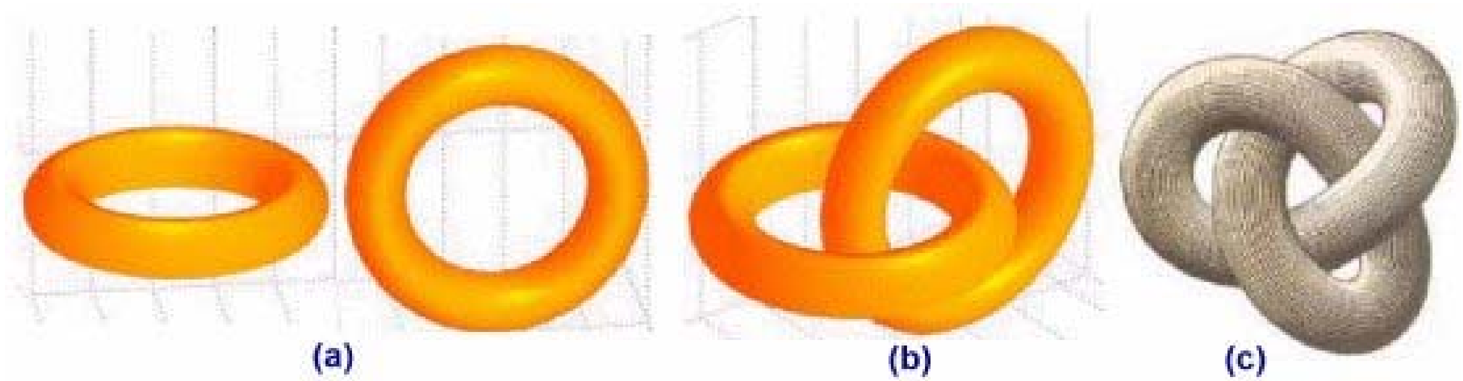
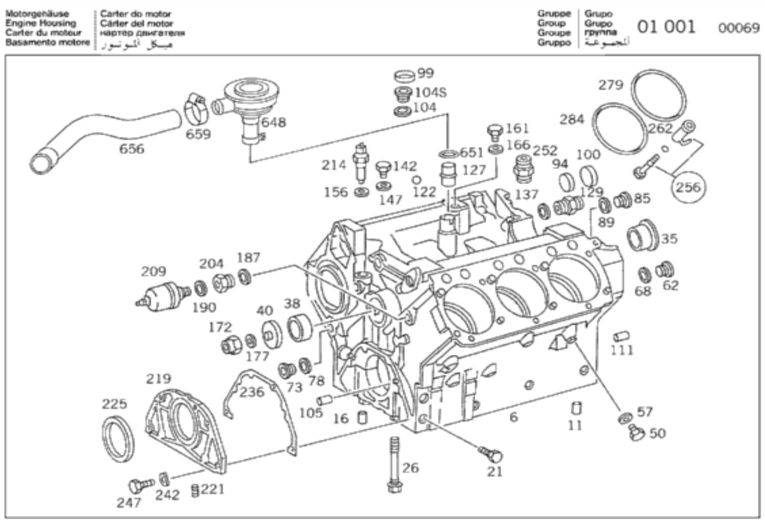
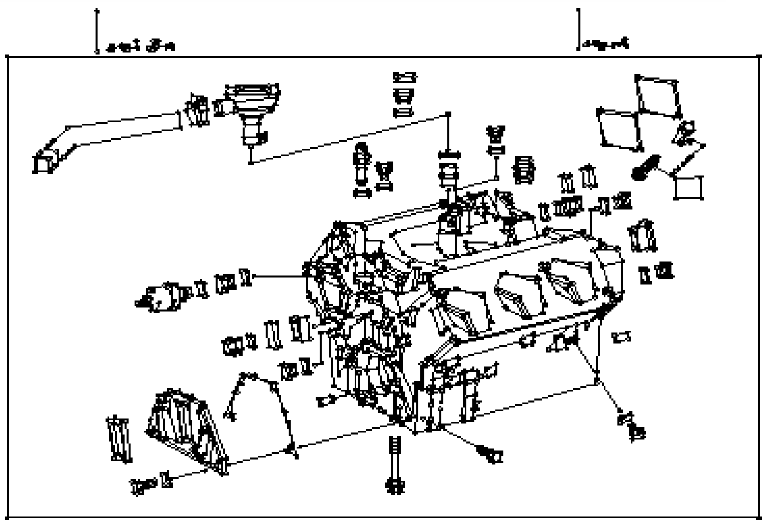
Hierarchies relating Topology and Geometry

Walter G. Kropatsch, Yll Haxhimusa, Adrian Ion
PRIP (183/2)
Pattern Recognition and Image Processing
Institute of Computer Aided Automation
Vienna Univ. of Technology- Austria

Graphs in Image Analysis







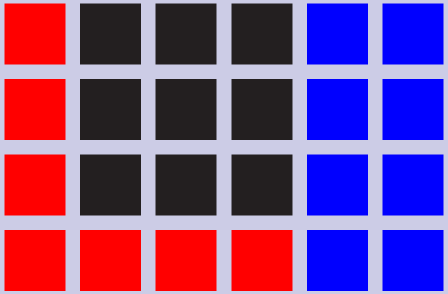
- Dual Graph Contraction
- Graph Pyramids

CONTENTS

- The Problem
- Partitions and Adjacencies
- How to Reduce the Descriptions
- Topology Preserving Operations
- Combinatorial Maps
- Abstract Cellular Complexes (ACC)
- Conclusions, Discussion

