Computer Science Department of The University of Auckland
CITR at Tamaki Campus (http://www.citr.auckland.ac.nz/)

CITR-TR-107 December 2001

Switches May Solve Adjacency Problems

Reinhard Klette[1]

Abstract

This paper presents a new method for defining grid-point adjacencies, called the switch approach. It is discussed how it relates to connectedness definition in multi-valued images. The paper illustrates how the method can be used, and provides a few experimental data illustrating the relevance and simplicity of the approach.

[1] Center for Image Technology and Robotics Tamaki Campus, The University of Auckland, Auckland, New Zealand. r.klette@auckland.ac.nz

You are granted permission for the non-commercial reproduction, distribution, display, and performance of this technical report in any format, BUT this permission is only for a period of 45 (forty-five) days from the most recent time that you verified that this technical report is still available from the CITR Tamaki web site under terms that include this permission. All other rights are reserved by the author(s).

Switches May Solve Adjacency Problems

Reinhard Klette
CITR, Department of Computer Science,Tamaki Campus
The University of Auckland, Private Bag 92019, Auckland, New Zealand
Email: r.klette@citr.auckland.ac.nz

Abstract

This paper presents a new method for defining grid-point adjacencies, called the switch approach. It is discussed how it relates to connectedness definition in multi-valued images. The paper illustrates how the method can be used, and provides a few experimental data illustrating the relevance and simplicity of the approach.

Keywords: grid-point adjacencies, pixel neighborhoods.

1. Introduction

Neighborhood or adjacency structures of digital images, and topological problems related to image analysis have been studied over the last thirty years [3]. This article suggests a simple and new model for data-dependent pixel adjacencies which might be an interesting alternative. Data-dependency of adjacencies is also discussed in [6] using hypergraphs, allowing even incorporations of local image data variances into the adjacency definition.

An *image* I is a discrete-valued function defined on a rectangular set $\mathbb{C}_{m,n} \subset \mathbb{Z}^2$ of grid points. The range is $\{0, \ldots, G_{max}\}$ with $G_{max} \geq 1$. In case $G_{max} = 1$ we have a *binary image*. To simplify our discussions, assume that any image defines different equivalence classes $\mathbb{C}_{u_0}$ of points $p \in \mathbb{C}$ by its values u_0, $0 \leq u_0 \leq G_{max}$: points p and q are *I-equivalent* iff $I(p) = I(q)$.

Papers [7, 8] defined connected subsets in the orthogonal grid, based on 4-adjacency or 8-adjacency. The separation problem in binary image analysis is solved by using two different adjacency definitions (called a *good pair* in [1]) on the image adjacency graph. For example, this suggests for $M = I^{-1}(1)$ to use 4-connectedness for M, and to use 8-connectedness for $\overline{M} = I^{-1}(0)$; or vice-versa. See Fig. 1 for an illustration of good pairs (8,4) and (4,8), where all object points are shown as filled dots and all background points are shown as hollow dots.

Valid adjacencies are between adjacent grid points which are labeled by identical image values. Valid adjacencies are shown by connecting line segments. *Invalid adjacencies* are between points in different I-equivalence classes defined by different values of image I.

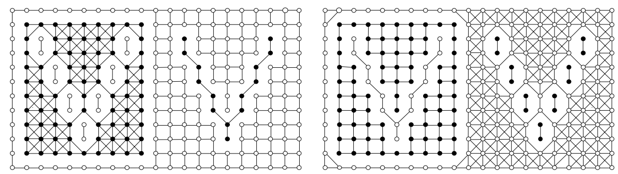


Figure 1. Left: good pair (8,4). Right: good pair (4,8) (this binary image example has been discussed in [5]). There are 'cuts' of the V-shape in both cases established by 8-adjacencies.

The left half of the binary image (an example from [5]) in Fig. 1 shows a 'background V', the right half shows an 'object V'. There is only one connected 'V' in both copies of this image, for good pair (8,4) and for good pair (4,8). The second 'V' is (already) disconnected by 8-connected pixels, i.e. it may happen that subsequent image analysis procedures have to disconnect these pixels again.

