Proceedings of the International Multiconference on Computer Science and Information Technology pp. 97–104
ISBN 978-83-60810-27-9
ISSN 1896-7094

Automatic Visual Object Formation using Image Fragment Matching

Mariusz Paradowski
Wroclaw University of Technology
Institute of Informatics, Poland
Nanyang Technological University
School of Computer Engineering, Singapore

Andrzej Śluzek
Nanyang Technological University
School of Computer Engineering, Singapore
Nicolaus Copernicus University
Faculty of Physics, Astronomy and Informatics, Poland

***Abstract*—Low–level vision approaches, such as local image features, are an important component of bottom–up machine vision solutions. They are able to effectively identify local visual similarities between fragments of underlying physical objects. Such vision approaches are used to build a learning system capable to form meaningful visual objects out of unlabelled collections of images. By capturing similar fragments of images, the underlying physical objects are extracted and their visual appearances are generalized. This leads to formation of visual objects, which (typically) represent specific underlying physical objects in a form of automatically extracted multiple template images.**

I. Introduction

Image understanding, although generally considered the highest level of machine vision applications, provides useful information for low–level image processing tasks. For example, noise removal, image segmentation, etc. can be performed more effectively if the presence of certain contents is known or assumed in the processed images. On the other hand, the results of low–level operations are usually indispensable to detect such contents in unknown images. This contradiction is apparently one of the most challenging issues in advanced applications of machine vision.

A similar, and equally challenging, problem exists in *content-based visual information retrieval* (CBVIR). Two operations can be relatively easily done for a given image, namely (1) image annotation (manually performed by a human) and (2) low–level feature extraction (performed automatically by dedicated algorithms). However, a *semantic gap* [1] exists between these two operations, i.e. it is usually very difficult to relate image features to semantically meaningful image tags.

The paper proposes a technique that handles the above contradictions from the low–level perspective. In general, we attempt to automatically build image semantics using purely visual characteristics of the images. The content of images is assumed unknown and random (although they may be collected from a certain "world"). The first step of the proposed approach, i.e. the automatic formation of *visual objects* [2] (which typically correspond to physical objects depicted in the images) is discussed in the paper in detail. Such *visual objects* are built as groups (clusters) of *prototypes* [2] containing multiple similar fragments identified in the available collection (database) of images. If the database is a representative description of the "world", we identify typical components and, thus, provide a certain level of understanding of that "world". The presented results are development of preliminary ideas highlighted in [3].

The similarity between image fragments is determined using sets of locally computed feature vectors, i.e. the proposed method falls into the *local* category. Each feature vector describes a small fragment of the image (usually an elliptical or circular *keypoint*). To calculate the keypoints, we use popular, widely discussed approaches such as *Harris-Affine* [4] and *SIFT* [5], [6].

Detection and clusterization of similar fragments in the image database is executed in four steps: (1) image pre-retrieval, (2) image fragment matching, (3) formation of *prototypes* from similar fragments, (4) formation of *visual objects* from prototypes. The first two steps are responsible for localization of similar image fragments within the database. Image pre–retrieval is introduced solely for a higher efficiency of the method, while image fragment matching is the key operation in finding similar image fragments.

In the remaining two steps, relations are established between similar image fragments identified in the first two steps. *Prototypes* are formed based on intersections of image fragments located within the same image. These multiple intersecting fragments come from matching of the same image with other images. The last step merges prototypes found in different images into *visual objects*.

The general idea of the proposed method is illustrated in Fig. 1 and detailed explanations of the underlying algorithms are provided in Section II.

Altogether, the automatic visual object formation is a grouping algorithm. Its input is a collection (database) of unlabelled images. The output is a set of groups (visual objects), where each group consists of image fragments that have been found mutually similar. The number of output visual objects is determined fully automatically based on the visual properties of database images.