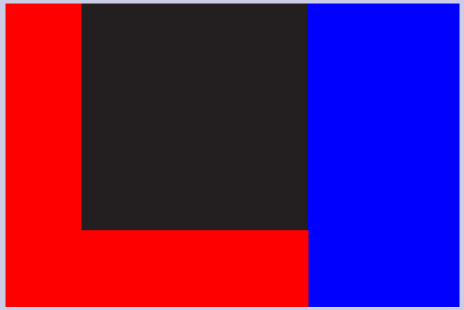
The PROBLEM **in 2D**

Label Array $\mathcal{P}_0$

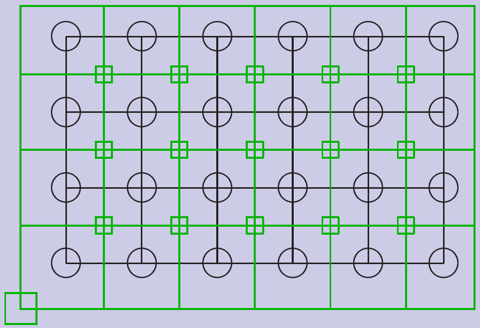


$\longrightarrow$

Regions $\mathcal{P}_1$

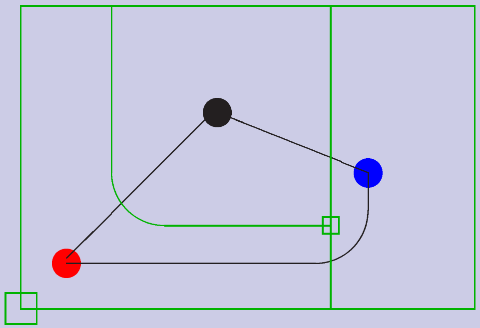


Pixel Adjacency $\mathcal{A}_0$



$\longrightarrow$

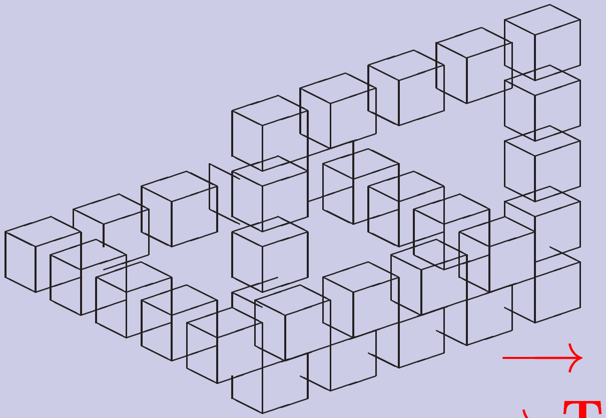
Region Adjacency $\mathcal{A}_1$



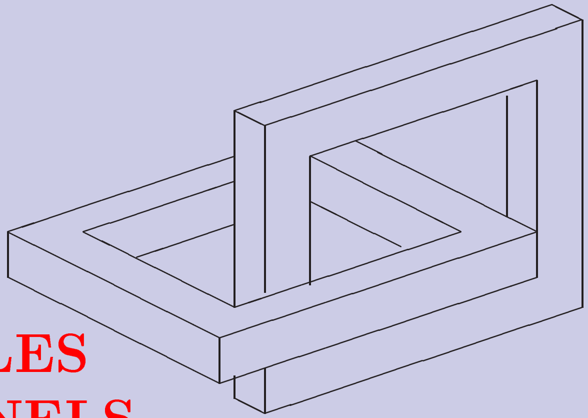
$lab : \mathcal{P}_0 \mapsto \mathcal{L}$	Partition	$lab : \mathcal{P}_1 \mapsto \mathcal{L}$
$\lvert\mathcal{P}_0\rvert > \lvert\mathcal{L}\rvert$		$\lvert\mathcal{P}_0\rvert > \lvert\mathcal{P}_1\rvert \geq \lvert\mathcal{L}\rvert$
$(\mathcal{P}_0, \mathcal{A}_0, \mathcal{L})$-Properties	$\longrightarrow$	$(\mathcal{P}_1, \mathcal{A}_1, \mathcal{L})$-Prop.

The PROBLEM in 3D

Label Array $\mathcal{P}_0$ $\longrightarrow$ Regions $\mathcal{P}_1$



$\longrightarrow$

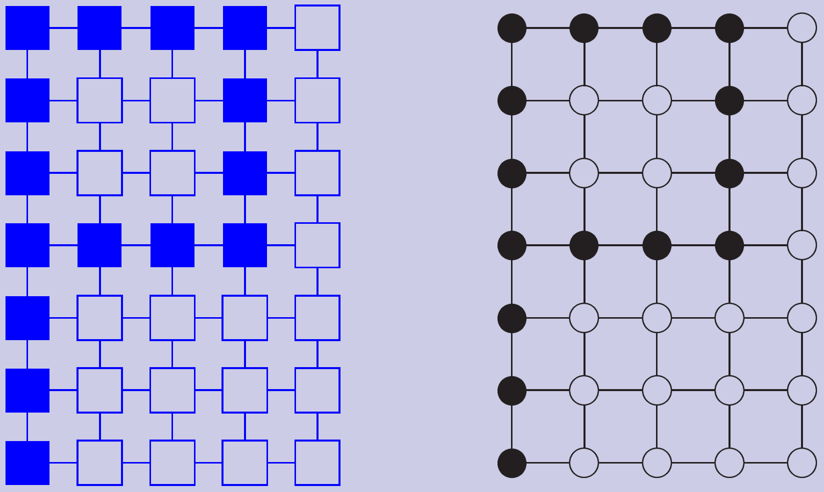


$\longrightarrow$ **HOLES**

$\longrightarrow$ **TUNNELS**

From Pixels To Graphs

IMAGE PIXELS NEIGHBORHOOD GRAPH



Properties/Relations between $(\mathcal{P}_0, \mathcal{A}_0)$ and $(\mathcal{P}_1, \mathcal{A}_1)$

$\mathcal{P}_0$	partition 1 Label/Cell	$\mathcal{P}_1$	partition (?) 1 Cell/Label
$c \in \mathcal{P}_0$	unit square simply connected simplex no inclusion 1 Label/CC ?	$c \in \mathcal{P}_1$	arbitrary shape (simply) connected simplex ? inclusion (tree) $\|CC\| = \|\mathcal{P}_1\|$ minimal
$\mathcal{A}_0$	4-connected ? well composed embedding Genus(CC(label)) orientation	$\mathcal{A}_1$	RAG (connected ?) connectivity preserved embedding Genus(CC(label)) orientation artefacts?
(c_1, c_2) $\in \mathcal{A}_0$	$lab(c_1) \neq lab(c_2)$ $lab(c_1) = lab(c_2)$ image edge Jordan boundaries	(c_1, c_2) $\in \mathcal{A}_1$	multiple (lab_1, lab_2) (lab_1, lab_1) (pseudo,self-loop) connected segment Jordan boundaries

next: How to perform $T : (\mathcal{P}_0, \mathcal{A}_0) \mapsto (\mathcal{P}_1, \mathcal{A}_1)$?

