Geometric Optimization in Grid Topology and Related Combinatorics

Takeshi Tokuyama(Tohoku University)

History (of our research)

- Goes back to 1994 (15 years ago)
- Tetsuo Asano, Naoki Katoh, and I tried to formulate and solve the **image segmentation** problem as a geometric optimization problem
  - A problem in digital geometry.
  - Important in image processing
- This talk recalls the idea and gives recent progress + open problems.

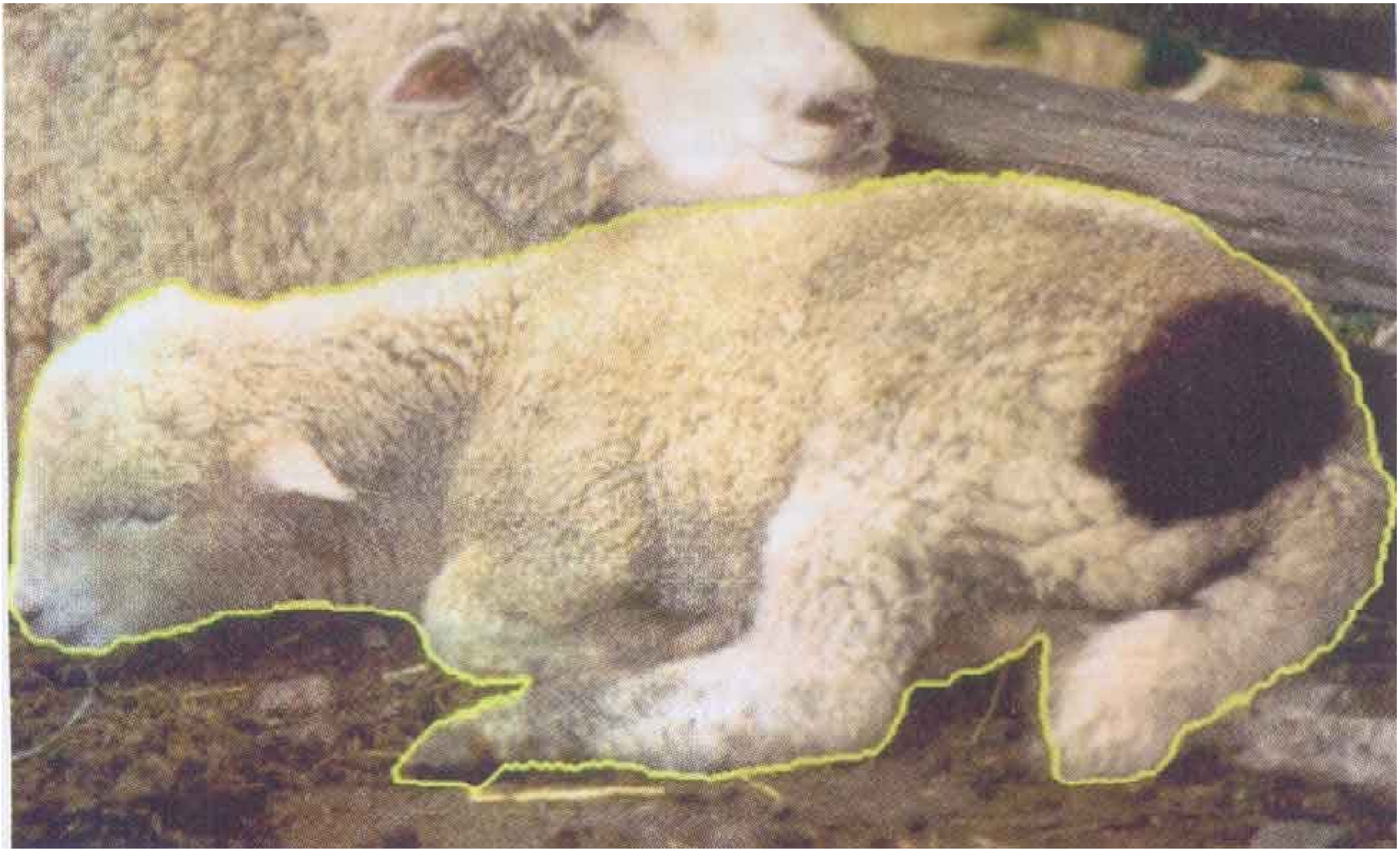
Image segmentation problem

- **G** = n x n pixel grid (for example, n =1024)
- A digital picture is a function **f (x)** on G to represent brightness/color of each pixel x
  - f(x) is real valued (monochromatic picture)
    - In RGB space for color pictures
- **Object image** is a subset S of G to represent an object in the picture.
- Image segmentation: Clip the object image

Popular methods in Image Processing

1. Intelligent Scissors

Trace the boundary with help of human input by mouse



Artificial picture constructed from the segmented images

2. A more automated system: SNAKES

Input initial boundary curve (rubber-band)

The rubber-band shrinks to minimize an energy function

Question: Can we solve the problem as a simple and intuitive combinatorial optimization problem?

Our formulation

- Approximation by two-valued function
  - Picture: function f from G to real values
  - Find the L_2 nearest two-valued function g to f

$$g(x) = a \quad (x \in R)$$

Image

$$g(x) = b \quad (x \notin R)$$

background

Minimize $|| f - g ||_2$

    - a and b become average values of f(x) in R and G-R.

- That is, minimize the intraclass variance

$$Var(R) = \sum_{x \in R} (f(x) - \mu(R))^2 + \sum_{x \in G-R} (f(x) - \mu(G-R))^2$$

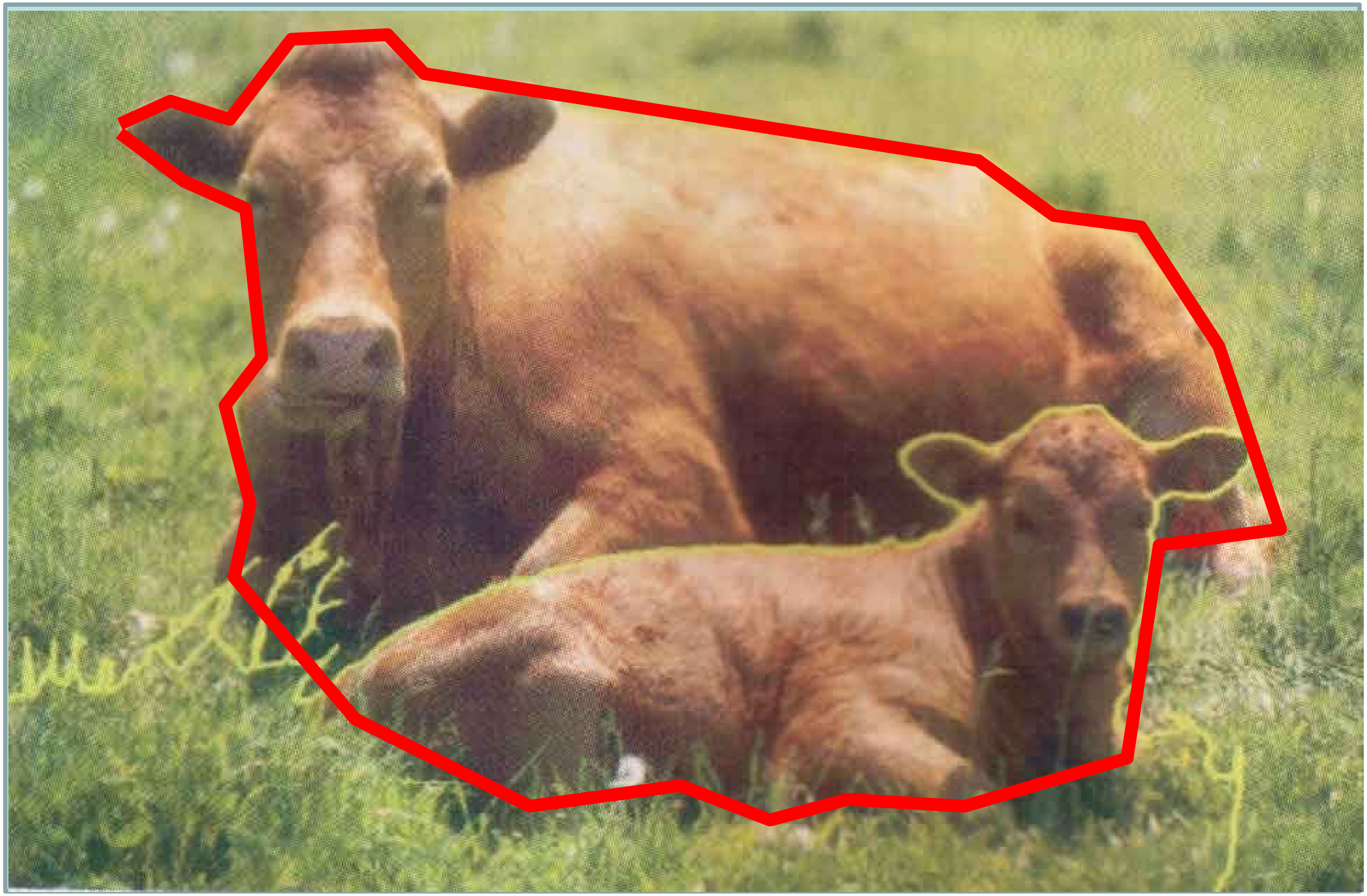
Intraclass variance minimization

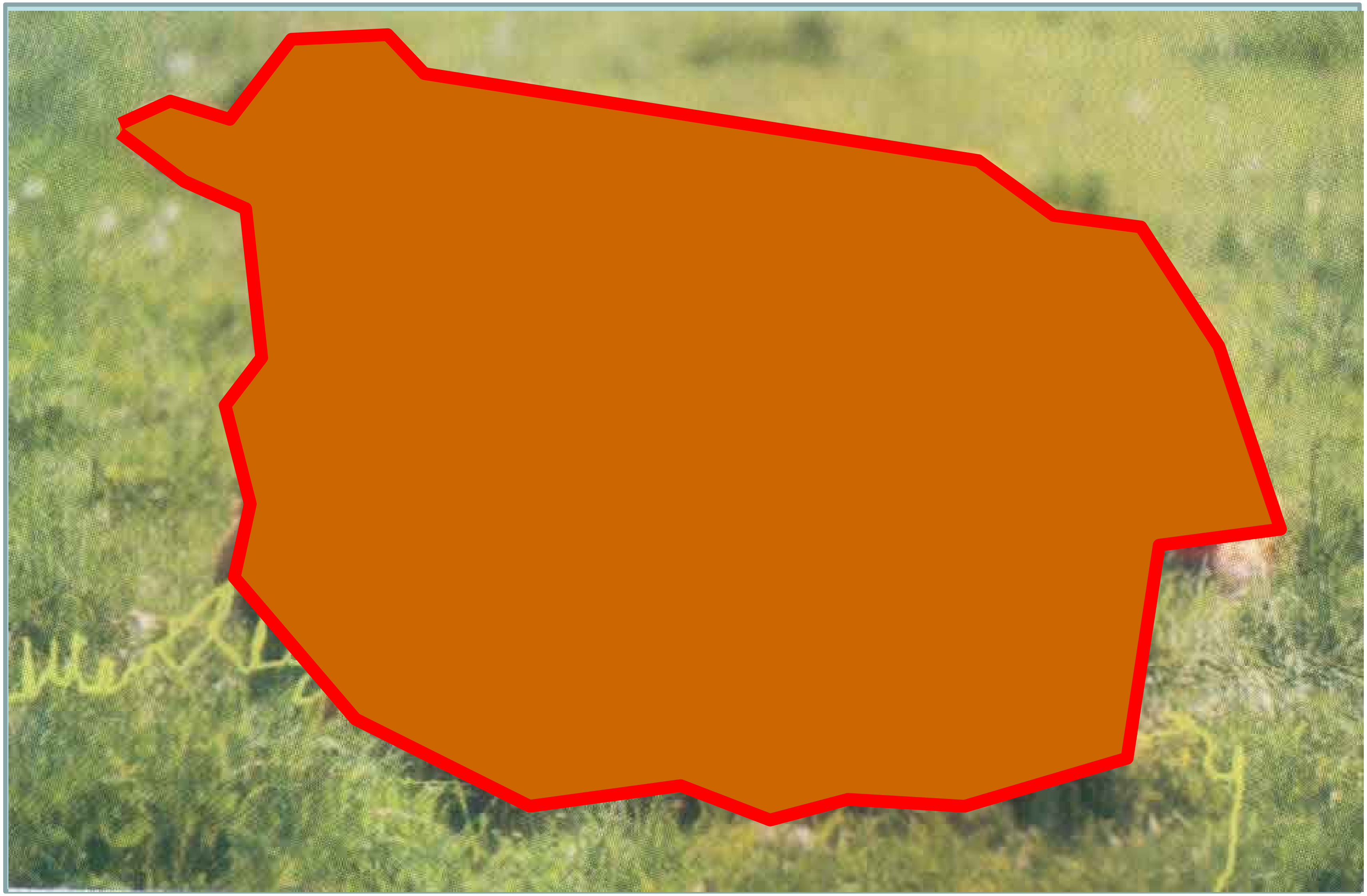
$$Var(R) = \sum_{x \in R} (f(x) - \mu(R))^2 + \sum_{x \in G-R} (f(x) - \mu(G-R))^2$$

