Metric and Combinatorial Searching in Multimedia Data

Gonzalo Navarro

Center for Web Research

December 2002

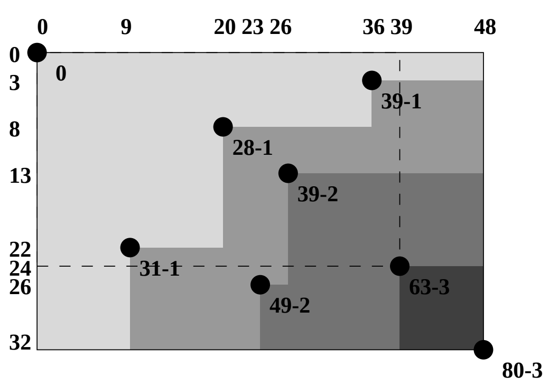
Searching Multimedia Data

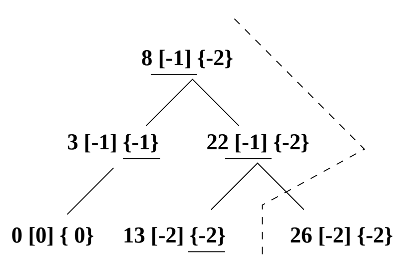
- Searching multimedia data is totally different from searching discrete data.
- Multimedia data represent continuous signals of the real world.
- Hence searching for equality is hopeless.
- We need a definition of *similarity* to asses how relevant is an answer to a query.
- Once such a definition is established, we need *similarity access methods* to quickly find the multimedia objects in a database that are most similar to a query.
- Two different approaches: *metric* and *combinatorial* search.
- The former is more general, the latter can be faster.

Approach	Computing similarity	Sequential searching	Indexed searching
Metric			X
Combinatorial	X	X	X

Searching Music

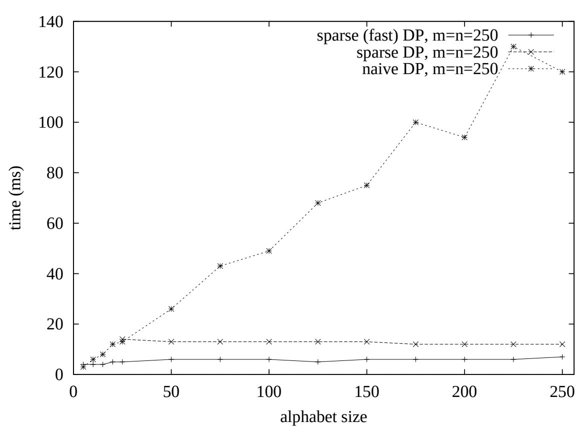
- Searching audio was traditionally approached from signal processing.
- Our aim is to try a combinatorial approach.
- This is especially relevant in the case of MIDI format.
- The algorithms should be resistant to changes of
  - Time warping
  - Transposition
  - Missing, extra or changed pieces
  - Volume and other distortions
- I have been working with Veli Mäkinen, Esko Ukkonen and Kjell Lemstrom, University of Helsinki.
- Transposition invariance:
  - We have obtained transposition invariant algorithms to compute different distances (LCS, Levenshtein, Hamming, SAD, ...).
  - The main techniques:
    - ∗ direct search for the optimum transposition
    - ∗ sparse dynamic programming
    - ∗ range searching

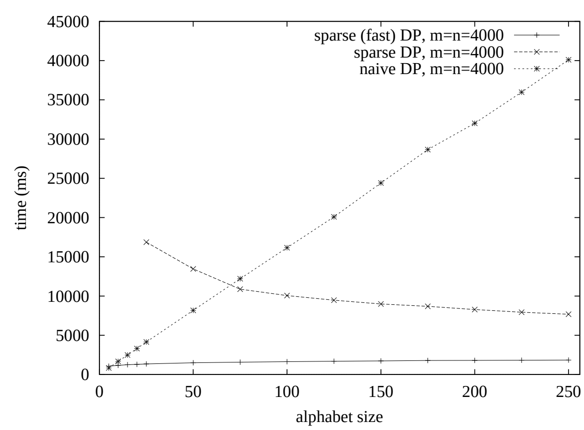




– This is much faster than the naive approach, both in theory and in practice:

