How Good Is My Test Data? Introducing Safety Analysis for Computer Vision

Oliver Zendel¹ · Markus Murschitz¹ · Martin Humenberger¹ · Wolfgang Herzner¹

Received: 13 April 2016 / Accepted: 18 May 2017 / Published online: 9 June 2017
© The Author(s) 2017. This article is an open access publication

Abstract Good test data is crucial for driving new developments in computer vision (CV), but two questions remain unanswered: which situations should be covered by the test data, and how much testing is enough to reach a conclusion? In this paper we propose a new answer to these questions using a standard procedure devised by the safety community to validate complex systems: the hazard and operability analysis (HAZOP). It is designed to systematically identify possible causes of system failure or performance loss. We introduce a generic CV model that creates the basis for the hazard analysis and—for the first time—apply an extensive HAZOP to the CV domain. The result is a publicly available checklist with more than 900 identified individual hazards. This checklist can be utilized to evaluate existing test datasets by quantifying the covered hazards. We evaluate our approach by first analyzing and annotating the popular stereo vision test datasets Middlebury and KITTI. Second, we demonstrate a clearly negative influence of the hazards in the checklist on the performance of six popular stereo matching algorithms. The presented approach is a useful tool to evaluate and improve test datasets and creates a common basis for future dataset designs.

Keywords Test data · Testing · Validation · Safety analysis · Hazard analysis · Stereo vision

1 Introduction

Many safety-critical systems depend on CV technologies to navigate or manipulate their environment and require a thorough safety assessment due to the evident risk to human lives (Matthias et al. 2010). The most common software safety assessment method is testing on pre-collected datasets. People working in the field of CV often notice that algorithms scoring high in public benchmarks perform rather poor in real world scenarios. It is easy to see why this happens:

1. The limited information present in these finite samples can only be an approximation of the real world. Thus we cannot expect that an algorithm which performs well under these limited conditions will necessarily perform well for the open real-world problem.
2. Testing in CV is usually one-sided: while every new algorithm is evaluated based on benchmark datasets, the datasets themselves rarely have to undergo independent evaluation. This is a serious omission as the quality of the tested application is directly linked to the quality and extent of test data. Sets with lots of gaps and redundancy will match poorly to actual real-world challenges. Tests conducted using weak test data will result in weak conclusions.

This work presents a new way to facilitate a safety assessment process to overcome these problems: a standard method developed by the safety community is applied to the CV domain for the first time. It introduces an independent measure to enumerate the challenges in a dataset for testing the robustness of CV algorithms.

The typical software quality assurance process uses two steps to provide objective evidence that a given system fulfills its requirements: verification and validation

Communicated by Rene Vidal, Katsushi Ikeuchi, Josef Sivic, Christoph Schnoerr.

✉ Oliver Zendel
oliver.zendel@ait.ac.at

¹ AIT Austrian Institute of Technology, Donau-City-Strasse 1, 1220 Vienna, Austria

(International Electrotechnical Commission 2010). Verification checks whether or not the specification was implemented correctly (i.e. no bugs) (Department of Defense 1991). Validation addresses the question whether or not the algorithm is appropriate for the intended use, i.e., is robust enough under difficult circumstances. Validation is performed by comparing the algorithm's output against the expected results (*ground truth*, *GT*) on test datasets. Thus, the intent of validation is to find shortcomings and poor performance by using "difficult" test cases (Schlick et al. 2011). While general methods for verification can be applied to CV algorithms, the validation step is rather specific. A big problem when validating CV algorithms is the enormous set of possible test images. Even for a small 8bit monochrome image of 640×480 pixels, there are already $256^{640 \times 480} \approx 10^{739811}$ possible images). Even if many of these combinations are either noise or render images which are no valid sensor output, exhaustive testing is still not feasible for CV. An effective way to overcome this problem is to find equivalence classes and to test the system with a representative of each class. Defining equivalence classes for CV is an unsolved problem: how does one describe in mathematical terms all possible images that show for example "a tree" or "not a car"? Thus, mathematical terms do not seem to be reasonable but the equivalence classes for images are still hard to define even if we stick to the semantic level. A systematic organization of elements critical to the CV domain is needed and this work will present our approach to supply this.

All in all, the main challenges for CV validation are:

1. What should be part of the test dataset to ensure that the required level of robustness is achieved?
2. How can redundancies be reduced (to save time and remove bias due to repeated elements)?

Traditional benchmarking tries to characterize performance on fixed datasets to create a ranking of multiple implementations. On the contrary, validation tries to show that the algorithm can reliably solve the task at hand, even under difficult conditions. Although both use application specific datasets, their goals are different and benchmarking sets are not suited for validation.

The main challenge for validation in CV is listing elements and relations which are known to be "difficult" for CV algorithms (comparable to optical illusions for humans). In this paper, the term *visual hazard* will refer to such elements and specific relations (see Fig. 1 for examples).

By creating an exhaustive checklist of these visual hazards we meet the above challenges:

1. Ensure completeness of test datasets by including all relevant hazards from the list.

Fig. 1 Examples for potential visual hazards for CV algorithms

2. Reduce redundancies by excluding test data that only contains hazards that are already identified.

Our main contributions presented in this paper are:

- application of the HAZOP risk assessment method to the CV domain (Sect. 3),
- introduction of a generic CV system model useful for risk analysis (Sect. 3.1),
- a publicly available hazard checklist (Sect. 3.7) and a guideline for using this checklist as a tool to measure hazard coverage of test datasets (Sec. 4).

To evaluate our approach, the guideline is applied to three stereo vision test datasets: KITTI, Middlebury 2006 and Middlebury 2014 (see Sect. 5). As a specific example, the impact of identified hazards on the output of multiple stereo vision algorithms is compared in Sect. 6.

2 Related Work

Bowyer and Phillips (1998) analyze the problems related to validating CV systems and propose that the use of sophisticated mathematics goes hand in hand with specific assumptions about the application. If those assumptions are not correct, the actual output in real-world scenarios will deviate from the expected output.

Ponce et al. (2006) analyze existing image classification test datasets and report a strong database bias. Typical poses and orientations as well as lack of clutter create an unbalanced training set for a classifier that should work robustly in real-world applications.

Pinto et al. (2008) demonstrate by a neuronal net, used for object recognition, that the currently used test datasets are significantly biased. Torralba and Efros (2011) successfully train image classifiers to identify the test dataset itself (not its content), thus, showing the strong bias each individual dataset contains.

A very popular CV evaluation platform is dedicated to stereo matching, the Middlebury stereo database. [Scharstein and Szeliski \(2002\)](#) developed an online evaluation platform which provides stereo datasets consisting of the image pair and the corresponding GT data. The datasets show indoor scenes and GT are created with a structured light approach ([Scharstein and Szeliski 2003](#)). Recently, an updated and enhanced version was presented which includes more challenging datasets as well as a new evaluation method ([Scharstein et al. 2014](#)). To provide a similar evaluation platform for road scenes, the KITTI database was introduced by ([Geiger et al. 2012](#)).

