An Image Similarity Measure based on Graph Matching

Ricardo Baeza-Yates
University of Chile
Department of Computer Science
Santiago, Chile
rbaeza@dcc.uchile.cl

Gabriel Valiente
Technical University of Catalonia
Department of Software
Barcelona, Spain
valiente@lsi.upc.es

Abstract

The problem of computing the similarity between two images is transformed to that of approximating the distance between two extended region adjacency graphs, which are extracted from the images in time and space linear in the number of pixels. Invariance to translation and rotation is thus achieved. Invariance to scaling is also achieved by taking the relative size of regions into account. Furthermore, the method provides a trade-off between pixel similarity threshold and approximation of the distance measure, which can be used to bound the error in image recognition as well as the time complexity of the computation.

1. Introduction

When comparing two images, it is a rather easy task for a human to render a qualitative judgement of the images being close or far apart. However, the efficient computation of the distance between digital images is one of the main open problems in low-level image recognition. There are many different approaches to image similarity, ranging from classical image recognition techniques to combinatorial pattern matching.

In this paper, yet another approach to image similarity is presented. A region adjacency graph is derived from each image and the problem of comparing two images is transformed to that of finding the best weighted matching in complete bipartite graphs. This technique of comparing images by matching region adjacency graphs is, to the knowledge of the authors, the first known reduction of image distance computation to a graph distance problem.

The rest of the paper is organized as follows. Distance measures over digital images are reviewed in Sect. 2. Extended region adjacency graphs are introduced in Sect. 3 and their extraction process is presented in detail. Image comparison based on matching extended region adjacency graphs is discussed in Sect. 4. In Sect. 5, experimental results concerning the extraction of extended region adjacency graphs from real images are presented. Finally, some conclusions are drawn in Sect. 6 and several open problems are discussed.

2. Distance Measures between Images

A digital image is a discrete function defined on a bounded regular grid of points in the plane and taking values in a set of gray levels. Without loss of generality, a digital image $A$ can be regarded as a set $\{A_{ij}\}$ of pixels, where each pixel $A_{ij} = (i, j, a_{ij})$ is defined by its spatial coordinates $(i, j)$ and gray level $a_{ij}$, or just as a set of gray levels, provided the topological structure imposed by the grid is remembered.

Comparison of binary images is a rather simple problem, because the image content can be extracted by assuming the set of black pixels to be the objects and the white pixels to be the background.

In the case of gray-level or color images, methods used for image comparison can be roughly divided into high-level image recognition and low-level image analysis. The former methods attempt to extract objects of interest by techniques such as thresholding, segmentation, and edge, shape, and texture detection, and then compare the objects. The latter methods, on the other hand, perform a direct comparison of images as whole entities by computing some measure of image similarity [6].

Definition 1. Let $X$ be a nonempty set. A function $\delta : X \times X \rightarrow \mathbb{R}$ is called a metric on $X$ if the following conditions hold, for all $x, y, z \in X$:

- nonnegative definiteness $\delta(x, y) \geq 0$, and $\delta(x, y) = 0$ if and only if $x = y$;
- symmetry $\delta(x, y) = \delta(y, x)$; and
- triangularity $\delta(x, y) \leq \delta(x, z) + \delta(z, y)$. 
The real number \( \delta(x, y) \) is called the distance between \( x \) and \( y \). The pair \( (X, \delta) \) is called a metric space if \( \delta \) is a metric on \( X \).

The most widely used distance-based measures for grayscale image comparison are the metrics based on the Hausdorff distance [8]. Given two subsets \( A, B \) of a metric space \( (X, d) \), the normalized Hausdorff distance between \( A \) and \( B \) is defined as
\[
H(A, B) = \max(h(A, B), h(B, A)),
\]
where
\[
h(A, B) = \max_{a \in A} \min_{b \in B} d(a, b).
\]
An image forms a metric space by taking \( d \) to be the Euclidean distance between pixels, for instance.