distance	distance evaluation	searching
exact	$O(m)$	$O(m+n)$
$d_{\mathrm{H}}^{\mathrm{t},\delta}$	$O(m \log m)$	$O(mn \log m)$
$d_{\mathrm{SAD}}^{\mathrm{t},\kappa}$	$O(m + \kappa \log \kappa)$	$O((m + \kappa \log \kappa)n)$
$d_{\mathrm{MAD}}^{\mathrm{t},\kappa}$	$O(m + \kappa \log \kappa)$	$O((m + \kappa \log \kappa)n)$
(δ, γ)–matching	$O(m)$	$O(mn)$
$d_{\mathrm{ID}}^{\mathrm{t},\delta}$	$O(\delta mn \log \log m)$	$O(\delta mn \log \log m)$
$d_{\mathrm{ID}}^{\mathrm{t},\delta,\alpha}$	$O(\delta mn \log n)$	$O(\delta mn \log m)$
$d_{\mathrm{L}}^{\mathrm{t},\delta}$	$O(\delta mn \log \log n)$	$O(\delta mn \log \log m)$
$d_{\mathrm{L}}^{\mathrm{t},\delta,\alpha}$	$O(\delta mn \log^2 n \log \log m)$	$O(\delta mn \log^2 m \log \log m)$
$d_{\mathrm{D}}^{\mathrm{t},\delta,\alpha}$	$O(\delta mn)$	$O(\delta mn)$





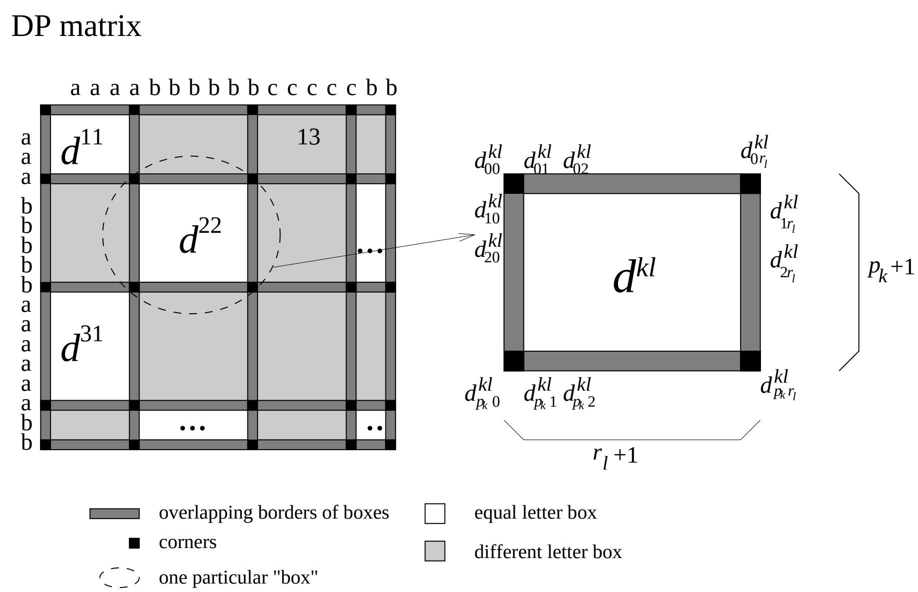
– Results will appear in STACS’03 and the extended version has been sent to the Journal of Algorithms.

– With Kjell we are working on a bit-parallel algorithm.

– Veli will probably spend March 2003 in Chile to continue this work.

– Immediate goals: extend some distances to images (lighting invariance, next part of the talk), and work on time warping.

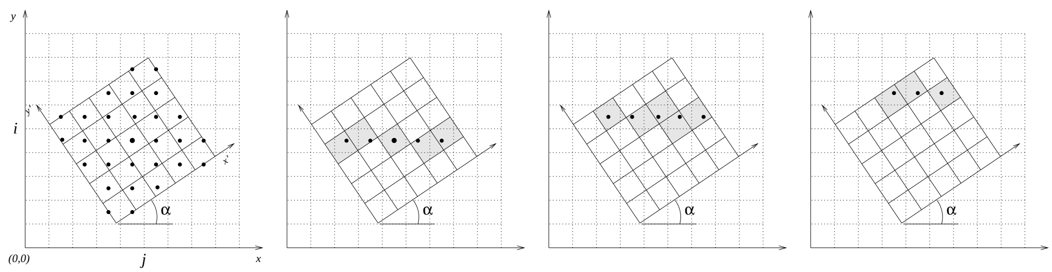
- Run-length compressed representations
  - We developed algorithms to compute LCS and Levenshtein distances over run-length compressed sequences.
  - Main idea: work only on the borders/corners of the “boxes”.



