Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 52-9 (2019) 127–132

Computer Vision in Control and Robotics for Educational Purposes ⋆

Sergei V. Shavetov * Isabella I. Merkulova * Anton A. Ekimenko * Oleg I. Borisov *,** Vladislav S. Gromov *,**

* *ITMO University, Faculty of Control Systems and Robotics, Saint Petersburg, Russia (e-mail: s.shavetov@corp.ifmo.ru).*
** *Center for Technologies in Robotics and Mechatronics Components, Innopolis University, Innopolis, Russia*

Abstract: The paper is devoted to computer vision education for ITMO University students specializing in control systems and robotics. The set of laboratory works are attached. Several examples of finished works are provided. The topics of the laboratory course consists of histograms, profiles, projections, image geometric transformations, filtering and edges detection, images segmentation, Hough transformation, image morphologic analysis.

Keywords: Computer vision, Laboratory education, Computer applications, Robots sensing systems, Control systems applications.

1. INTRODUCTION

Realization of most control systems is impossible without various sensors. Nowadays, control systems using one or several cameras and video processing together instead of sensors become more popular. The popularity growth is due to increasing computers and controllers calculation power, information quality, cameras miniaturization, new image and video processing algorithms, quantity of different labeled images datasets (e.g. KITTI Geiger et al. (2013), PASCAL VOC Mark Everingham (2012), Caffe2 (2019), etc.) for various object classes, neural network training using datasets, and machine learning.

One of the examples of control system with computer vision (CV) for robotic system is robot ABB Flexpicker. The control system, using computer vision, detects different products on the conveyor, and robot grips it for sorting, see Connolly (2007). The second example of implementation computer vision to control systems is unmanned vehicles (cars, trains, drons, etc). This process started a couple decades ago and growing rapidly in recent years. Janai et al. (2017).

We are specialists in developing of control systems, for example, see Bobtsov et al. (2009), Pyrkin et al. (2015), and Pyrkin and Bobtsov (2016). But modern control and robotics tasks require new skills in an area of computer vision. In our robotics lab some computer vision tasks were performed by students during laboratory work realization on remote control, see Shavetov et al. (2016). But it is hard to solve practical tasks without theoretical training. So, we decided to create our own theoretical and practical course on computer vision specially for the students who study at the ITMO University on Robotics and Control specializations. For practical part of this course six laboratory works were developed, see Shavetov (2018). The purpose of this course is a balance between depth of computer vision studying and necessity for students to study control and robotics. The practical course consists of six tasks covering basic concepts of computer vision.

Works are performed by using the MATLAB Computer Vision Toolbox. Students can work in any development environment that is convenient for them, such as Aforge on C#, OpenCV on C++, JavaCV on Java, IMAQ Vision on LabVIEW, etc. A student with an interest in this topic can perform a complex project combining several of these topics.

The rest of the paper consists of each laboratory task, theory block and examples of it realization.

2. LABORATORY WORKS

2.1 Histogram, profiles, projections

A lot of operations with digital images can be performed without complex algorithms. But in this case we should provide corresponding weak conditions, like homogeneous background, single or split objects, soft light, etc. For number of different tasks it is enough.

Theory. An each pixel of the digital image has three parameters: (x, y, I), where pair of integer numbers (x, y) describes pixel geometric position on the image plane, and value I characterizes pixel brightness. It is convenient to investigate the image using brightness parameter I and geometric parameters (x, y) separately. Due to this approach the investigated image order is decreased from $n = 3$ (x, y, I) up to $n = 2$ (x, y) in case of studying geometrical properties, and up to $n = 1$ (I) in case of studying brightness properties.

⋆ This work was supported by the Ministry of Education and Science of Russian Federation (Project 2.8878.2017/8.9) and the Government of Russian Federation (grant 08-08).

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2019.08.136

The brightness component of the digital image is characterized by one-dimensional array of *histogram.* Histogram is the frequency distribution of same brightness pixels occurrence in the image. Using image histogram we can correct brightness characteristics, stretch or compress the dynamic range, make some transformations. Let's consider typical task of stretch dynamic range for low contrast image. We can stretch it using a formula (1).

$$I_{new} = \left(\frac{I - I_{min}}{I_{max} - I_{min}}\right)^{\alpha}, \tag{1}$$

where I and I_{new} — brightness arrays of source and new images respectively; I_{min} and I_{max} — minimum and maximum brightness values of source image correspondingly; α — nonlinearity coefficient.

