Efficient and robust shape retrieval from deformable templates

Nezhinsky, A.E.; Verbeek, F.J.; Margaria, T.; Steffen, B.

Efficient and Robust Shape Retrieval from Deformable Templates

Alexander E. Nezhinsky and Fons J. Verbeek

Section Imaging and Bioinformatics,
Leiden Institute of Advanced Computer Science, Leiden University,
Niels Bohrweg 1, 2333CA, Leiden, The Netherlands
{anezhins,fverbeek}@liacs.nl

Abstract. Images with known shapes can be analyzed through template matching and segmentation; in this approach the question is how to represent a known shape. The digital representation to which the shape is sampled, the image, may be subject to noise. If we compare a known and idealized shape to the real-life occurrences, a considerable variation is observed. With respect to the shape, this variation can have affine characteristics as well as non-linear deformations. We propose a method based on a deformable template starting from a low-level vision and proceeding to high-level vision. The latter part is typically application dependent, here the shapes are annotated according to an ideal template and are normalized by a straightening process. The underlying algorithm can deal with a range of deformations and does not restrict to a single instance of a shape in the image. Experimental results from an application of the algorithm illustrate low error rate and robustness of the method. The life sciences are a challenging area in terms of applications in which a considerable variation of the shape of object instances is observed. Successful application of this method would be typically suitable for automated procedures such as those required for biomedical high-throughput screening. As a case study, we, therefore, illustrate our method in this context, i.e. retrieving instances of shapes obtained from a screening experiment.

Keywords: Content-based Indexing, Search, and Retrieval, Object detection and Localization, Object Recognition.

1 Introduction

In this paper we focus on the problem of object detection and localization in digitized images for structures that are deformed instances of an archetypal shape (Verbeek, 1995); as an extension to the case of single instance we have investigated the effect of the presence of more than one instance, intersecting or otherwise obscured. In all cases instances need be properly separated from the other content in the digital image.

In order to accomplish the detection in a robust and reproducible manner, we present a framework consisting of two steps for the recognition and annotation of the deformed instances of a predefined shape. Annotation is required so as to be able to compare different instances of objects in a comparable (reproducible) manner.

T. Margaria and B. Steffen (Eds.): ISoLA 2012, Part II, LNCS 7610, pp. 42–55, 2012.

Therefore as part of our solution we also elaborate a straightening normalization of the shape according to a predefined template. Our framework will be applied to a case study in biology, i.e. high throughput screening of zebrafish larvae.

A large number of context based image retrieval systems have been described (Zhong *et al.*, 2000). These can be divided in the *free-form* and the *parametric* approaches. A popular approach for shape retrieval is the Active Contour, a.k.a. the active snake, (Kass *et al.*, 1987) which is a typical *free-form* model (Jain *et al.*, 1996). Free form models require a correct global initialization in the image and optimize the local shape. Free-form class models do not have global shape limitation, but focus mainly on attraction towards certain image features (Jain *et al.*, 1996). In our approach we focus on the cases in which a global shape is known and therefore we do not consider active snake as a possibility.

In research dealing with recognition of known shapes (Garrido *et al.*, 2000; Ng *et al.*, 2006; Felzenswalb, 2003(1,2)) the use deformable templates is emphasized. Deformable templates typically, are *parametric class* models; they start from a set of predefined parameters. The representation of the parameters might differ, but often a template is used, consisting of a set of contour points to which the basic shape outlineis approximated. So, if the basic shape that is looked for is known, it still needs be localized in the images. Therefore the prior knowledge can be exploited by choosing the parametric deformable shape template matching method as a basic approach (Bronkorsta *et al.*, 2000, Jain *et al.*, 1996, Zhong *et al.*, 2000, Felzenswalb, 2003(1)).

Such approach is used in the segmentation of cells in a microscope images (Garrido *et al*, 2000). In this application the Hough transform approach is reformulated to be used as a deformable template. However, if the shapes are more complex than a circular shape like object, it is difficult to adapt to this approach.

An example of a more complex shape is the segmentation of the masseter on the basis of a predefined template (Ng *et al.*, 2006); locally deformed instances of the template can be successfully extracted from input images.

On the basis of this approach we propose a further generalization with which it is also possible to deal with multiple objects in one image as well as with global deformations; e.g. bending of the entire object.

