Object may collide, but not safe to make further observations. Issue precautionary warning</td>
</tr>
<tr>
<td>S4</td>
<td>Object will definitely collide with the ego-vehicle. Issue necessary warning</td>
</tr>
</tbody>
</table>

Table 3.1: System states.

Judging a Collision

When an object is first observed (Frame $k = 0$ for that object), its state is $S_0$. After observations ($k > 0$), the system computes $(\rho_L, \rho_R)$ for the nearest possible location of the $j^{th}$ feature point, $\overrightarrow{O_{2,k,j}}$. If $(\rho_L, \rho_R)$ includes the collision trajectory, $\zeta$, the system flags a possible collision (states $S_2$, $S_3$ or $S_4$), otherwise it is safe ($S_1$).

The advantages of considering a set of feature points over the extent of object instead of a single reference point are: Firstly, constraints between the feature points narrow the trajectory range. Secondly, range of trajectories immediately narrows when a feature point crosses a disparity change boundary. Lastly, trajectories which are avoiding for one feature point, i.e. trajectories outside $(\zeta_L, \zeta_R)$, could be colliding within $(\zeta_L, \zeta_R)$ for the other feature points.

So, if the extremes of the trajectory range for any feature point intersect the exclusion zone, then the system issues a necessary warning (state $S_4$). Otherwise, the system is either at state $S_2$ or $S_3$.

The model uses Algorithm 1 for computing the maximum tolerable speed. Algorithm 1 calls system module Algorithm 2 to identify states $S_2 \cup S_3, S_4$ for a partial or all colliding trajectories. Figure 3.6 illustrates the interaction of Algorithms 1 and 2 with Algorithm 1. Depending on the output state, the following steps of the system vary: If state $S_2 \cup S_3$ then Algorithm 3 is called to decide whether a precautionary warning is due or the system can safely wait for an observation. Otherwise, if state is $S_4$ then the system issues a necessary warning. Later, the model decides whether the warning issued by the system is timely, hence the colliding speed is tolerable too.
Algorithm 1 Computing the maximum tolerable speed

\[
\text{maxTolerableSpeed}(\vec{O}(0), \vec{V}, \vec{V}_i, \vec{V}_{max}, r_{exc}, n, L, H, W, f, b, \tau, d_{max}, \delta s, t_b, t_d, t_{p_f}, D_b) \rightarrow \text{speed}
\]

- Initialize \( \vec{O}^t = \vec{O}(0), \vec{V}^t = \vec{V} - \vec{V}_i \) and tolerable speed, \( \vec{V}_{safe} = \vec{0} \);
- \( t = 0, S \leftarrow S_0 \);
- Initialize the set of feature points over the object extent \( [\vec{O}_0^r, \vec{O}_1^r, \vec{O}_2^r, \ldots, \vec{O}_n^r] \);

\begin{algorithmic}
\For{each Observation \( k \) in \{0, 1, \ldots\}}
    \For{each feature point \( j \) in \{1, 2, \ldots, n\}}
        \State Update feature point position \( \vec{O}_{k,j}^t = \vec{O}_{0,j}^r + \vec{V}_r^t \);
        \If{\( \vec{O}_{k,j}^t \) in stereo CFoV (see Section 3.2.4)}
            \State Determine \( \hat{\vec{O}}_{k,j}^t \);
            \State Determine \( \min_j \vec{V}_r^t \) and \( \max_j \vec{V}_r^t \);
        \EndIf
    \EndFor
    \State Select feature points \( (j_x, j_z) \) at nearest X and Z distances, \( \hat{X}_{2,k,jx}^r \) and \( \hat{Z}_{2,k,jz}^r \);
    \If{\( t > 0 \)}
        \State Compute \( \min V_r^t \) (Equation 3.12) and \( \max V_r^t \) (Equation 3.13) and apply threshold \( V_{max}^r \);
        \State Compute \( (\rho_L, \rho_R) \) - trajectory angle range;
        \State Compute tangent angle arrays \( (\zeta_L, \zeta_R) \) for each feature point to the exclusion zone;
        \State \( S \leftarrow \text{CollisionDecision}(\rho_L, \rho_R, \zeta_L, \zeta_R, n) \) (see Algorithm 2);
        \If{\( S \in \{S_2, S_3\} \)}
            \State Compute \( \min \vec{V} \) from \( \min V_r^t \);
            \State \( S \leftarrow \text{canWait}(\hat{X}_{2,k,jx}^r, \hat{Z}_{2,k,jz}^r, \min V_z, \min V_z, V_r^t, t_b, t_d, \delta s, r_{exc}, D_b) \) (see Algorithm 3);
        \EndIf
        \If{\( S \in \{S_3, S_4\} \)}
            \State Check if vehicle can avoid a collision by braking;
            \If{collision is avoidable}
                \State speed \( |\vec{V}_r^t| \) is safe, set \( \vec{V}_{safe} = \vec{V} \);
            \Else
                \State Collision detected, but too late;
                \State \( \text{return} \) (previous) \( |\vec{V}_{safe}| \);
            \EndIf
        \EndIf
    \EndIf
\EndFor
\State Consider another observation \( t = t + \delta s \);
\end{algorithmic}
3.2. The Model

Figure 3.6: Steps followed by the model to interact with the system for a colliding object travelling at a constant velocity.

The system uses the nearest observed feature points in the directions $X - (j = jx)$ and $Z - (j = jz)$ and checks whether it is safe to consider additional observations (state $S2$) or not (state $S3$) (see Algorithm 3). Considering additional observations allows the system to refine the object trajectory range $(\rho_L, \rho_R)$ which narrows with each new observation.

An example of the transitions between states is presented in Fig. 3.5 which shows how the estimated trajectory range narrows with each observation and the maximum speed truncation is applied in Frame 2.

3.2.6 Tolerable Speed Contour Generation

In order to generate safe speed contours, I first set the $z$-component of the object’s velocity, $V_i^z = 0$, compute the trajectory for a collision, $\zeta = \tan^{-1}\left(-\frac{O_x(0)}{-O_y(0)}\right)$ and use it to compute $V_x^r = (V_z - V_i^z) \tan \zeta$. I then determine if the stereo system would issue a
warning in time to avoid a collision by braking alone. If it could, then $V_z$ is increased until warnings cannot be issued in time.

### 3.3 Results and Discussion

#### 3.3.1 Model Input

The model takes the optical and vehicle parameters as inputs from Table 3.2. Few values in the Table 3.2 are derived from the inputs: $V_{max}, Z_{min}(d_{max}), \theta, t_b, D_b$. And few others are just random numbers: $n = 9$. 

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Typical value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f$</td>
<td>Focal length</td>
<td>5mm</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Pixel size</td>
<td>4.7$\mu$m</td>
</tr>
<tr>
<td>$b$</td>
<td>Baseline length</td>
<td>750 mm</td>
</tr>
<tr>
<td>$d_{max}$</td>
<td>Maximum disparity</td>
<td>127</td>
</tr>
<tr>
<td>$\phi$</td>
<td>Vergence angle</td>
<td>0°</td>
</tr>
<tr>
<td>$\delta s$</td>
<td>Sampling interval</td>
<td>0.03 s</td>
</tr>
<tr>
<td>$r_{exc}$</td>
<td>Radius of vehicle exclusion zone</td>
<td>1 m</td>
</tr>
<tr>
<td>$V^i$</td>
<td>Vehicle speed</td>
<td>17 $ms^{-1}$(60kmh)</td>
</tr>
<tr>
<td>$V_{crit}$</td>
<td>Maximum collision speed</td>
<td>2.77 $ms^{-1}$(10kmh)</td>
</tr>
<tr>
<td>$V_{limit}$</td>
<td>Maximum speed limit</td>
<td>17 $ms^{-1}$(60kmh)</td>
</tr>
<tr>
<td>$s$</td>
<td>Speeding factor</td>
<td>1.5</td>
</tr>
<tr>
<td>$t_d$</td>
<td>Driver response time</td>
<td>0.5 $s$</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Coefficient of friction</td>
<td>0.4</td>
</tr>
<tr>
<td>$t_p$</td>
<td>Object detection and classification time</td>
<td>1.5ms</td>
</tr>
<tr>
<td>$L \times H \times W$</td>
<td>Object size</td>
<td>(3 $\times$ 0 $\times$ 2)$m$</td>
</tr>
<tr>
<td>$n$</td>
<td>Number of feature points</td>
<td>9</td>
</tr>
<tr>
<td>$V_{max}$</td>
<td>Object maximum speed</td>
<td>25.5 $ms^{-1}$(90kmh)</td>
</tr>
<tr>
<td>$Z_{min}(d_{max})$</td>
<td>Minimum depth in CFoV</td>
<td>8.4m</td>
</tr>
<tr>
<td>$\theta$</td>
<td>Half angle of stereo field of view</td>
<td>25.6°</td>
</tr>
<tr>
<td>$t_b$</td>
<td>Vehicle braking time</td>
<td>1.8 $s$</td>
</tr>
<tr>
<td>$D_b$</td>
<td>Maximum safe braking distance</td>
<td>44.4 m</td>
</tr>
</tbody>
</table>

Table 3.2: System parameters used in the model.
3.3. Results and Discussion

3.3.2 Analysis

Depth resolution degrades with distance, so the system needs to take more observations to observe a disparity change for an object that is farther from the ego-vehicle. The range of trajectories for an object with feature points close to disparity change boundaries narrows faster than that for an object which lies entirely in a disparity region so that all its points have the same disparity value.

With the change in observed disparity, the system now knows that the object is approaching the ego-vehicle but it would still be uncertain if it is going to collide with the ego-vehicle or could safely cross path with the ego-vehicle. After the disparity change the system infers a velocity that is consistent with all previous observations.

An example is presented in Fig. 3.7 where the system parameters as shown in Table 3.2 are used for a single reference point. The reference point first appears at $[0, 0, 118]^T_m$ and is moving with a velocity $V = [0, 0, -0.2]^T ms^{-1}$. For all the initial observations up to $k = 8$ it is observed at the same disparity $d = 9$ with its range of velocities $[\max V_x^r - \min V_x^r]$ gradually narrowing down. Due to the introduction of new uncertainty around the newly observed disparity after $k = 9$, the velocity limit $\min V_x^r$ goes up to $-71 ms^{-1}$ instead of the previously estimated velocity $-21 ms^{-1}$. However, since the new estimate is inconsistent with the previous observations the system chooses the last consistent velocity limit of $-21 ms^{-1}$.

3.3.3 Results

The model proposed in the study takes a set of input parameters (Table 3.2) and outputs a 2D contour map for each set of parameter values. All maps generated in the course of this study are presented in Appendix A.

Figures (3.8 to 3.11) present only a few of the contour maps generated by the model. The typical values presented in Table 3.2 with two focal lengths ($f = 5, 9mm$), two baseline lengths ($b = 750, 1000mm$), three pixel sized sensors ($\tau = 7.2, 4.7, 2.4 \mu m$) and several object sizes (a) point ($n = 1$), (b) pedestrian ($1 \times 1m, n = 9$) and (c) vehicle ($5 \times 2m$) are used.

The ego-vehicle uses two cameras in a canonical stereo configuration.

The tolerable speeds are higher for objects appearing farther away. Note that I have constrained the object speed to $25.5 ms^{-1}$ (90 kmh in a 60kmh zone).

The following sections, explain in detail how some representative results are derived: first for different sized objects, then different $V_{\text{crit}}^i$ for same sized object and finally for different locations. For simplicity, $Y$ components have been omitted from all vectors as I only consider motion on the XZ-plane (see Table 3.3).
Avoiding Constant Velocity Objects

Figure 3.7: Trajectory Range Narrowing: The range of possible velocities narrows down with each new observation. Note that once the disparity changes after \( k = 9 \), the measured velocity limits should remain consistent with previous observations. So, \( \max V'_z \) rightly becomes zero. Whereas, initially \( \min V'_z < -21 \) (shown as red line), so the system instead chooses \( \min V'_z = -21 \) (shown by green line).

Different Sizes

For the chosen scenarios, the braking time is \( t_{\text{brake}} = t_d + t_b + \delta s = 2.3s \), so that if time to collision is greater than this, the system can wait for another observation. I use a typical configuration (cf. Table 3.2) and show how results differ for two different sized colliding objects: (a) a single point object (\( n = 1 \)) and (b) a vehicle ((5×2)m, \( n = 9 \)). From Fig. 3.8 and Fig. 3.9 for point and vehicle: both first
Table 3.3: Scenarios: Highlights different parameters along with Table 3.2 configuration parameters used in the discussed scenarios. $|\vec{V}| = 25.5 ms^{-1}$ in all cases.

<table>
<thead>
<tr>
<th>Sizes</th>
<th>Point</th>
<th>X (m)</th>
<th>Z (mm)</th>
<th>$b$</th>
<th>$V_{i}^{i_{crit}}$ (ms$^{-1}$)</th>
<th>$\vec{V}$ (ms$^{-1}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Vehicle</td>
<td>10</td>
<td>144</td>
<td>750</td>
<td>10</td>
<td>(-2.9, -25.3)</td>
</tr>
<tr>
<td>$V_{i}^{i_{crit}}$</td>
<td>Vehicle</td>
<td>10</td>
<td>134</td>
<td>1000</td>
<td>10</td>
<td>(-2.9, -25.3)</td>
</tr>
<tr>
<td>Locations</td>
<td>Point</td>
<td>28</td>
<td>144</td>
<td>750</td>
<td>10</td>
<td>(-5.7, -24.8)</td>
</tr>
</tbody>
</table>

Table 3.4 shows the output of the system at each frame. The system uses Algorithm 3 to compute $Z_{safe}$ and $cO$, and then determines whether a warning is due (at state S3).

At Frame 0, the single point is initially observed at $d = 6$. For the vehicle, eight of the nine feature points are initially observed at $d = 6$, while one is observed at $d = 5$. At Frame 1, that point has crossed the disparity change boundary from 5 to 6. This change narrows the range of trajectories significantly as the range $(\rho_L - \rho_R)$ goes from $(108^\circ - 266^\circ)$ for the point to $(180^\circ - 266^\circ)$ for the vehicle.

For both objects, at Frame 15, a disparity change is observed for the nearest reference point, but the system remains at state S2. By Frame 19, the single point still has a very wide range of trajectories $(180^\circ - 267^\circ)$, whereas the vehicle’s range is only $(264^\circ - 266^\circ)$. At Frame 19, the vehicle’s range decreases further to $(266.1^\circ - 266.8^\circ)$ and one of the feature points is definitely tracking between the ego-vehicle’s exclusion zone tangents $(266.1^\circ - 266.9^\circ)$, so the system goes to state S4 and a warning is issued in time.

For the single point the system does not go to state S3 (causing a warning) until Frame 27, but the object is safely avoided. Thus, as the object size increases, warnings at state S4 are issued earlier. For small objects, the system can also issue a timely warning at state S3, providing the object’s speed is not higher than the tolerable speed shown in the contour map (Fig. 3.10).

**Different $V_{i}^{i_{crit}}$**

Varying $V_{i}^{i_{crit}}$ also varies the braking distance $D_b$. Hence, it also effects the timing of a precautionary warning at state S3. In this experiment, I show that if the system
can safely consider additional observations, then it can also go to a state $S_4$.

I use a typical configuration (Table 3.2) with $b = 1000\,mm$ and show how the type of warnings differs for a colliding vehicle when (a) $V_{\text{crit}}^i = 10\,kmh = 2.8\,ms^{-1}$, and (b) $V_{\text{crit}}^i = 30\,kmh = 8.3\,ms^{-1}$. The colliding vehicle first appears at $(10, 134\,m)$ moving with $V = [-2.9, -25.3]^T$ (speed $|V| = 25.5\,ms^{-1}$), $V_f = [-2.9, -42.3]^T\,ms^{-1}$ (see Table 3.5).

For $V_{\text{crit}}^i = 10\,kmh$, $D_b = 44.4\,m$ and $t_b = 1.8s$, whereas for $V_{\text{crit}}^i = 30\,kmh$, $D_b = 36.5\,m$ and $t_b = 1.1s$.

From Frame 0 to Frame 6, the object is observed at $d = 8$. At Frame 7, the first disparity change is observed but the range of trajectories is still very wide ($180^\circ$–$267^\circ$).

At Frame 16, a second disparity change is observed, which narrows the possible
3.3. Results and Discussion

Figure 3.9: Tolerable speeds (in $\text{m s}^{-1}$) on the ground for a vehicle with $n = 9$ and Table 3.2 configuration parameters with $f = 5\text{mm}$, $b = 750\text{mm}$, $(w \times h) = (1024 \times 768)$ and $\tau = 4.7\mu\text{m}$.

trajectories range to ($264^\circ$–$266^\circ$), but it is still wider than the tangents to the ego-vehicle’s exclusion zone ($265^\circ$–$266^\circ$). For $V_{\text{crit}}^i = 10\text{kmh}$, $t_{\text{brake}}$ is longer, therefore the predicted worst case $Z$-position is $42.4\text{m}$ compared to $60\text{m}$ for $V_{\text{crit}}^i = 30\text{kmh}$. So, for $V_{\text{crit}}^i = 10\text{kmh}$, the system goes to state $S3$ and collision is avoided in time.

At Frame 19, for $V_{\text{crit}}^i = 30\text{kmh}$ the system goes to state $S4$ as the range of trajectories ($265.2^\circ$–$265.8^\circ$) is definitely tracking between the exclusion zone tangents ($265^\circ$–$266^\circ$) and the object is safely avoided in time.

Different Locations

For a typical configuration and a colliding point object ($n = 1$), I show how truncation of the maximum velocity to some ‘reasonable’ limit ($V_{\text{max}}$) causes the tolerable speed
to be the same even though one colliding object initially appears at farther distance than the other one.

Consider two colliding objects first appearing from locations \( P = (28, 144)m \) and \( Q = (28, 148)m \) (see Fig. 3.11). \( \vec{V} = [-5.7, -24.8]^T \) (speed \( |\vec{V}| = 25.5ms^{-1} \)) is tolerable for both objects. At Frame 0, \( P \) is first observed at \( d = 6 \) while \( Q \) is observed at \( d = 5 \). By Frame 2, \( P \) has moved to \( (19.6, 142.6)m \) and is still observed at \( d = 6 \) because it is still within the limits for \( d = 6 \), namely 123.1m to 145.4m.