To reduce geometric components (x, y) to one-dimensional array $(n = 1)$ we can use *profiles* and *projections* of the image. The image profile along a certain line is the image brightness function distributed along the given line. The projection of the image onto a certain axis is called the pixels brightness sum of the image in the direction perpendicular to this axis. Row profile is:

$$Profile\ i(x) = I(x, i), \tag{2}$$

where i — image row number I. Column profile is:

$$Profile\ j(y) = I(j, y), \tag{3}$$

where j — image column number I. For the result see fig. 1.

The projection simplest case of a two-dimensional image is the vertical projection on the Ox axis, which is the brightness sum of the image pixels *by columns*:

$$Proj\ X(y) = \sum_{y=0}^{\dim Y - 1} I(x, y), \tag{4}$$

and the horizontal projection on the Oy axis, which is the brightness sum of the image pixels *by rows*:

$$Proj\ Y(x) = \sum_{x=0}^{\dim X - 1} I(x, y). \tag{5}$$

If there are contrasting objects on the image, then in the projection there will be visible function drops or extremes corresponding to the each object position. An example of a machine-readable document projection onto the Oy axis is shown in fig. 2. Two machine-readable text lines generate two significant projection function extremes. Such projections can be used for detection and segmentation simple objects.

Task.

(1) Choose an arbitrary low contrast image. Perform histogram alignment and stretch;
(2) Select an arbitrary image containing monotone regions and single or splitted objects. Make projection on the axes, calculate the object boundaries;
(3) Select an arbitrary image containing a bar code. Make projection along the bar code and calculate it.

Example. There are examples of the performed first laboratory work except of first task of three are shown in Fig. 1.

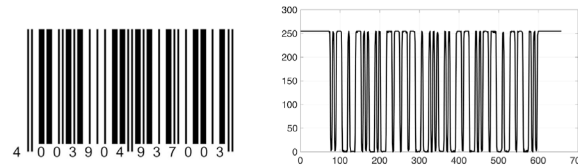


Fig. 1. Bar code profile

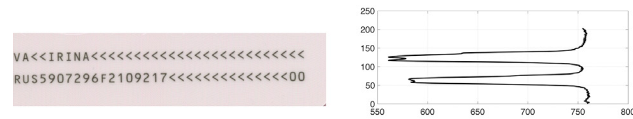


Fig. 2. Text strings projection

Advantages and skills. Students can perform searching simple disjoint objects without using complex algorithms required high computational power.

2.2 Image geometric transformations

Image geometric transformations imply a spatial location changing of a pixels set with coordinates (x, y) to another set with coordinates (x', y') with pixels brightness preserving. Typical tasks are distorted image correction, image stitching, etc.

Theory. For further transformations we will use *homogeneous coordinates.* Due to using these coordinates, object does not changes if all coordinates are multiplied to the same non-zero number. So, the required coordinates number for the represent points is always one more than the space dimension $\mathbf{P}^n$ where these coordinates are used. For example, to represent a point $X = (x, y)$ on a plane in the two-dimensional space $\mathbf{P}^2$, it is need to use three coordinates $X = (x, y, w)$. Let's illustrate it with the following example:

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = w \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Leftrightarrow \begin{bmatrix} x' & y' & w \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} w, \tag{6}$$

where w — scalar arbitrary factor, $x = \dfrac{x'}{w}$, $y = \dfrac{y'}{w}$. Using homogeneous coordinates triples and third-order matrices, any linear transformation of the plane can be described. Pixels coordinates of the transformed and original images are related by the following matrix relation $X' = XT$ (in a row form) or $X' = T^{\mathrm{T}}X$ (in a column form). Write out these matrix relations in a row form:

$$\begin{bmatrix} x' & y' & w' \end{bmatrix} = \begin{bmatrix} x & y & w \end{bmatrix} \cdot \begin{bmatrix} A & D & G \\ B & E & H \\ C & F & I \end{bmatrix}. \tag{7}$$