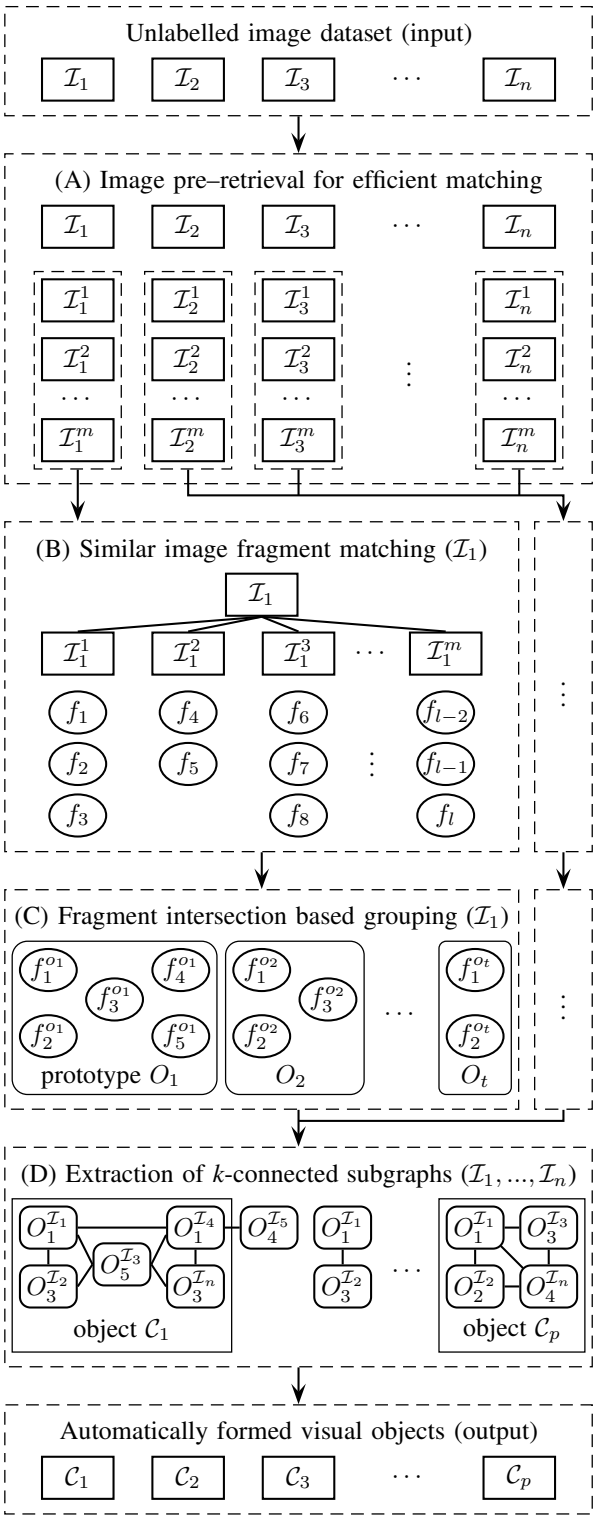


Fig. 1. Steps of the proposed automatic visual object formation method; (A) image pre–retrieval to limit number of matches, (B) image fragment matching to extract similar fragments in different images, (C) formation of prototypes out of intersecting fragments in the same image, (D) formation of objects out of prototypes from different images.

II. Visual object formation

A. *Efficient image pre–retrieval (step A)*

Given a collection $\{\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_n\}$ of n unlabelled images, our objective is to identify the presence of similar fragments within these images. Therefore, for a collection of n images we have to match up to n^2 image pairs. This can be a time–consuming operation even for a small database. Given a pair of images $\mathcal{I}$ and $\mathcal{J}$, similar fragments are identified using an *image fragment matching* method (described in Section II-B). The results of such a matching are accurate, but the operation is computationally intensive. Therefore, we have introduced an efficient image pre–retrieval scheme so that the system can be applied to image collections of relatively large sizes. The proposed pre–retrieval mechanism is one of the novelties presented in this paper.

For a query image $\mathcal{I}_x : x \in \{1, ..., n\}$, we attempt to identify the most relevant (candidate) images $\{\mathcal{I}_x^1, \mathcal{I}_x^2, ..., \mathcal{I}_x^m\} : m \ll n$ from the whole database $\{\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_n\}$. Detection of fragments similar to unspecified fragments of $\mathcal{I}_x$ query is attempted within those candidate images only.

The candidate images are identified using a specialized *similarity function* $s(\mathcal{I}, \mathcal{J})$ defined for a pair of images $\mathcal{I}$ and $\mathcal{J}$. The similarity function is called *image topology similarity*. The function can be converted (if necessary) into an image–distance function using a variety of approaches. Unlike many classic image similarity measures, the proposed similarity function is able to identify images that are very different but contain similar fragments. The idea of the proposed function is conceptually similar to the previously proposed topological image fragment matching algorithm [7], i.e. it is based on pairs of matched keypoints. The main difference is that in here we use *weaker topological constrains*.

The proposed similarity function is defined as follows: First, we obtain matched pairs of keypoints $P(\mathcal{I}, \mathcal{J})$ between the input images (details to be discussed later). Having a non-empty set of such pairs P, we check the topological constrains for each keypoint pair $(p_I, p_J) \in P$. The topological constrains are verified within spatial neighbourhoods $\mathcal{N}(p_I)$ and $\mathcal{N}(p_J)$ of both keypoints from the pair. The spatial neighbourhood is defined as a set of r keypoints being the nearest neighbours in terms of image coordinates (Euclidean distance d) of a given keypoint. The neighbourhoods are found off–line and the nearest neighbours can be cached for each keypoint (and quickly retrieved on demand). Formally, the spatial neighbourhood $\mathcal{N}(p_X)$ for a keypoint p_X from an image (set of keypoints) $\mathcal{X}$ is equal to:

$$\mathcal{N}(p_X) = \arg\min_{N \in \mathcal{X}} \sum_{n \in N} d(n, p_X), \quad |N| = r. \tag{1}$$

Given $\mathcal{N}(p_I)$ and $\mathcal{N}(p_J)$ neighbourhoods of p_I and p_J keypoints, we check *how many* pairs of matched keypoints $P(\mathcal{I}, \mathcal{J})$ can be found within these neighbourhoods. The larger number of found keypoint pairs, the more credible (topologically) is the selected keypoint pair $(p_I, p_J) \in P$. We can now define a *topological verification function* $t(\mathcal{I}, \mathcal{J}, p_I, p_J)$ for

a pair of keypoints $(p_I, p_J) \in P$ as the normalized number (r is the neighbourhood size) of matched pairs found in the neighbourhoods ($\times$ stands for Cartesian product):

$$t(\mathcal{I}, \mathcal{J}, p_I, p_J) = \frac{1}{r}\Big|\big[\mathcal{N}(p_I) \times \mathcal{N}(p_J)\big] \cap P(\mathcal{I}, \mathcal{J})\Big|. \quad (2)$$

An illustrative example is shown in Fig. 2.

