Информационные процессы, Том 10, № 3, стр. 197-211

From Pattern Recognition to Image Understanding (Language, Segmentation, Concepts)[1]

S.A. Guberman*, **V.V. Maximov****, **A.V. Pashintsev*****

**PO Box 2411, Cupertino, CA 95915, USA*
e-mail: guboil@hotmail.com
***Institute for Information Transmission Problems, Russian Academy of Sciences,*
19 Bolshoi Karetny line, 127994 Moscow, Russia
****Evernote Corp, 333 W. Evelyn ave, Mountain View, CA 94041, USA*

Received August, 31, 2010

Abstract – To resolve the image search problem, an algorithm has to understand images, i.e., be able to describe their content using an adequate language. With that approach the comparison of images becomes comparison of descriptions. To accomplish the above goal the software was developed,, which at the first step segmented the image. The segmentation is based on Dual Clustering procedure, which generates a limited number of segmentations and chooses the best according criteria. At the second step segments are recognized as notions "sky", "vegetation", "water", "ground", "mountains", "buildings" and some more.

"I stand at the window. Theoretically I might see there were 327 brightnesses and nuances of color. Do I have "327"? No. I have sky, house, and trees."

M. Wertheimer (1923)

INTRODUCTION

Phrase "pattern recognition" originating in the field of AI was, at first, considered only in a visual context, as the recognition of visual patterns [1]. Soon that method was generalized to abstract patterns represented by a set of numbers [2, 3]. Over the time as pattern recognition was successfully applied to abstract patterns in geology, geophysics, medicine, sociology and other fields it has created an assumption that the increasing stream of visual tasks should also be approached with pattern recognition methods. That meant that the approach was "learning through examples". Nowadays, most programs that can distinguish human faces on photographs use this approach. It requires a couple of hundred examples of full-face photographs to be used for learning to produce a decision rule. The decision rule is applied to every rectangle of a given size on the image. The search and the learning process is repeated for other sets of human-face images where each set has the face in a different position, i.e. face turned at an angle, head tilted, size varied, etc.

At the same time another idea appeared and began to develop – the idea of an adequate language [2, 4]. It started as just an idea but very soon it started to find practical support through a whole series of applications. The first successes in implementing this idea were in medical diagnosis, earthquake prediction and oil exploration. The reason for breakthroughs in each of these cases was the use of a new language adequate for describing the phenomena being studied. But the algorithmic basis of all these solutions was pattern recognition based on learning through examples.

[1] This paper is an extended version of the paper submitted to the Fifth National Conference on Systems Science, Fermo, Italy, 14-16 October 2010.

However, when considering visual objects the choice of an adequate language looks different. In technical applications (as in geology, seismology, and medical diagnosis) the number of possible descriptions can be very large but always finite. In the case of visual images the set of possible descriptions is practically infinite. Thus the task of finding in image processing an adequate language becomes the key problem.

In late 60's M. Bongard proposed a very general principle of constructing an adequate language – the imitation principle: the right way of describing things is to describe how they were created [2]. This principle has been successfully implemented in the learning algorithm for classification of black-and-white "geometrical" drawings made by humans – so called "Bongard-problems" [5]. However, for more complicated images – photos of the real world – it is difficult to apply the imitation principle to find an adequate language of description. The imitation principle can be applied to man-made objects, but how the sky was made?

LANGUAGE

Image search engine: Melchuk's approach

Let us consider the problem of an image search – a method of filtering out a subset of images within a given database that is the most similar to a user specified image. Most popular image search technology today is the Google Images web service. The following is how the scientists at Google Inc. have described the state of the image recognition technology in 2008:

"Although image search has become a popular feature in many search engines, including Yahoo, MSN, Google, etc., the majority of image searches use little, if any, image information to rank the images. Instead, commonly only the text on the pages in which the image is embedded (text in the body of the page, anchor-text, image name, etc.) is used. A fundamental task of image analysis is yet largely an unsolved problem: human recognizable objects are usually not automatically detectable in images. Although certain tasks, such as finding faces and highly textured objects like CD covers, have been successfully addressed, the problem of general object detection and recognition remains open" [6].

To come up with a successful image search algorithm one must understand and simulate a process that a human goes through to resolve this problem. It is fairly clear what that process is like for a person who is given a picture and is then asked to find similar pictures in an archive of never previously seen images. Most images that a person rejects are filtered out without a need to ever look at the original image again. Only occasionally a person must look at the original picture more carefully to make a decision. Such behavior can be explained by suggesting that humans do not compare each image to the original side by side. The filtering in one's mind seems to be performed by comparing descriptions of the images rather than comparing the actual images.

A similar problem in linguistics is the problem of translation from one language to another. An original approach to this problem was offered by a prominent linguist I. Melchuk [7]. The main idea of the approach was to translate a given phrase to a language of meanings before it is translated into the second language. The phrase in language of meanings is the representation of the meaning of the initial phrase. The main challenge of this approach, of course, was to define what exactly the language of meanings was. Melchuk has recognized that there was no problem in translating nouns from one language to another but there was a great deal of difficulty with other word classes such as verbs, adverbs, adjectives, etc. The breakthrough observation by Melchuk was that all of the word classes were organized in clusters called lexical functions. Each cluster is a meaning. For example, the cluster "to create" contains the following verbs with the same essential meaning as applied to different subjects: build, construct, assemble, produce, grow, raise, cultivate, give birth, originate, set up, establish, and many others.

