Texture Detection by Genetic Programming

Mario Köppen
Fraunhofer IPK Berlin
Pascalstr. 8-9
10587 Berlin
mario.koeppen@ipk.fhg.de

Xiufen Liu
Technical University Berlin
Straße des 17. Juni 135
10623 Berlin
xiufen@cs.tu-berlin.de

Abstract- This paper presents an approach to blind texture detection in images based on adaptation of the 2D-Lookup algorithm by Genetic Programming. The task of blind texture detection is to separate textured regions of an image from non-textured (as e.g. homogeneous) ones, without any reference to a priori knowledge about image content. The 2D-Lookup algorithm, which generalizes the well-known co-occurrence matrix approach of texture analysis, is based on two arbitrary image processing operations. By Genetic Programming, those image operations can be designed and adapted to a given recognition goal of the whole algorithm. The idea to employ such a framework for texture detection is to use a random image as adaptation goal. Despite of the fact that such a task has no exact solution, the system is able to fulfill this task to a certain degree. This degree is related to textureness in the image: the more texture, the higher the degree. The paper exemplifies this approach.

1 Introduction

Textures are homogeneous visual patterns that we perceive in natural and synthetic scenes. They are made of local micropatterns, repeated somehow, producing the sensation of uniformity. Texture perception plays an important role in human vision. It is used to detect and distinguish objects, to infer surface orientation and perspective, and to determine shape in 3D-scenes.

The restriction of approaches to image understanding on texture processing has recently been paid some attention [1] [2] [3]. The reason is quite simple. On the one hand, there are many image understanding applications on the wishlist of pattern recognition researchers, for which semantic propositions of a given image are a necessary prerequisite, on the other hand there is the current state-of-the-art being far away from meeting even the basic requirements for developping such applications. At least, this can be exemplified by the complete absence of any such applications, as e.g. content-based image search engines, on the internet[1]. Thus, it seems to be more favourable to restrict the application field on applications more resembling the current state-of-the-art.

The basic task of texture detection is to obtain a spatial separation of parts of an image that contain textured regions. Basically, this should be a kind of *blind* image processing, means without using any *a priori* knowledge about the prospective image contents. So far, a great deal of work was devoted to the accompanying non-blind task: describing textures in images based on semantic texture models, or segmentation of images assuming the image to be wholly covered by several texture classes [4] [5] [6] [7]. All of those approaches assume the processed image to be fully-textured.

Figure 1: Example for texture image containg fault.

Figure 2: Goal image for the texture fault in fig. 1, as given by the user.

However, in many contexts, there is a clear advantage of being able to segment an image into textured and non-textured regions. Just to name a few: restricting image manipulations to textured regions, or excluding them from the processing; constructing animations from still images; object separation against a background; obtaining primary semantic descriptions of images asf. This directly leads to the question about a joint property of textured regions, which may be employed to design such a detection procedure.

Since the early days of texture analysis, second-order statistics derived from the co-occurrence matrix of a texture image has been proven to be the most succesfull approach to many kinds of texture processing (especially the detection of texture faults, and the texture segmentation) [8] [6]. Hereby, the co-occurrence matrix is the histogram of the number of occurrences of grayvalue pairs at neighbouring pixel positions. For allocating positions in an image, which are responsible for particular feature values of the co-occurrence

[1] We consider the feature-vector based approaches given by e.g. IBM's QBIC or AltaVista Image Search not a contribution to *content-based* image retrieval.

matrix, a lookup-procedure may be considered, which marks all values in the image leading to a corresponding entry in the co-occurrence matrix.

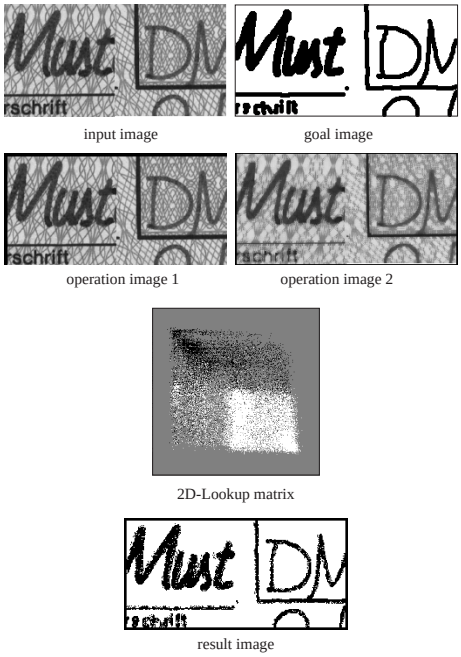

Figure 3: Adapted 2D-Lookup framework applied to bankchecks texture background removal (from [9]).