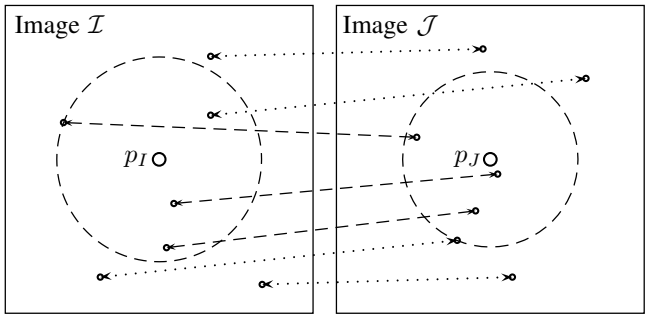


Fig. 2. Illustration of the topological constrains. Only three (dashed lines) out of $r = 4$ neighbours are matched keypoint pairs, $t(\mathcal{I}, \mathcal{J}, p_I, p_J) = \frac{3}{4}$.

The *similarity function* $s(\mathcal{I}, \mathcal{J})$ between two images $\mathcal{I}$ and $\mathcal{J}$ is defined using the *topological verification function* $t(\mathcal{I}, \mathcal{J}, p_I, p_J)$ for all matched keypoint pairs $P(\mathcal{I}, \mathcal{J})$ of both images (Eq. 3). The normalization factor $P^{max}(\mathcal{I}, \mathcal{J})$ is the maximum possible number of matched keypoint pairs generated by the matching routine, given images $\mathcal{I}$ and $\mathcal{J}$.

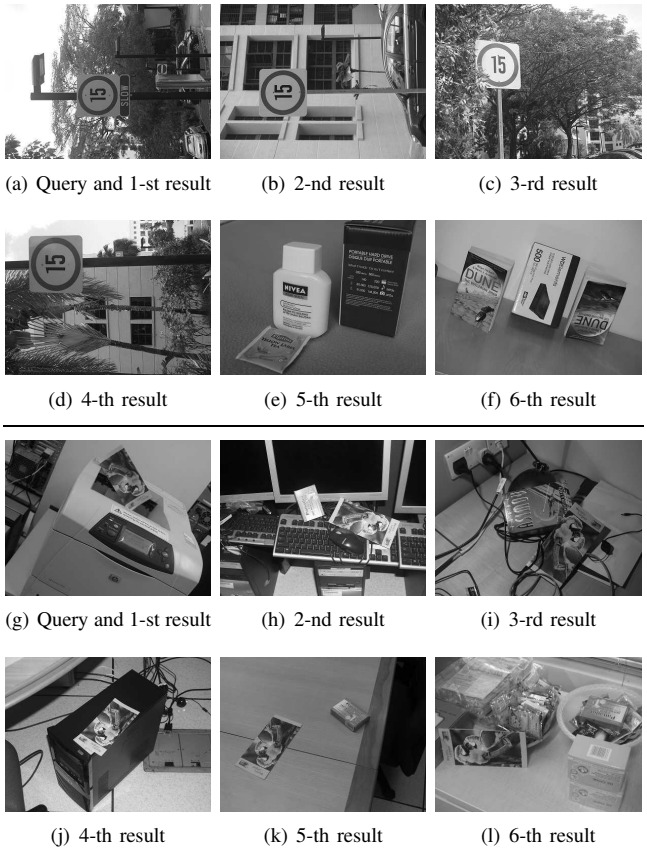
$$s(\mathcal{I}, \mathcal{J}) = \sum_{(p_I, p_J) \in P(\mathcal{I}, \mathcal{J})} \frac{t(\mathcal{I}, \mathcal{J}, p_I, p_J)}{P^{max}(\mathcal{I}, \mathcal{J})}. \quad (3)$$

An important property of the proposed *similarity function* is that it can detect the presence of similar fragments (even very small ones) in images with very complex and diversified backgrounds. The function is also computationally effective; it requires only $O(pr)$ operations, where p is the number of keypoint pairs $P(\mathcal{I}, \mathcal{J})$ and r is the size of the neighbourhood (the costs of keypoint matching are not included into the complexity of the function). Additionally, it is not sensitive to a certain level of inaccuracies in keypoint matching.

