

Bài 4

CÁC GIẢI THUẬT CƠ SỞ



NỘI DUNG

1. Các giải thuật xén tỉa
2. Thuật toán tô miền kín
3. Phép xử lý Antialiasing

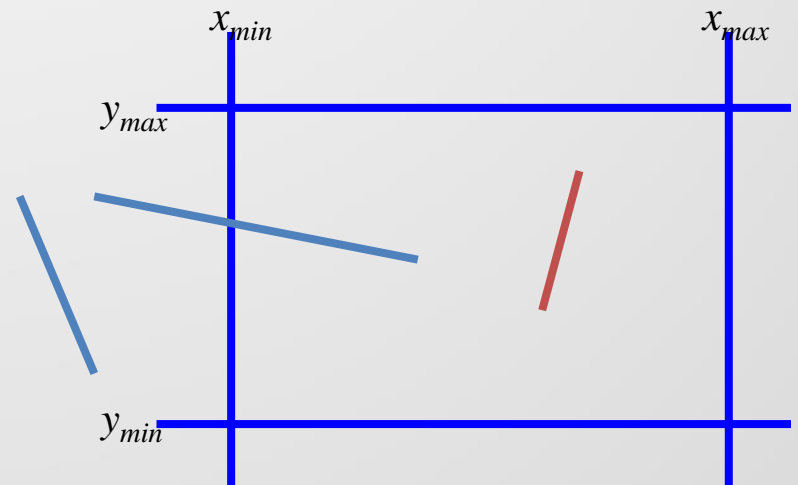
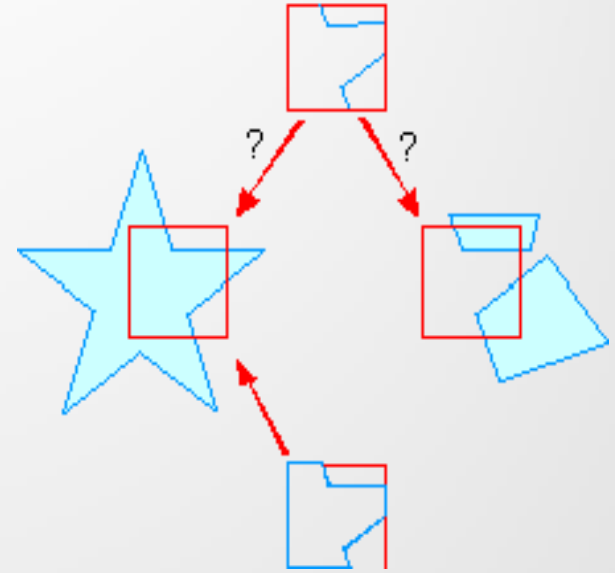


1

CÁC GIẢI THUẬT XÉN TỈA

Xén tỉa - clipping

- Khái niệm
Xén tỉa là tiến trình tự động xác định các điểm của 1 đối tượng nằm trong hay ngoài cửa sổ hiển thị
- Tiết kiệm thời gian tiến trình rasterize bỏ qua phần nằm ngoài cửa sổ hiển thị
- Clipping điểm
 - $x_{min} \leq x \leq x_{max}$
 - $y_{min} \leq y \leq y_{max}$



Clipping đoạn thẳng

- Tiến trình, giải thuật kiểm tra chấp nhận các đoạn thẳng nằm trong và loại bỏ các đoạn thẳng nằm ngoài dựa trên 2 điểm đầu cuối
- Lý do:
 - Không kiểm tra mọi điểm trên đoạn thẳng
 - Hầu hết các đoạn thẳng với 1 màn hình hiển thị đều được chấp nhận hoặc loại bỏ
 - Rất ít các đoạn thẳng cắt cửa sổ hiển thị



GIẢI THUẬT COHEN SUTHERLAND OUTCODE

- Giải thuật Cohen-Sutherland thực hiện nhanh với các trường hợp đoạn thẳng nằm trong hay ngoài cửa sổ hiện thị
- Mỗi điểm đầu cuối được gán mã code phụ thuộc vào vị trí trong mặt phẳng mã