The point with Cartesian coordinates (x, y) in homogeneous coordinates is written as $(x, y, 1)$:

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \cdot \begin{bmatrix} A & D & 0 \\ B & E & 0 \\ C & F & 1 \end{bmatrix} \tag{8}$$

or in the form of equations system:

$$\begin{cases} x' = Ax + By + C, \\ y' = Dx + Ey + F. \end{cases} \tag{9}$$

Using this matrix formalism we can perform linear or nonlinear transformations. In case of nonlinear projective transformation the equation system and matrix T have the form:

$$\begin{cases} x' = \dfrac{Ax + By + C}{Gx + Hy + I}, \\ y' = \dfrac{Dx + Ey + F}{Gx + Hy + I}, \end{cases} \Rightarrow T = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & 1 \end{bmatrix}. \quad (10)$$

In other terms we can rewrite (9) in a polynomial form. For the second order:

$$\begin{cases} x' = a_1 + a_2x + a_3y + a_4x^2 + a_5xy + a_6y^2, \\ y' = b_1 + b_2x + b_3y + b_4x^2 + b_5xy + b_6y^2, \end{cases} \quad (11)$$

where $a_1 \dots a_6$, $b_1 \dots b_6$ — transformations coefficients. It is necessary to point t_{min} pairs the same pixels on the two images to transfer the image from one coordinate system to the second image coordinate system in case of unknown parameters $a_1 \dots a_6$, $b_1 \dots b_6$:

$$t_{min} = \frac{(n+1)(n+2)}{2}, \quad (12)$$

where n — transformation order. If we have second order polynomial transformation and t_{min}-pairs the same pixels coordinates are known we can find transformation parameters using matrix form of (11). Using the calculated parameters we can recalculate pixels and transfer it from the first image coordinate system to the second image coordinate system. For performing two images automatic "gluing" into the one, we may calculate correlation function for each row or column of both images in the one coordinate system, for example.

And the last point of the work is "distortion correction". Typical distortions are "cushion distortion" with positive distortion sign and "barrel distortion" with the negative sign. Let $\boldsymbol{r} = (x, y)$ is a vector defines two coordinates in a plane perpendicular to the optical axis. For an ideal undistorted image, all rays passed through the optical system will fall into an image point with coordinates $\boldsymbol{R}$ as follows:

$$\boldsymbol{R} = b_0\boldsymbol{r}, \quad (13)$$

where b_0 — linear scale parameter. In case of higher order distortion (distortion can only be of odd order (third, fifth, seventh, etc) for axisymmetric optical systems) obtain:

$$\boldsymbol{R} = b_0\boldsymbol{r} + F_3r^2\boldsymbol{r} + F_5r^4\boldsymbol{r} + \dots, \quad (14)$$

where r — vector length $\boldsymbol{r}$; $F_i, i = 3, 5, \dots, n$ — n-order distortion parameters. In case of $n = 3$ order distortion and if $sign(F_3) = sign(b_0)$ is cushion distortion and vice versa — barrel distortion. For the distortion correction we can use the similar approach described above.

Task.

(1) Choose an arbitrary image. Perform conform, affine, and nonlinear transformations;
(2) Choose an arbitrary image, either with cushion or barrel distortion. Perform image correction;
(3) Select two images (snapshots from the camera, scanned image fragments, etc.) with intersection areas. Transform the second image and transfer coordinate system to the first one; perform two images automatic "gluing" into the one.

Example. There is an example of the performed second laboratory work, the two images automatic "gluing" is shown

Fig. 3. Left — itmoTop.jpg, center — itmoBot.jpg, right — result_img

Advantages and skills. Students can perform image distortion correction and solved frequently used task "gluing" images.

2.3 Filtering and edges detection

Digital images may contain distortions variety caused by different interference kinds, usually called *noise*. Noise in the image may have a different structure. For its successful suppression it is necessary to determine an adequate noise mathematical model. Usually are used low frequency filters or nonlinear approaches for it. High frequency filters enhance the high frequency components (areas with a strong brightness changes) and weak the low frequency image components. Usually are used to edges (contours) detection on images.

