How to make nD images Well-composed without interpolation

Nicolas Boutry, Thierry Géraud, Laurent Najman

HOW TO MAKE nD IMAGES WELL-COMPOSED WITHOUT INTERPOLATION

Nicolas Boutry[1,2], *Thierry Géraud*[1]

[1] EPITA Research and Development Laboratory (LRDE)
14-16, rue Voltaire
FR-94270 Le Kremlin-Bicêtre, France
firstname.lastname@lrde.epita.fr

Laurent Najman[2]

[2] Université Paris-Est,
Laboratoire d'Informatique Gaspard-Monge (LIGM),
A3SI, ESIEE Paris, Cité Descartes, BP 99
FR-93162 Noisy-le-Grand, France
l.najman@esiee.fr

ABSTRACT

Latecki *et al.* have introduced the notion of well-composed images, *i.e.*, a class of images free from the connectivities paradox of discrete topology. Unfortunately natural and synthetic images are not *a priori* well-composed, usually leading to topological issues. Making any nD image well-composed is interesting because, afterwards, the classical connectivities of components are equivalent, the component boundaries satisfy the Jordan separation theorem, and so on. In this paper, we propose an algorithm able to make nD images well-composed without any interpolation. We illustrate on text detection the benefits of having strong topological properties.

***Index Terms*—** Digital Topology; nD images; Well-Composed Sets; Mathematical Morphology.

1. INTRODUCTION

Well-composedness has been proposed as a practical way to deal with the well-known connectivities paradox encountered with the 4- and 8-connectivities [1, 2, 3]: the $2n$- and (3^n-1)-connectivities are equivalent in a well-composed image [4, 5]. Furthermore, well-composed images enjoy many more properties: the Jordan Separation theorem [3] holds true, we can design much simpler thinning algorithms [6], make the Euler characteristic locally computable [7], simplify graph structures resting from skeleton algorithms [3], and simplify the algorithm of the marching cubes [5, 8].

Two approaches exist to make binary images well-composed. The first one is to keep the original space of the image and to change some of the values of the initial image in such a way that the modified image becomes well-composed [9]. The second is to make an interpolation to preserve the topology of the original image [10, 11]. However this second approach needs a subdivision of the original space and measurably increases the computational costs of the algorithms.

In this paper, we propose a fast method that produces a well-composed image by modifying the original values. The plan of the paper is the following. Section 2 is a short reminder on well-composedness in nD for sets, binary images, and gray-valued images. Section 3 describes an algorithm producing well-composed images. Before concluding the paper, we illustrate with a 2D application to text detection.

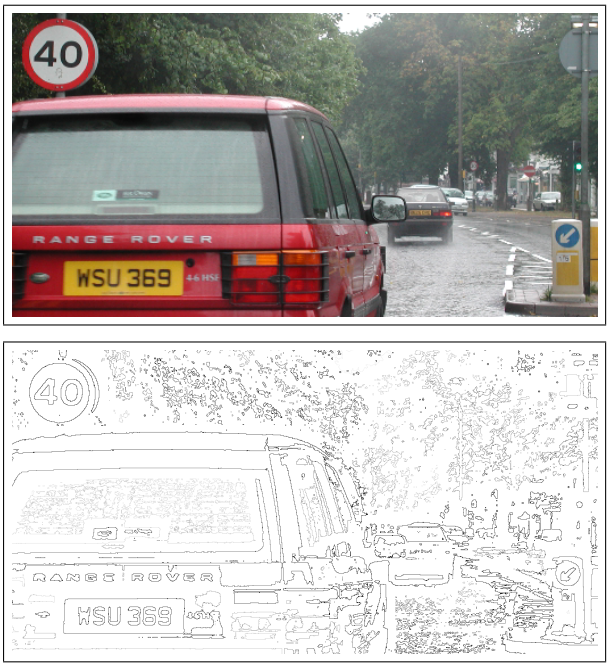


Fig. 1: Hierarchical representation of an image: since component boundaries are simple closed curves on well-composed images, two boundaries are either disjoint or in an inclusion relationship; thus, the delimited regions naturally form a tree.