Among the desired properties of image distance measures, both for theoretical reasons and because of their usefulness in image comparison applications, are the following:

**Normalization** Their values lie in the closed unit interval [0, 1]. This property allows to minimize the effects due to different and non-uniform illumination [6].

**Robustness** Their values are invariant to translation, rotation, and scaling of the images [12].

**Metric** The distance measure satisfies the (topological) properties of a metric function [16]. These properties are useful, though not necessary, for searching in a metric space [5].

**Efficiency** The complexity of most measures based on the Hausdorff distance for comparing grayscale images is \( O(n^2) \), where \( n \) is the number of pixels in the images, although it depends on the operations used to compare the distance between single pixels in the images.

Computation of the normalized Hausdorff distance between two images takes, at first look, \( O(n^2) \) time, where \( n \) is the number of pixels in the images, although algorithms taking \( O(n) \) time are known [2, 17, 18]. A major drawback, however, is the lack of invariance to image rotation and scaling. Only invariance to translation has been achieved with the normalized Hausdorff distance [9].

The desired properties of invariance to translation and rotation are obtained automatically, though, when using a graph-based representation together with graph matching techniques. This fact is well known in the field of structural pattern recognition, where objects are represented by symbolic data structures, such as strings, trees, or graphs, and recognition is performed by matching these structures [4].

Moreover, invariance to scaling is obtained by matching extended region adjacency graphs, because vertex matching takes the relative size of the regions into account, as explained in Sect. 3.

## 3. From Images to Planar Graphs

A region adjacency graph is a directed, simple, planar graph extracted from a digital image. A matrix of pixel values defines a set of connected regions containing similar pixels, where two pixel values \( a_{ij} \) and \( a_{k\ell} \) are similar if \( |a_{ij} - a_{k\ell}| \leq \theta \) for a fixed threshold \( \theta \) of pixel similarity. Vertices are regions, and two vertices are adjacent if the respective regions have at least one neighboring pixel, where the four neighbors of a pixel \( A_{ij} \) are \( A_{i+1,j} \) and \( A_{i,j+1} \).

**Definition 2.** Let \( \theta \) be a fixed pixel similarity threshold. Two pixels \( A_{ij} \) and \( A_{k\ell} \) are neighbors in a digital image \( A \) if \( |i - k| + |j - \ell| = 1 \), and they are similar if \( |a_{ij} - a_{k\ell}| \leq \theta \).

Region adjacency graphs have been introduced in [3] for approximating region boundaries, and further used in [15] to find all region boundaries in a digital picture.

In an extended region adjacency graph, vertices are labeled by the relative size of the region they belong to with respect to the size of the whole image, and arcs are directed by increment in average pixel value of the respective regions.

**Definition 3.** Let \( A \) be a digital image, let \( \theta \) be a pixel similarity threshold, and let \( R = \{R_1\} \) be a partition of \( A \) such that two pixels \( A_{ij}, A_{k\ell} \in R_k \) for some \( k \) if and only if the pixels \( A_{ij} \) and \( A_{k\ell} \) are neighbors and similar. The extended region adjacency graph associated to \( A \) is a directed, labeled graph \((V, E, \alpha, \beta)\), where

- there is a vertex \( v_i \in V \) for each region \( R_i \);
- there is a directed arc \((v_i, v_j) \in E \) if \( \text{avg}(R_i) < \frac{\text{avg}(R_j)}{2} \), where \( \text{avg}(R) \) is the average pixel value in region \( R \);
- the label \( \alpha(v_i) \) of vertex \( v_i \in V \) is the size of region \( R_i \) relative to the size of the whole image; and
- the label \( \beta((v_i, v_j)) \) of arc \((v_i, v_j) \in E \) is the increment \( \text{avg}(R_j) - \text{avg}(R_i) \) in average pixel value from region \( R_i \) to region \( R_j \).

Notice that \( \beta \) could be less than \( \theta \), but in most cases it will be larger.