Efficiency of keypoint matching is, obviously, another important aspect of the algorithm. The classic approach (e.g. *coherent pairs* method, i.e. *one-to-one* matching) is extremely slow and takes as much as $O(fp^2)$, where f is the length of keypoint descriptor vectors. This makes the classic (exact) approach not applicable to the pre–retrieval step, and we need to search for an approximate keypoint matching approach. There is a variety of approximate nearest neighbour algorithms e.g. [8], [9]. We have implemented a method which has been experimentally found time–efficient (although we did not compare its performances against the available alternatives). Exemplary pre–retrieval results are shown in Fig. 3.

B. Image fragment matching (step B)

The key factor in automatic visual object formation is a reliable detection of multiple similar fragments in a collection images (without any prior knowledge about the image contents). The assumption regarding the complete lack of prior knowledge is very important, because the system has to explore and learn unknown environments.



(a) Query and 1-st result (b) 2-nd result (c) 3-rd result

(d) 4-th result (e) 5-th result (f) 6-th result

(g) Query and 1-st result (h) 2-nd result (i) 3-rd result

(j) 4-th result (k) 5-th result (l) 6-th result

Fig. 3. Examples of pre–retrieval based on *image topology similarity* function $s(\mathcal{I}, \mathcal{J})$. The proposed similarity function is able to detect a presence of small similar fragments in images with very complex and diversified backgrounds.

Let us now formalize the similar fragments detection routine. Given a set of images $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_n\}$ we would like to detect all existing similarities within the set. To make the process more efficient, for each image $\mathcal{I}_x : x \in \{1, ..., n\}$, we filter out only a subset of most similar images $\{\mathcal{I}_x^1, \mathcal{I}_x^2, ..., \mathcal{I}_x^m\} : m \ll n$ (see Section II-A). Thus, there are nm possible image pairs to be checked. For a single pair of images, a reliable detection of similar fragments can be solved by a *image fragment matching* method. The fragment matching method generates a set of image fragment pairs, first element of each pair represents a fragment on the first image, second element of the pair represents the similar fragment on the second image. When all similar images in the database are matched, the resulting set (union of sets from all pairs) contains all similarities found within the database.

Image fragment matching the most important and the most time consuming operation of the whole approach. We use two such methods, namely: *geometric* and *topological* keypoint matching. We will shortly present both of them, now.

1) The geometrical method with triangles: The objective of geometric image fragment matching is to reconstruct a set of *affine* transformations relating similar planar surfaces present on both input images. The transformations are reconstructed from triangle pairs built over both images using pairs of matched keypoints. Affine transformations are decomposed into elementary transformations (rotations, translations and scales). A six–dimensional histogram (affine transformations have six degrees of freedom) of all transformations is built for a pair of images. A 2D subspace (two elementary rotations) of an exemplary histogram for a selected pair of images is visualized in Fig. 4. A non–parametric approach is used to find peaks of the histogram, which represent dominant affine transformations relating both images (i.e. relating similar fragments in the images). Sets of triangle pairs which contribute to these peaks form the outlines (convex hulls) of similar fragments shared by both images. Further details can be found in [10], [11].

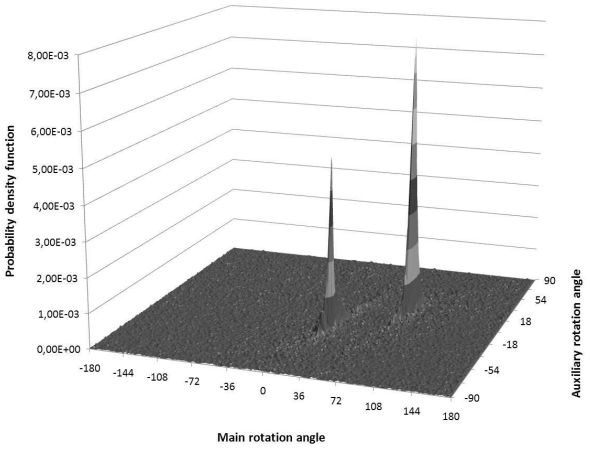


Fig. 4. Histogram of two rotations extracted from affine transformations [11]. Two peaks are visible, they represent two similar fragments.

2) The topological method: An alternative approach is the topological image fragment matching. Instead of recreating exact geometrical transformations between fragments of two images, this method focuses only on image topology. A *topological constrain* is introduced which, in general, reliably represents shape distortions of physical objects. We assume that neighbouring keypoints have to obey this constrain to be considered a similar fragment. Locally similar fragments are, therefore, extracted according to the results of topological constrain verification. The proposed topological constrain is the *matching order of vector orientations connecting keypoints from a selected pair to the neighbouring keypoints*. The constrain for a given keypoint pair is illustrated in Fig. 5. The topological method is more flexible than the geometric one; it is able to detect non-planar and deformed fragments of similar objects. However, the detection is less accurate in terms of generated fragment outlines (also represented as convex hulls). Further details on the topological method can be found in [7].