Based on silhouettes or boundary representations of prototype templates a considerable amount of research has been completed (Felzenswalb, 2003). Usually the silhouettes are defined by contour points and make up the template. These templates can then be deformed by a set of parametric transformations, including both local and global transformations (Zhong *et al,* 2000). We have taken this traditional representation as a starting point; however, we have replaced the contour points in the silhouette by a contour area (cf. Figure 2). The reason for using this representation is that we would like to allow multiple overlapping instances of the object in one input image and therefore we have to accommodate for missing contour points.

In addition to the template matching, we also address the problem of shape normalization; in particular for applications of biological objects. The combination of shape localization and normalization has been successfully applied for the round worm, i.e. *C.elegans* (Peng *et al.*, 2007). It is known as the BDB+ method. On the basis of the object boundary a straightening is applied so as to ease the further analysis of the objects. In our application, however, this method cannot be used; it starts from a predefined shape and then straightens the shape assuming the boundary has already been extracted. We want to investigate the recognition and straightening of more complex elongated shapes and, as indicated, account for the presence of multiple instances in one image.

The framework that we have elaborated consists of two steps. First, a preprocessing step including a segmentation of an input image in order to separate the object(s) of interest from the background is applied. Segmentation alone, however, does not give satisfactory results, as we are not only interested in separating background from foreground, but we also want to recognize position and best possible representation of the object. This is realized in the second step consisting of a matching of a deformable template to the segmented image. This step is the main focus of this paper. Finally, a post-processing step includes shape normalization through straightening of the extracted shape. Such is possible from contextual information about the object in the image that we have gained. Deformations are known and therefore deformations can be normalized according to the template. The framework was implemented in C++ using the OpenCVgraphics library (http://opencv.willowgarage.com).

2 Method

The starting point of our algorithm is variation; i.e. a shape has variation, it can be inflicted with noise and it can be deformed or partially occluded. Our framework detects deformed instances of a predefined structure by means of Deformable Template Matching and these are subsequently extracted from an input image. In Figure1, an overview of the process is presented.

The method consists of two steps: the preprocessing step and the template matching step. First, during the preprocessing step, the input image is converted to a strict binary representation. In the main process the deformable template matching is applied in the binary image obtained from the preprocessing. This entails looking for the best match of a prototype template in the image. If a match is found the result is annotated according to the prototype template and henceforth, straightened.

2.1 Pre-processing

The first step in the analysis is retrieving foreground and background: i.e. an operation that converts an input image to a binary representation by marking the pixels which belong to foreground objects 1 and the background pixels 0. Different binarization methods are described in the literature; i.e. based on the usage of global or adaptive threshold methods, color or edge based segmentation. The choice of the method depends on the input image at hand, its properties and quality (Gonzales *et al.*, 2001). In the cases where prior spatial information is known this given can be exploited and the threshold value set can be based on this given.

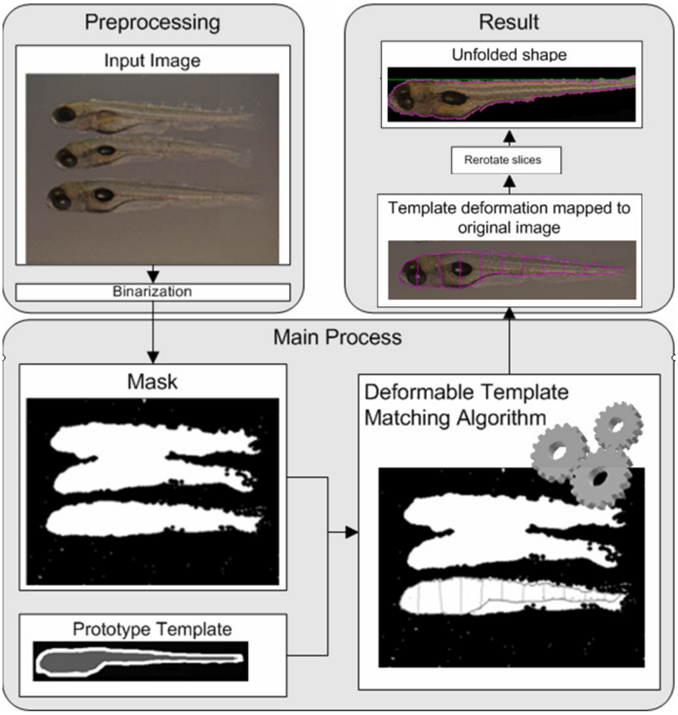

Fig. 1. Proposed framework for automatic shape retrieval and straightening

2.2 Deformable Template Matching

After separation of foreground and background in the image, the contextual information still needs to be retrieved in order to recognize the objects of interest.