Giải thuật Cohen Sutherland outcode

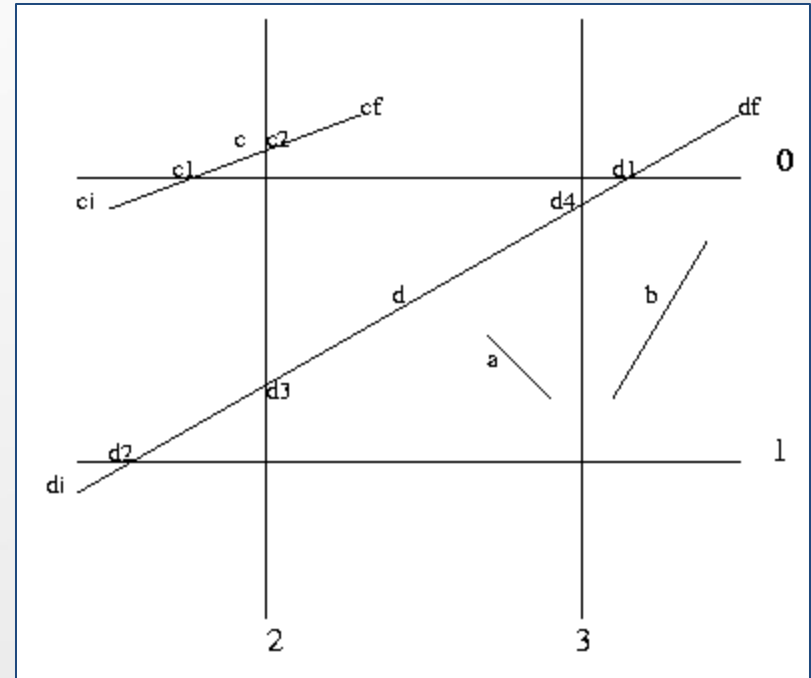
p.code = 0000

- If $p.x \leq x_{min}$ >> P.code or 0001
- If $p.y \leq y_{min}$ >> P.code or 0100
- If $p.x \geq x_{max}$ >> P.code or 0010
- If $p.y \geq y_{max}$ >> P.code or 1000



Giải thuật Cohen Sutherland outcode

- If $P1.code \text{ OR } P2.code == 0000$
 - Chấp nhận toàn đoạn thẳng
- If $P1.code \text{ AND } P2.code != 0000$
 - Loại
- Với trường hợp cắt, giải thuật xác định lại điểm đầu cuối là giao của đoạn thẳng và khung bao của cửa sổ hiển thị



Giải thuật Cyrus-beck Liang Barsky

- Giải Cohen-Sutherland yêu cầu cửa sổ là hình chữ nhật, các cạnh là cạnh của màn hình
- Vấn đề nảy sinh khi cửa sổ clip là 1 đa giác bất kỳ hoặc hình chữ nhật quay đi 1 góc
- Giải thuật Liang-Barsky tối ưu khi tìm giao điểm của đoạn thẳng với cửa sổ hiển thị
- Nicholl-Lee-Nicholl reducing redundant boundary clipping by identifying edge and corner regions

Liabarsky

- $x = x_1 + (x_2 - x_1)u = x_1 + uDx$
- $y = y_1 + (y_2 - y_1)u = y_1 + uDy$
- $x_{\min} \leq x_1 + Dx.u \leq x_{\max} \Leftrightarrow x \in [x_m, x_M]$
- $y_{\min} \leq y_1 + Dy.u \leq y_{\max} \Leftrightarrow y \in [y_m, y_M]$

- $P_k u \leq q_k \quad k = 1, 2, 3, 4$

$$\begin{cases} q_1 = x_1 - x_m \\ q_2 = x_M - x_1 \\ q_3 = y_1 - y_m \\ q_4 = y_M - y_1 \end{cases} \quad \begin{cases} P_1 = -Dx \\ P_2 = Dx \\ P_3 = -Dy \\ P_4 = Dy \end{cases}$$

Liabarsky

- Nếu $P_k = 0$: điều đó tương đương với việc đoạn thẳng đang xét song song với cạnh thứ k của hình chữ nhật clipping.
 - a) Nếu $q_k < 0 \Rightarrow$ Đường thẳng nằm ngoài cửa sổ (hệ bất phương trình trên vô nghiệm)
 - b) Nếu $q_k \geq 0$ thì đoạn thẳng nằm trong hoặc nằm trên cạnh của cửa sổ clipping.
- Hệ bất phương trình luôn thoả mãn.

Liabarsky

- Nếu $P_k \neq 0$: đoạn thẳng đang xét sẽ cắt cạnh k tương ứng của cửa sổ clipping tại vị trí trên đoạn thẳng $u_k = q_k/P_k$.
 - $P_k < 0$ đoạn thẳng có dạng đi từ ngoài vào trong
 - bất phương trình sẽ có dạng $u \geq q_k/P_k \Leftrightarrow u \geq u_k$.
 - $P_k > 0$
 - $u \geq u_k$ sẽ thuộc cửa sổ hiển thị.
 - bất phương trình sẽ có dạng $u \leq q_k/P_k$
 - $u \leq u_k$ với $u_k = q_k/P_k$ là giao của đoạn thẳng với cạnh k của cửa sổ clipping
 - đoạn thẳng có dạng đi từ trong ra ngoài so với cạnh k .