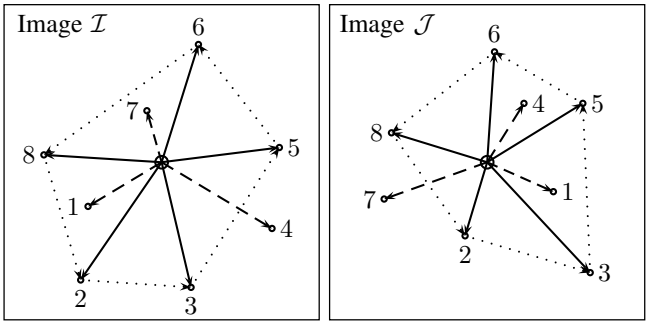


Fig. 5. Topological image matching concept. For each pair of matched keypoints, the largest subset of orientation-ordered neighbors is found. An exemplary subset of size 5 is shown.

3) Performances of image fragment matching: Reliable visual object formation is possible only if image fragment matching is performed with a high quality. In fact, any *false positive* matching error is very problematic for the future processing based on graph analysis. Such errors result in false connections within the graph and may result in incorrect contents of visual objects. The proposed routines are somewhat resistant to such errors, but this resistance is rather weak. On the other hand, *false negative* fragment matching errors are much less problematic. In case of missing graph connections, some visual objects may not be formed correctly. To solve this problem, it is usually enough to pre–retrieve more images or to deliver more images which would contain the corresponding physical objects. However, both of these solutions increase the computational costs of the method.

TABLE I
AVERAGE RECALL AND PRECISION FOR THE TEST DATASET.

Detector / *Descriptor*	HarAff SIFT	HarAff GLOH	HarAff Mom.	SURF SURF	MSER SIFT
Method	Geometrical method				
Prec. (area)	0.96	0.96	0.97	0.90	0.95
Recall (area)	0.64	0.50	0.47	0.49	0.53
F-m. (area)	0.77	0.66	0.64	0.63	0.68
Prec. (obj.)	0.97	0.97	0.97	0.98	0.94
Recall (obj.)	0.81	0.71	0.70	0.61	0.68
F-m. (obj.)	0.88	0.82	0.81	0.75	0.79
Method	Topological method				
Prec. (area)	0.64	0.62	0.78	0.50	0.71
Recall (area)	0.79	0.74	0.64	0.70	0.63
F-m. (area)	0.71	0.67	0.70	0.59	0.67
Prec. (obj.)	0.98	0.97	0.99	0.97	0.98
Recall (obj.)	0.92	0.88	0.86	0.79	0.78
F-m. (obj.)	0.95	0.92	0.92	0.87	0.87

The achieved matching results (the geometric and topological approaches) on the processed database are shown in Table I. Two measurement modes are used: in the first one (object-wise) we check if the similar fragments are matched, in the second one (area-wise) we measure how accurately the shapes of matched fragments are outlined. The geometric method is more precise in terms of area measurement, due to very strict mathematical foundations. The topological method is less precise, but it is able to find more matching fragments.

In Fig. 6 we show two exemplary cases of similar fragment matching using the geometric approach.

Fig. 6. Examples of image fragment matching using geometrical method

C. Automatic formation of prototypes (step C)

Image fragment matching provides us with a set of image fragment pairs. While image fragments within a pair are related (similar) there is not direct visual relation between fragments belonging to different pairs (even if detected in the same image). We need, nevertheless, to establish such relations because some of those fragments may represent the same physical object; the first step is to build relations between fragments extracted within the same image.

As shown in Fig. 1 a single image $\mathcal{I}_x : x \in \{1, ..., n\}$ is matched (i.e. it shares similar fragments) with a subset of images from the database. These fragments depicts physical objects present in the corresponding pairs of images. If we consider physical objects located in a single image $\mathcal{I}_x$, there might be several image fragments depicting each of these objects (they come from different matching processes against the same fragments of image $\mathcal{I}_x$). Such fragments should be very similar in shape, size and location (they are on the same image and approximate the same underlying physical object). Therefore, we assume that similar fragments represent the same physical object of image $\mathcal{I}_x$ and, thus, such groups of similar fragments are called *prototypes*; this is an important concept in the proposed approach. In fact, *prototypes* are intermediate structures required for form **visual objects**.

The process of prototypes construction is a grouping problem. To extract prototypes in a single image $\mathcal{I}_x$ we analyse intersections of image fragments. The larger is the relative size of the intersection, the higher chance that both fragments represent the same physical object. Two fragments are merged (to form a prototype) if: (1) both have similar sizes (areas), (2) the intersection of fragments is relatively large, i.e.

$$\min\left(\frac{c_1}{c_2}, \frac{c_2}{c_1}\right) > t_R, \quad \frac{c_I}{\min(c_1, c_2)} > t_I, \tag{4}$$

Fig. 7. Different image fragments (convex hulls) belonging to the same *prototype*. They come from matching of a single image with other images from the collection.

where: c_1 – area of the first fragment, c_2 – area of the second fragment, c_I – area of fragment intersection, t_R – area ratio threshold, t_T – intersection area ratio threshold.

Prototypes are formed from multiple fragments using a graph analysis technique. Each fragment is represented by a single graph node. Two graph nodes are connected by an edge if the underlying fragments satisfy the above merging criterion (Eq. 4). At least two fragments are necessary to form a prototype. Given the graph representation, prototype construction can be simply solved using graph connected component search. Various fragments of the same prototype formed in an exemplary image are given in Fig. 7.

