Muli-Scale Discrete Geometry

Mouhammad Said[12], Jacques-Olivier Lachaud[1], Fabien Feschet[2]

[1] Laboratoire de Mathématiques, UMR 5127 CNRS, Université de Savoie,
73376 Le Bourget du Lac, France

[2]LAIC, Univ. Clermont-Ferrand,
IUT, Campus des Cézeaux, 63172 Aubière Cedex, France

24/04/2009

Outline

1. Introduction
2. Covering of a Digital Line
   - Naive Digital Line
   - Standard Digital Line
3. A faster DSS recognition Algorithm
   - DSS, patterns, irreductible fractions and continued fractions
   - Fast DSS recognition when DSL container is known
   - Example
4. Results
   - Example of Multi-Scale covering of a digital contour
5. Conclusion and Perspective

Outline

Introduction

? A Common problem in image processing is to estimate derivatives in an image, along the boundary of a shape or along a digital curve.

Previous work

- Derivative lead to the construction of geometric estimates such as tangents or curvature.
  - Finite number of scales.

Introduction

? A Common problem in image processing is to estimate derivatives in an image, along the boundary of a shape or along a digital curve.

Previous work

- Derivative lead to the construction of geometric estimates such as tangents or curvature.
  - Finite number of scales.
- Convolution of a parameterized family of Gaussian kernels.
  + Provide a multiscale representation of the feature geometry.
  - Gaussian convolutions does not process well binary data such as digital curves.

Introduction

? A Common problem in image processing is to estimate derivatives in an image, along the boundary of a shape or along a digital curve.

Previous work

- Derivative lead to the construction of geometric estimates such as tangents or curvature.
  - Finite number of scales.
- Convolution of a parameterized family of Gaussian kernels.
  - \+ Provide a multiscale representation of the feature geometry.
  - \- Gaussian convolutions does not process well binary data such as digital curves.
- Consider the multiscale approach of (Vacavant, Coeurjolly and Tougne) in the sense that geometric objects are represented by a multi-resolution set of rectilinear tiles.
  - \+ Determine the minimal number of rectangles whose union covers the considered object.
  - \+ Define geometric primitives such as lines and use them to analyse thick digital objects.
  - \- Unclear for multiscale analysis of digital object features, and the constructed objects are not analytically defined.

Introduction

Tools for our works

- Tangent estimators (pieces of digital lines) and circular arcs estimate curvature.
- Keep them in the multiscale analysis of digital boundaries.

Our point of view

The work of Figueiredo who studied the behavior of 8-connected lines when changing the resolution of the grid.

Outline

Covering of a Naive Digital Line by a lower resolution grid

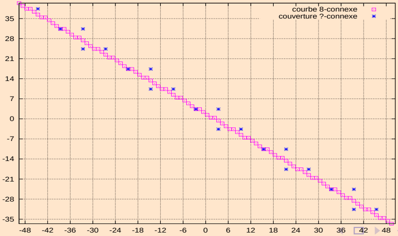
Theorem : The discrete line Δ of $S(h,v)$ covering the naive digital line $D(a,b,\mu)$ of $\mathbb{Z}^2$ is defined by :

$$-\left[\frac{-(\mu+a+b)}{g}\right]-\alpha-\beta \leq \alpha X+\beta Y < \left[\frac{\mu+b-1}{g}\right]+1$$

where $\alpha = \frac{ah}{g}$, $\beta = \frac{bv}{g}$ and $g = gcd(ah,bv)$

Example : $D(7,9,6)$ with $(h,v)=(6,7)$

$$D : 6 \leq 7x+9y < 15$$
$$\Delta : -3 \leq 2X+3Y < 1$$



Covering of a Standard Digital Line by a lower resolution grid

Theorem1

The digital straight line Δ of $S(h, v)$ covering the standard digital line $D(a, b, \mu)$ of $\mathbb{Z}^2$ is defined by :

