# RISCOS: Retrieving Images using Sketches

Luís Carlos Mendes Rosado

Instituto Superior Técnico, Portugal

May 2014

***Abstract:*** Currently we store a lot of images, because of the huge availability of capture devices (mobile phones, cameras) and also because of the daily growth of digital images. Due to this large number of images, it becomes a difficult task for users to find the desired images.

Existing solutions perform search based on text and examples images. Thus it is essential to develop a solution that allows searching and retrieval of images using sketches. Given the absence of a solution, with our work we tried through a common representation of a sketch and an image to obtain a possible comparison between them. We began by performing a pre-processing of the image and the sketch so that we could simplify them and obtain an image and a sketch very similar between them. This was followed by a conversion to a common format so that it was possible to identify the spatial and geometric relationships of the objects in the image and sketch. Identified the characteristics, we converted them in descriptors, to make a comparison between the initial images and sketches.

The evaluation of the solution consisted in analyzing the retrieval of images using of sketches and images.

***Keywords:*** Recovery of images, Queries using sketches, Edge detection, Simplification of images

## 1 Introduction

Due to the high quantity of accessible images, there is a need to innovate the technology related to its search. Currently it is possible to perform the search of images through text and sample images. Users can through keywords find images and can add color to better translate their desire. It is also possible to perform the search through an image, and through this one, find images that are similar to the query image.

The searches that the users have access become difficult, because they cannot express in words what they pretend, or they do not have an example of an image that identify or express their thoughts. Due to this, the results are not always the most desired, presenting inefficiencies by the information given, and not being related to the search required by the user.

Areas like images retrieval and edge detection on images, have been explored solutions that develop and enhance the existent searches, but fall from being limited and do not provide a solution to the problem. The limitations of these techniques are due to the lack of a solution to perform searches of images using sketches.

Having the need to explore other forms of searches that fit in the visual perception of the user and facilitate the specification of the interrogations, we explored a new solution to overcome these challenges. The solution that we developed, reconciles the two areas, *recovery of images through sketches and techniques for detecting edges in images*, giving the possibility of image search using sketches as queries.

Seeking to develop a solution that allows users to find images using sketches, facilitating the search of images that belong to a collection of images, it will be necessary to find a representation that will enable the comparison of images with sketches. From this idea, for the next section, we studied solutions that made retrieval of images through sketching, and solutions that convert images into sketches by edge detection, to aim to find a common representation. In section 3, developed solution, we describe the processes that allow to convert an image into a common outline format starting by performing the pre-processing of images or sketches by combining techniques of simplifying images, yielding images to outlines in order to approach a sketch. After it, we converted the images to common format. Subsequently, we applied an algorithm for the extraction of information that gives the spatial and geometric relationship of the elements of images and sketches. Obtained this information, we resorted to the area of information retrieval, which contains techniques that allowed the storage of information and make a comparison between the images and sketches, presenting the results.

In section 4, user evaluation, we make a comparison between *TopoGeo* and *GeoDiff* algorithms and evaluate them in Section 5, results and findings, presenting the results obtained in the same solution.

In Section 6, conclusions and future work, we carried out a comprehensive assessment of the solution and present some work to do in the near future.

## 2 Related Work

In this section we will introduce some work related with the target solution. Focusing on the objective of the image retrieval through sketching, we centered the research in the areas of *edge detection* and *image retrieval*.

### 2.1 Image Retrieval by Sketches

The authors Yang Cao et al. describe a technique that converts an image into a sketch from the edge detection [1]. Subsequently it extracts the most relevant features of the image and compare with the sketch drawn by the user. The technique chosen for the comparison between images was *Chamfer Matching* and its variants oriented for efficiently presenting the edges comparison. The *Chamfer Matching* technique consists in calculating the distance between the outline of the image and the edge outline. The technique *Oriented Chamfer Matching* is used to detect the orientation of the edge of the image in relation to the outline. Each picture is assigned a contour map, comprising six types of guidance: from -15º to 15º, 15º to 45º, 45º to 75º, 75º to 105º, 105º to 135º, and from 135º to 165º. Depending on the similarity between the direction of the image edge and the sketch, a hit is assigned to the corresponding edge technique *Indexable Oriented Chamfer Matching*.

This technique is an array of inverted indices composed by all images, to which the similarity between the sketch and all images are calculated.

The use of these techniques allows us to process an image in a format similar to a sketch, being this a important factor for the solution. It is apparent from the results produced by the use of this solution they are not always efficient (see figure 1).