How to perform $T : (\mathcal{P}_0, \mathcal{A}_0) \mapsto (\mathcal{P}_1, \mathcal{A}_1)$?

$(\mathcal{P}_0, \mathcal{A}_0)$	$(\mathcal{P}_1, \mathcal{A}_1)$
Pixelarray: $I(x, y)$	Combinatorial 2-Maps $(\boldsymbol{\mathcal{D}}_1, \alpha_1, \sigma_1)$ + Incl.Tree
ACC	Abstract Cellular Complex-graph?
	G-Maps + Incl.Tree $(\boldsymbol{\mathcal{D}}_1, \alpha_0, \alpha_1, \alpha_2)$
Dual Graphs: $((V_0, E_0), (F_0, \overline{E_0}))$	Dual Graphs $((V_1, E_1), (F_1, \overline{E_1}))$
2-Maps $(\boldsymbol{\mathcal{D}}_0, \alpha_0, \sigma_0)$	$(\boldsymbol{\mathcal{D}}_1, \alpha_1, \sigma_1)$

Possible Realizations of T:

1. direct construction
   (a) Connected Component Labeling
   (b) 2-Maps by Pixelscan
   (c) 2-Maps by Precode-Guided FUSION
   (d) G-Maps by Precode-Guided FUSION?
2. DS conversion + repeated reductions
   (a) 2-Maps by Precode-Guided (sequential) FUSION
   (b) Dual Graphs by Parallel Dual Contractions

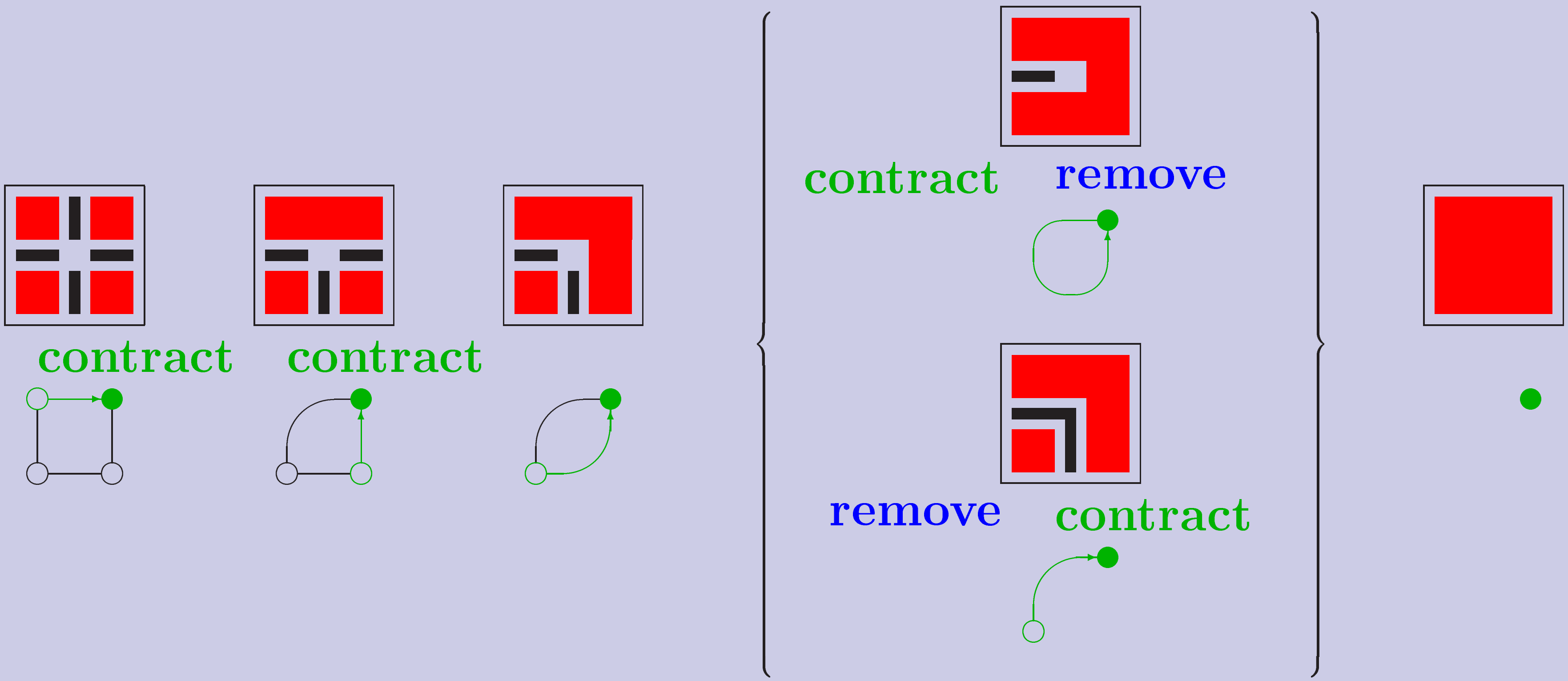
2D Removal and Contraction

Configuration	Primal Graph	Dual Graph
A ▮ B	Remove ▮	Contract A ⟶ B
$A + B$	merges A and B	

