Construction of Combinatorial Pyramids

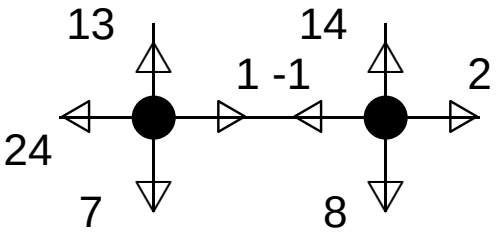
Luc Brun

Walter Kropatsch

Content

- Combinatorial maps,
- Build kernels : Union & Find,
- Encode the pyramid : implicit,
- Conclusion.

Combinatorial maps



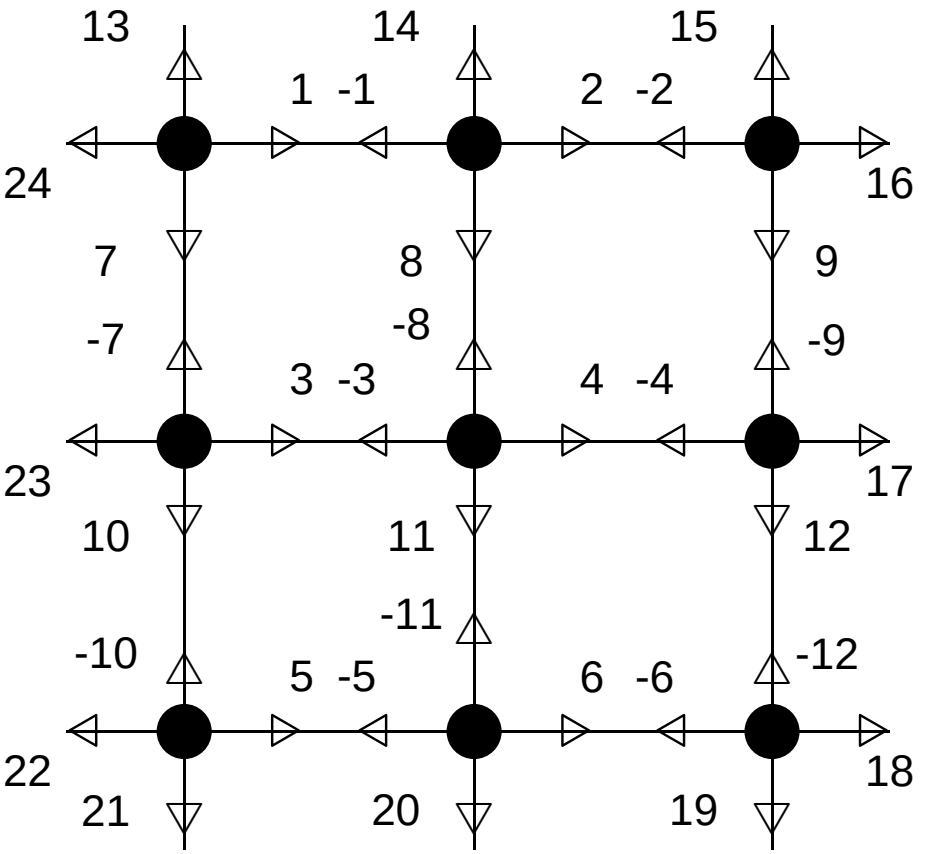
$$\alpha = (1,-1)\ldots$$
$$\sigma = (13,24,7,1)(14,-1,8,2)\ldots$$

$$G = (\mathcal{D}, \sigma, \alpha)$$

$$\sigma = \underbrace{(13,24,7,1)}_{1} \quad \underbrace{(14,-1,8,2)}_{2} \quad \ldots$$
$$\mu: \qquad 1 \qquad\qquad 2$$