At Frame 2, \( Q \) has reached \( (19.6, 145.2)m \) and is observed at \( d = 6 \) having crossed the disparity change boundary at \( Z = 145.4m \). However, even after the first disparity change the range of trajectories is still very wide for both objects and the system considers further observations (state \( S2 \)).
### 3.3. Results and Discussion

<table>
<thead>
<tr>
<th>Frame # (k)</th>
<th>Actual position ([X, Z]^{T} m)</th>
<th>(d)</th>
<th>Object (cO_z)</th>
<th>Trajectory range ((\rho_L - \rho_R))</th>
<th>State Trajectory range ((\rho_L - \rho_R))</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>((10.0, 144.0))</td>
<td>–</td>
<td>–</td>
<td>S0</td>
<td>–</td>
<td>S0</td>
</tr>
<tr>
<td>1</td>
<td>((9.9, 142.6))</td>
<td>6</td>
<td>64</td>
<td>((108°-266°))</td>
<td>(180°-266°)</td>
<td>S2</td>
</tr>
<tr>
<td>2</td>
<td>((9.8, 141.2))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>((9.7, 139.8))</td>
<td>104</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>((9.6, 138.4))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>((9.5, 136.9))</td>
<td>17</td>
<td>6</td>
<td>((109°-266°))</td>
<td>(180°-266°)</td>
<td>S2</td>
</tr>
<tr>
<td>6</td>
<td>((9.4, 135.5))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>((9.3, 134.1))</td>
<td>47</td>
<td>8</td>
<td>((112°-266°))</td>
<td>(266°-266.2°)</td>
<td>S4</td>
</tr>
<tr>
<td>8</td>
<td>((9.2, 132.7))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>((9.1, 131.3))</td>
<td>5</td>
<td>10</td>
<td>((109°-266°))</td>
<td>(180°-266°)</td>
<td>S2</td>
</tr>
<tr>
<td>10</td>
<td>((9.0, 129.9))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>((8.9, 128.5))</td>
<td>4</td>
<td>18</td>
<td>((110°-266°))</td>
<td>(265°-266°)</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>((8.8, 127.1))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>((8.7, 125.7))</td>
<td>3</td>
<td>33</td>
<td>((112°-266°))</td>
<td>(266°-266.2°)</td>
<td>S4</td>
</tr>
<tr>
<td>14</td>
<td>((8.6, 124.3))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>((8.5, 122.8))</td>
<td>2</td>
<td>23</td>
<td>((180°-266°))</td>
<td>(264°-266°)</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>((8.4, 121.4))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>((8.3, 120.0))</td>
<td>1</td>
<td>21</td>
<td>((180°-266°))</td>
<td>(265°-266°)</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>((8.2, 118.6))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>((8.1, 117.2))</td>
<td>0</td>
<td>19</td>
<td>((180°-267°))</td>
<td>(266°-266.2°)</td>
<td>S4</td>
</tr>
<tr>
<td>20</td>
<td>((8.0, 115.8))</td>
<td>7</td>
<td>47</td>
<td>((180°-267°))</td>
<td>(266°-266.2°)</td>
<td>S4</td>
</tr>
<tr>
<td>21</td>
<td>((7.9, 114.4))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>((7.8, 113.0))</td>
<td>6</td>
<td>14</td>
<td>((180°-268°))</td>
<td>(266°-266.2°)</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>((7.7, 111.6))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>((7.7, 110.2))</td>
<td>5</td>
<td>12</td>
<td>((180°-268°))</td>
<td>(266°-266.2°)</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>((7.6, 108.8))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>((7.5, 107.3))</td>
<td>4</td>
<td>10</td>
<td>((180°-268°))</td>
<td>(266°-266.2°)</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>((7.4, 105.9))</td>
<td>8</td>
<td>35</td>
<td>((265.5°-266°))</td>
<td>(266°-266.2°)</td>
<td>S3</td>
</tr>
</tbody>
</table>

Table 3.4: Point vs. Vehicle: Tolerable speeds of objects first appearing at \([10, 144]^{T} m\) for a point object \((n = 1)\) and a vehicle \(((5 \times 2)m, n = 9)\) for Table 3.2 configuration parameters. \(O_z(t_d + t_b) = 45m\). \(d\) represents disparity.
Avoiding Constant Velocity Objects

$V_{\text{crit}} = 10\text{kmh}$ and $30\text{kmh}$

<table>
<thead>
<tr>
<th>Frame # ($k$)</th>
<th>Actual Position $[X, Z]^T\text{m}$</th>
<th>$d$</th>
<th>Observed Position $(\tilde{X}<em>{k,iz}, \tilde{Z}</em>{k,iz})$</th>
<th>Trajectory range $(\rho_L - \rho_R)$</th>
<th>State 10kmh</th>
<th>State 30kmh</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(10.0, 134.0)</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>S0</td>
<td>S0</td>
</tr>
<tr>
<td>1</td>
<td>(9.9, 132.6)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>(9.8, 131.2)</td>
<td>8</td>
<td>(8.9, 125.5)</td>
<td>(108°–266°)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>(9.7, 129.8)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>(9.6, 128.4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>(9.5, 126.9)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>(9.4, 125.5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>(9.3, 124.1)</td>
<td>9</td>
<td>(8.1, 112.3)</td>
<td></td>
<td>S2</td>
<td>S2</td>
</tr>
<tr>
<td>8</td>
<td>(9.2, 122.7)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>(9.1, 121.3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>(8.9, 119.9)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>(8.8, 118.5)</td>
<td></td>
<td>(7.9, 112.3)</td>
<td>(180°–267°)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>(8.7, 117.1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>(8.6, 115.7)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>(8.5, 114.3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>(8.4, 112.9)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>(8.3, 111.4)</td>
<td>10</td>
<td>(7.4, 101.6)</td>
<td>(264°–266°)</td>
<td>S3</td>
<td>S4</td>
</tr>
<tr>
<td>17</td>
<td>(8.2, 110.0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>(8.1, 108.6)</td>
<td></td>
<td>(7.1, 101.6)</td>
<td>(265°–266°)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>(8.0, 107.2)</td>
<td></td>
<td></td>
<td>(265.2°–265.8°)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.5: $V_{\text{crit}}$: Tolerable speeds of a colliding vehicle first appearing at $[10, 134]^T$ for Table 3.2 configuration parameters with $b = 1000\text{mm}$. For $V_{\text{crit}} = 10\text{kmh}$ $Z_{\text{safe}} = 45\text{m}$ and for $V_{\text{crit}} = 30\text{kmh}$ $Z_{\text{safe}} = 37\text{m}$. At $d = 8, 9$ and $10$ object worst case trajectories $cO_z$ in braking time for $V_{\text{crit}} = 10\text{kmh}$ are $66\text{m}, 53\text{m}$ and $42\text{m},$ whereas for $V_{\text{crit}} = 30\text{kmh}$ are $84\text{m}, 71\text{m}$ and $60\text{m}. d$ represents disparity.

At Frame 27, both trajectory ranges are still very wide, but the system can not wait longer so it issues a precautionary warning (state S3) for $P$ as the object worst case trajectory would take less than the braking time to collide with the ego-vehicle. However, for $Q$ the system can still safely consider additional observations (S2). At the time of warning $P$ is at $(14.8, 106.4)\text{m}$ and is safely avoided by braking in time.

At Frame 30, $Q$ goes to S3 (precautionary warning). At this time $Q$ is at $(14.4, 106.2)\text{m}$ and is also safely avoided.
3.3. Results and Discussion

Figure 3.11: Tolerable speeds (in \( m s^{-1} \)) on the ground for a single reference point for Table 3.2 configuration parameters with \( f = 5 \text{mm} \), \( b = 750 \text{mm} \), \((w \times h) = (640 \times 480)\) and \( \tau = 7.2 \mu \text{m} \).

3.3.4 Warning Types

This stereo based safety system is safe but it issues many unnecessary warnings at state \textbf{S3}. Due to the stereo uncertainties the system issues late warnings at state \textbf{S4}. Figure 3.12 shows the tolerable speeds for the parameters presented in Table 3.2 with \( b = 1000 \text{mm} \) for warnings issued at either: (\textbf{S3} or \textbf{S4}) or (\textbf{S4} only). The colliding object is a vehicle \((5 \times 2) \text{m}\).

Note that Fig. 3.12 top, is a copy of Fig. 13(a) in Appendix 6.2 and is shown again here only for direct comparison. It is for warnings at states \textbf{S3} and \textbf{S4}. Fig. 3.12 bottom, is for warnings at state \textbf{S4} only.
3.4 Conclusions for This Chapter

The intent of this study was to provide a tool for the safety engineer, not to prescribe stereo configurations as there are many competing constraints (e.g. economic, social, etc.) beyond the scope of this work. The tool described here enables a designer to assess the effect of competing configuration parameters (e.g. focal length, baseline length, sensor pixel size) against other criteria such as desired opposing object speed. I consider here some general observations on the generated maps.

Higher speeds are not tolerated for smaller size objects, but (luckily!) smaller objects would generally be relatively slowly moving pedestrians. For larger objects, stereo systems can more accurately determine an object’s course and issue a state S4 (definitely needed!) warning. A key factor for the issuing of state S4 warnings (compared to state S3 ones, which are often unnecessary) is observing a disparity change, so the system designer should try to provide high depth resolution in critical regions to increase the probability of disparity changes being observed. As I show, a stereo based safety system can be made ‘safe’ in the sense that no warnings are missed or late, but may also issue too many warnings due to the initial uncertainty in estimation of the opposing object’s speed and direction. This raises the possibility that simple inexpensive auxiliary devices (e.g. SONAR) with limited capabilities (e.g. able to locate objects moving in one direction only) could be effectively used to reduce the false warnings.
Algorithm 2  Definite or possible collision decision

\textbf{CollisionDecision}(\rho_L, \rho_R, \zeta_L, \zeta_R, n) \textbf{ returns } system state

Initialize state, \( S \leftarrow S_0 \):

\textbf{for} each feature point \( j \) in \( \{1, 2, \ldots, n\} \) \textbf{ do}

\hspace{1em} if no trajectories collide - \((\rho_L > \zeta_{R,j} \text{ OR } \rho_R < \zeta_{L,j})\) then

\hspace{2em} Consider next feature point;

\hspace{1em} else

\hspace{2em} if \((\rho_L \geq \zeta_{L,j} \text{ AND } \rho_R \leq \zeta_{R,j})\) then

\hspace{3em} return \( S_4 \) (definite collision);

\hspace{2em} else

\hspace{3em} Set \( S \leftarrow S_2 \);

\hspace{3em} Compute the overlap \((\varrho_{L,j}, \varrho_{R,j})\) between \((\rho_L, \rho_R)\) and \(\zeta_{L,j}, \zeta_{R,j}\);

\hspace{2em} end if

\hspace{1em} end if

\hspace{1em} end for

\textbf{if} \( S = S_2 \) \textbf{ then}

\hspace{1em} Compute the minimum angle \( \varrho_L \) and \( \varrho_R \) from arrays \( \varrho_{L,j} \) and \( \varrho_{R,j} \);

\hspace{1em} if \( \varrho_L > \rho_L \) OR \( \varrho_R < \rho_R \) \textbf{ then}

\hspace{2em} return \( S_2 \setminus S_3 \);

\hspace{1em} else

\hspace{2em} Sort \( \varrho_{L,j} \) and the corresponding \( \varrho_{R,j} \) in ascending order with respect to \( \varrho_{L,j} \);

\hspace{2em} Compute the length \( ll \) of array \( \varrho_{L,j} \);

\hspace{2em} Initialize \( mm = 1 \);

\hspace{2em} \textbf{while} \( \varrho_{R,mm} \geq \varrho_{L,(mm+1)} \) AND \( mm < ll \) \textbf{ do}

\hspace{3em} All angles between \( \varrho_{L,(mm+1)} \) and \( \varrho_{R,mm} \) are colliding; \( mm = mm + 1 \);

\hspace{2em} \textbf{end while}

\hspace{2em} if all trajectories are avoiding with \( mm = 1 \) \textbf{ then}

\hspace{3em} return \( S_1 \);

\hspace{2em} end if

\hspace{2em} if all trajectories are colliding with \( mm = ll - 1 \) \textbf{ then}

\hspace{3em} return \( S_4 \);

\hspace{2em} else

\hspace{3em} return \( S_2 \);

\hspace{2em} end if

\hspace{2em} end if

\hspace{1em} end if

\hspace{1em} return \( S \);
Algorithm 3 Algorithm for precautionary warnings

\textbf{canWait}(\hat{X}^r_{2,k,jx}, \hat{Z}^r_{2,k,jz}, \text{min}V_x, \text{min}V_z, V_i^t, t_b, t_d, \delta s, r_{\text{exc}}, D_b) \text{ returns state;}

Determine whether to issue precautionary warning (S3) because its nearest point will collide in less than the braking time plus a sample time, \(t_{bs} = t_d + t_b + \delta s\). Object worst case position after braking would be

\[
\overrightarrow{cO} = \begin{bmatrix}
\hat{X}^r_{2,k,jx} + \text{min}V_x t_{bs} \\
\hat{Z}^r_{2,k,jz} + \text{min}V_z t_{bs} - (D_b + V_i^t \delta s)
\end{bmatrix}
\]

\([c_{\zeta_L}, c_{\zeta_R}] = \text{computeTangents}(\overrightarrow{cO})\) (see Section 3.2.5);

Maximum vehicle Z-distance to reach \(V_i^t\) exit: \(Z_{\text{safe}} = V_i^t \delta s + D_b - r_{\text{exc}} \cos c_{\zeta_L}\);

Minimum object Z-distance after \(t_{bs}\): \(cO_z = \hat{Z}^r_{2,k,jz} + \text{min}V_z t_{bs}\);

if \(cO_z \leq Z_{\text{safe}}\) then

\text{ return } S3 \text{ to Algorithm }[1]

else

\text{ return } S2 \text{ to Algorithm }[1]

end if
Figure 3.12: Difference between tolerable speeds if warnings are only at S4 compared to the safe situation where the timely warnings are given at (S3 or S4). Tolerable speeds are in $ms^{-1}$. 
In this chapter, I have generalized my previous tool for assisting a safety engineer in assessing collision trajectories. Here I have extended the tool from colliding objects with constant velocity to include more general variable velocity. I have also highlighted that a linear system cannot be relied upon for handling a colliding object with variable velocity. To deal with such velocity, a weighted system is implemented where past observations are given weights depending on the estimated velocities at those locations; priority is given to locations with reduced velocity. Based on this hypothesis, I have shown that the weighted system outperforms a linear one. The benefit is that it always issues a timely warning, even if the trajectory of the colliding object keeps changing over time.

Parts of this chapter have been published in IEEE Intelligent Vehicles Symposium 2013. Paper [8] was a joint work with Reinhard Klette.

4.1 Introduction

As pointed out in [5, 4], for a collision scenario with a colliding object moving with constant velocity, a stereo-based safety system can issue timely warnings. The system described in [5] used the estimated worst case trajectory to issue a precautionary warning (true positive warning). However, a similar trajectory within the uncertainty region could actually be on safe course, yet still be considered as colliding (false positive warning). Table 4.1 shows the details of the type of warnings. In order to check the robustness of a linear system, I have generalized the model proposed in Chapter 3 by considering an object moving with constant speed but variable direction leading to a variable velocity.

The assumption of a constant velocity for a colliding object is possible but sometimes it is hard for both object and ego-vehicle to maintain a constant speed and trajectory. A constant speed can be achieved through cruise-control. However, wind, wheel misalignment and human factors would still cause variation in its trajectory. Therefore, here I have improved the model in order to make it possible to determine
4. More General Trajectories

<table>
<thead>
<tr>
<th>Warning</th>
<th>Actual trajectory</th>
<th>Estimated trajectory</th>
</tr>
</thead>
<tbody>
<tr>
<td>False positive warning</td>
<td>Avoiding</td>
<td>Colliding</td>
</tr>
<tr>
<td>False negative warning</td>
<td>Colliding</td>
<td>Avoiding</td>
</tr>
<tr>
<td>True positive warning</td>
<td>Colliding</td>
<td>Colliding</td>
</tr>
</tbody>
</table>

Table 4.1: Types of warnings issued by a warning system.

![Collision scene observed by the reference camera. Left: Bicyclist first observed. Right: Bicyclist is very close to collide with ego-vehicle.](image)

the accuracy of estimated trajectory of an object moving with variable velocity. Note that the ego-vehicle is still assumed to be moving with constant velocity. Figure 4.1 shows an example of such real-world collision scenario. In this scenario, a bicyclist is moving in a parabolic path towards the path of ego-vehicle which is moving at constant speed.

The model now takes an additional input parameter to control the change in object trajectory over time. This parameter is rate of relative change of angular velocity $\delta c$ with reference to the ego-vehicle. The sub-pixel accuracy in measured disparities is also considered as an input to the model. Due to variable trajectory, the system takes longer to move to the warning states $S_3$ or $S_4$. Therefore, for an object that exits the stereo CFoV earlier with system still at state $S_2$, the system continues to determine its state based on the previous measurements.

4.2 Single vs. Multiple Feature Points

Observing a change in feature point position is one of the crucial aspects of issuing a timely warning. Any feature point closer to the disparity change boundary would
4.2. Single vs. Multiple Feature Points

Figure 4.2: Left column shows the reference images, while the right column shows the colour-coded disparities (the streaking effects are due to a DP matcher) after background subtraction. Pose of the car is changing clockwise while observed from Y-axis.

be more likely to come across such change than the farther one. For a rigid object moving with variable velocity and with variable pose, using multiple feature points can be beneficiary.

Figure 4.2 shows an example scenario. In this scenario, a toy-car with dimensions $[L \times W \times H] = [0.1, 0.1, 0.3]$ m is rotating around its XZ-centre. I use a stereo system in canonical configuration, defined by parameters $f = 16$ mm, $b = 60$ mm, $\delta s = 0.03$ s and $\tau = 4.65 \mu$m. I assumed that the stereo matcher generates integral disparities without sub-pixel accuracy. The stereo system uses a DP matcher to generate dense disparity maps at each pixel (no sub-pixel accuracy). After background subtraction, I colour-code the disparity levels over the surface of the toy-car.

