Detection of Symmetric Features in Images

Karel Horak

Abstract—**Features detection plays a significant role in many computer vision tasks. There exist a plenty of various methods for features detection in an image processing domain. Symmetric features form a substantial class of them which takes an advantage of robust approach and fast implementation. In this paper a basic concept of a mirror and rotational symmetry is introduced and several implementation aspects are discussed. Symmetry detection process is transparently compared to the selected standard approach and results are demonstrated on the rotational symmetry task.**

Keywords—**Feature point, Mirror Symmetry, Point of Interest, Rotational Symmetry, Symmetry.**

I. Introduction

DETECTION of features in images is one of the most important and the most frequently used activity in order to understand content of an image. There was performed a significant research particularly on the field of so-called point of interest in the past [1], [2]. Several other concepts have emerged from time to time when better performance or higher accuracy was required [3]. The recent research attention has been paid for so-called symmetry constellations, which gives a new idea about detection of symmetrical points of interest [4], [5]. The basic concept of symmetry features detection is introduced in this paper besides implementation issues which are described at the end of the paper.

II. Concept of Symmetry Detection

The basic concept of symmetry detection in images of the real world is based on straightforward idea of matching symmetry couples of feature points. There is a plenty of robust and efficient methods for points of interest detection in the modern image processing area [6]. Symmetry importance of each pair is assessed on the basis of the position, orientation and optionally scale of the features. All obtained symmetries are accumulated in a voting space similar to the Hough transform where the major symmetries are localized.

A. Problem Background

As it was mentioned, many robust and effective methods for feature point detection have been developed and the most of them have already been implemented as well. These methods usually provide a dense set of feature points over the image. In the past a lot of effort has been dedicated only to matching feature points between individual images of a sequence. Pairing of feature points in only one isolated image was slightly suppressed in the past.

The methods for feature point detection often result in list of records related to detected points of interest. Each record represents one feature point and usually contains a foursome of values: image coordinates x and y, orientation ϕ and scale s of the point. The last two values of orientation and scale are normalised to ensure independence of these parameters. A set of orientation and scale invariant feature points detected in a robust and effective way, moreover with a high repeatability, is well suited input for symmetries detection [7]–[9].

There are the two major classes of symmetry in real world images. First of them is a so-called direct symmetry, which is related to the pairs of similar feature points only under translational and rotational transform. A simple example of the direct symmetry is a car wheel. The similar feature points are spread around the perimeter of the wheel. So-called mirror symmetry is the other kind of the general symmetry, which is related to such pairs of different feature points where the first feature point matches only with the mirrored version of the other one. Human faces and butterfly's wings are good examples of mirror symmetry, because almost each detected feature point has its own mirrored twin on the other side of the object. Fig. 1 shows both the direct symmetry and the mirror symmetry example.

A descriptor of the mirrored feature point is in [7] defined as a descriptor of mirrored copy of the local image patch associated with the original feature point. Matching pairs of feature points then generate a set of matched pairs for the next processing step. Following chapters discuss the details of detecting both the direct and the mirror symmetries.

B. Feature Points

A feature point is similar to the idea of point of interest, i.e. each feature point can be defined as a pixel with local patch very dissimilar to the nearest neighbourhood. The feature points are generally computed by means of some robust and rotationally invariant method with high repeatability such as well-known SIFT method [10]. In addition to feature point's coordinate values x and y, the orientation ϕ of each feature point has to be always determined in contrast to the scale value s, which is important only if some scale invariant method is used for feature points detection.

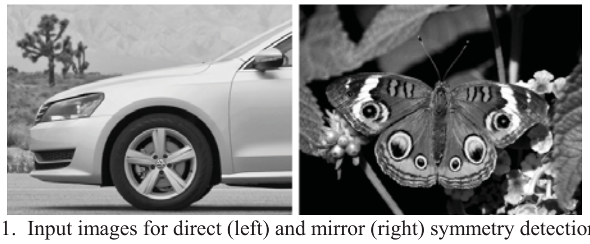

Fig. 1. Input images for direct (left) and mirror (right) symmetry detection.