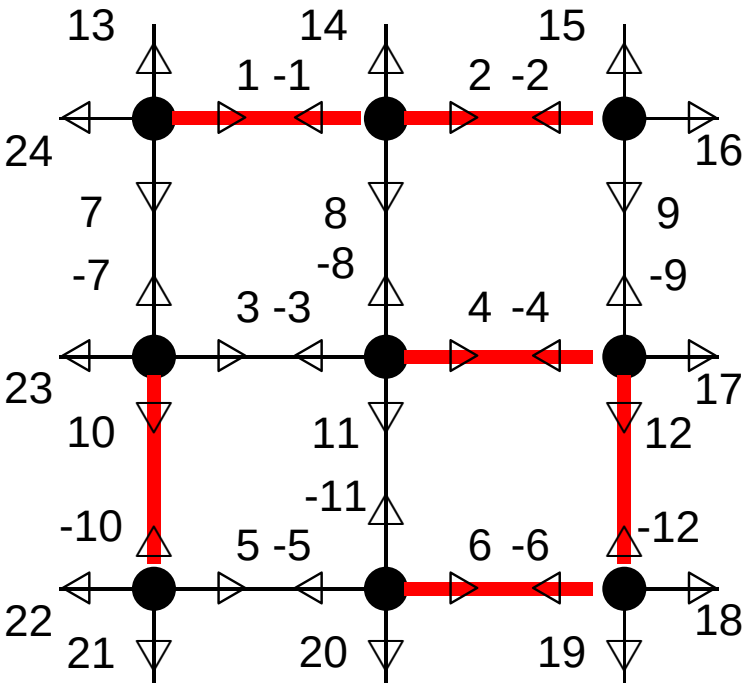
$$\mu(13) = \mu(24) = \mu(7) = \mu(1) = 1$$
$$\mu(14) = \mu(-1) = \mu(8) = \mu(2) = 2$$

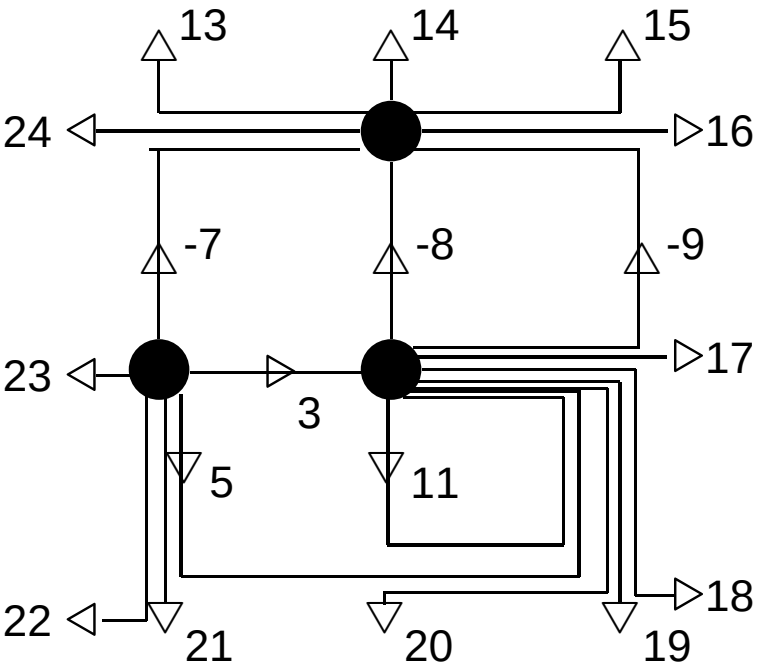
A 3×3 grid



$$G = (\mathcal{D}, \sigma, \alpha)$$

Reduction : Contraction kernel





Definition of a contraction kernel

K : forest of $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$.

- Montanvert 1991,
- Jolion 92, 2001,
- Haximusa 2002,2003

$\rightarrow$ Iterative decimation

$\vdots$

If the function λ s.t.

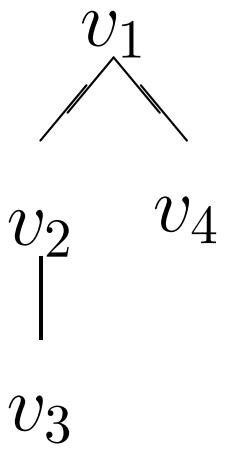
$\lambda(v, v') = 1$ if v and v' should be merged, 0 otherwise.

is known and if few processors available. . .

$\rightarrow$ Union & find.

