PhD Thesis Proposal

Searching in Compressed Image Databases

PhD Student: Violeta N. Chang Camacho (`vchang@dcc.uchile.cl`)
Advisors: Prof. Gonzalo Navarro (`gnavarro@dcc.uchile.cl`)
Prof. Benjamin Bustos (`bebustos@dcc.uchile.cl`)

Dept. of Computer Science, University of Chile,
Blanco Encalada 2120, Santiago, Chile.

Abstract

Content-based image retrieval consists in retrieving from an image database the most similar image with respect to a query, according to some similarity measure. This scenario has numerous specific applications that include bioinformatics and medical imaging, among others. However, because the size of image repositories grows very fast, finding patterns in images requires an index to avoid a sequential scan, thus reducing the search time. Considering that small (and high) levels are faster than larger ones in the memory hierarchy, it would be very beneficial to have a compressed representation of the image database that can give fast access and pattern matching functionalities in high memory levels.

In this thesis we propose a deep study of searching in compressed image databases, contributing with new algorithms for approximate indexed search based on template matching. Our specific objectives are to determine the feasibility of extending existing approaches for indexed exact template matching to approximate template matching working on compressed image databases, in particular allowing *don't care* pixels; to design a similarity measure based on information theory to be used for approximate template matching; and to evaluate the impact on the performance and quality of results, in feature vector based similarity search, when the index is compressed in lossy form.

1 Introduction

The advancement of multimedia services has emphasized the importance of tools for visual information retrieval. Recently, searching for image information on multimedia databases have drawn tremendous attention, and there have been many research works on image retrieval methods. In essence, image retrieval consists in retrieving from an image database the most similar ones with respect to a query image, according to some specific criteria that determine similarity. Image retrieval has numerous specific applications that include fingerprint recognition, optical character recognition, and medical imaging.

There are two approaches for searching an image in a database: *exact* or *approximate*. Each of them has different characteristics of flexibility and speed. The exact search looks for an exact match between the given query and the retrieved images. In real-world applications, exact search results are not very useful: despite an image contains occurrences of the objects in the query image, they may not match exactly due to slight variations in orientation, scale, and lighting. Approximate search allows us to associate the query image with a similar one from the image database, according to a tolerance level specified by the user. It can be flexible to variations in orientation, scale, and lighting, among others, and still provide similar enough results for a wide range of applications. Yet, the available algorithms for approximate search are slow in practice due to a normalization process in both the images in the database and the query that would allow us to have different types of invariance.

Up to now, there are two major directions for image retrieval research. One direction is a technique that involves human-supplied text annotations describing image semantics. These text annotations are then used as the basis for searching. However, they have been shown to be inadequate for searching images, because the less expressive richness of the text about the visual characteristics cannot exploit the full abilities of the human memory. Moreover, different people may input different textual annotations for the same image.

The other direction involves content-based image retrieval. Such methods, that are classified as *template matching* approaches, look for occurrences of a pattern (template). That is, the problem is that given an image that contains the pattern to search for, one must return all the regions in the image containing the pattern. Alternativelly, techniques that are collectively referred to as *feature descriptor* approaches characterize what is really observed in an image by exploiting the color, intensity, or luminance information in an image. Yet, it is difficult to determine which descriptors should be used, and what should be the representation of a particular image database in the form of descriptors.

There is no clear consensus among researchers about which technique is the best for implementing a general image retrieval system. The answer to this problem depends on many factors, such as the number and complexity of the objects present in an image, and the amount of a priori information about the scene. However, template matching techniques, as now available, have proven to be a very useful tool for content-based image retrieval. In this thesis we propose a deep study of searching in compressed image databases, contributing with new algorithms for approximate indexed searching based on template matching.

2 Template Matching

Template matching consists in, given a reference image of an object (template), deciding whether that object appears in an image, according to some similarity measure, and finding its location if it does [CK02]. Formally, let us have an image I and a pattern template P, which are two-dimensional arrays of size $n \times n$ and $m \times m$, respectively. The template matching problem is to find a transformation of pixel coordinates of P, bringing P to a location on top of I that gives a *good enough* match for their corresponding superimposed intensities, or in a more general case, to evaluate the *distance* between P and I in every location [Fre01].