$$-p + Q_2 - Q_1 + SI \leq \alpha X + \beta Y < Q_3 - Q_2 + SS$$

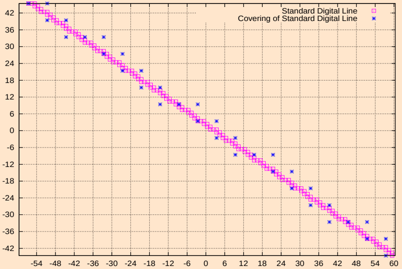
where $\alpha = \frac{ah}{g}$, $\beta = \frac{bv}{g}$, $g = gcd(ah, bv)$, $p = \alpha + \beta$,
$Q_k = \left[\frac{(k-1)\mu + k(a+b) - 1}{g}\right], k = 1, 2, 3,\ Q = \left[\frac{\mu + a + b}{g}\right]$,
$R_k = \left\{\frac{(k-1)\mu + k(a+b) - 1}{g}\right\}, k = 1, 2, 3,\ R = \left\{\frac{\mu + a + b}{g}\right\}$, and

$$SI = \begin{cases} 0 & \text{if} \quad R_2 \leq R_1 \\ 1 & \text{otherwise} \end{cases} \qquad SS = \begin{cases} 0 & \text{if} \quad R_3 \leq R_2 \\ 1 & \text{otherwise} \end{cases}$$

Example

$D(7,9,6)$ with $(h,v) = (6,6)$

$$D : 6 \leq 7x + 9y < 15$$
$$\Delta : -12 \leq 7X + 9Y < 4$$



Standard Digital Line

Theorem2

The covering line Δ is standard.

$$(Q_3 - Q_2 + SS) - (-p + Q_2 - Q_1 + SI) = \alpha + \beta\,?$$

Standard Digital Line

Theorem2

The covering line Δ is standard.

$$(Q_3 - Q_2 + SS) - (-p + Q_2 - Q_1 + SI) = \alpha + \beta\,?$$

Proposition1

$$\frac{R_2 - R_1 - R}{g} = \begin{cases} 0 & \text{if} \quad R_1 \leq R_2 \\ -1 & \text{otherwise.} \end{cases} \quad \text{and} \quad \frac{R_3 - R_2 - R}{g} = \begin{cases} 0 & \text{if} \quad R_2 \leq R_3 \\ -1 & \text{otherwise.} \end{cases}$$

Proof :

- $-2 < \frac{R_2 - R_1 - R}{g} < 1$.
- If $R_1 \leq R_2$, then $R_2 = (\mu + 2a + 2b - 1) \bmod g = (\mu + a + b) \bmod g + (a + b - 1) \bmod g = R + R_1$. Therefore $R_2 - R_1 - R = 0$ and $\frac{R_2 - R_1 - R}{g} = 0$.
- If $R_1 > R_2$, then $R_2 = (\mu + 2a + 2b - 1) \bmod g = (\mu + a + b) \bmod g + (a + b - 1) \bmod g - g = R + R_1 - g$. Therefore $R_2 - R_1 - R = -g$ and $\frac{R_2 - R_1 - R}{g} = -1$.

Standard Digital Line

Lemma1

$$Q_2 - Q_1 = Q + \begin{cases} 0 & \text{if} \quad R_1 \leq R_2 \\ 1 & \text{otherwise.} \end{cases}, \quad Q_3 - Q_2 = Q + \begin{cases} 0 & \text{if} \quad R_2 \leq R_3 \\ 1 & \text{otherwise.} \end{cases}$$

Proof :

- $\mu + a + b = (\mu + 2a + 2b - 1) - (a + b - 1)$.
- $\mu + a + b = gQ + R$, $a + b - 1 = gQ_1 + R_1$, and $\mu + 2a + 2b - 1 = gQ_2 + R_2$.

Then,

$$g(Q - Q_2 + Q_1) = R_2 - R_1 - R$$

- If $R_1 \leq R_2$, then by Proposition 1, $\frac{R_2 - R_1 - R}{g} = 0$ which implies that $Q = Q_2 - Q_1$,
- If $R_1 > R_2$, then by Proposition 1, $\frac{R_2 - R_1 - R}{g} = -1$ which implies that $Q = Q_2 - Q_1 - 1$.