Union-Find: Data structure

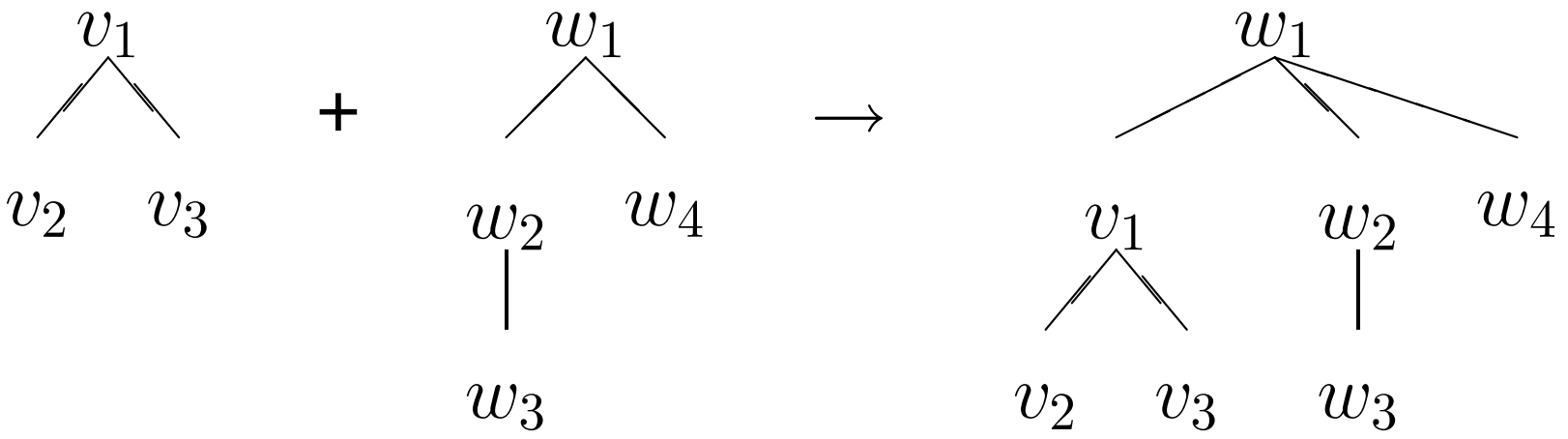
$father$: Array of integers (pointers) encoding the trees.



$$
\begin{aligned}
father(v_3) &= v_2;\\
father(v_2) &= v_1;\\
father(v_4) &= v_1;\\
father(v_1) &= v_1.
\end{aligned}
$$

Union-Find operations

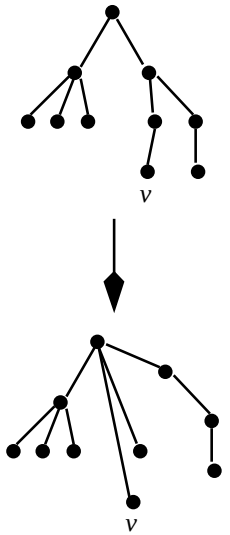
- Union :



$$father(v_1) = w_1$$

Union-Find operations

- Find - Compress:



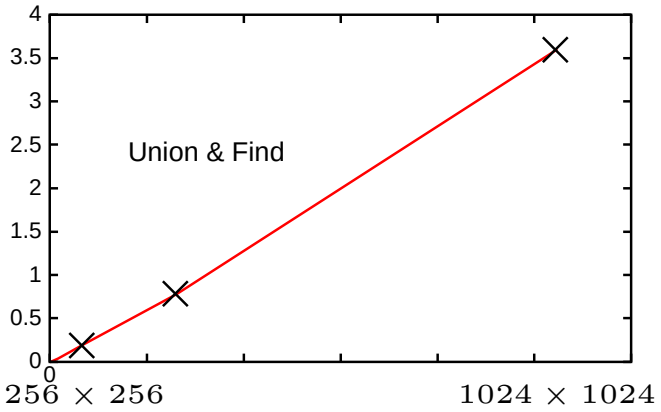

```
1   find_compress(array father,vertex v)
2   {
3       if (father(v) == v )
4           return v;
5       else
6       {
7           father(v) = find_compress(father,father(v))
8           return father(v)
9       }
10  }
```