Notice that in Def. 3, if pixels \( A_{pq} \) and \( A_{rs} \) lie in different regions, they have to be dissimilar. This means that the extended region adjacency graph associated to a digital image \( A \) with pixel similarity threshold \( \theta \), is unique, what allows to compare two images by comparing their extended region adjacency graphs. Also, the points of dissimilarity intuitively capture the borders, where there are abrupt gray-level changes. On the other hand, if the changes in gray level are smooth, only one region will be generated. Formally:
Figure 1. Fragment of the digital image reproduced in Fig. 4, consisting of 8 rows and 8 columns of 8-bit pixel values.

**Lemma 1.** Let $G = (V, E, \alpha, \beta)$ be the extended region adjacency graph associated to a digital image $A$, with pixel similarity threshold $\theta$. Then $G$ is unique, up to isomorphism.

**Proof.** If a complete graph is constructed, where each pixel has an adjacency arc with all its neighbors and arcs joining similar pixels are colored red and dissimilar pixels blue, then by contracting all the red arcs, independently of the order of the process, the same region adjacency graph is obtained. \[\square\]

As pointed out in the previous Lemma, construction of a region adjacency graph associated to an image can be seen as an edge contraction process on a grid graph derived from the image, by which adjacent, similar pixels are collapsed [10]. The resulting region adjacency graph is simple by construction and it is planar, since grid graphs are planar and edge contraction preserves planarity [14]. Extended region adjacency graphs are not only planar, they are also acyclic.

**Lemma 2.** Let $G = (V, E, \alpha, \beta)$ be the extended region adjacency graph associated to a digital image $A$. Then $G$ is acyclic.

**Proof.** Suppose there is a cycle. If so, picking any vertex in the cycle, its gray level is smaller than the gray level of the next vertex in the cycle. By transitivity, the previous vertex in the cycle has a gray level larger than the initial vertex, which is a contradiction. \[\square\]

**Example 1.** The image fragment reproduced in Fig. 1 consists of 8 rows and 8 columns of pixel values, each representing one of 256 possible gray levels, from 0 (black) to 255 (white). An extended region adjacency graph can be extracted from that image at different values of pixel similarity threshold. The resulting regions are illustrated in Fig. 2 on top of the original grid graph of the image fragment, for different threshold values. Figure 3 shows the region adjacency graph for the case $\theta = 20$, where vertex labels have been multiplied by the size of the whole image for clarity.

The following algorithm is a simple, more efficient procedure for constructing extended region adjacency graphs.

**procedure** extract planar graph
find all regions
build region adjacency graph

Regions of similar pixel values are grown until their neighborhood consist of dissimilar pixels only.

**procedure** find all regions
mark all pixels as non-visited
for each non-visited pixel $i, j$ do
mark pixel $i, j$ as visited
assign pixel $i, j$ to a new region
process neighborhood of pixel $i, j$
Figure 2. Construction of an extended region adjacency graph. Regions of neighboring, similar pixels are shown on top of the grid graph corresponding to the image fragment in Fig. 1, for different pixel similarity threshold values.
end for
The neighborhood of a pixel is grown recursively, as long as neighboring pixels are similar.

procedure process neighborhood of pixel \(i, j\)
for each pixel \(k, \ell\) neighbor to pixel \(i, j\)
if pixel \(k, \ell\) is non-visited and similar to pixel \(i, j\) then
mark pixel \(k, \ell\) as visited
assign pixel \(k, \ell\) to current region
process neighborhood of pixel \(k, \ell\)
end if
end for

After all regions of similar, neighboring pixels have been grown, the directed graph of adjacent regions is built by creating a vertex for each region found, and collecting a directed arc between pixels in a same region that are neighbors in the image. Vertices are labeled by the relative size of the region they encompass, while arcs are labeled by the increment in average pixel value between the regions.