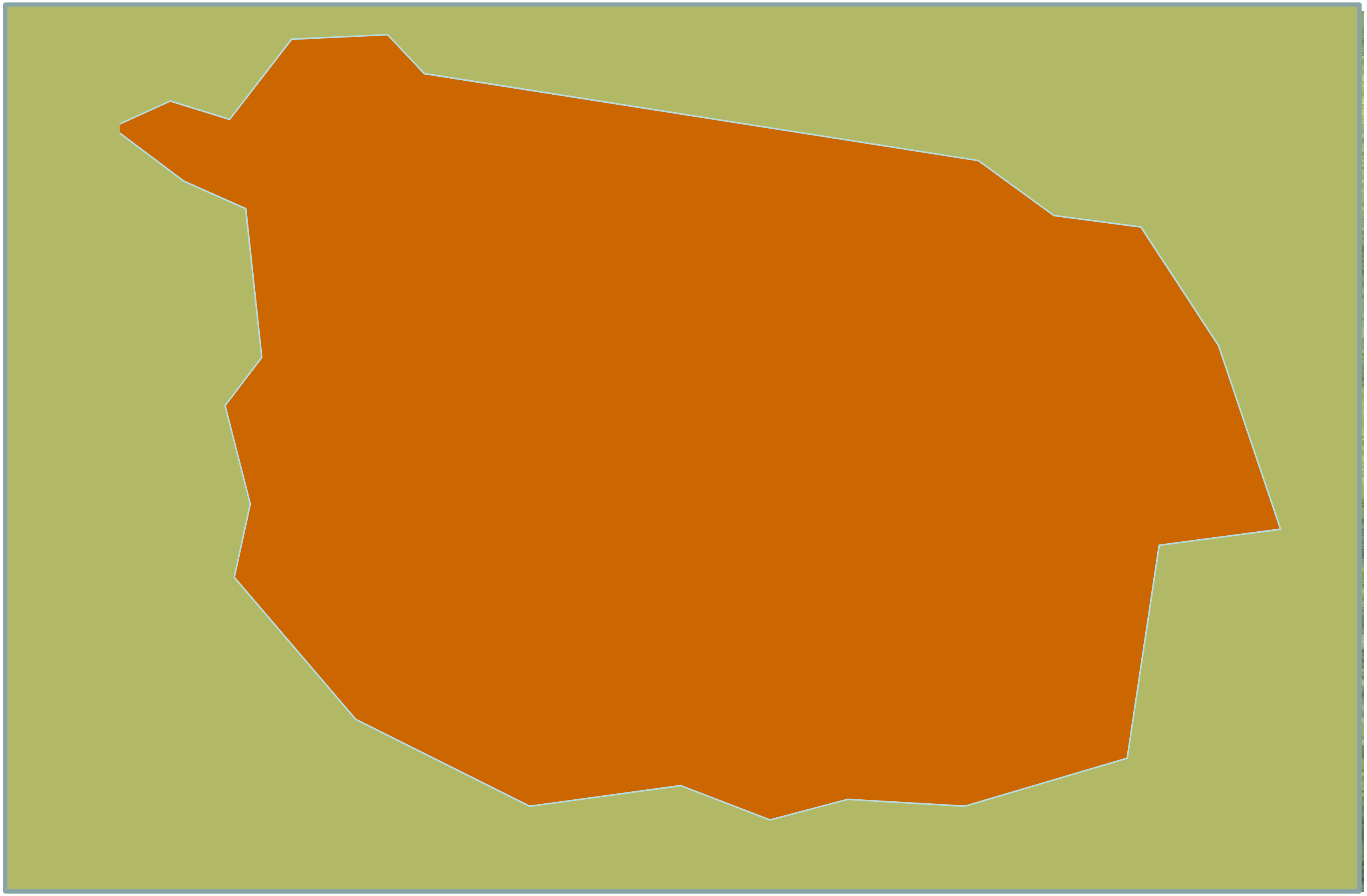
- A kind of 2-center problem.

- Easy if R can be arbitrary (disconnected).
  - Least-square threshold selection (Ohtsu, 1978)
    - Collect pixels brighter than a threshold θ

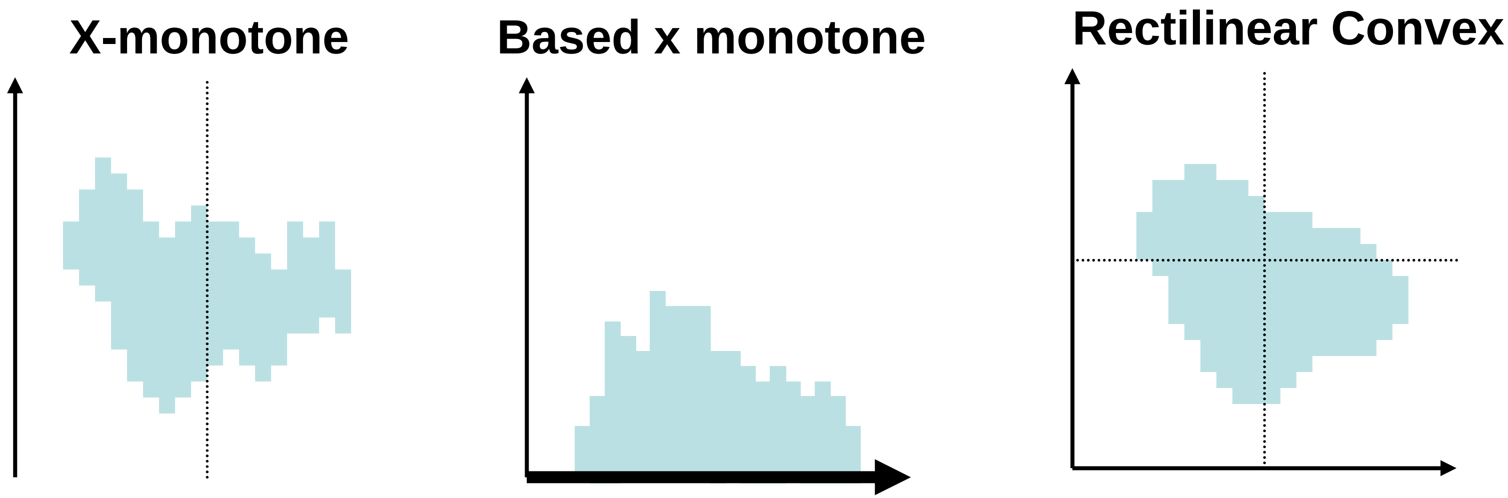
- Reasonable formulation: Consider a family **F** of ***nice*** regions, and find R ∈ **F** minimizing Var (R)







Typical Region Families



X-monotone: Intersection with any vertical line is a segment. (bounded by two x-monotone chains)

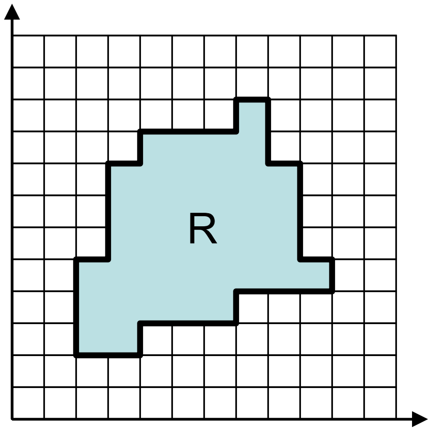
Based (x-)monotone: Region bounded by a monotone chain and a baseline (x-axis)

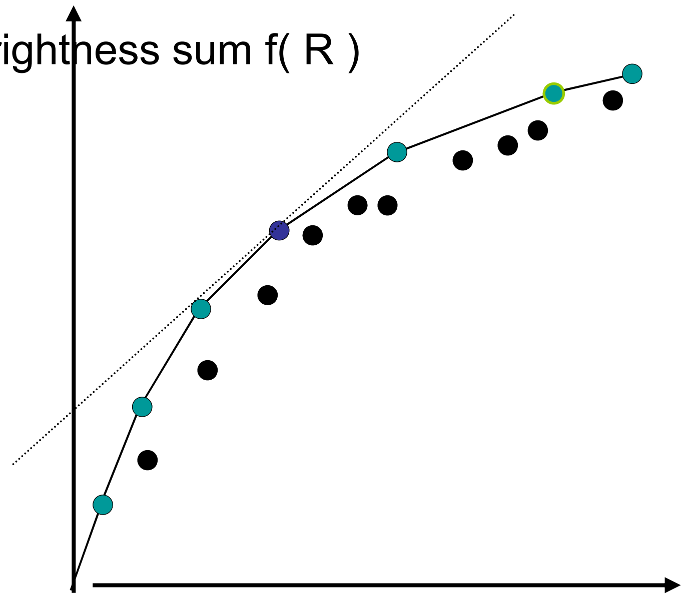
Rectilinear Convex: X-monotone and Y-monotone region.

Solution (Asano-Chen-Katoh-T 96)

- Idea :
  - If we fix the number |R| of pixels in R, Var(R) is minimized if the sum $f(R) = \sum_{p \in R} f(p)$ is maximized (or minimized).
    - To compute such R is a knapsack-type problem, and NP-hard even for the base monotone regions
  - Use parametric method: replace *f(x)* by *f*(x) = f(x) - t*
    - *f (R)* is replaced by *f*(R) = f (R) – t | R |*
  - *Maximization of f* (R) is easier to solve*

The idea for computing the optimal regions





- Consider (k, y(k)) for k=1,2,.. where y(k) = max f (R) s.t. |R|= k.
- Computation of y(k) is NP-hard
- However, we can compute the convex hull of the point set
  - Finding all the tangent points
  - **Maximizing f*(R)** = f (R)– t |R| finds the tangent points with slope t
  - We can find all slopes efficiently
- Var (R) is minimized at a vertex of the convex hull

The problem we need to solve

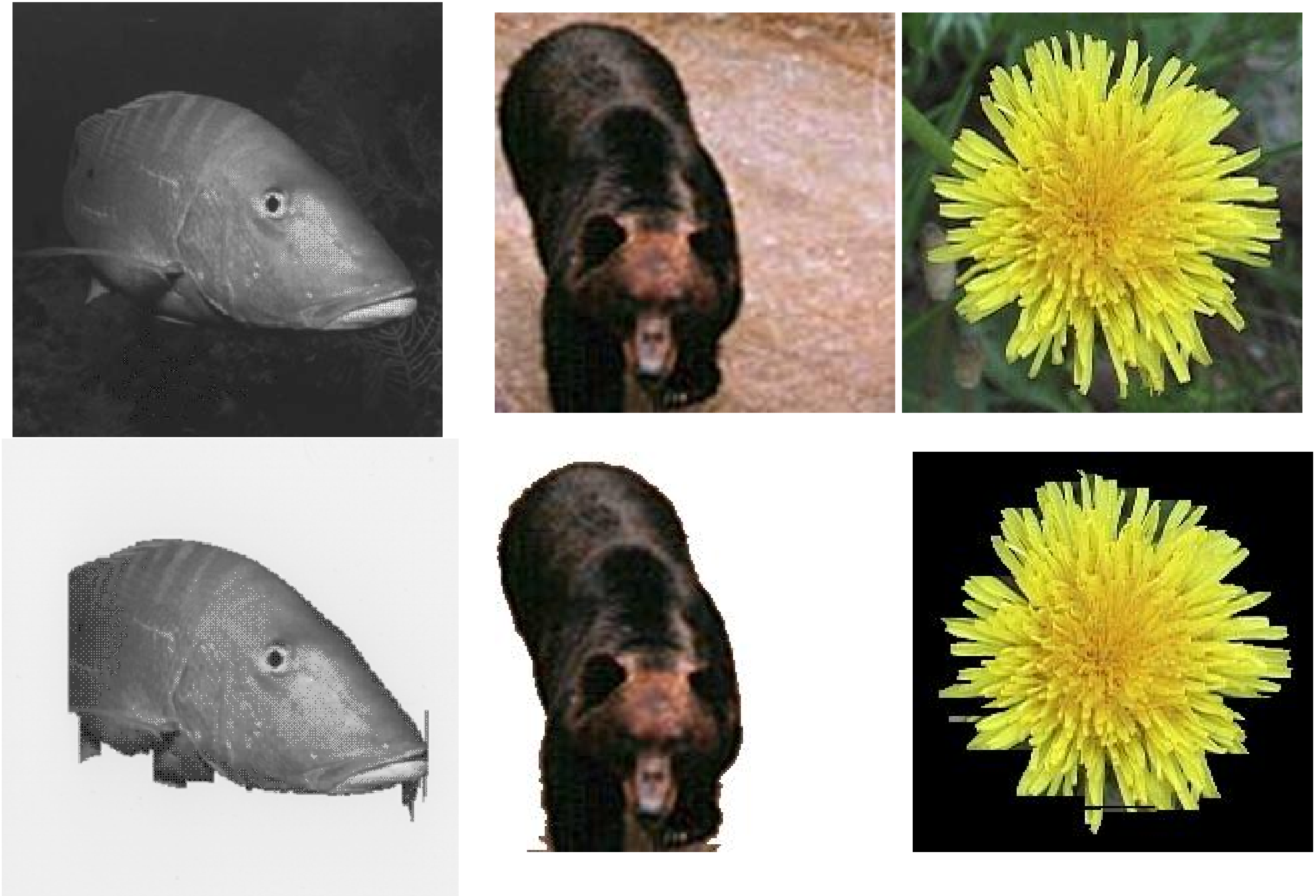
Thus, our image segmentation problem is reduced to the following :

Maximum weight region problem:
Given a function f*(x) on G, find the region R in the region family **F** maximizing f* (R)