D. Automatic formation of visual objects (step D)

Each prototype depicts (usually) a physical object located within a single image. Our ultimate objective is, however, to establish relations between prototypes form all database images, i.e. to form groups of prototypes that are referred to as *visual objects*.

Fortunately, the connections between images are already established in a form of similar fragment pairs (see Section II-B and Fig. 1). Since similar fragments are matched using a high–precision matching process (see Table I) the generated inter–image connections are mostly correct (this is a fundamental requirement for the visual object formation).

Formally, the formation of visual object is another graph–based grouping problem. We simply build a graph representing the above–mentioned connections between images. Each prototype $O_y \in y = \{1, ..., q\}$ is represented by a single graph node (note that prototypes are groups of image fragments from the same image). Each image fragment is matched with a similar fragment in another database image. Therefore, graph edges are created between nodes (prototypes) according to the matches between image fragments. Each prototype has a number multiple outgoing edges, equal to the number of similar fragments within this prototype. However, because the precision of similar fragment matching is still below 100%, a verification mechanism has to be put in place. The proposed verification mechanism is based on the analysis of graph *k-edge-connectivity*. A graph is *k-edge-connected* if it remains connected when less than k edges are deleted from the graph. In other words, a new prototype can be added to an existing visual object *if and only if* it is connected with at least k prototypes from the visual object. Such an approach is effective in eliminating random matching errors, because they usually form only a single connection to another

Fig. 8. Visual objects are formed out of many prototypes present on different images.

Fig. 9. Exemplary image fragments (instances of prototypes) representing visually formed objects.

prototypes. The resulting grouping algorithm is, therefore, a *k-edge-connected* subgraph search routine.

Exemplary image fragments belonging to different prototypes (but within the same visual object), found in various images are presented in Fig. 8.

In Fig. 9 we show a subset of visual objects automatically formed in the analysed collection of images (due to size limitations, each object is represented just by a single image fragment from one of the prototypes belonging to the object). Although no semantics is used during the object formation process, one my find that the created visual objects represent the actual physical objects appearing in multiple images within the database.

III. Discussion

Before presenting the experimental results we would like to discuss the applicability of the method and note its limitations. As stated above, the key requirement for successful object formation is very high precision of matching. Nowadays, precision near 100% may only be reached for image (fragment) matching problem, i.e. localization of image fragments containing identical objects. State-of-the-art approaches applicable for similarity based retrieval and grouping could not be used for the stated object formation problem, because their precision is still too low. Thus, the proposed method applies only for image collections depicting the same objects. Popular image databases such as *Caltech 101/256* may not be processed by the method, because they mostly contain similar (but *not identical*) underlying physical objects. For such databases *nothing would be found*.

Due to the mentioned, specific requirements, we have tested the method on a dataset containing images depicting a set of physically identical objects. For the reference and test purposes, the dataset may be downloaded from a web site[1]. First, we will discuss this dataset, later on we will show how to set up method's parameters. In the last part of this section we will summarize the achieved results.

A. The dataset and the objective of experiments

The dataset consists of 100 diversified images, captured both indoors and outdoors, containing a variety of objects. Most images in the database contains more than one object of interest, appearing in different configurations with diversified backgrounds. Camera settings and lighting conditions also differ between images.

Some of the physical objects repeat in several database images and thus, they are the candidates for visual objects. We have identified a set of physical objects present on at least three different images. These objects are: four different books, a notepad, a leaflet, two different medicine boxes, tissue pack, three different bottles, tea bag, two road signs, an exit sign and a street advertising poster. There are also other physical objects repeating only twice in the database (we consider them irrelevant because the assumed minimum number of prototypes in a visual object is $k = 3$, see Section III-B).

Our objective is to find all those repeating objects in the database and form visual object out of them. We expect that there will be no errors in the created objects, i.e. each visual object may represent *only one* underlying physical object. If a single physical object is represented by more than one visual object, we consider it a problem of lesser grade, because it is easily solvable in the proposed framework (see Section II-B3). Of course, the more correctly formed objects (without errors and object duplications) the better.

B. Method parameters

The proposed method has five parameters, related to (1) image pre–retrieval, (2) prototype formation and (3) visual object formation. All parameters, their short description and suggested values are listed in Table II.

The parameter m defines the size of the subset of most similar pre–retrieved images. The larger value of the parameter, the higher chance to capture important visual connections

[1]Image fragment matching dataset: http://www.ii.pwr.wroc.pl/∼visible

TABLE II
DEFAULT VALUES OF THE METHOD'S MAIN PARAMETERS.