Theory. There are several types of noises: additive, multiplicative, gaussian (normal), quantization and speckle noises, and something else. Typical noise filetring approach is a "sliding window" method. Sliding window is also called "mask" and has the sizes 3×3, 5×5, etc. Filtering an I image with the dimension $M \times N$ using a mask $w(s, t)$ $m \times n$ dimension is described by the formula (15).

$$I_{new}(x, y) = \sum_s \sum_t w(s, t)I(x + s, y + t), \quad (15)$$

where s and t are items mask coordinates relatively mask center $(s = t = 0)$. These transformations are *linear.*

For noise suppression are used low frequency filters with the follows mask properties:

(1) non-negative mask coefficients;
(2) coefficients sum is equal to one.

There are some low filter mask 3×3 examples:

$$w = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, w = \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 20 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \quad (16)$$

One of the well-known generalized averaging filter is a "counterharmonic averaging filter":

$$I_{new}(x, y) = \frac{\sum_{i=0}^{m}\sum_{j=0}^{n} I(i, j)^{Q+1}}{m \cdot n \cdot \sum_{i=0}^{m}\sum_{j=0}^{n} I(i, j)^{Q}}, \quad (17)$$

where Q — order filter. With $Q > 0$ the filter suppresses noises "pepper", with $Q < 0$ — noises "salt". It is impossible using the filter suppress "salt and pepper" simultaneously. With $Q = 0$ the filter is "arithmetic", and with $Q = -1$ — "harmonic". The most famous low frequency filter is "gaussian" filter:

$$G_\sigma = \frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}} = \frac{1}{\sigma\sqrt{2\pi}}e^{\frac{-x^2}{2\sigma^2}} \cdot \frac{1}{\sigma\sqrt{2\pi}}e^{\frac{-y^2}{2\sigma^2}}. \quad (18)$$

The bigger σ the more blurring image. As a rule filter radius is $r = 3\sigma$. In this case mask size $2r + 1 \times 2r + 1$ and matrix dimension is $6\sigma + 1 \times 6\sigma + 1$.

High frequency filters enhance high frequency components (areas with strong brightness changes) and weak low frequency components. Using high frequency filter we can detect contours in the images. Typical high frequency filter properties for edge detection are:

(1) filter mask coefficients can be negative;
(2) coefficients sum is equal to zero.

There are several typical high frequency masks (differentials operators). The first one is Roberts filter with 2×2 mask. It is very fast and effective filter. Some mask variants for finding the gradient along the axes Ox and Oy respectively:

$$G_x = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}, G_y = \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix}. \tag{19}$$

Another one is Sobel filter with masks:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}. \tag{20}$$

One of the popular edge detection approach is a Canny algorithm. It allows to detect connected boundary lines and consists of several steps:

(1) Low frequency filtering using gauss filter;
(2) Using Sobel filter for detecting gradient modulus of each image pixel;
(3) Analysis modulus gradient: if the value more, than orthogonal pixels — pixel is an edge, vice versa — non-maximum;
(4) Performing double threshold filtering for edge pixels from the previous step;
(5) Suppresion of all ambiguous pixels using orthogonal and diagonal neighborhood.

Task.

(1) Choose an arbitrary image. Make noised images using different noise types;
(2) Correct noised images using low frequency filters;
(3) Apply nonlinear filters for noised images correction;
(4) Apply high frequency filters to source image for edges detection.

Example. There are examples of the performed third laboratory work, result of several high frequency filters are shown.