2. STATE-OF-THE-ART

Latecki *et al.* introduced in [3] the notion of well-composedness for 2D sets. A 2D set is well-composed iff it does

not contain any *critical configuration* $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ or $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. Then Latecki extended the well-composedness to 3D sets in [7]. A 3D set is well-composed iff it does not contain any critical configuration ot *type 1*: $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, or of *type 2*: $\left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right)$ and $\left(\begin{array}{cc|cc} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{array}\right)$, modulo rotations and axial symmetries.

We have proposed in [12] an extension to dimension n of the notion of well-composed sets and images. Let us now recall this extension. With z a point in $\mathbb{Z}^n$ and $\mathcal{F} = \{f_1, \ldots, f_k\}$ a subset of the canonical basis $\mathbb{B}$ of $\mathbb{Z}^n$, we define the *block* associated to the couple $(z, \mathcal{F})$ by:

$$S(z, \mathcal{F}) = \left\{ z + \sum_{j=1}^{\text{card}(\mathcal{F})} \lambda_j . f_j \text{ with } \lambda_j \in \{0, 1\} \right\}, \quad (1)$$

Moreover, we call a block of $\mathcal{D} \subseteq \mathbb{Z}^n$ any element of the set:

$$\mathcal{B}(\mathcal{D}) = \left\{ S(z, \mathcal{F}) \;\middle|\; \exists z \in \mathcal{D},\, \exists \mathcal{F} \subseteq \mathbb{B},\, S(z, \mathcal{F}) \subseteq \mathcal{D} \right\}. \quad (2)$$

Finally, two points p and p' are said *antagonist* in a block S iff they maximize the Euclidian distance between two points in the block S.

A set $X \subseteq \mathbb{Z}^n$ contains a *primary* critical configuration of dimension k iff there exists a block S of dimension $k \in [2, n]$ such as $X \cap S = \{p, p'\}$ where p and p' are antagonist in S. A set $X \subseteq \mathbb{Z}^n$ contains a *secondary* critical configuration of dimension k iff there exists a block S of dimension $k \in [2, n]$ such as $X^c \cap S = \{p, p'\}$ where p and p' are antagonist in S. Then a set is said *well-composed* iff it does not contain any critical configuration.

Let $\mathcal{D} \subseteq \mathbb{Z}^n$ be the domain of the image. For $n = 2$, Latecki *et al.* defined in [3] a well-composed image $u : \mathcal{D} \to \mathbb{Z}$ as an image whose for any $\lambda \in \mathbb{R}$, the *upper threshold sets* $[u \geq \lambda] = \{x \in \mathcal{D} \;|\; u(x) \geq \lambda\}$ and the *lower threshold sets* $[u \leq \lambda] = \{x \in \mathcal{D} \;|\; u(x) \leq \lambda\}$ are well-composed. This definition can straightforwardly be extended in nD [12]. Latecki also introduced a characterization of a well-composed 2D image $u = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$: u is well-composed iff $\text{iv}(a, d) \cap \text{iv}(b, c) \neq \emptyset$, where the interval between a and b is denoted by $\text{iv}(a, b) = [\min(a, b), \max(a, b)]$. We have extended in [13, 12] this characterization to nD images: u is well-composed iff for any block S and for any $p \in S$:

$$\text{iv}(u(p), u(p')) \cap \text{span}(u(p'') \,|\, p'' \in S \setminus \{p, p'\}) \neq \emptyset, \quad (3)$$

with $p' = \text{antag}_S(p)$ and $\text{span}(A) = [\min(A), \max(A)]$, $\forall A \subseteq \mathbb{R}$.

Several methods have been proposed to obtain well-composed images: the interpolation methods [12, 11, 14] which increase the size of the domain of the original image, and the *repairing method* of Siqueira [9], *i.e.*, a method which selects an image close to the original one and which is defined on the same domain. However, the method of Siqueira works only on binary images and is restricted to the 2D and the 3D cases. The goal of this paper is to extend on this previous work, and we propose a method that produces, without any interpolation, well-composed gray-valued images in nD.