Let's take a look at a simple example of translating two short phrases with the same verb used in both. The phrases, as pronounced in Russian, are "postroit' dom" and "postroit' grafik". Translating the nouns is easy. The noun that follows the verb in the first phrase, "dom", means a "house". The noun that follows the verb in the second phrase, "grafik", means a "graph". Translating verbs is more complicated. The Russian verb "postroit" is used in both phrases and if translated on its own (without any context) means "to build". In the language of meanings it falls under the cluster, or lexical

function, “to create”. Now we are ready to translate this two phrases from the language of meanings into English. Lexical function “to create” applied to the noun “house” gives in English the word “to build”. Lexical function “to create” applied to the noun “graph” gives the word “to draw”. So, our translation would be “to build a house” and “to draw a graph”.

The most surprising discovery in the Melchuk’s theory was the fact that in such languages as English, French and Russian the number of clusters in the language of meanings was less than 100. This finding by Melchuk has had a crucial impact on shaping up our method for understanding images. The general concept that was fundamental to our approach was Melchuk’s discovery that beneath the surface of any natural system which looked complex there had to be a simpler description if expressed in an adequate language. This principle has guided to a number of relatively simple solutions to very complex problems such as handwriting recognition, earthquake forecasting, oil exploration and a few others [8].

Another related problem that Melchuk has prosed was to generate all possible phrases that had the same meaning as a given phrase. To achieve that goal, Melchuk has proposed to translate the original phrase into the language of meanings first, much like you would with the above translation method. However, in this case, the phrase in the language of meanings would then be translated back into the same language. Since each meaning, or “lexical function”, can usually be expressed with different words, the backward translation would yield a number of different phrases with about the same meaning as the original. For example, if the original phrase was, once again, “to build a house” then the similar phrases as translated back from the language of meanings would be “to construct a house”, “to build a home”, “to raise a building”, “to erect a skyscraper”, “to assemble a cottage”, and many others.

It is clear that some of the newly produced phrases are closer to the original phrase than others or, in other words, are at a different “distance” from the original meaning. Such distance can be measured using the syntactic structure of each phrase based on each word’s position in the structural hierarchy of the phrase.

To apply the above method of translation to our original problem of image search we now have to translate our original image into the language of meanings. It is that meaning that we must compare to meanings of other images in our database. The search result images will be different from the original one but will have the same meaning. Lastly, the search result images are arranged according to their proximity to the given image. The proximity of each image in the search results to the original image would be calculated similarly to how the “distance” between meanings would be calculated in our previous example.

Adequate language

One of the conclusions produced by the above analysis is that the algorithm for understanding images must be able to generate descriptions of those images. This approach poses two problems. The first problem is to create an adequate language to describe images. The second problem is to select a procedure, or a metric, which to employ when comparing images between each other.

Since our method relies on simulating the way a human approaches this problem, let’s get back to our example of how a human would go about finding pictures similar to one given to him/her by looking through an album of never before seen pictures. If you would ask that person why he/she has rejected a picture then the answer would most likely be something like this: “I’ve rejected this picture because in the original picture I saw a person in a park and in the picture I’ve rejected I saw a car on a street which is very different from one another”. As a matter of fact, picking out objects from a picture in order to describe that picture is an essential skill that children are taught in schools from an early age. So, it seems that the first step in accomplishing a successful image search is to program the computer to recognize objects like “person”, “sky”, “river”, “room”, table”, etc.

At this point our approach must be taken a step further and we must recognize that we are not really looking for objects per say but we are more interested in so called notions. For example, we are not as interested in specifically and unmistakably outlining an exact part of the picture where we find the sky. It is more important for us to simply determine that the picture does indeed contain the sky. Other notions that we are interested in are not objects at all. For example, we are interested in

determining whether or not an image was taken indoors or outdoors. Was it sunny or cloudy when a picture was taken?

Of course, it is impossible to create a full list of all notions that a human being possesses to describe a picture. However, some notions are more powerful than others. A powerful notion is the one that is present in roughly half of all pictures in a database. So, when a powerful notion is applied during a search the presence of such notion narrows the search results by roughly half. For example, a notion of an "indoor" (as opposite to the "outdoor") is very powerful. If we have determined that a given image was taken indoors we can now eliminate roughly half of the images in the database that were taken outdoors. In contrast, a notion of an "airplane" is not a very powerful notion. In most searches it is very unlikely that such a specific object as an airplane is present on the original image. Thus, its absence from the picture will not narrow the search results by a significant percentage. Of course, a combination of two or three notions will reduce the number of possible matches even more dramatically (example: "outdoor" + "sunny").

We have determined that a crucial step of performing a successful image search is to recognize notions in a given image. However, at this point, we should mention that there are two other steps that must be accomplished in order to find a solution for our original problem of image understanding. In the second step we must now describe the relationships between the objects in an image. It is not enough to determine that an image contains, for example, a bird and a branch. One must determine a relationship between the two, if that relationship exists. Is a bird flying near the tree? Is a bird sitting on a tree branch? Or is the bird far far away from the tree where there is no immediate relationship between the two?

The third step that must be accomplished is to understand what has happened before, or will happen after, the moment that has been instilled in the picture. For example, on a picture where an old lady is laying on a pavement with her cane to her side it must be determined that the old lady has fallen down in an accident. In another example, if a picture shows a man trapped in a car on a railroad tracks with a train within a small distance from the car then it must be determined that the train will hit the car and the man inside the car will most likely die or will be seriously hurt.