In this paper, an approach to texture detection is presented, which is based on a generalization of the co-occurence matrix, the so-called 2D-Lookup, and which uses Genetic Programming in order to make the heuristic assignment of 2D-Lookup matrix features to image regions.

The initial reason to develop such a Genetic Programming based adaptation of the 2D-Lookup algorithm was the design of texture filters from user-provided examples. The generated filters are complex image operators, which, in combination with the 2D-Lookup algorithm, gives a foreground-background separation of the image, with a special texture marked as image foreground (e.g. black pixels against a white background). This is assumed to work for *any* kind of texture. The same framework comes out to be able to detect textured regions by changing the user-provided goal, no matter what the particular texture is. Basically, this is achieved by giving the separation of *arbitrary* image regions as adaptation goal to the Genetic Programming. Of course, there is no texture filter possible to solve such a task. But, and this is the basic idea behind the approach presented here, it is able to solve this task to a certain degree: the better the 2D-Lookup adaptation works, the more texture is in the image region. This will be given in more detail in the following text.

Section 2 of the paper recalls the 2D-Lookup based framework for texture filter generation. The following section than reveals the basic idea of using random images in the same framework for separating textured regions from non-textured ones. Then, in section 4, examples for the approach using images of photographs are given and discussed. The paper ends with a summary and the naming of superposing procedures, which employ the given approach.

2 Texture Filter Generation by Genetic Programming

Recently, a framework for the automated generation of texture filters using evolutionary computation, was presented [10]. The framework employs the so-called 2D–Lookup algorithm, which goes as follows.

Given a texture image containing a foreground structure (e.g. a texture fault, handwriting). This input image is assumed to be a grayscale one, with each grayvalue from the set $\{0, 1, \ldots, 255\}$ (consider figure 1). The task is to design a filter, which separates the foreground structure from its textured background. The task is specified by a second binary image, the so-called goal image, which is manually crafted (e.g. by a photo retouching program like Photoshop, consider figure 2 as an example for such a goal image for the image in figure 1). In the goal image, the shape of the foreground structure is painted black, the texture–for–removal region is painted white. This is all, what is given. There is no special texture model known to the user and there are no further requirements for the goal image.

For specifying the 2D-Lookup algorithm, two image processing operations op_1 and op_2 are to be named. If $I(x, y)$ is the image funktion of the input image, for which the filter has to be designed, the two operations, if applied to I, gives two result images $op_1(x, y)$ and $op_2(x, y)$, both of the same size as I.

Now, as a second specification of the algorithm, a set of (crisp) rules is given. Each rule is parametrized by the image coordinates x and y and gives an entry in the result image of the algorithm according to the grayvalue outcome of the two operations. Such a rule has the general form

$$if\ op_1(x, y)\ is\ g_1\ and\ op_2(x, y)\ is\ g_2$$
$$then\ res(x, y)\ is\ lt(g_1, g_2) \in \{0, 1\}$$

There is a maximum of 256×256 rules necessary to specify the algorithm. It is convenient to display all rules by means of an image of size 256×256, the so-called 2D-Lookup matrix, with $lt(g_1, g_2)$ as image function.

Figure 3 gives an example for this procedure, including input image, goal image, result of op_1, result of op_2, 2D-Lookup matrix and result of the algorithm.

The purpose of the framework presented in [10] was to configure the 2D-Lookup algorithm properly just by means of input image and goal image, in order to supply the most simple user interface for texture filter generation. This goal was achieved by using evolutionary algorithms, especially Genetic Programming [10], and a variant of the Nessy algorithm capable of multi–objective optimization [11]. Each individual of the population of the evolutionary algorithm represents

one configuration of the 2D-Lookup algorithm. Therefore, an approach has to be given for the following issues:

- Coding: The individual must specify op_1, op_2 and the 2D-Lookup matrix lt in order to fully specify the 2D-Lookup algorithm.
- Fitness: A measure has to be given for comparing the result of the 2D-Lookup algorithm, as configured by an individual, with the goal image.