3. AN INCREASING PROCESS PRODUCING WELL-COMPOSED IMAGES

3.1. Principle

Let $u : \mathcal{D} \to \mathbb{Z}$ be a given image. We want to find a well-composed image u^* which minimizes the deformation of u,

$$u^* = \arg\min_v \{ \|v - u\|_1 \;|\; v \text{ is W.C.} \} \quad (4)$$

However, to the best of our knowledge, such a combinatorial problem does not have a solution reachable in a reasonable time. To find an approximate solution to this problem, we propose to iteratively select critical configurations and correct them, one by one. To prevent oscillation, we impose that, at each step of the algorithm, the current solution is greater than the previous one. Our process is thus *increasing*.

As we modify a critical configuration, our algorithm is local, in the sense that we only need to look at a block and modify the pixel in the block. However, the modification of the value of a given pixel can create a novel critical configuration in its neighborhood. Hence, there is potentialy a propagation effect, and thus several passes on the image are in principle necessary to achieve convergence.

Due to this propagation effect, the convergence of the algorithm is only ensured if the process is increasing. Indeed, if we allow the modifications to either decrease or increase the image, then oscillation effects could appear.

3.2. Correction Step

We want to correct a given critical configuration in the block $S \in \mathcal{B}(\mathcal{D})$. By definition of a critical configuration, there exists two points $p \in S, p' \in S$ with $p' = \text{antag}_S(p)$, verifying:

$$\text{iv}(u(p),\, u(p')) \cap \text{span}\{\, u(q) \;|\; q \in S \setminus \{p, p'\} \,\} = \emptyset. \quad (5)$$

Then two cases are possible. Either we have:

$$\max(u(p),\, u(p')) < \min\{\, u(q) \;|\; q \in S \setminus \{p, p'\} \,\}, \quad (6)$$

and we set $p^* \leftarrow \arg\max_q \{\, u(q) \;|\; q \in \{p, p'\} \,\}$ and $u(p^*) \leftarrow \min\{\, u(q) \;|\; q \in S \setminus \{p, p'\} \,\}$, or we have:

$$\max\{\, u(q) \;|\; q \in S \setminus \{p, p'\} \,\} < \min(u(p),\, u(p')), \quad (7)$$

then we set $p^* \leftarrow \arg\max_q \{\, u(q) \;|\; q \in S \setminus \{p, p'\} \,\}$ and $u(p^*) \leftarrow \min(u(p),\, u(p'))$. In both cases, u has been made well-composed on S.

Algorithm 1: The correction process.

SOLVECC (u, S) : p
begin
  $p' \leftarrow \text{antag}_S(p)$
  $m_1 \leftarrow \min(u(p),\, u(p'))$
  $M_1 \leftarrow \max(u(p),\, u(p'))$
  $m_2 \leftarrow \min\{\, u(p'') \mid p'' \in S \setminus \{p, p'\}\,\}$
  $M_2 \leftarrow \max\{\, u(p'') \mid p'' \in S \setminus \{p, p'\}\,\}$
  /* Primary case: */
  if $M_1 < m_2$ **then**
    $p^* \leftarrow \arg\max\{\, u(q) \mid q \in \{p, p'\}\,\}$
    $u(p^*) \leftarrow m_2$
  /* Secondary case: */
  if $M_2 < m_1$ **then**
    $p^* \leftarrow \arg\max\{\, u(p'') \mid p'' \in S \setminus \{p, p'\}\,\}$
    $u(p^*) \leftarrow m_1$
  return p^*

3.3. Convergence

The convergence of the method is easy to prove. Indeed, let us define $u_{min} = \min\{\, u(p) \,|\, u(p) \in \mathcal{D}\,\}$ and $u_{max} = \max\{\, u(p) \,|\, p \in \mathcal{D}\,\}$. As the algorithm increases the function u by at least one, we have a maximum of $(u_{max} - u(p))$ corrections for each $p \in \mathcal{D}$. The total number of corrections is then inferior or equal to $\sum_{p\in\mathcal{D}}(u_{max} - u(p)) \leq (u_{max} - u_{min}) \times \text{card}(\mathcal{D})$. This ensures the convergence of the algorithm, since the domain of u is finite.

Algorithm 2: The increasing nD algorithm.

