

CAROL - New Algorithmic Tools for Comparing 2D Electrophoresis Gel Images

supported by Deutsche Forschungsgemeinschaft, grant FL 165/4-1

H. Alt, F. Hoffmann, K. Kriegel, C. Wenk (Freie Universität Berlin, Inst. f. Informatik)

K.-P. Pleißner (Deutsches Herzzentrum Berlin)

Task: Design a new tool for the computer assisted analysis of 2D electrophoresis gel images; especially, find new algorithmic solutions to the **local** and **global matching** problem between spot patterns using techniques from **computational geometry**.

The algorithm should solely use **geometric information** derived from gel images via a preprocessing with a **spot detection** algorithm; gel images are then simply given as lists of spots, each spot represented by its geometric **x-y-coordinates** and a real value that represents the spot **intensity**.

The algorithm should NOT rely on neither given landmarks nor an a priori image alignment.

The **CAROL system** should be eligible to match 2D gel images across the Internet.

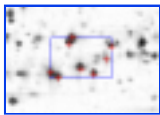
The Matching Problem for Spot Patterns:

Given gel images **source S** and **target T** in form of **spot lists** we want to solve the following tasks:

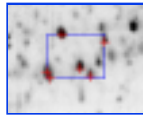
Local Matching:

Given in **S** a **local pattern P** of spots find all **local patterns P'** in **T** that "resemble" at least partially both the **geometric shape** and the **intensities** of **P**.

Spots in **P** (red):



Matched Spots in **P'** (red):



Global Matching:

Given both **S** and **T** represent **comparable samples**; compute an overall list of spot pairs that correspond to each other.

Local Matching:

Source: (HEART-2DPAGE)

Selected Pattern:

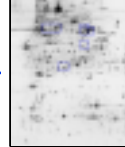
Target: (HP-2DPAGE)



Best Pattern:



Local Matchings:



Mathematical Model and Restrictions:

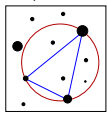
Assume we want to choose a pattern **P** from a small rectangular window in the source image **S**. Source and target image can have significantly different resolutions but intensive spots tend to appear first. Therefore it makes only sense to choose and restrict oneself to such patterns **P** that consist of the **locally most intensive spots**.

On the other side, the target pattern **P'** should also consist of locally intensive spots. Moreover, we should also accept solutions in which **P'** resembles **only a large portion** of **P**. This way we can also try to correct errors made by the spot detection algorithm.

To model the pure geometric resemblance between **P** and **P'** we make use of the following empirical fact. One can expect that if (s,t) is a line segment connecting spots in **P** and if there are corresponding spots s',t' in **P'** then both the absolute slope difference and the relative length of (s,t) and (s',t') can be bounded by rather small constants α and λ , respectively. We call a pair of edges (λ,α) -similar if they satisfy these conditions. Let a (λ,α) -matching between **P** and **P'** be a matching such that each corresponding pair of edges is (λ,α) -similar.

The Algorithmic Ideas:

A triangle of spots is **intensive**, if its enclosing disk does not contain a spot that is more intensive.



An edge is **intensive** if it is part of an intensive triangle.

Why to use intensive edges?

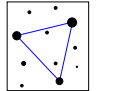
If there are **n** spots there is an expected number of only **6n** intensive edges compared to a total number of **n²** target edges.

Intensive edges can be computed efficiently by the **incremental Delaunay triangulation**.

If an intensive pattern **P** has an intensive matching partner pattern **P'** in the target, then edges of **P** will be (λ,α) -similar to **intensive edges** in **T**.

The Incremental Delaunay Triangulation:

1st Step:



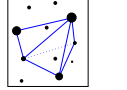
The 3 most intensive spots form an intensive triangle.

2nd Step:



Insert next intensive spot; one edge is deleted, three new ones inserted.

3rd Step:



5th spot inserted, two new edges, one new flipped diagonal.

Last Step:



Last spot inserted, 3 new triangulation edges, 2 flipped diagonals.

All triangulation edges in this construction are intensive!

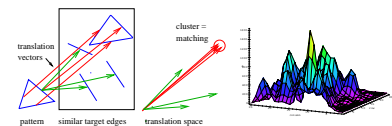
Let $Hist^*(T)$ denote a data structure of all intensive edges and all flipped diagonals; it contains **12n** elements.

Another Algorithmic Trick: 2-Step Geometric Hashing

Standard Approach: If e is a pattern edge and e' is a (λ,α) -similar edge in $Hist^*(T)$ one tries to extend this to a matching of the whole pattern (**alignment technique**).

Drawback: Many attempts will be useless.

Our solution: Instead of storing each vector we determine with a **scoring** procedure the accumulated number of hits sufficiently close to a grid node in the translation space. The actual matching is **locally** recomputed for the best scores.



The Implementation of the CAROL System:

The CAROL system is divided into two parts:

The algorithmical part (**server**)

The graphical user interface (**client**)

The communication between these parts is established via internet sockets. Both subprograms will usually run on different machines:

The **server** will run on the web server of the computer science institute of Freie Universität Berlin. It will wait for matching requests, perform the computation and send back the results.

The **client** will run on the user's machine, probably as an applet started out of an internet browser. The client will handle the user input, send matching requests to the server and display the results.

The Implementation of the Algorithm:

The combinatorial and geometrical kernel of the matching algorithm is implemented in C++.

It makes essential use of the **Standard Template Library (STL)**

Several data structures are based on the **Computational Geometry Algorithms Library (CGAL)**

The implementation of the incremental Delaunay triangulation also makes use of CGAL functions.

The Graphical User Interface:

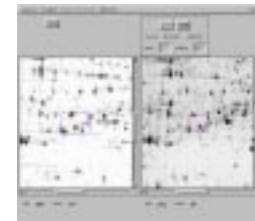
The graphical user interface is implemented in the **java** programming language. It can be run as an **applet** started out of an internet browser or as an **application**.

Due to the client-server-architecture of the CAROL system the graphical user interface (**client**) is only able to run, if the server program is already running.

After including a spot detection algorithm it will be possible to compare various gel images from all over the internet.

The applet can be started from the project's page at <http://www.inf.fu-berlin.de/gelmatching/index-eng.html>

Options and Features:



Select pattern region, pattern spots, search range, landmarks

Zoom picture, show detected spots, show matchings

Set matching parameters (λ, α , pattern size)

E-mail addresses :

hoffmann, kriegel, wenk @inf.fu-berlin.de; pleiss@ dhzb.de