VITRO – Model Based Vision Testing for Robustness

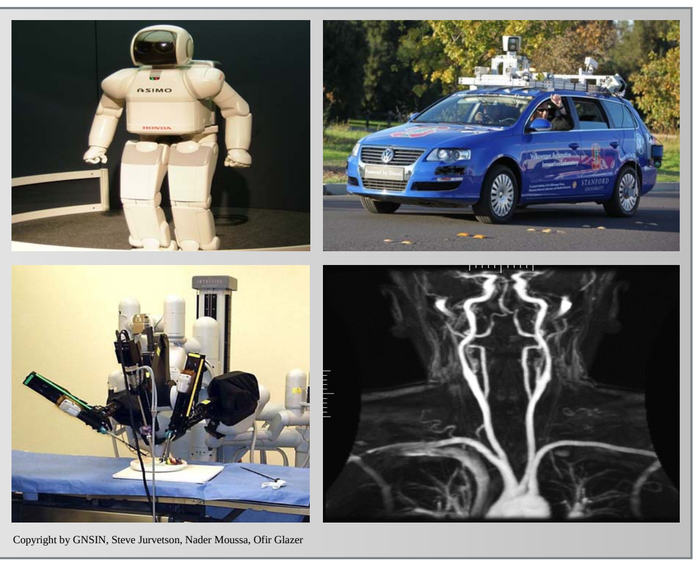
AUSTRIAN INSTITUTE OF TECHNOLOGY

INTRODUCTION

Computer vision (CV) is a key technology in many upcoming critical systems. Example applications include care robots, autonomous cars, assembly lines, logistics, robotic surgery and automated medical diagnostics. Errors in one of these systems could result in the loss of human life and therefore they are considered safety-critical. This means that their vision components and vision algorithms have to be dependable too. Verifying if the implementation of a CV algorithm fulfills the specifications is comparable to the verification of any other piece of software. But assuring that the implementation can solve the task at hand (validation) presents a special case.

Today, CV algorithms are usually tested by applying many real test images and comparing the results against a (manually) established ground truth. This method lacks computable coverage measures, and is hence insufficient for certification of CV algorithms, which is required for their commercial use in safety-critical applications.

This work deals with increasing the safety of CV systems and to enable their certification thus allowing their use in critical real world scenarios.

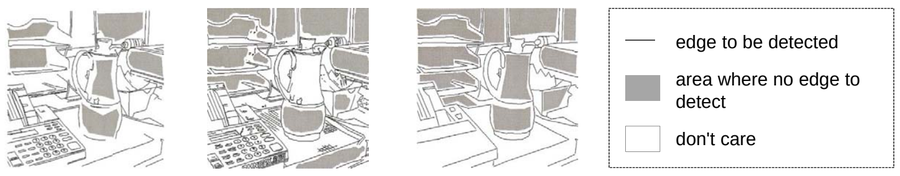


Copyright by GNSIN, Steve Jurvetson, Nader Moussa, Ofir Glazer

STATE OF THE ART

Currently a lot of test images are taken from reality to assemble a large dataset and the needed ground truth is generated manually by humans.
This is very slow and creates bias in multiple levels:
Input images show only typical views; e.g. Caltech 101 [1]:

Ground truth data can vary depending on the testing person [2]:

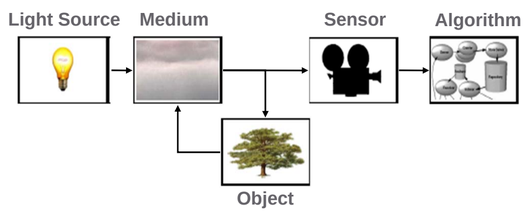


Is the test data set diverse / broad enough to uncover potential flaws?
How can one measure the potential flaw coverage of a given test set?