Outline

Patterns, irreductible fractions and continued fractions

Definition 1

Given a standard line (a, b, μ), we call pattern of characteristics (a, b) the succession of Freeman moves between any two consecutive upper leaning points. The Freeman moves defined between any two consecutive lower leaning points is the previous word read from back to front and is called the reversed pattern.

Definition 2

A *simple continued fraction* is an expression of the following form :

$$z = \frac{a}{b} = [0, u_1, u_2, ..., u_i, ..., u_n] = 0 + \frac{1}{u_1 + \frac{1}{\ldots + \frac{1}{u_{n-1} + \frac{1}{u_n}}}}$$

Update DSS

Proposition2

The slope evolution in **DR95** depends on the parity of the depth of its slope, the type of weakly exterior point added to the right. This is summed up in the table below, where the slope is $[0, u_1, ..., u_k]$, $k = 2i$ even or $k = 2i + 1$ odd, δ pattern(s) and δ' reversed pattern(s) :

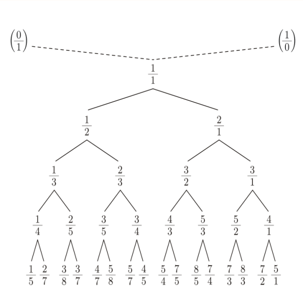


FIG.: Stern-Brocot Tree, hierarchy containing all the positive irreductible rational fractions

	Even k	Odd k
Upper weakly exterior	$[0, u_1, ..., u_{2i}, \delta]$	$[0, u_1, ..., u_{2i+1} - 1, 1, \delta]$
Lower weakly exterior	$[0, u_1, ..., u_{2i} - 1, 1, \delta']$	$[0, u_1, ..., u_{2i+1}, \delta']$

Proposition3

If $D'(a', b', \mu')$ is the covering of some digital line D by a tiling $S(h, v)$, then any segment $S \subset D$ induce a segment $S' \subset D'$ by the same tiling, such that the slope of S' is $\frac{a'}{b'}$ or one of its ancestors in the Stern-Brocot tree.

? When the slope of S' is exactly the slope of D'

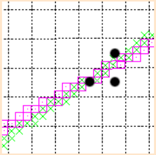
Proposition4

Assume S contains δ patterns in D. Let us also suppose the first and last points of S' are upper or lower learning points. If v divides δa and h divides δb, then the slope of S' is equal to the slope of D' if and only if by cases :

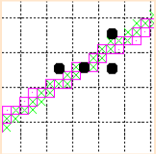
- $h = v$ and $\delta \geq h$.
- $\gcd(h, v) = 1$ and
  - $\gcd(a, v) = 1$, $\gcd(b, h) = 1$ and $\delta \geq hv$.
  - $\gcd(a, v) = 1$, $\gcd(b, h) \neq 1$, $\delta \geq \frac{hv}{g_1}$, and $g_1 = \gcd(b, h)$.
  - $\gcd(a, v) \neq 1$, $\gcd(b, h) = 1$, $\delta \geq \frac{hv}{g_2}$, and $g_2 = \gcd(a, v)$.
  - $\gcd(a, v) \neq 1$, $\gcd(b, h) \neq 1$, and $\delta \geq \frac{hv}{g_1 g_2}$.
- (v is multiple of h) and
  - $\gcd(a, v) \neq 1$, $\gcd(b, h) = 1$, and $\delta \geq h$.
  - $\gcd(a, v) = 1$ and $\delta \geq v$.
  - $\gcd(a, v) \neq 1$, $\gcd(b, h) \neq 1$, and $\delta \geq \frac{v}{g_2}$.

