A Multi-Tool for Ground-Truth Stereo Correspondence, Object Outlining and Points-of-Interest Selection

Boguslaw Cyganek
AGH University of Science and Technology
Al. Mickiewicza 30
30-059 Kraków, Poland
cyganek@agh.edu.pl

Katarzyna Socha
The Strata Mechanics Research Institute
Polish Academy of Sciences
Reymonta 27, Krakow30-059, Poland
socha@img-pan.krakow.pl

ABSTRACT

In this paper we present architecture and functionality of the visual multi-tool designed for acquisition of ground-truth and reference data for computer vision experimentation. The multi-tool allows three main functions, namely manual matching of the corresponding points for multi-view correlation, outlining of the image objects with polygons, as well as selection of characteristic points in specific image areas. These functions allows gathering of experimental data which are used for training and/or verification in such computer vision methods as stereo correlation, road signs detection and recognition, as well as color based segmentation. We present overview of the experimental results which were made possible with this multi-tool, as well as we discuss its potential further applications and extensions. The presented software platform was made available on the Internet.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous; D.2.8 [Software Engineering]: Metrics—complexity measures, performance measures

General Terms

Experimentation

Keywords

Ground-truth stereo, object recognition, segmentation, user interfaces.

1. INTRODUCTION

Acquisition of experimental data is a prerequisite for majority of scientific research. However, in many cases this means long and tedious process. In Computer Vision reference data are almost always referred to the Human Visual Systems which shows performance unattained to the artificial systems. This means however, that acquisition of such reference data involves human activities. Thus, depending on the task the ground-truth data is obtained from user operated platforms. However, to make this process as facile as possible suitable software platforms need to be developed.

In this paper we present and discuss an application, called a multi-tool, aimed at ground-truth data acquisition. Our platform allows three types of functionality

- Gathering point correspondences in the multi-view matching task;
- Outlining objects in images with polygons;
- Collecting a large number of points in specific image areas;

All these functions were used in a number of CV experiments, as described in [1][2][4]. These will be also outlined in this paper.

The paper is organized as follows. Section 2 presents architecture of the multi-tool. In section 3 its functionality is discussed, divided into three sub-sections, each devoted to different category of functions related to CV. In this section we also briefly outline experiments conducted with the presented multi-tool. The paper ends with conclusions and bibliography.

2. SYSTEM ARCHITECTURE

Figure 1 depicts architecture of our software framework. It is based on the OS API (Windows in our case). Upon this there are two libraries used. The first one is the Hardware Image Library (HIL) which implements object-oriented image processing and computer vision base algorithms. The library is described in [3]. The second supporting component is responsible for user's graphical interface widgets. In our framework we used the Microsoft Foundation Classes, which is common choice when using C++ and Windows [8]. However, other libraries such as Qt, FLTK, WPF, etc. can be also useful [6][8]. Nevertheless, choice of the programmers platform for GUI usually depends on the operating system and available libraries.

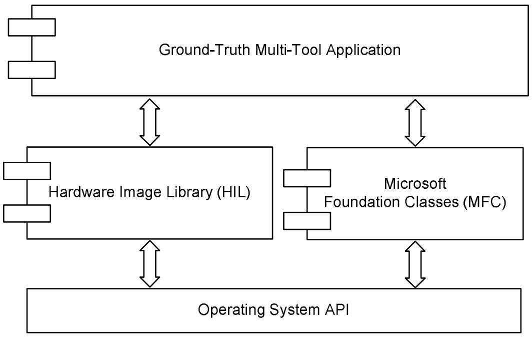

Ground-Truth Multi-Tool Application
Hardware Image Library (HIL)
Microsoft Foundation Classes (MFC)
Operating System API

Figure 1. Architecture of the software framework.

VIGTA '12, May 21 2012, Capri, Italy
Copyright 2012 ACM 978-1-4503-1405-3/12/05…$10.00.

The problem of design methodology of user interfaces gains much attention [8][9][10]. In the context of applications for manual image annotation the main problems are as follows

- Tiredness of operating persons;
- Precision of selections;

We believe that these are very important factors which frequently are not stressed enough in implementations of scientific tools. Therefore these should be accounted for in the GUI design. In our framework we implemented the following features which address the above issues:

- Each step can be undone; This makes a possibility to correct errors;
- At each stage the work can be saved, and then recovered and continued in a later time;
- The views allow image magnification;

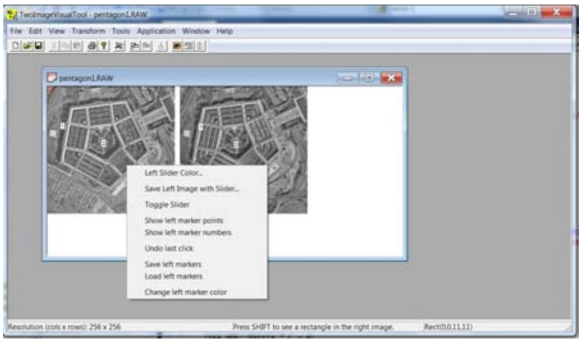
Thanks to the above it is possible to obtain high accuracy of manual point selections, although human errors are inevitable. The goal is to keep them as low as possible.