The algorithms used in modern image search technologies do not refer to notions or much less try to understand images. Those algorithms mainly try to compare characteristics of segments within an image based on geometry, coloring, positioning, textures, etc. Such methods can only work for problems that are very limited in their scope. When performing an image search such algorithms can mainly identify images that were taken at the same setting and within the same time of day. Of course, this kind of search results could simply be accomplished by extracting a date stamp of those pictures and, in some cases, a camera's location.

In conclusion, despite the primitive nature and the haziness of some of our reasoning, we strongly believe that the practical applications of our methods would be very valuable in developing successful image search algorithms. Even if our approach is implemented with just one notion, the pool of images in a given database will be significantly reduced.

SEGMENTATION

Objects and borders

The overwhelming majority of images in modern data bases are in color. So, for the recognition of scenes it is natural to use the fact that the sky is blue, clouds are white, vegetation is mainly green, roads are mainly gray, faces are yellow-red, shadows are mainly black, seas and lakes are blue. Consequently, the initial algorithms of image segmentation frequently used the colors of objects. But at the same time, behind all the activities of developing, testing and improving programs for image understanding, lay a simple fact: all these colored objects could be recognized on a black-and-white photograph. This simple fact persistently led us to look for structural, geometrical, and positional features which somehow identify sky, forest, trees, mountains, etc. Moving in that direction presents a question: do we need to know the values of brightness of the gray picture? Is it not enough to have the gradients of brightness only? It seems that in many many cases the answer is "Yes".

So, it seems that we came to the starting point in history of image processing: the gradients of brightness are the basic elements of finding objects in the image. But after our long journey in image processing we interpret the situation different. We are convinced that the initial procedure of image processing is not finding borders of an object, but finding an object and then defining its borders. In other words, we are going not from bottom to top, but from top to bottom. As a matter of fact, we see and recognize many objects despite them being only partially confined by clear visible borders (i.e. with large gradients). That is how we see trees, or clouds on the sky. That is how objects look in X-ray photos. That is how geologists outline tectonic plates – with borders partially defined. That is how a water spot appears on pants – with no borders at all. That is why arts of pen drawing and engraving exist. Therefore the starting point of image processing and understanding has to be finding areas with clear borders not finding points with high gradients, connecting them in lines, and enclosing the lines.

Our current approach looks similar: we generate a small number of hypothetical objects and chose one with borders of best quality. These hypothetical areas are generated using differences in brightness and color, and we need to do it knowing only points of big gradients. In other words, the set of points of big gradients have to serve at the same time as generator of hypothetical areas (future objects), and as a measure of quality of these objects.

All above means that segmentation (in its precise meaning) is not adequate as initial procedure for image processing and understanding, because it defines all borders of prospective objects. It has to be a more fuzzy procedure: define position of objects, show clearly expressed borders, but leave some areas between objects in the fog.

Looking at the image with gradients one can see that points of high gradient form not only lines (potential borders of objects), but some kind of texture as well (consisting on short breaking lines). One can see that such texture helps interpreting the objects: it helps separate trees from the sky, and the sky from the water. It is also obvious that texture can seldom help in defining borders. That is why we used some measure of texture for interpreting the spots, and not for segmentation the image. In the problem of finding objects on image with gradients only, texture can help in initial outlining potential objects.

Color

The use of color in the image segmentation is complicated more by the vector nature of the color space. Colors of individual pixels in digital images are usually specified in a coordinate system RGB, which is device dependent, being used in systems based on electronic displays (TV, video, computers). However, an independent use of the coordinates of this space is unsuited for image processing. The use of notions based on human perception, such as brightness, hue and saturation, instead of the amount of each primary color (red, green, or blue) is more fruitful. In particular, the brightness of the surface depends on the orientation of the surface with respect to the light source. Therefore, in order to locate on the image an area corresponding to the same surface, it is useful to abstract from the brightness. The same goes in case of saturation. In photos of open spaces the saturation usually decreases with distance, and remote trees or mountains look unsaturated.

We have used a coordinate system of brightness (lightness), hue and saturation CIE-Lhs in the color space, developed by the International Commission on Illumination (CIE) specifically for classification of colors according to human visual system [9]. This color space is almost linear with visual perception, and the CIE-Lhs coordinate system is perceptually uniform, its brightness parameter having a good correlation with perceived brightness. A variety of simplified system of color coordinates (HSL, HSV, etc.) developed for computer graphics also describes colors using the same names brightness, hue and saturation. These representations appear to be less useful because they suffer from perceptual nonlinearities and the uneven distribution of their components. Another reason why we have used CIE color space, specifically the coordinate system CIE-L*u*v*, is that it possess a Euclidean metric and the notion of Euclidean distance between colors is determined. The color metric is necessary for calculation of the scatter of color in proximity of a particular point on the image (as a characteristic feature of textured surfaces) and for calculation of value of the gradient of color.

Finding objects

In image processing, a pivotal problem is outlining objects within an image. Usually, a thesis is adopted either explicitly or implicitly that to identify a body in an image means pre-assigning its boundaries and, in this way, the problem of determining the boundaries of an object is substituted for the problem of identifying an object. The boundaries are determined by points where there is a sharp increase in image brightness: gradient of brightness exceeds some threshold. A typical situation occurs when the given value of the threshold is small: boundary points of an object to be identified are contained within this set, but a large number of points which are not boundaries of the object also appear in this set. If the threshold is given a large value, only some, but not all, boundary points will appear in that set. Therefore, in processing complex images one usually has to analyze a set of boundary points obtained to filter out spurious points and include missing ones. This approach is not able to adequately solve problems of object identification in natural images.