Template matching is an important issue in image retrieval with many applications including remote sensing [Bro92], medical imaging [FNU02], automatic inspection in industry [LC08], optical character recognition [Bro92], fingerprint identification [FNU02], signature verification [Bro92], and also can be an important low level building block of a high level vision system [Fre01, AV01]. In addition to the applications already mentioned, template matching is a very useful tool in Bioinformatics, because tasks such as gel electrophoresis [SAO$^+$98, KSR02], detection and classification of microscopic entities [ZCP04] and DNA microarray images analysis [GFP06], among others, can be greatly enhanced by template matching methods.

2.1 Supported Invariance Types

Rotation invariance is one of the desirable features in template matching and a basis for many applications, such as medical imaging and computation of visual motion. The traditional approach to the problem is to compute the cross-relation between each image location and each rotation of the pattern template [FNU02]. This can be done reasonably fast using the Fast Fourier Transform (FFT), requiring time $O(Kn^2 \log n)$ where K is the number of rotations sampled. Typically, K is $O(m)$ which makes the FFT approach very slow in practice [FNU05]. Rotation invariant moment-based methods require too much computing power for practical purposes and are sensitive to noise [LP96, FS94]. Various approaches have been used for achieving rotation invariance, such as generalized Hough transform, geometric hashing, graph matching, geometric moment-based matching, Zernike moments, wavelet descomposition, and combinatorial template matching.

Orientation codes have been used for *rotation invariant* template matching [UK04]. Histograms of orientation codes have been used for *lighting and rotation invariant* template matching [UKI01]. Using another approach, Fredriksson et al. [FMN07] address this problem considering approximate matching under several tolerance models.

Tanaka et al. [TSOO00] proposed a *rotation and scale invariant* template matching method. However, the matching scheme becomes impractical when both arbitrary rotations and scale changes are present in the template.

There are some proposed methods to achieve multi-invariance properties. Lin et al. [LC08] proposed a method for template matching, invariant under *translation, rotation, and scaling*. Although this method could be fast, precision is not good enough for real applications. A FFT-based matching method, which is an extension to phase-correlation technique, is presented by Reddy and Chatterji [RC96]. Their algorithm is characterized by being a *translation, rotation, scaling, and noise invariant* template matching method.

In summary, the template matching research field faces great challenges to design a method that is computationally efficient and highly accurate, in addition to provide a degree of tolerance for real applications with respect to variations in rotation, scale, lighting, and translation of both the image and the template.

2.2 Similarity Measures

The accuracy of a template matching process depends on the fitness of the similarity measure that is used to determine the similarity between a template and an image. Although different similarity measures have been developed, there is not a single similarity measure that is known to produce the best result in all situations. Depending on the types of images provided, one similarity measure may work better than another in template matching.

The most common similarity measures used in template matching are *Sum of absolute intensity differences* [DK82], *Cross-correlation coefficient* [Anu70, Pra74] and *Mutual information* [MCV+97, Vaj89]. It has been shown that cross-correlation coefficient produced more accurate matches than the sum of absolute differences when the images had no rotational differences [SMA78], and also that sum of absolute differences produced more accurate matches than cross-correlation coefficient when the images had some rotational differences [PWL+98]. Compared to cross-correlation coefficient and sum of absolute differences, mutual information is more sensitive to image noise. It has been found that slight image smoothing, which reduces image noise, improves the template matching accuracy of mutual information [DGS01], and that mutual information did not perform as well as the cross-correlation coefficient or the sum of absolute differences in matching of same modality images [PWL+98].

2.3 Template Matching with *Don't Cares*

It is common that, in practice, images may differ in a number of ways including being rotated, scaled, or affected by noise. Also, it should be considered the case where the intensity or brightness of an image ocurrence is unknown and where parts of either image contain *don't care* or *wild card* pixels, i.e., pixels that are considered to be irrelevant as far as template matching is concerned [CC07].

One general problem for this kind of template matching algorithms is that the template may contain a different background compared to the ocurrence of the template in the image, or some parts of the template or its counterpart in the image may be occluded by some other objects. One possible way to solve this consists in specifying which parts of the template are actually relevant and which are background, so, we could requiere that only some pieces match [FNU05].

3 Indexed Searching

Searching for all the ocurrences of pattern P in image database D can be carried out in two forms:

- **Sequential Searching**: we search for the pattern P directly on the plain representation of images of D. That is, we do not construct any data structure on the images, mainly because the image collection is small, or it is not available in advance.