As the pose of the toy-car changes over time, the observed disparities over its surface also change accordingly. Due to integral disparities, only few pixels over the surface are observed at a different disparity in a frame time $\delta s$. Therefore, instead of using a single feature point, multiple feature points can assist in improving the system’s decision.

As common for a safety system, the nearest object position is considered to safely avoid the worst case scenario. For the sequence used in Figure 4.2 with some post-processing, the nearest feature point can be chosen as a top-corner point, central point, or a bottom-corner point. Post-processing involves excluding smaller real-world contours on the toy-car, i.e. smaller than $[L \times W \times H] = [0, 0.10, 0.05]$ m.

The output is the measured distance between each chosen feature point and the centre of the baseline over time, and is shown in Figure 4.3. There is very little variation in these distances apart from the few occasions where a DP mismatch occurs. In order to handle mismatches, it would also be good to use more than one feature point to vote for the best outcome; but this is not implemented in these experiments.
4. More General Trajectories

Figure 4.3: Distance of the three nearest points on the toy-car to the centre of the baseline.

4.3 Assumptions

Since the model and the weighted system presented here are derived from the model and system proposed in Chapter 3, a few steps are similar. Figure 4.4 presents the interaction of this weighted system with the model.

A colliding object is still rigid and is travelling with a constant speed $V$ on a flat surface with $Y = 0$, however its direction may vary along $XZ$-components over time. The object’s $L$ dimension is always in the direction of the object’s motion. As the direction of motion changes, so would the pose of the object.

The collision trajectory of an object varies with its velocity so the object remains on a colliding trajectory within an exclusion zone of radius $r_{exc}$ around the ego-vehicle.

4.4 Model

4.4.1 Always Colliding Trajectory

The object is initially on a collision course $\zeta_L$ as a left tangent

$$\zeta_L = \tan^{-1}\left(\frac{-O_{rz}(k)}{-O_{rx}(k)}\right) - \sin^{-1}\left(\frac{r_{exc}}{D^r(k)}\right)$$

to the ego-vehicle’s exclusion zone, where $D^r(k) = \sqrt{(O_{rx}(k))^2 + (O_{rz}(k))^2}$ is the distance of the object to the ego-vehicle i.e. centre of the baseline (see Fig. 4.5). Note
4.4. Model

Figure 4.4: Interaction between model and weighted system. The position of the object reference point determines the object position in the model.

that angles \( \zeta_r^L \) and \( \zeta_r^R \) are in polar co-ordinates.

The initial object velocity at \( k = 0 \) is:

\[
V^r(k) = \begin{bmatrix} V^r \cos \zeta_r^L \\ 0 \\ V^r \sin \zeta_r^L \end{bmatrix}
\]  \hspace{1cm} (4.1)

with object’s \( L \) dimension in the direction of \( \overrightarrow{V(k)} \). After each system Observation
$k - 1$, the object moves to a new position

$$O^r(k) = O^r(k - 1) + (V^r(k - 1) \cdot \delta s)$$

The model assumes that the object changes its trajectory before every system observation except the first one. This effectively would also change the object pose that is derived by repositioning $n$ feature points (see Section 4.4.2) on the object’s surface. The object trajectory changes from $\zeta_{k-1}$ to $\zeta_{k-1} \pm \delta c$, where $\delta c$ is an input to the model (see Fig. 4.5).

Figure 4.5: Object velocity changes with each observation. An object is on a colliding trajectory when it crosses the exclusion zone which is indicated by a circle of radius $r_{exc}$.

Figure 4.6 illustrates the use of the $\pm$ notation. The same procedure is also later explained in Algorithm 4 in Section 4.4.2. At $k = 0$, labelled as position A in Fig. 4.6, the object trajectory direction is $\zeta_0 = \zeta^L_0$. For the following observations, $\pm = +$ is
4.4. Model

Figure 4.6: Collision trajectory: the trajectory direction of an object varies with each Observation \( k \). Four locations A, B, C and D show where the object changes its trajectory. The path and pose of the object at each instance is different relative to the ego-vehicle. The object collides in time \( t_c \).

used by the model to change object trajectories for all observations between locations A to C. Later, if \( \pm = + \) is used any further, then \((\zeta_{k-1}^r + \delta c) > \zeta_R^r\), where

\[
\zeta_R^r = \tan^{-1}\left(\frac{-O_r^x(k)}{-O_r^z(k)}\right) + \sin^{-1}\left(\frac{r_{exc}}{D^v(k)}\right)
\]

is the right tangent to the ego-vehicle exclusion zone from the object at Observation \( k \). At this path, the object would pass behind the ego-vehicle. So, for a collision scenario, \( \pm = - \) is used for changing object collision trajectories for all observations between locations C and E.
4. More General Trajectories

4.4.2 Object Pose

The pose of an object is determined based on its real-world velocity derived from Equation 4.1. The real-world velocity is defined as follows:

\[
\overrightarrow{V}(k) = \begin{bmatrix}
V_x^r(k) \\
0 \\
V_z^r(k) + V_z^i
\end{bmatrix}
\]  

(4.2)

The angle of the actual object trajectory, relative to the X-axis, is \( \eta_k = \tan^{-1}\left(\frac{V_z(k)}{V_x(k)}\right) \), and it is used by the model to determine the object pose before Observation \( k \) (and after system Observation \( k - 1 \)).

In the model, we consider \( n \) equidistant feature points on the surface of the colliding object. The nearest observed point in real-world co-ordinates is the reference point \( \overrightarrow{O}_j(k) \) (at \( j = 1 \)), and it is at position \( \overrightarrow{O}_r(k) \).

As the object trajectory changes with respect to \( \overrightarrow{V}(k) \), \( \eta_k \) also changes. This leads to a change in positions of each feature point on the object’s surface with respect to the object reference point \( \overrightarrow{O}_1(k) \).

The feature point \( j \) is at distance \( c \) to the object reference point \( \overrightarrow{O}_1(k) \) along the \( W \) dimension and it moves to a new position

\[
\overrightarrow{O}_j(k) = \begin{bmatrix}
O_{1x}(k) + c \cos (\eta_k - \frac{\pi}{2}) \\
0 \\
O_{1z}(k) + c \sin (\eta_k - \frac{\pi}{2})
\end{bmatrix}
\]  

(4.3)

Similarly, the feature point \( j \) at distance \( c \) to the object reference point \( \overrightarrow{O}_1(k) \) along the \( L \) dimension and it moves to a new position

\[
\overrightarrow{O}_j(k) = \begin{bmatrix}
O_{1x}(k) + c \cos (\eta_k - \pi) \\
0 \\
O_{1z}(k) + c \sin (\eta_k - \pi)
\end{bmatrix}
\]  

(4.4)

It is assumed that the system makes Observation \( k \) after the object pose has been adjusted by the model. Algorithm 4 shows the steps followed by the model to change object velocity and its pose before each system observation.

The model still adopts Algorithm 1 in (see Section 4.4.2) but uses Equation 4.3 and Equation 4.4 instead of

\[
\overrightarrow{O}_{k,j} = \overrightarrow{O}_{0,j} + \overrightarrow{V}t
\]

to represent the object’s new position.
Algorithm 4 Modelling a collision scenario with variable velocity and weighted system decision making.

\textbf{GeneralCollisionTrajectories}(\delta s, V^r, \delta c, r_{exc}, \vec{V}, O(0)) \ returns \ system \ state \ S \ at \ each \ Observation \ k; 
Initialize state, S \leftarrow S0 \ and \ Observation \ k = 0; \nObject \ initial \ relative \ position \ \vec{O}^r(0) = \vec{O}(0); 
for each Observation k do 
\begin{align*}
\text{Relative object distance} \ D^r(k) &= \|\vec{O}^r(k)\|; \\
\text{Relative collision trajectory to the ego-vehicle is} \ \zeta_k &= \tan^{-1}\left(\frac{-\vec{O}^r(k)}{\vec{O}^r_z(k)}\right); \\
\text{Left tangent to exclusion zone in polar co-ordinates is} \ \zeta^r_L &= \zeta_k - \sin^{-1}\left(\frac{r_{exc}}{D^r(k)}\right); \\
\text{Right tangent to exclusion zone in polar co-ordinates is} \ \zeta^r_L &= \zeta_k + \sin^{-1}\left(\frac{r_{exc}}{D^r(k)}\right); \end{align*}
\begin{align*}
\text{if} \ First \ observation \ at \ k = 0 \ \text{then} & \\
\text{Initial collision trajectory is the left tangent,} \ \zeta^r_k &= \zeta^r_L; \\
\text{For \ \pm \ notation \ SET} \ \text{MinusFlag} \leftarrow TRUE; \\
\text{else} \ & \\
\text{if} \ \zeta^r_k \ \text{is supposed to increase} \ (\text{MinusFlag} == FALSE) \ \text{then} \\
\text{if} \ \left(\zeta^r_{k-1} + \delta c\right) > \zeta^r_R \ \text{then} & \\
\zeta^r_k \ \text{has to decrease, SET} \ \text{MinusFlag} \leftarrow TRUE; \\
\text{else} & \\
\zeta^r_k \ \text{has to increase, SET} \ \text{MinusFlag} \leftarrow FALSE; \\
\text{end if} \ & \\
\text{else} \ & \\
\text{if} \ \left(\zeta^r_{k-1} + \delta c\right) < \zeta^r_L \ \text{then} & \\
\zeta^r_k \ \text{has to increase, SET} \ \text{MinusFlag} \leftarrow FALSE; \\
\text{else} & \\
\zeta^r_k \ \text{has to decrease, SET} \ \text{MinusFlag} \leftarrow TRUE; \\
\text{end if} \ & \\
\text{end if} \ & \\
\text{if} \ \text{MinusFlag} == TRUE \ \text{then} \\
\text{Decrease} \ \zeta^r_k &= \zeta^r_{k-1} - \delta c; \\
\text{else} & \\
\text{Increase} \ \zeta^r_k &= \zeta^r_{k-1} + \delta c; \\
\text{end if} \ & \\
\text{end if} \ & \\
\text{Use Equation 4.1 to derive the object collision velocity} \ \vec{V}^r(k) \ \text{at Observation} \ k; \\
\text{Use Section 4.4.2 to determine the pose of the object as positions of} \ n \ \text{feature points from reference point} \ j = 1 \ \text{at} \ \vec{O}^r(k) \ \text{using} \ \vec{V}(k); \\
\text{Use Section 4.5.2 to determine the system state} \ S; \\
k &= k + 1; \\
\text{Object reference point} \ (j = 1) \ \text{moves to new position} \ \vec{O}^r(k) &= \vec{O}^r(k-1) + \vec{V}^r(k-1) \cdot \delta s; \\
\text{end for}
4.5 Warning System

4.5.1 Design Constraints

This system has the same design constraints as discussed in Section 3.2.4 and in [5]. The uncertainty of measurement leads to a possible range of trajectories derived from the range of velocities ($\hat{\text{min}}_j V_r$ to $\hat{\text{max}}_j V_r$) (see Equation 3.10 and Equation 3.11 and [5]). To handle the changing trajectories of an object travelling with variable velocity, I compute the weighted average of velocity ranges. This average is computed after assigning a weight $\vec{w}_k = [w^k_x, 0, w^k_z]^T$ to each feature point at Observation $k$.

4.5.2 Weighted System

For simplicity, I assume that the procedure described further below would be applicable to each feature point independently. Later, feedback from each feature point would be taken into consideration for computing the velocity extrema. Furthermore, because the object is assumed to be moving on the flat surface hence measurement uncertainty along v-axis is insignificant, hence it is no longer considered in this section. So, instead of $(u, v, d)$ for position in image plane, I use $(u, d)$ instead.

At Observation $k$, weight $\vec{w}_k$ is directly related to the time $\Delta t$ for which a feature point is observed at the same position $(u, d)$ with some uncertainty ($\pm \nu_u, \pm \nu_d$). The weight $\vec{w}_k$ is also indirectly related to the uncertainty at the measured position $(\Delta X(u, d), \Delta Z(d))$,

$$\vec{w}_k = \begin{bmatrix} \Delta t_x \\ \frac{\hat{X}_{1,k} - \hat{X}_{0,k}}{\hat{Z}_{1,k} - \hat{Z}_{0,k}} \\ 0 \\ \frac{\Delta t_x \hat{Z}_{1,k} - \hat{Z}_{0,k}}{\hat{Z}_{1,k} - \hat{Z}_{0,k}} \end{bmatrix}$$

(4.5)

where $\hat{Z}_{1,k}$ is the observed $Z$-position at $\hat{Z} = (u, d)$ at Observation $k$. Similarly, $\hat{X}_{0,k}, \hat{X}_{1,k}, \hat{Z}_{0,k},$ and $\hat{Z}_{1,k}$ are the observed positions at the boundary of the measured pixel $\hat{X} = (u + \nu_u, d - \nu_d)$, $\hat{X} = (u - \nu_u, d + \nu_d)$, $\hat{Z} = (u, d - \nu_d)$, and $\hat{Z} = (u, d + \nu_d)$. $\hat{Z}_{1,k}$ is used in Equation 4.5 as a scaling factor for the $Z$-term of $\vec{w}_k$.

Due to the uncertainty of stereo measurement, the measured velocity is a range as already discussed in Chapter 3 in Equation 3.12 and Equation 3.13 (also in [5]). To compute their weighted outcome at observation $k$, I generalize the measured velocity
4.5. Warning System

Term as $mV^k$. While, $wmV^k$ as their corresponding weighted velocity for $k > 0$,

$$wmV^k = \begin{bmatrix}
\frac{wV^k - 1}{\sum_{i=0}^{k-1} w_i + kW^k \sum_{i=0}^{k}} \\
\sum_{i=0}^{k-1} w_i + kW^k \sum_{i=0}^{k}\\
0\\
\frac{wV^k - 1}{\sum_{i=0}^{k-1} w_i + kW^k \sum_{i=0}^{k}} \\
\sum_{i=0}^{k-1} w_i + kW^k \sum_{i=0}^{k}
\end{bmatrix}$$

(4.6)

where $wV^{k-1}$ keeps the feedback from previous observations and is computed at previous observation $k - 1$ by assuming that at $k = 0$, $wV^k = 0$, $mV^k = 0$, and $wmV^k = 0$ are initialized by 0,

$$wV^k = \begin{bmatrix}
\frac{wV^k - 1}{\sum_{i=0}^{k-1} w_i + kW^k \sum_{i=0}^{k}} \\
\sum_{i=0}^{k-1} w_i + kW^k \sum_{i=0}^{k}\\
0\\
\frac{wV^k - 1}{\sum_{i=0}^{k-1} w_i + kW^k \sum_{i=0}^{k}} \\
\sum_{i=0}^{k-1} w_i + kW^k \sum_{i=0}^{k}
\end{bmatrix}$$

(4.7)

**Velocity Extrema:** For each feature point $j$, the velocity extrema (before weighted average) ranged between $\underline{\underline{\text{min}}}_j V_r$ (computed through Equation 3.10) and $\underline{\underline{\text{max}}}_j V_r$ (computed through Equation 3.11). After obtaining the weighted average using Equation 4.6, the weighted velocity extrema range between $\underline{\underline{\text{min}}}_{wm,k} V_r$ and $\underline{\underline{\text{max}}}_{wm,k} V_r$.

In Chapter 3 the object was assumed to be moving with constant velocity hence the system computed the velocity extrema to be the maximum values consistent with all previous observations. Here, however, since the object is moving with variable velocity, the weighted system chooses the best outcome based on the weight derived in Equation 4.7.

The object is still assumed to be rigid. Therefore, the system computes the largest minimum and the smallest maximum as a range of velocities being consistent with all feature point observations. Thus, for the whole object we have that

$$\underline{\underline{\text{max}}} V_r^T = \begin{bmatrix}
\text{min} (\underline{\underline{\text{max}}} V_r^x, \ldots, \underline{\underline{\text{max}}} V_r^x) \\
0\\
\text{min} (\underline{\underline{\text{max}}} V_r^z, \ldots, \underline{\underline{\text{max}}} V_r^z)
\end{bmatrix}$$

(4.8)

$$\underline{\underline{\text{min}}} V_r^T = \begin{bmatrix}
\text{max} (\underline{\underline{\text{min}}} V_r^x, \ldots, \underline{\underline{\text{min}}} V_r^x) \\
0\\
\text{max} (\underline{\underline{\text{min}}} V_r^z, \ldots, \underline{\underline{\text{min}}} V_r^z)
\end{bmatrix}$$

(4.9)
The range of trajectory angles \((\rho_L, \rho_R)\) is then computed from the truncated \(\min \overrightarrow{V}\) and \(\max \overrightarrow{V}\).

In order to make a decision if a collision is likely, the weighted system follows the same procedures as the ones described in Section 3.2.4, Section 3.2.4, and Section 3.2.5.

### 4.6 Experiments and Discussion

#### 4.6.1 Model Input

As discussed previously in Section 3.2.6, the model has a wide range of parameters. In the experiments discussed below I have used the standard parameters presented in Table 3.2, and the additional input parameter rate of relative change of angular velocity \(\delta c = 0.1^\circ\) with respect to the ego-vehicle.

#### 4.6.2 Experiment 1: Comparing Linear and Weighted Systems

The objective of this experiment is to show that the longer a linear system observes the object at same position, the better its estimate would be of collision likelihood. Thus, more weight is assigned to such observations in a weighted system.

In this experiment, a few of the typical parameters presented in Table 3.2 are varied: \(r_{exc} = 2m\), \(b = 1500mm\) and \(f = 9mm\). In order to control the variable collision trajectory a new input parameter \(\delta c = 0.1^\circ\) is also introduced. I also assume that the disparity map is up to half pixel. To compare and analyse the behaviour of the linear and weighted systems, I simplify the problem by using a single feature point \(n = 1\).

I assume that the colliding object first appears at location \(A_1 = (20.0, 110.0)m\) and is on a collision parabolic path. Initially, for both linear and weighted systems the feature point is observed at \((20.1, 110.7)m\) from the ego-vehicle while the object is moving at \(\overrightarrow{V} = [-3.41, -0.04]ms^{-1}\) (speed \(|\overrightarrow{V}| = 3.4ms^{-1}\)). At this stage, the object is on a trajectory to collide with the ego-vehicle as a left tangent to the ego-vehicle exclusion zone. After the second observation the object is observed closer to the ego-vehicle at \([20.1, 108.7]m\) due to the change observed in disparity at a sub-pixel level. After this observation, the systems can determine the range of trajectories for the first time.