It seems reasonable to separate the problems of identifying objects from problems of defining its boundaries, putting the identification of objects first. Once the problem of identifying objects is solved, the problem of defining the boundaries of these objects is radically transformed. Previously the search for boundaries in general (and only then, the elucidation of what exactly these boundaries delimit) came first, but now the problem of defining the boundaries of a certain object is posed.

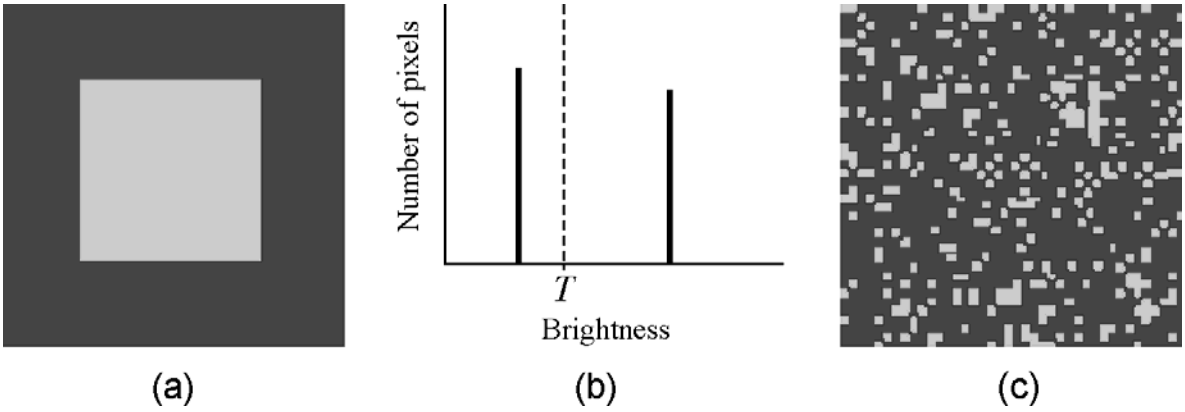

Fig. 1

There is an old technique of finding objects by using histograms of brightness. A simple example is shown in Fig. 1 (a). The histogram of the brightness for that image has two spikes – Fig. 1 (b). A slice of the image at any level of brightness $B = T$ between these two spikes produces a bitmap, which outlines the object. On real photos the brightness of the objects is never a constant. Representation of the object on histogram will not be a sharp spike but a bell-like curve as well as the representation of the background. Still the minimum of the histogram between the two maximums will provide a reasonable threshold T and will outline the object. The goal of this procedure is to single out clusters of brightness. M. Bongard considered this procedure as one of the basic tools of our intelligence [2]. He used the term “heap” for “cluster” and “breaking down into heaps” for “clustering”. We use the following measure of clustering

$$k(T) = \frac{\sqrt{n_1 \cdot D_1} + \sqrt{n_2 \cdot D_2}}{\sqrt{n_0 \cdot D_0}},$$

where D_1 is dispersion of the left part of the histogram ($B < T$), D_2 is dispersion of the right part of histogram ($B \geq T$), n_1 and n_2 are number of pixels in each part of the histogram, and D_0 and n_0 are dispersion and total number of pixels for the entire histogram. When clusters overlap, the best dividing threshold corresponds to the minimum $k_{\min} = \min_T k(T)$.

Majority of real photos are more complex: the object is represented on histogram by more then one maximum, the clusters are asymmetric and overlapping, the difference in brightness of different objects is small etc. All that makes this tool useful in a limited number of cases. Not all these

difficulties originate from the complexity of the reality. There is a shortcoming in the procedure alone: the algorithm does not care about ***position*** of pixels with particular brightness. Histogram for the image in Fig. 1 (c) (known as "salt and-pepper") is identical to the histogram of the image in Fig. 1 (a) although there are no objects in Fig. 1 (c).

To overcome this defect, a pair of spaces has to be introduced: one space is the one-dimensional histogram of brightness $H = H(B)$, the second space – the dual 3-dimensional space of the original image itself $B = B(x, y)$. The first space allows to measure how compact is distributed the **brightness** of the image by calculating minimal clustering k_{min}. Threshold brightness T corresponding to k_{min} defines the binary (black-and-white) image – bitmap $b = \varphi(x, y)$, where $\varphi(x, y) = 0$, if $B(x, y) < T$, and $\varphi(x, y) = 1$, if $B(x, y) \geq T$. The bitmap b is an object in dual space. On that bitmap a measure has to be defined reflecting how compact distributed black (or white) ***pixels*** are. For example, the measure of compactness for the bitmap in Fig. 1 (a) has to be much higher then for the bitmap in Fig. 1 (c).

A number of measures for compactness can be discussed.

1. *Number of spots N on the bitmap*. The less N the higher is the compactness of the bitmap. It works well for "salt-and-pepper" image – Fig. 1 (c).

2. *Length of all borders L on the bitmap for a given threshold T*. That measure separated the Fig. 1 (a) and Fig. 1 (c) as well: the shorter the border the more is the compactness on the bitmap.

3. *Value of the gradients on the object's borders*. The ideal situation is when all objects have big gradients on their borders. But the reality is far from the ideal. That is why the common approach that starts with finding areas of high gradients and then proceeds to find objects has so many difficulties. The Dual Clustering (DC) approach starts with finding objects (spots on bitmap b) and then estimates in dual space the "quality" of their borders. In other words, we are not looking for points of high gradient, but for objects with good borders. By the way, this measure, which is very useful in gray and color images, doesn't work on Fig. 1 (c).