Easy to solve if **F** is the family of

- (Connected) x-monotone regions
- Rectilinear convex regions

NP-hard for the family of all connected regions



Lucky to find several unexpected applications and extensions

•Data Mining Application: Optimized Numeric Association Rules (SIGMOD 96 ,VLD96,98, KDD 97)

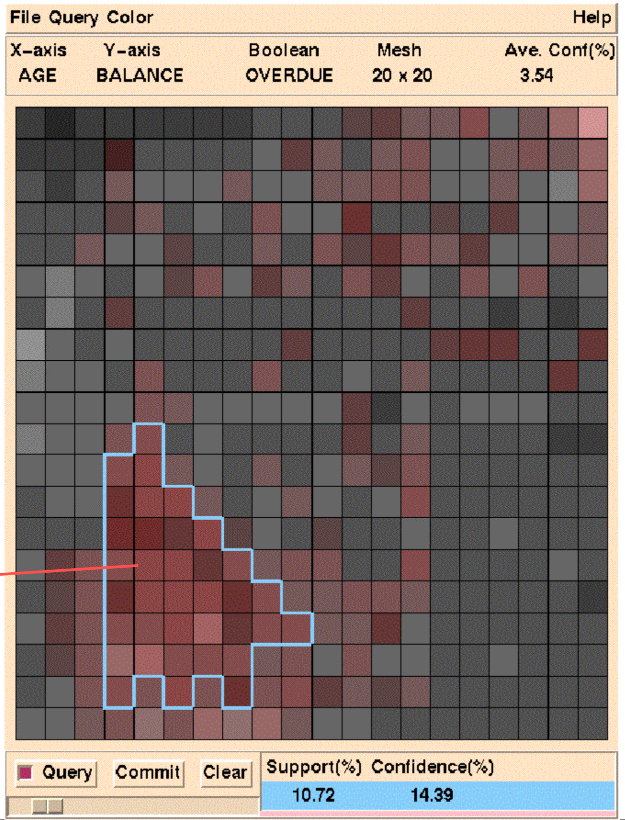
SONAR

(System for Optimized Numeric Association Rules)

Find a rule to detect unreliable customers using a customer database

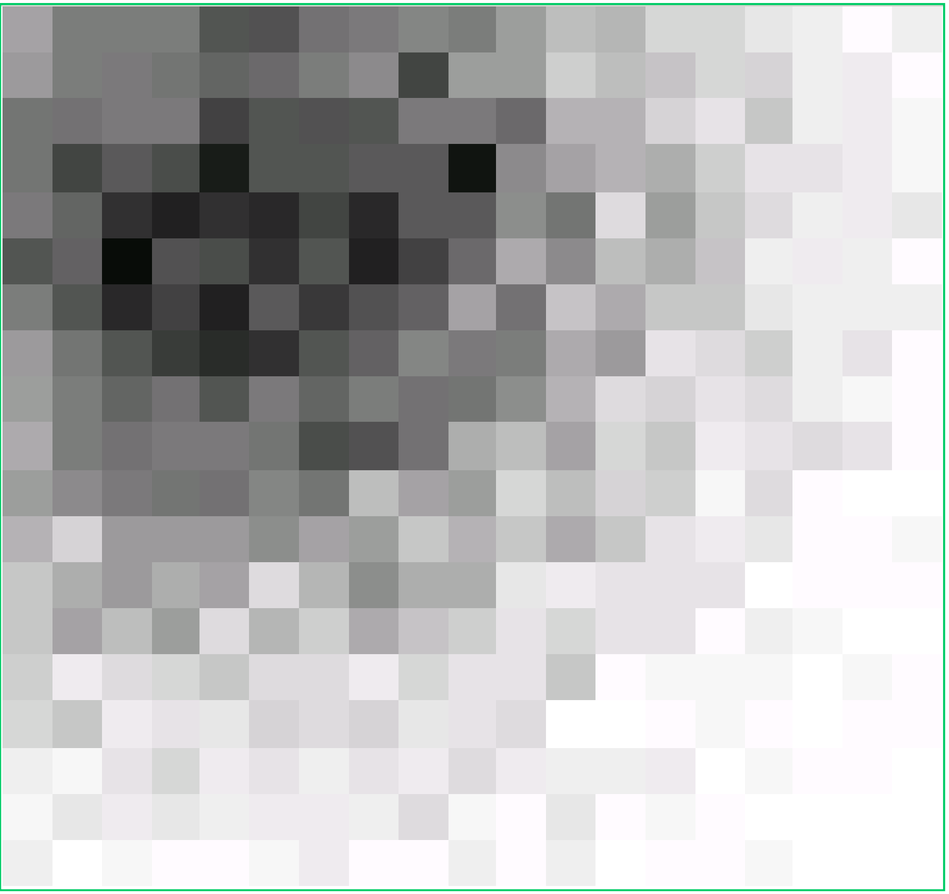
(Age, Balance) $\in$ R

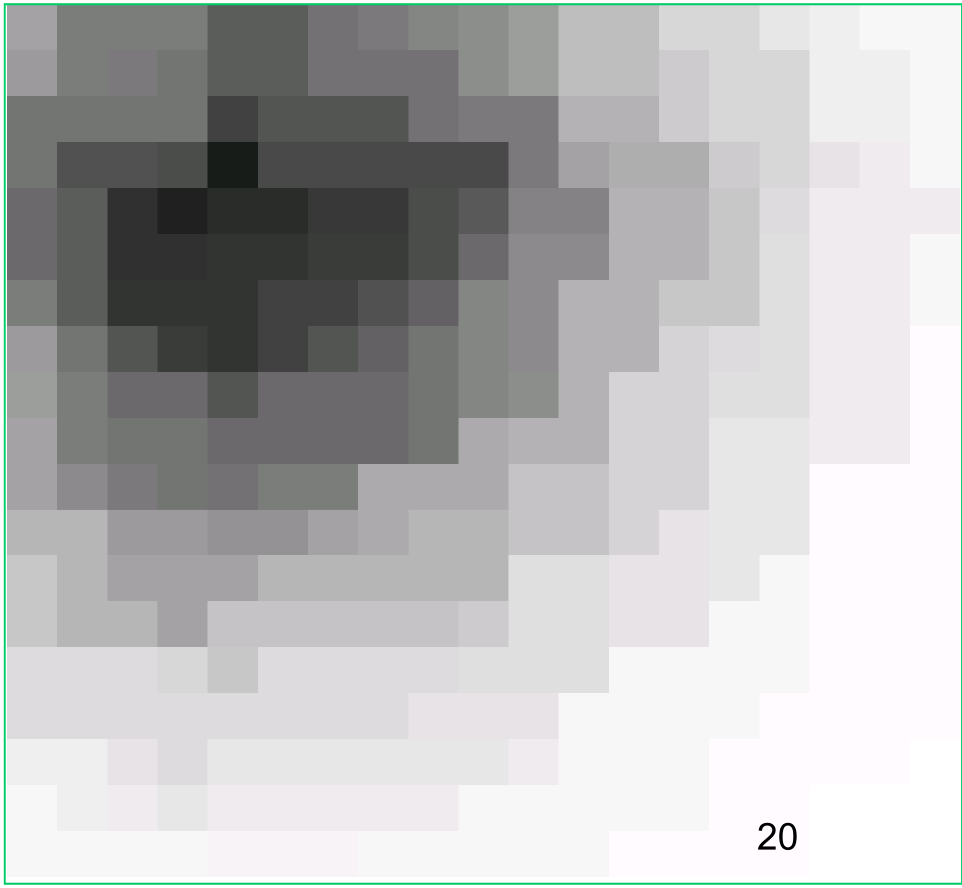
$\Rightarrow$ (CardLoanDelay = yes)



Pyramid approximation and layered rule (Chun-Sadakane-T 03, Chen-Chun-Katoho-T 04)

Instead of two-valued function, we can construct the optimal multilayer function to approximate the input f.



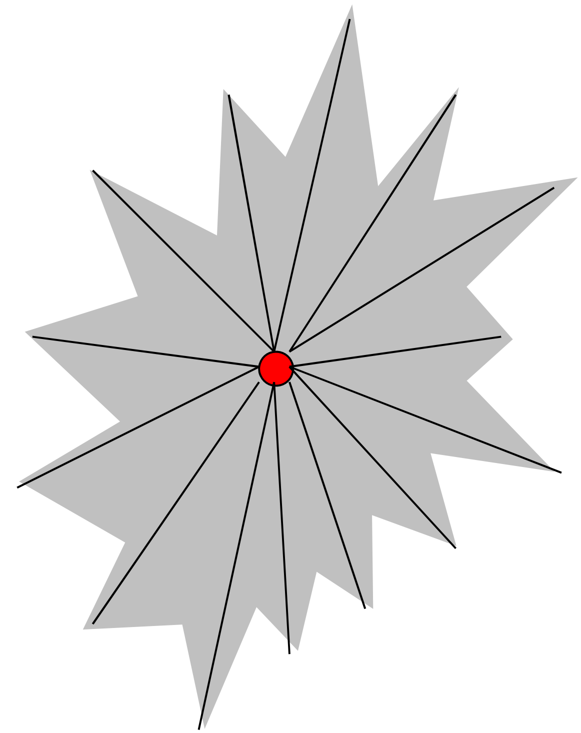


Remained problems

- The region families are in “rectilinear world”
- However, a digital picture should visually simulate the usual (Euclidean) world.
  - Convex region, Star-shaped region
- Segmentation of a region consisting of a few basic shapes.

Mount Fuji taken by NASA.

http://en.wikipedia.org/wiki/Mount_Fuji



Recent progress

Segmentation of

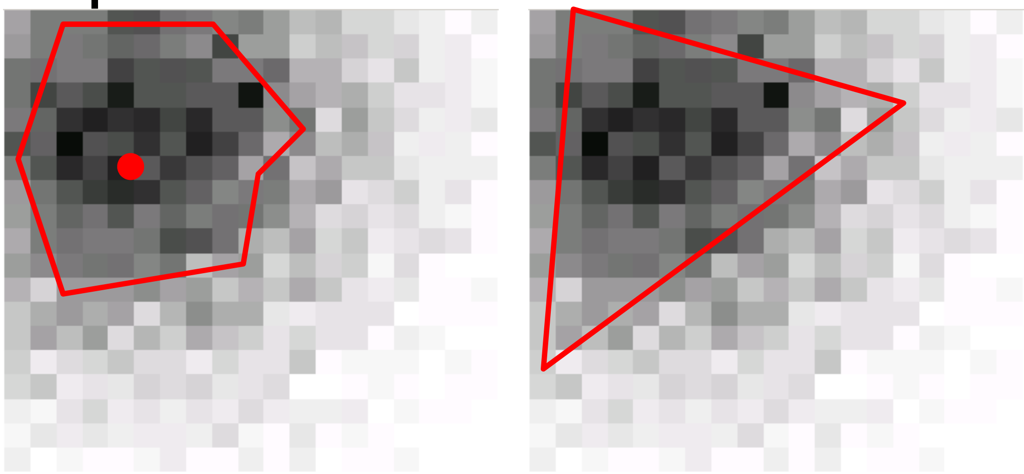
- Star-shaped region
  - Joint work with Jinhee Chun, Matias Korman, and Martin Noellenberg
- Regions decomposable into a few basic regions
  - Joint work with Jinhee Chun, Ryosei Kasai, and Matias Korman

Segmentation of a star-shape

First idea:

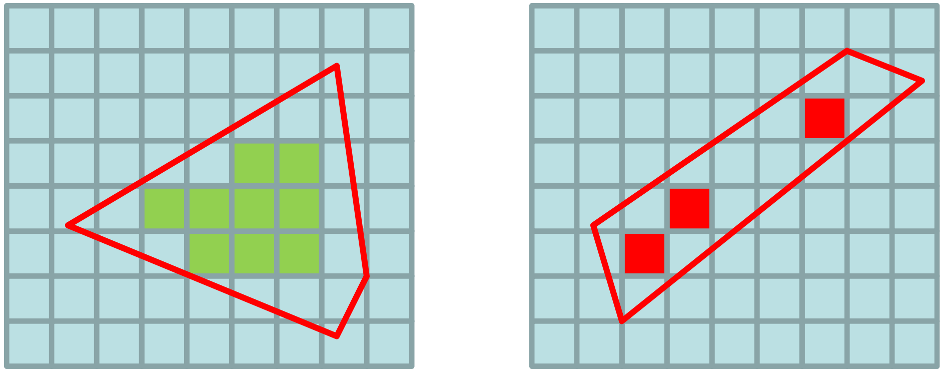
- Consider a real star-shaped region P that has a pixel o as its center.
- Minimize/maximize the measure of P with respect to the pixel distribution f* (x)

- Unfortunately, this looks very difficult
  - Complicated even if we P is a triangle



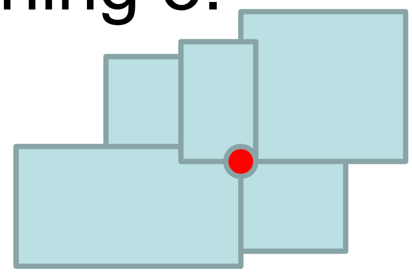
Segmentation of a digital star-shape

- Second idea:
  - Define the “digital star-shape region” as the set of pixels inside a real star-shaped region
  - Unfortunately, I have no idea how to efficiently find such P maximizing f* (P)