- **Indexed Searching**: we build a data structure (or index) on the image database to restrict the search to a small portion of the collection, improving search time but increasing the space requirement to solve the problem. This approach is used when the image collection is so large that a sequential scan is prohibitively costly, many searches (using different patterns) must be performed on the same images, and there is sufficient storage space to maintain the index and provide efficient access to it.

The last consideration about indexed searching is a necessary condition to consider indexing at all. At first sight, the storage issue might not seem significant given the common availability of massive storage. The last decades have witnessed an exponential growth rate of the data available in digital form, as well as a parallel growth of the storage capacities available for moderate prices. The real problem, however, is efficient access. CPU speeds have been doubling every 18 months, while disk access times have stayed basically unchanged, and larger and faster cache memories have appeared, which are smaller, faster, and more expensive than the flat RAMs, and there have appeared even several layers of caches. This multi-layered memory architecture with sharp efficiency differences from one level to the next needs to be considered in indexed searching research.

For many years, there has been a lot of research on indexed search, focused on text search. Modern text databases have to provide fast access to the text using as little space as possible. However, these goals are opposite, because to provide fast access an index has to be built on the text, increasing the space requirement. As an obvious consequence, there has been much research on compressed text databases, focusing on techniques to represent the text and the index using little space, yet permitting efficient text searching.

With the huge texts available nowadays (for example, the human genome consists of about 3×10^9 base pairs, where each base pair needs 2 bits), one solution is to store the indexes on secondary memory. However, this has a significant influence on the running time of an application, as an access to secondary memory is considerably slower than an access to main memory. An alternative is to use compressed text indexes, which increases processing time. However, given the relation between main and secondary memory access times, it is preferable to handle compressed indexes entirely in main memory, rather than handling them in uncompressed form but in secondary storage.

In the area of text indexing, a new branch of compressed indexing has emerged in recent years: Given a sequence of characters, it is possible to obtain a compressed representation of the sequence that operates also as an index [FM05, GGV03, NM07, Sad03]. The indexing features of these structures can range from random access to elements of the original sequence to fast pattern matching capabilities. These compressed representations are often called *self-indexes*, as they replace the original data and can, thus, be considered as compression methods with some added value.

The combinatorial approach to searching for patterns on image databases relies on many of the foundations of text searching. There is already an extensive development in searching for patterns in two-dimensional sequences [GG95, Gia95]. At first, search algorithms were on-line, and as in the case of unidimensional sequences, the indexed search was studied later [FNU05, FNU02]. In the latter scenario, the images of the database are known in advance and can be preprocessed to speed up searches later.

In the case of two-dimensional data (image) collections, the memory usage is not a trivial matter even for medium-size image collections. The random access to pixel values is very significant because of the large difference in the cost to access to different levels of memory. For example, a very large image can be stored in compressed form in a large but slow memory level, and to access the values of its pixels, it should be uncompressed and copied to a faster memory level. To search efficiently in a collection of data (in this case, images) of regular size, it is necessary to use some kind of index. However, if the index involves spending even more memory, it would be stored at a slower memory level and the search time could be considerably affected.

Recently, two data structures have emerged to build self-indexes for a collection of images, that is, compressed representations of images that provide fast access with support for exact search [MN08]. Thus, considering the huge size of image databases, having the entire database in main memory would not be feasible, but it could be stored in a compressed form that allows direct access and accurate pattern search.

3.1 Self-indexing for filtering image searching

This technique uses a classical text index for a string $T[1,n]$ as a filter for the two-dimensional case of images. The suffix array $A[1,n]$ of T is an array of pointers to all the suffixes of T in lexicographic order [MM93]. Given A and T, the occurrences of a pattern $P = p_1p_2 \ldots p_m$ can be counted in $O(m \log n)$ time. The occurrences form an interval $A[sp, ep]$ such that suffixes $t_{A[i]}t_{A[i]+1} \ldots t_n$, for all $sp \leq i \leq ep$, contain the pattern P as a prefix. Self-indexes [NM07] replace the suffix array (and also the text) with a structure using $O(n \log \sigma)$ bits instead of the $O(n \log n)$ bits required by the suffix array, where σ represents the number of different gray intensities or different colors (commonly referred as the *alphabet*).