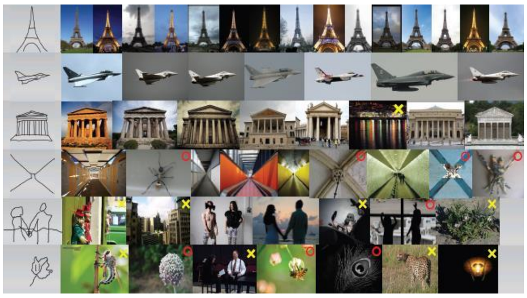


**Figure 1.** Example queries and the corresponding top results. According to subjects' labeling, images marked by yellow cross indicate that they are irrelevant with the corresponding sketch query in terms of both structure and conceptual meaning; while images marked by red circle mean that although they are not the same concept of the sketch query, these images still have very similar structures to the query.

In [2] John Collossome, et al. describe a technique that retrieves images and locates objects similar to a sketch in a picture. The authors first use a *Laplace Gaussian* operator, to soften the surroundings of the object of the image and of the sketch. After building a gradient histogram for the image and the outline, using a window of n x n coursing the images. The construction of the histogram is performed according to the orientation of the pixels in each image. Then the distance is calculated between these histograms, using the standard $L^2$ to find existing intersections between images. The location of the object in the image is performed by *RANSAC* technique [2]. Once identified the object in the image, a frame is created using the template outlined object. This technique demonstrates that works for color images (see figure 2), locating the object outlined.

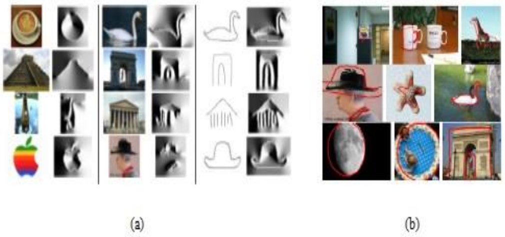


**Figure 2.** (a) Localizing the query sketch within retrieved images. (b) Sample images and query sketches.

In [3] it is described a technique that retrieves images of faces based on sketches. The authors Hamed Galoogahi and Terence Sim, use a *Gabor filter* to attenuate thick textures, such as eyes, eyebrows, nose, chin and ears, characteristics that form a face. Subsequently, these textures are modeled by applying the *Radon Transform*.

By applying these two techniques, we eliminate fine textures such as wrinkles, freckles, shadows or reflections. Given a black and white image of a face, it is applied a *Gabor filter* that calculates the *magnitudes* of *Gabor*. Each magnitude is divided into blocks, and a histogram is constructed that represents a set of *Radon* samples. Then, we concatenate all histograms, yielding a single histogram representing the form of *Gabor*. This representation is the image of the face obtained in draft form (see figure 3).

The use of the technique is limited to the contribution of the solution, for only fit to images with faces.

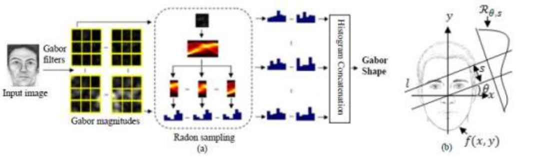


**Figure 3.** (a) The framework of Gabor Shape face representation, (b) Radon transform of an image.

## 2.2 Edge detection in images

The authors Mathias Eitz, et al. in [4], describes a technique for extracting the edges named *Edge histogram descriptor*, which was proposed for the study of MPEG 7 standard. Although it seem useful for such studies, it can be applied to our case which helps the extracting of the edges of an image. This approach consist in dividing the image into blocks, which is proportional to the image size division. In each block all pixels are traversed by five different shades. These masks are characterized as vertical edges, horizontal, direction in 45 degrees, 135 degrees and edges without direction. Each mask has a size of two by two, to subsequently be calculated by using a distance. Considering the greatest value by calculating the distance, becomes a decision as to the direction of the edge of the pixel in the image. That said, it is constructed for each block of the image a histogram containing the number of directions of each edge. When analyzing the histogram, we verify which route has the highest value in the histogram, being this route, considered the strongest in the block boundary. This technique applied to an image and a sketch gives the histograms edges of each block. Its use shows the direction of edges in each block (see figure 4). Applying the difference between histograms, one can measure the similarity between both the images.