Common(A,B) true $\Longleftrightarrow$

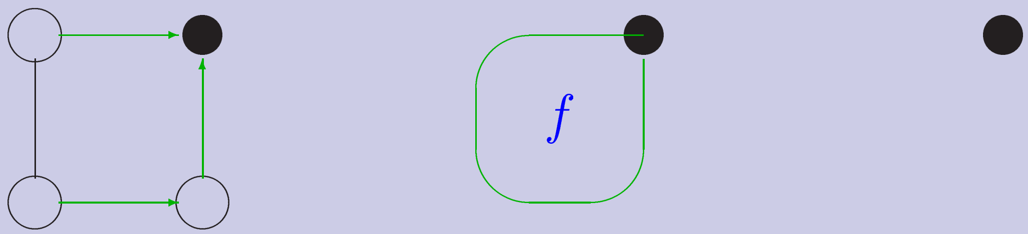
Same Label	$\longrightarrow$	CCL
Similar Color	$\longrightarrow$	Segmentation
'belong together'	$\longrightarrow$	Grouping

2 × 2 CCL



Contraction Kernel, Simplification

contract more **simplified**



contraction kernel self-loop removed

Spanning tree $\deg(f) < 3$

Irregular Graph Pyramid
Building Algorithm

Dual Graph Pyramid Algorithm

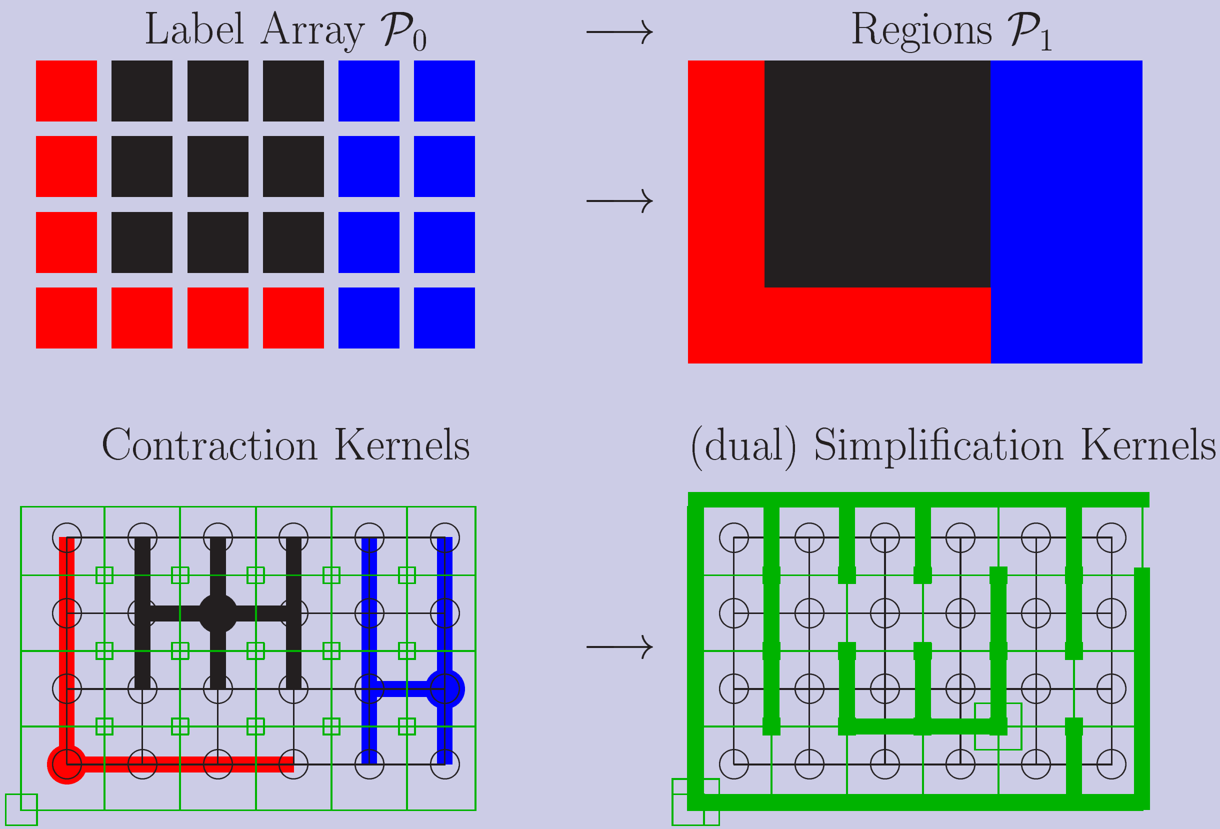
Input: Graphs $(G_0, \overline{G_0})$

while further abstraction is possible **do**
    select contraction kernels
    perform graph contraction and simplification of dual graph (DGC [Kropatsch, 1995a])
    apply reduction functions to compute new reduced level
end while

Output: Graph pyramid – $(G_k, \overline{G_k})$, $0 \leq k \leq h$.

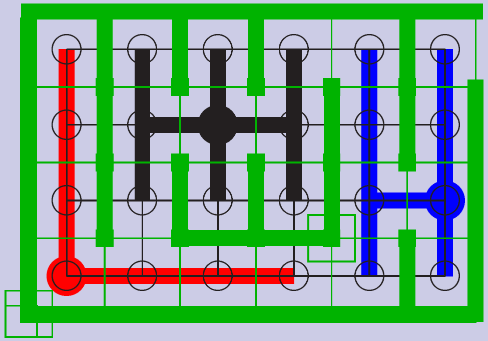
- Graph pyramid is a stack of $(G_k, \overline{G_k})$, $0 \leq k \leq h$

Dual Graph Contraction — Example 1



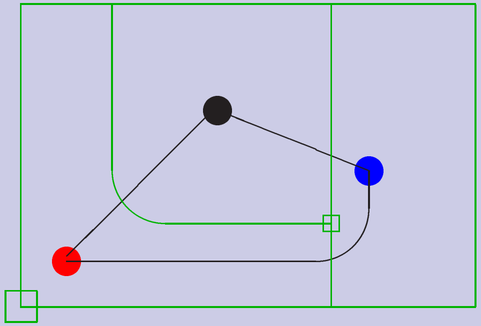
What Remains...