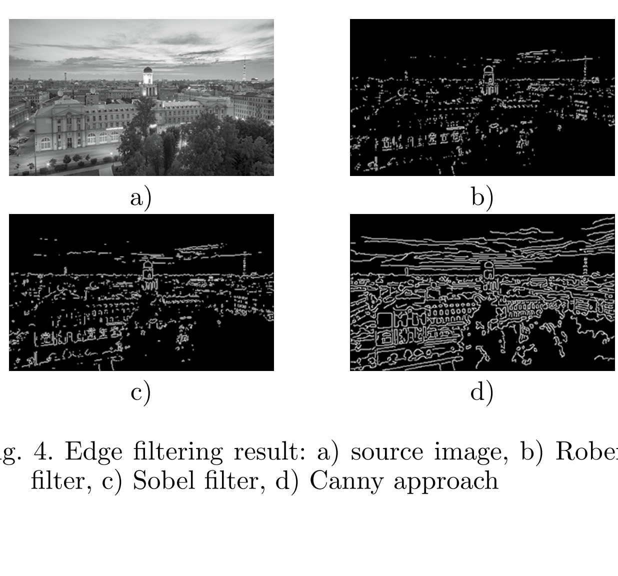


Fig. 4. Edge filtering result: a) source image, b) Roberts filter, c) Sobel filter, d) Canny approach

Advantages and skills. Students can perform low and high frequency images filtering to denoise images and edge detection.

2.4 Images segmentation

Segmentation is an operation of splitting image to several semantic areas by one or several criterions. It can be performed using a lot of methods and approaches, neural networks, machine learning, etc. For study basics we consider binarization (the simplest type of segmentation), k-means method, nearest neighbor approach and texture segmentation.

Theory. The simplest type of segmentation is binarization. It can be performed as threshold binarization (21) or double threshold binarization (22).

$$I_{new}(x, y) = \begin{cases} 0, I(x, y) \leqslant t, \\ 1, I(x, y) > t, \end{cases} \tag{21}$$

where I — source image, I_{new} — binarized image, t — binarization threshold.

$$I_{new}(x, y) = \begin{cases} 0, I(x, y) \leqslant t_1, \\ 1, t_1 < I(x, y) \leqslant t_2, \\ 0, I(x, y) > t_2, \end{cases} \tag{22}$$

where t_1 and t_2 — up and bottom binarization thresholds.

Nearest neighbor segmentation using colors implies transfer from RGB color space to CIE Lab typically. In the space the lightness component is separated from chromatic components. After that, we can use euclidean metrics $\sqrt{(a(x, y) - a_{mark})^2 + (b(x, y) - b_{mark})^2}^2$ for comparison distance between each pixels and pre-calculated labels. The minimum distance is corresponding to a right class.

In case of texture segmentation will consider image brightness I as a random value z with probability distribution $p(z)$ (can be obtain from image histogram). A *central moment* of order n of a random variable z is the parameter $\mu_n(z)$, calculated using the formula:

$$\mu_n(z) = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i), \tag{23}$$

where L — number of brightness levels, m — average value of random variable z:

$$m = \sum_{i=0}^{L-1} z_i p(z_i). \tag{24}$$

From (23) follows $\mu_0 = 1$ and $\mu_1 = 0$. To describe the texture it is very important to know *dispersion*. The dispersion is equal to $\sigma^2(z) = \mu_2(z)$ and is a measure of brightness contrast. To evaluate the textural features the *entropy* E function is used. It determines the scatter of the neighboring pixels brightness:

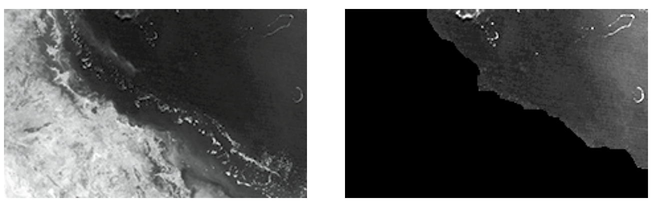
$$E = -\sum_{i=0}^{L-1} p(z) \log_2 p(z_i). \tag{25}$$

After calculating any feature or set of features, it is necessary to make a binary mask for image segmentation. For example, you can use the entropy E in the each pixel (x, y) neighborhood.

Task.

(1) Choose an arbitrary image. Perform binarization using different approaches for choosing threshold (average brightness, Otsu approach, etc.);
(2) Choose an arbitrary image with a finite colors number. Perform image segmentation using k-means or nearest neighbor methods;
(3) Choose an arbitrary image with two textures. Perform texture segmentation, define textures parameters.

Example. There is an example of the performed forth laboratory work consists of the texture segmentation result.



a) b)

Fig. 5. a) Source image, b) water texture and land mask

Advantages and skills. Students can perform simple images segmentation using different classical and frequently used approaches.