Manuscript received February 28, 2013. This work was supported by the grant Research of Modern Methods and Approaches in Automation from the Internal Grant Agency of Brno University of Technology (FEKT-S-11-6).

K. Horak is with the Department of Control and Instrumentation, Brno University of Technology, Brno, 61600, Czech Republic (phone: 00420-5-4114-6417; fax: 00420-5-4114-6451; e-mail: horak@feec.vutbr.cz).

Nevertheless, assume the *i-th* feature point in the image contains all four values in general and so forms a point vector $\mathbf{p}_i = (x_i, y_i, \phi_i, s_i)$. The vector $\mathbf{p}_i$ describes unambiguously location, orientation and scale of the *i-th* feature point detected in the image. Further a descriptor usually denoted k_i is computed for each vector $\mathbf{p}_i$. These descriptors represent appearance of the local image patches around x_i and y_i coordinates. There are many feature descriptor methods designed for matching purposes and based on the very different theory in the literature [3], [6].

C. Mirror Symmetry

The mirror symmetry case is in [7] denoted as bilateral to emphasize similarity between a certain feature point $\mathbf{p}_i$ in an image and some second feature point $\mathbf{p}_j$ in the same image but flipped over given axis. It means two feature points are treated as a feature pair ($\mathbf{p}_i$, $\mathbf{p}_j$) only if original image patch of the first point matches with the mirrored image patch of the other one. There is only one true feature point $\mathbf{p}_j$ related to the other feature point $\mathbf{p}_i$ in the image containing an object of the mirror symmetry.

For matching purposes a mirrored feature descriptor m_i is computed for each known descriptor k_i. To obtain a mirrored version of the original image patch we can choose an arbitrary mirroring axis because of orientation normalisation. A construction of the normalized and mirrored versions of the original feature point is schematically shown in the following Fig. 2. Formation of the mirror feature m_i is only illustrative in the mentioned figure. In fact, there are two fundamentally different ways how to compute the mirrored feature descriptors m_i. The first approach is to simply flip the original image patch related to the descriptor k_i about *x* or *y* axis and compute additional (mirror) descriptor m_i. This approach is clearly very simple and naïve but it has not been always efficient.

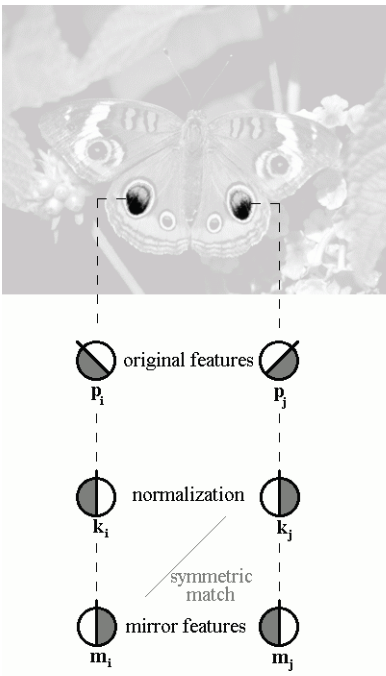

Fig. 2. The detection of the symmetric pair (p_i,p_j).

The other approach requires knowledge of a structure of the feature descriptor k_i, because the mirrored descriptor m_i is generated directly by modifying all the proper values in k_i. Clearly, this way is substantially more efficient in comparison to the previous one, but it is not so straightforward. Moreover it is usable only with the convenient descriptor's structure e.g. the SIFT fits this demands completely.

The next processing step is to recognize potentially symmetric features and to form a set of pairs ($\mathbf{p}_i$, $\mathbf{p}_j$) of feature points related to k_i and m_j descriptor respectively. Note that matching the descriptor k_i with the descriptor m_j gives the same result as matching the descriptor k_j against the descriptor m_i. It follows the pair ($\mathbf{p}_i$, $\mathbf{p}_j$) is always formally interchangeable with the reversed pair ($\mathbf{p}_j$, $\mathbf{p}_i$), because they are truly equivalent.