Figure 4.7 shows the graph plots of the range of velocities estimated by a linear system. The graph consists of peaks and valleys. Since the system estimates the precautionary warning using \(\min \overrightarrow{V}\) (derived from \(\min \overrightarrow{V}\)), a peak leading \(\min \overrightarrow{V}\) closer to actual velocity is more important than a valley. Such peak is developed
gradually for observations of the object at constant position. However, it changes to a valley with a change in observed position. This describes the reason for the weighted system giving priority to observations of the object at constant position for a longer time. After the first valley with a first change in observed position, the linear system only considers the previous observations since last change as to allow for changing object velocity. However, that reduces the number of observations to an insufficient number for this experiment to have a good estimate.

The linear system fails to correctly identify the colliding object as its range of trajectories ($246.5^\circ$–$258.3^\circ$) is definitely not being tracked between the exclusion zone tangents ($258.6^\circ$–$260.8^\circ$). Hence the system issues a false negative warning.
(see Table 4.1).

On the other hand, Fig. 4.8 shows the plots of the range of velocities estimated by the weighted system. As the system assigns appropriate weight to each observation, it gradually narrows down the range towards the actual velocity. The weighted system issues a timely precautionary warning at state \( S_3 \) after \( k = 111 \). Furthermore, for an object appearing from location \( A_1 \) the weighted system can safely tolerate speeds up to \( |\vec{V}| = 19 \text{ms}^{-1} \).

### 4.6.3 Experiment 2: Linear vs. Weighted Systems Contours Plot

For this experiment I have used the typical parameters presented in Table 3.2, with \( n = 1 \). Figure 4.9 (a) and (b) show the maximum tolerable speeds for a linear and for
a weighted system respectively. The tolerable speeds are only for a collision scenario for objects first appearing at any of these locations within stereo CFoV.

Figure 4.9 shows that the a weighted system outperforms the linear one. Both systems have similar behaviour for closer distances as both issue timely precautionary warnings. However, for farther distances, when the object would take longer time before colliding with the ego-vehicle, the system can safely delay the precautionary warning. The longer time before collision should allow the systems to improve their estimates. However, as the trajectory keeps changing, the linear system estimates that the object is safely avoiding the ego-vehicle and fails to recognize a collision scenario. On the other hand the weighted system assigns weight to each measurement based on the time it is observed at the same location, hence it avoids false estimates.

**Experiment 3: Anomalies of a Linear System**

When a linear system uses the typical configuration parameters as presented in Table 3.2, it manifests certain anomalies for objects appearing at greater distances (see Fig. 4.9(a)). To analyse these anomalies I assume that two similarly sized objects are first appearing at two different locations $A = [4, 112]^T m$ and $B = [4, 114]^T m$ in front of the ego-vehicle. Since $A$ is closer it should have a lower tolerable speed than $B$; however, the experimental results show that the maximum tolerable speed of $A$ ($25.4 \text{ m/s}$) is unexpectedly higher than that of $B$ ($1.1 \text{ m/s}$). To analyse this anomaly, I assume the collision speed of $V = 1.2 \text{ m/s}$ which is intolerable at $B$ but tolerable at $A$.

$A$ being closer to the ego-vehicle is first observed at $d = 13.5$, while $B$ being farther away is first observed at $d = 13.0$ (sub-pixel level). After the second Observation, the system can determine the range of trajectories for the first time. At this stage, the system estimates that the objects could possibly collide, but as the objects are quite far from the ego-vehicle, the system can safely wait for additional observations.

Object appearing from location $A$, is at $(2.6, 76.6)m$ from the ego-vehicle after 61 Observations. The system can no longer wait for additional observations and hence issues a timely precautionary warning. On the other hand, object first appearing from location $B$ is still farther away at $(2.7, 78.6)m$ from the ego-vehicle, so the system chooses to consider additional observations. After Observation 64 for this object, the system has the estimated range of trajectories $(\rho_L, \rho_R) = (262.9, 266.8)^\circ$. Since this range does not cross the ego-vehicle’s exclusion zone tangents $(\zeta_L, \zeta_R) = (266.9, 269.9)^\circ$, the system estimates that object from location $B$ is safe and issues a false negative warning.
4.7 Conclusions for This Chapter

The designed tool can assist a safety engineer in assessing the effect of various stereo configurations, including the effects of various vehicle and object parameters. Previously I designed a similar tool that could only estimate constant object trajectories. Now, by modelling variable collision trajectories, I have shown that a linear system can falsely estimate a safe travel.

To mitigate this problem, I designed a weighted system which always issues a timely warning for a collision scenario, provided the object is binocularly visible CFoV. Although these warnings are timely, they are mostly precautionary. As the system cannot work out the exact object trajectory, thus it issues a warning based on a possible worst case trajectory.
4.7. Conclusions for This Chapter

Figure 4.9: Tolerable speeds (in $\text{ms}^{-1}$) of a linear and a weighted system for a collision scenario. Parameters not specified in captions are taken from Table 3.2 configuration parameters. The dark-blue region within stereo CFoV has tolerable speeds less than or equal to $V_{\text{crit}} = 2.7 \text{ ms}^{-1}$.

(a) Linear system: $b = 750 \text{mm}$, $f = 9 \text{mm}$, $\tau = 4.7 \mu\text{m}$, $w \times h = 1024 \times 768$ pixels and $n = 1$

(b) Weighted system: $b = 750 \text{mm}$, $f = 9 \text{mm}$, $\tau = 4.7 \mu\text{m}$, $w \times h = 1024 \times 768$ pixels and $n = 1$
So far I assumed in my models in this thesis that feature points are tracked without any errors. I used the tracked feature points for estimating the object trajectory. However, in this chapter I show that their tracking is not a simple task in reality. First, to track them, I have to match them frame-by-frame and then, by using the off-line disparity maps, I can derive their real-world position and finally an estimated velocity.

Unfortunately, a feature-point matcher cannot find (reliable) matches in all frames. In fact, the performance of a matcher varies with type of feature point detector and descriptor used. In this chapter, I briefly evaluate the detection performance of various detectors, namely difference of Gaussian (DoG), speeded up robust features (SURF) detector, features from accelerated segment test (FAST), oriented binary robust independent elementary features (ORB), and the star detector, and various descriptors, namely scale invariant feature transform (SIFT), SURF, binary robust independent elementary feature (BRIEF), fast Retina keypoint (FREAK), and binary robust invariant scalable keypoints (BRISK).

Their comparison gives a general impression of how detector and descriptor performance degrades as a rigid object approaches the ego-vehicle in a collision scenario video sequence. To handle the mismatches, I use a Kalman-filter-based tracker for each tracked feature point. The tracker with the maximum number of matches and with a most recent match is chosen as the optimal tracker. The role of the optimal tracker is to assist in updating the tracker of a feature point which had no match. The optimal tracker is also used in estimating the velocity.

The recorded sequence has the object static, while the ego-vehicle moves towards the object for a collision. Hence, the estimated velocity is that of the ego-vehicle, which is used by the system to determine the safe braking distance. If it is no longer safe to wait for an additional observation, then the system issues a warning. The warning is timely if it is issued before the ego-vehicle crosses a marker on the road. To understand the behaviour of the safety system, I used the DoG detector in combination with SURF, FREAK, and BRIEF descriptors, while linBP and iSGM are used as stereo matchers.
5.1 Objective

The objective of this chapter is to practically test the findings from the models described in Chapters 3 and 4 for a collision scenario. The colliding object considered in my experiments is an opposing vehicle. To ensure the health and safety of the driver of the ego-vehicle (which in my case is HAKA1) and of the opposing vehicle, an actual collision cannot be recorded.

Therefore, instead a bigger exclusion zone around the ego-vehicle, with $r_{exc} = 3.6m$, is taken into consideration. Even with this set-up, an opposing vehicle crossing the in front of the path of our moving ego-vehicle and nearly missing the collision is not a safe exercise either. Therefore, the only safe collision experiment that can be performed is to keep the opposing vehicle static, while moving the ego-vehicle towards it.

5.1.1 Choreographing a Collision Scenario

The real-world experiments, to be reported in this chapter, are not as ideal as assumed in the theoretical models before. For example, the road surface is hardly planar leading also to an observed movement along the Y-axis. The constant speed maintained by the ego-vehicle is based on the HAKA1 speedometer whose measurements can vary based on various factors including tyre pressure and so forth. Nevertheless, there is a maximum target speed the driver has to maintain based on which the safe braking distance has been precomputed and marked on the road (see Figure 5.1).

In the experiment, my helpers and I kept a safe braking distance from the opposing vehicle. The marked position is used by the driver of the ego-vehicle to apply brakes after the ego-vehicle crosses it to safely avoid collision (without a warning system). Similarly, the safety system must recognise the collision scenario and issue a warning before the ego-vehicle crosses this mark on the road, hence issuing a timely warning. To validate this, an observer is used, who raises a flag after the ego-vehicle has crossed the marker. This observer is also visible to reference and match cameras in the ego-vehicle.

So, if the safety system can issue a warning (based on estimated trajectories) before the observer raises the flag, then the system is proven to be issuing a timely warning for that scenario. Figure 5.1 illustrates the choreography and an example of an iSGM disparity map for this scenario.
5.2 Introduction

To estimate the opposing vehicle or object trajectory, which may be a hazard for the ego-vehicle, the object has to be tracked as it moves around in a scene. Like stereo matching between reference and match cameras, tracking also involves searching for correspondences between images captured over time on the same camera. Generally, object tracking is considered to be a challenging task. Difficulties in tracking arise due to the following factors: object motion, camera motion, change in object pose, changes in the scene, non-rigid objects, and object occlusions.

The approach to tracking may vary based on the application. Few assumptions are used based on the application of tracking [134]. For example, in my theoretical models, I assumed that the object was rigid. I will continue with this assumption in this chapter as well. I will also keep the assumption that the tracked feature points are always binocularly visible. Due to the lack of time, I assumed that the object is pre-detected with a bounding box around the object.

Figure 5.1: Choreographed sequence. The observer is a person holding a flag. The flag is raised after the ego-vehicle, which is on a collision course, crosses the marker on the road. The marker on the road is at a safe braking distance from the colliding object. Bottom image: The iSGM disparity map for the stereo pair illustrated by one frame in the top image.
Before tracking the object, it has to be represented by either its shape or appearance. An object shape may be represented by a single point that usually is the centroid of the object \([127]\). As discussed previously in Chapters 3 and 4, for an object with its extent distributed over multiple disparities, it is better to represent the object by a set of points \([115]\). For example, Nedevschi et. al. detected 3D points over object edges to detect obstacles under real-time constraints \([101]\). If multiple feature points are used, then it becomes a challenge to distinguish which set of feature points belong to the same object over the course of time. Often, for a rigid object, a common motion constraint is applied where neighbouring features that are seen moving together are grouped to represent the same object \([36]\). For more complex non-rigid objects, a contour representation might also be used, where a contour outlines the boundary of an object \([133]\).

There are a number of ways to represent the object appearance. For example, estimating the probability densities of object features such as colour or texture within the region represented by a shape model. These estimates can either be parametric, for example Gaussian \([138]\) or a mixture of Gaussians \([105]\), or they can be non-parametric. Another way is to use templates formed by geometric shapes. However, it is less useful in collision scenarios where due to moving ego-vehicle and object, the pose of an object observed by the stereo system in the ego-vehicle changes over time.

Subspace methods such as principal component analysis (PCA) are widely used in pattern recognition applications \([1, 2]\). Similarly, independent component analysis (ICA) can be used to represent multiple object views and their shapes \([99, 24]\). However, a limitation of PCA or ICA is that they require appearances to be learned in a training phase – before object tracking.

I recall that the tracker is a module of the collision avoidance system. With the object assumed to be pre-detected, feature points are first identified on the detected object at observation Time \(k = 0\).

To track an object over time, these feature points are tracked in the following in observed stereo frames over time. In order to match the feature points, each feature point is represented by a descriptor. A descriptor can describe salient features also known as attributes. So, at \(k = 0\) the initially identified feature points are represented by their descriptors.

For discussing a tracking situation, consider the subsequent frame at \(k = 1\). The object’s position may change, leading to a change in positions of previously identified feature points.

There are two ways to track the feature points in general.

First, for each feature point in the previous frame, match with any possible hypothesis called particles of a particle filter (PF) \([31, 108]\). In such cases, each hypothesis is represented by a descriptor. The descriptor describes a feature point and
5.2. Introduction

Figure 5.2: Feature point tracking on the reference images.

is used to identify the confidence of each hypothesis, and the best hypothesis is chosen. The advantage of this approach is that it can work for non-linear measurements provided by stereo vision. However, the limitation is that a descriptor is still a representation of discrete pixels. Hence, multiple hypotheses at non-integer pixel positions may have the same descriptor. In addition to that, a chosen hypothesis may no longer be represented by the same distinctive feature as the original one. Hence, the feature points tend to be mis-tracked in the following frames.

This problem can be mitigated by using a greater number of particles, and, more importantly, by using a good descriptor which is not prone to scaling. For example, the previously used census transform in Chapter 2 is a poor descriptor for PF based feature points tracking. The only limitation a good descriptor could have, that it might consumes more processing time compared to a weaker one.

Second, the other alternative is to first detect feature points represented by distinctive features and descriptors in each frame, followed by matching among the detected feature points only. So, for an object detected in the first frame, the feature points detected over its region are the \textit{query feature points}, while the feature points detected in the following frame are the \textit{train feature points} (see Fig. 5.2). The advantage of this approach is that instead of pixel-to-pixel matching, the features already describe the regions which are most important. Hence, it reduces the search

1 The naming convention is consistent with the one used in \textit{Open Source Computer Vision} (OpenCV) library for matching the feature points.
space to a set of detected features. Therefore, I will be using query training feature points matching, for the object tracking as part of my driver assistance system.

Matches between the query-train feature points leads to 2-dimensional \(I_L(u,v)\) image space tracking. Provided the detection is performed on rectified reference images, then their disparity maps along with the cameras calibration data can be used to compute the real-world position of each detected feature point. However, the only limitation of feature points matching is that it can mismatch as well. Thus, an additional step of outlier removal is necessary to select only the correct matches. Due to this, at each following frame, not all query feature points have a correct match. Hence, to track each query feature point in real-world 3D space, a Kalman filter (KF) is used, which in the absence of correct match can predict the real-world position instead of the corresponding train feature point.

There are many existing feature point detection algorithms, few of which will be discussed here. Similarly, there are many ways to form a descriptor for describing a feature point. Chekhlov et al. used Scale-Invariant Feature Transform (SIFT) like spatial gradient descriptor with the Shi-Tomasi-Kanade Feature Detector (STK) and Unscented Kalman Filter (UKF) tracker. Davison et al. used NCC in an 11 × 11 window with the STK detector and Extended Kalman Filter (EKF) tracker. DiVerdi et al. used Speeded Up Robust Features (SURF) and optical flow with the STK detector with Random Sample Consensus (RANSAC) matcher. Similarly Se et al. used scale and orientation attributes with the DoG detector and KF tracker. While, Skrypnyk and Lowe used SIFT descriptor with the DoG detector with RANSAC matcher. Also, Lee and Hilger used SIFT, and optical flow descriptors with the DoG detector with RANSAC and KF tracker. Table 5.1 summarizes few of the existing feature point trackers.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Detector</th>
<th>Descriptor</th>
<th>Tracker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chekhlov et al.</td>
<td>STK</td>
<td>SIFT</td>
<td>UKF</td>
</tr>
<tr>
<td>Davison et al.</td>
<td>STK</td>
<td>NCC (11 × 11)</td>
<td>EKF</td>
</tr>
<tr>
<td>DiVerdi et al.</td>
<td>STK</td>
<td>SURF + optical flow</td>
<td>RANSAC</td>
</tr>
<tr>
<td>Se et al.</td>
<td>DoG</td>
<td>SIFT like</td>
<td>KF</td>
</tr>
<tr>
<td>Skrypnyk and Lowe</td>
<td>DoG</td>
<td>SIFT</td>
<td>RANSAC</td>
</tr>
<tr>
<td>Lee and Hilger</td>
<td>DoG</td>
<td>SIFT + optical flow</td>
<td>RANSAC + KF</td>
</tr>
<tr>
<td>Wu et al.</td>
<td>DoG</td>
<td>SIFT + col-histogram</td>
<td>PF</td>
</tr>
<tr>
<td>Yan et al.</td>
<td>DoG</td>
<td>SIFT + col-histogram</td>
<td>PF</td>
</tr>
</tbody>
</table>

Table 5.1: Various feature point trackers. Showing the detector, descriptor, and tracker. col-histogram stands for colour histogram.
5.3 Feature point Detector, Descriptor and Matcher

In my vision based driver assistance system, feature point matching is performed on the rectified reference images starting from the pre-detected object. Using disparity maps, the system derives the real-world position of each feature point. Later, by using the KF, it estimates the future position and velocity of each feature point. The estimated velocity is later used to determine whether the ego-vehicle is on a collision course. If it is then the system issues a braking warning to the driver, who can later apply brakes to avoid the collision.

5.3 Feature point Detector, Descriptor and Matcher

Given the reference rectified image as input, feature point detector looks for regions of interest within the image. Many feature detection algorithms in general look for rapid changes in image gradient. Such changes can form edges [30], corners [58] etc.

The feature point orientation is computed with respect to the direction of strong image gradient in a region. So, even if the image is rotated, due to the strong image gradient, the orientation of detected feature point can be changed. Furthermore, due to rotated orientation of the detected feature point, the descriptor can be computed, independent of the image rotation. Such feature point based algorithms are called rotation invariant. Furthermore, if the feature point based algorithm computes only fixed sized features instead of independently computing the optimal size for every feature point then that algorithm can detect same features, even if the image is scaled. Such algorithms are called scale invariant. Figure 5.3 shows an example scenario of two different views of the scene. There is a difference in the orientation and scaling of images. Still, the detected feature is rotation and scale invariant.