procedure build region adjacency graph
create a new graph \(g\)
for each region \(R\) do
add a new vertex to \(g\)
set relative size of region
set average pixel value encompassed by region
end for
for each row \(i\) do
for each column \(j\) do
let \(m\) be the region to which pixel \(i, j\) belongs
let \(n\) be the region to which pixel \(i + 1, j\) belongs
if \(m \neq n\) then
if there is no arc in \(g\) from \(m\) to \(n\) then
add an arc in \(g\) from vertex \(m\) to vertex \(n\)
set increment in average pixel value
else if there is no arc in \(g\) from \(n\) to \(m\) then
add an arc in \(g\) from vertex \(n\) to vertex \(m\)
set increment in average pixel value
end if
end if
let \(n\) be the region to which pixel \(i, j + 1\) belongs
if \(m \neq n\) then
if there is no arc in \(g\) from \(m\) to \(n\) then
add an arc in \(g\) from vertex \(m\) to vertex \(n\)
set increment in average pixel value
else if there is no arc in \(g\) from \(n\) to \(m\) then
add an arc in \(g\) from vertex \(n\) to vertex \(m\)
set increment in average pixel value
end if
end if
end for
end for

The algorithm runs in space quadratic in the number of pixels in the image, in order for the arc existence test to take constant time. Moreover, since every pixel is visited at most a constant number of times (its neighborhood size plus one), it also runs in time linear in the number of pixels in the image.

The alternative view of constructing an extended region adjacency graph by edge contraction can be realized by the following algorithm, where \(\text{avg}(R)\) denotes the average pixel value in region \(R\):

procedure extract planar graph
let \(G\) be the grid graph defined by the image \(A\)
remove all edges in \(G\) joining dissimilar pixels
let \(\{R_1, \ldots, R_n\}\) be the connected components of \(G\)
create a new graph \(g\)
for each \(R_i\) in \(G\) do
add a vertex \(v_i\) to \(g\)
label \(v_i\) with the size of \(R_i\) relative to the size of \(A\)
end for
for each \(R_i, R_j\) in \(R\) do
if \(R_i, R_j\) are adjacent then
if \(\text{avg}(R_i) < \text{avg}(R_j)\) then
add a directed arc \((v_i, v_j)\) to \(g\)
label \((v_i, v_j)\) with \(\text{avg}(R_j) - \text{avg}(R_i)\)
end if
if \(\text{avg}(R_j) < \text{avg}(R_i)\) then
add a directed arc \((v_j, v_i)\) to \(g\)
label \((v_j, v_i)\) with \(\text{avg}(R_i) - \text{avg}(R_j)\)
end if
end if
end for
end for

This alternative algorithm runs in time and space linear in the number of pixels in the image, provided that the connected components of the resulting graph are collected using an algorithm that takes time and space linear in the number of vertices and edges, such as those described in [19, 20].

4. Matching Extended Region Adjacency Graphs

Computing the distance between arbitrary graphs is an NP-complete problem, even for planar graphs [11]. However, the similarity between two images can be computed in polynomial time by approximating the distance between the associated extended region adjacency graphs.

The distance between two labeled graphs, as a generalization of edit distance between strings, sequences, and trees [1], is the least cost sequence of elementary graph edit operations that transform one graph into the other. The elementary edit operations between two graphs \(G_1\) and \(G_2\) are the following:

- deleting a vertex or an arc from \(G_1\);
- inserting a vertex or an arc into \(G_2\); and
• substituting a vertex or an arc from $G_1$ by a vertex or an arc from $G_2$.

**Definition 4.** An approximate graph matching from a graph $G_1 = (V_1, E_1, \alpha_1, \beta_1)$ to a graph $G_2 = (V_2, E_2, \alpha_2, \beta_2)$ is a bijective function $f : W_1 \rightarrow W_2$, where $W_1 \subseteq V_1$ and $W_2 \subseteq V_2$.

An approximate graph matching $f : W_1 \rightarrow W_2$ can be seen as a sequence of elementary edit operations that transform graph $G_1$ into graph $G_2$. All vertices in $V_1 \setminus W_1$ are deleted from $G_1$, all vertices in $W_1$ are substituted by vertices in $W_2$, all vertices in $W_2 \setminus W_1$ are inserted into $G_2$. Let $X_1 \subseteq E_1$ be defined as $X_1 = E_1 \cap (W_1 \times W_1)$, and let $X_2 \subseteq E_2$ be given by $X_2 = E_2 \cap (W_2 \times W_2)$. Then all arcs in $E_1 \setminus X_1$ are deleted from $G_1$, all arcs in $X_1$ are substituted by arcs in $X_2$, all arcs in $E_2 \setminus X_2$ are inserted into $G_2$.

