VIALAB Machine vision benchmark
Preliminary results

Fabio Gori*, Simone Giardino**, Claudio Salati
T3LAB

Bologna, Museo del Patrimonio Industriale
23 Gennaio 2013

* Currently with Infovision
** Currently with System SpA
Goals of VIALAB machine vision benchmark

• Provide an independent benchmark of the most relevant machine vision libraries

“The machine vision benchmark that is part of the VIALAB project is different from academic benchmarks in that it is specifically aimed at commercial libraries and their overall characteristics: its main goal is to support users in the selection of the software that best fits their needs.”

• Construct a framework that allows an easy repetition of performance measures
  • On new libraries
  • On new versions of libraries that have already been tested

• Performance metrics and measures must be comparable
Benchmark framework

• Not an episodic assessment but a tool (a test bench) that supports and assesses the progress of the technology

• Explicit and detailed definition of each challenge task

• Supporting SW is open
  ➢ Based on OpenCV

• Purely image/ground truth based to make it repeatable
Stakeholders

• Users, OEMs and integrators
• Vision library vendors

➤ “The Middlebury Stereo Dataset and the NIST Face Recognition Grand Challenge have succeeded well in advancing their respective research algorithms by providing well-defined challenge tasks with ground truth and evaluation metrics.”

➤ When possible, we want to do it in cooperation with library vendors
What libraries?

• Primary target libraries identified in a survey of machine vision users, vendors, integrators, OEM

• Primary target libraries
  • Cognex VisionPro
  • Matrox MIL
  • MVTec Halcon
  • Teledyne/DALSA Sapera

• OpenCV? Also! But also a special role for it!
• Open to all vendors that are willing to participate
Expected from library suppliers

- Review of methodology
  - In general
  - Specific challenges
- Availability of their library (and calibration targets)
- Construction of datasets

➢ Provide library specific procedures for the solution of challenge tasks (*solutions*)

*It didn’t work!*
Comparable results

• Toolkit suppliers report tests they perform using the standard benchmark (MVTec proposal)
  • Standard procedure to report results
• Toolkit suppliers are invited to provide solutions for the benchmark: we run them on our test bench
  • If they don’t, we implement these solutions
  • Running test cases in a single site guarantees that results are comparable
• Results “include the code fragment used to solve the benchmark task. Beside transparency, this will allow users to learn more about the use of a given system.”
Benchmark platforms

- Different HW platforms
  - Simple embedded platform
  - PC-based
    - Single core vs. multi core
    - 32 vs. 64 bit
- Different SW platforms
  - Linux
  - Windows
- HW acceleration
  - GPU
  - FPGA
  - Dedicated boards

How do libraries manage multiprocessing and GPUs?
Algorithms

• Low level operators (e.g. Sobel) vs. Application functions

• Classes of problems vs. List of problems

• To avoid ad hoc solutions and optimizations

• Only some examples are provided to solution developers for a given class of problems

• Example:
  detection of textured objects vs. detection of untextured objects
Challenges

• 1st set of challenges: March 2012
  • 3D camera calibration
  • 2D object detection in orthogonal view images
  • 2D object detection under perspective distortion (without rectification)
  • 2D object localization in the image space in orthogonal view images
  • 2D object localization in the image space under perspective distortion (without rectification)

• 2.5D object localization in real world coordinates
• 2D camera calibration
• 2D object localization in the image space in orthogonal view images without scale variations
Performance metrics

• Speed
• Accuracy / trueness
• Precision (repeatability)
• Resolution
• Robustness
  • Parameters
  • Deformations
Performance metrics

From an MVTec presentation
Ground truth

- Localization in the real world
  - Thanks to the help of Marposs
  - Accuracy of 150 μm
- Localization in the image space: 2 approaches
  - We ended up using a procedure based on OpenCV: we define the ground truth using the same tools that we want to evaluate using that ground truth
  - Known a priori with synthetic images
Accuracy for object detection

• ROC (Receiver Operating Characteristic) curve

• TPR against FPR
  • FPR = False Positive Rate
    = False Positives / (False Positives + True Negatives)
  • TPR = True Positive Rate
    = True Positives / Ground Truth Positives

• Evaluation metric (academic): Area Under Curve (AUC)