Example 1

$D(2, 3, 0)$ and $(h, v) = (3, 4) \Rightarrow \Delta(1, 2, -3)$ (Theorem 1)
As $(h, v) = 1$, $(a, v) = (2, 4) \neq 1$ and $(b, h) = (3, 3) \neq 1$, then $\delta \geq 2$ (Proposition 4)



$\delta = 1$ and $S'(1, 1)$



$\delta = 2$ and $S'(1, 2)$

Example 2

$D(2,3,-2)$ and $(h,v) = (6,6) \Rightarrow \Delta(2,3,-4)$
As $h = v$, then $\delta \geq 6$

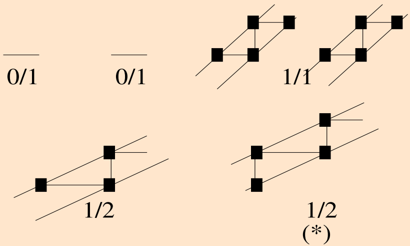


FIG.: (*) $\delta = 6$, $S'(1,2)$ and the slope of S' does not equal to the slope of Δ, because the first and last points does not upper or lower leaning points

Example 2

$D(2, 3, -2)$ and $(h, v) = (6, 6) \Rightarrow \Delta(2, 3, -4)$
As $h = v$, then $\delta \geq 6$

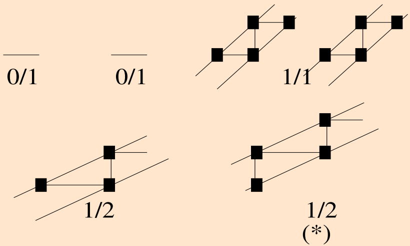


FIG.: (*) $\delta = 6$, $S'(1, 2)$ and the slope of S' does not equal to the slope of Δ, because the first and last points does not upper or lower leaning points

It is harder to compute analytically the exact characteristics of a DSS.

Coefficients of Bezout

Lemma2

The coefficients of Bezout relating to the couple of integers (a, b) are written according to the parity of the complexity of $\frac{a}{b}$, and denoting by $\frac{p_k}{q_k}$ the k-ieme convergent of $\frac{a}{b}$:

$$\begin{array}{cc} (q_{2n}, p_{2n}) & \text{if } \frac{a}{b} \text{ is of complexity } 2n+1 \\ (b - q_{2n-1}, a - p_{2n-1}) & \text{if } \frac{a}{b} \text{ is of complexity } 2n \end{array}$$

Algorithm (Update Slope)

Action UpdateSlope(**In** uw, δ, **InOut** k, u, p, q);
uw : boolean ; /* *True* iff upper weak leaning point */
δ : integer ; /* number of (reversed) patterns */
k : integer ; /* depth of slope continued fraction */
u, p, q : array of integers ; /* slope cont. fraction */
begin
  if *(wu = true and k is odd) or (wu = false and k is even)* **then**
    $u_k \leftarrow u_k - 1$, $p_k \leftarrow p_k - p_{k-1}$, $q_k \leftarrow q_k - q_{k-1}$;
    $u_{k+1} \leftarrow 1$, $p_{k+1} \leftarrow p_k + p_{k-1}$, $q_{k+1} \leftarrow q_k + q_{k-1}$;
    $k \leftarrow k + 1$;
  if $\delta = 1$ **then**
    $u_k \leftarrow u_k + 1$, $p_k \leftarrow p_k + p_{k-1}$, $q_k \leftarrow q_k + q_{k-1}$;
  else
    $u_{k+1} \leftarrow \delta$, $p_{k+1} \leftarrow \delta p_k + p_{k-1}$, $q_{k+1} \leftarrow \delta q_k + q_{k-1}$;
    $k \leftarrow k + 1$;
end

Algorithm 1: Updates in $O(1)$ the slope of a DSS according to the addition of an upper leaning point (*uw* is *true*) or lower leaning point (*uw* is *false*), to the number of patterns or reversed patterns δ, and to the current continued fraction of the slope.