By using Genetic Programming (GP), best results were achieved. In GP, each individual is represented by an expression tree, with image operations as node functions. For details of the GP approach see [10]. The result of a GP run is a fully specified filter program, which performs the two operations on an arbitrary input image, and the 2D-Lookup matrix. Figure 4 shows an example for the tree structure of an individual of the Genetic Programming population, and fig. 5 the same tree with the processing images replaced.

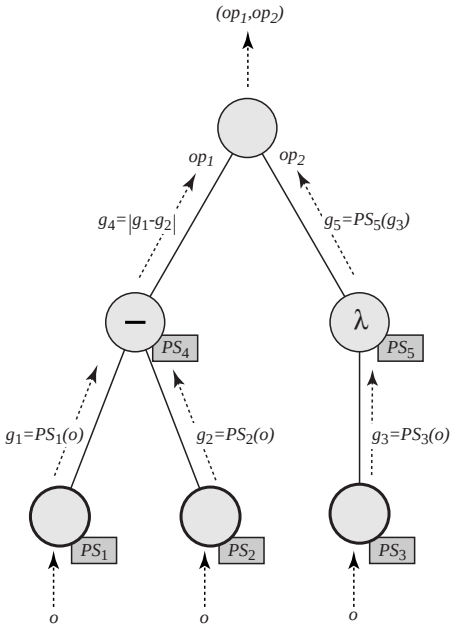

Figure 4: Example for the representation of a complex image processing operation by an individual tree structure.

In [12], an approach for improving the performance of the framework by using neural networks, esp. the Unit RBF neural network [13], was presented.

Meanwhile, this framework was used in several applications, including texture inspection and bankcheck processing [9].

It has to be noted that the approach will not work on the borders of the images. In all cases, there is a border of 5 to 7 pixels width, for which the complex operators will not behave correctly.

Also it should be noted that several runs of the Genetic Programming will give several results. It is not expected to have a means for finding globally optimal solutions of the filtering problem. Moreover, slightly imperfect results gain better generalization performance, as it was demonstrated in [12].

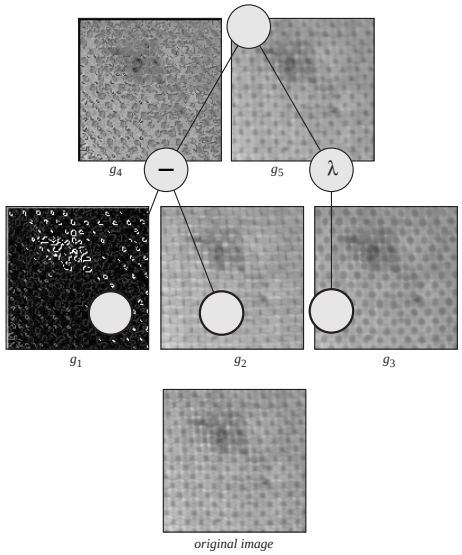

Figure 5: Same tree as for figure 4, with the processing images replaced.

To summarize: the 2D-Lookup adaptation framework creates complex image processing operators by Genetic Programming. The objective of the adaptation is given by means of a goal image. Given a textural scene with a foreground structure of interest, the user can, without referring to any model, draw a goal image, where she marks all positions of the foreground structure in black and all other white.

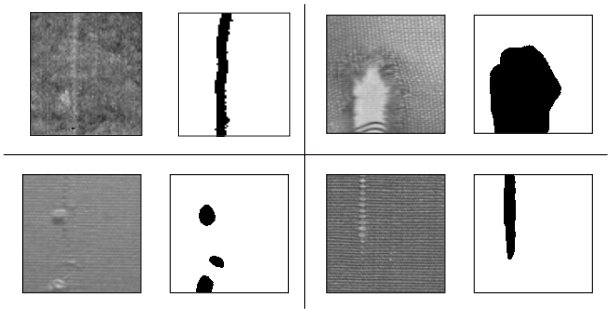

Figure 6: The user interface of the 2D-Lookup adaptation framework: only the texture image and a hand-crafted binary image, which indicates the region of interest, has to be provided by the user.