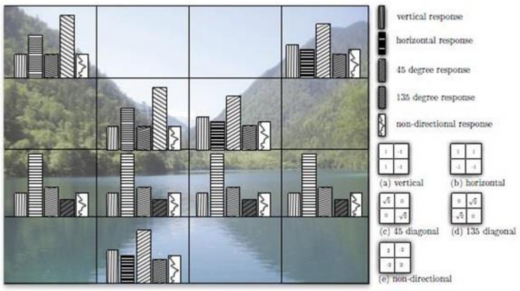


**Figure 4.** Left: the EHD distinguishes 5 types of edges for each cell and stores the values in a 1D array. Bottom right: five edge filters used for the EHD.

## 2.3 Related work discussion

After describing the work carried out above, other techniques were analyzed. Regarding the techniques [5] and [6] it was found that they only deal with monochrome images. The technique [4] to use edges masks, can analyze the different features present in the image, while the technique [7] refers to the chain code algorithm to extract the edges in an image. The technique [8] starts by performing a pre-processing to eliminate the noise present in the image, followed by detection of the edges through the application of an edge detector. For exchange, the techniques [9] and [10], stand out from the rest because of the way they can deal with colored images. In the case of the technique [10], it has become limited due to the application of the edge detector, which results in a black and white image and therefore in the loss of the colors in the image.

Cited and analyzed the techniques studied, it appears that all have limitations. As mentioned, some techniques are limiting because only deal with sketches and monochrome images, discarding the color images. Another limitation is evident that none presents a technique to convert an image and a sketch to a common format to both. Such a process, is essential to compare the similarity between both, and there are no techniques that gave characteristics related to images and sketch.

## 3 Developed Solution

The solution is divided into two main parts, one to address images and another to treat the sketches (see figure 5). Regarding images, we apply a pre-processing and a simplification to obtain a picture closer to a sketch. After, we converted it to a format which allows to extract information from objects and compare it with the sketch. In the case of the sketch, we made a pre-processing making it simpler, converting it into a format that allowed extracting its features.

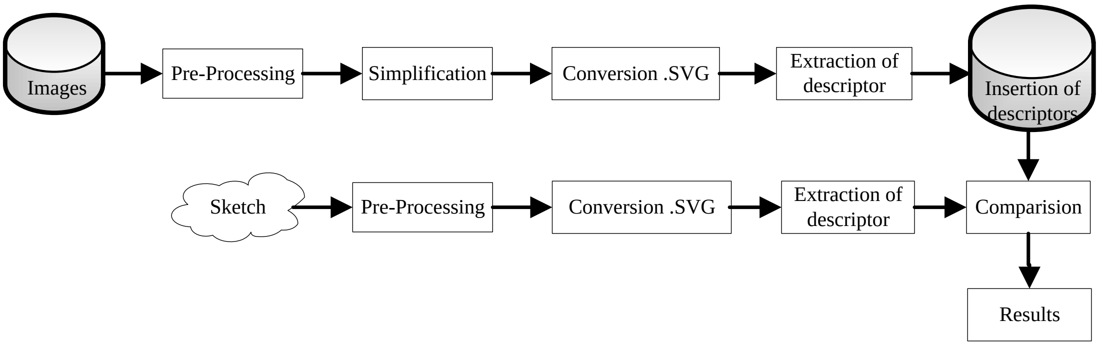


**Figure 5.** Architecture of the solution

## 3.1 Converting *raster* image to *vector* image

This section provides all the processes used to convert a *raster* image into two *vector* images. We began by implementing a pre-processing to the image by applying two distinct techniques *Canny Filter* and *Combination of techniques*. Concerning the first, it does the detection of all the edges in the image, preserving all the detail that forms the *raster* image. The application of this filter yielded a greater number of objects associated with the image (see figure 6).

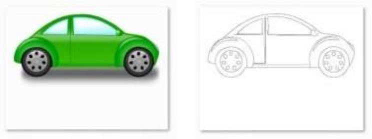


**Figure 6.** Process Canny Filter. Left. Image to process. Rigth. Canny Filter

Concerning the second technique, this consisted of a combination of a *Gaussian Blur Filter* with a set of techniques. This combination provided the removing of some noise in the image as isolating or eliminating irrelevant objects in the image, resulting in an image edges. The application of *Gaussian Blur Filter* softened the image while preserving the edges of the same. Then we used the technique *binarization* that eliminated isolated and irrelevant objects from objects. Finally, we created the *Find Edges* function evidencing the overall edge of the object in the image (see figure 7).