  - Time is $O(m'n')$ (LCS, conjectured) and $O(m'n + mn')$ (Levenshtein).
  - These can be useful for audio and image data.
  - They are also useful to deal with time warping.
  - We published this in CPM’02 and in Algorithmica.

Searching Images

- The problem: find a small subimage inside a larger image.
- Traditionally, similarity has been based on FFT.
- Recently, a combinatorial approach has emerged.
- This approach is less flexible but computes the similarity much faster.
- Our aim is to push on this flexibility/speed tradeoff.
- Ideally one would like to be resistant to
  - pixel differences,
  - rotations,
  - stretching and morphing,
  - scaling,
  - lighting.
- I have worked with Kimmo Fredriksson and Esko Ukkonen, Univ. of Helsinki.
- We have focused on permitting rotations and pixel differences.
- Sequential searching:
  - We have obtained optimal algorithms, $O(n^2 \log_\sigma m/m^2)$, if no pixel differences are permitted
  - We have obtained fast algorithms under several models of pixel differences (Hamming, SAD...)
  - For example, $O(n^2 k \log_\sigma m/m^2)$ for Hamming distance and $O(n^2(k/\sigma) \log_\sigma m/m^2$ for SAD.
  - Main techniques
    - Multipattern search of linear features
    - Incremental verification
    - Reduction to exact searching

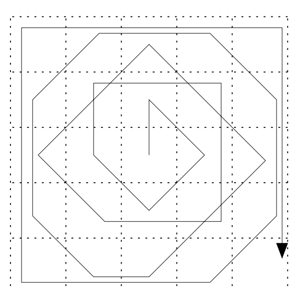


- We have published this in CPM’02.
- We plan to work on lighting invariance at first. Combination with existing work to permit some morphing (but not rotations) is in our plans.

- Indexed searching
  - The goal is to index a collection of images to permit fast searching.
  - Main idea is to use suffix trees that read the images at every possible angle.

22	16	10	17	23
15	5	1	6	18
9	4	0	2	11
14	8	3	7	19
21	13	12	20	24

(a)



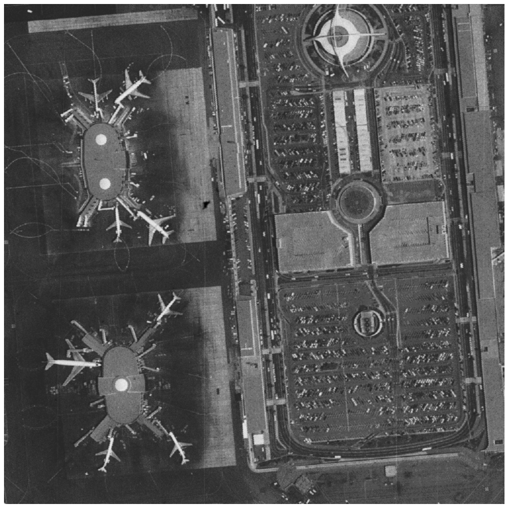
(b)

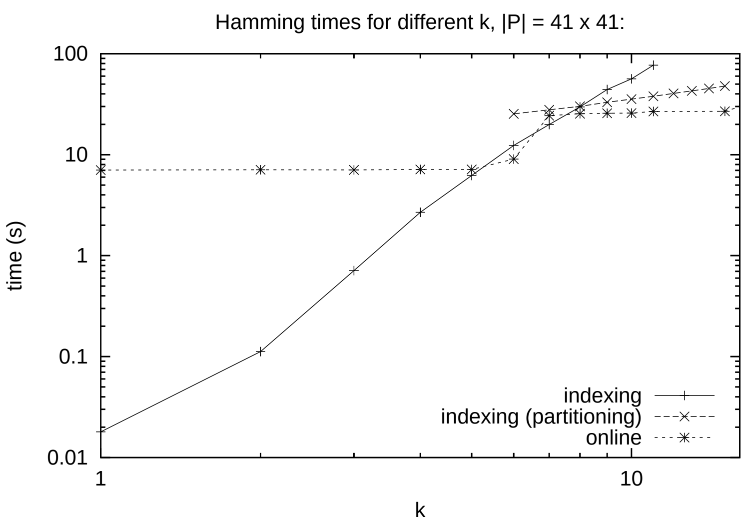
2	56	12	1	3
1	7	2	5	9
3	6	3	19	1
31	28	2	5	23
2	13	12	22	4