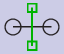
Both Contraction Trees



$\longrightarrow$

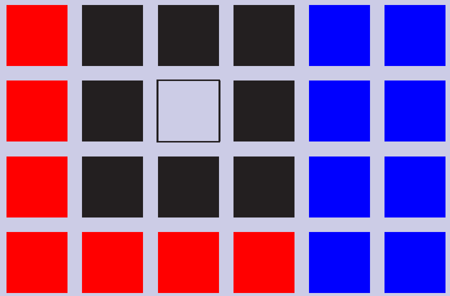
Contraction Result



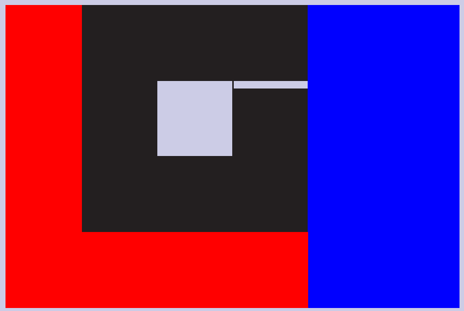
local characterization		1 cell/CC(label)
		$lab(c_1) \neq lab(c_2)$
		≥ 3 regions meet or background
		removal/contraction

Dual Graph Contraction — Example 2

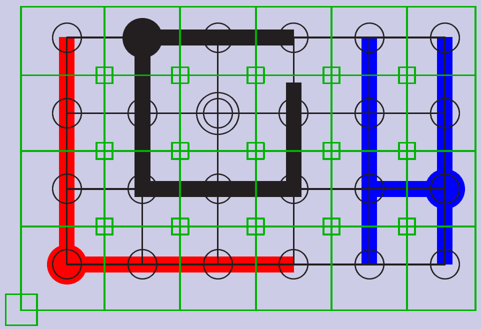
Label Array $\mathcal{P}_0$ $\longrightarrow$ Regions $\mathcal{P}_1$



$\longrightarrow$

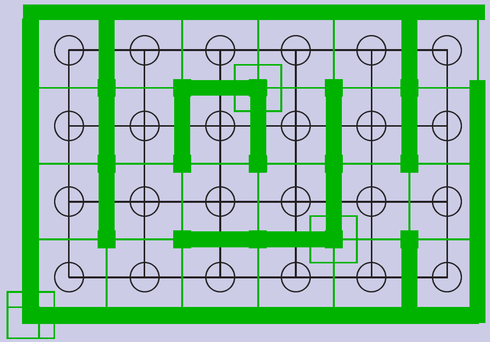


Contraction Kernels



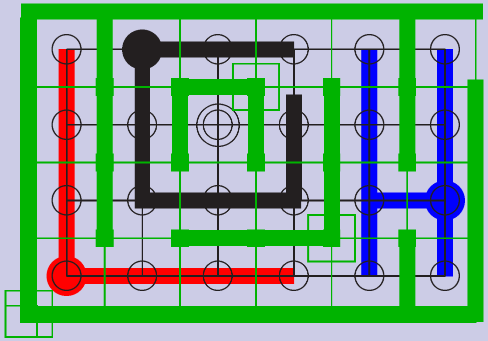
$\longrightarrow$

(dual) Simplification Kernels



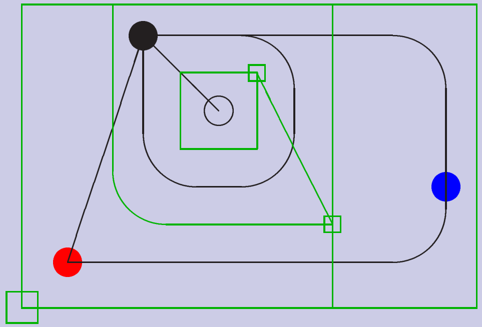
What Remains...

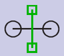
Both Contraction Trees



→

Contraction Result



local characterization	•	1 cell/CC(label)
		$lab(c_1) \neq lab(c_2)$
	except	self-loop, multi-edge with inclusion
	▪	≥ 3 regions meet
		or background
	+	removal/contraction

Topology Preserving Operations in 2D

	Points	Lines	Faces	Cond.	PRE-CONDITION
Euler	# P	$-\#L$	$+\#F$	= const.	to preserve:
Incr.	ΔP	$-\Delta L$	$+\Delta F$	$= 0$	Euler; Orientation
Contract(l, p_0)	-1	-1		(p_1, l, p_0)	$p_1 \neq p_0$;
Remove(l, f_0)		-1	-1	(f_x, l, f_0)	$(f_x \neq f_0)$; $\deg(f_0) \leq 2$
Any Incr.	$(-a$	$-b$	$-c)$		$b = a + c$;
by a contr.	$(-1$	$-1)$		$\times a$	
by c remov.		$(-1$	$-1)$	$\times c$	

A Few Comments

1. $p_0 \neq p_1 \Longrightarrow$ no contraction of self-loop.
2. (f_x, l, f_0): self-loop in dual = bridge in primal, Removal $\Longrightarrow$ Disconnection.
3. removal: l not bridge, since $(f_x, l, f_0 \neq f_x)$ not self-loop
   $\Longrightarrow$ removal does NOT disconnect.
4. $\deg(f_0) > 2 : (f_x, l, f_0, l_i), i = 1, 2, \ldots$: order of (f_x, l_i) may change.
5. Contraction and removal are the ONLY operations needed.
6. Any other topology-preserving operation can be achieved by appropriate combinations of contraction and removals.
7. Note that negative contractions and negative removals are possible as the inverse operations under certain pre-conditions.
8. Pre-conditions for individual operations can be extended to sets of operations: FOREST requirement (no cycle) for contraction.
9. CC: additional pre-condition: $lab(p_1) \neq lab(p_2)$.

What remains after repetition?