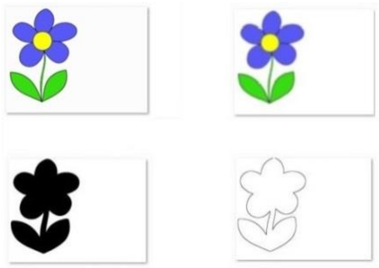


**Figure 7.** Combination of techniques Process. Up Left. Image to Process. Up Right. Gaussian Blur Filter. Down Left. Binarization. Down Right. Find Edges

This combination of techniques resulted in an overall edge of the object in the image. After performing the pre-processing of the image, it was exposed to a simplification, this step consists in reducing some noise, as small holes that may exist in the image. The completion of this step implies three operations, a *threshold*, to minimize image noise, a *close* to eliminate small holes that still exist and the application of the technique *skeletonize*, who got a thin contour of the objects and is crucial to perform the passage of the *raster* image into *vector*. After the steps of pre-processing and simplification are made, we resorted to techniques of the library *Batik* that allowed the conversion of a *raster* image into a *vector* format.

## 3.2 Converting *raster* sketch to *vector* sketch

This section sets up the processes undertaken to the sketches, since the pre-processing, to the conversion to a *vector* format.

The pre-processing performed to sketches, can be divided into two distinct processes, *Detailed Filtering* and *Simplified Filtering*. Describing *Detailed Filtering*, this consist in preserving all the detail in the sketch, just removing the imperfections contained in a drawing in sketch form. This process combined the *Gaussian Blur Filter* that softened the outlines of the sketch with, the *Binary* techniques, that converted the sketch to a level of 8-bit colour and the *Skeletonize* which resulted in getting a thin contour of the objects outline (see figure 8).

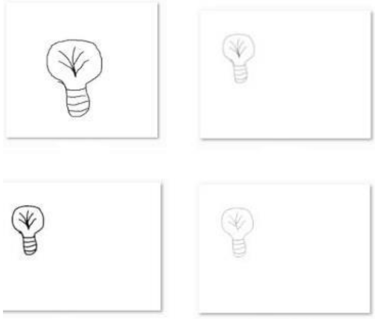


**Figure 8.** Detailed Filtering Process. Up Left. Image to process. Up Right. Gaussian Blur Filter. Down Left. Binary. Down Right. Skeletonize

Regarding *Simplified Filtering*, this consists of a combination of techniques that allow the elimination of information describing the object, maintaining only the outline surrounding the object of the sketch. After we applied a *Gaussian Blur Filter* to soften the contours of the outline. Next, we used the Binary Data and technical *Fill Holes*, resulting in a sketch in black and white with existing holes between the objects of the sketch. Finally, we applied the technique *Find Edges* and *Skeletonize*, obtaining this way a thin outer contour to proceed to the conversion of the sketch to a vector format (see figure 9). Performed the steps of pre-processing to the sketches, made up a conversion of these to a *vector* format, as image, to extract the descriptors.

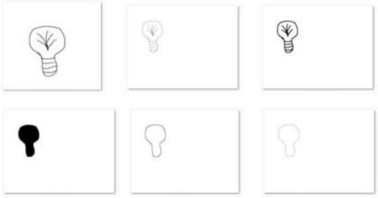


**Figure 9.** Filtering Process Simplified Results. Up Left: Image to process; Gaussian Blur Filter; Binary. Down Left: Fill Holes; Find Edges; Skeletonize

## 3.3 Extracting the descriptors

In this section we deal with all the processes involving the extraction of the characteristics of objects in the image and sketches. These procedures were carried out after the filtering performed by implementing the algorithm *GeoDiff* to images and sketches in which was preserved all the detail. This algorithm consists in extracting topological and geometric information of objects. The *Geometry* algorithm applied to the pictures and drawings in which we only obtained the overall outline of the object, consisting in calculating the overall outline of the object geometry of the images and sketches.

### 3.3.1 GeoDiff

The *GeoDiff* algorithm consists in generating a graph from the ratio of the different polygons image resulting from the filtering process. The graph consists of a root node, this node corresponding to the image as a whole and for several objects, representing each of the different polygons that make up the image. In the formation of the graph there is a relationship between nodes, this link may be considered for inclusion or adjacency. A node is included when a polygon is inside another polygon. Node adjacency are those whose vertices of two polygons are neighbours to each other. When a polygon is included in another, it is considered a parent- child relationship, being the son of the contained polygon that contains the parent polygon. In this case, it adds to the graph a new level of depth, resulting in an adjacency relation between all nodes that are at the same level of depth (see figure 10).