Figure 6 shows possible pairings of texture image and goal image. Those images constitute the user interface to the adaptation framework.

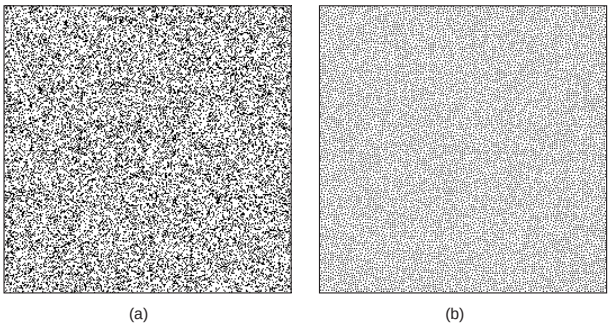

Figure 7: Random dot patterns generated from equally distributed coordinates (a) and from equally distributed coordinates with a minimum distance constraint for nearby dots (b). Only (b) appears to be random in the sense of undistinguished positions, while (a) gives visual cues for the human perception.

3 Texture Detection by 2D-Lookup Adaptation

Basing the approach is the following observation: in figure 7, there are two displays of 20,000 randomly selected pixels in a 256×256 image. The difference between partial figures (a) and (b) is that for the case (b) it was not allowed to have direct neighbours selected. Hence, partial figure (a) is a random dot pattern in the statistical sense, while random pattern (b) seems to be a better representation of "randomness", since there are no regions differentiated from other regions in the image. The observation is that fig. 7 (a) appears to be more structured for the human eye than fig. 7 (b), despite of the fact that only fig. 7 (a) is truely a random pattern.

This fact is underlying the visual and subjective perception of texture. Hence, it also provides a means for detecting textures as well. As the human eyes perceives micropatterns in the dot distribution, the 2D-Lookup procedure can be forced to detect such micropatterns as well.

The approach is demonstrated by a small experiment. For a grayvalue random pattern, a goal image is given, for which an arbitrary region is assigned to black. Despite of the fact that there is no such distinction possible in the random pattern image, the 2D-Lookup adaptation framework is configured in order to achieve such a segmentation. Of course, the more micropatterns there are, the better the chance to achieve a result matching the goal image to a higher degree. In fig. 8, the same experiment is performed for a more homogeneous image, and the lower matching degree after adaptation can be clearly seen. Also, fig. 8 gives the result for a gradient image: the system is only able to select some border-like structures.

From this, the general procedure for texture detection can be derived.

Given an image I with unknown content. From I, a grayvalue image I_g is derived. Then, random binary images are created, as e.g. two opposing chessboard images as given in fig. 9 (the regularity of the patterns in those images is not important for the approach, but technically convenient). Now,

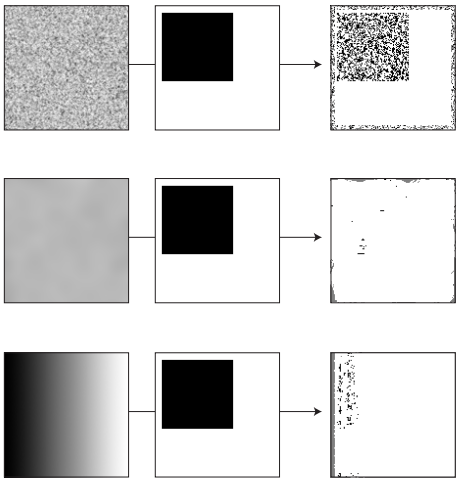

Figure 8: Adapting the 2D-Lookup algorithm by providing an arbitrary goal image and a random pattern image (upper row), a homogeneous image (middle row) and a gradient image (lower row).

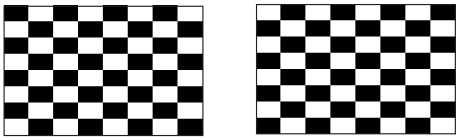

Figure 9: The two goal images with opposing chessboard patterns SW and WS, which were used for the texture detection.

the 2D-Lookup adaptation framework is run twice with those random images as goal images, and the two result images are fused. There is no need to consider the intermediate operation images or the 2D-Lookup matrix, all adapted by the Genetic Programming procedure, at all.