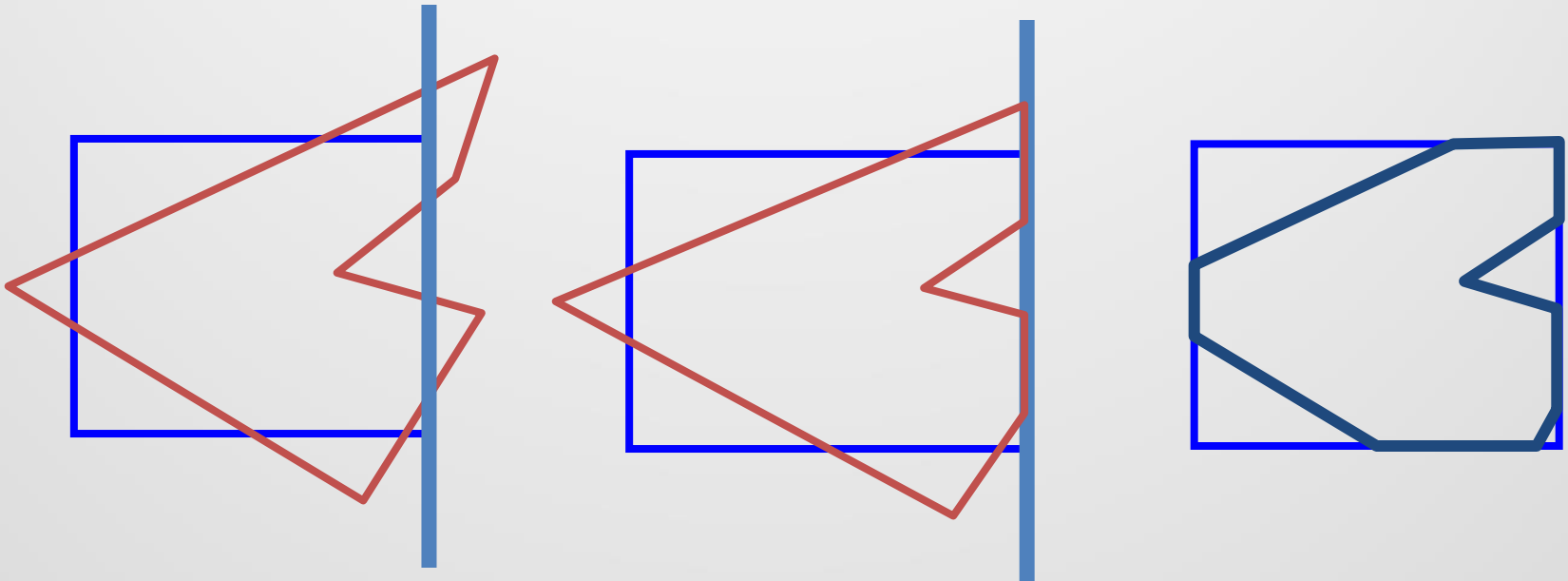
Liabarsky

- $P_k < 0$ và $u_k < 0$
 - cạnh k của cửa sổ clipping cắt đoạn thẳng tại phần mở rộng nằm ngoài đoạn thẳng.
 - $u_k \leq u < 0$ thoả mãn bất phương trình sẽ không nằm trên đoạn thẳng cần xét.
 - $\Rightarrow u_k$ sẽ nhận là 0 khi $u_k < 0$
- $P_k > 0$ và $u_k > 1$
 - $\Rightarrow u_k$ tương ứng sẽ nhận giá trị 1.
- điểm nằm trong cửa sổ clipping sẽ có dạng như sau:
 - $U_1 \leq u \leq U_2$

$$U_2 = \min \left(\{1\} \cup \left\{ u_k : u_k = \frac{q_k}{P_k}, P_k > 0 \right\} \right)$$
$$U_1 = \max \left(\{0\} \cup \left\{ u_k : u_k = \frac{q_k}{P_k}, P_k < 0 \right\} \right)$$

Sutherland-Hodgman Clipping

- Basic idea:
 - Consider each edge of the viewport individually
 - Clip the polygon against the edge equation
 - After doing all planes, the polygon is fully clipped