The costs of these elementary graph edit operations are non-negative real numbers defined as follows:

• deletion of vertex $v \in V_1 \setminus W_1$ has cost $c(v \rightarrow \Lambda)$;
• insertion of vertex $v \in V_2 \setminus W_2$ has cost $c(\Lambda \rightarrow v)$;
• substitution of vertex $v \in W_1$ by vertex $u \in W_2$ has cost $c(v \rightarrow u)$;
• deletion of arc $e \in E_1 \setminus X_1$ has cost $c(e \rightarrow (\Lambda, \Lambda))$;
• insertion of arc $e \in E_2 \setminus X_2$ has cost $c((\Lambda, \Lambda) \rightarrow e)$; and
• substitution of arc $e \in X_1$ by arc $a \in X_2$ has cost $c(e \rightarrow a)$.

**Definition 5.** The cost of an approximate graph matching $f : W_1 \rightarrow W_2$ from a graph $G_1 = (V_1, E_1, \alpha_1, \beta_1)$ to a graph $G_2 = (V_2, E_2, \alpha_2, \beta_2)$ is given by

$$
\gamma(f) = \sum_{v \in V_1 \setminus W_1} c(v \rightarrow \Lambda) \\
+ \sum_{v \in V_2 \setminus W_2} c(\Lambda \rightarrow v) \\
+ \sum_{e \in E_1 \setminus X_1} c(e \rightarrow (\Lambda, \Lambda)) \\
+ \sum_{e \in E_2 \setminus X_2} c((\Lambda, \Lambda) \rightarrow e) \\
+ \sum_{(v, u) \in X_1} c((v, u) \rightarrow (f(v), f(u)))
$$

**Definition 6.** The edit distance between two graphs $G_1$ and $G_2$ is the (cost of the) least cost approximate graph matching from $G_1$ to $G_2$.

$$
\delta(G_1, G_2) = \min \{ \gamma(f) \mid f : G_1 \rightarrow G_2 \}.
$$

The edit distance between extended region adjacency graphs is the basis of the similarity measure between images. The edit distance can be approximated by computing a minimum cost matching in a weighted, complete bipartite graph built out of the extended region adjacency graphs.

**Definition 7.** Let $g_1 = (V_1, E_1, \alpha_1, \beta_1)$ and $g_2 = (V_2, E_2, \alpha_2, \beta_2)$ be extended region adjacency graphs. The weighted, complete bipartite graph $g = (V, U, E)$ associated to $g_1$ and $g_2$ is defined as follows:

• $V = E_1 \cup \{(\Lambda, \Lambda)\}$;
• $U = E_2 \cup \{(\Lambda, \Lambda)\}$;
• $E = V \times U$; and
• each arc $(e_i, e_j) \in E$ has weight $w_{ij}$, where

$$
w_{ij} = k[\beta_1(e_i) - \beta_2(e_j)] \\
+ (1 - k)[\alpha_1(v_p) - \alpha_2(u_r)] \\
+ (1 - k)[\alpha_1(v_q) - \alpha_2(u_s)]
$$

and $e_i = (v_p, u_r)$ and $e_j = (u_r, u_s)$. The label on the dummy vertices is $\alpha_1(\Lambda) = \alpha_2(\Lambda) = 1$, and the label on the dummy arc is $\beta_1((\Lambda, \Lambda)) = \beta_2((\Lambda, \Lambda)) = k - 1$, where $k$ is the number of possible gray levels in the images.

Notice that the values $\alpha_1(\Lambda) = \alpha_2(\Lambda) = 1$ represent the size of the largest possible region relative to the size of the
whole image, while the values $\beta_1((\Lambda, \Lambda)) = \beta_2((\Lambda, \Lambda)) = 255$ represent the largest possible increment in average pixel value between two regions.