A general overview of CV performance evaluation can be found in ([Thacker et al. 2008](#)). They summarize and categorize the current techniques for performance validation of algorithms in different subfields of CV. Some examples are shown in the following: [Bowyer et al. \(2001\)](#) present a work for edge detection evaluation based on receiver operator characteristics (ROCs) curves for 11 different edge detectors. [Min et al. \(2004\)](#) describe an automatic evaluation framework for range image segmentation which can be generalized to the broader field of region segmentation algorithms. In [Kondermann \(2013\)](#) the general principles and types of ground truth are summarized. They pointed out, that thorough engineering of requirements is the first step to determine which kind of ground truth is required for a given task. [Strecha et al. \(2008\)](#) present a multi-view stereo evaluation dataset that allows evaluation of pose estimation and multi-view stereo with and without camera calibration. They additionally incorporate GT quality in their LIDAR-based method to enable fair comparisons between benchmark results. [Kondermann et al. \(2015\)](#) discuss the effect of GT quality on evaluation and propose a method to add error bars to disparity GT. [Honauer et al. \(2015\)](#) reveal stereo algorithm-specific strengths and weaknesses through new evaluation metrics addressing depth discontinuities, planar surfaces, and fine geometric structures. All of these are examples of visual hazards.

Current test datasets neither provide clear information about which challenges are covered nor which issues remain uncovered. Our approach can fill both gaps: By assigning a reference-table entry with a unique identifier to each challenging hazard, we create a checklist applicable to any dataset. To the best knowledge of the authors there is no published work considering the vision application as a whole, which identifies risks on such a generic level.

2.1 Robustness

Depending on context, *robustness* can refer to different characteristics of the considered system. In the safety context, robustness is about the correct handling of abnormal situations or input data. For instance, in the basic standard

for functional safety ([International Electrotechnical Commission 2010](#)), it is defined via the system's behavior in a hazardous situation or hazardous event. This also includes the ability to perform a required function in the presence of implementation faults (internal sources) or cope with faulty and noisy input data (external sources). A method to evaluate robustness against the first type is fault injection ([Hampel 1971](#)) (e.g., bit flips in registers or bus) while fuzz testing ([Takanen et al. 2008](#)) can be used for assessing the robustness against abnormal input data.

In computer vision, robustness usually refers to coping with distorted or low-quality input. Popular methods are random sample consensus (RANSAC) ([Fischler and Bolles 1981](#)), M-Estimators ([Huber 1964](#)), or specific noise modeling techniques, which arose from the need to use systems in “real-world applications”. In the work described in this paper, we do not exclude these issues, but aim to cover all influences that may cause a degraded or false performance of a CV solution. This in particular includes aspects that can usually be part of observed scenes, such as lacking or highly regular textures, reflections, occlusions, or low contrasts. Figure 1 illustrates some examples.

2.2 Risk Analysis

Risk-oriented analysis methods are a subset of validation and verification methods. All technical risk analysis methods assess one or several risk-related attributes (e.g. safety or reliability) of systems, components or even processes with respect to causes and consequences. Some techniques additionally try to identify existing risk reduction measures and propose additional measures where necessary.

Originally, risk identification techniques have been developed by the chemical industries, but nowadays they are successfully applied to software quality assurance as well (see [Fenelon and Hebborn 1994](#) and [Goseva-Popstojanova et al. 2003](#) for UML models). The most commonly used methods are:

- HAZOP [7], ([Kletz 1983](#))—hazard and operability analysis,
- FME(C)A ([Department of Defense 1949](#))—failure modes, effects, (and criticality) analysis,
- FTA ([Vesely et al. 1981](#); [Laprie 1992](#))—fault tree analysis.

Each risk analysis method defines a systematic process to identify potential risks. The first step in a HAZOP is to identify the essential components of the system to be analyzed. The parameters for each component, which define its behavior, have to be identified. These parameters often describe the input output characteristics of the component. A set of predefined guide words which describe deviations

are applied to the parameters (e.g. “less” or “other than”) and the resulting combinations are interpreted by experts in order to identify possible consequences (potential hazards) and counteractions. While FME(C)A also starts with identifying the systems components and their operating modes, it then identifies the potential failure modes of the individual components. Further steps deal with identifying potential effects of these failures, their probability of occurrence, and risk reduction measures similar to HAZOP. FTA starts with a hazardous “top event” as root of the fault tree. Leaves are added recursively to the bottom events representing Boolean combinations which contain possible causes for their parent event (e.g. “own car hits the front car” if “speed too high” and “braking insufficient”). This refinement is executed until only elementary events are encountered.

3 CV-HAZOP

The identification and collection of CV hazards should follow a systematic manner and the results should be applicable to many CV solutions. The process has to be in line with well-established practices from the risk and safety assessment community to create an accepted tool for validation of CV systems. The most generic method HAZOP (Kletz 1983) is chosen over FME(C)A and FTA because it is feasible for systems for which little initial knowledge is available. In addition, the concept of guide words adds a strong source of inspiration that all other concepts are missing. The following Sections address the main steps of a HAZOP:

1. Model the system.
2. Partition the model into subcomponents, called locations.
3. Find appropriate parameters for each location which describe its configuration.
4. Define useful guide words.
5. Assign meanings for each guide word/parameter combination and derive consequences from each meaning.
6. Give an example clarifying the entry using for a specific application (e.g. in the context of stereo vision, object tracking, face detection).

3.1 Generic Model

The first step of any HAZOP is deriving a model of the system that should be investigated. In case of this HAZOP, the generic CV algorithm has to be modeled together with the observable world (its application). Marr (1982) proposes a model for vision and image perception from the human perception perspective. Aloimonos and Shulman (1989) extended it by the important concepts of stability and robustness. We propose a novel model which is entirely based on the idea of information flow: The common goal of all CV

algorithms is the extraction of information from image data. Therefore “information” is chosen to be the central aspect handled by the system. It should be noted, that “information” is used in the context “Information is data which has been assigned a meaning.” Van der Spek and Spijkervet (1997) rather than in a strict mathematical sense (Shannon and Weaver 1949). In this context, *hazards are all circumstances and relations that cause a loss of information*. Even though hazards ultimately propagate to manifest themselves in the output of the algorithm, an effective way to find a feasible list of hazards is to look at the entire system and attribute the hazard to the location where it first occurred (e.g. unexpected scene configuration or sensor errors). Multiple inputs from different disciplines are used to create the system model:

Information Theory Communication can be abstracted according to information theory (Shannon and Weaver 1949) as information flow from the transmitter at the source—with the addition of noise—to the receiver at the destination.

Sampling Theorem Sampling is a key process in the course of transforming reality into discrete data. Artifacts that can be caused by this process, according to (Nyquist 1928; Shannon 1949), will result in a loss of information.