Figure 5.3: Example of rotation and scale invariant feature from two different views of the same scene [55].
5.3.1 Feature Point Detectors

Difference of Gaussian

Inspired by human object recognition process, Lowe proposed Scale Invariant Feature Transform (SIFT) [83]. The detector part of Lowe’s algorithm is discussed here, while the descriptor part will be discussed later. To detect the scale invariant feature points, a scale space is constructed by convolving the image with Gaussian filters at various scales. Then, the Difference of Gaussian (DoG) images are computed from the scaled images. Candidate feature points are chosen based on the minima and maxima of DoG images at various scales. Candidate feature points location are further refined by interpolating the neighbouring image intensities. Candidate feature points that are with low contrast or are at the edge are excluded [134], while the remaining form the set of detected scale invariant feature points.

Speeded Up Robust Features Detector

Speeded Up Robust Features (SURF) detector is said to be a rotation and scale invariant detector and was first proposed by Herbert Bay et al. [17]. The same also proposed SURF descriptor which will be discussed later.

This detector uses Haar wavelet approximation of the blob detector based on the determinant of Hessian matrix [82]. To reduce the computation time it also uses Viola and Jones integral images [128] that are formed by addition of all pixel intensities in the rectangular neighbourhood of a desired pixel location.

Features from Accelerated Segment Test

Features from Accelerated Segment Test (FAST) is an efficient corner detector [41]. For a candidate point \( p = (u_p, v_p) \), it uses a 16 pixels in a clockwise circle around \( p \) to detect whether \( p \) is a corner.

Let intensity of \( p \) be denoted as \( I_L(p) \). The candidate point is a corner, if there are \( C \) contiguous pixels in the circle all meeting the same intensity criteria of either all darker or all brighter than \( I_L(p) \).

Intensity criterion can be: darker, brighter or similar intensities. A corner point is detected, if it fulfills the brighter intensity criteria defined as: if all selected pixels intensities are brighter than \( I_L(p) + a \). Or, it fulfills the darker intensity criteria defined as: if all selected pixels intensities are darker than the \( I_L(p) - a \), where, \( a \) is the threshold.

To efficiently detect corner points and exclude non-corner points, a high-speed test is applied. The objective of high-speed test is to improve the performance, so, instead of choosing all 16 pixels in the neighbourhood, a step wise procedure is adopted.
Keeping the 12 contiguous pixels criteria, it starts by choosing neighbouring pixels at locations 1, 9, 5, and 13.

Firstly using pixels 1 and 9, it checks whether both meet the same intensity criteria. If they don’t, then \( p \) is not a corner. Otherwise, pixels 5 and 13 are also checked. If three out of the four chosen pixels meet the same intensity criteria, then the rest of neighbouring pixels are also tested.

Although the high-speed test improves the detector’s performance. However, there are few limitations. First, multiple features are detected very close to each other. Second, the efficiency of the detector varies with the choice of the neighbouring pixels as well as the number of neighbouring pixels. To avoid these limitations, FAST detector also incorporates a machine learning approach.

**Oriented Binary Robust Independent Elementary Features**

*Oriented Binary Robust Independent Elementary Features* (ORB) detector also known as Oriented FAST \([56]\), is an extension of FAST detector. Although FAST detector can detect corner feature points at various scales, but because it does not consider the orientation of a feature point, therefore it is not rotation invariant \([123]\). Whereas, ORB uses FAST on different scales of images of a pyramid to detect strongest corners \([109]\). Additionally it also computes the orientation of each corner point. The orientation is computed by first order moments.

**Star Features**

*Star detector* is part of OpenCV library and it is an extension of the scale invariant *Center Surrounded Extrema* (CenSurE) detector originally proposed by Agrawal et al. \([15]\). Instead of computing a more computationally expensive DoG or even *Laplacian of Gaussian* (LoG), CenSurE approximates the circular LoG by using a set of three integral images (including a square image and two slanted images) and an octagon filter of different sizes.

Instead of using an octagon filter, Star detector uses two squares, one of which is 45° rotated \([122]\). Figure 5.4 illustrates the difference in the filter shapes.

### 5.3.2 Feature Point Descriptors

A descriptor is a combination of attributes of the detected feature point. Often, feature point detectors themselves offer complimentary attributes to the descriptor. Attributes like edge orientation, multi scaling information and gradient magnitude are few examples. Due to this a more subtle analysis is necessary to differentiate whether a descriptor independent of the detector used, contains which attributes. Although,
Kingsbury argued that a descriptor cannot be rotation invariant as it is dependent on orientation detected by feature point detectors \cite{122}. Still, a descriptor on its own can determine orientation before computing the other attributes.

**Scale Invariant Feature Transform**

Lowe proposed a local image SIFT descriptor which is invariant to rotation and illumination \cite{83,84}. Each feature point descriptor is a 128-dimensional feature vector which elaborates the spatial structure and orientation of the patch surrounding the feature point. Eight different gradient magnitudes are computed for eight major directions. The gradient magnitude of each orientation is represented by 16 gradient histograms. Each gradient histogram itself is a representation of $4 \times 4$ grid superimposed on $16$pixel $\times$ $16$pixel patch surrounding the feature point.

To be scale invariant, the size of the patch has to be normalized in a scale-invariant fashion. Similarly, to be rotation invariant, a dominant gradient of the patch is first determined which is later used to compute the directional histograms.

This descriptor is also considered as illumination invariant, as like many others intensity based histogram classifiers \cite{104}, SIFT descriptor is also a combination of histograms.

**Speeded Up Robust Features Descriptor**

The SURF descriptor describes the intensity distribution in the neighbourhood of the detected feature point. Like SURF detector, SURF descriptor also uses integral images along with Haar wavelets. To be rotation invariant, the gradient values in $u$ and $v$ direction are computed from Haar wavelets in a circular neighbourhood. The radius of this neighbourhood is derived from the scale at which the feature point was
detected [17]. Figure 5.5 illustrates the Haar wavelet filters used to compute the responses in these directions.

Figure 5.5: The darker region has the weight -1, while lighter region has the weight of +1. Left image: Filter to compute response in $u$ direction. Right image: Filter to compute response in $v$ direction.

After weighting the responses with a Gaussian whose standard deviation is also derived from the scale at which feature point was detected. A sliding orientation window of size $\frac{\pi}{3}$ is used, to compute the sums of all weighted responses in both directions. The two summed responses are used to form a local orientation vector. The longest orientation vector of all the orientation windows is used to define the orientation of the detected feature point [17].

**Binary Robust Independent Elementary Feature Descriptor**

Calonder et al. proposed *Binary Robust Independent Elementary Feature* (BRIEF) descriptor [29]. Calonder et al. suggested that instead of creating a long vector (for example, SIFT has 128-dimensional descriptor), a short binary descriptor vector can be constructed where all bits are independent.

To reduce sensitivity to noise a Gaussian smoothing is applied by a 9pixels × 9pixels averaging filter centred on the feature point. Later, using the Gaussian distribution around the detected feature point, random pixels are chosen for comparison. The proposed bitwise descriptor vector is obtained by comparing the intensity of 512 pairs of pixels in a 48pixels × 48pixels region.

One of the advantages of a binary descriptor is that the matching of descriptors is efficient. It is replaced from the usual Euclidean distance to a simpler exclusive OR based Hamming distance. However, the disadvantage is that the descriptor is neither invariant to orientation nor scale changes. The solution to this is to couple it with a detector which is invariant to changes in orientation, for example ORB detector.
5. Accuracy of Trajectory Estimation

<table>
<thead>
<tr>
<th>Detector</th>
<th>Rotation invariant</th>
<th>Scale invariant</th>
</tr>
</thead>
<tbody>
<tr>
<td>DoG</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>SURF</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>FAST</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>ORB</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Star</td>
<td></td>
<td>yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Rotation invariant</th>
<th>Scale invariant</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>SURF</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>BRIEF</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>FREAK</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>BRISK</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 5.2: Summary of rotation and scale invariance.

**Fast Retina Keypoint Descriptor**

Alahi et al. originally proposed *Fast Retina Keypoint* (FREAK) descriptor that is an extension of BRIEF descriptor [16]. Inspired by human eye retinal pattern, the sampling of chosen points follows a specific pattern with more chosen points closer to the detected feature point and as the distance increases the number of chosen points reduce exponentially. Due to the specific sampling pattern approach the feature descriptor allows for the ‘coarse to fine’ approach. The descriptor is a binary vector consisting of sum of estimated local gradients over selected point pairs.

**Binary Robust Invariant Scalable Keypoints**

Leutenegger et al. originally proposed *Binary Robust Invariant Scalable Keypoints* (BRISK) as an extension of BRIEF descriptor [31]. However, unlike BRIEF descriptor, BRISK uses a symmetric pattern to choose the sample points in the form of concentric circles around the detected feature point. Each sampled point is represented after applying a Gaussian blur. The standard deviation of this blur is increased with the sampled point’s distance from the detected feature point.

Table 5.2 summarizes the rotation and scale invariance properties of the before mentioned feature point detectors and descriptors.

### 5.3.3 Feature Point Matching

Correspondence between feature points in different frames can be done by matching the feature point descriptors. For doing so, I adopted the exhaustive *brute-force* (BF)
matching approach. So that, to find the correspondence of a query feature point descriptor, all train feature point descriptors are tested and only $f$ nearest neighbours are chosen as matching descriptors.

### 5.3.4 Outlier Removal

Feature point matching often results in mismatching. Most common type of mismatch occurs when feature points correspondence is incorrect. To remove such outliers I follow a series of steps.

Firstly, I use $f = 2$ to find two nearest descriptors for each query feature point. I adopt the known approach of ratio test to remove initial outliers. After computing these distance of two nearest neighbours from the query point, I compute the ratio of the distances. The nearest point is chosen, only if the ratio is greater than 1.5. The ratio test will remove most of the outliers.

To remove any remaining outliers, I apply the RANSAC approach\ [43]. RANSAC approach is often used in removing outliers between the corresponding points. And, due to its wide application, there have been various optimizations of this approach\ [25, 26]. This approach is particularly useful when the object is rigid, which in my case is. So,

\[ \text{Figure 5.6: Top: First rectified reference image at } k = 50. \text{ Bottom: Last rectified reference image at } k = 110 \text{ before crossing safe braking distance marker.} \]

I start by finding the rigid perspective transformation between the filtered matched train feature points and their corresponding query feature points. In case of the outliers, few corresponding points will not fit the rigid perspective transformation. For the outlier removal, RANSAC method considers various subsets of four random
corresponding points pairs. And estimates the homography matrix for these pairs. Based on the re-projection error, the outliers are removed.

5.3.5 Detector and Descriptors Matching Performance

The matching performance of each detector after outlier removal is evaluated for all descriptors. The dataset used is the collision sequence with pre-detected object at \( k = 50 \). Detected query feature points at \( k = 50 \) are matched with detected train feature points for each of the following frames up to \( k = 110 \) (maximum time to issue a timely warning).

Figure 5.6 illustrates the rectified reference images at \( k = 50 \) and \( k = 110 \), as the first and last frames of the collision sequence.

For the comparison of detector descriptor combination. I use a single detector for the whole sequence and compare the percentage of matching points in each frame for all descriptors.

It is a common practise to compare DoG with SURF detector. Also, the star detector is an extension of the DoG detector. So, I compare these three detectors first. Figure 5.7 illustrates the matching performances: top: DoG detector, middle: SURF detector, bottom: star detector. Note that for each detector, the number of query feature points in the first frame can vary. For example, the DoG detector detects 49 query feature points, the SURF detector detects 32 query feature points, and star detector detects 27 query feature points.

SIFT descriptor has very good matching performance with DoG detector and DoG detector's extension i.e. star detector. The only limitation SIFT descriptor has, is to be very slow, compared to other descriptors. Whereas, the much faster SURF descriptor, seems to be a robust descriptor, as it always produces some matches, independent of the detector being used.

For all three detectors, BRIEF descriptor has similar performance: good matching only if the scaling and camera pose differences are not significant between query feature points and train feature points. So, after 40 frames have passed, it has close to negligible matches.

Figure 5.8 illustrates the comparison of all descriptors for Fast and ORB detectors. It is important to note that, the number of query feature points for ORB detector is three times the number for FAST detector. Even with this greater number of query feature points, the matching performance of SURF and BRISK are comparable, and SURF seems to be winning more (at the end of sequence).

The plots in Fig. 5.7 and Fig. 5.8 can only show how good the descriptor is in finding the matches through the course of time. However, it fails to show the consistency of a match for a particular feature point, in the whole sequence. Consistency in the
matches is very crucial for the safety system, especially in cases when the opposing trajectory keeps on changing with time.

### 5.4 Tracker

A common limitation, that the feature point tracker has to overcome, is that the matcher does not confirm that the matched points in one frame would also be matched in the following frames. Furthermore, mismatches in positions can also occur, if the stereo correspondence algorithm fails to determine the correct disparity for the matching train feature point.

![Figure 5.7: Percentage of matched train feature points for three detectors. Top: DoG detector for 49 query feature points, Middle: SURF detector for 32 query feature points, Bottom: Star detector for 27 query feature points.](image)
5. Accuracy of Trajectory Estimation

Figure 5.8: Percentage of matched train feature points for two detectors. Top: Fast detector for 86 query feature points. Bottom: ORB detector for 255 query feature points.

Hence, to track each feature point a tracker is also needed. KF is the simplest option, but there are other advanced trackers as well like PF. However, a PF tracker needs to compute weight for large number (around 1000) hypothesis (or particles) to choose the best outcome for tracking a single feature point. Therefore, instead of using computationally intensive PF, I use KF tracker.

So that, when there is a match for a feature point, then its tracker has to be updated with the new observation (real-world position). And, when there is no match found, then the tracker has to predict feature point’s real-world position. Due to the rigid object assumption, all trackers should portray the similar real-world velocity estimate. So, during the prediction phase the neighbouring feature point trackers can assist as well.

5.4.1 Feature Point Tracking by Kalman Filter

A KF is usually defined in three steps (see Algorithm 5 for the complete object tracking algorithm):

Initialization of KF is the first step. Given the detected object in the first frame on virtual image plane of reference camera (also see Fig. 1.1). Let $n$ be the
Algorithm 5 Feature point tracker

\[
\text{collisionDecisionSystem}(f, b, d_{\text{max}}, \tau, \delta s, w, h, t_d) \quad \text{returns state } S
\]

for each Observation \( k \) in \{0, 1, \ldots \} do

if \( k = 0 \) then

Input detected object;

Detect \( n \) query feature points in this frame;

for each query Feature point \( j \) in \{1, 2, \ldots, n\} do

Initialize \((\text{KF}_j, \overrightarrow{O}_j, w_j, k_j) = \text{initializeTracker}(I_L(u, v, d))\) using Algorithm 6

end for

System state \( S \leftarrow S_0 \) (see Table 3.1);

else

for each query Feature point \( j \) in \{1, 2, \ldots, n\} do

\( \text{KF}_j \) prediction of position \( \overrightarrow{wO}_j \) and velocity \( \overrightarrow{wV}_j \);

end for

\( m = \text{FIND optimal tracker } j \) with maximum confidence criteria: \( w_j / ((k - k_j)\delta s) \);

\( l = \text{FIND } j \) with smallest predicted distance: \( ||\overrightarrow{wO}_j||\);

\( S \leftarrow \text{canWait2}(\overrightarrow{wO}_l, \overrightarrow{wV}_m, t_d, V_{\text{crit}}, \delta s, r_{\text{exc}}, \mu, g) \) using Algorithm 7

if \( S = S_3 \) then

Possible collision: issue precautionary warning as not safe to make further observations;

return

end if

if \( S = S_4 \) then

Definite collision: issue necessary warning;

return

end if

for each query Feature point \( j \) in \{1, 2, \ldots, n\} do

\((\text{KF}_j, \overrightarrow{O}_j, w_j, k_j) = \text{updateTracker}(k, w_j, k_j, \overrightarrow{O}_j, \overrightarrow{wV}_m)\) using Algorithm 8

end for

end if

end for

number of query feature points detected in this frame. Then, for \( j = \{1, 2, \ldots, n\} \),
initialize KF\textsubscript{j} with the real-world location \( \mathbf{\hat{O}}_j \) of query Feature point \( j \) at \( I_L(u,v,d) \) using Equation 3.8.

**Algorithm 6** Initialize feature point tracker

\texttt{initializeTracker}(I_L(u,v,d)) \texttt{returns} (KF\textsubscript{j}, \( \mathbf{\hat{O}}_j \), \( w_j \), \( k_j \))

For each Feature point \( j \), compute \( \mathbf{\hat{O}}_j \) using Equation 3.8.

Initialize KF\textsubscript{j} with \( \mathbf{\hat{O}}_j \).

Let \( w_j \) be the observation frequency and \( k_j \) be the last frame with successful match for \( j \). Initialize \( w_j = 0 \) and \( k_j = k \).

Algorithm 6 describes the initialization of KF for each query feature point.

**Prediction by KF** is the second step. In the following frame, feature points are matched and outliers are removed. Each KF\textsubscript{j} is asked to predict the new position \( \mathbf{\hat{O}}_j \), which after first observations is still the same \( \mathbf{\hat{O}}_j \), as there is only one observation. However, with more observations, it is based on the KF\textsubscript{j} estimated velocity \( \mathbf{\hat{V}}_j \).

Previously, in Chapter 4, I assumed that a feature point is matched in every frame, and I assigned more weight to a feature point which was observed at the same position for longer. However, on real-world data, it is not possible for the matcher to guarantee the match of every query feature point throughout the video sequence. Therefore, I assign weight \( w_j \) to a Feature point \( j \) as a counter of its matches, i.e. every time there is a match, \( w_j \) is incremented.

In a case, when the matcher fails to find the match, then optimal tracker \( m \) is chosen as the one with maximum \( w_j \). In a case, when two or more feature points having the same \( w_j \) count, then priority is given to the point with the most recent successful match. Hence, the optimal Tracker \( m \) will have maximum \( \sum w_j \), where \( k_j \) denotes the observation number for Feature point \( j \) with recent successful match.

Previously in my models, I also assumed that the ego-motion is known, while object trajectory was estimated. However, in my experiment, the object is already static, but the ego-vehicle is moving. So, instead the system uses the measurements from the optimal tracker KF\textsubscript{m} to determine the ego-vehicle velocity \( \mathbf{\hat{V}}_m \).