Prototype Template. The initial contour sketch of the object of interest is defined by the prototype template T_0. The construction of T_0 is based on prior knowledge and is an approximate representation of how a typical object contour should look like and represents a contour location area. In Figure 2 a few examples are shown.

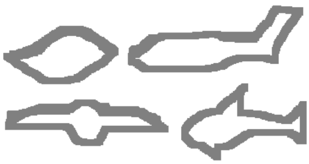

Fig. 2. Some examples of prototype templates of different objects. Gray area represents the space where connected component boundaries might be located.

A contour location area represents the region of interest within which the local template contour can exist and may change. In doing so, local deformations become limited by evaluating only at the pixels that are located within the contour location area (depicted in grey in Figure 2). Introducing a limitation to the template boundary location of the deformed template is necessary to predict image boundary location in cases where it is missing, incomplete or overlapping. As a result of this representation only global deformations are remaining, which will be described in the next subsection.

Parametric Transformation. Biological shape instances are often bended and rotated. In order to cope with these global deformations T_0 is distributed in n smaller sub-templates; these are hereafter referred to as slices t (Nezhinsky and Verbeek, 2010):

$$T_0 = t_0, t_1, .., t_n \tag{1}$$

A single slice can be seen as a rectangular matrix $t(i, j)$, consisting of binary values. This is shown in Figure 3.

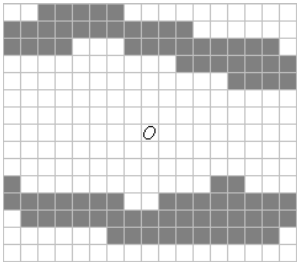

Fig. 3. An example of matrix $t(i, j)$ representing a template slice. Fields with value 1 are marked with grey color. Other fields have the value 0.

The horizontal medial axis of the slice $O_{horizontal}$, is defined at [i/2,*], and the slice origin as O at [i/2, j/2]. Within the whole template the slices can rotate around their origin O to allow matching against a rotated shape. The origins are linked together as a chain (Figure 4); at any deformation of the total shape the distance between sequential slice origins remains the same.

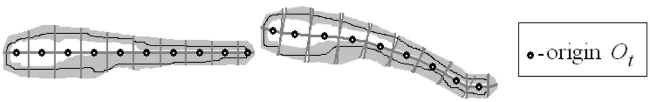

Fig. 4. A prototype template as a chain of slices and a deformed instance. All shape boundaries that fit in the grey area fit the template. Black lines represent two example shapes that fit these templates.

A deformed template T' is derived from T_0 and is represented as:

$$T'(T_0, \underline{\xi}) \tag{2}$$

A deformation $\underline{\xi}$ of the slice chain is encoded by the following state-sequence:

$$\underline{\xi} = (x, y, \alpha_0, \alpha_1, .., \alpha_n) \tag{3}$$

x is the shift in the *X*-axis, *y* the shift in the *Y*-axis direction of the first slice t_0 and α_i the angle of rotation of each slice t_i. Due to the proposed slice based representation our deformable template approach is very suitable for use with elongated shapes.

Objective Function. The fitness of a template matching of an input image is measured by an objective function (Jain *et al.*, 1996). In the *C.elegans* application (Peng *et al.*, 2008) a parametric representation is used in which the algorithm marches along the backbone of a representation to calculate the objective function.

A similar approach is applied, by comparing simultaneously the matrix $S(i, j)$ of binary slices to a selected image region of the same size, i.e. the binary matrix $R(i, j)$ (Figure 5a,b). First, the matrix is considered in which both the template and the image region have overlapping foreground pixels, which are the result of $R(i, j) \cdot S(i, j)$ (Figure 5c).