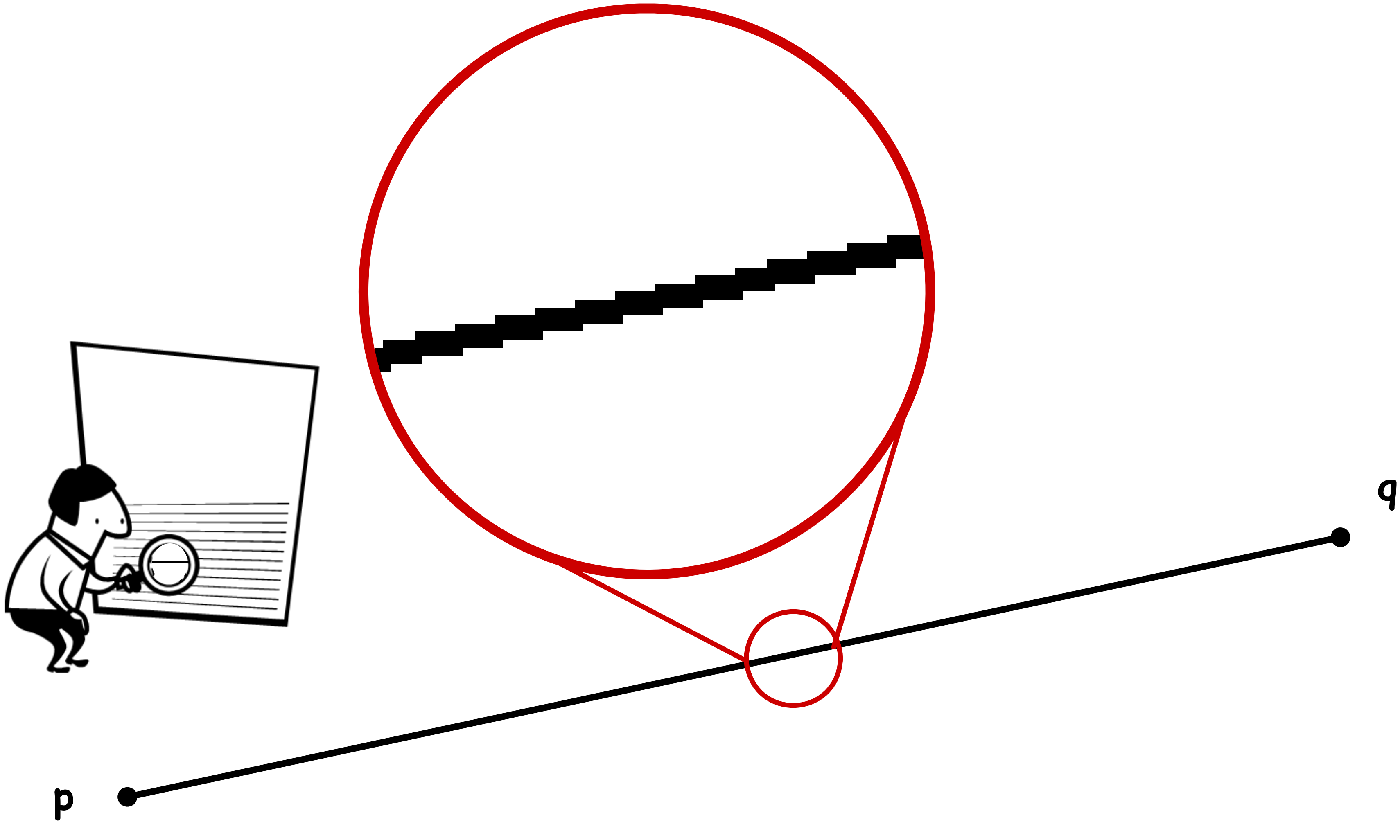


Definition of digital star shape

- Third idea (our choice)
- Give a definition of digital star-shape analogously to the Euclidean star-shape
  - For any p in P, the **digital ray** dig(op) from o to p is in P
- Problem: What is the "digital ray?"
  - If P contains **all** shortest paths from o to p, we have a union of rectangles containing o.
    - Staircase convex region
    - Too "fat" as a star-shape



Digital line

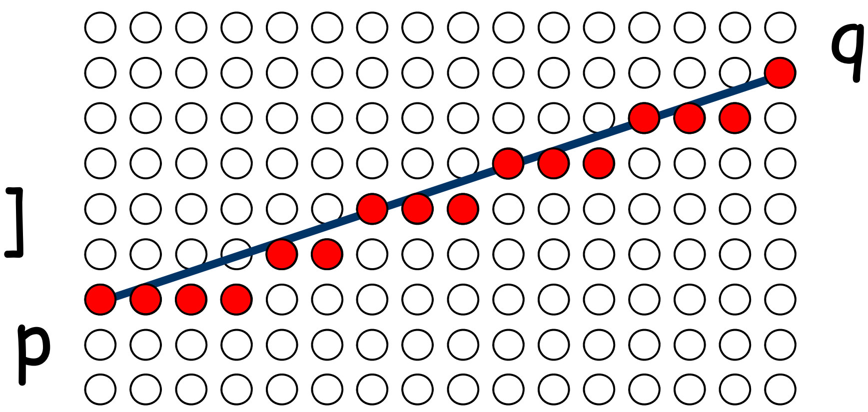


Digital Straight Segment

- Digital line segment
  - Many different formulations to define a line in the digital plane, started in (at latest) 1950s.
  - A popular definition: DSS (Digital straight segment)

line : $y=ax+b$

digital line : $y=[ax+b]$

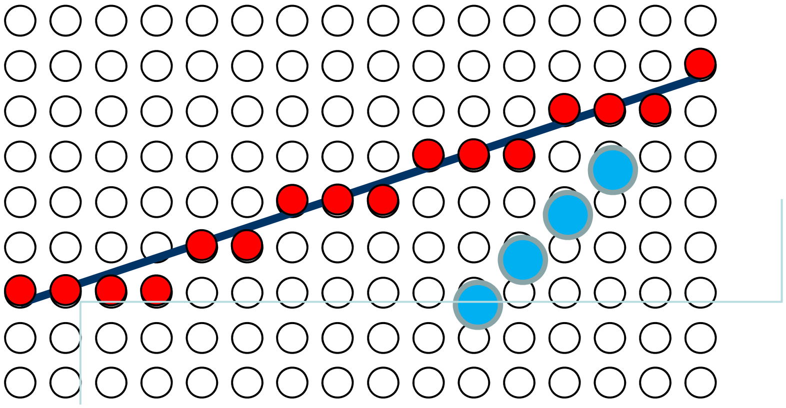


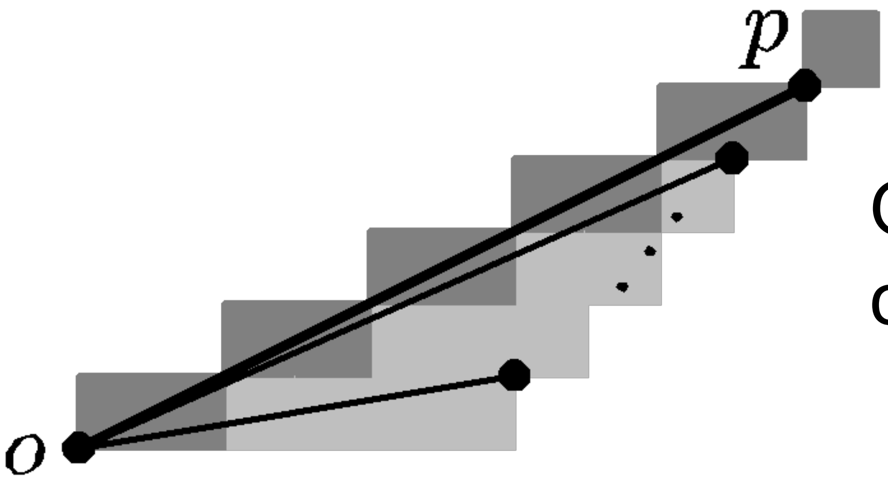
Digital Straight Segment

- DSS is not star-shaped

line : y=ax+b

digital line : y=[ax+b]





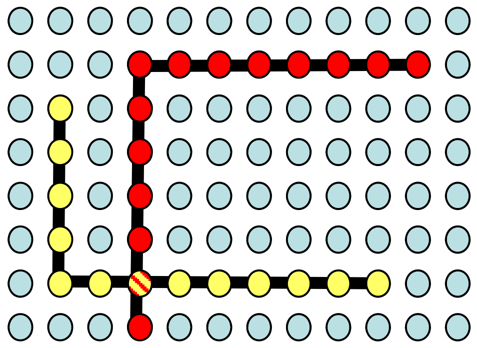
Only fat star-shapes can be obtained

Axioms for consistent digital line segment

- **(s1)** A digital line segment dig(pq) is a connected path between p and q under the grid topology.
  (connectivity)
- **(s2)** There exists a unique dig(pq)=dig(qp) between any two grid points p and q.
  (existence)
- **(s3)** If $s,t \in$ dig(pq), then dig(st) $\subseteq$ dig(pq).
  (consistency)
- **(s4)** For any dig(pq) there is a grid point $r \notin$ dig(pq) such that dig(pq) $\subset$ dig(pr).
  (extensibility)

Consistent digital segments

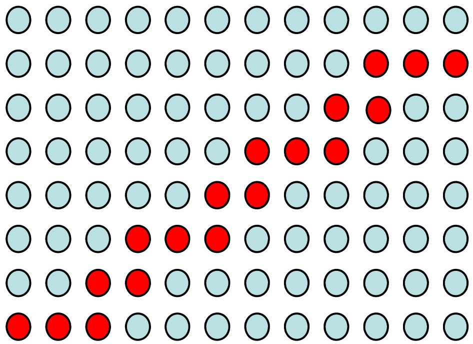
- DSS is not consistent
  - Intersection of two digital segments is not always connected



- Known consistent digital segments
  - L- path system
  - Defect of the L-path system
    - Does not approximate line segments visually
    - Hausdorff distance from real line is O(n)
- L-path system is not suitable to define visually nice digital star-shape regions.

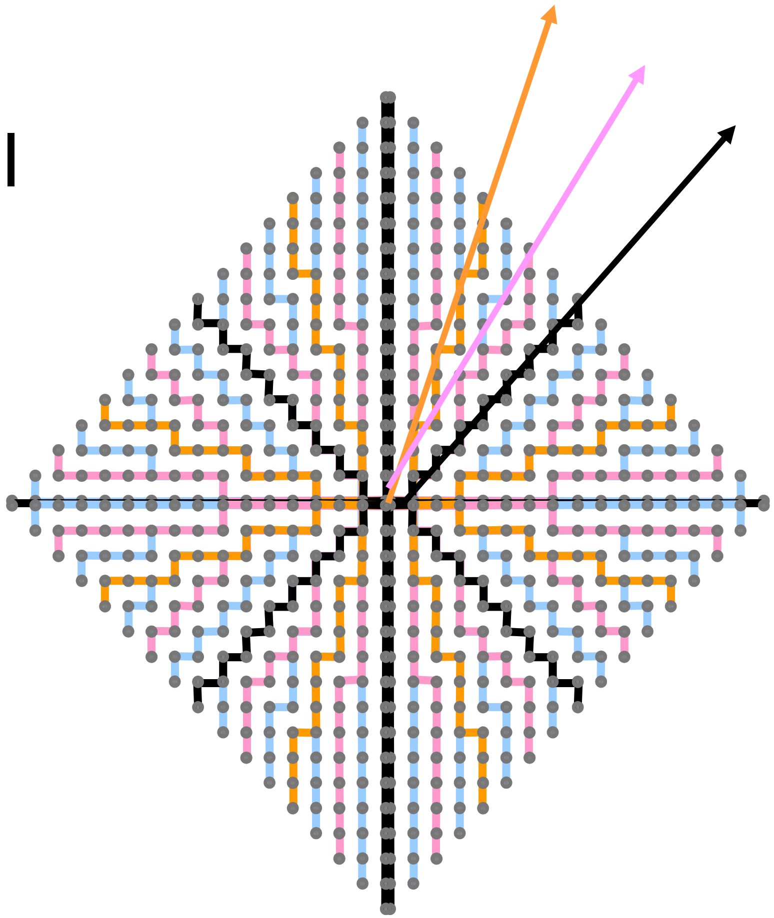
Digital segments vs rays

- We need a visually nice consistent digital segments
- But, this is a big challenge (more than 50 years)
  - No system of consistent digital segments with o(n) Hausdorff distance error is known (Impossible ?)
- Hopeless approach again??
- But we only need RAYS from the center to define the digital star-shapes
  - Isn’t this easier ?
    - Yes, it is easier.
  - How easy it is ?