The amount of symmetry of arbitrary pair ($\mathbf{p}_i$, $\mathbf{p}_j$) is quantified by symmetry magnitude M_{ij} [7] defined as:

$$M_{ij} = \begin{cases} \Phi_{ij} \cdot S_{ij} \cdot D_{ij} & \text{if } \Phi_{ij} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where Φ_{ij} denotes an angular symmetry weight, S_{ij} denotes a scale weight and finally D_{ij} denotes a distance weight. In accordance with the previous equation, the symmetry value of each pair ($\mathbf{p}_i$, $\mathbf{p}_j$) is assessed as a function of the relative position of the feature points in the image, their orientation and scale. All the three weights newly introduced as the angular, scale and distance contributions are defined by the following equations as:

$$\Phi_{ij} = 1 - \cos\left(\phi_i + \phi_j - 2 \cdot \theta_{ij}\right) \tag{2}$$

$$S_{ij} = \exp\left(\frac{-\left|s_i - s_j\right|}{\sigma_s \cdot (s_i + s_j)} \right)^2 \tag{3}$$

$$D_{ij} = \exp\left(\frac{-d^2}{2\sigma_d^2} \right) \tag{4}$$

where angles ϕ_i and ϕ_j represent angles between main orientations of the feature points $\mathbf{p}_i$ and $\mathbf{p}_j$ and the *x* axis and the symbol θ_{ij} corresponds to an angle between a line joining $\mathbf{p}_i$ and $\mathbf{p}_j$ and the *x* axis and so it represents a relative orientation of the pair [7]. Concerning previous equation (3), the scale weight S_{ij} quantifies the relative similarity in scale of the two feature points. Finally, the letter *d* in the last equation of the distance weight stands for the distance separating the feature pair. It follows the distance weighting function D_{ij} penalizes matching pairs that are far off the symmetry axis and vice versa (i.e. favours such pairs that are closer to the axis).

The distance weight D_{ij} was originally motivated by psychophysical findings that symmetric features close to the symmetry axis play more important role in human perception of symmetry than outlying features [11]. In fact, from a computer vision point of view, it is not necessary to use parameter D_{ij} explicitly and so it can be completely omitted in the equation (1) or equally value of D_{ij} can be set to 1.

At this point each pair ($\mathbf{p}_i$, $\mathbf{p}_j$) of all detected feature points in the image has assigned corresponding symmetry magnitude M_{ij}. The higher similarity of the descriptors k_i and m_j are, the higher value M_{ij} becomes. It means the symmetry magnitude quantifies the strength of symmetry of an individual pair of feature points. Lastly a symmetry map can be generated by accumulating individual values of symmetry magnitude M_{ij} for all detected feature points. The accumulation process is performed in the Hough style of a voting space. Symmetry maps related to the two input images depicted in the Fig. 1 are shown in the next Fig. 3. There are three different images of symmetry maps for each input image in the figure below depending on the α parameter. The α value controls a sensitivity of an algorithm about a radial symmetry. The smaller α value is, the less sensitive algorithm is.

More specifically lower values of α accepts feature points of the mirror symmetry (i.e. bilateral) whilst higher values of α accepts strictly only the radial feature points. As you can see from the two sets of bottom images, α values higher than 3 have only small influence on symmetry map whilst smaller α values results in significantly different symmetry maps.