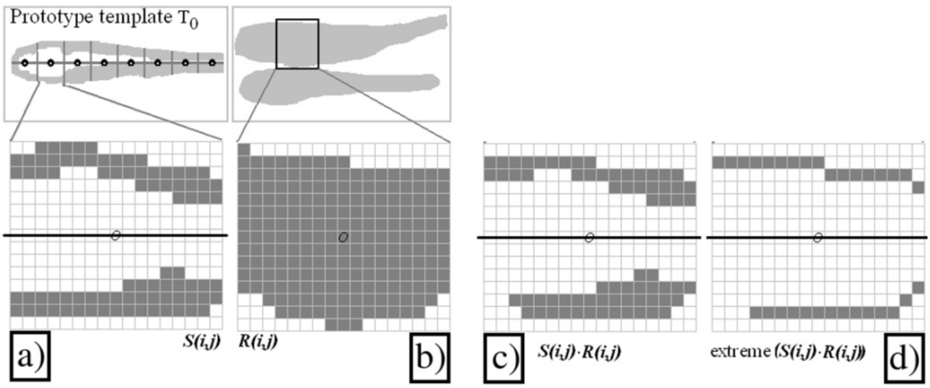

Fig. 5. An example of template matching of a template slice ***S(i,j)*** and a region ***R(i,j)***

During this step the shape is a filled binary object, therefore matches that are farthest from the slice center but still in the silhouette contour area, are assumed to define the object. In order to get the actual border, the algorithm then marches along the horizontal medial axis of $R(i, j) \cdot S(i, j)$ iterating over *0* till *j*. Each orthogonal image plane pixel columns p_i (iterating from *0* to *i*) is compared to the template. It is assumed that the silhouette has only one silhouette pixel in the top and the bottom of each column. Therefore, per column, the two extreme points that are of value 1 (as measured from the horizontal medial axis) remain 1, all other values are set to 0 (Figure 5d). The result thus obtained is considered an intermediate result. The quality, i.e. the objective function, of this result is then measured by the length of the retrieved border. Objective function is 1 if all pixels of the silhouette have been retrieved. To that end the objective function for a slice $S(i, j)$ is defined as:

$$\phi(S(i,j)) = \frac{0.5}{j}\left(\sum (S(i,j)\cdot R(i,j))\right) \tag{4}$$

A template consists of n slices of the same size, and therefore $F(T, \underline{\xi})$ depends on the fitness function of each slice:

$$F(T, \underline{\xi}) = \frac{1}{n}\sum_{k=0}^{n} \phi_k(S(i,j)) \tag{5}$$

Matching the Template to the Input Image. In order to check all possible occurrences, T_0 must be transformed, rotated and deformed by all possible parameters. To find the best solution there is a need to retrieve the global maximum of the fitness function (cf. Eq. 5) for the input image. That is, to compare each $S(i, j)$ to all possible regions $R(i, j)$ within the image.

A global search is computationally complex (Kim *et al.*, 2007), especially when the search space image is large. To that end Genetic algorithms (Tagare *et al.*, 1997) and dynamic programming approaches (Liu *et al.*, 2000), have been used for optimization.

2.3 Post-processing: Straightening the Template

After a sequence of slices is found, the shape can be normalized through straightening by back-rotation of slice found. Since each deformed template T has a deformation defined by $\underline{\xi} = (x, y, \alpha_0, \alpha_1, .., \alpha_n)$ each of its slices t_i is rotated back by the angle slice $-\alpha_i$. In this manner the global deformation of the deformed template can be reverted to the prototype template T_0.

3 Case Study

The framework was developed in the context of high throughput imaging applications. Therefore, as a case study we will elaborate on shape analysis and retrieval of the zebrafish larvae (Stoop *et al.*, 2011). Typically, zebrafish are employed in high throughput studies to investigate new factors for mycobacterial infection. Such approach requires a screening of thousands of larvae.

The shapes of zebrafish larvae are similar, yet each individual (instance) is little different. Moreover, shapes are often slightly bent and rotated. Without proper localization and annotation of the regions in the shapes, the measurement of features within each instance is severely hampered.

The framework that we present fulfills the need for solutions in high-throughput applications in which shapes can be recognized in images and subsequently annotated in such way that these can be compared to other retrieved shapes.

The images for this case study were acquired using Leica MZ16FA light microscope as 24-bit color images with a size of 2592×1944 pixels. On average, each image contains up to 3 larvae. The orientation of the instances in the images is random; however they are not touching the image border. Given the experimental set-up we can assume that the images only contain zebrafish larvae and some incidental noise/debris.