CV HAZOP

An important means of increasing the efficiency of testing is the use of previous experience: flaws and hazards that effected previous systems under comparable conditions are likely to challenge a new systems as well.
A HAZOP [3] was conducted to systematically assess potential hazards the general CV algorithm is facing:

Partition System into distinct “locations”:

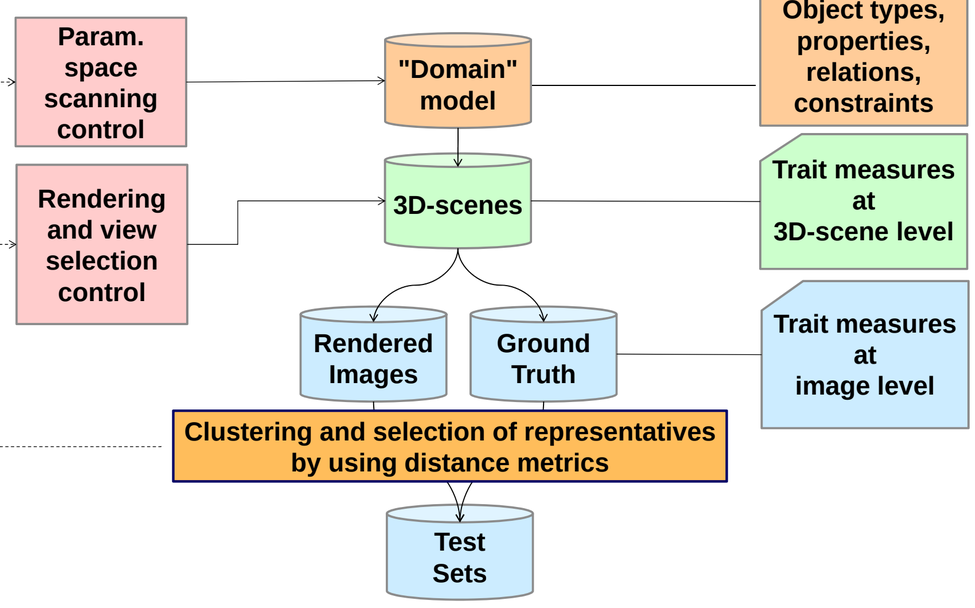


Define “parameters” for each location:

Parameter	Meaning
Number	Number of (distinguishable) light sources.
Position	One emphasized point of light source; e.g. COG
Area	The radiating area of the light source.
Spectrum	Colour, i.e. light source emission spectrum
Texture	Contrast distribution of emitted light.
Intensity	Scaling factor for Texture
Beam width	Opening angle of light source beam.
Wave prop.	Polarization, coherence

Together with well-defined “guidewords” this allows many combinations, each representing a potential hazard

THE BIG PICTURE



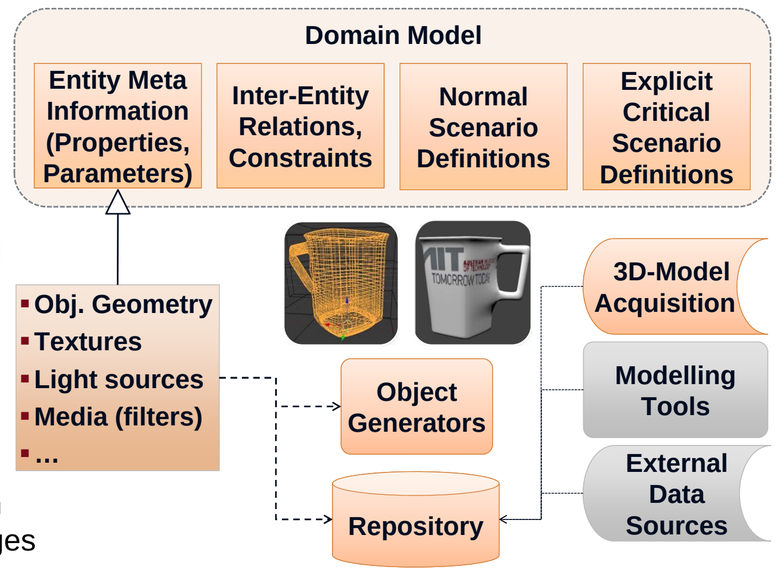
THE DOMAIN MODEL

Object description: notations from CG and Virtual Reality, e.g. [4]

Relations, constraints: own XML notation (UML[5]/OWL[6] based)

Object generators: parameterized, for obj. families like cups

3D-model acquisition: object descriptions from high-quality stereo images



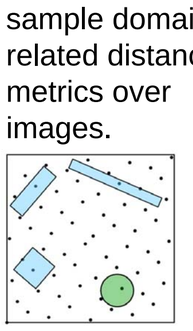
COVERAGE AND REDUNDANCY

Test data should cover the domain and criticalities without including too much redundancy so that test efforts are acceptable.