Image analysis normally deals with multi-level input images (actually even with multi-channel images in an increasing number of applications), i.e. we have $G_{max} > 1$. See Fig. 2 for an example; pixel values are shown as shaded

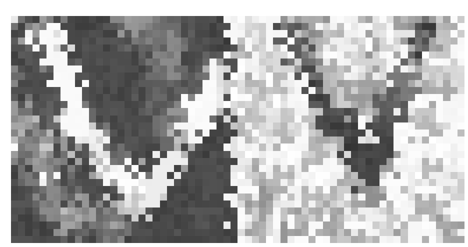


Figure 2. Multi-level input image as normally given in image analysis.

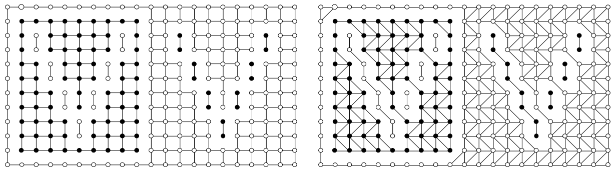


Figure 3. Pair (4,4) and good pair (6,6): there are still a few ‘cuts’ as in Fig. 1 for (6,6). Missing connections may be obtained by subsequent image analysis approaches.

squares. The concept of good pairs cannot be extended to such multi-level images I where a similar consistency of different neighborhood definitions for $I^{-1}(u)$, for $0 \leq u \leq G_{max}$, is impossible if $G_{max} > 1$.

Alternative orientations of diagonals defining 6-adjacencies introduce a (systematic) directional bias into the resulting 6-components. Figure 3 illustrates the pair (4,4) and the good pair (6,6). Both, $(4,4)$ and $(6,6)$ are planar graph structures. The connectedness approach defined by the pair $(4,4)$ is used in several major commercial image processing systems sold worldwide, see Fig. 4: pixels are shown again as squares; an 8-curve 4-separates one interior 4-component from one exterior 4-component, but the 8-curve itself is not connected according to the system.

2 Switches

Undirected grid edges represent a symmetric and irreflexive adjacency relation.

We use all isothetic grid edges representing 4-adjacency, plus selected diagonals in grid squares specified in the following definition.

Definition 1 *Take the lower left corner of a grid square as the reference point for a* switch *which is a grid diagonal*

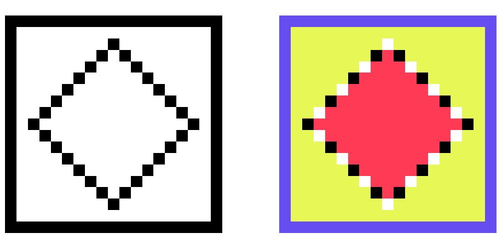


Figure 4. Connected components (right) as produced by a major commercial image processing system for input image shown on the left.

being either in an on-, or in an off-state, see Fig. 5. The state of a switch needs to be such that the grid diagonal connects grid points being in the same equivalence class (i.e. having identical image values) if there is such a pair of diagonal points in the given grid square; if both diagonals connect grid points in identical equivalence classes then a state may be chosen.

Note that we only allow one grid diagonal per grid square. The resulting (inhomogeneous in general) planar graphs are examples of adjacency graphs as studied in [2], examples of two-dimensional strongly normal digital picture spaces in the sense of [4], and also examples of planar generic axiomatized digital surface-structures (GADSs) as discussed in [1],

Figure 5 shows on the right all possible 2×2 image value configurations: filled dots illustrate pixels $(p, I(p))$ belonging to one equivalence class C, and hollow dots illustrate pixels belonging to different equivalence classes (not necessarily just to one category different to C). The state of the switch is unimportant in cases (a) (both diagonals connect points in class C), (e) and (f) (both diagonal pairs are points in different classes). The state of the switch is uniquely defined in cases (b), (c) and (d) because there is just one diagonal pair of points which are in the same class. In situation (h) we choose the off-state because the connected diagonal pair might be in the same class.