These dummy vertices and arcs model vertex and arc deletion and insertion. The constant $k$ models the ratio of pixel similarity mismatch to region size mismatch in an optimal match between the two images.

A minimum cost matching in the complete bipartite graph $g$ associated to $g_1$ and $g_2$ approximates the edit distance between $g_1$ and $g_2$. Edges in the matching (not touching any dummy arc) correspond to arc substitutions. Edges in the matching touching the dummy arc in $V$ correspond to arc insertions. Edges in the matching touching the dummy arc in $U$ correspond to arc deletions.

Similar approximations of graph matching by weighted matching in bipartite graphs are found, for instance, in [21, 7], the former using a maximum flow formulation.

Now, a similarity measure between two images $A_1$ and $A_2$ can be defined in terms of a minimum cost matching $M$ in a complete bipartite graph $g$, associated to the extended region adjacency graphs $g_1$ and $g_2$ of the images.

**Definition 8.** Let $g_1$ and $g_2$ be extended region adjacency graphs associated to images $A_1$ and $A_2$, respectively. The similarity $\sigma$ between $A_1$ and $A_2$ is given by

$$\sigma(A_1, A_2) = \sum_{(e_i,e_j)\in M} w_{ij},$$

where $w_{ij}$ is the weight of arc $(e_i,e_j)$ in $M$, according to Def. 7.

Let $g_1$ and $g_2$ be the extended region adjacency graphs associated to the images $A_1$ and $A_2$, respectively. The following algorithm computes the similarity between $A_1$ and $A_2$ by finding a minimum cost matching in the complete bipartite graph $g$ associated to $g_1$ and $g_2$.

**procedure** compute similarity between images

let $g$ be the bipartite graph from $g_1$ and $g_2$

let $M$ be a minimum cost matching in $g$

set $\sigma(A_1, A_2)$ to zero

for each arc $(e_i,e_j)$ in $M$

add $w_{ij}$ to $\sigma(A_1, A_2)$

end for

Weighted bipartite matching is known to run in time $O(n(m + n \log n))$ for a graph with $n$ vertices and $m$ edges [13], and therefore in time $O(n^2 \log n)$ for planar graphs, because a planar graph with $n$ vertices cannot have more than $3n - 6$ edges [14].

Since weighted complete bipartite matching is the dominating factor in the complexity of the previous algorithm, and considering that the bipartite graph is complete, the overall time is $O(n^4 \log^2 n)$ time, where $n$ is the number of regions in the two extended region adjacency graphs. As the number of regions depends on the parameter $\theta$, there is a trade-off between the quality of the similarity measure and the computational complexity of its computation.

The number of regions depends on $\theta$. It can range from 1 to $T$, the total number of pixels in an image. As the probability of two adjacent pixels being similar only depends on $\theta$ and $T$ for random images, changing $\theta$ only changes the constant term in the number of regions, which will be of the form $N/f(\theta, T)$, where $N$ is the number of pixels in the image and $1 \leq f(\theta, T) \leq N$. Hence, for not too large $N$, the parameter $\theta$ can be set such that the number of regions is constant, logarithmic or sublinear in $N$, in order for $O(n^4 \log^2 n)$ to be at most $O(N)$. This can be seen in the next section, where for $\theta = 10$ only a few hundred regions are obtained for the images considered. On the other hand, for large images, the value for $\theta$ can be large.

Invariance to translation and rotation is achieved by matching region adjacency graphs, because of the graph representation itself. Moreover, invariance to scaling is obtained by matching extended region adjacency graphs. The same image at different scalings has associated the same extended region adjacency graph.

**5. Experimental Results**

The algorithm for extracting an extended region adjacency graph from an image, subject to a given threshold of pixel similarity, and the algorithm for computing the similarity between two images, given their extended region adjacency graphs, have been implemented in C++ using the LEDA library of efficient data structures and algorithms [13] and the Ldoc literate programming environment.

In order to assess the adequacy of the approach, a series of experiments have been carried out on a standard digitized image of the Mona Lisa famous painting by Leonardo da Vinci, shown in Fig. 4. The image consists of 360 rows and 250 columns of pixel values, each representing one of 256 possible gray levels, from 0 (black) to 255 (white).