INCREASING (u) : Image
/* Makes the image W.C. */
begin
  /* Initialization of the queue: */
  for all $S \in \mathcal{B}(\mathcal{D})$ **do**
    if CRITICALCONFIGURATION(S, u) **then**
      PUSH(Q, S)
  while $Q \neq \emptyset$ **do**
    $S \leftarrow$ POP(Q)
    /* Correction process: */
    $p \leftarrow$ SOLVECC(u, S)
    /* Detection of the direction of the propagation: */
    for all $S' \in \mathcal{B}(\mathcal{D})$ s.t. $p \in S'$ **do**
      if CRITICALCONFIGURATION(S', u) **then**
        PUSH(Q, S')

3.4. Proposed Algorithm

Given the correction step, the algorithm is straightforward, it is detailed in Algorithm 2. It proceeds in two steps. First, the *initialization step* detects all the critical configurations of the threshold sets $\{[u \geq \lambda]\}_\lambda$ on $\mathcal{D}$ and enqueue them into Q. Second, the *correction step* solves one by one the critical configurations listed into Q using Algorithm 1 and enqueue the new critical configurations which appeared in the neighborhood of the modified value. This algorithm iterates until there is no longer any critical configuration in $\mathcal{D}$; the resulting image u is then well-composed.

3.5. Experimental Results and Complexity of the 2D Case

We used the test set of 100 natural images of the Berkeley image database [15]. Their sizes are (sx, sy) with $\text{sx} = 481$ and $\text{sy} = 321$ pixels or the converse. We cropped each image with ten different windows (for each image) to obtain images of various sizes. The size $(\text{newsx}, \text{newsy})$ of the crop window is randomly chosen into $[\![2, \text{sx}]\!] \times [\![2, \text{sy}]\!]$ and its position is randomly chosen into $[\![1, \text{sx} - \text{newsx} + 1]\!] \times [\![1, \text{sy} - \text{newsy} + 1]\!]$.

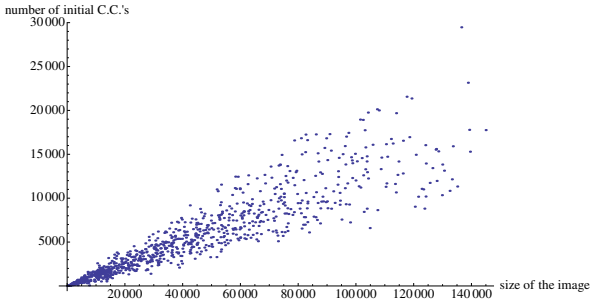


Fig. 2: Number of critical configurations as a function of the size of the image given in number of pixels.

We experimentally assessed the percentage of critical configurations contained in a given image. Fig. 2 shows that between 0.88% and 24.84% of the domain of the original images are covered by critical configurations. From a statistical point of view, a image contains on average $0.1246(\pm 0.036)$ critical configuration by pixel. It is rare to have a well-composed image.

Queue initialization. To initialize the queue of critical configurations, we simply have to detect among the $(\text{newsx} - 1) \times (\text{newsy} - 1)$ blocks which one contains a critical configuration, and in this case we insert it in the queue Q. Each detection and each insertion in the queue is in constant time. This implies that the initialization step in linear time relatively to the size of the image.

Correction process. Concerning the correction step, we had to proceed to c corrections by pixel, with $\text{c} \in [9.10^{-3}, 0.2386]$. From a statistical point of view, an average number of $0.1203(\pm 0.0348)$ corrections by pixel has been observed. Numerical experiments show that the correction step is linear on average with respect to the image size.

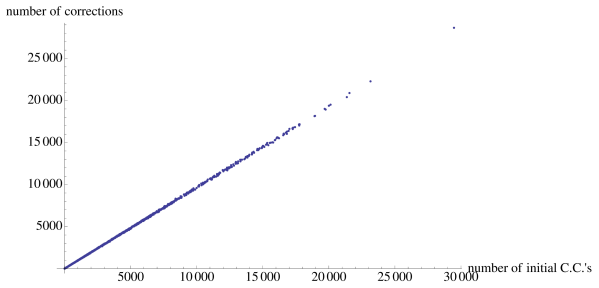


Fig. 3: Number of corrections as a function of the number of initial critical configurations.