Action SmartDSS(**In** D, **In** P, Q, **Out** S);
D : DSL (α, β, μ'), P, Q : Point of $\mathbb{Z}^2$ S : DSS (a, b, μ);
Var u, p, q : array of integers /*Cont. fraction $\frac{a}{b} = [u_0, \ldots, u_k] = \frac{p_k}{q_k}$ */
Var U, L, U', L' : Point of $\mathbb{Z}^2$; $ulu, lul, inside$: boolean ; $k, loop$: integer ;
begin
  $k \leftarrow 0,\ u_0 \leftarrow 0,\ p_0 \leftarrow 0,\ q_0 \leftarrow 1,\ p_{-1} \leftarrow 1,\ q_{-1} \leftarrow 0$;
  $U \leftarrow P,\ L \leftarrow P$; $inside \leftarrow true,\ ulu \leftarrow true,\ lul \leftarrow true$;
  while *inside and* $p_k \neq \alpha$ **do**
    $(a, b) \leftarrow (p_k, q_k)$;
1    $(b', a') \leftarrow$ Bézout(p, q, k) /*$ab' - ba' = 1$*/;
    $U' \leftarrow U + (b - b', a - a')$;
    $L' \leftarrow L + (b', a')$;
    $\delta \leftarrow 1,\ loop \leftarrow 0$;
2   **repeat**
      $U' \leftarrow U' + (b, a),\ L' \leftarrow L' + (b, a)$;
      if $U'_y \leq Q_y$ *and* $U' \in D$ **then** $loop \leftarrow 1$;
      else if $L'_x \leq Q_x$ *and* $L' \in D$ **then** $loop \leftarrow 2$;
      $\delta \leftarrow \delta + 1$;
    until $U'_y \geq Q_y$ *or* $L'_x \geq Q_x$;
3   **if** $loop = 1$ /*Increase slope with weak upper leaning point U'*/ **then**
      UpdateSlope$(true, \delta, k, u, p, q)$;
4      $L \leftarrow L' - (b', a')$;
      if *not lul* **then** $L \leftarrow L - (b, a)$;
      $ulu \leftarrow true,\ lul \leftarrow false$;
5   **if** $loop = 2$ /*Decrease slope with weak lower leaning point L'*/ **then**
      UpdateSlope$(false, \delta, k, u, p, q)$;
6      $U \leftarrow U' - (b - b', a - a')$;
      if *not ulu* **then** $U \leftarrow U - (b, a)$;
      $ulu \leftarrow false,\ lul \leftarrow true$;
    else
      $inside \leftarrow false$
    ;
  $a \leftarrow p_k,\ b \leftarrow q_k,\ \mu \leftarrow aU_x - bU_y$;
end

Number of tested points

Proposition5

Let S be a DSS of slope $\frac{a}{b} = [u_0, u_1, ..., u_k]$, $T(n)$ the number of points on S tested by Algorithm 2 to recognize $\frac{a}{b}$ with $\sum_{i=0}^{k} u_i = n$, then $T(n) \leq 2n$ (it only depends on the sum of u_i).

Proof

- $T(0) = 0$ is obvious,
- Assume that $T(n) \leq 2n$ for all j and we shall prove that $T(n+1) \leq 2n+2$,
- As $T(n) = 2n$, then $\frac{a}{b}$ is some $[u_0, u_1, ..., u_{2i}]$. According to Stern-Brocot Tree, there are only two possible evolution for the slope, either $[u_0, u_1, ..., u_{2i}, 1]$ or $[u_0, u_1, ..., u_{2i} - 1, 1, 1]$.
  - Upper weakly exterior, the new slope increases to $[u_0, u_1, ..., u_{2i}, 1]$ and $T(n+1) = T(n) + 1$.
  - Lower weakly exterior, then the slope decreases to $[u_0, u_1, ..., u_{2i} - 1, 1, 1]$ and $T(n+1) = T(n) + 2$.
- Finally, we conclude that $T(n+1) \leq T(n) + 2 = 2n + 2$.

Example

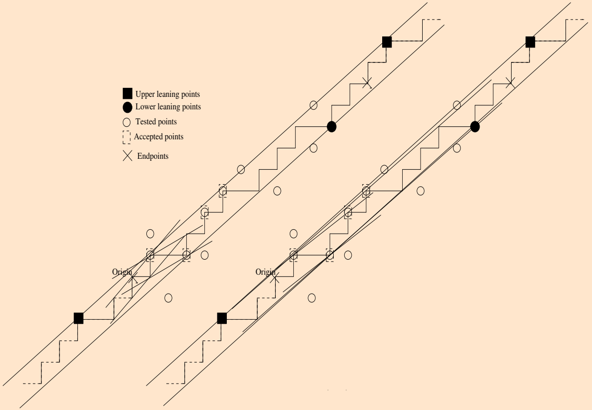


FIG.: A digital straight line $D(13, 17, -5)$ with an odd slope. Computes the characteristics (a, b, μ) of a DSS S that is the subset of D between the origin and the point $(12, 9)$. The intermediate slopes are drawn with solid lines on the left and on the right, tested points are circled.