• In real applications false positives are not acceptable
  ➢ Only the leftmost part of the curve, with small or null FPR, is significant
2D object detection:
untextured, orthogonal view, single instance

ROC curves computed over the entire dataset - Low Texture - Single Instances

- Matrix ME 9.00 R2: Geometric Model Finder
  AUC = 0.883
- MVTec HALCON 11.0: Shape-Based Matching
  AUC = 0.914
- Cognex VisionPro 7.2: PatMax
  AUC = 0.963

➤ "sparse" plot.

detection dataset

localization dataset
2D object detection: untextured, orthogonal view, single instance

➢ “sparse” vs. “dense”: 0.3Mp-8mm camera, detection dataset.
2D object detection: untextured, orthogonal view, single instance

How realistic are solutions? Polarity?

0.3Mp-8mm camera, detection dataset

ROC curves computed over the 0.3 MP / 8 mm camera - Low Texture - Single Instances

- Matrox ML 9.00 R2
- Geometric Model Finder
- MVTec HALCON 11.0
- Shape-Based Matching
- Cognex VisionPro 7.2
- PlatMax

no polarity info with polarity info
2D object detection: untextured, orthogonal view, multiple instances

• Results don’t change much.
2D object detection: untextured, orthogonal view, multiple instances

“sparse” vs. “dense”: 0.3Mp-8mm camera, detection dataset.
System related metrics

- Interoperability related to complexity of adapting to OpenCV based challenge APIs (in C/C++)
- Usability related to complexity of solutions (in C/C++)
- Footprint (code and data)
- Modularity
- Portability
  - Supported target systems
  - Requirements for port

Library specific IDE!
Robustness

• Different degrees of deformation
  • “As a starting point, MVTec proposes a number of benchmarks, each of which consists of a set of image sequences.
  • Within each sequence the influence of a “defect” is continuously increased.
  • For example, in template matching, an original image of a PCB could be generated and then successively defocused to provide a specific image sequence.
  • The quality of specific software can then be measured by the number of images that can be processed correctly.”

• In our benchmark summary results are averaged over all images but raw data are available for further analysis
Robustness to what?

- Similarity (also scale changes?)
- Perspective deformation
- Intra-class variance
- Clutter
- Occlusion
- Noise (several models)
- Defocus/Motion Blur
- Inhomogeneous illumination (several models)
- Under/Over Exposure
- Contrast variation
**Synthetic images**

- How can we handle all these deformations and their different degrees?
- How can we handle the combination of different deformations with different degrees?
- How do we determine the ground truth?
  - Synthetic images
  - We have created a tool for the introduction of calibrated deformations in a base image
Parameters

• Generally applicable
• Camera resolution (from VGA to 5Mp)
• Optics (6.5 and 8 mm)
• Computational platform
• Challenge specific, e.g. for the calibration challenge
  • Quality of the physical calibration target
  • Number of calibration images (or equivalent)
Definition of a challenge

- **General definition**
  the functionality that is evaluated and the metrics that are applied

- **Main Parameters**
  the parameters and deformations that are considered and the values they can have

- **Component tasks**
  the procedure that is followed to perform the assessment

- **Dataset and ground truth**
  rationale for the structure of the dataset and way the ground truth is created

- **Evaluation metrics**
  how the results computed by a library are compared against the ground truth and how the comparison data are summarized

- **Task interface**
  the API of the challenge, an example solution based on OpenCV, the benchmark framework that invokes the API, a development dataset
Protocol

1. VIALAB distributes the challenge definition
2. VIALAB waits for comments
3. VIALAB distributes a revised definition of the challenge
4. VIALAB waits for solution (or develops solution)
5. VIALAB runs solutions on the benchmark framework
6. VIALAB discusses individual results with participating vendor
7. All results are published
Summary

performance indices

• Not necessarily a single performance index but, possibly, different indices for different facets
  • Averaging distances in pixels over images with different resolutions?
• All raw performance data will remain available
  • Different summarization strategies
  • Review of anomalous cases
Byproducts

- A huge dataset (2TB) of images with an associated ground truth
- An OpenCV-based software for the generation of synthetic images with the associated ground truth, supporting similarity and perspective transformations and the introduction of a controlled amount of nuisances
- VIALAB extensions to OpenCV
- VIALAB utility procedures for other libraries
3D camera calibration