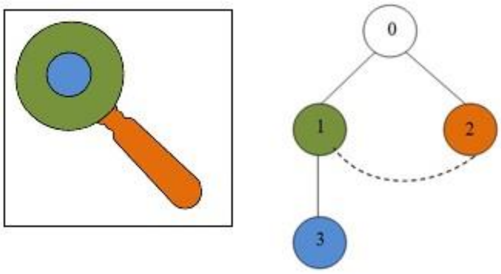
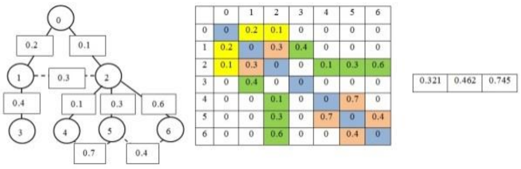


**Figure 10.** Left. Image composed of included objects and adjacent. Right. Formation of the graph from an image

After building the graph, which contains a list of the different objects of the image, we calculated the geometry of each node, using the *Cali* library [11]. In the case of the root node, and this being the initial node of the graph, we considered the geometric of its descriptor has all its components with the maximum value 1. We assumed this value, since we consider this, as the largest amount that any node can get from the geometric calculation. Obtained the geometry of the nodes, we assign weights to links between nodes that will form the graph. There is a relationship of adjacency and inclusion with different geometrical connections between the weights, we constructed an adjacency matrix of the graph, representing the number of rows and columns by the number of adjacent nodes and included, respectively. Each array position corresponds to the value of the weights assigned to the links. From the adjacency matrix, we obtain a vector that allows the calculation of similarity between the image and the outline (see figure 11).



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0.2 | 0.1 | 0 | 0 | 0 | 0 |
| 1 | 0.2 | 0 | 0.3 | 0.4 | 0 | 0 | 0 |
| 2 | 0.1 | 0.3 | 0 | 0 | 0.1 | 0.3 | 0.6 |
| 3 | 0 | 0.4 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0.1 | 0 | 0 | 0.7 | 0 |
| 5 | 0 | 0 | 0.3 | 0 | 0.7 | 0 | 0.4 |
| 6 | 0 | 0 | 0.6 | 0 | 0 | 0.4 | 0 |

| | | |
|---|---|---|
| 0.321 | 0.462 | 0.745 |

**Figure 11.** Process used to calculate the descriptor image and outline

### 3.3.2 Geometry

The algorithm *Geometry* consists in calculating the overall contour of the object in images and sketches geometry. This algorithm was applied using the *Cali* library that has techniques for extracting the points which form the contour of the object (see figure 12), yielding a geometric descriptor of the image and outline.

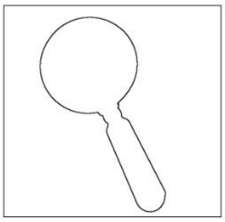


**Figure 12.** Image captured by the overall contour of the object

## 3.4 Image Retrieval

In this section we present a comparison between an image and a sketch, recurring to a multidimensional index structure of the images *NB -Tree* [12]. This structure allows a greater speed in the image storage using indexing techniques that store a considerable amount of images. Then it is made the calculations of similarity between images and sketches from the topology and geometry descriptors obtained in the step extraction of descriptors.

### 3.4.1 Insertion of the descriptors

The insertion of the descriptor occurs after obtaining the descriptors topology and geometric difference of objects, such as the geometry descriptor from the geometric calculation of the overall contour of the object in the images. To store up this information, we used a *NB - Tree* indexing structure considered by Basgalupp [13] a great tool for storing and finding information. This allowed the storage directory to save the images, the *id* and the values of the descriptors.

### 3.4.2 Comparison

Obtained the database containing the collection of images, we make a correspondence between all images and the sketch. To the sketch it is performed the step of pre-processing, followed by calculating the geometric descriptors and the topology of the difference mapping the overall contour of the object geometry. After obtaining the descriptors, we compared them using the same metrics. This comparison was made by pairing the descriptors of topology and geometry of the images with the sketch as well as the descriptors calculated by the overall contour of the object of the image and the sketch geometry. After obtaining the results, we sort the values and increasingly present the best values, these correspond to the most similar images to the sketch.