2.1 Switch State Matrix

The only remaining problem is the *flip-flop case* (g) (in fact absolutely analogously to the Euclidean plane when two curves intersect at one point, and the assignment of the intersection point decides how these two curves subdivide the Euclidean plane!): if both diagonal points shown by hollow dots are in different classes then the switch will be in on-state. Otherwise we call a procedure *SetSwitches* to chose either the on- or the off-state. Important is that the

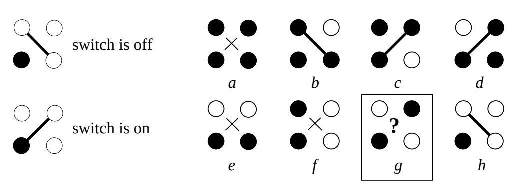
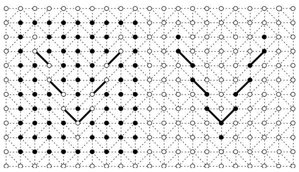


Figure 5. The reference point is at the lower left corner: states of a switch (left), and for all possible image value assignments on a grid square, there is only flip-flop case (g) where the position of the switch needs to be decided. The cross stands for a don’t-care-situation.



-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1	0	1	1	0	0	1	0	1	0	0	1	0	0	0	1	0	1	0	0	-
0	1	0	1	1	0	1	1	0	0	1	1	0	1	1	0	0	1	0	0	-
1	1	1	0	1	0	0	1	0	1	0	1	0	1	1	0	1	0	1	0	-
1	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	1	1	1	-
0	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	1	1	1	0	-
0	0	0	0	0	1	1	1	0	0	1	0	0	0	0	1	1	1	0	1	-
0	0	0	0	1	1	1	0	1	1	0	0	1	0	0	0	1	1	1	0	-
0	1	0	0	0	1	1	0	1	0	1	0	1	0	0	1	1	1	1	1	-
1	1	1	0	0	1	0	0	1	0	0	1	0	0	0	1	1	0	1	1	-
0	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0	-
0	0	1	0	1	1	0	1	0	1	0	1	1	0	1	0	0	0	0	0	-

Figure 6. States of switches are uniquely defined, or can be chosen randomly in most of the cases. Just a few flip-flop switches (12 in this example) are decided by a procedure *SetSwitches* defined by the local templates shown in Fig. 7. The binary matrix $\mathbb{S}$ on the right encodes the states of all switches.

state can't be changed again during one topological operation on a picture after it has been set.

The procedure *SetSwitches* may, for example, analyze larger neighborhoods of the reference point for defining the state of its switch. See Fig. 6 for a possible specification of switches, where a procedure *SetSwitches* has been used in a bottom-up, left-to-right fashion: assigning switches randomly in don't-care situations, and using the local templates shown in Fig. 7 for the flip-flop case (g). These templates are such that the state of the switch in the grid square below is simply copied as the new state of the switch in the recent grid square. The example shows that it is possible to assign switches such that both V-shapes remain connected. Of course, more advanced procedures *SetSwitches* may be designed, using larger neighborhoods for enforced control about switch settings. In an implementation of the approach a binary image may be used with value 0 at point p iff the switch at reference point p is in off-state, and value 1 otherwise (see left of Fig. 6). The topology of a digital image is then specified by such an accompanying binary *switch state matrix* $\mathbf{S}$, which defines an $\mathbf{S}$-adjacency between grid points and, subsequently, $\mathbf{S}$-connectedness. Figure 8 shows the resulting subgraph of valid adjacencies between $\mathbf{S}$-adjacent grid points p, q having identical image values, i.e. $I(p) = I(q)$, where $\mathbf{S}$ as shown in Fig. 6.

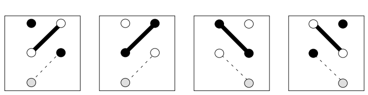


Figure 7. Set of simple templates for defining flip-flop switches.

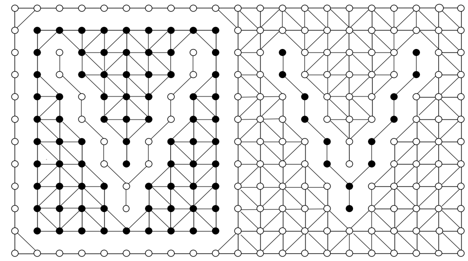


Figure 8. Valid $\mathbb{S}$-adjacencies for the planar adjacency graph shown in Fig. 6.

2.2 Practical Aspects

The discussed switches ensure that the resulting $\mathbf{S}$-adjacency graph is always planar. Any image processing step, e.g. a simple local filter, contrast enhancement, or an interactive modification of single image values, will/may create a need to update the switch state matrix $\mathbf{S}$ of a given image. Of course, matrix $\mathbf{S}$ is only needed if a topological operation is called, and its calculation is very simple. The *switch-approach* can be summarized as follows: every grid square contains one grid diagonal only, as in case of the good pair (6,6). However, to avoid a directional bias, or to reflect the given image data, the diagonals (switches) may be either set randomly or based on rules as discussed above. Figure 10 shows two possible results depending on the chosen set of templates for defining switches.