3. MULTI-TOOL SYSTEM FUNCTIONALITY

3.1 Multi-View Point Correspondence

In stereo and multi-view processing the key task is matching of the corresponding points among the views. Based on their coordinates the multi-view epipolar geometry in tensor representation can be computed [5]. This, in turn, helps in finding depth information in the observed scene [3]. However, all computations depend on a quality of matches. To evaluate algorithms aimed at this task it is useful to have some ground-truth point correspondences. For this purpose in the multi-tool we included a feature of manual point matching in the multi-views.

Figure 3 shows the multi-tool in the stereo correspondence mode. In this case a pair of stereo images is loaded and the operator selects a number of point correspondences. These can be saved or used for further processing, such as computation of the fundamental matrix, multi-view tensor, etc. Available functions are visible in the pop-up menu.



Resolution (cols x rows): 256 x 256

Press SHIFT to see a rectangle in the right image.

Rect(0,0,11,11)

Figure 2. General view of the multi-tool with stereo pair and context menu for selection and saving chosen points.

Figure 6 depicts a well known "Tsukuba" stereo pair from Tsukuba University [11]. The point correspondences can be chosen arbitrarily. Nevertheless, for the sake of computational stability it is better to choose corresponding pairs from different parts of a stereo pair. Selected points are assigned consecutive numbers. Thus we have correspondences 1-1, 2-2, 3-3, etc.

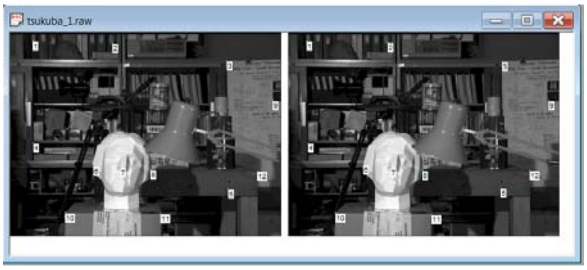


Figure 3. A stereo pair with manually selected points. These can be used to compute a reference epipolar geometry.

Naturally, the selected pairs can be saved to a text file.

3.2 Object Outlining

The other frequent CV task is object detection. For this operation a computer algorithm should correctly find in an image an object based on some specification, such as shape, color, texture, or other formal models [5]. Evaluation of such algorithms is possible if some ground-truth data is available. Usually this comes in a form of an image database with manually outlined objects of interest, such as persons, faces, cars, etc.

In the reported multi-tool manual object outlining can be done with two slightly different features:

- Extendable rectangular tool;
- A polygon tool;

These can serve the same purpose but have a bit different properties, especially for the operator.

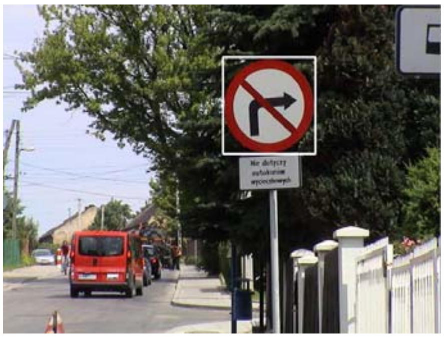


Figure 4. A road sign manually outlined with an extendable rectangle (white) which is simpler in manipulation than a polygon. This is saved always with four values.

Figure 4 depicts an image of a road with visible "no right turn" sign. In this case the sign was outlined with the expandable rectangle tool. When used for selection, first the top left corner is set upon an object, and then pressing *Ctrl* button the right bottom corner is dragged by an operator to tightly outline the object of interest. If *Ctrl* button is not pressed, then the tool can be grasped and dragged to new location. However, its operation is very simple and only four integers are necessary to describe the selected region. However, for some more complicated objects, the polygon tool can be more appropriate.