Extended region adjacency graphs have been extracted from these images for different pixel similarity threshold values $\theta$, with $0 \leq \theta \leq 50$. The number of vertices of the resulting extended region adjacency graphs are given in Fig. 5 as a function of the pixel similarity threshold.

These extended region adjacency graphs turned out to be rather sparse. The density or ratio of the number of arcs to the number of vertices is given in Fig. 6 as a function of the pixel similarity threshold. The density of extended region adjacency graphs has been always less than 3 in all these experiments. As a matter of fact, extended region adjacency graphs are sparse because they are planar, and these graphs, as mentioned before, have a linear number of edges.

---

1 This probability is exactly $(2\theta + 1)/T - \theta(\theta + 1)/T^2.$
Figure 4. Digitized image of the Mona Lisa, taken from [10]. The whole image (top) consists of 360 rows and 250 columns of 8-bit pixel values, while the eyes image fragment (bottom, left) has 20 rows and 50 columns of pixel values, and the smile image fragment (bottom, right) has 16 rows and 32 columns of pixel values.

Figure 5. Number of vertices in extended region adjacency graphs, for different pixel similarity thresholds.
Figure 6. Ratio of number of arcs to number of vertices in extended region adjacency graphs, for different pixel similarity thresholds.

The maximum vertex degree in the resulting extended region adjacency graphs are also given in Fig. 7, again as a function of the pixel similarity threshold. Actually, relatively few vertices achieve the maximum vertex degree, and most vertices have degree not greater than 5, as shown in Fig. 8.

Furthermore, in order to assess the invariance of the similarity measure to rotation, another series of experiments have been carried out on the same digitized image of the Mona Lisa. The distance between the image and a rotated version of the image has been computed for different pixel similarity threshold values θ and for different ratios k of pixel similarity mismatch to region size mismatch. The image has been rotated 90 degrees using the proflip program by Jef Poskanzer.

The similarity between the image and a rotated image is shown in Fig. 9 as a function of the pixel similarity threshold, and in Fig. 10 as a function of the ratio of pixel similarity mismatch to region size mismatch.

These experiments confirm the hypothesis of a trade-off between the quality of the similarity measure and the computational complexity of its computation.

6. Conclusions

The problem of computing the similarity between two images is transformed to the problem of computing the distance between two extended region adjacency graphs, for which an approximation algorithm is given. The extended region adjacency graphs are extracted from the images in time and space linear in the number of pixels.

The method presented in this article exhibits the desired
Figure 8. Number of vertices realizing each vertex degree in extended region adjacency graphs, for different pixel similarity threshold values ranging from 0 to 10.

Figure 9. Similarity between the image and a rotated version of the image, for a fixed ratio of pixel similarity mismatch to region size mismatch of 0.5 and for different pixel similarity thresholds.

Figure 10. Similarity between the image and a rotated version of the image, for a fixed pixel similarity threshold of 50 and for different ratios of pixel similarity mismatch to region size mismatch.
properties of invariance to translation, rotation, and scaling. Another interesting property is that a trade-off is provided between pixel similarity threshold and approximation of the similarity measure, which can be used to bound the error in image recognition as well as the time complexity of the computation.

The method can also deal with color images, as each color is also coded as an integer. In this case $\theta$ is only used to separate different colors. Similarly, it can also deal with texture, as well as other attributes.

Current research consists of using the similarity measure presented in this article to find a subimage in an image. This includes adding variable scaling such that if the subimage is larger or smaller it can also be found. For this, the relative sizes between subimages instead of the absolute sizes are taken in consideration.

The new similarity measure is not perfect, but can be used as a fast filter. That is, regions of the image where the subimage cannot exist can be eliminated and a more expensive algorithm is used only in the areas where there are several potential candidates.

Acknowledgments

The authors would like to acknowledge with thanks the anonymous referees, whose comments and criticism have led to a substantial improvement of this paper.

References