It is possible to apply the structure described above to an image I by concatenating all rows of I into a sequence of length $N = n^2$. To search for an $m \times m$ pattern P, we search for each of its rows independently using the self-index of concatenated I [MN08].

3.2 Self-index based on 2D Suffix Arrays

This self-indexing technique is built on a well-known technique to index images based on so-called L-suffixes [GG97, KKP03]. The idea is that each position $I[i, j]$ of an image defines a two-dimensional suffix. By building a suffix tree over those L-suffixes, it is possible to find all the positions where any $m \times m$ pattern occurs in time $O(m^2)$. It is not hard to derive a suffix array technique from the suffix tree. The suffix array just points to all the (i, j) image positions, in lexicographical order of the corresponding L-suffixes.

It is also possible to index a collection of images using just one suffix array, so that each suffix array position points to some pixel of some image [MN08]. In this case, if N is the overall size of the collection (in pixels), the search cost is $O(m^2 \log N)$. The suffix array and extra structures occupy $O(N \log N)$ bits of space. This is too high compared to the $N \log \sigma$ bits to store the plain representation of images, which must also be maintained to permit searching. In order to reduce space, Mäkinen and Navarro [MN08] use the concepts developed in Sadakane's Compressed Suffix Array (CSA) [Sad03].

4 Similarity Search based on Feature Vectors

When using a feature vector approach in a content-based image retrieval system, the image objects are often represented as d-dimensional feature vectors. One accesses the images using the feature vectors and an appropiate similarity measure. The usually high dimensionality of the feature vectors leads to high computational complexity in distance calculation for similarity retrieval, and inefficiency in indexing and searching [CW06]. An important difference between multimedia indexing structures and the general indexing structures is that the former is confronted with the influence of the *curse of dimensionality* [LLYY05]. In recent years, several methods have been proposed to speed up similarity queries in high-dimensional image databases [CZPC02, Lu02]: dimensionality reduction, multidimensional indexing, and filter-based approaches. A retrieval algorithm may include a combination of two or more of the mentioned methods. For example, one promising approach is to first perform dimensionality reduction and then use appropiate multidimensional indexing techniques [CW06].

Dimensionality reduction is one of the most direct methods to deal with the curse of dimensionality. This approach first condenses most of information in a dataset to a few dimensions by applying singular value descomposition (SVD) [HSE$^+$95] or Hilbert curve filling [CC98], among others. However, these methods have their own drawbacks, because they do not preserve the distances in the original space and significant errors may result when lower-dimensional transformed feature vectors are used to approximate the original feature vectors. In addition, traditional dimensionality reduction techniques have a high computational complexity.

Multidimensional indexing methods treat d-dimensional feature vectors as points in a d-dimensional vector space, and the similarity measure can be viewed as a measure of distance within that space. Furthermore, these methods work by partitioning the data space, clustering data according to partitioons, and using the partitions to prune the search space for querying. The existing multidimensional indexing techniques include the KD-tree, KDB-tree, R-tree (and its variants R$^+$-tree and R*-tree), and X-tree.

The multidimensional indexing approach faces a challenge when accessing an image database: the performance of existing multidimensional indexing schemes degrades dramatically as the dimensionality increases [WLW00, WSB98].

The filter-based approach searches the nearest k neighbors of a query image by filtering the vectors, so that only a small amount of them must be visited [CZPC02]. Clustering, classification, and latent semantic analysis are commonly used filter methods [LLYY05]. Also, the triangle inequality is another efficient filter approach [Lu02], when the distance measure is a metric. However, there are important drawbacks because the design of an approximation is not a trivial work, the precision of the approximation is not good while applying to image data of good locality, and additional information should be added to enhance the filtering rate when the database is getting larger [CW06].

5 Thesis Proposal

In the following subsections, for each part of the thesis, we give the problem statement, and then we define the objectives, and the main aspects of the proposed research.

5.1 Approximate template matching in image databases

Problem

What will be the performace of an indexed template matching using image self-indexes, with some degree of tolerance k, compared to that of a sequential search? What will be the impact of reducing image precision? What will be the impact of allowing *don't care* pixels?

Justification

There are only a few research works on approximate template matching with (a combination of) different degrees of tolerance, whose impact would be much larger than that of the exact approach in real-world applications. Also, there have been proposed self-indexes for images, which support only exact template matching, that could be extended to approximate template matching.