The consistent digital rays

- Consider a system of digital rays, that are digital line segments emanated from the origin o
- Satisfying axioms
- Approximating real straight rays

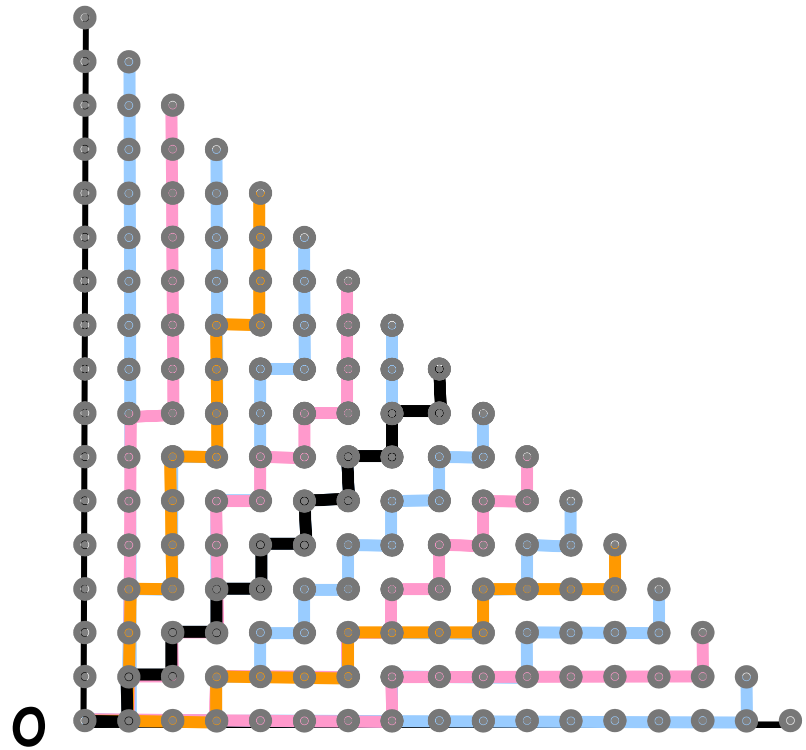


Axioms for digital ray

- **(R1)** A digital ray dig(op) is a connected path between o and p. (connectivity)
- **(R2)** There is a unique digital ray dig(op) between o and any grid point p. (existence)
- **(R3)** If $r \in$ dig(op), then dig(or) $\subseteq$ dig(op). (consistency)
- **(R4)** For any dig(op), there is a grid point $r \notin$ dig(op) such that dig(op) $\subset$ dig(or). (extensibility)
- **(R5)** For any $r \in$ dig(op), $|\overline{or}| \leq |\overline{op}|$ (monotonicity)

Digital rays form a tree

- The union of all digital rays form an infinite spanning tree T of G
  - dig(op) is the unique path between o and p in the tree.



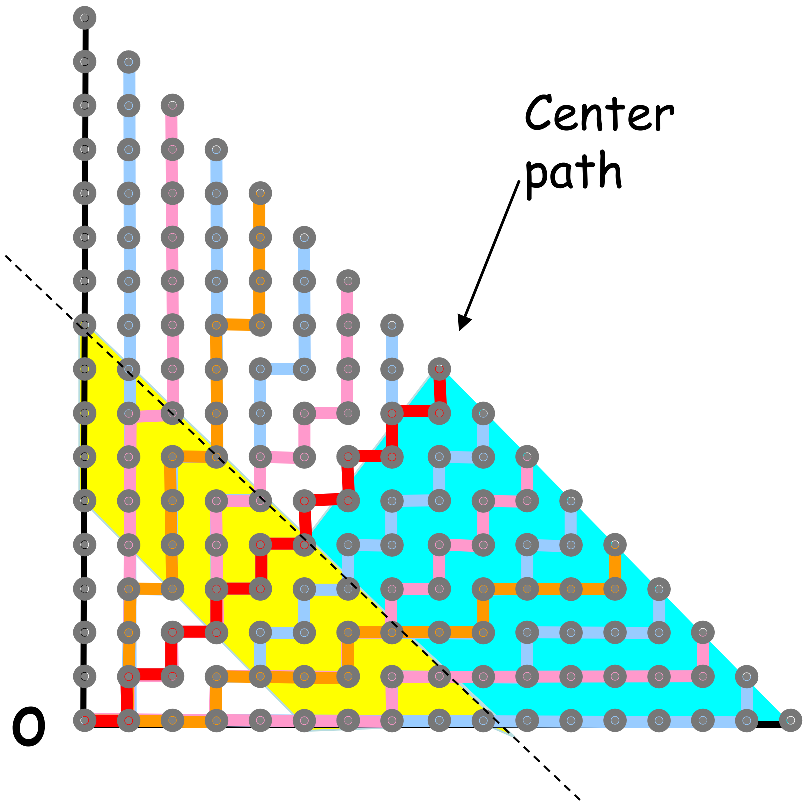
- **(R3)** If $r \in dig(op)$, then $dig(or) \subseteq dig(op)$.
- **(R4)** For any dig(op), there is a grid point $r \notin dig(op)$ such that $dig(op) \subset dig(or)$.

The distance bounds

- $\Theta(\log n)$ bound for the Hausdorff distance between digital ray and the corresponding real ray.
  - The construction gives the upper bound
  - The lower bound comes from the **discrepancy theory**
- The same distance bound holds for the digital star-shaped regions.
- The same bound holds in d-dimensional grid

Upper bound construction

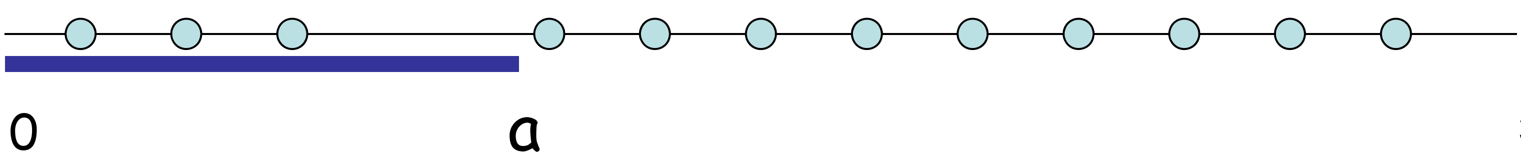
- Construct the center path
- Recursively construct the two parts divided by the center path, copying the structure of size n/2



Lower bound comes from Discrepancy of sequence

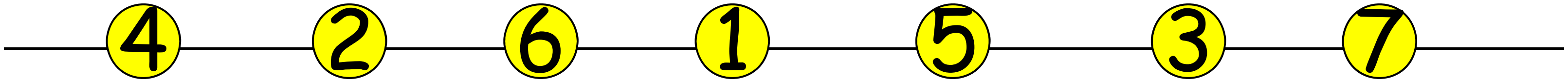
- Consider a sequence $a_1, a_2, a_3,\ldots\ldots a_m \ldots$in the interval [0,1] such that each prefix sequence gives a (nearly) uniform distribution
- Discrepancy
  - Difference between the number $X_m(a)$ of elements in [0,a] in $a_1, a_2, a_3,\ldots\ldots a_m$ and am (expected number for the ideal uniform distribution)

$$\max_{m<n} \max_{0<a<1} | X_m(a) - am |$$

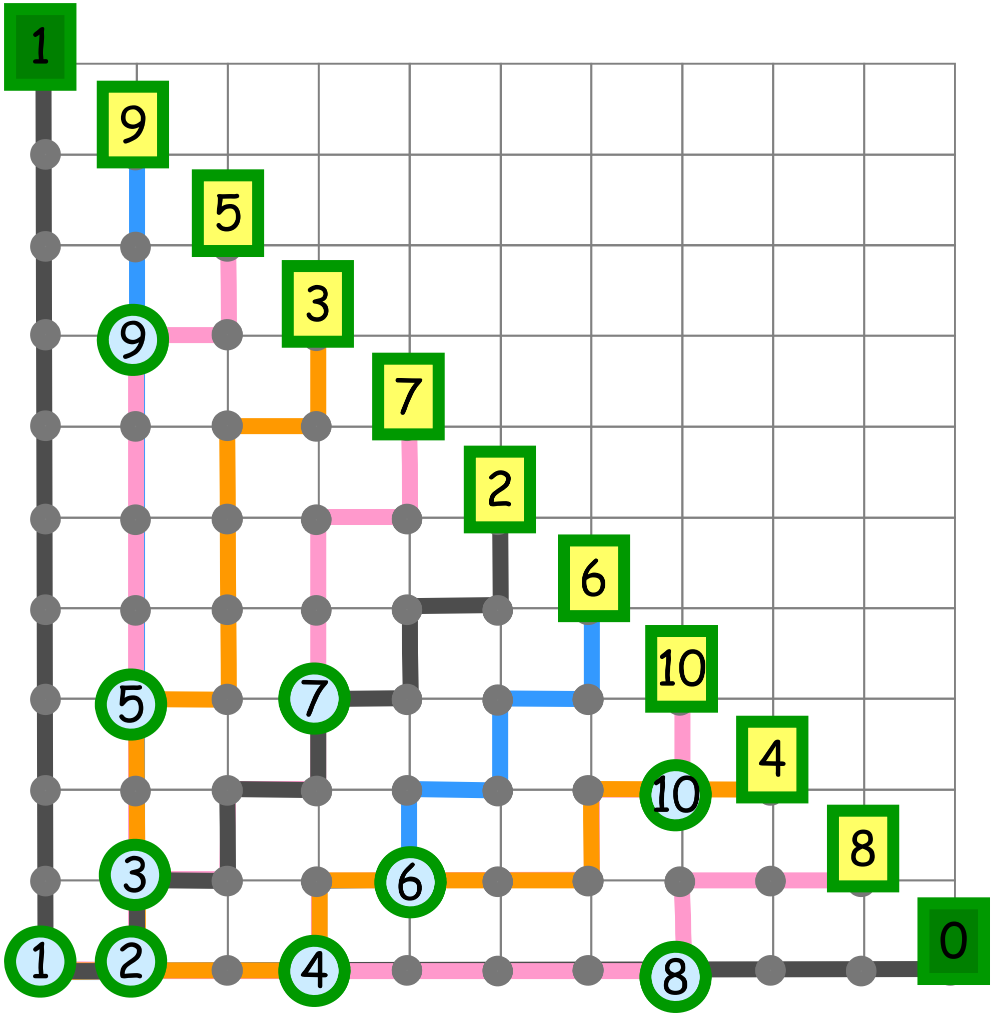


Low discrepancy sequence

- Van der Corput sequence: $O(\log n)$ discrepancy (1933)
- $\Omega(\log^{1/2} n)$ lower bound (Roth, 1954)
- $\Omega(\log n)$ lower bound (Schmidt, 1972)



Theorem (Chun-Korman-Noellenberg-T 08).
The Hausdorff-distance between a real ray and digital ray in any system of consistent digital rays in the size n x n grid cannot be smaller than the discrepancy of a sequence of length n



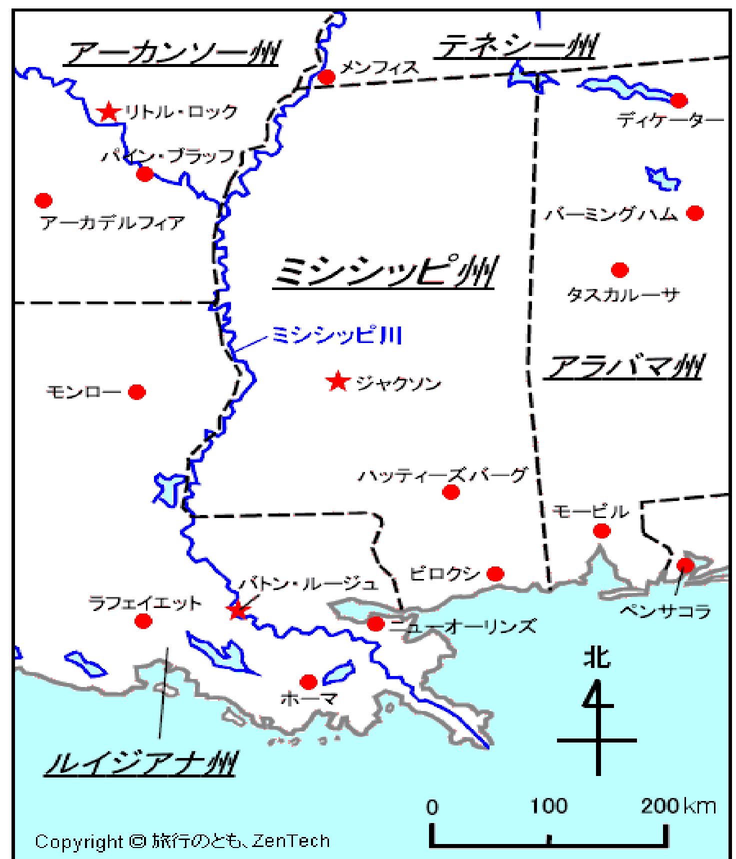
- Give labels to nodes on the diagonal
- Read the x-values of diagonal nodes ordered by the labels
- The discrepancy of this sequence equals the distance bound

10, 0, 5, 3, 8, 2, 6,……

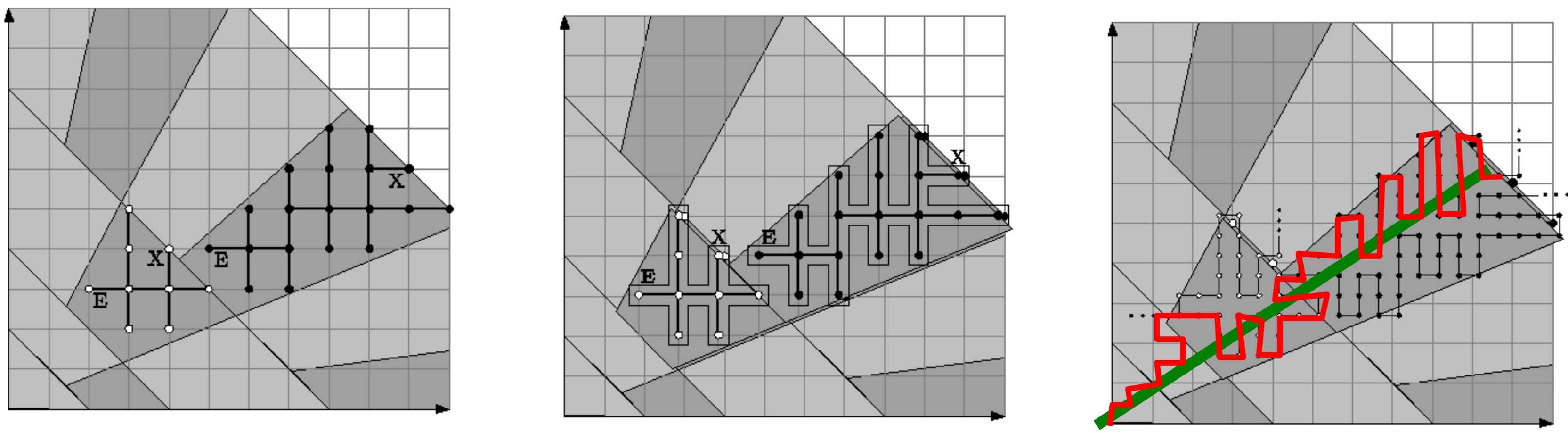
We get Van der Corput sequence from our upper bound tree construction

Snaky river can go straight

- If we ignore monotonicity, we can reduce the distance error to O(1)



How and why snaky ray can behave well?