• Only Halcon and OpenCV can provide a solution to the challenge (in the way we defined it)

• General results: as a function of the number of calibration images, averaging all other parameters (camera resolution, optics, target quality)

<table>
<thead>
<tr>
<th>Number of images</th>
<th>OpenCV</th>
<th>Halcon</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Backward projection (error in pix)</td>
<td>Forward projection points (error in mm)</td>
</tr>
<tr>
<td>5</td>
<td>0,719</td>
<td>0,285</td>
</tr>
<tr>
<td>7</td>
<td>0,704</td>
<td>0,282</td>
</tr>
<tr>
<td>10</td>
<td>0,693</td>
<td>0,280</td>
</tr>
<tr>
<td>17</td>
<td>0,723</td>
<td>0,294</td>
</tr>
<tr>
<td>24</td>
<td>0,713</td>
<td>0,290</td>
</tr>
<tr>
<td>30</td>
<td>0,711</td>
<td>0,288</td>
</tr>
</tbody>
</table>
3D camera calibration

- Forward projection yields quantitative results that are meaningful also when all parameters (including resolution) are averaged.

- The error (< 0.3mm) with both libraries is comparable with the accuracy of the ground truth (≈ 0.15mm), so they are practically equivalent.
2D object detection: untextured, orthogonal view, single instance

<table>
<thead>
<tr>
<th>Library</th>
<th>AUC over regular images</th>
<th>Overall AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>0.3 MP</td>
</tr>
<tr>
<td>VisionPro</td>
<td>0.978</td>
<td>0.963</td>
</tr>
<tr>
<td>MIL</td>
<td>0.945</td>
<td>0.883</td>
</tr>
<tr>
<td>HALCON</td>
<td>0.886</td>
<td>0.914</td>
</tr>
</tbody>
</table>

Effects of nuisances and camera resolution

- Based on AUC!
- “Regular images” is a dataset of real images, under good lightning conditions, with no addition of synthetic nuisances
- Basic images in different datasets are different
- 5Mp cameras are perceptibly very noisy
2D object detection: untextured, perspective view, single instance

- How likely in industrial applications?
  - Previous image rectification based on 2D calibration is usually performed
- MIL doesn’t provide a specific method to handle perspective deformation
2D object localization:
untextured, orthogonal view, single instance

<table>
<thead>
<tr>
<th></th>
<th>VisionPro</th>
<th>MIL</th>
<th>Halcon</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE in x-position (pixels)</td>
<td>0.95</td>
<td>0.83</td>
<td>1.01</td>
</tr>
<tr>
<td>RMSE in y-position (pixels)</td>
<td>0.93</td>
<td>0.79</td>
<td>1.00</td>
</tr>
<tr>
<td>RMSE in rotation (degrees)</td>
<td>0.12</td>
<td>0.17</td>
<td>0.11</td>
</tr>
<tr>
<td>RMSE in scale</td>
<td>0.008</td>
<td>0.012</td>
<td>0.006</td>
</tr>
<tr>
<td>Average Accuracy in x-position (pixels)</td>
<td>0.97</td>
<td>1.11</td>
<td>0.91</td>
</tr>
<tr>
<td>Average Accuracy in y-position (pixels)</td>
<td>0.98</td>
<td>1.10</td>
<td>0.90</td>
</tr>
<tr>
<td>Average Precision (pixels)</td>
<td>0.05</td>
<td>0.49</td>
<td>0.36</td>
</tr>
</tbody>
</table>

RMSE = Root Mean Square Error
x- and y- RMSE based on centroid
Accuracy and precision based on vertices of the bounding box

• Based on a data subset with no nuisances added
• This configuration represents the most common case in industrial applications
• In case, previous image rectification assumed
What next?

• How can we get vendors **really involved**?
  • Increasing the relevance of the benchmark: journals, conferences, exhibitions ...
  • Making the benchmark a web accessible, development tool
• Building a community
• Some technical chores
  • Reviewing challenges
  • New challenges
Thank you for your attention

- T3LAB
  via Bassanelli 9/11
  40129 Bologna
  http://www.t3lab.it

- VIALAB
  http://www.progetti.t3lab.it/vialab