Objectives

- To determine the feasibility of flexible search in images, while maintaining reasonable efficiency according to a certain degree of tolerance in real applications and allowing *don't care* pixels.
- To determine whether the exact search on two-dimensional data self-indexes can be extended to approximate search using time and space efficiently.
- To determine whether the reduction of image precision has a favorable impact on approximate search performance.

Aspects to consider in the research

- To design and implement algorithms for template matching in images considering different models of tolerance in the matching, such as invariance to rotation, to scale, to brightness, to color and to deformations, and combinations thereof.
- To design and implement algorithms for approximate template matching in images allowing *don't care* pixels.
- To evaluate the performance of the implemented algorithms in terms of required resources (time and space) and retrieval quality:

- To compare combinatorial pattern matching techniques with classical techniques.
- To consider sequential and indexed search, focusing on indexed search.
- To consider classic and compressed indexes, with a special focus on the compressed ones, and on representations that encompass both image and index.

5.2 Template Matching Similarity Measures

Problem

What will be the impact on the retrieval quality of template matching when using a similarity measure based on information theory with a compression approach? Will it be practical to use an information theory based similarity measure for approximate template matching?

Justification

In the case of text compression, if two very similar texts are concatenated together and compressed, the required space for the compressed text is very close to the space needed for one of those texts in compressed form. This idea has not been used in any similarity measure for template matching.

Objectives

- To determine the effectiveness of template matching results using a similarity measure based on information theory involving a compression approach.
- To determine the feasibility of combining the similarity measure based on information theory with template matching techniques for searching on images with some degree of tolerance.

Aspects to consider in the research

- To design a similarity measure based on information theory with compression approach to be used for template matching.
- To compare the designed similarity measure to other similarity measures that are suitable for template matching.
- To evaluate the performance of the implemented algorithms for aproximate template matching using the designed similarity measure based on information theory, in terms of resource usage (time and space) and achieved retrieval quality.

5.3 Compressed indexes for feature vector based similarity search

Problem

Will feature vector index compression have favorable impact on required resources and on result precision?

Justification

Many of currently available multidimensional indexing techniques fail to handle high-dimensional data efficiently. For example, the R*-tree works well up to 10 dimensions, but its performance degrades quickly with incresing dimension and data sizes. In addition, there have been proposed some text index compression techniques with very good retrieval results that could be extended to work with two-dimensional data. This approach could be a solution to make the content-based image retrieval approach truly scalable to large image repositories. Furthermore, some recent studies aim to reduce the precision of inverted indexes for searching documents. These theoretical and experimental studies show that this technique saves storage space while the quality of responses is maintained (and improved, in some cases).

Objective

- To determine the impact of the compression of the index in the required resources and precision of image retrieval results using feature vectors.

Aspects to consider in the research

- To study feature vector based similarity models and associated indexes.
- To study retrieval algorithms for the reviewed models, especially considering the impact of index compression on index performance, both in efficiency and in retrieval quality.

6 Deliverables

- Our main contributions to scientific community are expected to be:
  - Algorithms for approximate template matching in compressed image databases.
  - A new similarity measure based on information theory for template matching.
  - Empirical evaluation of impact in quality results of index compression using feature vectors for similarity search.
- We plan to implement a prototype of our algorithms and to make them freely available to the scientific community.
- We hope to publish our main results in at least two high-level international conferences.

7 Conclusions

The type of search in images with more realistic applications is approximate seaching. However, because the size of image repositories grows very fast, finding patterns in images requires an index to avoid a sequential scan, thus reducing the search time. Considering the memory hierarchy, it would be very beneficial to have a compressed representation of the image database that can give fast access and pattern matching functionalities in high memory levels. Thus, the development of pattern matching algorithms with compression support would impact the efficiency of many real-world applications.

There is much work on exact template matching and some in approximate template matching, and many goals have been obtained separately. In this proposal, we have defined the working plan for our thesis. As a general objective, we hope to contribute in the track of approximate searching in compressed image databases. Our specific objectives are to determine the feasibility of extending existing approaches for indexed exact template matching to approximate template matching working on compressed image databases, in particular allowing *don't care* pixels; to design a similarity measure based on information theory to be used for approximate template matching; and to evaluate the impact on the performance and quality of results, in feature vector based similarity search, when the index is compressed in lossy form.