Rendering Equation The rendering equation (Kajiya 1986) is a formal description of the process of simulating the output of a virtual camera within a virtual environment. The different parts of the standard rendering equation amount to the different influences that arise when projecting a scenery light distribution into a virtual camera.

Control Theory The general system theory (e.g. Von Bertalanffy 1968) and especially cybernetics interpret and model the interactions of systems and the steps of acquiring, processing, as well as reacting to information from the environment.

The entire flow of information is modeled as follows:

1. Since in CV the sensor is a camera, all data within the observed scene available to a CV component can only be provided by the electromagnetic spectrum (simply referred to as *light* in this paper) received by the *observer* (i.e. the sensor/camera) from any point in the scene. Hence, light represents data and, as soon as a meaning is assigned, information.
2. At the same time, any unexpected generation of light and unwanted interaction of light with the scene distorts and reduces this information.
3. The sensing process, i.e. the transformation of received light into digital data, further reduces and distorts the information carried by the received light.
4. Finally, the processing of this data by the CV algorithm also reduces or distorts information (through rounding errors, integration etc.).

Fig. 2 Information flow within the generic model. Light information is shown as *solid arrows* and digital information as a *dashed arrow*

In essence, two information carriers are distinguished: light outside of the system under test (SUT) and digital data within the SUT. This is visualized in Fig. 2 by two types of arrows: solid arrows for light and a dashed line for digital data. At each transition, this information can potentially be distorted (e.g. by reduction, erasure, transformation, and blending). Benign interactions, e.g. interaction of a pattern specifically projected to create texture for structured light applications, are not entered into the list. We are interested in situations and aspects that can potentially reduce output quality. Nevertheless, the failing of such expected benign interactions (e.g. inference effects of multiple projectors) are a risk and, thus, included in the analysis.

3.2 Locations

The system model is now partitioned into specific locations (i.e. subsystems) of the overall system. Light sources that provide illumination start the process flow (illustrated in Fig. 2). The light traverses through a medium until it either reaches the observer or interacts with objects. This subprocess is recursive and multiple interactions of light with multiple objects are possible. The observer is a combination of optical systems, the sensor, and data pre-processing. Here the light information is converted into digital data as input for a CV algorithm. The CV algorithm processes the data to extract information from it.

Each entity (box in Fig. 2) represents a location for the HAZOP. The recursive loop present in the model results in an additional location called “Objects” for aspects arising from the interactions between multiple objects. The observer is modeled by two components: “Observer—Optomechanics” and “Observer—Electronics”. This reduces complexity for the analysis and allows to focus on the different aspects of the image capturing process.

3.3 Parameters

Each location is characterized by parameters. They refer to physical and operational aspects describing the configuration of the subcomponent. The set of parameters chosen for a

single location during the HAZOP should be adequate for its characterization. Table 2 shows the parameters chosen for the location “Medium” as an example. Too few parameters for a location means that it is insufficiently modeled and that the analysis will likely contain gaps. Performing an analysis with too many parameters would require too much effort and create redundancy. A full listing of all parameters is available at the website vitro-testing.com.

3.4 Guide Words

A guide word is a short expression to trigger the imagination of a deviation from the design/process intent. Number and extent of guide words must be selected to ensure a broad view on the topic. Nevertheless, their number is proportional to the time needed for performing the HAZOP, so avoiding redundant guide words is essential. The provided examples in Table 1 show all guide words we used in the analysis. Exemplary meanings for the deviations caused by each guide words are given, but the experts are not limited to these specific interpretations during the risk analysis. The first seven “basic” guide words are standard guide words used in every HAZOP. The remainder are adaptations and additions that provide important aspects specific for CV: spatial and temporal deviations (Table 2).

3.5 Implementation

The actual implementation of the HAZOP is the systematic investigation of each combination of guide words and parameters at every location in the system. It is performed redundantly by multiple contributors. Afterwards, the results are compared and discussed to increase quality and completeness. Each HAZOP contributor assigns at least one meaning to a combination. In addition, for each meaning found the contributors investigate the direct consequences of this deviation on the system. One meaning can result in multiple consequences at different levels. Each entry in the list represents an individual hazard which can lead to actual decreases in the total system’s performance or quality. Combinations that result in meaningful interpretations by any contributor are considered to be “meaningful” entries while combinations without a single interpretation are considered to be “meaningless”.

3.6 Execution

The execution of the CV-HAZOP, including various meetings and discussions by the contributors (with expertise in testing, analysis, and CV), took one year. Each location is covered by at least three of the authors. The additional experts are mentioned in the acknowledgments. The 52 parameters from all seven locations, combined with

Table 1 Guide Words used in the CV-HAZOP

Guide word	Meaning	Example
Basic		
No	No information can be derived	No light at all is reflected by a surface
More	Quantitative increase (of parameter) above expected level	Spectrum has a higher average frequency than expected
Less	Quantitative decrease below expected level	Medium is thinner than expected
As well as	Qualitative increase (additional situational element)	Two lights shine on the same object
Part of	Qualitative decrease (only part of the situational element)	Part of an object is occluded by another object
Reverse	Logical opposite of the design intention occurs	Light source casts a shadow instead of providing light
Other than	Complete substitution—another situation encountered	Light source emits a different light texture
Additional—spatial		
Where else	“Other than” for position/direction related aspects	Light reaches the sensor from an unexpected direction
Spatial periodic	Parameter causes a spatially regular effect	A light source projects a repeating pattern
Spatial aperiodic	Parameter causes a spatially irregular effect	The texture on object shows a stochastic pattern
Close/remote	Effects caused when s.t. is close to/remote of s.t. else	Objects at large distance appear too small
In front of/behind	Effects caused by relative positions to other objects	One object completely occludes another object
Additional—temporal		
Early/Late	Deviation from temporal schedule	Camera iris opens too early
Before/after	A step is affected out of sequence, relative to other events	Flash is triggered after exposure of camera terminated
Faster/slower	A step is not done with the right timing	Object moves faster than expected
Temporal periodic	Parameter causes a temporally regular effect	Light flickers periodically with 50 Hz
Temporal aperiodic	Parameter causes a temporally irregular effect	Intensity of light source has stochastic breakdowns

Table 2 Parameters used in the location Medium

Parameter	Meaning
Transparency	Dimming factor per wavelength and distance unit
Spectrum	Color, i.e. richness of medium with respect to absorption spectrum (isotropic or anisotropic)
Texture	Generated by density fluctuations and at surfaces (e.g. water waves)
Wave properties	Polarization, coherence
Particles	Influences and effects of the particles that make up the medium

the 17 guide words, result in 884 combinations. Each combination can have multiple meanings assigned to it. Finally, 947 unique and meaningful entries have been produced. Table 3 shows an excerpt of entries from the final HAZOP and Fig. 3 shows visualizations for each hazard

mentioned. The entries in the list can include multiple meanings for each parameter as well as multiple consequences and hazards per meaning. The whole resulting dataset of the CV-HAZOP is publicly available at www.vitro-testing.com.