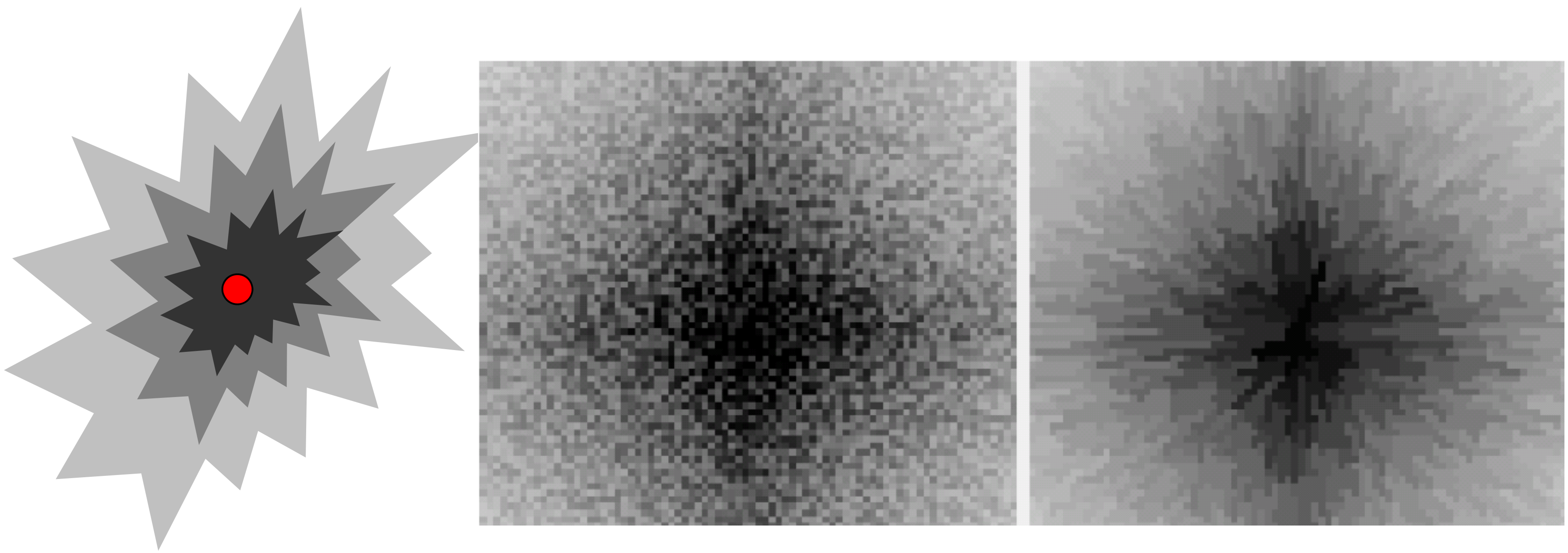


Each ray can control the direction by snaking without violating the consistency.

Visually, each ray is seen as a bold line segment.

Use of digital star-shapes

- Optimal approximation of a function by a layer of digital star shapes (like Mount Fuji)

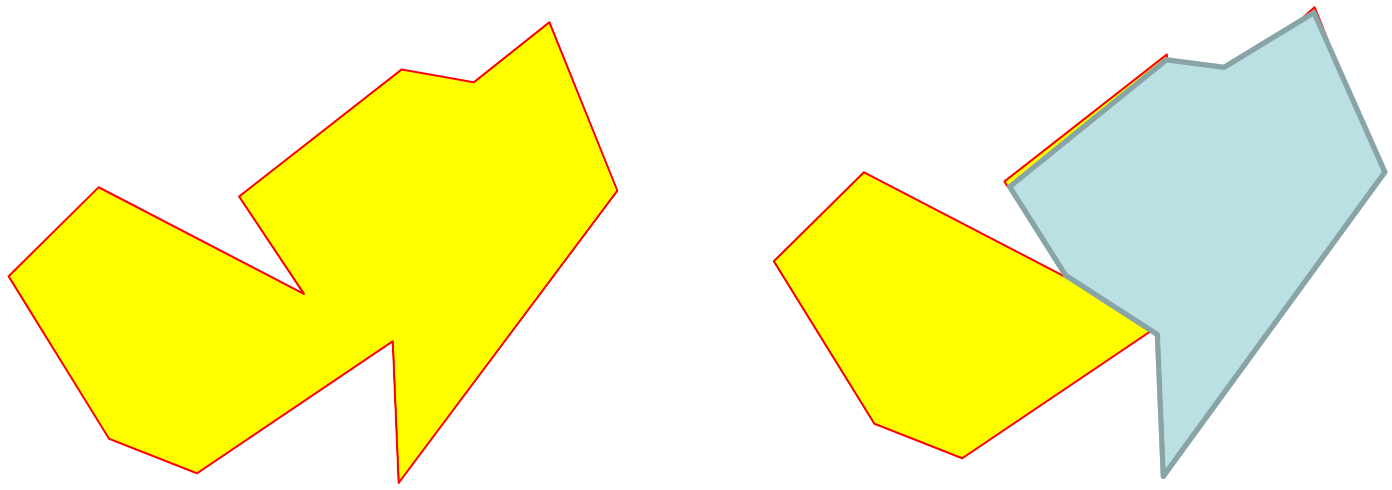


input : f(x)

output
unimodal function

Another advancement:

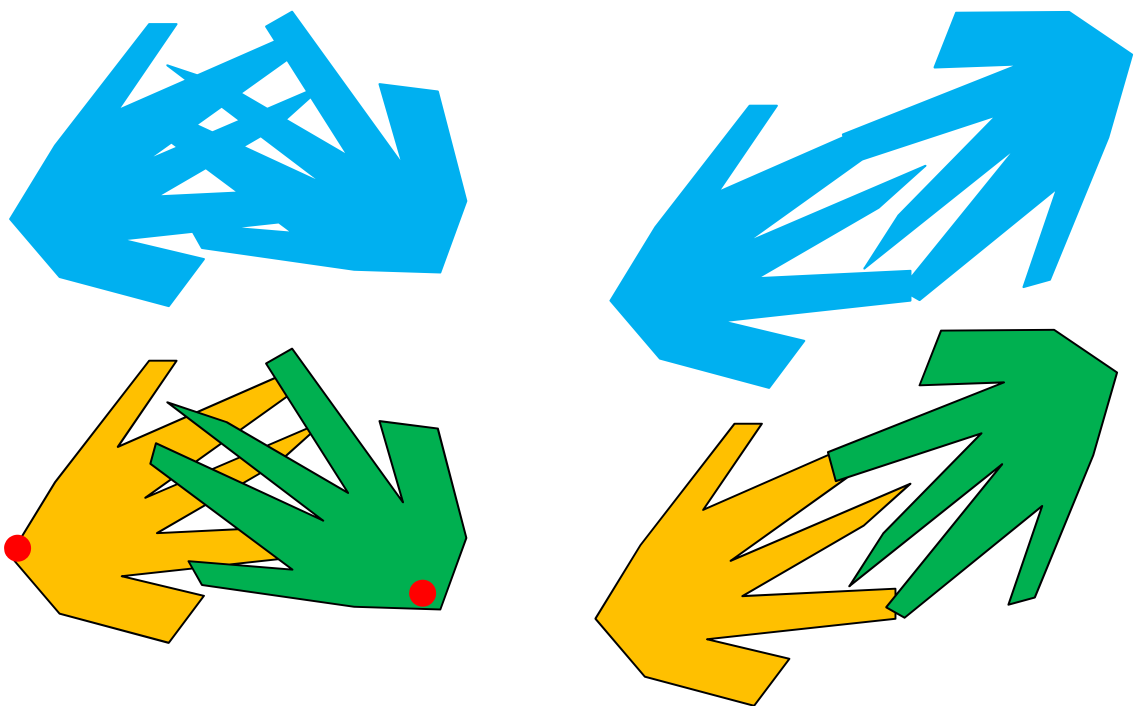
Segmentation of an image consisted from basic shapes



Segmentation of union/composition

1. Find the max-weight region that is a union of two digital star shapes.
2. Find the max-weight region that is decomposable into two digital star shapes.

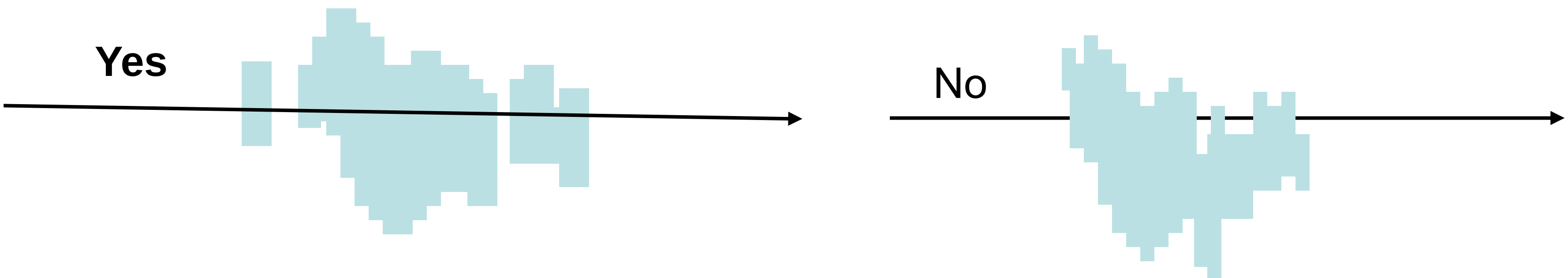
- Problem 1 is NP-hard. Problem 2 is in P.



It is open if we consider three digital star shapes .

Image consisting of k basic regions

- Basic region 1: Base-monotone region with a base-line

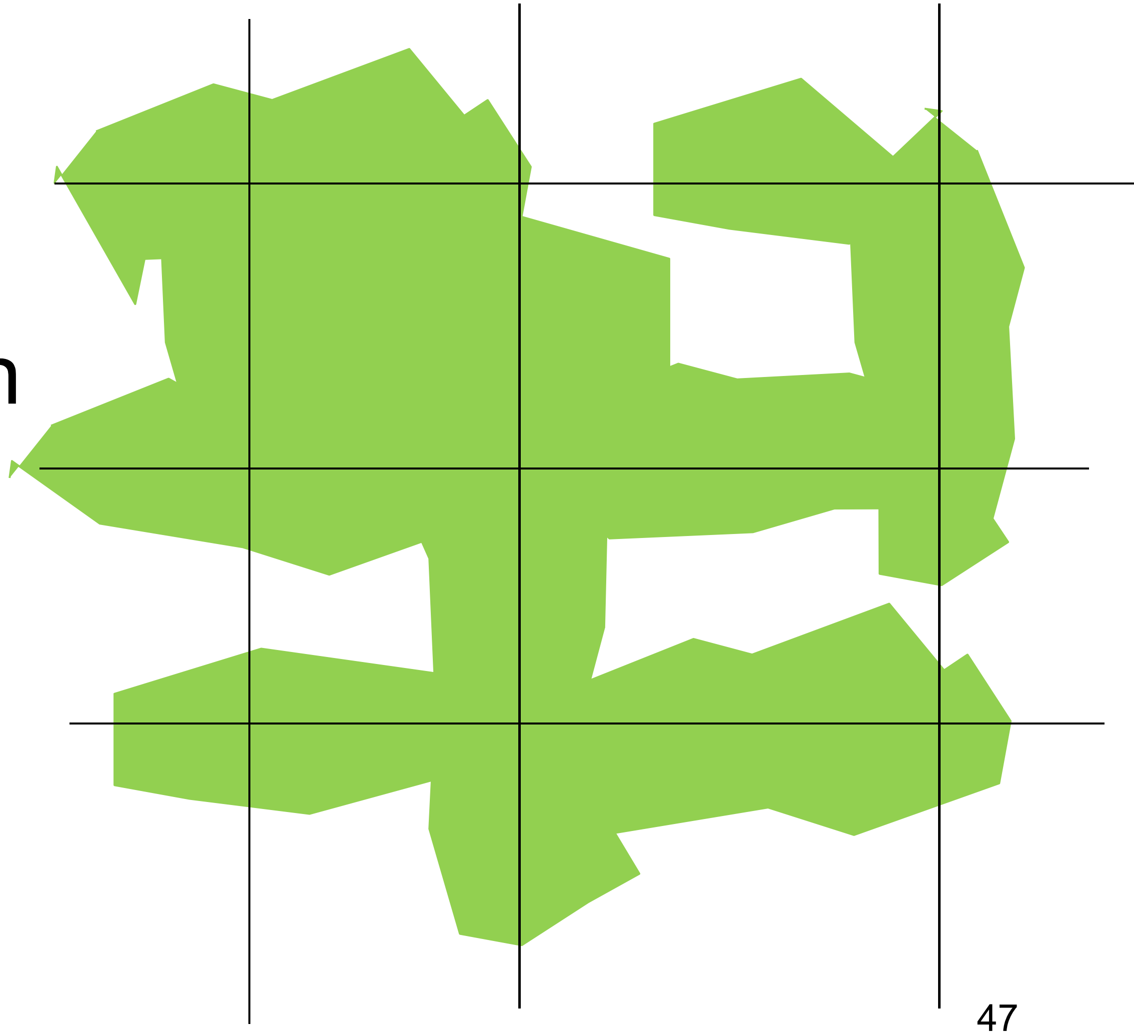


Find the max-weight region decomposable into base-monotone regions with a given set of baselines

Computed in $O(n^3)$ time in an n x n grid.