Example

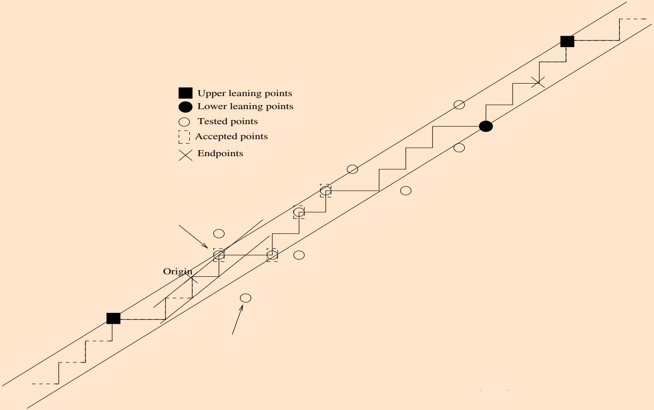


FIG.: DSS(1,1,0)

Example

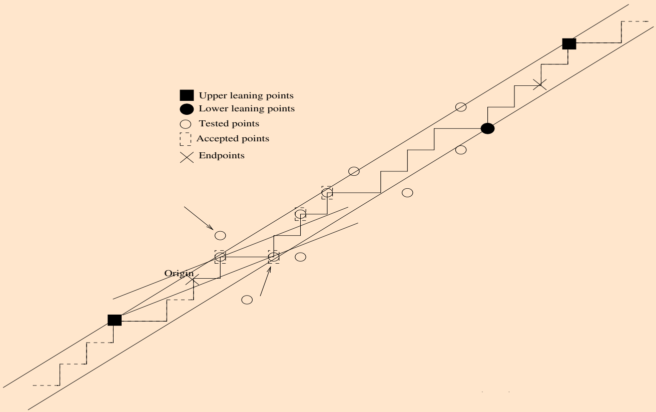


FIG.: DSS(1,2,-1)

Example

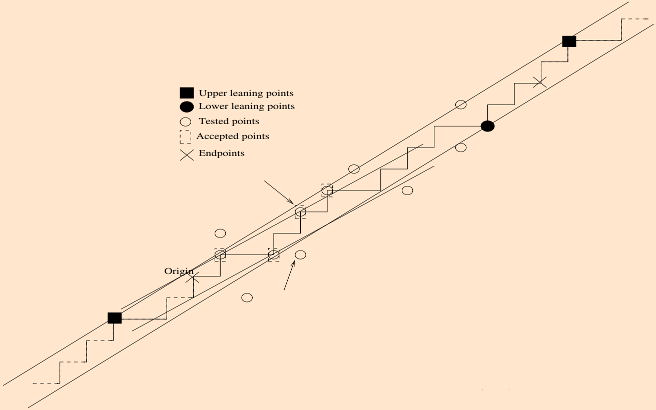


Fig.: DSS(2,3,-1)

Example

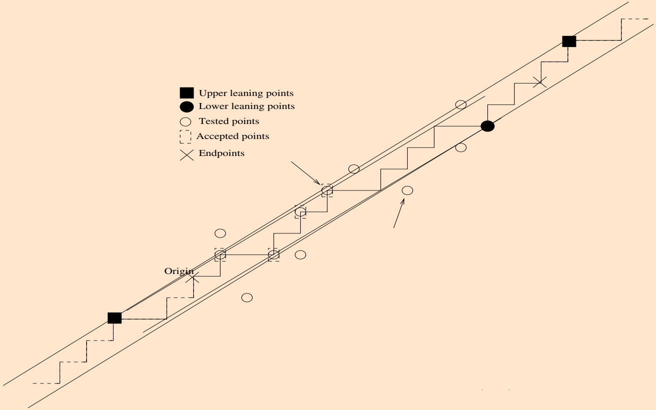


FIG.: DSS(3,4,-1)