References

[Anu70] P. Anuta. Spatial Registration of Multispectral and Multitemporal Digital Imagery Using Fast Fourier Transform Techniques. *IEEE Transactions on Geoscience Electronics*, 8(4):353–368, 1970.

[AV01] M. Agustí and J. Valiente. Bases de datos para multimedia: Recuperación por contenido. In *Preprints and Electronic Proceedings of the International Workshop on Multimedia Applications, INTERMEDIA 2001*, volume CD, pages TEC–01–28, 2001.

[Bro92] L. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.

[CC98] G. Cha and C. Chung. Multi-mode indices for effective image retrieval in multimedia systems. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, pages 152–159, 1998.

[CC07] P. Clifford and R. Clifford. Self-normalised Distance with Don't Cares. In *Proceedings of the 18th Annual Symposium on Combinatorial Pattern Matching, Lecture Notes in Computer Science 4580*, pages 63–70, 2007.

[CK02] M. Choi and W. Kim. A novel two stage template matching method for rotation and illumination invariance. *Pattern Recognition*, 35(1):119–129, 2002.

[CW06] S. Cheng and T. Wu. Fast indexing method for image retrieval using k nearest neighbors searches by principal axis analysis. *Journal of Visual Communication and Image Representation*, 17(1):42–56, 2006.

[CZPC02] G. Cha, X. Zhu, D. Petkovic, and C. Chung. An efficient indexing method for nearest neighbor searches in high-dimensional image databases. *IEEE Transactions on Multimedia*, 4(1):76–87, 2002.

[DGS01] L. Ding, A. Goshtasby, and M. Satter. Volume image registration by template matching. *Image and Vision Computing*, 19(12):821 – 832, 2001.

[DK82] P. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice-Hall International, 1982.

[FM05] P. Ferragina and G. Manzini. Indexing compressed text. *Journal of the ACM*, 52(4):552–581, 2005.

[FMN07] K. Fredriksson, V. Mäkinen, and G. Navarro. Rotation and lighting invariant template matching. *Information and Computation*, 205(7):1096–1113, 2007.

[FNU02] K. Fredriksson, G. Navarro, and E. Ukkonen. *Recent Research Developments in Pattern Recognition*, chapter Faster than FFT: Rotation Invariant Combinatorial Template Matching, pages 75–112. Transworld Research Network, 2002.

[FNU05] K. Fredriksson, G. Navarro, and E. Ukkonen. Sequential and indexed two-dimensional combinatorial template matching allowing rotations. *Theorical Computer Science*, 347(1-2):239–275, 2005.

[Fre01] K. Fredriksson. *Rotation Invariant Template Matching*. PhD thesis, Department of Computer Science, University of Helsinki, Finland, 2001.

[FS94] J. Flusser and T. Suk. Affine moment invariants: a new tool for character recognition. *Pattern Recognition Letters*, 15(4):433–436, 1994.

[GFP06] N. Giannakeas, D. Fotiadis, and A. Politou. An Automated Method for Gridding in Microarray Images. In *Proceedings of the 28th Annual IEEE International Conference of Engineering in Medicine and Biology Society (EMBS '06)*, pages 5876–5879, 2006.

[GG95] R. Giancarlo and R. Grossi. *Automata, Languages and Programming*, volume 944, chapter On the construction of classes of suffix trees for square matrices: Algorithms and applications, pages 111–122. Springer Berlin / Heidelberg, 1995.

[GG97] R. Giancarlo and R. Grossi. *Pattern Matching Algorithms*, chapter Suffix tree data structures for matrices, pages 293–340. Oxford University Press, 1997.

[GGV03] R. Grossi, A. Gupta, and J. Vitter. High-order entropy-compressed text indexes. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '03)*, pages 841–850, 2003.

[Gia95] R. Giancarlo. A Generalization of the Suffix Tree to Square Matrices, with Applications. *SIAM Journal on Computing*, 24(3):520–562, 1995.

[HSE+95] J. Hafner, H. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient Color Histogram Indexing for Quadratic Form Distance Functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):729–736, 1995.

[KKP03] D. Kim, Y. Kim, and K. Park. Generalizations of suffix arrays to multi-dimensional matrices. *Theoretical Computer Science*, 302(1-3):223–238, 2003.