Because each of proposed measures has its own pro and contras we construct a combination M_{DC} that reflects 1) difference in brightness between the object and the background measured by k, 2) length of all borders L reflecting the geometry of the object, and 3) mean gradient on the borders G, which reflects quality of the border:

$$M_{DC} = \frac{G}{k \cdot L}.$$

The bigger is M_{DC} the better is the quality of segmentation

Image segmentation

Principles described above were implemented in a program, which executes the following steps.

1. Input image is split in three channels: Hue, Saturation, and Brightness (Lightness).
2. Gray areas on the image are found (as areas with low saturation). These areas are excluded from the image in the Hue channel.
3. For segmentation the DC procedure is applied to each channel (H, S, L), i.e. for each channel $M_{DC}(T)$ was calculated and the maximum M_{DC} and corresponding threshold were kept ($\{M^B_{DC}\ T^B\}$, $\{M^H_{DC}, T^H\}$, and $\{M^S_{DC}, T^S\}$).
4. The largest of three M_{DC} values was chosen and appropriate T was used to create the bitmap representing a chosen segmentation. That bitmap divided the complete image into two segments: all black pixels and all white pixels. Each segment is then divided in non-overlapping connected sets of pixels – spots.
5. The algorithm continues by applying recursively the Dual Clustering procedure to each spot of the image obtained at the previous step.
6. At each step spots are eliminated if (a) the spot is too small, or (b) the measure of clustering M_{DC} for that spot sunk below some threshold.

Segmentation stops when all spots are eliminated.

The most time consuming part of Dual Clustering is finding maximum calculating M_{DC} for each modality (*L, H*, and *S*). For that purpose 255 black-and-white bitmaps have to be generated (for each of 255 values of given modality). On each map borders of all spots have to be identified. Each pixel of an image has 4 neighbors. The brightness of that pixel and of all its neighbors is known. Having these values one can find at which thresholds T that pixel will be a border pixel. According to the definition, a pixel is a border pixel if at least one of its neighbors belongs to the spot and at least one of its neighbors belong to the background.

Let B_0 be the brightness of the given pixel. Let B_j ($j = 1, 2, 3, 4$) be the brightnesses of its neighbors. Let B_{min} be the minimum of B_j. First, it has to be noted that a pixel with brightness B_0 can be a border point of some spot on bitmap only if the threshold T, that created that bitmap, is less the B_0. Now, if the bitmap was created by the threshold T, which is smaller then B_{min} ($T < B_{min}$), then the central pixel and all neighboring pixels will belong to the spot, and the central pixel is not a border point. In case the threshold is between B_{min} and B_0 the central point will belong to the spot and at least one pixel (with brightness = B_{min}) will belong to the background, i.e. the central pixel will be a border point. That information has to be defined only once for each pixel and then become known on which bitmaps (i.e. for which thresholds) it will be a border point.

CONCEPTS

From objects to notions

As soon as the image is segmented into spots we can work on farther interpretation: to find the notions. As it was mentioned before, the list of notions, which are useful in outdoor scenes without people or animals, is as follows.

1) sky,
2) vegetation (trees, bushes, grass),
3) building,
4) road,
5) car,
6) mountains,
7) water (sea, lake, pool, river).

Above mentioned notions are of a varied nature. Some of them are well-defined objects which could be described by a small number of features. For example, a car has four wheels and a body. Another example is the human face (two eyes, nose, mouth). Difficulties in recognition are caused by the fact that they are 3D objects and appear on scene at different angles and therefore look different. Nevertheless, it is possible to teach the computer to recognize these objects using a limited number of views at different angles and of different size for learning purposes. It is not very sophisticated but it could work.

What about sky, or vegetation, or water (seas, lakes, rivers)? They can not be represented by a limited number of views, as they are not physical objects but concepts. The sky does not exist as a physical object, the sky is a universal background, it has no shape. The concepts of vegetation, buildings, and human bodies have the same problem: too many appearances.

When we confronted the image understanding problem we decided to develop simple and reasonable algorithms to understand the reality of images, with a readiness to change our understanding of visual objects, colors, scenes, and recognition. And unlike our segmentation algorithm, in finding notions we use the simplified HSL color space. We had no difficulties in creating notions induced by color problems. We also modified our segmentation algorithm by replacing CIE representation of color to simplified HSL coordinates. At first glance it caused minor changes, which is crucial, and didn't change the list of found notions. We believe that it happened because we were trying to imitate the human perception at a very low level, and simplicity of tools turned out adequate for the simplicity of the task.

Here are the short descriptions of algorithms for finding notions. Examples are shown in Fig. 5, where descriptions generated by our software are given below each image.

Sky

Every spot found by segmentation could be described by shape of its borders, by color, by brightness, by position relatively to the frame of the image, and by position relative to other spots. Appearance of sky varies dramatically in color and shape. Geometrical characteristics of the borders of sky spot in an image are borders of other objects: buildings, mountains, trees.

In search for the sky the analysis began with spots (result of segmentation) of significant brightness and particular color (in HSL coordinates from H=130 to H=170). Usually it is connected to the upper border of the frame. As a rule, it covers a significant area of the image or touches a significant part of the top border of the image. It is often found at some distance from the bottom border of the image. Sky could be represented by one spot, or by a number of spots. Of course, for each of these “rules” a number of contrary examples exist, but still the rules cover the majority of real outdoor pictures. We would like to reiterate that in the beginning it is preferable to develop simple and reasonable algorithms and clarify the obstacles of real image understanding.