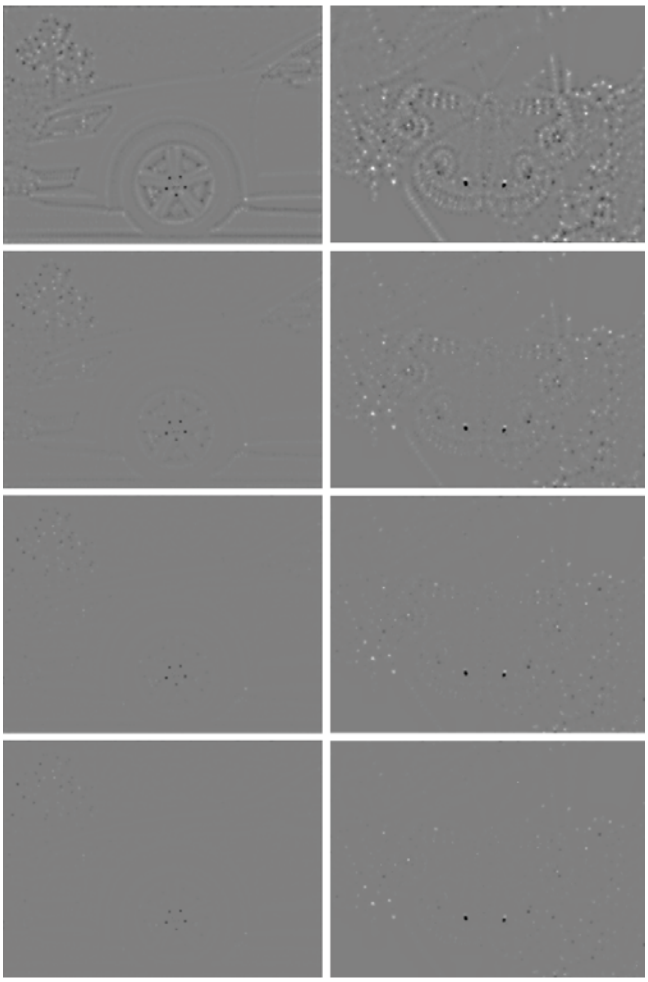

Fig. 3. The symmetry maps of the input images for α values of 1, 2, 3 and 4 from the top to the bottom respectively. The “wheel” input image relates to the left column and the “butterfly” input image to the right column.

The above mentioned butterfly image is the exemplary case of the mirror symmetry. Each pair of corresponding features in such image represents a small contribution of an imaginary symmetry axis. More explicitly the two dark points in the lower right image in the Fig. 3 unambiguously defines their mid-point, which lies on the major axis of the butterfly. In this way the dominant symmetry axis in the image can be computed by the linear Hough transform applied on all mid-points of the symmetry pairs. In fact each symmetric pair casts a vote in a Hough accumulator weighted by corresponding symmetry magnitude M_{ij}. The accumulator is processed in a proper way of a non-maximal suppression and dominant symmetry axes are then detected as isolated peaks.

D. *Rotational Symmetry*

A detection of rotational symmetry (somewhere radial) is slightly less complicated in comparison with the mirror symmetry, because there is no need to generate the mirrored version m_i of the original feature descriptor k_i [12]. The rotational symmetry is detected directly by matching each feature descriptor k_i against all remaining descriptors k_j. If normalized vectors of an arbitrary pair of the feature points ($\mathbf{p}_i$, $\mathbf{p}_j$) are not parallel there is a point about which they are rotationally symmetric. Further if more such pairs are present in the image they define a central point of the rotational symmetry. Formally it is done by omitting the θ_{ij} value in the equation (2) corresponding to an angle between the line joining $\mathbf{p}_i$ and $\mathbf{p}_j$ and the x axis.

Results of the rotational symmetry detection were shown in the previous Fig. 3 depending on the mentioned α value. Crucial issue for the next processing step is to obtain a binary representation of the symmetry map. Almost arbitrary segmentation method can be used because all rotational symmetry points are represented by significantly darker or brighter spots than the others points in the symmetry map.

The three binary representations of the previous images of symmetric maps are shown in the following Fig. 4. It is clear that several different threshold values ε were used to obtain more or less messy images with the rotational symmetry points. Similarly as in the previous case of mirror symmetry this binary representations can be used as input for the Hough transform. As we are interesting in the rotational symmetry detection we have to use, however, the circle Hough transform.

All the three accumulators of the circle Hough transform corresponding to the binary images in the Fig. 4 are depicted in the following Fig. 5. Processing of the accumulators are the same as in the previous case i.e. non-maximal suppression followed by the peaks detection. Number of peaks detected depends on implementation and is almost arbitrary. The main difference between Hough transform applied on the pure edge image (e.g. by the Canny edge detector) and the symmetry map is that in the first case all circles presented in the image are detected whilst only circles with the symmetrical features are taken into account in the other case. Because of the feature descriptors k_i and m_i computation and also of the Hough transform usage, the effectiveness of such image processing method for symmetry detection is not always enough sufficient.

