

# Graph Isomorphism applied to Fingerprint Matching

Regina de Cássia Nandi, André Luiz Pires Guedes  
Universidade Federal do Paraná - Departamento de Informática  
{regina, andre}@inf.ufpr.br

## 1. Introduction

This work presents a methodology to compare fingerprints using graphs to represent features named minutiae. The comparison methodology adopted in this work is flexible since it prevents some problems in the comparison process: minutiae not detected and geometric differences such as rotation, translation and scale of the images [JAI2002; NAN2006].

Each vertex represents one minutiae found in the fingerprint image and is labelled with its type and geometric position in the image. The edges represent a relation of neighborhood between these features. The matching is done through the computation of graph isomorphism [BOL1998].

The implemented algorithm returns true when the spanning tree of a graph GF, that represent the fragment fingerprint, is isomorphic to a subgraph of GC that represents the candidate fingerprint. The algorithm was programmed and tested using images downloaded from the Internet for which the minutiae had been found in a manual process. The tests had been done based on a combination of diferents kind of graphs, built from the minutiae of the fingerprint. In this work is also shown an improvement based on the size of edges to reduce the number of comparisons.

## 2. Methodology discussion

The fingerprint matching process could be done in 3 levels: i) level of fingerprint, ii) level of set of points and, iii) level of graphs.

The current work carries through the matching at level iii), that is better than level ii), since the graph with edges defined on the correctly way, could supply a more limited matching. However, we must be care choosing graphs, since a bad choice of edges can make a matching impossible.

We must notice that the comparison in level ii) and iii) can lead to a false "yes".

## 3. Algorithm

Let X represent the fragment fingerprint minutiae set and let S represent the candidate fingerprint minutiae set, the matching process is a search of a transformation T, such that  $T(X) \subseteq S$ .

To find a possible solution, graphs  $GF=(VF,EF)$  and  $GC=(VC,EC)$  are constructed, where  $VF=X$  and  $VC=S$ . The edge sets EF and EC represent a particular neighborhood of the minutiae. These graphs represent the fingerprint and are the base for the matching process.

From the graphs GF and GC, we intended to verify if GF is a subgraph of GC, proving that X is a subset of S and the minutiae that compose the two graphs represent the same fingerprint. Thus we have:

$$\text{If } S \in GF=(VF,EF) \subseteq GC=(VC,EC) \rightarrow \exists T \mid T(X) \subseteq S$$

To verify if GF is a subgraph of GC, validating the proposed methodology, graph isomorphism is applied. In the isomorphism algorithm to solve the matching problem in this work, beyond the topology of the graphs, the geometry is also analyzed in the comparison process.

The fingerprint matching happens in the follow sequence: i) choose an edge "a" of GF, ii) for each "b" edge of GC, iii) calculate a T transformation between "a" and "b" and iv) test T for the others vertices of GC and GF. The algorithm run sequence code can be seen on table 1.

The graph matching returns true when the spanning tree of GF is isomorph to a subgraph of GC. Therefore when defining GF – case the data represents the same fingerprint – it must be guaranteed that the edges of this one are in GC.

The algorithm complexity depends directly on the order and size of the graphs used to represent the fingerprints. For this reason, to improve the isomorphism performance, a study was realized to minimize the number of edges to be evaluated. So, we propose a cut in the number of edges in GC to be used when choosing a transformation T that is determined in accordance with the size of its edges.

Both approaches, with and without cuts, are quite good but with cuts the numbers of comparisons are minimized.

<ol style="list-style-type: none"> <li>1. isIsomorph(GF Graph, GC Graph)</li> <li>2. Choose a edge <math>a=ij \in EF</math> ..... (base edge)</li> <li>3. For each edge <math>b=kl \in EC</math></li> <li>4. <math>MT = \text{getTransformationMatrix}(T(a,b))</math></li> <li>5. Pré-associates: <math>MAP[j] \leftarrow k</math> e <math>MAP[j] \leftarrow l</math></li> <li>6. Test_remaining(MT,MAP,GF,GC)</li> <li>7. If test finish and is OK, so return MAP</li> </ol>	<ol style="list-style-type: none"> <li>10. Test_remaining(MT Array, MAP Array, GF Graph, GC Graph)</li> <li>11. For <math>i</math> in <math>2</math> a <math> VF </math></li> <li>12. <math>w \leftarrow \text{OrderF}(i)</math>;</li> <li>13. <math>z \leftarrow MAP[\text{FatherF}(w)]</math></li> <li>14. <math>ListVizF \leftarrow \text{getNeighborhood}(GF,z)</math></li> <li>15. For each neighbor vertice <math>y</math> not marked of <math>z</math> (<math>ListVizF</math>)</li> <li>16. <math>marriedW \leftarrow \text{Test_TransformationMatrix}(T,w,y)</math>;</li> <li>17. If <math>marriedW</math></li> <li>18. <math>MAP[w] \leftarrow y</math></li> <li>19. mark <math>y</math></li> </ol>
---	--

Table 1: Implemented algorithm

### 3. References

- [BOL1998] BOLLOBÁS, Béla. **Graduate Texts in Mathematics: Modern Graph Theory**. Springer, 1ª Ed., 1998.
- [JAI2002] JAIN, Anil K., PANKANTI, Sharath, PRABHAKAR, Salil. **On the individuality of fingerprints**. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, New York, v. 24, n. 8, p-1010-1025, 2002.
- [NAN2006] NANDI, Regina de Cássia Nandi. **Isomorfismo de Grafos aplicado à comparação de Impressões Digitais**. Universidade Federal do Paraná, 2006 (Dissertação de Mestrado em Informática)