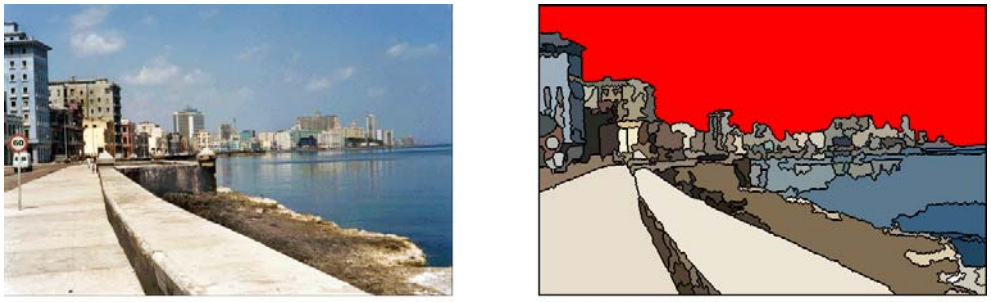

Fig.2

Fig.3

Figs 2 and 3 illustrate the operation of the algorithm. Original color images presented in Figs 2 (a) and 3 (a), were first subjected to segmentation, and each spot was painted in its average color. Then there were found spots corresponding to the sky and painted in red. Segmentation results are shown in Figs 2 (b) and 3 (b), where the sky is marked with red.

Our experience with the program has shown that in most cases of segmentation of outdoor images by this algorithm the first division occurs by hue channel. Typically, it is a division into two segments with warm and cool colors. In this case one or more spots, which form the segment of cold colors, meet the “rules” stated in this section, i.e., they represents the "sky". Thus, with “top-to-bottom” scheme to find the sky it is unnecessary to carry the segmentation procedure to the end. All is revealed on the very first steps.

Clouds

Sometimes there are clouds on the sky – sometimes they are light with fuzzy borders and our segmentation fails to represent them as distinctive objects (Fig. 2). Sometimes clouds are well defined and create objects, which are part of the "sky" – Fig. 3 (b). Commonly they are white or gray, and completely or partly surrounded by sky. There are some other objects that could appear in the sky and be misrecognized by the above mentioned rules as clouds: balloons, airplanes, blimps. The distinctive features of such artificial objects are sharp borders, color, and texture.

Vegetation ("Green")

Green (trees, bushes, grass etc) is a very common part of outdoor non-urban scenes. It is clear that not all trees belong to that notion: the trees without leaves (trees in the winter or after a fire) are excluded, and trees in late fall if covered with red and yellow leaves. Here once more we face the reminder of limitations of our approach to image understanding – dependence on color: the human eye can recognize vegetation on gray image.

There are three main difficulties in identifying the notion of "green":
1) objects belonging to that notion have no definite shape,
2) same object appears quite different on different distances,
30 texture and color saturation are extremely variable.

Naturally, the first obstacle is the existence of artificial objects colored green. It could be overcome by measuring the smoothness of the green surface (in contrast to vegetation, which is characterized by sharp changes in brightness and saturation). That is due to the essential three dimensional structure of vegetation, which exposes to the observer deep dark pockets between the brightly illuminated leaves.

Another fundamental feature of the "green" notion is the size. To be an important part of the scene, the vegetation has to occupy a significant part of the scene. And that is one more option for differentiating "green" from many green artificial objects. To mention ahead, presence of such notions in the scene as sky or lake increases probability that green spot is vegetation. Important features for "green" (like size, homogeneity, and position) can be defined only after the spot itself is defined. For that purpose a "green" channel was established by cutting out from Hue channel the green interval (from H=50 to H=130). It solves the problem in significant number of scenes but in many cases it extracts a mosaic of separated small spots – a tiny part of the vegetation visible on the image. The rest of vegetation is closer to blue or red part of the spectrum not to mention the parts of vegetation in deep shadows, which appear dark gray. If we try to expand the "green" area by expanding the interval extracted from the Hue channel, it picks out a lot of spots, which are not part of vegetation.

We choose the following practical solution:
1) get spots from the "green" channel (from H=50 to H=130),
2) get spots from expanded channel (from H=30 to H=150),
3) add spots from the expanded channel, that have common borders to the "green" spots,
4) add gray spots, which fill holes in spots created in point 3.

Examples are shown in Fig. 3 (b) and Fig. 5.

Trees

Between a broad variety of vegetation trees are most distinctive, particularly the stand alone tree. Tree appears green in the center (around the stem), when covered with leaves, and expose separated branches at the edges. Between the branches on the edge one can see the sky (or other background), which constitute bays in the spot interpreted as sky. Between the sky bays and the green mass in the center of the tree there is a silent not interpreted zone. That zone contains small green spots isolated from the big central green spot, and because of their small size it was excluded from farther analysis. Similarly, that zone is occupied by small blue spots – sky visible through openings in the leaves – and consequently dropped from the analysis. Example of segmentation and interpretation resulting in "trees" extraction is shown in Fig. 5.

Water

Water is created from the same material as "sky", i.e. from blue spots and from gray spots minus spots recognized as sky or clouds. It was postulated that images that have "water" must have "sky" (evident restriction on the class of recognizable images). Each of selected spots went through number of tests analyzing geometrical and positional characteristics of the spot: width, height, touching the frame on left or right, flatness of the spot's top border. Example is shown in Figs 2, 5 (d) and 5 (i).