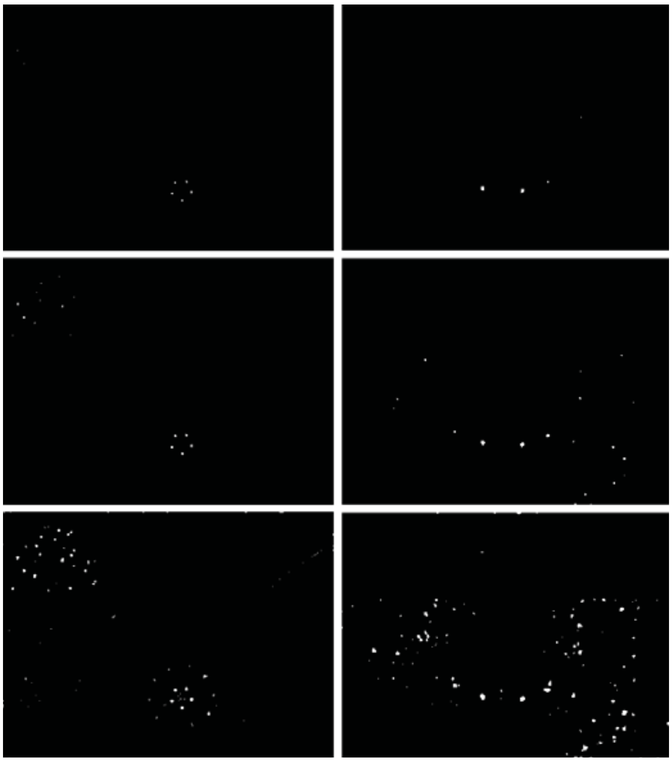

Fig. 4. The binary representations of the symmetry maps for thresholding ε values of 0.2, 0.3 and 0.4 respectively. The “wheel” input image relates to the left column and the “butterfly” input image to the right column again.

Fig. 5. Three accumulators of the circle Hough transform related to the three previous binary representations of the “wheel” input image.

There are rather more effective methods of so called “Fast” Hough transform in the image processing area, but this step is still the bottleneck of the processing chain. Due to computational problem mentioned in the previous paragraph, several aspects for process speed-up are discussed in the next two chapters.

III. Modification Towards Robustness

A robustness and stability of the symmetry detection process can be simply a crucial aspect in some application. For example scientific evaluation of the symmetrical features in organic substances is often very sensitive about noise in an image. In our case of the rotationally symmetric car-wheel in the Fig. 1, at least two evident approaches exist. First, a direct computation of the edge image by some of well-known methods can be used. It is not important whether the edge image was generated either by the simple Sobel operator or by the more robust Canny operator or even by some morphological operation. The differences between results of these methods are not significant for the next symmetry detection. More important is difference between first approach of the edge image computation and the other approach of the symmetry map generation.

Suppose the Canny edge detector as a convenient method for robust edge detection in our input image even though it is very time consuming way. Besides this edge image we can use the symmetry map with approximately same computational costs as an input of the Hough transform for circles detection. Both input images are shown in the next Fig. 6.

Among others a density and a structure of pixels in the input image affect the resulting accumulator of the Hough transform. Each one pixel in the input image increases energy of the output accumulator. The both accumulators of the Hough transform for circle detection related to the images in the Fig. 6 are shown in the next Fig. 7.