Figure 9. Upper left: this 2014×1426 picture, i.e. 2,872,964 pixels, with $G_{max} = 255$, possesses 14,359 flip-flop cases, i.e. 0.50$\%$ of all grid points. Upper right: 0.38$\%$. Lower left: 0.38$\%$. Lower right: 0.22$\%$.

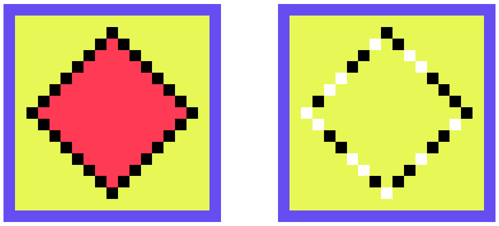


Figure 10. Possible components.

A switch state matrix **S** needs to be available at the time of a topological operation in an image such as contour tracing or thinning. The matrix **S** ensures that only planar adjacency graphs are used, and it can be

(i) always the same switch state matrix (look-up table $\mathbf{S}_{m,n}$), just defined by the size $m \times n$ of the image and calculated by using a random number generator,

(ii) a function which produces a binary pseudo-random number based on the coordinates of the reference point $p = (i, j)$, allowing that actually no switch state matrix is needed, just a local calculation of the pseudo-random switch state (e.g., if the size of the images varies frequently),

(iii) an updated switch state matrix **S** using the rules as discussed for Fig. 5 and an image data-dependent procedure for dealing with the flip-flop cases (which, in fact, appear very rarely in captured images, see Fig. 9, i.e. this option might be of interest for cases of very high-precision image capturing only).

Due to a certain degree of randomness in captured image data (due to sensor noise, uncertainties in image data, illumination changes etc. and the low percentage of locally (i.e. in a 2×2 window) undecidable flip-flop states (note: typically less than 0.5% for grayscale images, see Fig. 9, and less than 0.2% for color images) it is normally appropriate to use one of the first two options. Of course, just a small number n of flip-flops at the border of an image component defines 2^n different ways of allowing connections to neighboring components etc.

3 Concluding Remarks

This report informed about the switch-approach which is a simple, data-dependent method for specifying adjacencies in multi-level images. The switch-approach has shown that context-dependent connectivities may be achieved by adding more structural elements (namely grid diagonals) to the 4-adjacency graph. Experimental studies have shown that additional needs in computing time are minor. Different sets of templates support different models for defining connected components (e.g. a preference towards line-like components): the *SetSwitches* procedure may be designed such that curve-like patterns are preferably connected, for example using larger neighborhoods than in the simple set of templates shown in Fig. 7. The switch-approach is not designed for extreme cases such as ‘chessboard-like’ binary image segments.

References

[1] S. Fourey, T. Y. Kong, G. T. Herman: Generic axiomatized digital surface-structures. In: *Electronic Notes in Theoret. Comp. Science* **46** (2001) 20 pages.

[2] R. Klette, K. Voss, P. Hufnagl: Theoretische Grundlagen der digitalen Bildverarbeitung. Part II. Nachbarschaftsstrukturen. *Bild und Ton* **38** (1985) 325–331.

[3] R. Klette: Digital topology for image analysis - Part 1. CITR-TR-101, The University of Auckland (2001).

[4] T. Y. Kong, A. W. Roscoe, A. Rosenfeld: Concepts of digital topology. *Topology and Its Applications* **46** (1992) 219–262.

[5] V. Kovalevsky: Finite topology as applied to image analysis. *CVGIP* **46** (1989) 141–161.

[6] S. Rital, A. Bretto, D. Aboutajdine, H. Cherifi: Application of adaptive hypergraph model to impulsive noise detection. Proc. *CAIP 2001*, Springer, Berlin **LNCS 2124** (2001) 555–562.

[7] A. Rosenfeld, J. L. Pfaltz: Sequential operations in digital picture processing. *J. ACM*, **13** (1966) 471–494.

[8] A. Rosenfeld: Connectivity in digital pictures. *J. ACM* **17** (1970) 146–160.