2.5 Hough transformation

The classical Hough transformation was originally intended to lines detection on binary images and based on the voting points idea. The Hough transformation uses the parameter space for searching geometric primitives. Generalized Hough transformation allows to find arbitrary image contours.

Theory. Let in the Cartesian coordinate system the straight line is given by the equation (26), from which it is easy to calculate the radius-vector ρ and angle Θ (27). Then in the Hough parameters space the straight line will be represented by a point with coordinates (ρ_0, Θ_0), see fig. 6.

$$y = kx + b, \tag{26}$$

$$x \cos\Theta + y \sin\Theta = \rho, \tag{27}$$

where ρ — radius-vector from origin to the line; Θ — radius-vector angle. In Hough transformation for each

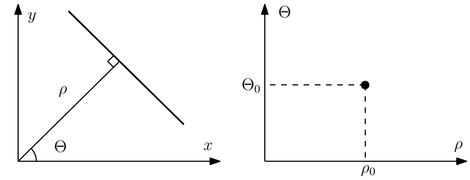


Fig. 6. Line in the Hough parameters space

parameter space point votes number for summing it up. Hough space is a matrix called an *accumulator* in a discrete form. It stores voting information of $A(\rho, \Theta)$. Through each point in the Cartesian coordinate system we can draw an infinite number of straight lines. These lines generate in the Hough parameter space a sinusoidal response function.

Two sinusoidal response functions in the Hough parameter space will intersect at the point (ρ, Θ) only if the points in the original space lie on a straight line, see fig. 7. To search for lines in the source space, it is necessary to find all local accumulator maximums.

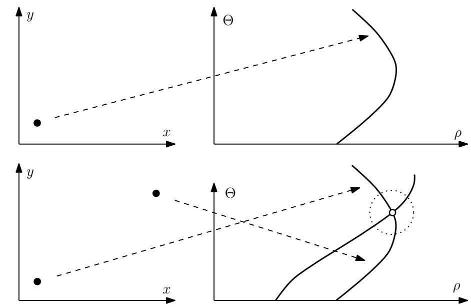
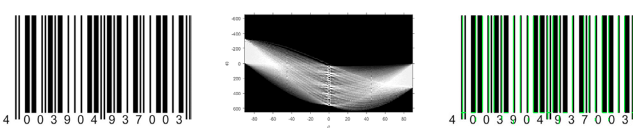


Fig. 7. Voting procedure

Task.

(1) Choose three arbitrary images with lines. Perform lines searching using Hough transformation, show found lines, point start and end lines pixels, find lines lengths, count lines quantity;
(2) Choose three arbitrary images with circles. Perform circles searching of given radius and from range of radiuses, show found circles.

Examples. There is an example of the performed fifth laboratory work. The result of Hough lines detection is shown.



a) b) c)

Fig. 8. a) Source image, b) Hough parameters space, c) detected lines

Advantages and skills. Students can perform geometric primitives searching and get its analytical describing.

2.6 Image morphologic analysis

Morphology is a science about forms. Morphologic analysis is an universal and powerful tool for a lot of computer vision applications. It can be used for filtering, segmentation, edge detection, searching given object, etc. This work quite different from previous ones and stands alone.

Theory. The basic morphologic operations are:

(1) Dilatation: $A \oplus B$, dilate binary figure A with structure element B;
(2) Eroding: $A \ominus B$, erode binary figure A with structure element B;
(3) Opening: $(A \ominus B) \oplus B$, removes binary figure external defects A with structure element B;
(4) Closing: $(A \oplus B) \ominus B$, removes binary figure internal defects A with structure element B,

where $\oplus$ and $\ominus$ are sum and subtraction of Minkowski respectively. Edges detection can be realized as follows:

(1) $C = A - (A \ominus B)$ — internal contour;
(2) $C = (A \oplus B) - A$ — external contour.

The task of splitting overlapped objects can be solved by consecutively performing several times eroding and then dilatation the result as much as possible. The intersection of the original and processed images allow to split the overlapped objects.