Param.	*Meaning*	*Value*
m	Number of pre–retrieved images (% of n)	0.25n
r	Size of topological neighbourhood	60
t_A	Minimum ratio of image fragments area size	0.75
t_I	Minimum ratio of image fragments intersection size	0.75
k	Visual objects graph edge connectivity	3

between images from the database. However, larger values of the parameter significantly slows down the method. The maximum value of this parameter is n; this represents a disabled pre–retrieval, i.e. all image pairs are matched. The parameter r defines the size of the topological neighbourhood (see Eq. 2). To choose the proper value of the parameter, we need to consider two issues. The topological neighbourhood has to be large enough to be informative and it has to be small enough due to memory requirements (to speed up the method all neighbourhoods are pre–computed off–line).

Another two parameters are related to prototypes. They are used in the image fragment merging routine. As mentioned in Section II-C, merging should happen only if two image fragments (they are on the same image) represent the same physical object. Both parameter values t_A and t_I have been experimentally set to 0.75.

The last parameter is related to the visual object formation. To eliminate potential error matches, we want each prototype to be linked to at least k other prototypes (i.e. each prototype should contain at least k image fragments). In the presented approach, we have assumed $k = 3$, which is the minimum possible value. Due to high precision of image fragment matching, it is fully sufficient. However, if matching errors are more frequent (lower precision of image fragment matching) especially on larger databases, one should consider a larger value of the parameter. As a result, it would be more difficult to form visual objects but they would be more credible.

C. Discussion on created visual objects

The minimum requirement to form a visual object is having a physical object *correctly matched* on at least three different images ($k = 3$, see Section III-B). The quality of matching (i.e. precision of matched area) is in fact the most important element of successful object formation. To measure how well the visual objects are formed, we use the ground truth information as discussed in Section III-A. In fact, we want to capture as much of underlying physical objects as possible.

Given the processed database and the proposed parameter set up, there are *no errors* in the generated visual objects, i.e. each formed visual object represents only one physical object. In some cases, a physical object is represented by more than one visual object, but such objects are merged together when more data is provided (the pre-retrieval size is increased). Also, larger amount of matching data leads to larger number of formed objects. Given the pre–retrieval mechanism, we can easily set up how many images will be given to image fragment matching and subsequently how many similar fragments may be found. Comparison of results for different pre–retrieval scenarios are given in Tab. III. We note that not all objects have been found (see Section III-A), non–planar ones were the most problematic. Low number of pre–retrieved images causes only a few objects to be created. The more pre–retrieved images, the more formed objects and the quality increases, but the computational cost also increases.

TABLE III
AUTOMATICALLY FORMED OBJECTS VERSUS GROUND TRUTH OBJECTS. "+" REPRESENTS A CORRECTLY FORMED OBJECT, "–" REPRESENTS A MISSING OBJECT, "±" MEANS THAT MORE THAN ONE VISUAL OBJECT HAVE BEEN FORMED FOR THE UNDERLYING GROUND TRUTH OBJECT.

Ground truth defined object	Pre–retrieval size (m) [% of n]							
	3	4	5	10	15	20	50	100
	Geometrical fragment matching							
Book 1	±	±	+	+	+	+	+	+
Book 2	–	+	+	+	+	+	+	+
Book 3	+	+	+	+	+	+	+	+
Book 4	+	+	+	+	+	+	+	+
Bottle	–	–	–	–	+	+	+	+
Box	–	–	–	±	±	+	+	+
Exit sign	–	+	+	+	+	+	+	+
Leaflet	+	+	+	+	+	+	+	+
Medicine 1	±	+	+	+	+	+	+	+
Medicine 2	+	+	+	+	+	+	+	+
Road poster	+	+	+	+	+	+	+	+
Road sign 1	+	+	+	+	+	+	+	+
Road sign 2	–	–	–	–	+	+	+	+
Tea bag	+	+	+	+	+	+	+	+
Tissues	–	–	–	+	+	+	+	+
	Topological fragment matching							
Book 1	–	±	±	+	+	+	+	+
Book 2	+	+	+	+	+	+	+	+
Book 3	+	+	+	+	+	+	+	+
Book 4	–	+	+	+	+	+	+	+
Bottle	+	+	+	+	+	+	+	+
Box	–	–	–	+	+	+	+	+
Exit sign	–	–	–	–	–	–	–	–
Leaflet	–	+	+	+	+	+	+	+
Medicine 1	–	–	–	+	+	+	+	+
Medicine 2	–	–	–	–	–	–	–	–
Road poster	+	+	+	+	+	+	+	+
Road sign 1	–	–	–	–	–	–	–	–
Road sign 2	–	–	–	–	–	–	–	–
Tea bag	–	–	–	–	–	–	–	–
Tissues	–	–	–	–	–	–	–	–

Having the geometrical method employed for the matching task, we may expect very high area precision (see Tab. I). The number of pixel–level false positive errors in generated image fragments is minimal. But there is also a cost, non–planar objects will be only partially captured and they will most probably not constitute correct prototypes. The topological fragment matching overcomes the problem of non–planar objects, but has much lower precision in terms of matched area. Statistics demonstrating the ability to recreate meaningful objects are shown in Tab. III. Due to high precision the geometrical method is a more suitable candidate for object formation than the topological one. The main problem with the topological approach is much smaller number of prototypes successfully used in object formation. Large differences in image fragments (the main criterion for successful prototype formation) prevent capturing identical parts on the same image. Even though the

number of matched fragments (mostly correct) is higher for the topological method, the overall number of formed objects is lower. Statistics for the numbers of created prototypes and visual objects are given in Tab. IV.