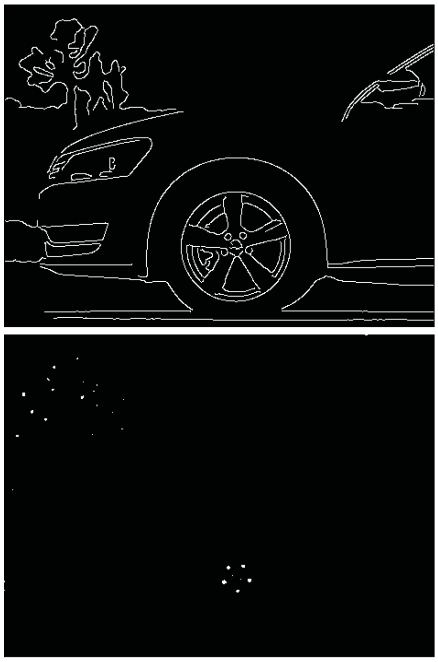

Fig. 6. The Canny edge image (left) and the symmetry map (right) as the alternative inputs of the Hough transform.

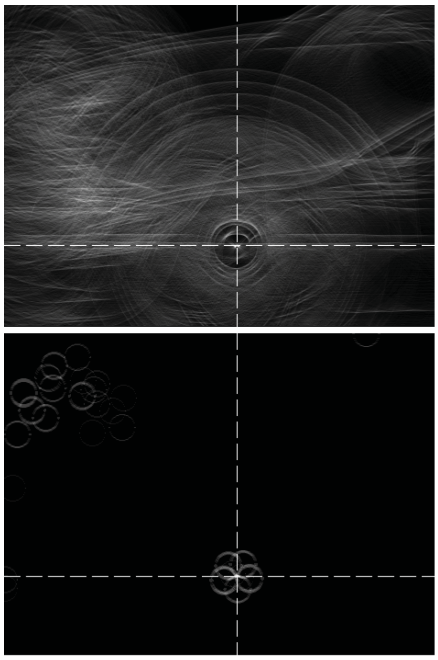

Fig. 7. The dense accumulator related to the Canny edge image (left) against

the sparse accumulator related to the symmetry map (right).

As can be seen from the previous figure, the accumulator based on the symmetry map (bottom image) is not so much cluttered as the accumulator based on the standard Canny edge detector (top image). Generally speaking, the more cluttered accumulator is, the more time next processing step spends. And furthermore, the more complicated is also recognition of individual (partially isolated) peaks in the accumulator. Moreover, because of the Hough transform is very time consuming in general, it is appropriate to generate a sparse accumulator as possible. It means the accumulator has to have a very low number of non-zero elements, because only such elements play role in the next step of seeking for maximum value and the rest elements of zero values are omitted. Similarly, also the accumulator matrix can be simply thresholded by considering only pixels of values over the predefined limit. In such case, finding of maximal element can be extremely speeded-up.

Using some additive “speed-up” mechanism or not, results have to be the same. Although very different images were used as the inputs of the Hough transform, the same peak was detected as the centre point of the car-wheel. Relevant peaks are depicted by a pair of dotted lines directly in the images of the both accumulators in the Fig. 7.

IV. Modification Towards Speed

A speed of a symmetry detector implementation generally depends on several different and principally independent aspects as well. Very crucial is a selection of an appropriate feature point descriptor. In order to achieve robust, rotationally and scale invariant descriptor we usually chose some well-known and often implemented method as the already mentioned SIFT (Scale Invariant Feature Transform) [10]. Nevertheless, to achieve a reasonable speed of the implementation some alternative descriptor method should be used instead of computational expensive one. For example the SURF (Speeded-Up Robust Features) method [13] suits to the both accuracy and speed, but it is not so simple to compare such different methods as the SIFT, SURF and some others. The SURF method takes advantage of an integral image [14], [15] which often satisfies time requirements. The detection of feature points is then performed on the basis of the Hessian matrix $H(\mathbf{x},\sigma)$ formed by a convolution of the input image and second derivation of the Gauss function $G(\sigma)$. Such similar approaches are significantly more effective but only for a limited set of applications.

Another aspect to speed-up the mentioned method for symmetry detection is to remove bottlenecks. In our case it should be goniometrical and exponential functions in equations (2), (3) and (4). This can be achieved by a priori computed look-up table with sufficiently dense set of values. For the circular symmetry detection, of course, is much more effective to use the symmetry map as the input of the Hough transform instead of the more general edge image.