**Warning decision** is the intermediate step which does not affect the KF, however does affect the output of the system. The system uses \( \mathbf{\hat{V}}_m \) to compute the safe braking displacement \( \mathbf{D}_b \) in XZ directions with \( \mathbf{\hat{V}}_{\text{crit}} = [0,0,0] \) ms\(^{-1}\).

Then, the system computes the feature point \( l \) at the nearest predicted position \( \mathbf{\hat{O}}_l \). The system uses Algorithm 7 to determine whether the it has to issue a necessary warning at state \( S_4 \), or a precautionary warning at state \( S_3 \), or it can safely wait for an additional observation at state \( S_2 \).
Algorithm 7 Determine if braking warning is due

\texttt{canWait2}(\vec{wO}_r^1, \vec{wV}_r^m, t_d, V_{crit}, \delta s, r_{exc}, \mu, g) \textbf{returns} state

Object distance: $D_{cc} = ||\vec{wO}_r^1||$

\textbf{if} $D_{cc} \leq r_{exc}$ \textbf{then}
\hspace{1em} \textbf{return} S4;
\textbf{end if}

Object worst case position: $\vec{cO}^f = \vec{wO}_r^1$

Estimated vehicle velocity: $\vec{V}_i = \vec{wV}_r^m$

Vehicle braking displacement after $t_d$: $\vec{D}_b = \vec{V}_i(t_d + \delta s) + \left(\left(V_i^2 - (V_{crit}^i)^2\right) / (2\mu g)\right)$

Maximum vehicle displacement to reach $V_{crit}^i$: $\vec{O}_{safe}^i = \vec{V}_i \cdot \delta s + \vec{D}_b + [r_{exc}, 0, r_{exc}]^T$

\textbf{if} $||\vec{cO}^f|| \leq ||\vec{O}_{safe}^i||$ \textbf{then}
\hspace{1em} \textbf{return} S3
\textbf{else}
\hspace{1em} \textbf{return} S2
\textbf{end if}

Algorithm 8 Update feature point tracker

\texttt{updateTracker}(k, w_j, k_j, KF_j, \vec{O}_j^r, \vec{wV}_r^m) \textbf{returns} (KF_j, \vec{O}_j^r, w_j, k_j)

\textbf{if} match found in train feature points after outlier removal \textbf{then}
\hspace{1em} $w_j = w_j + 1$ and $k_j = k$;
\hspace{1em} Use matched train feature point to compute $\vec{O}_j^r$ using Equation 3.8 and update KF_j;
\textbf{else}
\hspace{1em} Update KF_j with $\vec{O}_j^r + \vec{wV}_r^m \cdot (k - k_j)\delta s$
\textbf{end if}

Update of KF is the final step. Each KF_j is updated based on the new matched real-world position. In a case, when the matcher fails in finding the correct match or the disparity is zero, then the optimal tracker $m$ is used on the last observation $\vec{O}_j^r$ to find the new predicted position. KF_j is updated using this predicted position (see Algorithm 8).
5. Accuracy of Trajectory Estimation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Typical value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f )</td>
<td>Focal length</td>
<td>8.9( \text{mm} )</td>
</tr>
<tr>
<td>( \tau )</td>
<td>Pixel size</td>
<td>5.01( \mu\text{m} )</td>
</tr>
<tr>
<td>( w \times h )</td>
<td>Sensor pixel resolution</td>
<td>960 \times 320 pixels</td>
</tr>
<tr>
<td>( b )</td>
<td>Baseline length</td>
<td>395.8( \text{mm} )</td>
</tr>
<tr>
<td>( d_{\text{max}} )</td>
<td>Maximum disparity</td>
<td>60</td>
</tr>
<tr>
<td>( \phi )</td>
<td>Vergence angle</td>
<td>0°</td>
</tr>
<tr>
<td>( \delta s )</td>
<td>Sampling interval</td>
<td>0.04 s</td>
</tr>
<tr>
<td>( r_{\text{exc}} )</td>
<td>Radius of vehicle exclusion zone</td>
<td>3.6m</td>
</tr>
<tr>
<td>( V^i )</td>
<td>Vehicle speed</td>
<td>11.1 ( \text{ms}^{-1} ) (40( \text{kmh} ))</td>
</tr>
<tr>
<td>( V_{\text{crit}} )</td>
<td>Maximum collision velocity</td>
<td>0 ( \text{ms}^{-1} ) (0( \text{kmh} ))</td>
</tr>
<tr>
<td>( V_{\text{limit}} )</td>
<td>Maximum speed limit</td>
<td>17( \text{ms}^{-1} ) (60( \text{kmh} ))</td>
</tr>
<tr>
<td>( s )</td>
<td>Speeding factor</td>
<td>1.5</td>
</tr>
<tr>
<td>( t_d )</td>
<td>Driver response time</td>
<td>1.5 s [14, 39]</td>
</tr>
<tr>
<td>( \mu )</td>
<td>Coefficient of friction</td>
<td>0.45 [124, 147]</td>
</tr>
<tr>
<td>( n )</td>
<td>Number of feature points</td>
<td>Detector dependant</td>
</tr>
<tr>
<td>( V_{\text{max}} )</td>
<td>Object maximum speed</td>
<td>0( \text{ms}^{-1} ) (0( \text{kmh} ))</td>
</tr>
<tr>
<td>( t_b )</td>
<td>Vehicle braking time</td>
<td>To be estimated</td>
</tr>
<tr>
<td>( D_{\text{b}} )</td>
<td>Maximum safe braking distance</td>
<td>To be estimated</td>
</tr>
</tbody>
</table>

Table 5.3: System parameters used in the model.

5.5 System Parameters

I use a collision experiment to evaluate the driver assistance system. Table 5.3 summarizes the parameters used in this experiment. I use stereo matchers iSGM and linBP for this sequence for my evaluations.

5.6 Results and Discussion

Although the system is tracking all query feature points throughout the sequence, however, it will be much easier to explain the track of a single feature point. Therefore, as was done before in Chapter 3 and Chapter 4, I choose the nearest observed query feature point \( N \) in the first frame, and visually track it throughout the sequence. The representation of tracked position is in the ego-vehicle frame of reference. Note that, even for a single track, the system will be tracking all query feature points at the back-end. And, in case there is a mismatch for point \( N \) then one of these neighbouring trackers will become an optimal tracker for \( N \) at that instance.
The tracking starts at \( k = 50 \) where the object is first detected and continues until either a warning is issued by the system or the vehicle has crossed the safe brake distance marker at \( k = 110 \). So, basically, the system has to issue a braking warning within 60 observations to be deemed as timely.

The output is in the form of a track plot in the XZ grid. Both dimensions are measured in metres with reference to the ego-vehicle. The exclusion zone for the ego-vehicle is represented by a red circle. The \( r_{exc} \) is scaled based on X-axis only.

The observed or predicted position of point \( N \) is represented by a marker. Which in general is a combination of a circle and a horizontal line. The centre of circle presents the marked position.

A green marker on the track, highlights an observation after correct feature point match and with disparity greater than 0 at that instance.

A blue marker on the track highlights, that either the matcher did not find a match at that instance or the disparity of the corresponding train feature point was zero. Thus, optimal tracker prediction is used to mark at this position.

A red marker on the track highlights, that either the matcher did not find a match at that instance or the disparity was zero. Also, the considered number of observations for all trackers are less than 7 (thus optimal tracker may have error in its estimations due to unsettled KF).

Due to integral disparities, the measured location might not change, leading to a constant predicted position, hence multiple observation markers can be drawn on top of each other.

If the system issues a braking warning before the safe braking distance i.e. 60 observations, then, the system labels the considered number of observations next to the last marker in the plot.

Previously in Section 5.3.5, I compared five detectors using five descriptors. Independent of a detector being used, I can classify that matching performance of SURF and BRISK is comparable. However, as the object became closer to the ego-vehicle, SURF seemed more robust than BRISK. Similarly, BRIEF was the worst descriptor. While, FREAK descriptor was always the in-between one. SIFT descriptor although good for DoG and star detector, isn’t good for all detectors.

Therefore, I will evaluate the KF tracker’s performance for DoG detector using the robust option i.e. SURF descriptor, the worst option i.e. BRIEF descriptor and the in-between option i.e. FREAK descriptor. As, the detector is common for each experiment, so the detected points would also be common, however the matching points may vary depending on the descriptor.

For a given detector and descriptor, I compare the tracking outputs of both iSGM and linBP based driver assistance systems in parallel. SDPS is ignored in this comparison because, iSGM stereo being an extension of SDPS is already considered.
The track plots have iSGM based output on the left while linBP based output on the right.

5.6.1 DoG Detector and SURF Descriptor

Due to the common detector and descriptor, both systems are tracking the same feature points. However, the Fig. 5.9 illustrates that even with this similarity there is a difference in types of markers and even position of markers.

![Figure 5.9: DoG detector and SURF descriptor. Left: iSGM, Right: linBP.](image)

Firstly, the position of markers is different because the disparities computed for the matched points are different for iSGM and linBP. Secondly, the presence of red markers only in iSGM’s plot suggests that, iSGM had generated zero disparities for the correct matches of \( N \) while for the same points linBP had green markers.

There is no difference in the timing of the warning, as both iSGM and linBP based systems issue timely warnings after 52 observations. Even though there were zero disparities for iSGM, in the course of tracking, still the KF can accommodate them. This shows that iSGM becomes even more robust option with a KF tracker.

5.6.2 DoG Detector and FREAK Descriptor

The Fig. 5.10 illustrates that while the ego-vehicle approaches the object, the green markers in relative frame of reference are more evenly distributed for iSGM than for linBP. This highlights that initially while \( N \) had correct feature point matches, iSGM also had better disparities than that of linBP.
5.6. Results and Discussion

Figure 5.10: DoG detector and FREAK descriptor. Left: iSGM, Right: linBP.

FREAK descriptor fails to identify any matches for point \(N\) after \(Z < 45m\). Still, in both system (see Fig. 5.10), the tracked path of \(N\) rightly changes. This change in predicted trajectory is also consistent with real ego-vehicle trajectory (see Fig. 5.1).

This shows that in the absence of matches, the other feature points had correct matches and disparities, leading to correct feedback from the optimal tracker to the point \(N\). This also shows that, while the object trajectory changes, it is important to have matches either ideally for an optimal tracker or for any feature point.

Due to the correct disparities of iSGM, its system classifies the collision scenario earlier after 55 observations, while linBP based system takes longer (59 observations), but both are timely warnings.

It was highlighted previously in Chapter 2 that the linBP stereo has the tendency to fail more often (compared to the iSGM) at disparity discontinuities. Intensity discontinuity is also likely to occur at disparity discontinuity. Since DoG detector finds feature points at intensity discontinuities, therefore, due to disparity discontinuity near detected feature points, performance of linBP is degraded more compared to iSGM.

5.6.3 DoG Detector and BRIEF Descriptor

Previously in Fig. 5.7 I showed that BRIEF descriptor matching performance degrades as the object approaches the ego-vehicle. Similarly, here Fig. 5.11 also show consistent results.
5. Accuracy of Trajectory Estimation

It is clear that the BRIEF descriptor is able to match in the initial stages. However, as the scale of object increases while the object comes closer to the ego-vehicle. Then, there are no feature point matches, neither for the point $N$ nor for the optimal tracker. Hence, the track estimated by the systems does not include the change in trajectory of ego-vehicle as it approaches the safe braking distance.

The systems are still able to issue timely warnings, but this will be an unlikely case if the object was approaching the ego-vehicle from $X - distance > r_{exc}$.

Appendix B has the tracking plots for few other detector and descriptor combinations. All using the nearest query point for the tracking plot. Note that as the detector changes, so can the nearest detected point in the first frame.

5.7 Conclusions

The number of feature points matched in each frame, cannot be used to identify the best feature point matcher for the purpose of feature point based object tracking. To have a good estimate, the same feature point has to be correctly matched frame-by-frame. Most matchers do not offer such guarantee. So, instead multiple feature points can be used for tracking through a Kalman filter. So, when there is a mismatch, an optimal tracker for any neighbouring feature point with maximum number of matches can assist in determining the next real-world position.

If the observed trajectory of object is changing, then it is important that there is at least one up-to-date optimal tracker. So that in case of a mismatch, the changing

![Figure 5.11: DoG detector and FREAK descriptor. Left: iSGM, Right: linBP.](image)
5.7. Conclusions

trajectory is reflected in the estimated trajectory as well. Similarly, for evaluation of a tracker in general, it is essential to evaluate it on a variable trajectory dataset.

I designed and tested the stereo based safety system that can issue timely warning to avoid possible collision scenario. Due to the safety limitations, I only tested it for a static object. However, the ego-vehicle was allowed to change its collision trajectory with time. The performance of the safety system varied based on the stereo matcher used. In my experiments, I found out that iSGM based estimations were far more accurate, compared to linBP. Similarly, changing the feature point descriptor for frame-to-frame feature point matching would also change the system’s performance. SURF descriptor was found to be much better than FREAK and BRIEF descriptors for the DoG detector.

It will also be very interesting to validate the system performance with a laterally moving object towards the path of the ego-vehicle. The key limitation in recording such a sequence, is the synchronization of the ego-vehicle and colliding object. Nevertheless, the safe braking distance away from the point of collision can play an important role in choreographing such a sequence, and eventually evaluating the designed driver assistance system.
This thesis has presented an overall stereo vision based driver assistance system. This system is supposed to issue braking warning to the driver in a timely fashion, allowing the driver to respond and apply brakes to avoid an imminent collision.

The first component of this system, is the stereo measurement. Cameras are aligned in canonical stereo configuration geometrically defined by ideal central projection into both image planes, which are assumed to be coplanar with epipolar lines passing through the centres of pixels in the same image row in both images (see Fig. 1.1). Rays of light drawn from the same scene, project into two different camera sensors at two different pixel locations. The difference in the pixel columns forms the disparity, which is used along with other stereo configuration parameters to derive depth of the scene. The process of finding the disparity is called stereo matching or stereo correspondence.

6.1 Conclusions from Individual Chapters

In Chapter 2, I briefly described various stereo matchers including dynamic programming, semi-global matching, and belief propagation. I also explained a few data-cost functions followed by their analysis on various datasets. Hypertree based BP with absolute difference data term has already been implemented on hardware but for a limited range of disparities; for example, \[^{[106]}\] uses 16 disparity levels using $D_{AD}$. Whereas, SGM has been implemented already on FPGA for a wider range of disparities using $D_{census}$, and is already working in real-time (at 25 Hz) in driver assistance systems of Daimler A.G. I improved the existing BP by re-implementing it on a sequential computer so that it could be tested with $D_{census}$. I named it as linBP. I compared linBP with SDPS using pixel based absolute difference data cost function and the Middlebury dataset. The findings of this experiment were that linBP was a better matcher than SDPS. Using these findings, I tested linBP matching performance for various data cost functions on a synthetic dataset and after comparison with ground truth data, I concluded that linBP has the best performance when using the census transform. Later, I compared iSGM and linBP by using the census transform.
on the KITTI stereo-vision dataset. On this dataset, iSGM was performing better than linBP (on the test-dataset). However, for the training-dataset, with a matching error count of greater than 3 pixels, linBP seemed to be a better matcher. These findings became more clear after adopting the third-eye approach in evaluating the matching performance of linBP and iSGM on EISATS’s Set 9. This dataset consists of eight different yet equally challenging sequences. Matcher linBP turned out to be a better matcher for these sequences.

In Chapter 3, I constructed a model for a stereo-vision-based driver assistance system. Instead of considering stereo matching errors, I assumed a perfect world, where the object and the ego-vehicle were moving with constant velocity on a perfectly flat surface with Y-axis = 0. Feature points on the object’s surface were also assumed to be tracked perfectly by the safety system. The only limitation that I considered was that of integral disparities, leading to discrete measurable depths. In fact, this limitation formed the source of all stereo-measurement inaccuracies. This model was totally re-configurable as all stereo-configuration parameters could be varied, along with object-collision speed, ego-vehicle velocity, ego-vehicle braking performance parameters and even driver response time. The objective of this model is to determine the maximum safest speed a colliding object can have after first appearing at a given location in front of the ego-vehicle, and yet is safely detected as colliding by the safety system and the safety system can issue a timely warning for the driver who can apply brakes and avoid it.

In Chapter 4, the model from Chapter 3 was extended by considering more general object collision trajectories. The colliding object was now assumed to be moving with constant speed, yet variable velocity. It was also assumed to be always colliding within an exclusion zone defined around the ego-vehicle. The relative rate of change of angular velocity was an additional input of this model. To compute the trajectories of variable velocity object, I updated the previously devised linear system to a weighted system, where more weight was given to observations being observed at the same position but for longer time. After comparison of linear and weighted systems for variable object velocities, I concluded that, a weighted system is much more robust option in estimating the object trajectories.

In Chapter 5, I considered real-world data. To evaluate a stereo-based safety system, one needs a collision scenario. However, in the real-world, due to the safety reasons it is difficult to record a collision sequence. Another alternate could have been to prepare a synthetic sequence. However, as it was already concluded in Chapter 2, that the performance evaluation on synthetic data may portray a wrong impression about the best matcher.

Therefore, a proper choreographed sequence was required, where the object appears to be on a collision course, and yet, an actual collision does not occur. For doing
this, I assumed a much bigger exclusion zone around the ego-vehicle of 3.6\text{m} and instead of moving the object and ego-vehicle toward each other, I only moved the ego-vehicle towards the object. This way, any unplanned synchronization problems between colliding object and ego-vehicle were resolved. Furthermore, in order to be consistent with the models, I used the tracked feature points on a presumably pre-detected object as a reference for ego-motion estimation.

Based on the estimated ego-motion, the system computed the safe braking distance, and issued a warning, when it was no longer safe for it to consider any additional observations. For the evaluation of the system, iSGM and linBP were tested as the stereo matchers. Various feature point detectors and descriptors were also used for matching feature points between frames, while a Kalman filter was used for estimating the position and velocity. Within the brief comparison of descriptors, it was found out that the SURF descriptor was most robust in issuing a timely braking warning, while iSGM was much better than linBP as a stereo matcher for estimating object trajectories. One of the reasons for that was because iSGM could generate sub-pixel-accurate disparities.