3.1 Pre-processing

Because of uneven illumination, global threshold methods applied on gray-scale converted images of the zebrafish larvae will not produce satisfactory results. We, therefore, employ an edge map based method to the input image. Edges define the boundaries between objects and background without strong dependency to flaws in the illumination. There exist several algorithms to create an edge map. After creation of the edge map a threshold is applied to select for the strong edges.

The determination of the threshold value of an edge map can be a cumbersome task. To set the threshold automatically without prior spatial relationship knowledge of the image, the Otsu segmentation (Otsu, 1979) might be applied.

However, as we have prior spatial relationship knowledge of our particular dataset, we can exploit this for our *border based* method; i.e. the objects in the image are always located at some minimal distance d from image border. This characteristic is utilized. We can assume that a sheet of thickness d on the outside of an image contains only background pixels and some incidental noise. Of this sheet we retrieve a number of local maximal pixel values. Of all the collected values we take the median value to determine the threshold value for the edge map.

In order to select the best preprocessing approach we compare the performance of different simple edge detectors, i.e. Sobel gradient, Roberts gradient (Gonzales & Woods, 2001) in combination with Otsu segmentation (Otsu, 1979) and our *border based* method on 233 images. We count the number of objects in the segmented image. If the number of objects equals the number of objects in the original image we mark the prediction as correct. Both gradient methods in combination with our border method outperformed thresholding based on Otsu segmentation. Out of the images incorrectly segmented in 17 cases the zebrafish shapes were touching each other and thus connected. In all of these cases both edge detectors predicted 1 or 2 fish instead of 3 due to this connection.

After basic segmentation is completed, mathematical morphology operations are applied to get rid of noise and close up unwanted gaps. The closing operator is applied to connect small regions and close holes, then connected components labeling (with filling up holes) is applied to obtain the closed shapes. In order to eliminate remaining noise we use the fact that we know the minimal area covered by a zebrafish. This area size can be automatically retrieved from the template size. Thus we remove all objects smaller then this minimal area. We do not remove objects that are larger than the maximal area, since larger object might be intersecting zebrafish shapes.

3.2 Main Process: Deformable Template Matching

Prototype Template. Our initial zebrafish template (cf. Figure 6) is created from averaging a test set of training shapes (Cootes *et al.,*1994); here the template is created by averaging a set of 20 zebrafish larvae shapes.

Zebrafish larvae tend differ in length. Therefore, the length of the template is not fixed, but can vary between some minimal (*min*) and maximal (*max*) number of slices $t_{\min}$ and $t_{\max}$. If the number of found slices is smaller than *min* or larger then *max* we assume the shape is not found. All the slices t_x with $min < x < max$ are thus optional slices. In Figure 6 this is depicted. The *max* and *min* are set based on the length of the longest and shortest encountered zebrafish.

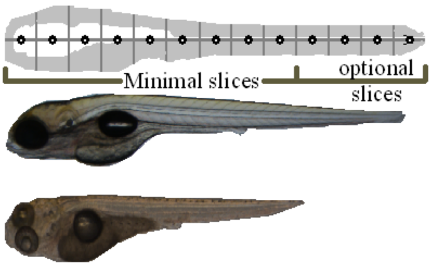

Fig. 6. Minimal slices and optional slices in a prototype template. The template is compared to an example of a short and a long larvae shape.

Objective Function for the First Slice. The described objective function is applicable for slices in which the important information is located above and below the slice center. While most of the zebrafish larvae template applies to this condition the very first slice, in which the head is located, does not. This is due to the fact that its shape is close to half circular. This is depicted in Figure 7. To cope with this case, instead of retrieving extreme values above and below the median axis extreme values are retrieved in a circular way as shown in Figure 7.

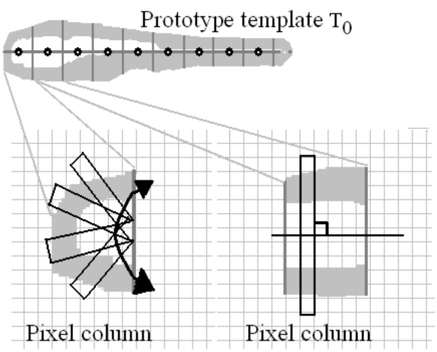

Fig. 7. Marching direction for the first slice depicted in the image pixel matrix

Matching the Template to the Shape. In order to obtain a global optimum a top-down dynamic programming approach is applied with a hash table saving for intermediate result. In our case study the larvae shapes are located in approximately vertical positions in the input image. This fact is used to reduce the search space by assuming that each slice can rotate between -45 to 45 degrees as measured from the image horizontal axis. Additionally, a discrete set of deformation angles for each slice is used.