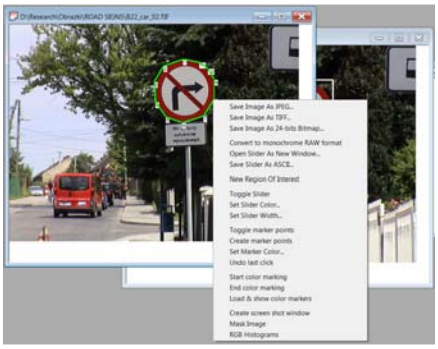


Figure 5. A road sign manually outlined with a polygon. This requires a number of points to be saved. A context menu allows a number of options.

Figure 5 shows the same object (road sign) selected with the polygon tool. The operator clicks upon consecutive points, and each time a polygon is drawn upon an object. The number of points depends on the operator and naturally on complexity of an object, such as an ellipse in Figure 5, or triangle in Figure 6.

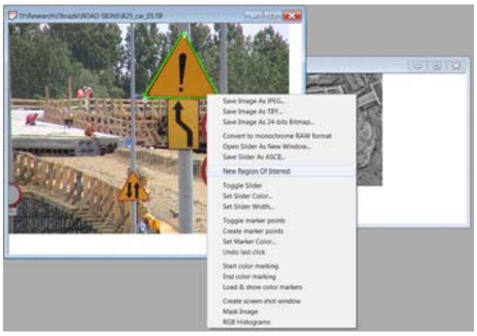


Figure 6. An image with manually outlined road sign (green frame around the yellow sign). For this purpose the polygon is used which local coordinates of points are saved. Visible context menu with a number of options.

For some more complicated objects, such as human faces shown in Figure 7, the polygon tool is the best option.

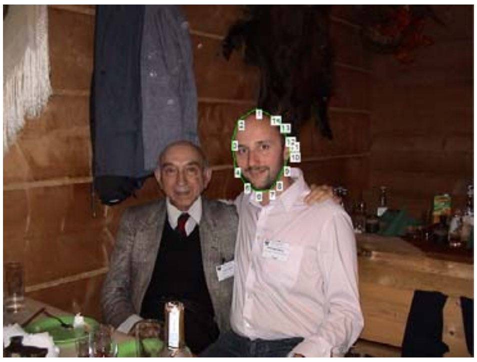


Figure 7. The multi-tool used for manual face selection with a polygon.

Once again, all the selected polygon points can be saved and then re-loaded or used in testing performance of the algorithms for automatic object detection. Point coordinates are local and related to the upper left corner of an image.

3.3 Point Sampling

The third functionality - called point sampling - was designed to allow selection of multiple points from a user's selected regions, objects, etc. These points can be used to train classifiers or to infer some knowledge on the marked areas of images

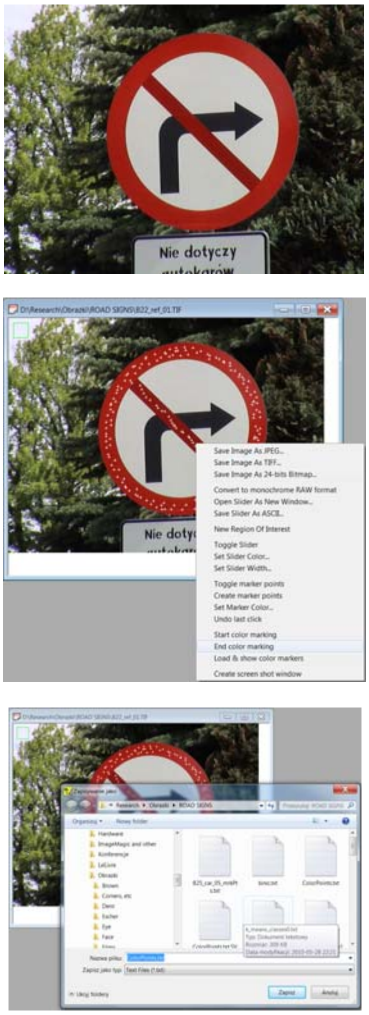


Figure 8. Multi-tool in a function of image point sampling. Both point coordinates, as well as their color values can be stored for further processing. The points can be reloaded.

Figure 8 shows the multi-tool in the point sampling mode. Both point coordinates, as well as their color values are stored for further processing. Also, the points can be reloaded and/or supplemented with newly selected ones.

For the points marked in a road sign presented in Figure 8 some of their color values in the color spaces RGB, HSI, and YCrCb are shown in Figure 9. Depending upon experiment, these can be used to train a classifier, such as the one-class support vector machines (OC-SVM), described in [2]. These were also used to compute separate histograms which can be also used for object segmentation, as described in [1].