## 4 User Evaluation

In this section we present the differences between two algorithms. Consisting the implementation of an algorithm *GeoDiff* an evolution of the algorithm *TopoGeo* developed in a previous work by Fonseca [14] and both focusing on extracting a descriptor topology and geometry of objects in an image, they differ in their implementation. Regarding *TopoGeo*, this uses a graph topology and geometry, consisting of links between nodes, being these included or adjacent. In the case of being included, it is attributed by default for all connections, the geometric value 1, being only calculated the geometry for the links between adjacent nodes. This geometry is given by a vector with 11 additional nodes, each corresponding to one of a geometric characteristic. Each of the nodes of the graph, which represents an element of the image, performs a match for the 11 additional nodes, assigning a weight equal to the value of this geometric feature for this element (see figure 13). In the case of *GeoDiff*, the weight of the link between the parent node and the child node is different, it is calculated by the geometric weight between the root node of the graph and the nodes that are descendants of this, as it is calculated for the remaining connections to the geometric weight between the nodes included and adjacent (see figure 14).

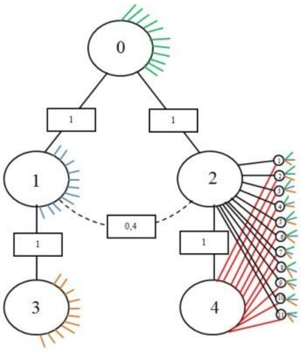


**Figure 13.** Formation of the graph using TopoGeo

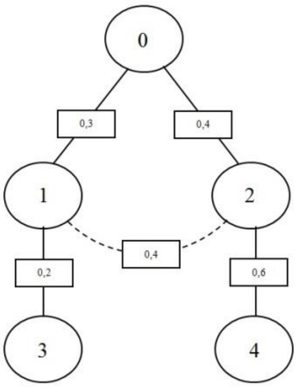


**Figure 14.** Formation of the graph using GeoDiff

Presented the differences of implementation between the algorithms *TopoGeo* and *GeoDiff*, it is evident that the *TopoGeo* presents a greater complexity with respect to *GeoDiff*, due to the way it describes the geometry of the picture elements. In the absence of this complexity, it was decided to implement the *GeoDiff*, consisting in grouping it in the topological and geometrical information of the graph, reducing the complexity of the same, yielding the same topology and geometry of the elements in the image.

# 5 Results and Findings

In this section we present the evaluation tests of performance from the solution compared to other existing algorithm. The searches were performed using images and sketches as queries, and a subsidiary data base composed of 100 images, divided into 10 categories. It was also used with other database images, about 13000, which included the 100 images of the previous database.

The realization of the tests was to evaluate the performance of two algorithms, *TopoGeo* and *GeoDiff*, as the combination of each of these with *Geometry*. For each test it was used the metrics *Precision & Recall*, the *Mean Average Precision* and *R- Precision*.

After we performed the analysis of the results obtained by the images and sketches as queries, it was concluded that the *GeoDiff* algorithm, though less complex against the *TopoGeo*, in some cases experienced similar behaviour to *TopoGeo*. When combined each of them with the *Geometry*, there was a greater number of images retrieved. This fact is due to the combination of a descriptor that extract the spatial relationship and geometry of objects of images that only preserved the detail with a descriptor that calculates the global geometry of the contour of the object in the images, enriching the solution through the use of two descriptors.

## 5.1 Discussion of results

Obtained the results, it is noted by the metric *Precision & Recall* (see figure 15 and figure 16) that the *GeoDiff* presented results similar to the *TopoGeo*. When combined each one of these with the *Geometry*, we verify that the results were better and the precision improved 10% in comparison with the algorithms without the geometry. The usage of the combination *GeoDiff* with *Geometry* became an advantage to the solution, given that the results obtained were better than when we used only *GeoDiff*. This combination allowed the conciliation of the information retrieved in the graph of the topology and the geometric difference with the information obtained by the calculation of the geometry of the object, making the results more accurate.