m [%]	Matched fragments	Formed prototypes	Prototypes in objects	Formed objects	True objects
Geometrical fragment matching					
3	203	106	39	11	9
4	291	131	64	12	11
5	357	138	72	11	11
10	589	162	95	14	13
15	**709**	**174**	**105**	**16**	**15**
20	**778**	**176**	**102**	**15**	**15**
50	951	188	108	15	15
100	1063	196	108	15	15
Topological fragment matching					
3	218	114	13	4	4
4	307	155	33	8	7
5	378	186	45	8	7
10	617	244	63	9	9
15	752	290	67	9	9
20	840	307	72	9	9
50	1059	348	80	9	9
100	1211	382	80	9	9

TABLE IV
CREATED PROTOTYPES AND VISUAL OBJECTS FOR VARIOUS SETTINGS OF PRE-RETRIEVAL (m).

An interesting case is present for geometrical matching, $m = 15$ and $m = 20$. The number of prototypes successfully used in object formation is in fact *decreasing*. Surprisingly, this is a correct behaviour. Some image fragments, which should create a single prototype, were not merged together for $m = 15$. Instead, multiple different prototypes are created. If the number of pre–retrieved images is increased to $m = 20$, missing links between these prototypes are found. Image fragments are properly merged and a single prototype is formed. Thus, we can also see a decrease (by 1) of the number of visual objects. In fact, similar behaviour happens quite often (see "$\pm$" in Tab. III), but in this case it is well captured.

We may also observe a much lower number of formed visual objects for the topological fragment matching. It is related to the already mentioned problem of lower area precision. Decreasing the values area related merging thresholds (t_A and t_I) results in creating of *false objects* and thus is not an acceptable approach. This confirms one of the initial statements, that successful visual object formation requires image fragment matching working with very high precision.

Apart of the listed features of the proposed method, there are also weak points. In some cases one detected prototype is a part of another prototype. Prototype formation criterion based on fixed merging thresholds (t_A and t_I) can not capture it correctly. Such prototypes (and later on visual objects) will not be joined, even though they represent the same underlying physical object (or a part of it). Therefore, we should consider building a *hierarchy* of prototypes and visual objects, instead of a plain structure.

IV. SUMMARY

A method for automatic formation of visual objects has been presented. The method is able to find meaningful image fragments from an unlabelled set of images. Visual objects are formed out of repeating, similar image fragments within the dataset. The proposed method employs *image fragment matching* techniques to extract such similar fragments of images. Two matching techniques are used, namely: the geometric and the topological ones. Apart from the visual object formation solution, we have also presented a novel image pre–retrieval method that effectively identifies images prospectively containing similar fragments. The method uses a topology–based similarity function. It is an important component of the system, because it significantly shortens the matching process.

The proposed solution creates a set of visual objects, without any kind of structure or hierarchy. This might be its weak point, because some similar fragments may indeed be structured (e.g. one object represents a visual fragment of another object). Our further research will focus on building such a hierarchy of visual objects.

ACKNOWLEDGEMENT

The research presented in this paper is a part of A*STAR Science & Engineering Research Council grant 072 134 0052. The financial support of SERC is gratefully acknowledged.

This work is partially financed from the Ministry of Science and Higher Education Republic of Poland resources in 2008–2010 years as a Poland–Singapore joint research project 65/N-SINGAPORE/2007/0.

REFERENCES

[1] A. W. Smeulders and A. Gupta, "Content-based image retrieval at the end of the early years," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.

[2] S. Dickinson, A. Leonardis, B. Schiele, M. Tarr, S. Edelman, and P. Valery, *Object Categorization: Computer and Human Vision Perspectives*, 2009.

[3] A. Śluzek and M. Paradowski, "A vision-based technique for assisting visually impaired people and autonomous agents," in *Proc. 3th Int. Conf. on Human System Inteaction HSI2010*, 2010, pp. 653–660.

[4] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, pp. 63–86, 2004.

[5] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Computer Vision*, vol. 2, 1999, pp. 1150–1157.

[6] ——, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[7] M. Paradowski and A. Śluzek, "Keypoint-based detection of near-duplicate image fragments using image geometry and topology," in *Proc. International Conference on Computer Vision and Graphics, LNCS*, 2010, in press.

[8] A. Andoni, M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni, "Locality-sensitive hashing using stable distributions," *Nearest Neighbor Methods in Learning and Vision: Theory and Practice*, 2006.

[9] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *International Conference on Computer Vision Theory and Applications*, 2009.

[10] M. Paradowski and A. Śluzek, "Detection of image fragments related by affine transforms: Matching triangles and ellipses," in *Proc. International Conference on Information Science and Applications*, vol. 1, 2010, pp. 189–196.

[11] ——, "Local keypoints and global affine geometry: triangles and ellipses for image fragment matching," *Innovations in Intelligent Image Analysis*, 2010, in press.