To further reduce the search space a *Multi resolution* algorithm (Leroy, 1996) is used. First the solution is located on a low resolution template and a low resolution input image. Then, the solution is used for initialization in a higher resolution.

4 Experiments and Results

An evaluation of our algorithm is performed on a dataset consisting of 233 images which were obtained from a running experiment. Out of the images 177(76%) contained 3 larvae, 33(14%) contained 2 larvae and 23(10%) 1 larva.

A first basic test is to check for how many of the tested images the number of larvae is predicted correctly: for all images (100%) the number of the larvae shapes (1,2 or 3 larvae, even if overlapping) with in the image was correctly retrieved.

The correct prediction of the number of shapes in an image is promising, however only retrieving the number of shapes is not sufficient for a proper analysis. To that end we also tested the accuracy of the algorithm, that is, how precise the shapes were retrieved. In Figure 8 representative results of retrieved shapes are depicted. We show shapes that are deformed in different way as well as shapes touching each other. The template in Figure 6 was used for the creation of all these images.

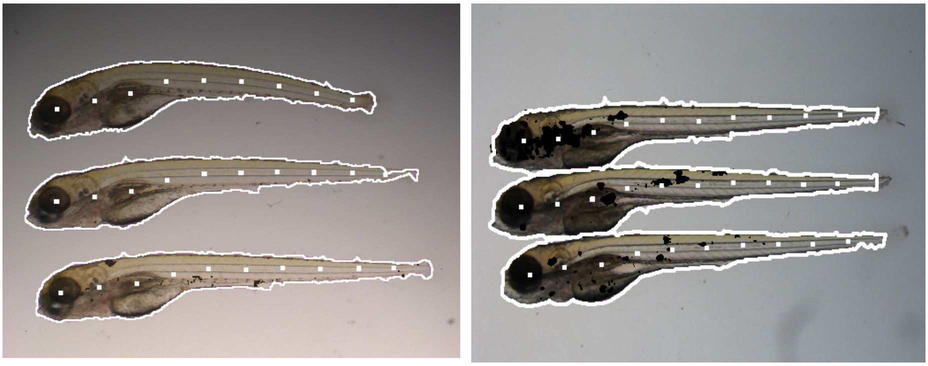

Fig. 8. Automatic localization of the zebrafish shapes using deformable template matching. The white line defines the shape as found by the algorithm. The white dots represent the slice centers found. In the right image the shapes are slightly touching each other, which complicates the recognition. These result were created using the same template as the left image; the black regions in each larva depicts a bacterial infection.

Methods that can be used for automated retrieval predefined shapes from images without an initialization have not been described, and therefore we have compared the resulting shapes of each retrieved zebrafish larva shape against ground truth images of the same shapes as annotated by experts. A comparison of human to automatic retrieval is regularly used for validation procedures (Peng *et al.*, 2008).

In order to have manageable proportions in the evaluation, we reduced our test set to a total of 104 zebrafish shapes (distributed over in 35 images, containing up to 3 shapes per image). Four experts (test-persons T1, T2, T3, T4) were asked to delineate the outline of the zebrafish larva. Drawing the shapes was realized with an LCD-tablet (Wacom) using the TDR software (Verbeek *et al.*, 2002).

Next, the precision of our method is compared by applying it to the same input data (algorithm output A). The accuracy of our shape retrieval algorithm is measured by the equation proposed in (Ng *et al.*, 2006):

$$accuracy = 2 \times \left(\frac{N(M_{area} \cap S_{area})}{N(M_{area}) + N(S_{area})} \right) \times 100\% \tag{6}$$

We have compared the accuracy of our algorithm to T1, T2, T3 and T4. The average accuracy was established as 96.71 (σ= 1.27). Note, that this accuracy could not be achieved by the segmentation step alone, as 35 (of the 104) shapes used for this test were touching each other and their boundary could only be derived through the template matching.

Table 1 presents the results of the comparison of the accuracy of our algorithm with the test persons. In addition the inter-observer variation is analyzed.