In some cases there is no detectable border between sky and water (no visible horizon). We use two features to divide the combined spot: 1) mostly brightness of the sky increases from zenith to horizon (from top of the image toward the bottom); brightness of water mostly decreases from horizon down, and 2) when the spot combines into one sky and water, the border between them (the horizon) is often located in the narrowest part of the spot (see Fig. 2). In case when water doesn't contact the sky another feature useful for identifying "water" appears – existence of shadows of objects located on the far banks of lakes or bay (see the same image Fig. 2). Shadows in water can be recognized by horizontal symmetry of contours of objects.

Ground

The Earth surface is a general background for vast majority of images. In some subclasses of images it is completely covered by other objects (like in indoor scenes), in majority of outdoor scenes the Earth surface is covered only partially (by trees, buildings, cars and so on). Visible surface can appear differently: as lawn, road, or plaza. We will call it "ground". It occupies significant area of image (greater than 4%) and touches bottom of the image. Despite the simplicity, it works in many cases.

Another kind of ground is not gray but green. Spots, that satisfy all positional and geometrical characteristics of ground and are qualified as "green", become "green ground". Various types of vegetation can appear as "green ground": it could be grass, plants, bushes or forest (as it is seen from mountains). It seems that differentiation of these classes could be done using texture characteristics (like autocorrelation function).

A particular kind of ground is the road in perspective. The color and texture characteristics of the road are the same as of the ground, but it has very specific geometrical characteristics. Because borders of the road in reality are parallel the width of the road on the image will decrease gradually as the distance to the observer increases. When the road is a straight line the width become a linear function of distance, and the position of the horizon could be found.

Mountains and snow

Take a look at Fig. 4 (a). The regular description of this image would be "polyline".

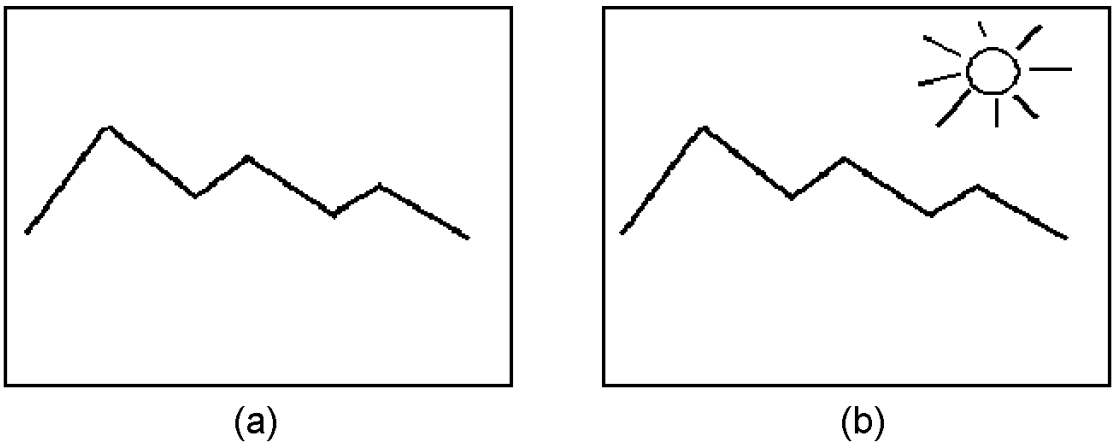

Fig. 4

Let's modify the image – ad a feature that would create perception of the "sky" – see Fig. 4 (b). Now mountains appear on the image. It shows that "sky" is a very creative ingredient of an image. The top, left, and right borders of sky in many cases are silent – they contain little or no information on the image. To the contrary, the bottom border is highly informative. In majority of landscapes the bottom border of sky is water, or green, or buildings, or mountains. Therefore the simple rule is: if the spot under sky was not recognized as water, or green, or buildings, it must be mountains. Of course, some restrictions on color, texture, size, and geometry have to be applied. Buildings that appear in front of sky can be identified by two procedures. One is the subroutine (detector), which finds buildings (see below), and the second one is a procedure that analyzes the bottom border of the sky spot looking for strait lines in general and vertical ones particularly.

If mountains will be covered with snow, spots that represent patches of snow are located between the mountains and the sky. If such spots exist and they satisfy some conditions then a statement is issued "mountains with snow" – Fig. 5 (c).

Buildings

The most characteristic features of images of buildings are straight lines – vertical and horizontal. Typically buildings are not represented on image as a single spot because different walls having different luminosity, different parts of building having different colors (windows, walls, roof). All these parts appear as spots with linear borders. Vertical borders are an invariant of buildings. Lines, which are horizontal in nature in most cases appear on images in perspective as a bunch of lines with a singular imaginary apex. Majority of buildings have a periodical structure of windows. In urban scenes perspective lines are a valuable source of information: 1) it helps to establish references between objects in the scene and its in-depth location, 2) if one object is recognized (like human, car or window), it allows estimating possible size of similar objects in other locations as shown in Figs 5 (e) and 5 (h).

Cars

In our list of basic notions, which we propose as the first step in building image understanding, "car" is the first notion of quite different nature – it is defined by its geometrical form. The difficulty of recognizing cars is caused by the fact that cars look essentially different from different points of view. Despite all cars having similar main components, proportions are quite different and that increases difficulties of recognition.

There are programs for car recognition that are based on total search of cars in every point of an image and comparison of given fragment with all possible appearances of a car (at different angles and different distances). We acknowledge that with gigantic speed and memory size of computers that problem can give satisfactory solutions in many applications. But from the history of AI we know that that kind of solutions have very restricted areas of application. What we try to do is to find more intellectual tools that could deliver more general solutions.