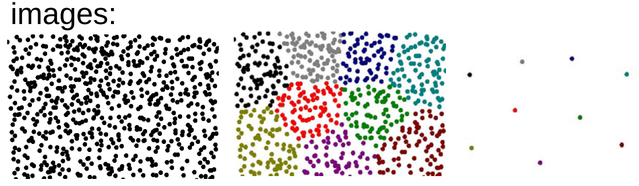
The similarity of images can be defined in many ways but when sticking to local pixel features [7], one can easily get unwanted results.

© by Chiswick Chap

We use low discrepancy sampling techniques [8] to sample domain parameters and to establish content-related distance metrics over images.



Calculate a vector representation of the sampling points for each view and use it to cluster images:

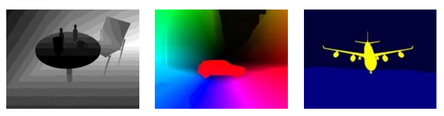


RESULTS

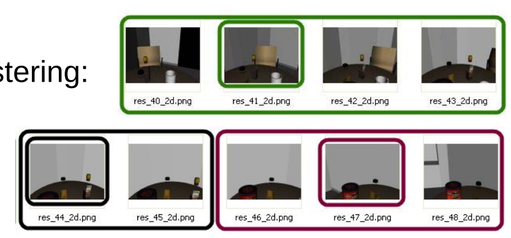
The VITRO tool chain allows the automatic generation of test images, the ground truth data (depth and segmentation) and its clustering.

Input:

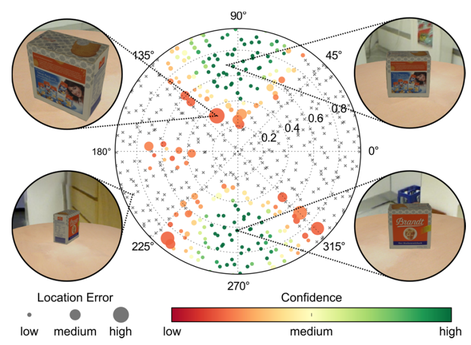
GT:



Clustering:



Evaluation example:
Object recognition depending on its orientation and distance:



REFERENCES

1. L. Fei-Fei, R. Fergus and P. Perona. Learning generative visual models from few training examples, IEEE. CVPR 2004, Workshop on Generative-Model Based Vision. (2004)
2. K. Bowyer, C. Kranenburg, S. Dougherty: Edge Detector Evaluation Using Empirical ROC Curves, Computer Vision and Image Understanding 84, 1–4, (2001)
3. F. Redmill, M. Chudleigh, J. Catmur; System Safety: HAZOP and Software HAZOP; John Wiley & Sons, ISBN: 978-0-471-98280-7 (1999;)
4. Khronos 3D Asset Exchange Schema COLLADA , http://www.khronos.org
5. Object Management Group (OMG): “UML Infrastructure”, v2.2. OMG document formal/2009-02-04 (2009);
6. Word Wide Web Consortium (W3C): “OWL 2 Web Ontology Language - Document Overview. (2009)
7. H. B. Mitchell, Image Similarity Measures, Image Fusion Part 3, 167-185, DOI: 10.1007/978-3-642-11216-4_14 (2010)
8. Matousek, Jiri, Geometric Discrepancy: An Illustrated Guide. Algorithms and Combinatorics. 2nd edition, Springer, ISBN: 978-3-642-03941-6 (2010)

This work was partially funded by ARTEMIS-Project R3-COP (contract nr. 100233)

Contact: Wolfgang Herzner, Oliver Zendel, Markus Murschitz [wolfgang.herzner@ait.ac.at]