Table 3 Excerpt from CV-HAZOP entries (simplified)

HID	Location/parameter	Guide word	Meaning	Consequence	Example
16	Light sources/position	No	Position of light source not known	Light source is not detected	Confusion of algorithm due to missing relation of shadows to the respective light source
271	Medium/particles	Faster	Particles move faster than expected	Motion blur of particles	Blurred particles obfuscate scene; image recognition severely reduced
370	Object/complexity	More	Object is more complex than expected	Object has features or feature combinations which make its correct recognition difficult	Object is not correctly recognized
537	Objects/number	No	Number of objects is not detectable/decidable	Scene with unknown number of objects	Nonexistent objects are reported (false positives)
724	Objects/transparency	More	Objects are more transparent than expected	More objects are visible than usual	Algorithm is confused by clutter in scene
997	Observer/transmittance of optics	Spatial periodic	The optical density of the lenses changes in a spatially periodic manner, e.g., microlens array	The scene appears periodically repeated over the image in shrunk size	Strong confusion of CV alg.
1154	Observer/resolution	More	The sensor resolution is higher than expected	More noise than expected	Increase error rate on scene interpretation due to higher noise rate
1350	Algorithm/models	Part of	Models are incomplete	System uses incomplete model data	Misdetections due to incomplete models

Fig. 3 Visualization of entries from Table 3

3.7 Resulting List

In total, 947 entries are considered meaningful by the experts. A detailed analysis of the meaningful entries achieved for each guide word/parameter combination is shown in Fig. 4. One goal is to maximize the meaningful entries—and the graphic shows reasonably high entries for most of the basic guide words (see Table 1). Lower valued entries in the matrix can be explained as well: The concepts of the spatial aspects “Close” and “Remote” are simply not applicable to the properties of the electronic part of the observer (obs. electronics) and the concept of space in general is not applicable to a number of parameters at various locations. This also holds true for the temporal guide words which do not fit to the optomechanical and medium locations. Nevertheless, even here the usage of guide word/parameter combinations inspire the analysts to find interpretations which would have been hard to find otherwise. Each hazard entry is assigned a unique hazard identifier (*HID*) to facilitate referencing of individual entries of the checklist.

4 Application

The remainder of this paper focuses on the application of the checklist as an evaluation tool for existing test datasets. On the one hand, we show that the CV-HAZOP correctly identifies challenging situations and on the other hand, we provide a guideline for all researches to do their own analysis of test data.

Initially, the evaluators have to clarify the intent and domain of the specific task at hand. This specification creates the conceptual borders that allow the following analysis to filter the hazards. The intent includes a description of the goals, the domain defines the conditions and the environment under which any algorithm performing the task should work robustly. With the intent and domain specified, the evaluators can now check each entry of the CV-HAZOP list to see if that entry applies to the task at hand. Often it is useful to reformulate the generic hazard entry for the specific algorithm to increase readability. In the following a process outline is given:

	Generic							Temporal							Spatial					
Light Source	1.00	1.00	1.00	1.00	0.63	0.56	0.78	1.00	1.00	0.67	0.73	0.60	0.25	0.67	0.88	0.75	0.63	0.38	0.56	0.56
Medium	1.00	1.00	1.00	1.00	0.60	0.80	1.00	0.80	0.80	0.20	0.20	0.83	0.80	0.80	1.00	1.00	0.80	1.00	0.40	0.40
Object	1.00	1.00	1.00	1.00	0.91	0.91	1.00	1.00	1.00	1.00	0.90	0.60	0.50	0.91	0.92	0.82	0.73	0.30	0.30	
Objects	1.00	1.00	1.00	1.00	1.00	0.82	0.88	1.00	1.00	0.71	0.75	1.00	0.71	0.92	1.00	0.86	1.00	1.00	1.00	1.00
Obs. Optomechanics	1.00	1.00	1.00	0.93	1.00	0.75	1.00	0.69	0.86	0.64	0.69	0.75	0.62	0.92	0.85	0.83	0.75	0.67	0.50	0.58
Obs. Electronics	1.00	1.00	1.00	1.00	1.00	0.80	1.00	1.00	1.00	0.83	1.00	1.00	1.00	0.80	1.00	1.00	0.00	0.00	0.20	0.20
Algorithm	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.83	0.57	0.50	0.71	0.83	0.33	0.33
	No	More	Less	As well as	Part of	Reverse	Other than	Temp. periodic	Temp. aperiodic	Before (temp.)	After	Faster	Slower	Where else	Spat. periodic	Spat. aperiodic	Close (spat.)	Remote (spat.)	In front of (spat.)	Behind (spat.)

Fig. 4 Ratio of meaningful combinations for each guide word per location (averaged over all parameters of each location)

1. Check if the preconditions defined by the column *Meaning* and the according *Consequences* apply.
2. Check if the *Example* matches the specific task at hand.
3. Each row represents a unique *Hazard* and has a unique *Hazard ID* (HID). If the *Hazard* is too generic to be feasible, add a new row for the specific task using a matching *Example*.
4. Evaluate if the *Hazard* can be detected (i.e. is visible in the test data).
5. Store the identity of test cases which fulfill relevant *HIDs*. Create a new test case should none of the current test cases fulfill this *Hazard*.

Previous evaluations for comparable tasks can be used as templates to speed up this process and to reduce the effort compared to evaluating the whole generic list. Specialized hazards can be added to the checklist so that they can be used directly in future evaluations.

With the reduced list of possible hazards, the evaluators are able to go through test datasets and mark the occurrence of a hazard. Usually a simple classification per test case is enough. Individual pixel-based annotations can also be used to indicate the location of specific hazards in test images (see Sect. 5). After this process, the missing hazards are known and quantifiable (e.g. 70% of all relevant hazards are tested using this test dataset). This is a measure of completeness which can be used to compare datasets. Even more important: If a hazard cannot be found in the test data, the CV-HAZOP entry states an informal specification for creating a new test case to complement the test dataset. The extensiveness of the checklist allows a thorough and systematic creation of new test datasets without unnecessary clutter.

Each hazard entry in the check list has a unique hazard identifier (HID). This allows to easily reference individual hazards and compare results from different CV implementations. The checklist approach allows for a top-down evaluation of CV (starting from the problem definition down to the pixel level). This is a good complement to regular benchmarks which tend to be focused on the detailed pixel level (bottom-up evaluation).