6.2 Future Work

Regarding future work, I like to work on object detection, ego-motion estimation, and the analysis of a driver-assistance systems, when both ego-vehicle and objects are moving.
Appendices

Appendix A: Additional Figures for Chapter 3

Figure 1: Tolerable speeds (in $ms^{-1}$) on the ground for a single reference point for Table 3.2 configuration parameters with $f = 9mm$, $b = 750mm$, $(w \times h) = (640 \times 480)$ and $\tau = 7.2\mu m$. 
6. Conclusions

(a) Pedestrian \((1 \times 1) \text{m}\): \(b = 750\text{mm}\), \(f = 5\text{mm}\) and \(n = 9\)

(b) Pedestrian \((1 \times 1) \text{m}\): \(b = 750\text{mm}\), \(f = 9\text{mm}\) and \(n = 9\)

Figure 2: Tolerable speeds (in \(\text{ms}^{-1}\)) on the ground for a pedestrian \((1 \times 1) \text{m}\) for Table 3.2 configuration parameters with \(f = (5\text{mm or 9mm})\), \(b = 750\text{mm}\), \((w \times h) = (640 \times 480)\), \(n = 9\) and \(\tau = 7.2\mu\text{m}\).
6.2. Future Work

(a) Vehicle\((5 \times 2) m\): \(b = 750\text{mm}, f = 5\text{mm}\) and \(n = 9\)

(b) Vehicle\((5 \times 2) m\): \(b = 750\text{mm}, f = 9\text{mm}\) and \(n = 9\)

Figure 3: Tolerable speeds (in \(ms^{-1}\)) on the ground for a vehicle \((5 \times 2) m\) for Table 3.2 configuration parameters with \(f = (5\text{mm or 9mm}), b = 750\text{mm}, (w \times h) = (640 \times 480), n = 9\) and \(\tau = 7.2\mu m\).
Figure 4: Tolerable speeds (in $ms^{-1}$) on the ground for a single reference point for Table 3.2 configuration parameters with $f = 9mm$, $b = 1000mm$, $(w \times h) = (640 \times 480)$ and $\tau = 7.2\mu m$. 
6.2. Future Work

(a) Pedestrian (1 × 1) m: \( b = 1000\text{mm}, f = 5\text{mm} \) and \( n = 9 \)

(b) Pedestrian (1 × 1) m: \( b = 1000\text{mm}, f = 9\text{mm} \) and \( n = 9 \)

Figure 5: Tolerable speeds (in \( \text{ms}^{-1} \)) on the ground for a pedestrian (1 × 1) m for Table 3.2 configuration parameters with \( f = \{5\text{mm or 9mm}\}, b = 1000\text{mm}, (w \times h) = (640 \times 480), n = 9 \) and \( \tau = 7.2\mu\text{m} \).
6. Conclusions

(a) Vehicle\( (5 \times 2) \): \( b = 1000\, mm \), \( f = 5\, mm \) and \( n = 9 \)

(b) Vehicle\( (5 \times 2) \): \( b = 1000\, mm \), \( f = 9\, mm \) and \( n = 9 \)

Figure 6: Tolerable speeds (in \( m s^{-1} \)) on the ground for a vehicle \((5 \times 2)\) for Table 3.2 configuration parameters with \( f = (5\, mm \text{ or } 9\, mm) \), \( b = 1000\, mm \), \((w \times h) = (640 \times 480)\), \( n = 9 \) and \( \tau = 7.2\, \mu m \).
Figure 7: Tolerable speeds (in $ms^{-1}$) on the ground for a single reference point for Table 3.2 configuration parameters with $f = 9\, mm$, $b = 2000\, mm$, $(w \times h) = (640 \times 480)$ and $\tau = 7.2\, \mu m$. 
6. Conclusions

(a) Pedestrian \((1 \times 1) m\): \(b = 2000 mm\), \(f = 5 mm\) and \(n = 9\)

(b) Pedestrian \((1 \times 1) m\): \(b = 2000 mm\), \(f = 9 mm\) and \(n = 9\)

Figure 8: Tolerable speeds (in \(ms^{-1}\)) on the ground for a pedestrian \((1 \times 1) m\) for Table 3.3 configuration parameters with \(f = (5 mm \ or \ 9 mm)\), \(b = 2000 mm\), \((w \times h) = (640 \times 480)\), \(n = 9\) and \(\tau = 7.2 \mu m\).
6.2. Future Work

(a) Vehicle(5 × 2) m: b = 2000 mm, f = 5 mm and n = 9

(b) Vehicle(5 × 2) m: b = 2000 mm, f = 9 mm and n = 9

Figure 9: Tolerable speeds (in m s⁻¹) on the ground for a vehicle (5 × 2) m for Table 3.2 configuration parameters with f = (5 mm or 9 mm), b = 2000 mm, (w × h) = (640 × 480), n = 9 and τ = 7.2 µm.
6. Conclusions

(a) Pedestrian (1 × 1) m: $b = 750\text{mm}$, $f = 9\text{mm}$ and $n = 9$

(b) Vehicle (5 × 2) m: $b = 750\text{mm}$, $f = 9\text{mm}$ and $n = 9$

Figure 10: Tolerable speeds (in $m s^{-1}$) on the ground for various object sizes for Table 3.2 configuration parameters with $f = 9\text{mm}$, $b = 750\text{mm}$, $(w \times h) = (1024 \times 768)$ and $\tau = 4.7\mu\text{m}$.
Figure 11: Tolerable speeds (in $m s^{-1}$) on the ground for a single reference point for Table 3.2 configuration parameters with $f = 5mm$, $b = 1000mm$, $(w \times h) = (1024 \times 768)$ and $\tau = 4.7\mu m$. 
6. Conclusions

(a) Pedestrian \((1 \times 1)\ m:\ b = 1000\ mm, f = 5\ mm\) and \(n = 9\)

(b) Pedestrian \((1 \times 1)\ m:\ b = 1000\ mm, f = 9\ mm\) and \(n = 9\)

Figure 12: Tolerable speeds (in \(ms^{-1}\)) on the ground for a pedestrian \((1 \times 1)\ m\) for Table 3.2 configuration parameters with \(f = \{5mm\ or \ 9mm\}\), \(b = 1000mm\), \((w \times h) = (1024 \times 768)\) and \(\tau = 4.7\mu m\).
6.2. Future Work

(a) Vehicle($5 \times 2$) m: $b = 1000$ mm, $f = 5$ mm and $n = 9$

(b) Vehicle($5 \times 2$) m: $b = 1000$ mm, $f = 9$ mm and $n = 9$

Figure 13: Tolerable speeds (in $m/s$) on the ground for a vehicle ($5 \times 2$) m for Table 3.2 configuration parameters with $f = (5$ mm or $9$ mm), $b = 1000$ mm, $(w \times h) = (1024 \times 768)$ and $\tau = 4.7\mu m$. 
Figure 14: Tolerable speeds (in $\text{ms}^{-1}$) on the ground for a single reference point for Table 3.2 configuration parameters with $f = 9\text{mm}$, $b = 2000\text{mm}$, $(w \times h) = (1024 \times 768)$ and $\tau = 4.7 \mu\text{m}$.
6.2. Future Work

(a) Pedestrian(1 × 1) m: \( b = 2000\, \text{mm}, f = 5\, \text{mm} \) and \( n = 9 \)

(b) Pedestrian(1 × 1) m: \( b = 2000\, \text{mm}, f = 9\, \text{mm} \) and \( n = 9 \)

Figure 15: Tolerable speeds (in ms\(^{-1}\)) on the ground for a pedestrian (1 × 1) m for Table 3.2 configuration parameters with \( f = (5\, \text{mm} \text{ or } 9\, \text{mm}), b = 2000\, \text{mm}, (w \times h) = (1024 \times 768) \) and \( \tau = 4.7\, \mu\text{m} \).
6. Conclusions

(a) Vehicle\((5 \times 2) \text{m}\): \(b = 2000 \text{mm}, f = 5 \text{mm}\) and \(n = 9\)

(b) Vehicle\((5 \times 2) \text{m}\): \(b = 2000 \text{mm}, f = 9 \text{mm}\) and \(n = 9\)

Figure 16: Tolerable speeds (in \(\text{ms}^{-1}\)) on the ground for a vehicle \((5 \times 2) \text{m}\) for Table 3.2 configuration parameters with \(f = (5 \text{mm or 9mm}), b = 2000 \text{mm}, (w \times h) = (1024 \times 768)\) and \(\tau = 4.7 \mu \text{m}\).
Figure 17: Tolerable speeds (in $m s^{-1}$) on the ground for a single reference point for Table 3.2 configuration parameters with $f = 9mm$, $b = 750mm$, $(w \times h) = (2048 \times 1152)$ and $\tau = 2.4 \mu m$. 
6. Conclusions

(a) Pedestrian (1 \times 1) m: b = 750 mm, f = 5 mm and n = 9

(b) Pedestrian (1 \times 1) m: b = 750 mm, f = 9 mm and n = 9

Figure 18: Tolerable speeds (in \text{ms}^{-1}) on the ground for a pedestrian (1 \times 1) m for Table 3.2 configuration parameters with f = (5 mm or 9 mm), b = 750 mm, (w \times h) = (2048 \times 1152) and \tau = 2.4 \mu m.
Appendix B: Additional Figures for Chapter 5

Figure 19: FAST detector, BRIEF descriptor and iSGM stereo matcher: Tracked path of the nearest query point throughout the sequence in ego-vehicle frame of reference. Distances are in m in XZ directions.
Figure 20: FAST detector, FREAK descriptor and iSGM stereo matcher: Tracked path of the nearest query point throughout the sequence in ego-vehicle frame of reference. Distances are in m in XZ directions.
Figure 21: ORB detector, BRISK descriptor and iSGM stereo matcher: Tracked path of the nearest query point throughout the sequence in ego-vehicle frame of reference. Distances are in m in XZ directions.
Co-Authored References


Non-Co-Authored References


[130] P. Wramborg. A new approach to a safe and sustainable road structure and street
[131] P. Wu, L. Kong, F. Zhao, and X. Li. Particle filter tracking based on color and SIFT
[132] Y. Yan, J. Wang, C. Li, and Z. Wu. Object tracking using SIFT features in a particle
384–388, 2011.
[133] A. Yilmaz, X. Li, and M. Shah. Contour-based object tracking with occlusion handling
in video acquired using mobile cameras. J. Pattern Analysis and Machine Intelligence
with the reduction of systematic errors. Pattern Recognition. Lett., vol. 26, no. 14, pages
correspondence. In Proc. IEEE Int. European Conf. on Computer Vision (ECCV),
[137] L. Zhang. Hierarchical block-based disparity estimation using mean absolute difference
and dynamic programming. In Proc. Int. Workshop on Very Low Bitrate Video Coding,
Bayes/MDL for multiband image segmentation. J. Pattern Analysis and Machine
Intelligence (PAMI), vol. 18, no. 9, pages 884–900, 1996.
List of Tables

2.1 Configuration parameters for my belief propagation program. No hierarchical processing, and use of the census cost function. ................................. 19
2.2 Summary of comparisons between stereo matchers on datasets. Entities used are represented by ‘y’, while the ones unused are represented by ‘n’. .... 22
2.3 Stereo matching accuracy (error rate) of SDPS and linBP for the Middlebury dataset. ................................................................. 25
2.4 linBP vs iSGM: Complete KITTI training dataset. ............................. 28
2.5 Evaluation on KITTI test images for linBP and iSGM. ......................... 29
2.6 Distance values of normalised data measures to SNCCmask. ................. 34
3.1 System states. .................................................................................. 47
3.2 System parameters used in the model. ............................................... 50
3.3 Scenarios: Highlights different parameters along with Table 3.2 configuration parameters used in the discussed scenarios. |V| = 25.5m s⁻¹ in all cases. ... 53
3.4 Point vs. Vehicle: Tolerable speeds of objects first appearing at [10, 144] m for a point object (n = 1) and a vehicle ((5 × 2)m, n = 9) for Table 3.2 configuration parameters. O(t_d + t_b) = 45m. d represents disparity. ...... 57
3.5 V_{crit}*:Tolerable speeds of a colliding vehicle first appearing at [10, 134]² for Table 3.2 configuration parameters with b = 1000mm. For V_{crit} = 10kmh Z_{safe} = 45m and for V_{crit} = 30kmh Z_{safe} = 37m. At d = 8, 9 and 10 object worst case trajectories O(t_d) in braking time for V_{crit} = 10kmh are 66m, 53m and 42m, whereas for V_{crit} = 30kmh are 84m, 71m and 60m. d represents disparity. ................................................................. 58
4.1 Types of warnings issued by a warning system. ................................. 66
5.1 Various feature point trackers. Showing the detector, descriptor, and tracker. col-histogram stands for colour histogram. ................................. 88
5.2 Summary of rotation and scale invariance. ........................................... 94
5.3 System parameters used in the model ....................................................... 102
List of Figures

1.1 Canonical stereo configuration: The decreasing depth resolution (larger $\delta Z$) for larger $Z$ values is evident [38]. The figure only shows one epipolar plane.

2.1 Left: reference image. Middle: match image. Right: overlapping reference and match images [75].

2.2 Belief propagation: $p$ influences pixels $q$ in one iteration step, and thus (indirectly) also pixels $r$ in the next iteration step [77].

2.3 Belief propagation: $pq$ message passing at time $t$.

2.4 Belief propagation: Four directions of message passing in a 4-neighbourhood.

2.5 Reference images for used stereo pairs from the Middlebury College datasets. Top left: Tsukuba. Top right: Venus. Bottom left: Teddy. Bottom right: Cones.

2.6 Columns, left to right: Ground truth disparity maps, SDPS disparity maps, linBP disparity maps. Rows, top to bottom: Tsukuba, Venus, Teddy, and Cones. Note that all disparities are scaled at the same level as the ground truth images.

2.7 Illustration of EISATS Set 2, Sequence 2. Left to right, top row: Reference image and match image. Left to right, bottom row: Ground truth disparity map and linBP disparity map.

2.8 Data cost functions: % of pixels with error $> 2$ on Sequence 2, Set 2 of EISATS.

2.9 KITTI training data example. Top: Reference image. Middle: Ground truth disparity map. Bottom: linBP disparity map.

2.10 Data cost functions: % of pixels with error $> 2$, KITTI training dataset.

2.11 linBP vs iSGM: % of pixels with error $> 2$ on KITTI training dataset.
2.12 The third-eye approach uses three cameras. Left most one forms the third image, while right two are for computing disparity map by stereo matching [94].

2.13 Functions $S_{NCC}$ and $S_{NCCmask}$ for iSGM on sequence Midday, and for linBP on sequence Dusk.

2.14 Comparison of the performance of iSGM and linBP on four of the sequences from Set 9. iSGM performs better on Bridge, but linBP wins on Dusk, Midday, and Wiper.

2.15 Comparison of NCC_mask of iSGM on set 9 of EISATS with NCC_mask-normalized functions $\sigma_{left}$, NCC_leftright, and $\sigma_{Sobel}$.

3.1 Example of a collision scene observed by the reference camera. Left image: The car on the right moves towards the path of the ego-vehicle for a potential collision. Right image: After few frames, this car is about to cause collision. Illustrated sequence is in Set 2 of EISATS.

3.2 Collision speed $V_{crit}$ varies with scenarios and the amount of protection needed for the road users [130].

3.3 Collision scenario.

3.4 Collision scenario.

3.5 Range of estimated paths of an object represented by the reference point on the object, initially at $\vec{O}_{k+1}(0)$ and the ego-vehicle first at $\vec{O}(0)$ in world co-ordinates.

3.6 Steps followed by the model to interact with the system for a colliding object travelling at a constant velocity.

3.7 Trajectory Range Narrowing: The range of possible velocities narrows down with each new observation. Note that once the disparity changes after $k = 9$, the measured velocity limits should remain consistent with previous observations. So, $\max V_z$ rightly becomes zero. Whereas, initially $\min V_z < -21$ (shown as red line), so the system instead chooses $\min V_z = -21$ (shown by green line).

3.8 Tolerable speeds (in $\text{ms}^{-1}$) on the ground for a single reference point for Table 3.2 configuration parameters with $f = 5\text{mm}$, $b = 750\text{mm}$, $(w \times h) = (1024 \times 768)$ and $\tau = 4.7\mu\text{m}$.

3.9 Tolerable speeds (in $\text{ms}^{-1}$) on the ground for a vehicle with $n = 9$ and Table 3.2 configuration parameters with $f = 5\text{mm}$, $b = 750\text{mm}$, $(w \times h) = (1024 \times 768)$ and $\tau = 4.7\mu\text{m}$.

3.10 Tolerable speeds (in $\text{ms}^{-1}$) on the ground for a pedestrian with $n = 9$ and Table 3.2 configuration parameters with $f = 5\text{mm}$, $b = 750\text{mm}$, $(w \times h) = (1024 \times 768)$ and $\tau = 4.7\mu\text{m}$.