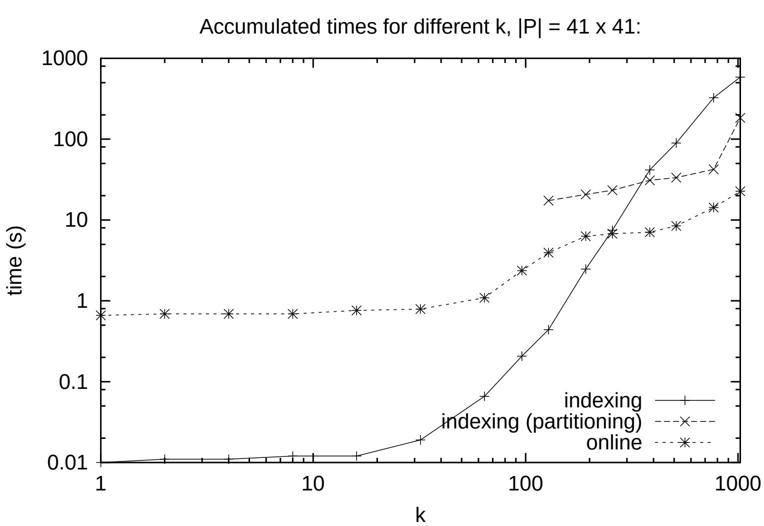
(c)

  - Good complexities are obtained, e.g. $O(\log_\sigma n)^{5/2})$ for exact searching
    and subquadratic for Hamming and SAD.
  - This has been extended to 3-dimensional searching.
  - This work predates the Millenium project, but we plan to integrate it with sequential search in a journal publication.

- Some real results:

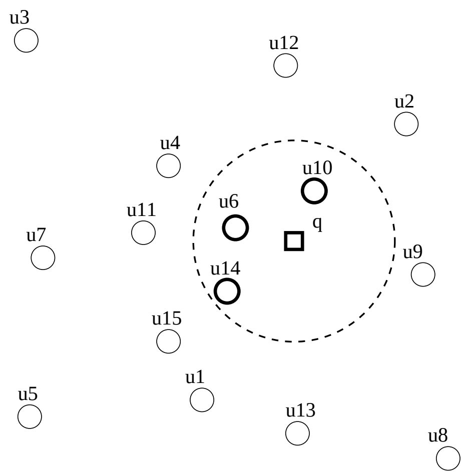




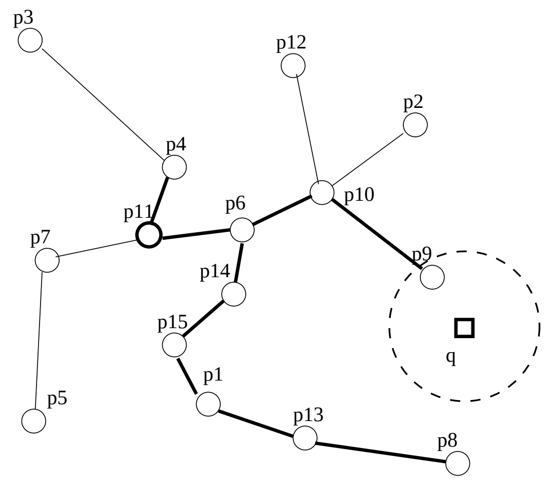


Metric Space Searching

- Computing the distance is only part of the problem
- We cannot compare the query against every database object!
- We seek to structure the database so as to quickly find the relevant objects.
- General methods are available via the metric space model.
- Metric spaces
  - A set of objects and a distance function that satisfies the triangle inequality
  - Many similarity searches can be modeled as a metric space search
  - General algorithms exist for metric space searching
  - However, the area is still immature and requires a lot of development
  - Apart from multimedia searching, this can be used for other Web related problems: lossy signal compression to transmit images by the network, document similarity to retrieve relevant Web pages, approximate text searching, etc.
- We have worked on several areas:
  - Dynamic data structures
  - Probabilistic and approximate methods
  - New approaches