5 Example

As proof of concept, the authors applied the described process to a specific task. We chose canonical stereo vision:

The intent of the algorithm is the calculation of a dense disparity image (correspondence between the pixels of the image pair) with a fixed epipolar, two camera setup. To further simplify the analysis, we only use greyscale information and assume that the cameras are perfectly synchronous (exposure starts and stops at the same instants), and omit the use of any history information so that many time artifacts

can be disregarded. The domains of the algorithm are indoor rooms or outdoor road scenarios. Conditions like snow, fog, and rain are included in the problem definition. This was done to keep the problem definition sufficiently generic to allow room for the analysis.

Note that this evaluation is not designed to compare stereo vision algorithms themselves or to compare the quality of the specific datasets (will be done in future works). However, this paper provides the first step: a clear proof of concept of the CV-HAZOP list as a tool for validation. The simplifications in domain/intent analysis and algorithm evaluation were performed to reduce complexity/workload and should be re-engineered for a specific stereo vision evaluation.

First, six experts in the field of CV (some had experience with the CV-HAZOP list, others were new to the concept) analyzed the initial 947 entries and identified those applying to the stereo vision use case. During this step, 552 entries were deemed to be not applicable and 106 entries were non-determinable (not verifiable by only surveying the existing test data; more background knowledge needed). The remaining 289 entries were deemed to be relevant for stereo vision. See Table 4 and Fig. 5 for examples from the datasets. About 20% of the hazard formulations were further specified to simplify the following annotation work while the rest were already specific enough. The experts analyzed three test datasets commonly used for stereo vision evaluation (see Table 5) individually for each of the identified hazard.

The hazard entries were evenly distributed among evaluators. All evaluators had the task to annotate each assigned hazard at least once in each dataset (if present at all). The step to annotate all occurrences of individual hazards in all images was omitted as the required effort would exceed the resources reasonable for this proof of concept. One representative of each hazard is deemed sufficient for the purpose of this proof-of-concept but certainly requires a larger sample size for a detailed evaluation of a CV algorithm. Ideally, a test dataset should include a systematically increasing influence of each hazard so that the algorithm's point of failure can be evaluated.

The annotation tool was set to randomly choose the access order to reduce annotation bias by removing the influence of image sequence ordering. Table 5 summarizes the results of the evaluation showing the number of images with hazards and the number of uniquely identified hazards. It is not a surprise that KITTI contains the most hazards: it is the largest dataset and is also created in the least controlled environment (outdoor road scenes). It contains many deficiencies in recording quality manifesting as hazards and it includes images with motion blur as well as reflections on the windshield.

Many effects stemming from interactions of multiple light sources, medium effects, and sensor effects are missing in all three test datasets. The majority of hazards present in the data

Table 4 Example for entries used during the example application of the CV-HAZOP list (simplified)

HID	Location/parameter	Guide word	Meaning	Consequence	Example
125	Light source/intensity	More	Light source shines stronger than expected	Too much light in scene	Overexposure of lit objects
481	Object/reflectance	As well as	Obj. has both shiny and dull surface	Diffuse reflection with highlight/glare	Object recognition distorted by glares
445	Object/texture	No	Object has no texture	Object appears uniform	No reliable correspondences can be found
706	Objects/reflectance	Close	Reflecting Obj. is closer to Observer than expected	Reflections are larger than expected	Mirrored scene taken for real
584	Objects/positions	Spatial periodic	Objects are located regularly	Same kind of objects appear in a geometrically regular pattern	Individual objects are confused
1059	Optomechanics/aperture	Where else	Inter-lens reflections project outline of aperture	Ghosting appears in the image	Aperture projection is mis-interpreted as an object
1123	Electronics/exposure	Less	Shorter exposure time than expected	Less light captured by sensor	Details uncorrelated due to underexposure

deal with specific situations that produce overexposure (HIDs 26, 125, 479, 482, 655, 707, 1043, 1120), underexposure (HIDs 21, 128, 651, 1054, 1072, 1123), little texture (HIDs 444, 445, 449) and occlusions (HIDs 608, 626).

6 Evaluation

In this section we evaluate the effect of identified hazards on algorithm output quality. The goal is to show that the entries of the CV-HAZOP are meaningful and that the checklist is a useful tool to evaluate robustness of CV algorithms. A specific hazard can only impact the system if it is visible in the image. Thus, we need to annotate areas in images corresponding to specific hazards to show that the annotated area itself (and, thus, the included hazard) is responsible for the output quality decrease. Initially it was unclear how accurate these areas have to be defined. For this purpose two different types of annotations were evaluated: a manually selected outline and a bounding box calculated from the outline.

We potentially add another bias to our analysis by evaluating only areas that contain annotations. This has two influences: (i) We only look at frames that have annotations while ignoring all other frames in the dataset without any annotations, (ii) We average over small sampling windows that often contain relatively little data due to missing values in the GT.

To quantify these influences we generated another set of control annotations: for each annotation in the dataset we generated a mask with a random position but the same size as the annotated hazard in the respective frame.

At last the overall performance of an algorithm was needed as a base line value. For this the whole image was evaluated. All in all we generated four types of masks from the annotations for our evaluation.

The different masks represent a step-by-step increase of influence of the annotated areas:

- *shape* masks with the annotated outlines as filled polygons,
- *box* masks with boxes of equal size and centroid as each annotated outline,
- *rand* masks with boxes of equal size as the annotated outlines but a randomly placed centroid,
- *all* masks with all pixels except the left border region (to exclude occlusions).

Figure 6 gives an example of the generated masks. Not every image in the test datasets contains annotations. The masks *shape*, *box*, and *rand* are evaluated for the subset of images containing at least one annotation while *all* is evaluated for all images of the datasets.

Fig. 5 Examples for each entry in Table 4. Images are taken from the datasets described in Table 5

Table 5 Stereo vision test datasets used in our evaluation, number of found hazards and percentage of masks covered by GT

Algorithm	Image pairs	Images with hazards	Found hazards	# Annotations	% GT all	% GT rand (avrg.)	% GT box	% GT shape
Middlebury Stereo Evaluation (MB06) (Scharstein and Szeliski 2002)	26	19	34	55	96.0	88.2	94.8	92.5
Middlebury Stereo Eval. “New” (MB14) (Scharstein et al. 2014)	23	17	57	80	96.9	93.0	93.5	91.2
The KITTI Vision Benchmark (KITTI) (Geiger et al. 2012)	194	62	76	101	45.7	42.0	36.8	37.0

Fig. 6 Example for annotation masks for hazard ‘No Texture’ (from left to right): input image, shape, box, rand, all

The *rand* masks only represent the annotated area’s size as well as the subset of annotated frames. A total of 100 random masks are generated for each annotation that share its size but are randomly displaced. Statistics can thus be evaluated over the whole set of random masks which increases the significance. Annotation *box* represents area and position while *shape* represents the full annotation.

The *rand* versus *all* masks verify if the output quality is affected by using smaller image parts for evaluation instead of the whole image as well as a subset of frames, while *box* versus *shape* evaluates the influence of specific shapes of the annotations.