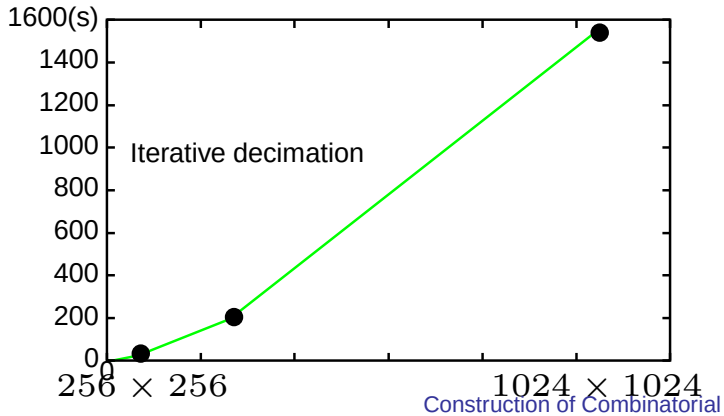
Construction scheme

```
1       contraction kernel K = ∅
2       top of the pyramid G_l = (SD_l, σ_l, α)
3
4       For any vertex v in G_l
5               father(v) = v
6       For any dart b ∈ SD_l
7       {
8           f_1= find_compress(father, μ(b))
9           f_2= find_compress(father, μ(α(b)))
10          if (f_1 != f_2 and λ(f_1, f_2))
11          {
12              K = K ∪ {α*(b)}
13              father(f_1) = f_2
14          }
15      }
16      return K
17
```

Execution Times

execution times	Union & Find	Iterative Decimati
64×64	0.01 s	0.39 s
128×128	0.04 s	3.13 s
256×256	0.19 s	24.65 s
512×512	0.78 s	3 min 14s
1024×1024	3.5 s	25 min 33 s





Which method should we use ?

	Advantages	Drawbacks
Union & Find	Kernels of any depth ■ Fast, ■ “logical levels”.	attachements
Iterative	selection of ■ survivors, ■ attachements.	iterative ■ long processing times, ■ no logical level.

Construction of the pyramid

G_i

$\downarrow$ contraction kernel

G_{i+1}

$\downarrow$ removal kernel

G_{i+2}

$\mathcal{P} = (G_0, G_1 \ldots, G_n)$.

Embedding : receptive field (recursive uses of the parent-child relationship).

A new construction scheme

$\mathcal{P} = (G_0, G_1 \ldots, G_n)$ with $G_i = (\boldsymbol{\mathcal{D}}_i, \sigma_i, \alpha)$ and :

$$\boldsymbol{\mathcal{D}}_n \subset \boldsymbol{\mathcal{D}}_{n-1} \cdots \subset \boldsymbol{\mathcal{D}}_0$$

Two functions :

$$\begin{cases} \forall b \in \boldsymbol{\mathcal{D}}_O & b \in \boldsymbol{\mathcal{D}}_{level(b)-1}, b \notin \boldsymbol{\mathcal{D}}_{level(b)} \\ \forall i \in \{1, \ldots, n\} & state(i) \in \{Contracted, Removed \end{cases}$$

$\rightarrow \mathcal{P} = (G_0, level, state)$

Construction scheme :$\mathcal{P} = (G_0, G_n, level)$

Retrieving features

- Bottom-up

	sequential	parallel
G_i	$\mathcal{O}(\lvert\partial G_i\rvert)$	$\mathcal{O}(\log(\lvert\partial G_i\rvert))$
$(G_1, \ldots, G_n)$	$\mathcal{O}(\lvert\boldsymbol{\mathcal{D}}_0\rvert)$	$\mathcal{O}(\log(\lvert\boldsymbol{\mathcal{D}}_0\rvert))$

- Top-down:

		sequential	parallel
$region$	$\sigma_i^*(b)$	$\mathcal{O}(\lvert R_{\sigma_i^*(b)}\rvert)$	$\mathcal{O}(\log(R_{\sigma_i^*(b)}))$
$border$	b	$\mathcal{O}(\lvert\partial RF_i(b)\rvert)$	$\mathcal{O}(\log(\lvert\partial RF_i(b)\rvert))$

Conclusion

- Contraction Kernel construction :

  iterative vs Union & Find

- Encoding of the pyramid :

  Explicit vs Implicit

C & C++ code :

www.univ-reims.fr/leri/membre/luc/PYRAMIDES/index.html