The number of corrections by initial critical configuration is not a constant: it can be seen on Fig. 3 that the number of corrections is between $m = 80\%$ and $M = 104\%$ of the number of initial critical configurations. Indeed, a given correction can repair several critical configurations at the same time, which explains that m is less than 100%. Conversely, the propagation effect is responsible for M being greater than 100%. Statistically, we obtain a mean ratio of $0.965764(\pm 0.014)$ corrections by initial critical configuration.

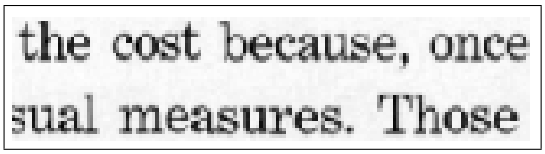
Detection of the direction of the propagation. For each processed correction, there exists only one position $p \in \mathcal{D}$ such as $u(p)$ is modified in the image, and then the propagation is possible in a bounded number of blocks, *i.e.*, in the blocks containing p. This means that the number of blocks processed in the detection step is proportional to the total number of corrections. Since the correction step is in linear time, so is the detection step.

Complexity. Since the 3 steps of the algorithm are in linear time with respect to the initial number of critical configurations, the complete algorithm is in linear time with respect to the size of the image (in number of pixels).

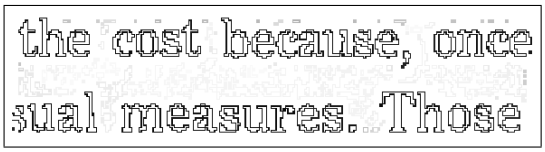
4. ILLUSTRATION AND CONCLUSION

We illustrate the interest of well-composedness to text detection with the morphological Laplacian in 2D. Let us recall that the *morphological Laplacian* $\mathcal{L}$ *of a given image* u is defined as $\mathcal{L}_{\mathrm{se}}(u) = \delta_{\mathrm{se}}(u) + \varepsilon_{\mathrm{se}}(u) - 2u$ where se is a given structuring element. The contours of u are the zero-crossing of the Laplacian. As they are boundaries of level-sets of the grayscale image, the zero-crossing are closed curves. We can set the gray-level of a given contour to the mean of the gradient of the original image along the contour.

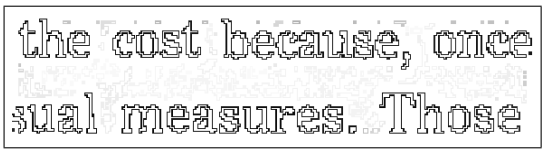
Without correction of the well-composedness, it can be seen on Fig. 4b and Fig. 4d that some characters are broken into several connected components. If we apply the proposed process on the Laplacian image, we observe that the contours are simple. In practice, it can be seen on Fig. 4c and Fig. 4e that the correction repairs many contours.



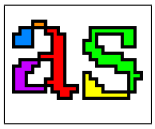
(a) Original image u.



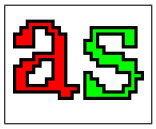
(b) Zero-crossings of the original Laplacian.



(c) Zero-crossings of the Laplacian modified by the increasing process.



(d) Crop of (b).



(e) Crop of (c).

Fig. 4: Results obtained without and with the algorithm; in (d) and (e) the connected components have been colorized.

In conclusion, we have presented a new algorithm that produces well-composed images without interpolation. Compare to the interpolation methods, the proposed algorithm is faster and less memory consuming. It can be seen as a natural extension of the algorithm of topological repair of Siqueira *et al.* [9] to gray-valued images.

Future work will apply the methods to segmentation problems, including text detection in natural images, where first tests are encouraging. Another research direction is to take advantage of the natural hierarchy provided by the closed contours of the zero-crossing of the Laplacian. This hierarchy is yet another promising tree-based image representation that fits into the morphological framework [16].

The source code of the proposed algorithm has been implemented using our image processing C++ library "Milena" [17, 18], which is free software under the GNU Public Licence v2. Since we advocate reproducible research, this source code will be released on our web site as supplementary material (if this paper is accepted).