Table 5 lists the resulting number of annotations created for each dataset. Some hazards require the selection of split areas, resulting in multiple annotations. We only use pixels with valid GT information for evaluation. Unfortunately, many of the hazards (e.g. reflections, transparencies, occlusions, very dark materials) also have a negative influence on the laser scanner used for the GT generation in KITTI. The GT data is generally sparse and even more sparse in the annotated areas.

6.1 Performance Evaluation

For evaluation of the stereo vision test dataset we used the following popular stereo vision algorithms: SAD + tex-

ture thresholding (TX) & connected component filtering (CCF) (Konolige 1998), SGBM + TX & CCF (Hirschmüller 2008), census-based BM + TX & CCF (Humenberger et al. 2010; Kadiofsky et al. 2012), cost-volume filtering (CVF) & weighted median post processing filtering (WM) (Rhemann et al. 2011), PatchMatch (PM) & WM (Bleyer et al. 2011), and cross-scale cost aggregation using census and segment-trees (SCAA) & WM (Zhang et al. 2014), (Mei et al. 2013). The resulting disparities of each stereo vision algorithm are compared to the GT disparities of the test dataset. The number of wrong pixels (with an error threshold of $>2\text{px}$) is then compared to the number of pixels within the respective mask that had valid ground truth values. Invalids in the result are counted as being above any threshold. We consider each disparity pixel $d_i \in \mathbb{R}^*$ to either be valid ($\in \mathbb{R}$) or invalid (denoted by the star value “ $\star$ ”). Where $\mathbb{R}^* = \mathbb{R} \cup \{\star\}$. The same holds for each corresponding ground truth pixel value $g_i \in \mathbb{R}^*$. We consider every d_i for which $\text{correct}(d_i, g_i) = \text{true}$ to be true, and $\text{correct} : \mathbb{R}^* \times \mathbb{R}^* \mapsto \text{true, false}$ to be defined by:

$$\text{correct}(g_i, d_i) = \begin{cases} \text{true} & \text{for } d_i = \mathbb{R} \\ & \wedge g_i \neq \star \\ & \wedge |d_i - g_i| < 2 \\ \text{false} & \text{else} \end{cases} \quad (1)$$

The actual comparison is performed for each dataset independently according to the average error $\bar{e}_m$ as defined by (2) where $\mathbb{D}_m, \mathbb{G}_m$ are the disparity and GT values selected by a given mask $m \in \{\text{“shape”}, \text{“box”}, \text{“rand”}, \text{“all”}\}$.

$$\bar{e}_m = \frac{|\{\forall d_i \in \mathbb{D}_m, g_i \in \mathbb{G}_m : \neg \text{correct}(d_i, g_i)\}|}{|\{\forall g_i \in \mathbb{G}_m : g_i \in \mathbb{R}\}|} \quad (2)$$

Fig. 7 Percentage of pixels with an error above 2px for all algorithms for the different masks and datasets (average error $\bar{e}$)

Figure 7 shows the result of the evaluation for all three datasets and all four mask types. The arithmetic average of the performance evaluated for 100 random masks are reported as *rand*. We chose to use a high threshold of 2pxl to distinguish the coarse cases “algorithm succeeded at finding a good correspondence” versus “algorithm could not determine a correct correspondence” as opposed to measuring small measurement errors. The performances of the different mask types creates a distinct picture. Section 6.2 will first interpret the results. The following Sect. 6.3 will then assign statistical significance to these interpretations.

6.2 Interpretation

The effect of applying the masks based on the identified hazards can be clearly seen. Table 6 summarizes the ratios between the error values of *shape* and *all*. The correctly masked areas (*shape*) have higher error ratios than the mean for the full image (*all*). The results for KITTI are much more erratic than the rest. The large amount of missing GT data in this dataset reduced its value for this evaluation drastically. The majority of *shape* mask areas have higher error ratios than the same-sized *box* mask areas. Newer and more complex algorithms generally score lower errors and have lower absolute differences between *shape* and *all* errors. There are two distinct groupings: *rand* masks have comparable results as *all* masks while *box* is comparable to *shape*. This suggests that box annotations can often be used instead of the time-consuming shape annotations. This allows for the following conclusions based on the different maskings: algorithms have higher error rates at annotated areas and score

even higher error rates if the annotation’s shape is preserved (*shape* vs. *box*). The effect of sampling patches of different sizes in each image is not prevalent (*rand* vs. *box*) and can be neglected.

6.3 Statistical Significance

The intuitive grouping of the mask into groups (*all*, *rand*) and (*shape*, *box*) is now evaluated for its statistical significance. The null hypothesis H_0 we will test is that the average performance evaluated at two different mask-types is not distinguishable. More specifically, that the differences between pairings of measurements (x_i, y_i) are symmetrically distributed around zero. This hypothesis should be valid between the grouped mask types and invalid between the other types.

To test the hypothesis, parametric and non-parametric tests can be used. Parametric tests (e.g. *T*-test) need to make assumptions about the underlying distribution. Such assumptions would be detrimental for our analysis as they could introduce bias. From the possible non-parametric tests we chose the Wilcoxon signed rank test (Wilcoxon 1945) because of its robustness and the possibility to evaluate over all three datasets in one joined analysis (see Demšar 2006) for a comparison between similar suited tests). The evaluation of all three datasets in one test statistic increases the sampling size and, thus, the test’s significance.

The Wilcoxon signed rank test works by calculating the absolute difference for each pair of measurements from the two distributions and sorting those differences in ascending order. The rank in this order is now summed up using the original sign of each of the differences and the absolute value of this sum is used as the test statistic W . Ties in the ranking receive all the same average over the tying ranks. The number of differences not equal to zero is denoted with N_r .

Distributions with a symmetry around zero will yield a sum that has an expected value of zero and a variance of

Table 6 Ratio between average errors of *shape* and *all*: $\frac{\bar{e}_{\text{shape}}}{\bar{e}_{\text{all}}}$

Dataset	SAD	CEN	SGBM	CVF	PM	SCAA
MB06	1.47	1.89	1.95	1.55	1.49	1.15
MB14	1.47	1.58	1.67	1.76	1.65	1.97
KITTI	1.17	1.69	1.86	1.28	2.31	1.07

Table 7 Probability values z_W obtained from the Wilcoxon signed rank test for different pairings

Pairing	SAD	CEN	SGBM	CVF	PM	SCAA	Overall
all, rand	0.46	0.08	−0.33	−1.17	−0.81	−1.18	−1.10
shape, box	−4.54	−3.22	−2.74	−0.36	−0.20	−0.58	−4.89
all, shape	4.79	5.95	4.89	2.19	4.03	0.87	9.64
all, box	4.00	5.72	4.92	2.50	4.01	0.84	9.43
shape, rand	−4.40	−4.98	−4.16	−2.36	−3.21	−1.28	−8.53
box, rand	−3.55	−4.70	−4.15	−2.48	−3.37	−1.40	−8.23

Bold entries represent rejected null hypothesis when using a typical significance level of 5% (translates to a z value of ± 1.96). Negative values mean the first entry of the pairing was more difficult than the second; positive values signify the opposite

$var_W = N_r(N_r + 1)(2N_r + 1)/6$. For $N_r > 9$ the distribution of W approaches a normal distribution with $\sigma_W = \sqrt{var_W}$ and $z_W = W/\sigma_W$. These resulting probability values z_W can be used as a measure for rejecting the null-hypothesis if z_W is larger than z_{W_c} based on the selected significance level.