RGB	HSI	YCrCb
186 123 17	27 215 108	129 -63 40
186 122 24	26 199 110	129 -59 39
183 124 24	27 199 110	130 -59 37
188 124 26	26 196 112	131 -59 39
185 125 29	26 189 113	131 -58 37
188 125 22	26 204 111	132 -62 39
186 121 27	25 193 111	129 -57 40
183 124 22	27 203 109	130 -60 37
188 124 26	26 196 112	131 -59 39
186 122 24	26 199 110	129 -59 39
188 125 22	26 204 111	132 -62 39
202 149 57	27 148 136	154 -54 33
187 149 50	31 155 128	149 -55 27
196 147 55	28 149 132	151 -54 31
187 138 46	28 160 123	142 -54 31
190 141 48	28 158 126	145 -54 32
181 134 54	27 143 123	138 -47 30
181 131 44	27 160 118	136 -51 32
184 135 30	29 189 116	137 -60 33
208 137 55	22 149 133	148 -52 42

Figure 9. Few first values of colors in three spaces at points marked by a user and saved to a file.

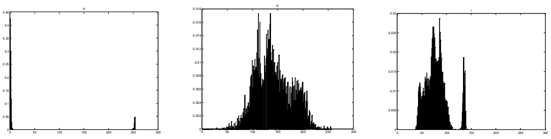


Figure 10. Histograms of the H, S, and I components of the prohibitive road signs from the gathered color samples.

In the experiments described in [2], all data were gathered for the dominating colours of the Polish road signs. These are: red – for prohibitive and warning signs, blue – for prohibitive sign B-35, and for the information and obligation signs, yellow – for the warning signs, and green – for the information signs.

The histograms presented in Figure 10 were created from the manually gathered data. Their purpose is at least twofold. Firstly, they allow assessment of the distributions for different colors of the (mostly Polish) road signs and in different color spaces. Secondly, they allow derivation of the border values for segmentation based on the simple thresholding. Although not perfect, this method is very fast and is often preferable for many machine vision tasks (e.g. due to its simple implementation). It works quite well in the road signs recognition system.

Based on the histograms in Figure 10 we observe that road signs detection based on threshold values on characteristic colors are work the best in the HSI space. This gives a very intuitive insight into the color representation and allows fast computations. However, usually it requires prior conversion from the RGB space. From these histograms we derived the empirical threshold values for the H and S channel for four colors encountered in the Polish road signs from every group, which can be accessed in [1].

4. CONCLUSIONS

In this paper a multi-tool application is presented which allows three types of ground-truth data collection for the purpose of training and/or testing CV methods. The first functionality allows manual selection of point correspondences in stereo images. The second functionality allows object outlining with the rectangle or polygon tools, respectively. The third functionality consists in selecting multiple points in regions of interest. The points and their corresponding color values can be then used to infer some additional information of the selected regions or objects.

There is an ongoing work an application of similar functionality available on mobile devices endowed with the Android OS. This would open new possibilities of gathering images in a terrain and then transferring them with potentially outlined objects of interest to the server for further storage or processing. The application is aimed mostly at processing of traffic images but some other applications can be also envisaged.

5. ACKNOWLEDGMENTS

The work was supported in the year 2012 from the funds of the Polish National Science Centre NCN, contract no. DEC-2011/01/B/ST6/01994.

6. REFERENCES

[1] Cyganek, B. 2007. Circular Road Signs Recognition with Soft Classifiers. Integrated Computer-Aided Engineering, IOS Press, Vol. 14, No 4 /, 2007, 323 - 343.

[2] Cyganek, B. 2008. Color Image Segmentation with Support Vector Machines: Applications To Road Signs Detection. Int. Journal of Neural Systems, Vol. 18, No. 4, World Scientific Publishing Company, 2008, 339–345.

[3] Cyganek, B., Siebert J.P. 2009. An Introduction to 3D Computer Vision Techniques and Algorithms, Wiley.

[4] Cyganek, B. 2012. One-Class Support Vector Ensembles for Image Segmentation and Classification. Journal of Mathematical Imaging & Vision, Vol. 42, No. 2-3, Springer 103–117.

[5] Fisher, R., Dawson-Howe, K., Fitzgibbon, A., Robertson, C., Trucco E. 2005. Dictionary of Computer Vision and Image Processing, Wiley.

[6] http://www.fltk.org/

[7] http://home.agh.edu.pl/~cyganek/CVMultiTool.zip

[8] http://msdn.microsoft.com/en-us/

[9] Johnson, J. 2011. Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules.

[10] Tidwell, J. 2011. Designing Interfaces. O'Reilly Media.

[11] Scharstein, D., Szeliski, R. 2002. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. Int. J. of Computer Vision, Vol. 47, No. 1, 7-42.