5. REFERENCES

[1] A. Rosenfeld and J.L. Pfaltz, "Sequential operations in digital picture processing," *Journal of The Association of Computing Machinery*, vol. 13, no. 4, pp. 471–494, Oct. 1966.

[2] T. Y. Kong and A. Rosenfeld, "Digital topology: Introduction and survey," *Comput. Vision Graph. Image Process.*, vol. 48, no. 3, pp. 357–393, Dec. 1989.

[3] L.J. Latecki, U. Eckhardt, and A. Rosenfeld, "Well-composed sets," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 70–83, January 1995.

[4] A. Rosenfeld, "Connectivity in digital pictures," *Journal of the ACM*, vol. 17, no. 1, pp. 146–160, January 1970.

[5] N.J. Tustison, B.B. Avants, M. Siqueira, and J.C. Gee, "Topological well-composedness and glamorous glue: A digital gluing algorithm for topologically constrained front propagation," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1756–1761, 2011.

[6] J. Marchadier, D. Arquès, and S. Michelin, "Thinning grayscale well-composed images," *Pattern Recognition Letters*, vol. 25, pp. 581–590, April 2004.

[7] L.J. Latecki, "3D well-composed pictures," *Graphical Models and Image Processing*, vol. 59, no. 3, pp. 164–172, May 1997.

[8] T. Etiene, L.G. Nonato, C. Scheidegger, J. Tierny, T.J. Peters, V. Pascucci, R.M. Kirby, and C.T. Silva, "Topology verification for isosurface extraction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 6, pp. 952–965, 2012.

[9] M. Siqueira, L.J. Latecki, N. Tustison, J. Gallier, and J. Gee, "Topological repairing of 3D digital images," *Journal of Mathematical Imaging and Vision*, vol. 30, no. 3, pp. 249–274, 2008.

[10] L.J. Latecki, "Well-composed sets," in *Advances in Imaging and Electron Physics*. 2000, vol. 112, pp. 95–163, Academic Press.

[11] P. Stelldinger and L.J. Latecki, "3D object digitization: Majority interpolation and marching cubes," in *Proc. of the 18th International Conference on Pattern Recognition*, 2006, vol. 2, pp. 1173–1176.

[12] N. Boutry, T. Géraud, and L. Najman, "How to make well-composed images in nD in a self-dual way with a front propagation algorithm (submitted)," in *Mathematical Morphology and Its Applications to Signal and Image Processing—Proceedings of the International Symposium on Mathematical Morphology (ISMM)*, 2015.

[13] N. Boutry, T. Géraud, and L. Najman, "On making nD images well-composed by a self-dual local interpolation," in *Discrete Geometry for Computer Imagery*. 2014, vol. 8668 of *Lecture Notes in Computer Science Series*, pp. 320–331, Springer.

[14] R. Gonzalez-Diaz, M.-J. Jimenez, and B. Medrano, "Well-composed cell complexes," in *Proceedings of Discrete Geometry for Computer Imagery*. 2011, vol. 6607 of *Lecture Notes in Computer Science Series*, pp. 153–162, Springer.

[15] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings of the 8th International Conference on Computer Vision (ICCV)*, July 2001, vol. 2, pp. 416–423.

[16] L. Najman and J. Cousty, "A graph-based mathematical morphology reader," *Pattern Recognition Letters*, vol. 47, pp. 3–17, Oct. 2014.

[17] R. Levillain, T. Géraud, and L. Najman, "Why and how to design a generic and efficient image processing framework: The case of the Milena library," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Hong Kong, September 2010, pp. 1941–1944.

[18] R. Levillain, T. Géraud, and L. Najman, "Writing reusable digital topology algorithms in a generic image processing framework," in *Applications of Discrete Geometry and Mathematical Morphology – First International Workshop, WADGMM 2010, Istanbul, Turkey, August 22, 2010, Revised Selected Papers*, Ullrich Köthe, Annick Montanvert, and Pierre Soille, Eds. 2012, vol. 7346 of *Lecture Notes in Computer Science*, pp. 140–153, Springer-Verlag Berlin Heidelberg.