Example

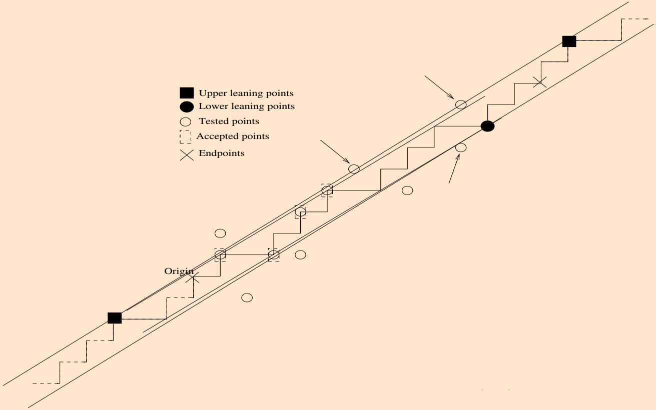


FIG.: DSS(3,4,-1)

Outline

Multi-Scale covering of a digital contour

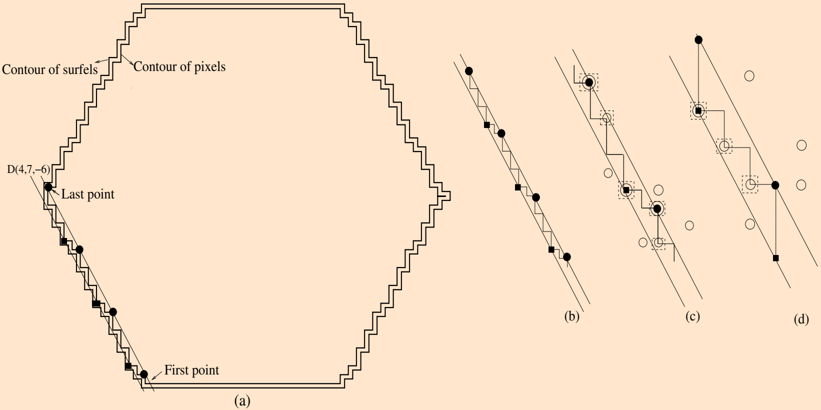


Fig.: (a) Multiscale computation of the boundary of a digital shape according to several tiling (h, v). (b) $(h, v) = (1, 1)$ and original DSL is $(4, 7, -6)$. (c) $(h, v) = (2, 2)$, DSL is $(4, 7, -8)$ and extracted DSS is $(4, 7, -8)$. (d) $(h, v) = (3, 4)$, DSL is $(16, 21, -32)$ and extracted DSS is $(3, 4, -6)$.

Outline

Conclusion and Perspective

Conclusion

- Presented new results about the covering of discrete objects by regular tilings (h, v).
- Presented a novel fast DSS recognition algorithm.
- computational complexity is $\Theta(\sum_{i=0}^{k} u_k)$.
- Compute the exact multiscale covering of a digital contour in a time proportional to $M \times \overline{U}$, where $\overline{U}$ is the average of the partial quotient sum of the *output* subsampled DSS
- In most cases, this is clearly sublinear, and at worst, linear in the size of the contour.

Conclusion and Perspective

Conclusion

- Presented new results about the covering of discrete objects by regular tilings (h, v).
- Presented a novel fast DSS recognition algorithm.
- computational complexity is $\Theta(\sum_{i=0}^{k} u_k)$.
- Compute the exact multiscale covering of a digital contour in a time proportional to $M \times \overline{U}$, where $\overline{U}$ is the average of the partial quotient sum of the *output* subsampled DSS
- In most cases, this is clearly sublinear, and at worst, linear in the size of the contour.

Perspective

- This work is a first step towards the multi-scale computation of the tangential cover, a fundamental representation of digital curves.

Questions ?

Thank you for your attention