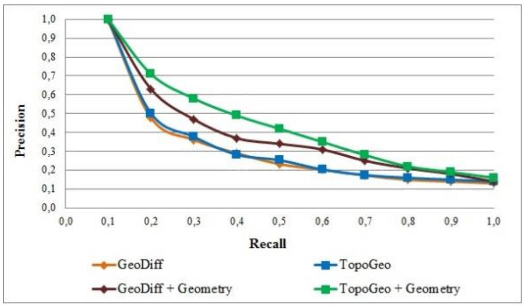


**Figure 15.** Results Precision & Recall of query by images

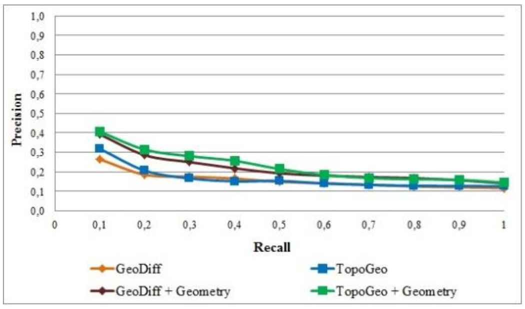


**Figure 16.** Results Precision & Recall of query by sketch

When the results are analysed by the metric *Mean Average Precision* we verify that the algorithms *TopoGeo* and *GeoDiff* present an average accuracy of 20% for images as queries and 5% to the sketches as queries (see figure 17 and figure 18). For its part, by combining these algorithms with *Geometry*, the average precision improved by 5% to the images with queries and 3% to the sketches with queries. As to *TopoGeo* this improved by 10% in the images with queries and 5% to the sketches with queries (see figure 17 and figure 18). After obtaining the results and analyzing the behaviour of the algorithms in a global perspective, we conclude that the usage of the *Geometry* improved and enriched the solution, resulting in a larger number of images relevant in the top 10 positions of each query.

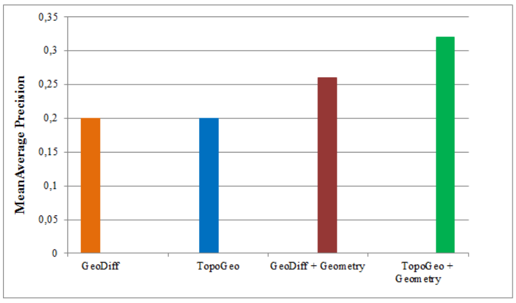


**Figure 17.** Results Mean Average Precision of query by images

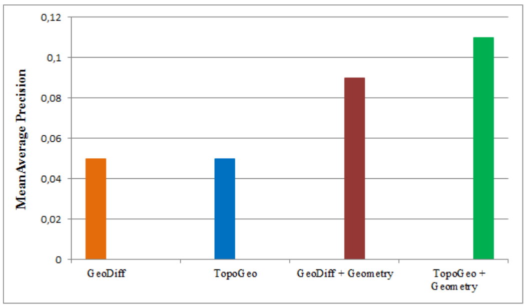


**Figure 18.** Results Mean Average Precision of query by sketch

After analyzing the results of the metrics *Precision & Recall* and *Mean Average Precision* it was analysed the results obtained by the metric *R-Precision*. It was verified that when using images as queries, the algorithm *TopoGeo* got the best *R-Precision* 11%, followed by the *GeoDiff* with 6% and the combination of *GeoDiff* and *Geometry* and *TopoGeo* and *Geometry* both with 3% (see figure 19). After this results were obtained, we conclude that the algorithms without the combination of *Geometry* got a better behaviour, due to the database of images possess a geometry very identical to the images used as queries.

Analysing the results obtained by the sketches as queries, it is noted by the figure 20 that the *R-Precision* obtained by all the algorithms was 1%. Such result is due to the reduced number of images recovered for the top 100 positions.

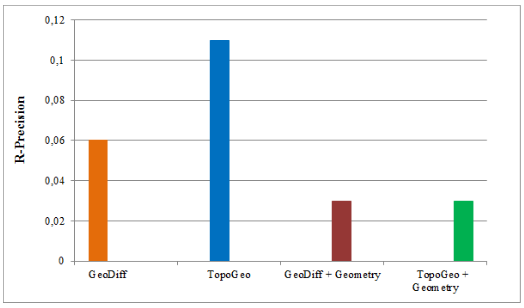


**Figure 19.** Results R-Precision of query by images

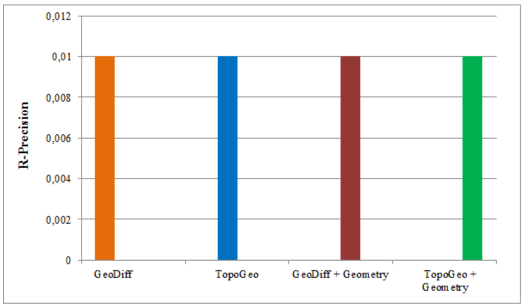


**Figure 20.** Results R-Precision of query by sketch

In short, and after the results are verified by the different metrics, we conclude that the *GeoDiff* algorithm, presents a lower complexity and strength in comparison to the *TopoGeo*, it managed to obtain similar results to *TopoGeo*. When combined each of these algorithms with *Geometry* it is noted that the results were even better, obtaining a bigger number of relevant images. This is due to the combination of an algorithm that extract the topological information and calculates the geometric difference between the objects with an algorithm that calculates the geometry of the global outline of the object in the image.