In the result, the cells of the chessboard image with an underlying texture in the corresponding region of image I_g will become more filled with black pixels than for the boxes with underlying non-textured region. By using the opposing chessboard patterns, just after two runs only each of the image pixels was part of one black position in the goal image.

Of course, other binary patterns for the goal images could be used as well. The only point is that there must not be a correlation between the spatial organisation of the image structures and the random pattern. Giving the chessboard patterns and photographic images, it is highly unlikely to get a correlation.

4 Results and Discussion

In fig. 10, a test image of New Caledonia was considered. The result of fusing four processing runs of the 2D-Lookup adaptation framework for each chessboard pattern as goal image is given, clearly distinguishing the central textured part in the image by allocating more black position to those region.

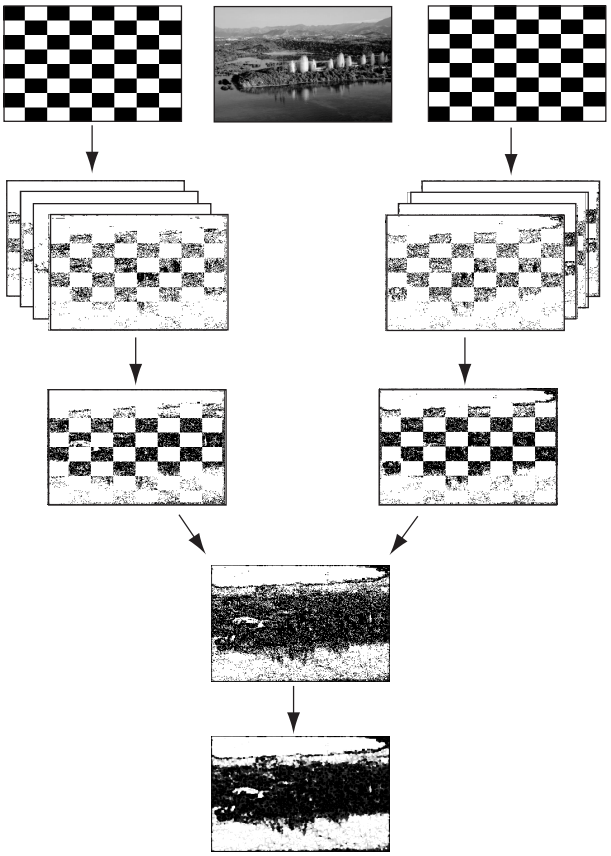

Figure 10: Texture detection on the test image: the figure shows the test image and the two goal images (upper row), the results of four runs of the 2D-Lookup adaptation for each goal image (second row), the fusion of those images into two images (third row), the minimum of both images (fourth row) and a postprocessed result, assigning darker grayvalues to regions with higher pixel density (last row).

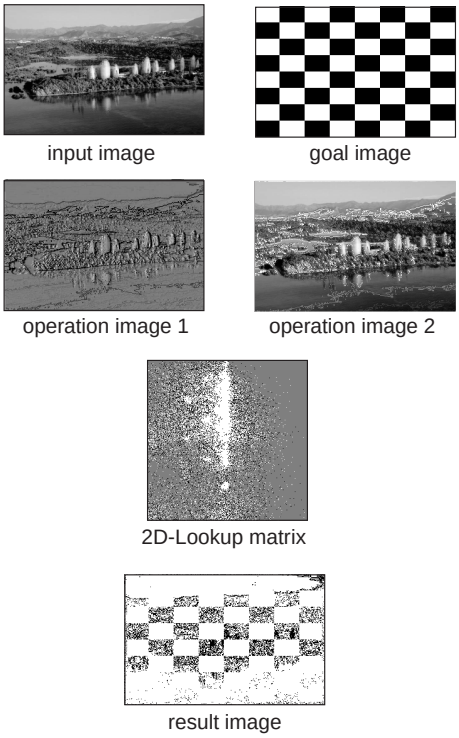

Figure 11: The intermediate result images for one run of the 2D-Lookup adaptation in fig. 10.

Figure 12: Variancy image of the test image.

Figure 11 shows the intermediate processing results for the same case. The 2D-Lookup matrix is surprisingly different from a random pattern, thus resembling underlying non-randomness in the grayvalue distribution of the corresponding image parts. For comparison, fig. 12 shows the corresponding variancy image, for which the value of the variancy of the grayvalues in an 5×5 neighborhood of each pixel is represented as a grayvalue. As it can be seen, this result relies on the original image structures and does not account for e.g. long distanced similarities among pixel patterns (what the given 2D-Lookup adaptation approach is able to do).