Let us deconstruct a modern vehicle to its ancient ancestor – keep only the wooden board and wheels – it is still a vehicle. There are some limitations that put restrictions of width of the base to wheels' diameter (roughly from 2 to 5). That means that the space under the vehicle is always in deep shadow. That feature doesn't depend on construction of the vehicle (with the exception of exotic cases). It is obvious that finding the darkest spots on the image will create a number of false alarms (deep shadows in trees, open doors in buildings, black paintings, etc). But it is also obvious that a number of natural restrictions can be applied. If the dark spot is inside the "green" area, or on the top of the image it is very unlikely to be an indicator of a car. There are also some general indications that there is a car over the dark spot: the car is mostly represented by a number of spots, and there are some geometrical restrictions on these spots (in size and relative locations). Majority of car images have straight lines. Some false alarms can be disaffirmed on the level of reinterpretation. To find out if this approach would work could be done only by testing. An illustration is presented in Fig. 5 (e).

(a) Blue sky, Montains, Green

(e) Green, Blue sky, Ground, City, Building, One tree on the left, One three on the centre, Cars

(b) Blue sky, Clouds, Green ground, One tree on the right, Distant forest

(f) Green, Blue sky, Ground, One tree on the right, Trees on the left, Distant forest

(c) Blue sky, Mountains, Snow, Green, One tree on the right, Ground

(g) Green, Blue sky, One tree on the left, One tree on the right, One tree in the centre, Distant forest, Water, Mountains

(d) Green, Blue sky, Ground, One tree on the right, Water, Mountains

Fig. 5 (a-g)

(h) Gray sky, Ground, City, Buildings

(i) Sky, Green, Ground

Fig.5 (h-i)

Reinterpretation

As Gestalt psychology claims, interpretation of a part of image depends on interpretation of the rest of the image. All described above is mainly interpretation of each spot independently of other objects in the image. But even on that basic level of image understanding we were forced to reinterpret some notions. When searching for the “sky” we take in consideration if the candidate for the sky is surrounded by trees. When looking for a car and finding one, we look around for more cars using less restrictions. If a car was found, we would look for ground with softer criteria. If sky is found, we would look on geometry of surrounding spots, and sometimes divide the sky spot into two spots: sky and water. If white spots are surrounded by sky, the spots are interpreted as clouds, but if white spots are surrounded by mountains, these spots are interpreted as snow.

Of course many other rules of reinterpretation must be (and will be) implemented at the first level of image understanding. According Gestalt psychology when dividing an image into parts two conditions have to be satisfied: 1) each part has to be meaningful, and 2) the whole has to be meaningful. On the first level we take care on the parts, on the next level we must take care on the meaning of the whole picture.

From description of DC algorithm it is clear that at first steps it outlines big spots. In the outdoor scenes it will be “sky”, or “ground”, or “green”, or “water” and they could be immediately categorized as one of these notions. As it was mentioned above, knowing one of these notions helps in segmentation and interpretation of other spots (completely in accordance with law of Gestalt psychology [8]). That is a big advantage of our “top-to-bottom” scheme of segmentation over “bottom-to-top” ones (like starting with gradients, or combining objects from pixels, which allow to start interpretation after segmentation is finished).

CONCLUSION

Last two decades were marked by appearance of programs that succeeded (or partly succeeded) in solving some sophisticated AI problems (like chess game, or search images for images of faces or cars). Solutions are mainly based on complete enumeration of possibilities plus a number of heuristic restrictions. The success is determined to a great extent by high speed of computers and tremendous size of memory. Chess programs are looking at many as possible positions going as deep as possible. Search for cars keeps in memory 8 (or as much as you can) different car views and expanded them to 4 (or more) different sizes. Every piece of the image is compared to stored images. The piece with good match is called “car”. Same approach was used in face finding.

But if we found reasonable to search in the image for such concepts as sky, green, ground, mountains etc, full search approach is useless because what examples will one store as representatives of such concepts? No way! We don’t claim that our approach is better, we only want to show that different approach is possible and sometimes inevitable.

REFERENCES

1. Rosenblatt F., The Perceptron: a probabilistic model for information storage and organization in brain. *Psychol. Review*, 1958, vol. 65, no 6, pp. 386–408.
2. Bongard M., *Pattern Recognition*. New York: Spartan Books, 1970.
3. Guberman S., Izvekova M., Holin A., Hurgin Y., Solving geophysical problems by mean of pattern recognition algorithm, *Proc of Acad. of Sci. of the USSR*, 1964, vol. 154, no 5.
4. Vasiliev Ju., Gelfand I., Guberman S., Shik M., Interaction in biological systems, *Priroda*, 1969, no 6 (in Russian).
5. Maximov V.V., A system learning to classify the geometrical images. In: *Models of Learning and Behaviour*. Ed M.S. Smirnov, M.: Nauka, 1975, pp. 29–120 (in Russian) – the article was translated in English by Marina Eskin in 2003 and is available (PDF, 3 MB) at the site http://www.cogsci.indiana.edu/farg/harry/res/bps/maksimov/Maksimov_Bongard_problems.pdf
6. Jing Y., Baluja S., PageRank for Product Image Search, In: *WWW Conference 2008*, http://www.esprockets.com/papers/www2008-jing-baluja.pdf
7. Mel'čuk I., *Dependency Syntax: Theory and Practice*. New York: State University of New York Press, 2010.
8. Guberman S., Minati G., *Dialogue about Systems*, Milan: Polimetrica, 2007.
9. Ford A., Roberts A., *Colour Space Conversions*,1998, http://www.poynton.com/PDFs/coloureq.pdf