Contraction inhibited

↙ ↘

BY CONTENT
(e.g. diff. LABELs)

BY TOPOLOGY
(e.g. DISCONNECT)

↓

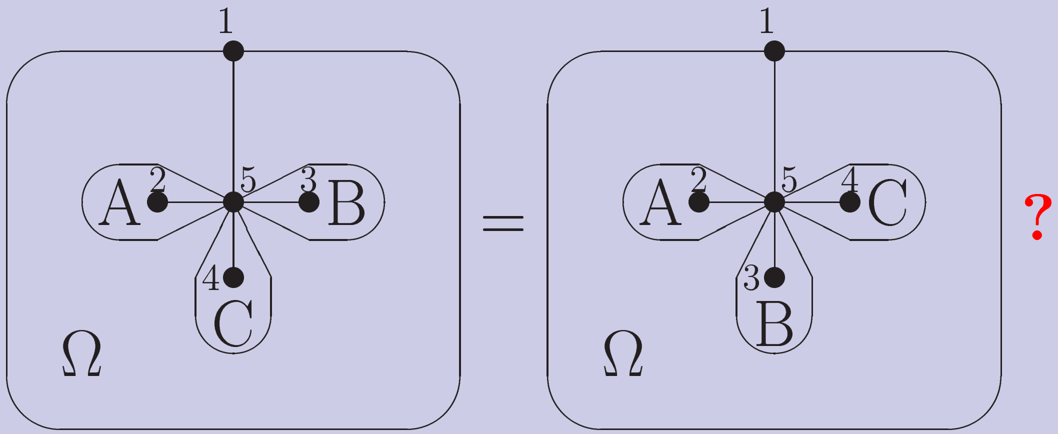
HOLES
(Self-loops, multi-edge)
Betti number: # holes = # fictive edges

↙ ↘

FICTIVE Edges
not UNIQUE!
unified data structure

Removal + Inclusion tree
well defined
separate data structure

Dual Graphs vs. Maps

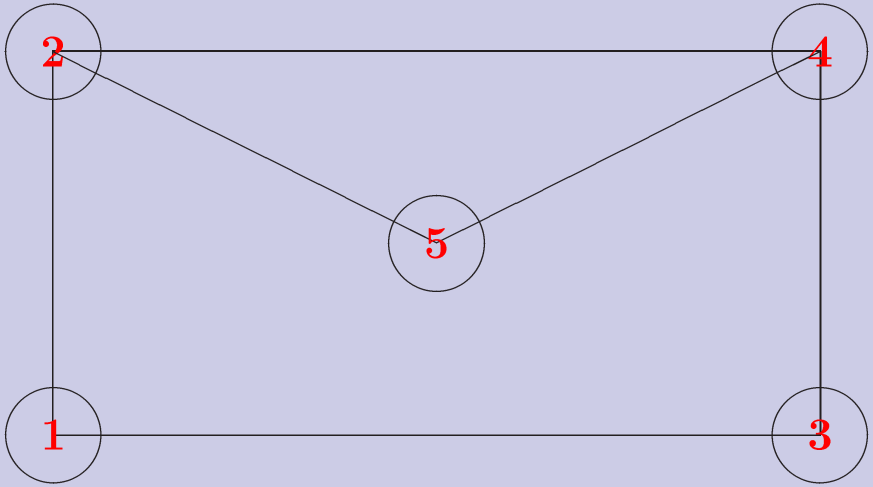


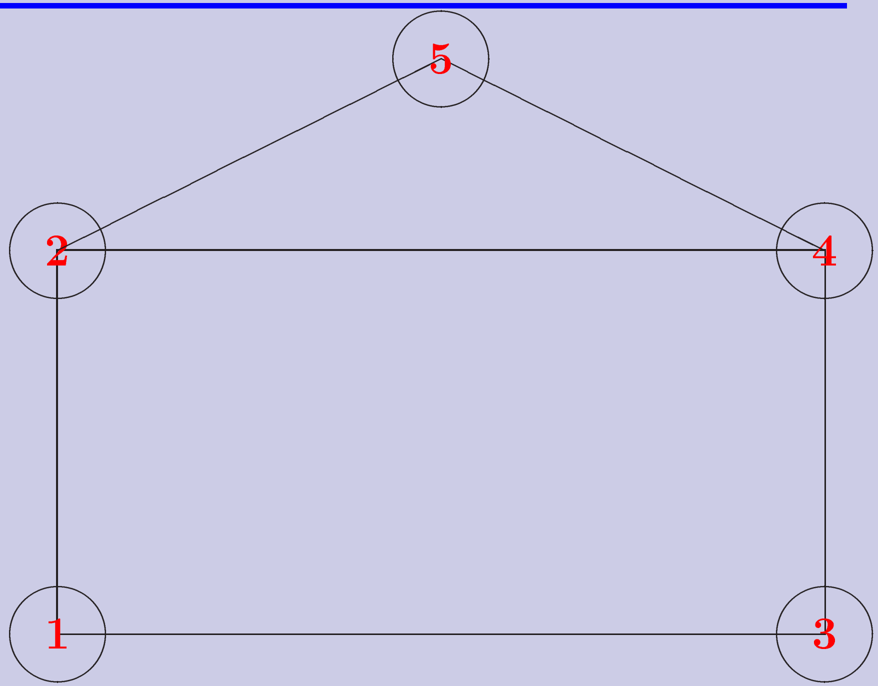
Dual Graphs CANNOT distinguish, MAPS can!