In our case we calculate the differences using average performance between two mask variants for each single test case (stereo image pair) from the datasets and then sort all differences by their absolute value. The resulting sum of the signed ranks is divided by σ_W for the corresponding N_r of that comparison yielding a single z value each. This test is performed for all relevant pairings of masks and for each algorithm, but we will combine the differences for all datasets. Finally we also calculate the overall z value for each pairing by evaluating the cumulation of all algorithm results. Table 7 shows the summarized results for all tests. The 100 samples of each mask generated for *rand* are used to calculate 100 times the value of z_W for each combination that contains *rand*. The table entry contains the arithmetic average of all 100 values. For this evaluation we keep the sign of the resulting test statistic to preserve the direction of each comparison. The decision whether to accept or reject the null hypothesis (distribution of results from different masks are the same) is based on the selected significance level. This percentage describes the probability of rejecting a true null hypothesis (*type I error*). We now apply a significance level of 5% to the data which translates to a z value of ± 1.96 . All null hypothesis with an absolute z_W value of higher than $z_{W_c} = 1.96$ can be rejected.

This results in the following observations:

- (*all*, *rand*) is not significantly different, the null-hypothesis that both conform to the same distribution can be accepted
- (*shape*, *box*) is significantly different, *shape* is more difficult than *box*
- (*all*, *shape*) has the most significant difference, *shape* is much more difficult than *all*. The pairing *all*, *box* is also presenting the same level of significant differences. (*shape*, *rand*) and (*box*, *rand*) show slightly less signifi-

cance but are still very definite: both *shape* and *box* are significantly more difficult than *rand*

- The significance of the results varies widely between the different algorithms. Older and real-time algorithms tend to show the highest test statistics. SCAA results in the same trends as the remaining algorithms but stays always below the significance level of 5%.

The evaluation paints a clear overall picture: areas identified by the CV experts as containing a visual hazard guided by the CV-HAZOP checklist are especially challenging for the selected CV algorithms. Focusing on these challenging areas is beneficial for robustness evaluations since it creates more meaningful test cases.

7 Conclusion

Many critical situations and relations have the potential to reduce the quality and functionality of CV systems. The creation of a comprehensive checklist containing these elements is a crucial component on the road towards systematic validation of CV algorithms. This paper presents the efforts of several experts from the fields of CV as well as risk and safety assessment to systematically create such a list. To the authors' best knowledge, this is the first time that the risk analysis method HAZOP has been applied extensively to the field of computer vision.

The CV-HAZOP is performed by first introducing a generic CV model which is based upon information flow and transformation. The model partitions the system into multiple subsystems which are called locations. A set of parameters for each location is defined, that characterize the location's individual influence on information. Additional special CV-relevant "guide words" are introduced that represent deviations of parameters with the potential to create hazards. The execution of the HAZOP was performed by a number of authors in parallel, assigning meanings to each combination of guide words and parameters to

identify hazards. The individual findings were discussed and merged into one resulting CV-HAZOP list. A guideline for using the hazard list as a tool for evaluating and improving the quality and thoroughness of test datasets is provided.

The CV-HAZOP has produced a comprehensive checklist of hazards for the generic CV algorithm with over 900 unique entries. Each individual hazard is now referable by a unique hazard identifier (HID). It supports structured analysis of existing datasets and calculation of their hazard coverage in respect to the checklist. We present an example by applying the proposed guidelines to popular stereo vision datasets and finally evaluate the impact of identified hazards on stereo vision performance. The results show a clear correlation: identified hazards reduce output quality.

8 Outlook

The creation or combination and completion of test datasets using our checklist is the logical next step. We plan to guide the creation of a stereo vision test dataset with known coverage of hazards from our checklist. Another idea is the creation of test data that gradually increases the influence of specific hazards (e.g. amount of low contrast textures). This allows to find the point of failure and get an accurate estimation about the robustness of an algorithm when facing a specific hazard. The usage of our checklist can also be streamlined. Pre-filtered lists for common applications and domains provide specific lists without the need of manual adjustments. We are also investigating the automatic detection of hazards, i.e. algorithmic checks to determine if and where a hazard is present in a test image. This will reduce the manual task of categorizing test data and in the long run should lead to a fully automatic CV validation framework.

Our HAZOP checklist is not considered final. It will be updated to include lessons learned during evaluations and testing or even after tested systems are put into operation. By sharing this information with the community over our public HAZOP database we hope to increase quality and reduce effort in CV robustness evaluation. At this stage, the CV-HAZOP becomes a structured and accessible reference hub for sharing experiences with CV algorithm development, usage, and maintenance.

Acknowledgements Special thanks for their extensive CV-HAZOP contributions go to Lawitzky G., Wichert G., Feiten W. (Siemens Munich), Köthe U. (HCI Heidelberg), Fischer J. (Fraunhofer IPA), and Zinner C. (AIT). Thanks to Cho J.-H. (TU Wien) and Beham M. (AIT) for their help with the example chapter. The creation of the CV-HAZOP as well as this work have been funded by the ARTEMIS Project R3-COP, No. 100233 and the European Initiative to Enable Validation for Highly Automated Safe and Secure Systems (ENABLE-S3) Joint Undertaking under grant agreement grant agreement No. 692455. This joint