[KSR02] N. Khan, A. Stockman, and S. Rahman. A Cooperative Environment for Genetic Variance Analysis Using Component Database for Database Integration. In *Proceedings of the 15th IEEE Symposium on Computer-Based Medical Systems (CBMS'02)*, pages 365–368, 2002.

[LC08] Y. Lin and C. Chen. Template matching using the parametric template vector with translation, rotation and scale invariance. *Pattern Recognition*, 41(7):2413–2421, 2008.

[LLYY05] H. Ling, W. Lingda, C. Yichao, and L. Yuchi. Indexing Structures for Content-Based Retrieval of Large Image Databases: A Review. In *Proceedings of the 2nd Asia Information Retrieval Symposium (AIRS 2005)*, pages 626–634, 2005.

[LP96] S. Liao and M. Pawlak. On Image Analysis by Moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(3):254–266, 1996.

[Lu02] G. Lu. Techniques and Data Structures for Efficient Multimedia Retrieval Based on Similarity. *IEEE Transactions on Multimedia*, 4(3):372–384, 2002.

[MCV+97] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, 16(2):187–198, 1997.

[MM93] U. Manber and G. Myers. Suffix Arrays: A New Method for On-Line String Searches. *SIAM Journal on Computing*, 22(5):935–948, 1993.

[MN08] V. Mäkinen and G. Navarro. On Self-Indexing Images - Image Compression with Added Value. In *Proceedings of the 18th Data Compression Conference (DCC)*, pages 422–431, 2008.

[NM07] G. Navarro and V. Mäkinen. Compressed full-text indexes. *ACM Computing Surveys*, 39(1):article 2, 2007.

[Pra74] W. Pratt. Correlation Techniques of Image Registration. *IEEE Transactions on Aerospace and Electronic Systems*, AES-10(3):353–358, 1974.

[PWL+98] G. Penney, J. Weese, J. Little, P. Desmedt, D. Hill, and D. Hawkes. A Comparison of Simularity Measures for use in 2D-3D Medical Image Registration. In *Proceedings of the 1st International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI '98)*, pages 1153–1161, 1998.

[RC96] B. Reddy and B. Chatterji. An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing*, 5(8):1266–1271, 1996.

[Sad03] K. Sadakane. New text indexing functionalities of the compressed suffix arrays. *Journal of Algorithms*, 48(2):294–313, 2003.

[SAO+98] Y. Sugahara, S. Akiyoshi, Y. Okazaki, Y. Hayashizaki, and I. Tanihata. An automatic image analysis system for RLGS films. *Mammalian Genome*, 9(8):643–651, 1998.

[SMA78] M. Svedlow, C. McGillem, and P. Anuta. Image registration: Similarity measure and pre-processing method comparisons. *IEEE Transactions on Aerospace and Electronic Systems*, AES-14(1):141–150, 1978.

[TSOO00] K. Tanaka, M. Sano, S. Ohara, and M. Okudaira. A parametric template method and its application to robust matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 620–627, 2000.

[UK04] F. Ullah and S. Kaneko. Using orientation codes for rotation-invariant template matching. *Pattern Recognition*, 37(2):201–209, 2004.

[UKI01] F. Ullah, S. Kaneko, and S. Igarashi. Orientation code matching for robust object search. *IEICE Transactions on Information and Systems, Special Issue on Image Recognition and Understanding*, 84(8):999–1006, 2001.

[Vaj89] I. Vajda. *Theory of Statistical Inference and Information.* Kluwer Academic Publishers, 1989.

[WLW00] J. Wang, J. Li, and G. Wiederhold. SIMPLIcity: Semantics-sensitive Integrated Matching for Picture Libraries. In *Proceedings of the 4th International Conference on Advances in Visual Information Systems (VISUAL '00)*, pages 360–371, 2000.

[WSB98] R. Weber, H. Schek, and S. Blott. A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. In *Proceedings of the 24rd International Conference on Very Large Data Bases (VLDB '98)*, pages 194–205, 1998.

[ZCP04] Y. Zhu, B. Carragher, and C. Potter. Contaminant detection: improving template matching based particle selection for cryoelectron microscopy. In *Proceedings of the IEEE International Symposium on Biomedical Imaging: Nano to Macro*, volume 2, pages 1071–1074, 2004.