5 Outlook

In this paper, an approach to blind texture detection was presented, which is based on the ability of a Genetic Programming procedure to solve a certain task. A framework for 2D-Lookup algorithm adaptation was employed. Normally, the framework is instructed by the provision of a texture image and a hand-drawn goal image. In this goal image, all pixels belonging to a region-of-interest of the texture image (e.g. texture fault region) are marked in black. Now, the 2D-Lookup adaptation designs a texture filter, which comes as close as possible to the given goal image.

For texture detection, a set of random images as goal images is used instead. This means that the 2D-Lookup algorithm is required to solve a basically contradictious goal. However, if there is the occurrence of a similarily textured region at two separated regions of the goal image, both in black, the task may be solved to a certain degree. So, the system will better reproduce the black patterns in the goal image at textured positions.

The approach was considered on a test image and gave a reliable result. The following could be noted:

- The higher the density of reconstructed pixels in the fused result image, the more textureness is in the corresponding image parts.

- For gradients e.g. object borders, the approach will also tend to indicate the object borders.
- It is reasonable to split the processing image into smaller subimages for getting better performance.

The result of such a processing may be used e.g. in the context of digital watermarking, for slight image modification will be less visible in textured regions. Also, from the detected textured regions, goal images for the "regular" use of the 2D-Lookup adaptation can be automatically designed. Then, all different textures in an image can be segmented. This gives primary information about the image content. Hence, we consider the presented procedure, despite of its complexity, as an important first step for the derivation of image descriptions by image processing operations.

Bibliography

[1] Mario Köppen and Javier Ruiz-del-Solar. Fuzzy-based texture retrieval. In *Proceedings FUZZ-IEEE'97, Barcelona, Spain*, pages 471–475, 1997.

[2] Javier Ruiz-del-Solar and Mario Köppen. Autopoiesis and image processing II: Autopoietic-agents for texture analysis. In Masoud Mohammadian, editor, *Computational Intelligence for Modelling, Control & Automation*, pages 72–76, Vienna, Austria, 1999.

[3] Javier Ruiz-del-Solar, Patricio Parada, and Mario Köppen. Texture synthesis using soft-computing. In *Proceedings of the 4th Asian Fuzzy Systems Symposium*, pages 605–610, Tsukuba, Japan, 2000.

[4] Andre Gagalowicz and Christine Graffigne. Blind texture segmentation. In *Intenational Conference on Pattern Recognition 1988*, pages 46–50, 1988.

[5] Roger W. Ehrich and Peichung F. Lai. Elements of a structural model of texture. In *Pattern Recognition and Image Processing 1978*, pages 319–326, 1978.

[6] Robert M. Haralick. Statistical and structural approches to texture. In *Proc. of Int. Joint Conference on Pattern Recognition*, pages 45–69, 1979.

[7] Fumiaki Tomita, Yochiaki Shirai, and Saburo Tsuji. Description of textures by a structural analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, (4)2:183–191, 1982.

[8] R. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man & Cybernetics*, 3(6):610–621, 1973.

[9] K. Franke and M. Köppen. Towards an universal approach to background removal in images of bankchecks. In *Proceedings IWFHR'98, Taejon, Korea*, pages 55–66, 1998.

[10] M. Köppen, M. Teunis, and B. Nickolay. A framework for the evolutionary generation of 2D-Lookup based texture filters. In *Proceedings IIZUKA'98*, pages 965–970, 1998.

[11] M. Köppen and S. Rudlof. Multiobjective optimization by Nessy Algorithm. In P.K. Chawdhry R. Roy, T. Furuhashi, editor, *Advances in Soft Computing*, pages 357–368, 1998.

[12] Mario Köppen, Achim Zentner, and Bertram Nickolay. Deriving rules from evolutionary adapted texture filters by neural networks. In *1999 IEEE International Fuzzy Systems Conference Proceeding*, volume 2, pages 785–790, Seoul, Korea, 1999.

[13] P.G. Anderson. The Unit RBF Network: Experiments and preliminary results. In *Proceedings NC'98*, pages 292–297, 1998.