## 6 Conclusions and Future Work

After analysing the experimental results done to the solution, we concluded that the application of two filtering, which consisted in preserving the detail, and another in the obtainment of the global outline of the object in the image, enriched the search of the images.

Another essential contribute was the combination of the two descriptors, *GeoDiff* and *Geometry*. In case of the *GeoDiff*, the contribute focused in the low complexity presented in the extraction of the topological and geometric information of the elements of the images, grouping in a graph both information, while *TopoGeo* is a lot more complex and contains separately the geometric information of the elements. For its part, the *Geometry* provided the calculation of the geometry of the global outline of the object obtained in the image. The combination of both descriptors *GeoDiff* and *Geometry*, turned the searches of images better, presenting a higher number of relevant images in the solution.

Although there are contributions, the solution has some limitations that can be exploited in the future. One of the gaps, concerns the filtering made with to the sketches and images, which would rule out the colours, but retain the same topology. Existing this condition in the solution, one of the paths to explore in the future is the use of colour. The combination with this alternative solution, enrich the researches, considering the images with the same topology but also with the same colour.

Another aspect that could enhance the solution would be to implement a new algorithm, which is more effective in extracting characteristics by providing an improved detection of the information in images, making the similarity calculations more accurate.

## References

[1] Y. Cao, C. Wang, L. Zhang e L. Zhang, "Edgel Index for Large-Scale Sketch-based Image Search," em *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR),* 2011.

[2] R. Hu, B. Mark e J. Collossome, "Gradient Field Descriptor For Sketch Based Retrieval and Localization," em *In Proceedings of Intl. Conf. on Image Proc. (ICIP),* Hong Kong, Setembro 2010.

[3] H. Galoogahi e T. Sim, "Face Photo Retrieval by Sketch Example," em *Proceedings of the 20th ACM international conference on Multimedia - MM '12,* Nova Iorque - USA, 2012.

[4] M. Eitz, K. Hildebrand, T. B. Boubekeu e M. Alexa, "An evaluation of descriptors for large-scale image retrieval from sketched feature lines," em *Computers & Graphics*, 2009.

[5] M. Silva e A. Candeias, "Um Novo Algoritmo para Detecção de Contorno de Melanomas em Imagens de Pele," em *Sibgrapi - Workshop of Undergraduate Work,* Brasil, Outubro 2008.

[6] H. Matos, "Reconhecimento Biométrico Baseado na Geometria da Mão," em *Tese de mestrado integrado em Engenharia Electrotécnica e de Computadores,* Porto, 2010.

[7] R. Santos, "Reconhecimento de Esboços de Formas Geométricas Contidas em Fluxogramas," em *S.W.I.B - Simpósios SBSC, WebMedia, IHC e SBBD*, Brasil, Outubro 2010.

[8] R. Zanin, A. Dal Poz e É. Martins, "Esqueletização na Modelagem e Extração de Cruzamentos Simples de Rodovias," em *VI Workshop de Visão Computacional*, 2010.

[9] Y. Chen, V. Roussev, G. R. I. G. e Y. Gao, "Content-Based Image Retrieval for Digital Forensics," em *In Proceedings of the First International Conference on Digital Forensics*, 2005.

[10] A. Chalechale, G. Naghdy e A. Mertins, "Sketch-Based Image Retrieval Using Angular Partitioning," em *Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology*, 2003.

[11] M. Fonseca e J. Joaquim, "CALI: Uma Biblioteca de Componentes para Interfaces Caligráficas," em *X Encontro Português de Computação Gráfica*, Portugal, 1999.

[12] M. Fonseca e J. Jorge, "NB-Tree: An Indexing Structure for Content-Based Retrieval in Large Databases," Portugal, 2003.

[13] M. Basgalupp, "LEGAL-Tree: um algoritmo genético multi-objetivo para indução de árvores de decisão," em *Tese de Douturado em Ciências de Computação e Matemática Computacional*, São Carlos, 2010.

[14] M. Fonseca, "Sketch-Based Retrieval in Large Sets of Drawing," em *Dissertation for the degree of Doctor of Philosophy in Information Systems and Computer Engineering*, Instituto Superior Técnico, 2004.