V. Conclusion

The basic concept of the symmetry constellations [7]–[9] has been introduced and further the two major types of symmetry have been discussed in the previous chapters. A theory of symmetrical features extraction is relatively new in the area of computer vision so it is encouraged to improve known implementations and to develop some original ones as well. A very high number of potential applications [14]–[15] can yields from the symmetry detection methods.

Only small portion of attention was also paid for robustness and speed aspects of the symmetry features detection at the very end of the paper. Besides theoretical basics, also several ideas about robustness and speed of algorithms for symmetry point detection have been shortly discussed. The important role of the paper was to schematically illustrate significant differences between two basic types of input images for the Hough transforms. An approach of symmetry maps as an input of the Hough transform was definitely selected as much more convenient than the standard edge-based approach, which have more general purpose.

References

[1] Hjelmas, E., Low, B.K. Face Detection: A Survey, Computer Vision and Image Understanding, Vol. 83, Issue 3, pp. 234-274, 2001, Elsevier, ISSN 1077-3142.

[2] Lienhart, R., Maydt, J. An Extended Set of Haar-like Features for Rapid Object Detection, Proceedings of the International Conference on Image Processing, Vol. 1, pp. 900-903, 2002, Rochester. ISBN 0-7803-7622-6.

[3] Wilson, P.I., Fernandez, J. Facial Feature Detection Using Haar Classifiers, Journal of Computing Sciences in Colleges, Vol. 21, Issue 4, pp. 127-133, 2006, Consortium for Computing Sciences in Colleges, USA. ISSN 1937-4771.

[4] Bai, L., Shen, L., Wang, Y. A Novel Eye Location Algorithm Based on Radial Symmetry Transform, Proceedings of the 18th International Conference on Pattern Recognition, Vol. 3, pp. 511-514, 2006, IEEE Computer Society, Washington. ISBN 0-7695-2521-0.

[5] Davis, L. S. Understanding shape, II: Symmetry. SMC, 7:204–212, 1977.

[6] Wang, P., Ji, Q. Multi-view Face and Eye Detection Using Discriminant Features, Computer Vision and Image Understanding, Vol. 105, Issue 2, pp. 99-111, 2006, Elsevier, ISSN 1077-3142.

[7] Loy, G., Eklundh, J. Detecting Symmetry and Symmetric Constellations of Features, Proceedings of the 9th European Conference on Computer Vision, Vol. 2, pp. 508-521, 2006, Springer-Verlag, Berlin. ISBN 3-540-33834-9.

[8] Loy, G., Zelinsky, A. Fast Radial Symmetry for Detecting Points of Interest, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 959-973, 2003, Washington. ISSN 0162-8828.

[9] Masuda, T., Yamamoto, K., Yamada, H. Detection of partial symmetry using correlation with rotated-reflected images. Pattern Recognition, 26(8):1245–1253, August 1993.

[10] Lowe, D.G. Object recognition from local scale-invariant features, Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on , vol.2, no., pp.1150-1157 vol.2, 1999.

[11] Tyler, C.W., L. Hardage, and R. T. Miller. Multiple mechanisms for the detection of mirror symmetry. Spatial Vision, 9(1):79–100, 1995.

[12] Liu, J., Mundy, J., Zisserman. A. Grouping and structure recovery for images of objects with finite rotational symmetry. In ACCV, volume I, pages 379–382, 1995.

[13] Bay, H., Ess, A., Tuytelaars, T., Gool, L. Speeded-Up Robust Features (SURF), Computer Vision and Image Understanding, Volume 110, Issue 3, June 2008, Pages 346-359, ISSN 1077-3142.

[14] Horak, K. Fatigue Features Based on Eye Tracking for Driver Inattention System, Proceedings of the 34th International Conference on Telecommunications and Signal Processing, pp. 593-597, 2011, Department of Telecommunications, Brno. ISBN 978-1-4577-1411-5.

[15] Horak, K. Honzik, P., Kucera, P. On Image Segmentation Techniques for Driver Inattention Systems, Proceedings of the 17th International Conference on Soft Computing, pp. 1-6, 2011, Institute of Automation and Computer Science, Brno. ISBN 978-80-214-4120-0.