Table 1. Accuracy comparison of our algorithm T0 and the test subjects T1, T2, T3, T4. The matrix is symmetrical, yet we have shown all the values for viewing convenience

	T0	T1	T2	T3	T4
T0	-	96.85	97.19	96.29	96.47
T1	96.85	-	97.61	97.21	96.81
T2	97.19	97.61	-	97.17	97.46
T3	96.29	97.21	97.17	-	96.68
T4	96.47	96.81	97.46	96.68	-

As can be seen from the table the accuracy between our algorithm and each test person is as close to the accuracy of the test persons to each other. This indicates that the algorithm retrieves shapes as good as or comparable to manual retrieval.

In the last part of the experiment the objects, i.e. zebrafish, are normalized; a straightening operation. This is accomplished using a template with a straight top border in order to align the slices found with their top to a horizontal line. In Figure 9 and Figure 10 the results are shown.

To retrieve and straighten a single zebrafish shape from a 2592×1944 image took our application about 35s CPU time on an Intel Dual Core 2.66 Ghz, 1.00 Gb.

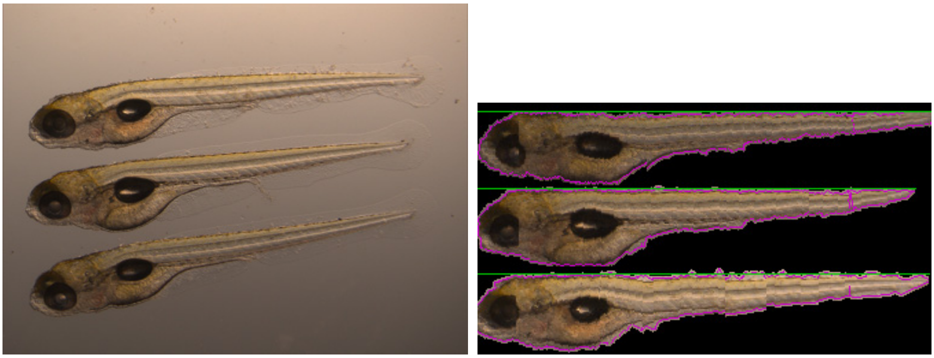

Fig. 9. Results showing the automated straightening of zebrafish larvae. Image (left) is the input image. Image (right) is the normalization result.

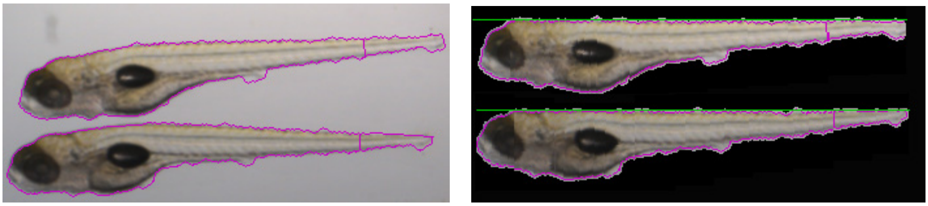

Fig. 10. Results of the automated straightening zebrafish larvae. Image (a) shows the retrieved shapes projected on the input image. Image (b) shows the automated normalization result.

5 Conclusions

In this paper we have described a framework for automated detection of archetypal object shapes in an image. Once detected a post-processing by straightening of each object on the basis of a predefined template is applied.

In our framework, the prototype template is represented as a bitmap and can easily be adapted to the needs of the application while the same algorithm is used.

The algorithm we propose does not rely on initial localization of the shape and therefore does not require any manual intervention or analysis.

The framework was applied in an experimental set-up for high throughput screening with a read-out in images. In the application to zebrafish screening average accuracy of about 96 percent has been achieved.

The framework can be easily adapted to work with other shapes, be in the life sciences or in other fields that require accurate and robust shape retrieval.

Further analysis of the validation and the precision in object straightening is part of the future work.

Acknowledgements. This work is partially supported through the Smartmix program.