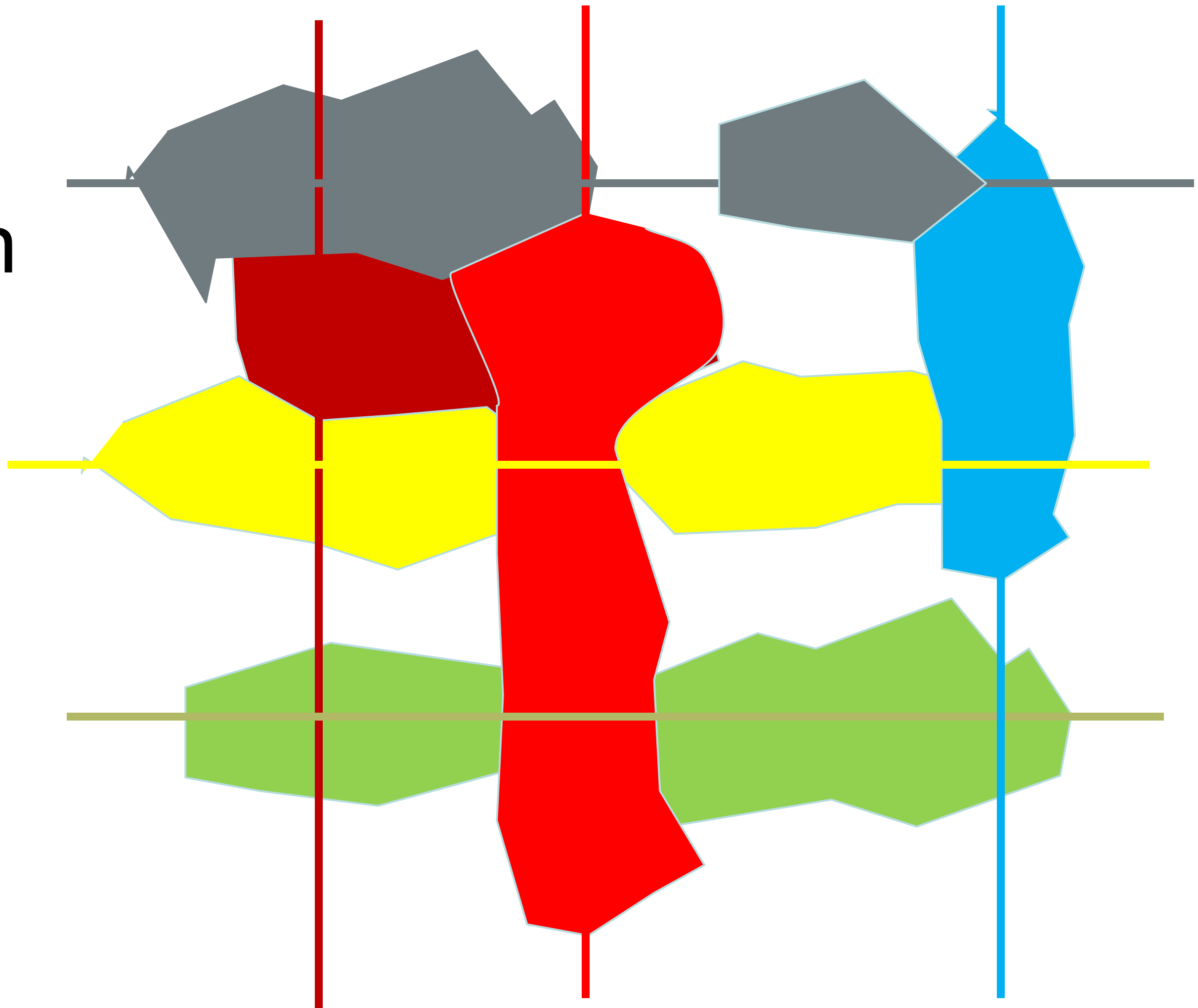
Composition of baseline monotone regions

- This picture is decomposed into baseline monotone regions of the given 6 baselines



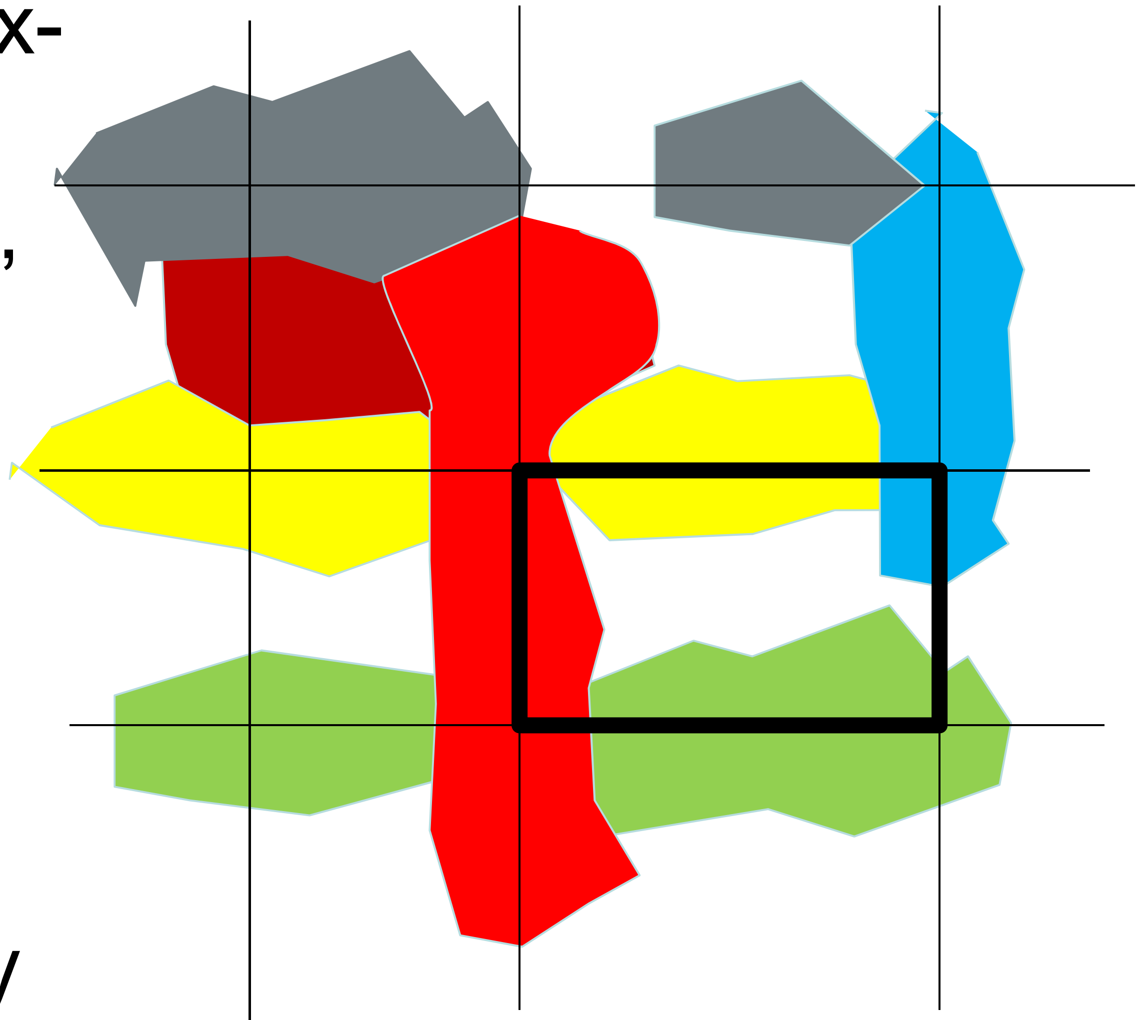
Composition of baseline regions

A possible decomposition



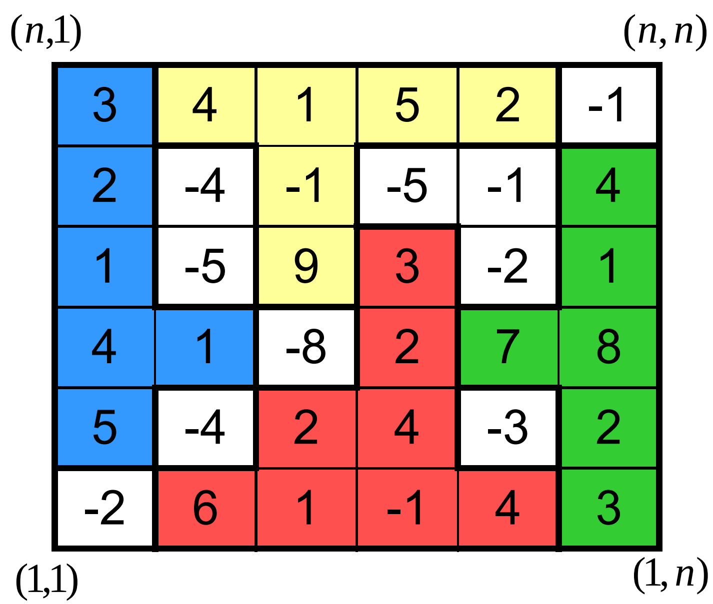
The algorithm

- We find the max-weight region in each rectangles, and combine them.
- In each rectangle, the problem (room problem) is solved efficiently



Room problem:

•Find the maximum weight region decomposable into four base-monotone regions corresponding to boundary edges.



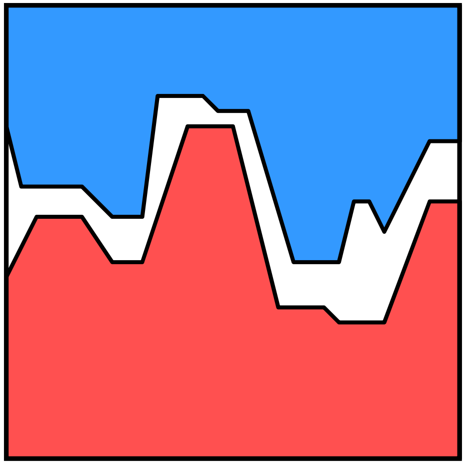
Room problem:

$(n,1)$ (n,n)

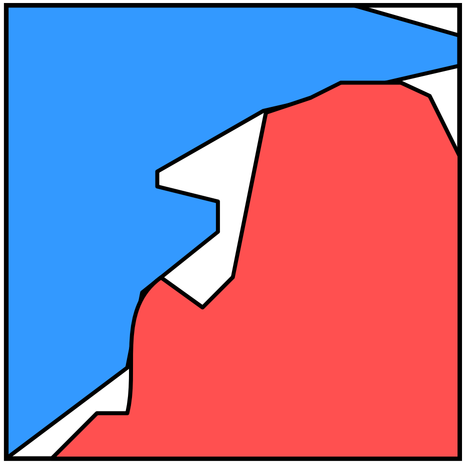
3	4	1	5	2	-1
2	-4	-1	-5	-1	4
1	-5	9	3	-2	1
4	1	-8	2	7	8
5	-4	2	4	-3	2
-2	6	1	-1	4	3

$(1,1)$ $(1,n)$

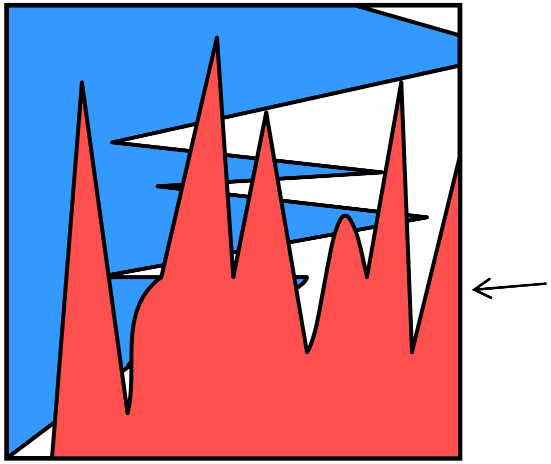
Idea: If two regions instead of four



Known: complement of x-monotone region

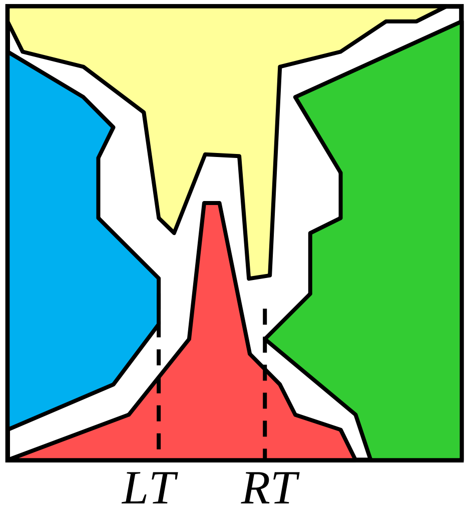


Linear time algorithm (Dynamic Programming)