3.11 Tolerable speeds (in $\text{ms}^{-1}$) on the ground for a single reference point for Table 3.2 configuration parameters with $f = 5\text{mm}$, $b = 750\text{mm}$, $(w \times h) = (640 \times 480)$ and $\tau = 7.2\mu\text{m}$. 
3.12 Difference between tolerable speeds if warnings are only at S4 compared to the safe situation where the timely warnings are given at (S3 or S4). Tolerable speeds are in $ms^{-1}$. .......................... 63

4.1 Collision scene observed by the reference camera. Left: Bicyclist first observed. Right: Bicyclist is very close to collide with ego-vehicle. .......................... 66

4.2 Left column shows the reference images, while the right column shows the colour-coded disparities (the streaking effects are due to a DP matcher) after background subtraction. Pose of the car is changing clockwise while observed from Y-axis. .......................... 67

4.3 Distance of the three nearest points on the toy-car to the centre of the baseline. 68

4.4 Interaction between model and weighted system. The position of the object reference point determines the object position in the model. .......................... 69

4.5 Object velocity changes with each observation. An object is on a colliding trajectory when it crosses the exclusion zone which is indicated by a circle of radius $r_{exc}$. .......................... 70

4.6 Collision trajectory: the trajectory direction of an object varies with each Observation $k$. Four locations A, B, C and D show where the object changes its trajectory. The path and pose of the object at each instance is different relative to the ego-vehicle. The object collides in time $t_c$. .......................... 71

4.7 Range of relative speeds estimated by a linear system for an object first appearing at A1 from the ego-vehicle. (a) lateral speeds, (b) longitudinal speeds. .......................... 77

4.8 Range of relative speeds estimated by a weighted system for an object first appearing at A1 from the ego-vehicle. (a) lateral speeds, (b) longitudinal speeds. .......................... 78

4.9 Tolerable speeds (in $ms^{-1}$) of a linear and a weighted system for a collision scenario. Parameters not specified in captions are taken from Table 3.2 configuration parameters. The dark-blue region within stereo CFoV has tolerable speeds less than or equal to $V_{crit} = 2.7 ms^{-1}$. .......................... 81

5.1 Choreographed sequence. The observer is a person holding a flag. The flag is raised after the ego-vehicle, which is on a collision course, crosses the marker on the road. The marker on the road is at a safe braking distance from the colliding object. Bottom image: The iSGM disparity map for the stereo pair illustrated by one frame in the top image. .......................... 85

5.2 Feature point tracking on the reference images. .......................... 87

5.3 Example of rotation and scale invariant feature from two different views of the same scene [85]. .......................... 89

5.5 The darker region has the weight -1, while lighter region has the weight of +1.

Left image: Filter to compute response in u direction. Right image: Filter to compute response in v direction. ........................................ 93

5.6 Top: First rectified reference image at k = 50. Bottom: Last rectified reference image at k = 110 before crossing safe braking distance marker. ......... 95


5.8 Percentage of matched train feature points for two detectors. Top: Fast detector for 86 query feature points. Bottom: ORB detector for 255 query feature points. ......................................................... 98

5.9 DoG detector and SURF descriptor. Left: iSGM, Right: linBP. ............. 104

5.10 DoG detector and FREAK descriptor. Left: iSGM, Right: linBP. ........ 105

5.11 DoG detector and FREAK descriptor. Left: iSGM, Right: linBP. ........ 106

1 Tolerable speeds (in $ms^{-1}$) on the ground for a single reference point for Table 3.2 configuration parameters with $f = 9mm$, $b = 750mm$, $(w \times h) = (640 \times 480)$ and $\tau = 7.2\mu m$. ...................................................... 113

2 Tolerable speeds (in $ms^{-1}$) on the ground for a pedestrian (1 x 1) m for Table 3.2 configuration parameters with $f = (5mm \ or \ 9mm)$, $b = 750mm$, $(w \times h) = (640 \times 480)$, $n = 9$ and $\tau = 7.2\mu m$. ...................................................... 114

3 Tolerable speeds (in $ms^{-1}$) on the ground for a vehicle (5 x 2) m for Table 3.2 configuration parameters with $f = (5mm \ or \ 9mm)$, $b = 750mm$, $(w \times h) = (640 \times 480)$, $n = 9$ and $\tau = 7.2\mu m$. ...................................................... 115

4 Tolerable speeds (in $ms^{-1}$) on the ground for a single reference point for Table 3.2 configuration parameters with $f = 9mm$, $b = 1000mm$, $(w \times h) = (640 \times 480)$ and $\tau = 7.2\mu m$. ...................................................... 116

5 Tolerable speeds (in $ms^{-1}$) on the ground for a pedestrian (1 x 1) m for Table 3.2 configuration parameters with $f = (5mm \ or \ 9mm)$, $b = 1000mm$, $(w \times h) = (640 \times 480)$, $n = 9$ and $\tau = 7.2\mu m$. ...................................................... 117

6 Tolerable speeds (in $ms^{-1}$) on the ground for a vehicle (5 x 2) m for Table 3.2 configuration parameters with $f = (5mm \ or \ 9mm)$, $b = 1000mm$, $(w \times h) = (640 \times 480)$, $n = 9$ and $\tau = 7.2\mu m$. ...................................................... 118

7 Tolerable speeds (in $ms^{-1}$) on the ground for a single reference point for Table 3.2 configuration parameters with $f = 9mm$, $b = 2000mm$, $(w \times h) = (640 \times 480)$ and $\tau = 7.2\mu m$. ...................................................... 119

8 Tolerable speeds (in $ms^{-1}$) on the ground for a pedestrian (1 x 1) m for Table 3.2 configuration parameters with $f = (5mm \ or \ 9mm)$, $b = 2000mm$, $(w \times h) = (640 \times 480)$, $n = 9$ and $\tau = 7.2\mu m$. ...................................................... 120

9 Tolerable speeds (in $ms^{-1}$) on the ground for a vehicle (5 x 2) m for Table 3.2 configuration parameters with $f = (5mm \ or \ 9mm)$, $b = 2000mm$, $(w \times h) = (640 \times 480)$, $n = 9$ and $\tau = 7.2\mu m$. ...................................................... 121
Tolerable speeds (in $ms^{-1}$) on the ground for various object sizes for Table 3.2 configuration parameters with $f = 9mm$, $b = 750mm$, $(w \times h) = (1024 \times 768)$ and $\tau = 4.7\mu m$. . . . . . . . . . . . . . . . . . . . . . 122

Tolerable speeds (in $ms^{-1}$) on the ground for a single reference point for Table 3.2 configuration parameters with $f = 5mm$, $b = 1000mm$, $(w \times h) = (1024 \times 768)$ and $\tau = 4.7\mu m$. . . . . . . . . . . . . . . . . . . . . . 123

Tolerable speeds (in $ms^{-1}$) on the ground for a pedestrian $(1 \times 1)m$ for Table 3.2 configuration parameters with $f = 5mm$ or $9mm$, $b = 2000mm$, $(w \times h) = (1024 \times 768)$ and $\tau = 4.7\mu m$. . . . . . . . . . . . . . . . . . . . . . 124

Tolerable speeds (in $ms^{-1}$) on the ground for a vehicle $(5 \times 2)m$ for Table 3.2 configuration parameters with $f = 5mm$ or $9mm$, $b = 1000mm$, $(w \times h) = (1024 \times 768)$ and $\tau = 4.7\mu m$. . . . . . . . . . . . . . . . . . . . . . 125

Tolerable speeds (in $ms^{-1}$) on the ground for a pedestrian $(1 \times 1)m$ for Table 3.2 configuration parameters with $f = 9mm$, $b = 2000mm$, $(w \times h) = (2048 \times 1152)$ and $\tau = 2.4\mu m$. . . . . . . . . . . . . . . . . . . . . . 126

Tolerable speeds (in $ms^{-1}$) on the ground for a single reference point for Table 3.2 configuration parameters with $f = 9mm$, $b = 750mm$, $(w \times h) = (2048 \times 1152)$ and $\tau = 2.4\mu m$. . . . . . . . . . . . . . . . . . . . . . 127

Tolerable speeds (in $ms^{-1}$) on the ground for a pedestrian $(1 \times 1)m$ for Table 3.2 configuration parameters with $f = 9mm$, $b = 750mm$, $(w \times h) = (2048 \times 1152)$ and $\tau = 2.4\mu m$. . . . . . . . . . . . . . . . . . . . . . 128

Tolerable speeds (in $ms^{-1}$) on the ground for a single reference point for Table 3.2 configuration parameters with $f = 9mm$, $b = 750mm$, $(w \times h) = (2048 \times 1152)$ and $\tau = 2.4\mu m$. . . . . . . . . . . . . . . . . . . . . . 129

Tolerable speeds (in $ms^{-1}$) on the ground for a pedestrian $(1 \times 1)m$ for Table 3.2 configuration parameters with $f = 9mm$, $b = 750mm$, $(w \times h) = (2048 \times 1152)$ and $\tau = 2.4\mu m$. . . . . . . . . . . . . . . . . . . . . . 130

FAST detector, BRIEF descriptor and iSGM stereo matcher: Tracked path of the nearest query point throughout the sequence in ego-vehicle frame of reference. Distances are in m in XZ directions. . . . . . . . . . . . . . . . . . . . . . 131

FAST detector, FREAK descriptor and iSGM stereo matcher: Tracked path of the nearest query point throughout the sequence in ego-vehicle frame of reference. Distances are in m in XZ directions. . . . . . . . . . . . . . . . . . . . . . 132

ORB detector, BRISK descriptor and iSGM stereo matcher: Tracked path of the nearest query point throughout the sequence in ego-vehicle frame of reference. Distances are in m in XZ directions. . . . . . . . . . . . . . . . . . . . . . 133
List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_L$</td>
<td>Left image or reference image or base image</td>
</tr>
<tr>
<td>$I_R$</td>
<td>Right image or match image</td>
</tr>
<tr>
<td>$w$</td>
<td>Image width in pixels</td>
</tr>
<tr>
<td>$h$</td>
<td>Image height in pixels</td>
</tr>
<tr>
<td>$d$</td>
<td>Disparity of a feature point</td>
</tr>
<tr>
<td>$u$</td>
<td>Pixel column number</td>
</tr>
<tr>
<td>$v$</td>
<td>Pixel row number</td>
</tr>
<tr>
<td>$L \times H \times W$</td>
<td>Object size</td>
</tr>
<tr>
<td>$O'(0)$</td>
<td>Ego-vehicle position at Observation 0</td>
</tr>
<tr>
<td>$\vec{V}$</td>
<td>Ego-vehicle velocity</td>
</tr>
<tr>
<td>$r_{exc}$</td>
<td>Radius of exclusion zone around ego-vehicle</td>
</tr>
<tr>
<td>$Z_{axis}$</td>
<td>Parallel to optical axes of stereo cameras</td>
</tr>
<tr>
<td>$Y_{axis}$</td>
<td>Perpendicular to both baseline – line between the optical centres of stereo cameras and optical axes</td>
</tr>
<tr>
<td>$V_{exit}$</td>
<td>Maximum tolerable collision speed</td>
</tr>
<tr>
<td>$\vec{V}$</td>
<td>Object real-world velocity</td>
</tr>
<tr>
<td>$O(0)$</td>
<td>Object position at Observation 0</td>
</tr>
<tr>
<td>$t$</td>
<td>Time since the first observation</td>
</tr>
<tr>
<td>$O(t)$</td>
<td>Object’s position after Time $t$</td>
</tr>
<tr>
<td>$\eta$</td>
<td>Object’s trajectory angle relative to the $X$-axis</td>
</tr>
<tr>
<td>$X_{axis}$</td>
<td>Parallel to the baseline – line between optical centres of stereo cameras</td>
</tr>
<tr>
<td>$n$</td>
<td>Number of feature points on the object’s surface at $k = 0$</td>
</tr>
<tr>
<td>$t_{cross}$</td>
<td>Time object takes to enter the ego-vehicle’s path since the first observation</td>
</tr>
<tr>
<td>$O(t_{cross})$</td>
<td>Object’s position after Time $t_{cross}$ relative to $O'(0)$</td>
</tr>
<tr>
<td>$Z_{cross}$</td>
<td>Object’s Z-position after Time $t_{cross}$ relative to $O'_i(0)$</td>
</tr>
<tr>
<td>$t_{leave}$</td>
<td>Time object takes to leave the ego-vehicle’s path since the first observation</td>
</tr>
<tr>
<td>$Z_{leave}$</td>
<td>Object’s Z-position after Time $t_{leave}$ relative to $O'_i(0)$</td>
</tr>
<tr>
<td>$O'(t_{cross})$</td>
<td>Ego-vehicle’s position after Time $t_{cross}$ relative to $O'(0)$</td>
</tr>
</tbody>
</table>
**List of Figures**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_c$</td>
<td>Distance between ego-vehicle and object at the time object first enters the ego-vehicle’s path.</td>
</tr>
<tr>
<td>$t_b$</td>
<td>Time taken by the ego-vehicle to reach $V_{v,\text{crit}}$ after braking.</td>
</tr>
<tr>
<td>$t_d$</td>
<td>Driver response time to apply brakes after warning.</td>
</tr>
<tr>
<td>$t_{\text{warn}}$</td>
<td>Maximum time to issue a timely warning.</td>
</tr>
<tr>
<td>$r$</td>
<td>Superscript to represent ego-vehicle frame of reference.</td>
</tr>
<tr>
<td>$w \times h$</td>
<td>Dimensions of stereo image pair as columns $\times$ rows.</td>
</tr>
<tr>
<td>$b$</td>
<td>Baseline length between the stereo cameras.</td>
</tr>
<tr>
<td>$f$</td>
<td>Focal length of the stereo camera lens.</td>
</tr>
<tr>
<td>$\phi$</td>
<td>Verge angle between the stereo cameras.</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Square pixel size of the stereo cameras.</td>
</tr>
<tr>
<td>$\tau_h$</td>
<td>Horizontal pixel size of the stereo cameras.</td>
</tr>
<tr>
<td>$\tau_v$</td>
<td>Vertical pixel size of the stereo cameras.</td>
</tr>
<tr>
<td>$\zeta$</td>
<td>Relative real collision angle in polar co-ordinates.</td>
</tr>
<tr>
<td>$\theta$</td>
<td>Half angle of the stereo common field of view.</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Maximum estimated velocity in the ego-vehicle frame of reference.</td>
</tr>
<tr>
<td>$\rho_{\text{max}}$</td>
<td>Maximum estimated velocity in the ego-vehicle frame of reference.</td>
</tr>
<tr>
<td>$\rho_t$</td>
<td>Minimum estimated trajectory angle in polar co-ordinates.</td>
</tr>
<tr>
<td>$\rho_{\text{max}}$</td>
<td>Maximum estimated trajectory angle in polar co-ordinates.</td>
</tr>
<tr>
<td>$\zeta_t$</td>
<td>Minimum tangent angle to exclusion zone in polar co-ordinates.</td>
</tr>
<tr>
<td>$\zeta_n$</td>
<td>Maximum tangent angle to exclusion zone in polar co-ordinates.</td>
</tr>
<tr>
<td>$S$</td>
<td>System state.</td>
</tr>
<tr>
<td>$jx$</td>
<td>Feature point $j$ with smallest X-distance.</td>
</tr>
<tr>
<td>$\zeta_{\text{L}}$</td>
<td>Array of left tangents for all feature points to the exclusion zone.</td>
</tr>
<tr>
<td>$\zeta_{\text{R}}$</td>
<td>Array of right tangents for all feature points to the exclusion zone.</td>
</tr>
<tr>
<td>$j_z$</td>
<td>Feature point $j$ with smallest Z-distance.</td>
</tr>
<tr>
<td>$cO$</td>
<td>Object position after braking in ego-vehicle frame of reference.</td>
</tr>
<tr>
<td>$Z_{\text{safe}}$</td>
<td>Maximum ego-vehicle Z-distance to decelerate to $V_{v,\text{crit}}$.</td>
</tr>
<tr>
<td>$\delta c$</td>
<td>Rate of change of object relative angular velocity.</td>
</tr>
<tr>
<td>$\zeta_r$</td>
<td>Real collision angle in polar co-ordinates, just before Observation $k$.</td>
</tr>
<tr>
<td>$D'(k)$</td>
<td>Object distance from the ego-vehicle, just before Observation $k$.</td>
</tr>
</tbody>
</table>
List of Figures

- $\min_{k} \mathbf{V}_r^j$: Minimum relative weighted velocity of a Feature point $j$ at Observation $k$  
- $\max_{k} \mathbf{V}_r^j$: Maximum relative weighted velocity of a Feature point $j$ at Observation $k$  
- $\mathbf{O}_j$: Measured position of matched Feature point $j$ from the ego-vehicle  
- $\mathbf{K}_F^j$: Kalman filter based tracker for query Feature point $j$  
- $\mathbf{wO}_j^r$: Kalman filter based relative predicted position of query Feature point $j$  
- $\mathbf{wV}_r^j$: Kalman filter based relative predicted velocity of query Feature point $j$  
- $D_b$: Braking displacement  
- $V_{crit}^r$: Maximum tolerable collision velocity  
- $w_j$: Number of times query Feature point $j$ is matched  
- $k_j$: Frame number at which query Feature point $j$ was last matched  
- $m$: Optimal tracker  
- $l$: Nearest predicted query feature point from ego-vehicle
Index

CFOV, 41
ABS, 4
ACC, 4
accuracy matching, 22
braking time, 41, 43
coefficient of friction, 43
collision, 41
collision trajectory, 41
Common Field of View, 41
depth, 4
descriptor matchers
  BF, 84
  optical flow, 88
  RANSAC, 88 95
descriptors
  BRIEF, 83
  BRISK, 84
  FREAK, 94
  SIFT, 88 92
  SURF, 88 92
detectors
  CenSurE, 91
  DoG, 88 90
  FAST, 90
  LoG, 91
  NCC, 88
ORB, 91
  Star, 91
  STK, 88
  SURF, 88 90
disparity, 5 12 14
driver response time, 11 12 30 102
description
  ego-vehicle, 6 37
  energy, 12
  exclusion zone, 39
feature point, 6
geometry
  epipolar, 4
  ground truth, 22
image
  match, 12
  reference, 12
  iSGM, 66
KITTI, 29
laser, 4
maximum tolerable collision speed, 39
measure
  similarity, 39
measurements
  X measurement error, 44
collision trajectory, 46 47
depth resolution, 43
disparity, 43
errors in trajectory estimates, 44
feature point, 6
query feature points, 87
region of uncertainty, 44
relative velocity, 44
tangents to ego-vehicle’s exclusion zone, 46
train feature points, 87
trajectory angles, 45, 47
uncertainty area, 44
uncertainty in depth, 43
NCC, 30
object, 6
OpenCV, 87
pixels
    corresponding, 4
RADAR, 4
rigid object, 45
rotation invariant, 89
safe braking distance, 50, 102
safe speed, 49
scale invariant, 89
SDPS, 15
speeding factor, 45
states, 44
stereo
    Belief propagation, 13
    BP, 13
disparity, 5
    DP, 13
dynamic programming, 13
    GC, 13
    iSGM, 13
    linBP, 13
linear BP stereo, 13
    SDPS, 13
    Semi-Global Matching, 13
Symmetric Dynamic Programming stereo, 13
stereo configuration
    canonical, 4
stereo photogrammetry, 5
term
    data, 12
    smoothness, 12
third-eye approach, 22, 29
timely warning, 41
timing of warning, 11
tracker
    EKF, 88
    KF, 88, 98
    PF, 86, 88, 98
    UKF, 88
trajectory angle, 49
TTC, 4
vehicle braking time, 50, 102