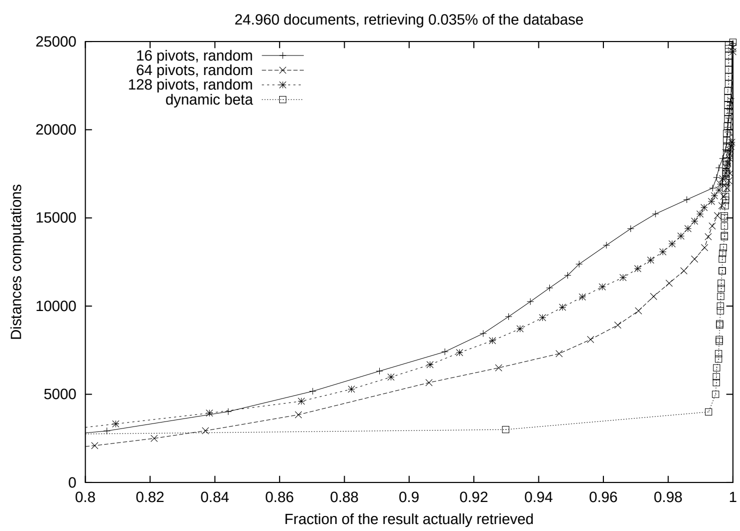

u3
u12
u2
u4
u10
u6
q
u11
u7
u9
u14
u15
u5
u1
u13
u8

- Dynamic data structures
  - We focused on the Spatial Approximation Tree



  - This data structure is competitive but markedly static
  - We developed algorithms for inserting and deleting elements
  - Main ideas:
    - ∗ limit reconstruction sizes,
    - ∗ use of timestamps,
    - ∗ use of empty nodes.
  - Construction times have in fact decreased!
  - Search times stay the same or improve
  - The cost of deletions is reasonable
  - Only now the structure is suitable for multimedia databases
  - We plan to develop a secondary memory data structure and to integrate it to a relational database.
  - We have also developed versions that become faster as more memory is given to them.
  - Work with my students: Nora Reyes (Univ. Nac. de San Luis, Argentina, MSc. and soon PhD. student) and Francisca Muñoz (U. de Chile, undergraduate).
  - Publications in SPIRE'02, CACIC'02, and others to come. In particular we plan to submit to SPIRE'03 and The VLDB Journal.
  - I hope that this structure can be useful to Javier?

- Probabilistic and approximate algorithms
  - In several applications, computing an approximate answer is as good as the “exact” answer.
  - Especially if it comes at a much lower cost!
  - In hard metric spaces, this is the only option
  - We have developed probabilistic versions of existing algorithms.
  - Main idea: traverse the set in a heuristically promising order, so the best elements are seen soon.
  - The results are encouraging:



  - Can we give an alternative to traditional IR methods?
  - Work with Edgar Chávez (U. Michoacana, Mexico) and Benjamin Bustos (U. de Chile, MSc. student).
  - Published in IPL, SPIRE’02, and soon in JDA.
  - We plan to keep working on a new line called PAC (Probably Approximately Correct) with Marco Patella and his group (Univ. di Bologna, Italy).
  - Maybe Karina (PhD student funded with the project) will work on this.

- New approaches: Graph-based methods
  - Precompute and store a subset of the distances of the database.
  - Regard that subset as a graph.
  - Depending on the properties of the graph we can bound the other distances.
  - We can also use the graph to navigate and approach the query.
  - For example we have worked on t-spanners.
  - These are memory-intensive techniques for hard spaces (typically not many objects with a very expensive distance function).
  - We will also consider neighbor graphs, an extension of the spatial approximation tree, and others.
  - MSc. thesis and ongoing PhD. thesis of Rodrigo Paredes, funded with the project.
  - We have already publications in SPIRE'02 and ALENEX'02.
  - Rodrigo gave more details this morning.

Handling Semistructured Information

- The structure of the text is a rich source of information that has not been fully exploited.
- It could be a useful tool to enhance the precision of Web queries.
- HTML has become a de facto standard to exchange structured documents.
- XML is the best candidate to emerge as the new standard for exchanging structured documents, replacing HTML.
- Query and manipulation languages to handle structured text are a current active research area.
- Several query languages for XML exist, but no efficient implementation.
- Only with an efficient implementation can this become a real choice for Web search engines.
- We have developed in the past indexing mechanisms for structured text.
- I guided an undergrad thesis (Manuel Ortega) this year that answers XPath queries over XML collections, using such an index.
- The software has been shown to be competitive against available alternatives.
- A demo can be seen at www.dcc.uchile.cl/ixpn.