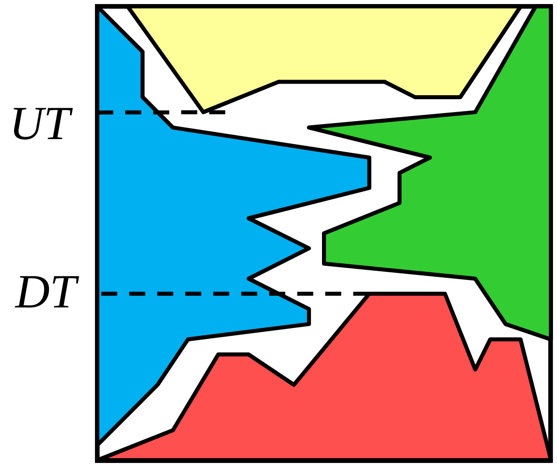


This is NP-hard

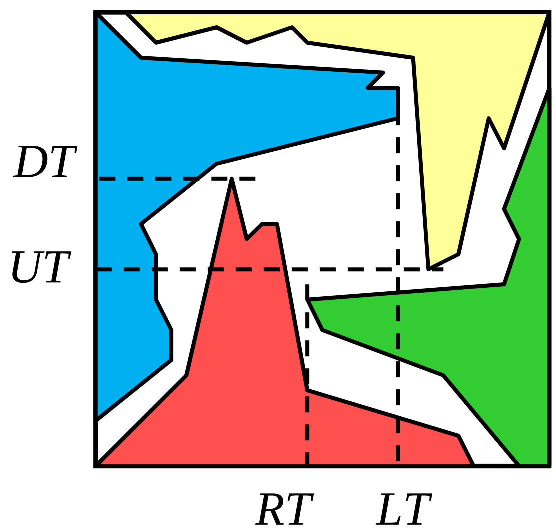
Possible patterns of four regions



$LeftTop < RightTop$

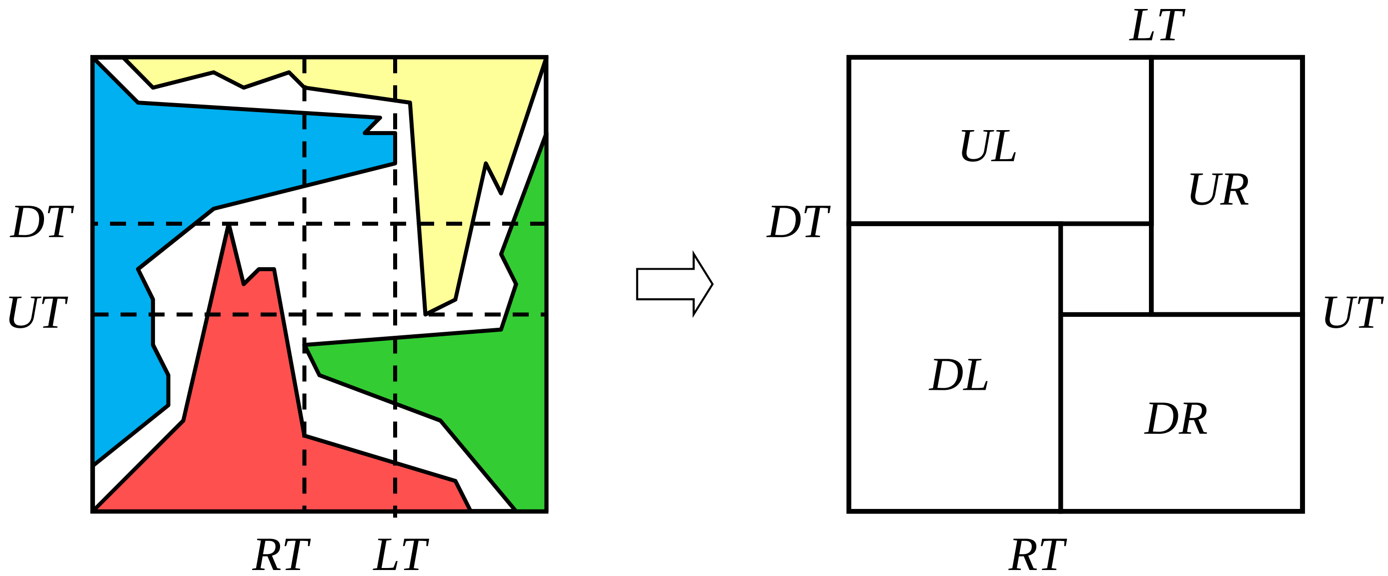


$DownTop < UpTop$



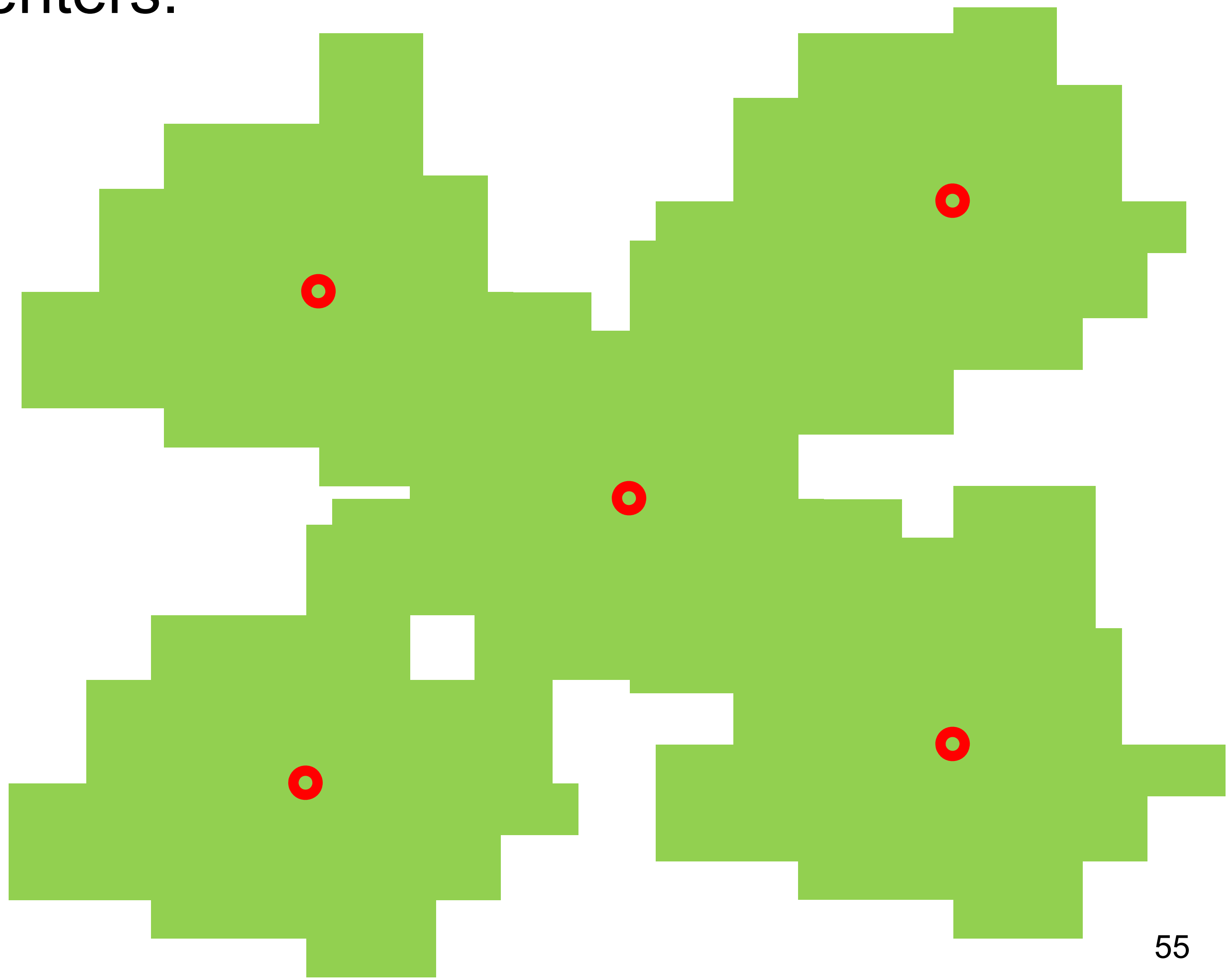
$(LeftTop \geq RightTop) \wedge (DownTop \geq UpTop)$

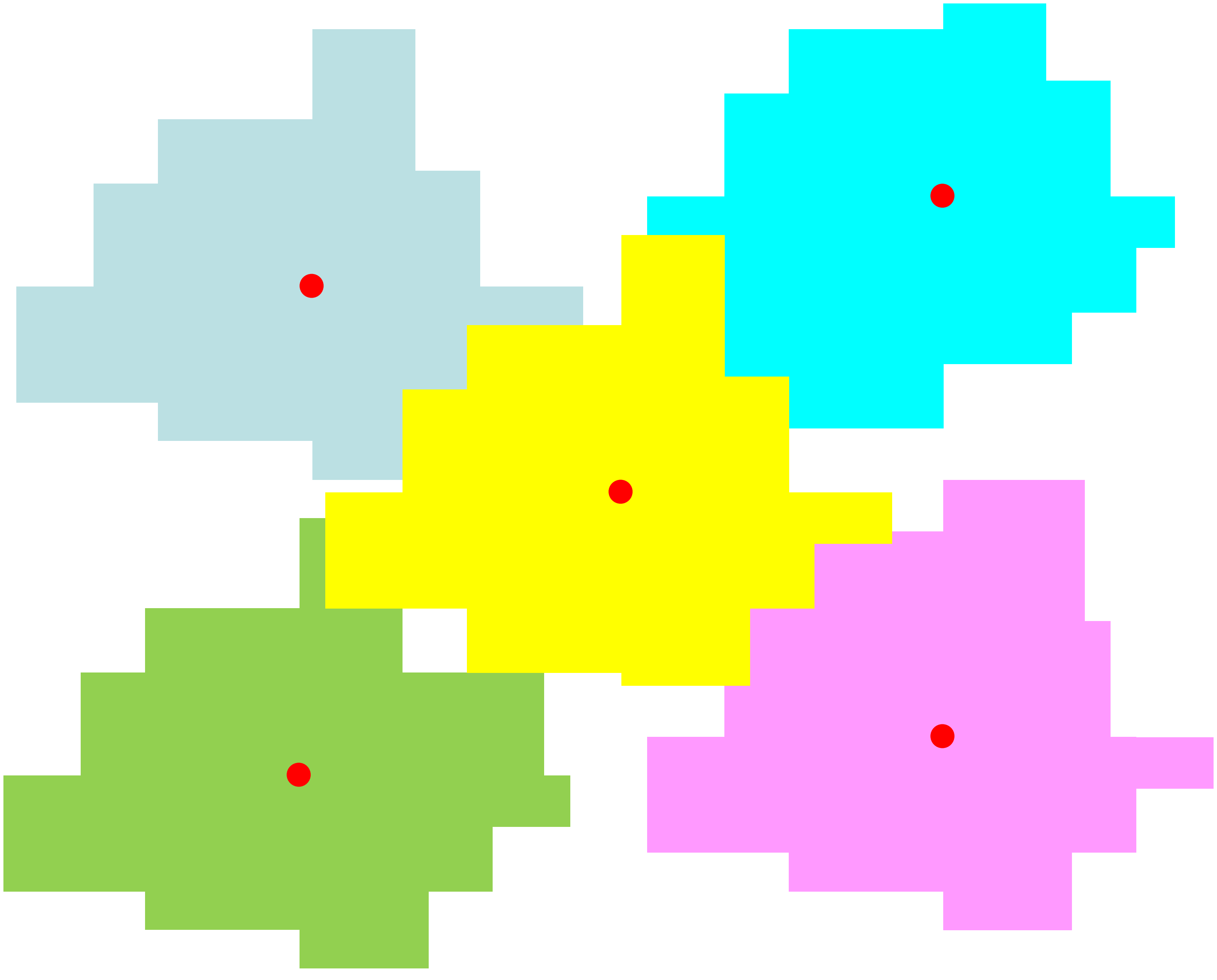
Four region case can be decomposed into two-region cases



Thus, apparently polynomial time solvable. We should use a better method to attain $O(n^3)$ time algorithm in an n x n grid.

Similar problem??: Find the max-weight region decomposable into k staircase convex regions for given k centers.





$O(n^{2k})$ time algorithm is not difficult to obtain

Open problems

- Digital line segments: No essentially better system than L-shape paths?
- Digital rays emanated from two centers.
- Composition of k staircase convex regions
  - Currently, only $O(n^{2k})$ time solution
  - Fixed Parameter Tractable algorithm ? $O(f(k)\, n^c)$
- Composition of three star-shape regions.
  - "Three forests" → NP-hard

Open problems

- Handling color images explicitly
  - Looks difficult
  - Currently, we project color vectors to transform to a monochromatic image
    - User can pick a color vector to determine the projection
  - 2-center problem in 3-d if we ignore the geometric shape of the image