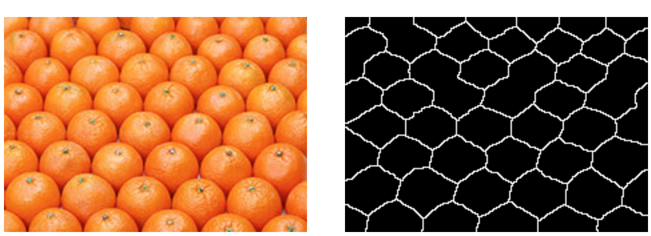
Another approach of segmentation is morphological watershed segmentation. In this approach the image is considered as a height map, and each pixel brightness describe heights relative to some level. Over the "place" it is "raining" forming a *drainage basins* multitude. Water from overflowing basins overflows, and the basins are combining into larger ones. Combined basins lines are marked as watershed lines. If the "rain" is stopped early, then the image will be segmented into small areas, and if it is late — into large ones. In this approach, all pixels are divided into three types:

(1) *local minimum*;
(2) *on the slope* (from which water rolls into the same local minimum);
(3) *local maximum* (from which water rolls into more than one minimum).

Task.

(1) Choose an arbitrary image with form defects. Perform morphological filtering for defects correction;
(2) Choose an arbitrary image with overlapping objects. Perform morphological objects splitting;
(3) Choose an arbitrary image. Perform watershed segmentation.

Example. There is an example of splitted overlapping objects.



a) b)

Fig. 9. a) Source image b) objects borders

Advantages and skills. Students can use morphological approach for solving different tasks of computer vision.

3. CONCLUSION

Described laboratory works are used for teaching of Control and Robotic ITMO University students. Two academic years of studying showed that these laboratory works are effective. In the future we plan to expand the course and complement laboratory works on modern methods and approaches of computer vision, research of detectors and descriptors (see Tuytelaars and Mikolajczyk (2008) and Krig (2016)), machine learning methods, several modern convolutional neural networks like R-CNN, Fast R-CNN Girshick (2015), Faster R-CNN Ren et al. (2015), Mask R-CNN He et al. (2017).

REFERENCES

Bobtsov, A., Pyrkin, A., and Kolyubin, S. (2009). Adaptive stabilization of a reaction wheel pendulum on moving lego platform. *Proceedings of the IEEE International Conference on Control Applications*, 1218–1223. doi: 10.1109/CCA.2009.5281045.

Caffe2 (2019). `https://caffe2.ai/docs/datasets`. [Online; accessed 16-February-2019].

Connolly, C. (2007). Abb high-speed picking robots establish themselves in food packaging. *Industrial Robot: An International Journal*, 34(4), 281–284.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

Janai, J., Güney, F., Behl, A., and Geiger, A. (2017). Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. *arXiv preprint arXiv:1704.05519*.

Krig, S. (2016). Interest point detector and feature descriptor survey. In *Computer vision metrics*, 187–246. Springer.

Mark Everingham, Luc van Gool, C.W.J.W.A.Z. (2012). Pascal voc. `http://host.robots.ox.ac.uk/pascal/VOC/`. [Online; accessed 16-February-2019].

Pyrkin, A. and Bobtsov, A. (2016). Adaptive controller for linear system with input delay and output disturbance. *IEEE Transactions on Automatic Control*, 61(12), 4229–4234. doi:10.1109/TAC.2015.2509428.

Pyrkin, A., Bobtsov, A., Nikiforov, V., Kolyubin, S., Vedyakov, A., Borisov, O., and Gromov, V. (2015). Compensation of polyharmonic disturbance of state and output of a linear plant with delay in the control channel. *Automation and Remote Control*, 76(12), 2124–2142. doi:10.1134/S0005117915120036.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99.

Shavetov, S.V. (2018). *Technical Vision: Laboratory Workshop*. ITMO University, Saint Petersburg, Kronverkskiy pr. 49, Russia.

Shavetov, S.V., Vedyakov, A.A., Pyrkin, A.A., Bobtsov, A.A., and Borisov, O.I. (2016). Advanced educational tool for remote control study. *IFAC-PapersOnLine*, 49(6), 303–308.

Tuytelaars, T. and Mikolajczyk, K. (2008). The examination of cell nuclei. *Foundations and Trends in Computer Graphics and Vision*, 3, 177–280.