undertaking receives support from the European Union's HORIZON 2020 research and innovation programme and Austria, Denmark, Germany, Finland, Czech Republic, Italy, Spain, Portugal, Poland, Ireland, Belgium, France, Netherlands, United Kingdom, Slovakia, Norway. Additional support was received by the project autoBAHN2020 funded by the Austrian Research Promotion Agency with Contract Number 848896.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Aloimonos, J. Y., & Shulman, D. (1989). *Integration of visual modules: An extension of the Marr paradigm*. Boston: Academic Press Professional Inc.
- Bleyer, M., Rhemann, C., & Rother, C. (2011). Patchmatch stereo—stereo matching with slanted support windows. In *British machine vision conference*.
- Bowyer, K., & Phillips, P. J. (1998). *Empirical evaluation techniques in computer vision*. Los Alamitos, CA: IEEE Computer Society Press.
- Bowyer, K., Kranenburg, C., & Dougherty, S. (2001). Edge detector evaluation using empirical ROC curves. *Computer Vision and Image Understanding*, 84(1), 77–103.
- Center for Chemical Process Safety. (1992). *Guidelines for hazard evaluation procedures, with worked examples* (2nd ed.). Hoboken: Wiley.
- Department of Defense. (1949). *Procedures for Performing a Failure Mode, Effects and Criticality Analysis, MIL-STD-1629A*.
- Department of Defense. (1991). *Reliability Prediction of Electronic Equipment: MIL-HDBK-217F*.
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7, 1–30.
- Fenelon, P., & Hebborn, B. (1994). Applying HAZOP to software engineering models. In *Risk management and critical protective systems: proceedings of SARSS* (pp. 11–116).
- Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381395.
- Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Computer vision and pattern recognition*.
- Goseva-Popstojanova, K., Hassan, A., Guedem, A., Abdelmoez, W., Nassar, D. E. M., Ammar, H., et al. (2003). Architectural-level risk analysis using UML. *IEEE Transactions on Software Engineering*, 29(10), 946–960.
- Hampel, F. R. (1971). A general qualitative definition of robustness. *The Annals of Mathematical Statistics*, 42(6), 1887–1896.
- Hirschmüller, H. (2008). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2), 328–341.
- Honauer, K., Maier-Hein, L., & Kondermann, D. (2015). The HCI stereo metrics: Geometry-aware performance analysis of stereo algorithms. In *The IEEE international conference on computer vision (ICCV)*.
- Huber, P. (1964). Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1), 73–101.

- Humenberger, M., Zinner, C., Weber, M., Kubinger, W., & Vincze, M. (2010). A fast stereo matching algorithm suitable for embedded real-time systems. *Computer Vision and Image Understanding*, 114, 1180–1202.
- International Electrotechnical Commission. (2010). *Functional safety of electrical/electronic/programmable electronic safety-related systems—part 4: Definitions and abbreviations: IEC 61508-4*.
- Kadiofsky, T., Weichselbaum, J., & Zinner, C. (2012). Off-road terrain mapping based on dense hierarchical real-time stereo vision. In *Advances in visual computing*. Lecture Notes in Computer Science (Vol. 7431, pp. 404–415). Berlin: Springer.
- Kajiya, J. T. (1986). The rendering equation. In *SIGGRAPH conference proceedings* (Vol. 20, No. 4, pp. 143–150).
- Kletz, T. A. (1983). *HAZOP and HAZAN notes on the identification and assessment of hazards*. The Institution of Chemical Engineers.
- Kondermann, D. (2013). Ground truth design principles: An overview. In *Proceedings of the international workshop on video and image ground truth in computer vision applications, VIGTA '13* (pp. 5:1–5:4). ACM, New York, NY, USA.
- Kondermann, D., Nair, R., Meister, S., Mischler, W., Güssefeld, B., Honauer, K., et al. (2015). Stereo ground truth with error bars. In *Asian conference on computer vision*.
- Konolige, K. (1998). Small vision systems: Hardware and implementation. In *Robotics research*. Berlin: Springer.
- Laprie, J. (1992). Dependability: Basic concepts and terminology. In *Dependable computing and fault-tolerant systems* (Vol. 5). Berlin: Springer.
- Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. San Francisco: W. H. Freeman.
- Matthias, B., Oberer-Treitz, S., Staab, H., Schuller, E., & Peldschus, S. (2010). Injury risk quantification for industrial robots in collaborative operation with humans. In *Proceedings of the 41st international symposium on robotics and 6th German conference on robotics*.
- Mei, X., Sun, X., Dong, W., Wang, H., & Zhang, X. (2013). Segment-tree based cost aggregation for stereo matching. In *Computer vision and pattern recognition* (pp. 313–320).
- Min, J., Powell, M., & Bowyer, K. W. (2004). Automated performance evaluation of range image segmentation algorithms. *IEEE Transactions on Systems Man and Cybernetics Part B Cybernetics*, 34, 263–271.
- Nyquist, H. (1928). Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 47(2), 617–644.
- Pinto, N., Cox, D. D., & DiCarlo, J. J. (2008). Why is real-world visual object recognition hard? *PLOS Computational Biology*, 4, e27.
- Ponce, J., Berg, T. L., Everingham, M., Forsyth, D. A., Hebert, M., Lazebnik, et al. (2006). Dataset issues in object recognition. In *Toward category-level object recognition* (pp. 29–48). Springer.
- Rhemann, C., Hosni, A., Bleyer, M., Rother, C., & Gelautz, M. (2011). Fast cost-volume filtering for visual correspondence and beyond. In *Computer vision and pattern recognition* (pp. 3017–3024).
- Scharstein, D., & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1), 7–42.
- Scharstein, D., & Szeliski, R. (2003). High-accuracy stereo depth maps using structured light. In *Computer vision and pattern recognition*.
- Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nescic, N., Wang, X., & Westling, P. (2014). High-resolution stereo datasets with subpixel-accurate ground truth. In *Pattern recognition* (pp. 31–42). Springer.
- Schlick, R., Herzner, W., & Jöbstl, E. (2011). Fault-based generation of test cases from UML-models approach and some experiences. In *Computer safety, reliability, and security*. Lecture Notes in Computer Science (Vol. 6894, pp. 270–283). Berlin: Springer.
- Shannon, C. E. (1949). Communication in the presence of noise. *Proceedings of the Institute of Radio Engineers*, 37(1), 10–21.
- Shannon, C. E., & Weaver, W. (1949). *The mathematical theory of communication*. Champaign: University of Illinois Press.
- Strecha, C., von Hansen, W., Van Gool, L., Fua, P., & Thoennessen, U. (2008). On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Computer vision and pattern recognition*.
- Takanen, A., DeMott, J., & Miller, C. (2008). *Fuzzing for software security testing and quality assurance*. Artech House on Demand.
- Thacker, N., Clark, A., Barron, J., Ross Beveridge, J., Courtney, P., Crum, W., et al. (2008). Performance characterization in computer vision: A guide to best practices. *Computer Vision and Image Understanding*, 109(3), 305–334.
- Torralba, A., & Efros, A. A. (2011). Unbiased look at dataset bias. In *Computer vision and pattern recognition* (pp. 1521–1528).
- Vesely, W. E., Goldberg, F. F., Roberts, N. H., & Haasl, D. F. (1981). Fault tree handbook. In *Systems and reliability research*. Office of Nuclear Regulatory Research: NRC.
- Von Bertalanffy, L. (1968). *General systems theory*. New York 41973, 40.
- Van der Spek, R., & Spijkervet, A. (1997). Knowledge management: Dealing intelligently with knowledge. In *Knowledge management and its integrative elements* (pp. 31–60).
- Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6), 80–83.
- Zhang, K., Fang, Y., Min, D., Sun, L., Yang, S., Yan, S., & Tian, Q. (2014). Cross-scale cost aggregation for stereo matching. In *Computer vision and pattern recognition*.