References

1. Brunelli, R.: Template Matching Techniques in Computer Vision: Theory and Practice. John Wiley & Sons, Ltd. (2009)
2. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models: their training and application. Computer Vision and Image Understanding 61(1), 38–59 (1995)
3. Cootes, T.F., Taylor, C.J., Lanitis, A.: Active Shape Models: Evaluation of a Multiresolution Method for Improving Image Searches. In: Proceedings of the British Machine Vision Conference, vol. 1, pp. 327–336 (1994)
4. Dormand, J.R., Prince, P.J.: A family of embedded Runge–Kutta formulae. J. Comp. Appl. Math. 6(6), 19–26 (1980)
5. Felzenszwalb, P.: Representation and Detection of Shapes in Images. Ph.D. dissertation, Massachusetts Institute of Technology (2003)
6. Felzenszwalb, P.: Representation and Detection of Deformable Shapes. In: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), vol. 1, p. 102 (2003)
7. Garrido, A., Perez de la Blanca, N.: Applying deformable templates for cell segmentation. Pattern Recognition 33 (2000)
8. Gonzales, R., Woods, R.: Digital Image Processing, 2nd edn. Addison-Wesley, London (2001)
9. Jain, A.K., Zhong, Y., Lakshmanan, S.: Object matching using deformable templates. IEEE Tran. on Pattern Analysis and Machine Intell. 18(3) (1996)
10. Jain, A.K., Zhong, Y., Dubuisson-Jolly, M.: Deformable Template Models: a Review. In: Signal Processing - Special Issue on Deformable Models and Techniques for Image and Signal. Elsevier (1998)
11. Zhong, Y., Jain, A.K.: Object localization using color, texture and shape. Pattern Recognition 33 (2000)
12. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Int. Journal of Comput. Vision 1(4) (1987)
13. Kim, H.Y., de Araújo, S.A.: Grayscale Template-Matching Invariant to Rotation, Scale, Translation, Brightness and Contrast. In: Mery, D., Rueda, L. (eds.) PSIVT 2007. LNCS, vol. 4872, pp. 100–113. Springer, Heidelberg (2007)
14. Leroy, B., Herlin, I., Cohen, L.D.: Multi-resolution algorithms for active contour models. In: Proceedings of the 12th International Conference on Analysis and Optimization of Systems Images, Wavelets and PDE'S, Rocquencourt (1996)
15. Liu, Z., Wang, Y.: Face detection and tracking in video using dynamic programming. In: Proceedings of International Conference on Image Processing (2000)
16. Nezhinsky, A.E., Verbeek, F.J.: Pattern Recognition for High Throughput Zebrafish Imaging Using Genetic Algorithm Optimization. In: Dijkstra, T.M.H., Tsivtsivadze, E., Marchiori, E., Heskes, T. (eds.) PRIB 2010. LNCS (LNBI), vol. 6282, pp. 301–312. Springer, Heidelberg (2010)
17. Ng, H.P., Ong, S.H., Goh, P.S., Foong, K.W.C., Nowinski, W.L.: Template-based Automatic Segmentation of Masseter Using Prior Knowledge. In: Proceeding SSIAI 2006 Proceedings of the 2006 IEEE Southwest Symposium on Image Analysis and Interpretation (2006)
18. Peng, H., et al.: Straightening Caenorhabditis elegans images. Bioinformatics 24, 234–242 (2008)
19. Ren, M., Yang, J., Sun, H.: Tracing boundary contours in a binary image. In: Image and Vision Computing (2002)

20. Ridler, T.W., Calvard, S.: Picture thresholding using an iterative selection method. IEEE Trans. System, Man and Cybernetics SMC-8, 630–632 (1978)
21. Stoop, E.J.M., Schipper, T., Rosendahl Huber, S.K., Nezhinsky, A.E., Verbeek, F.J., Gurcha, S.S., Besra, G.S., Vandenbroucke-Grauls, C.M.J.E., Bitter, W., van der Sar, A.M.: Zebrafish embryo screen for mycobacterial genes involved in the initiation of granuloma formation reveals a newly identified ESX-1 component. Dis. Model. Mech. 4(4), 526–536 (2011)
22. Tagare, H.D.: Deformable 2-D template matching using orthogonal curves. IEEE Transactions on Medical Imaging 16(1), 108–117 (1997), http://www.ncbi.nlm.nih.gov/pubmed/9050413
23. Otsu, N.: A threshold selection method from gray-level histogram. IEEE Transactions on Systems Man Cybernet (1978)
24. Verbeek, F.J.: Three-dimensional reconstruction from serial sections including deformation correction Delft University of Technology, Delft (1995)
25. Verbeek, F.J., Boon, P.J.: High-resolution 3D reconstruction from serial sections: microscope instrumentation, software design, and its implementations. In: Three-Dimensional and Multidimensional Microscopy: Image Acquisition and Processing IX (2002)