Reason: self-loops loose local orientation:

$$\begin{array}{llll}\{(1, & 1; & \Omega, & \infty), \\ (1, & 5; & \Omega, & \Omega), \\ (5, & 5; & A, & \Omega), \\ (5, & 2; & A, & A), \\ (5, & 5; & C, & \Omega), \\ (5, & 4; & C, & C), \\ (5, & 5; & B, & \Omega), \\ (5, & 3; & B, & B)\}\end{array}$$

Envelope: Graph





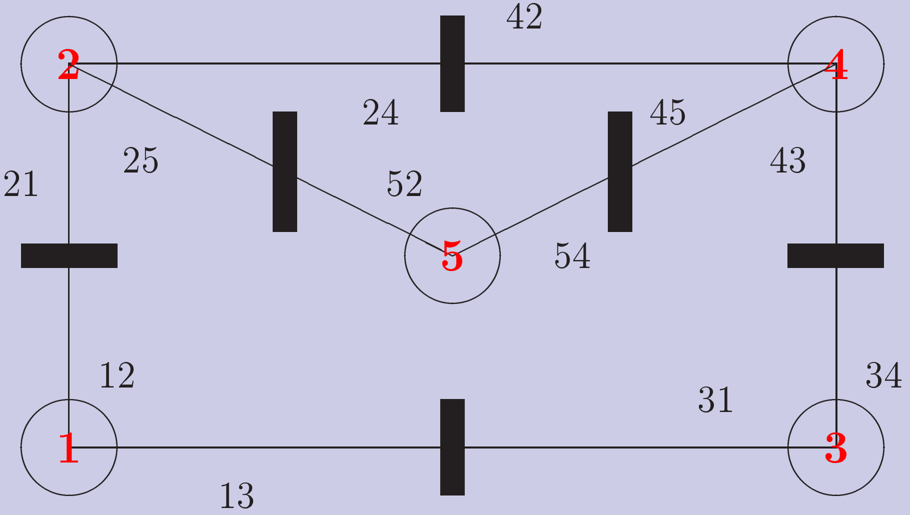
Adjacency Matrix:

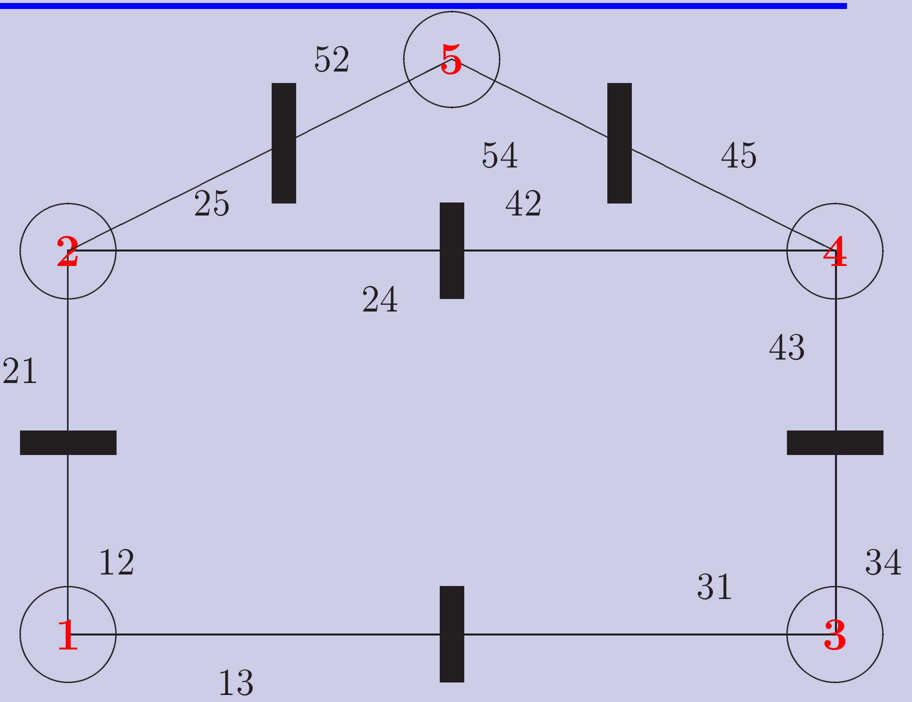
	1	2	3	4	5
1		1	1		
2	1			1	1
3	1			1	
4		1	1		1
5		1		1	

Adjacency Matrix:

	1	2	3	4	5
1		1	1		
2	1			1	1
3	1			1	
4		1	1		1
5		1		1	

Envelope: Combinatorial Map $(\mathcal{D}, \sigma, \alpha)$

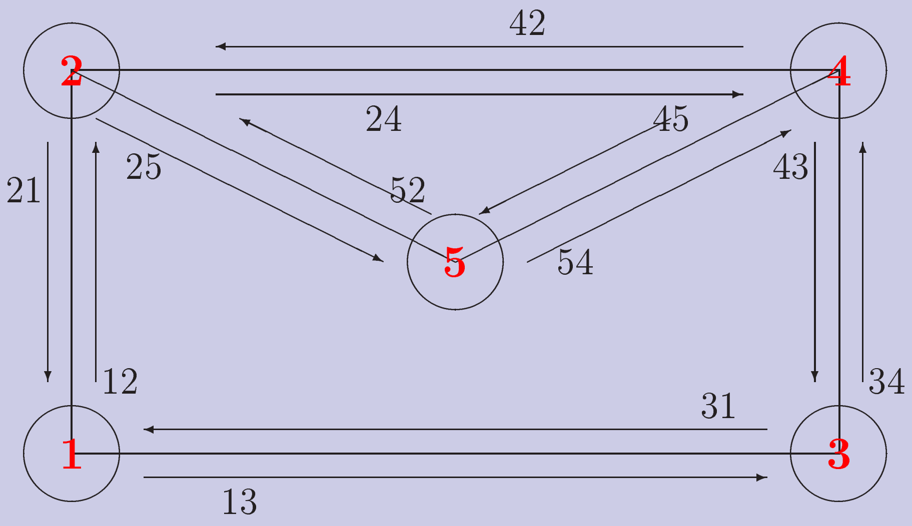


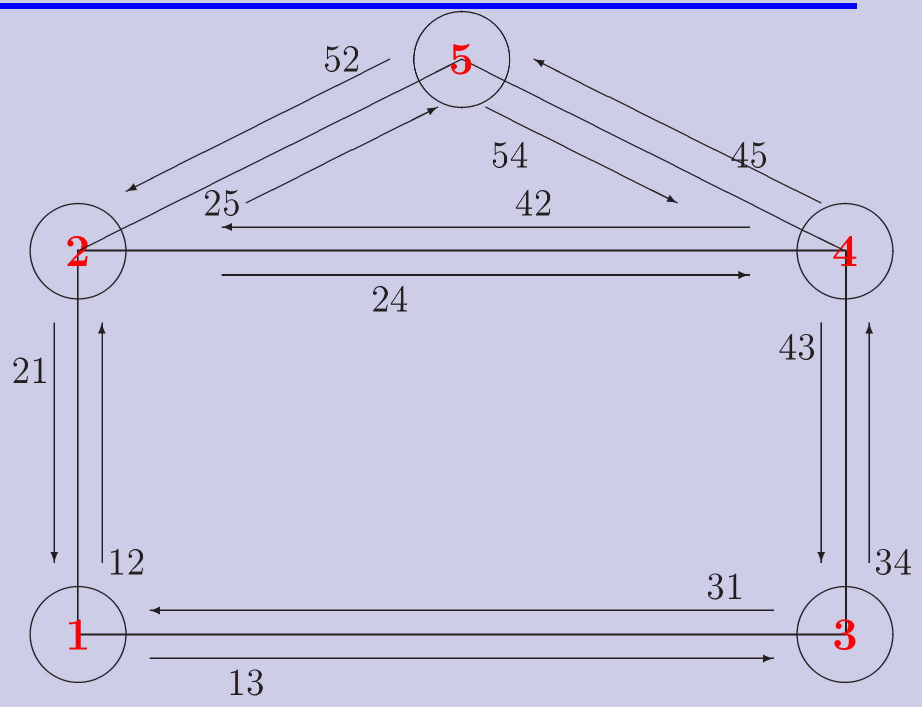


$\mathcal{D}$	12	13	21	24	25	31	34	42	43	45	52	54
α	21	31	12	42	52	13	43	24	34	54	25	45
σ	13	12	25	21	24	34	31	45	42	43	54	52

$\mathcal{D}$	12	13	21	24	25	31	34	42	43	45	52	54
α	21	31	12	42	52	13	43	24	34	54	25	45
σ	13	12	24	25	21	34	31	43	45	42	54	52

Envelope: Combinatorial Map $(\mathcal{D}, \beta_1, \beta_2)$





$\mathcal{D}$	12	13	21	24	25	31	34	42	43	45	52	54
β_2	21	31	12	42	52	13	43	24	34	54	25	45
β_1	13	12	25	21	24	34	31	45	42	43	54	52

$\mathcal{D}$	12	13	21	24	25	31	34	42	43	45	52	54
β_2	21	31	12	42	52	13	43	24	34	54	25	45
β_1	13	12	24	25	21	34	31	43	45	42	54	52

Topology Preserving Operations in 3D

	Pts	Lin.	Fac.	Vol.	Cond.	PRE-CONDITION
Euler	# P	$-\#L$	$+\#F$	$-\#V$	$=$ const.	
Incr.	ΔP	$-\Delta L$	$+\Delta F$	$-\Delta V$	$= 0$	pres.Euler
V-Fusion			-1	-1	(v_1, f, v_2)	$v_1 \neq v_2$? $(\deg f = 2)$
F-Fusion		-1	-1		(f_1, l, f_2)	$f_1 \neq f_2$? $\deg l \leq 2$
L-Fusion	-1	-1			(l_1, p, l_2)	$l_1 \neq l_2$? $\deg p \leq 2$
Any Incr.	$(-a$	$-b$	$-c$	$-d)$	pres.Euler	see below
by V-Fus.			$(-1$	$-1)$	$\times d$	
by F-Fus.		$(-1$	$-1)$		$\times(c-d)$	
by L-Fus.	$(-1$	$-1)$			$\times a$	with $b + c = a + d$

A Few Comments

1. V-Fusion, F-Fusion and L-Fusion are the ONLY operations needed.
2. Any other topology-preserving operation can be achieved by appropriate combinations.
3. Pre-conditions for individual operations can be extended to sets of operations: FOREST requirement (NOTHING INSIDE) and ??.
4. Note that negative Fusions are possible under certain pre-conditions as the inverse operations.
5. Pre-conditions 2 in 3D are not trivial except for faces: a line may delimit more than 2 faces, and a point may be the intersection of more than 2 lines.
6. Note that the L-fusion in 3D eliminates a point along a line sequence whereas the contraction in 2D eliminates a line between two points. As result the 2D contraction produces a point while 3D l-fusion produces a line.
7. CC: additional pre-condition: $lab(v_1) \neq lab(v_2)$, no further pre-condition for F-Fusion and L-Fusion.

What remains after repetition?

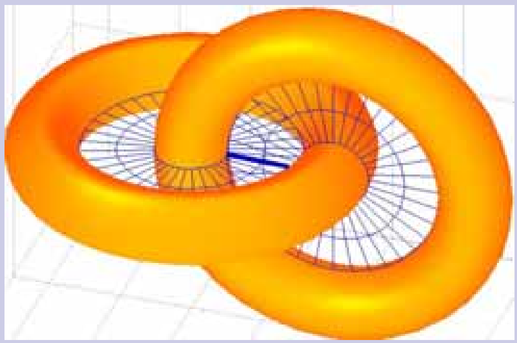
Contraction inhibited

↙ BY CONTENT (e.g. diff. LABELs)

↘ BY TOPOLOGY (e.g. DISCONNECT)

↙ TUNNELS ↓ FICTIVE Elements

↘ **HOLES** ↓ Removal + Inclusion tree



Some useful properties:

1. we need multi-edge and self-loop
2. combine operations (like ECK)
3. repeated contraction, termination criteria
4. pseudo/fictive elements characterize topological relations
   pseudo edge $\longleftrightarrow$ hole, pseudo face $\longleftrightarrow$ tunnel, ...
5. independence of operations allows for:
   - parallelism
   - optimized scan
   - divide and conquer

next: Open problems:

Open problems:

1. 3D, 4D
2. interlaced thori
3. Re-insertion of removed edges/darts (a la wavelet)
4. pre-condition (single OP) $\Longrightarrow$ pre-condition (set of OPs)
5. repeated contraction:
   - different selection criteria
   - different termination criteria
   - different attributes
   - different reduction functions
6. Betti numbers? homology groups? generators?
7. appropriate applications?

Thank you