Sutherland-Hodgman Clipping

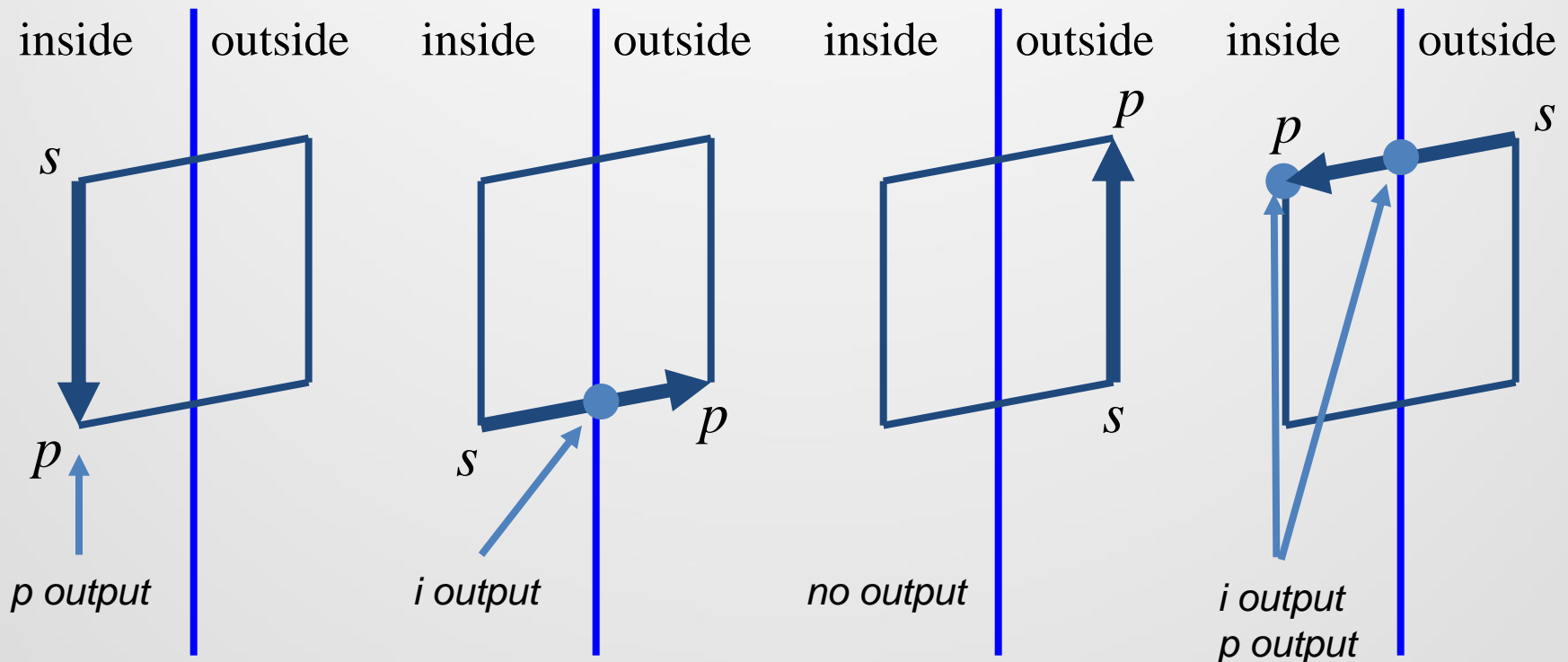
- **Input/output for algorithm:**
 - Input: list of polygon vertices in order
 - Output: list of clipped polygon vertices consisting of old vertices (maybe) and new vertices (maybe)
- **Note: this is exactly what we expect from the clipping operation against each edge**

Sutherland-Hodgman Clipping

- Sutherland-Hodgman basic routine:
 - Go around polygon one vertex at a time
 - Current vertex has position p
 - Previous vertex had position s , and it has been added to the output if appropriate

Sutherland-Hodgman Clipping

- Edge from s to p takes one of four cases:
(Purple line can be a line or a plane)



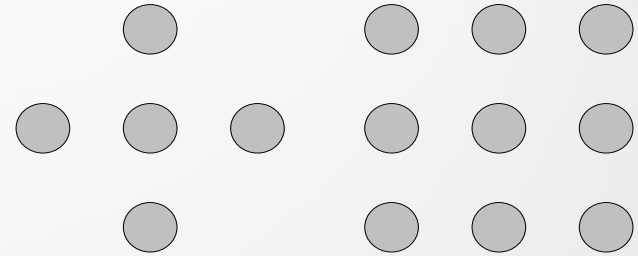
Sutherland-Hodgman Clipping

- Four cases:
 - s inside plane and p inside plane
 - Add p to output
 - Note: s has already been added
 - s inside plane and p outside plane
 - Find intersection point i
 - Add i to output
 - s outside plane and p outside plane
 - Add nothing
 - s outside plane and p inside plane
 - Find intersection point i
 - Add i to output, followed by p

2

GIẢI THUẬT ĐƯỜNG BIÊN

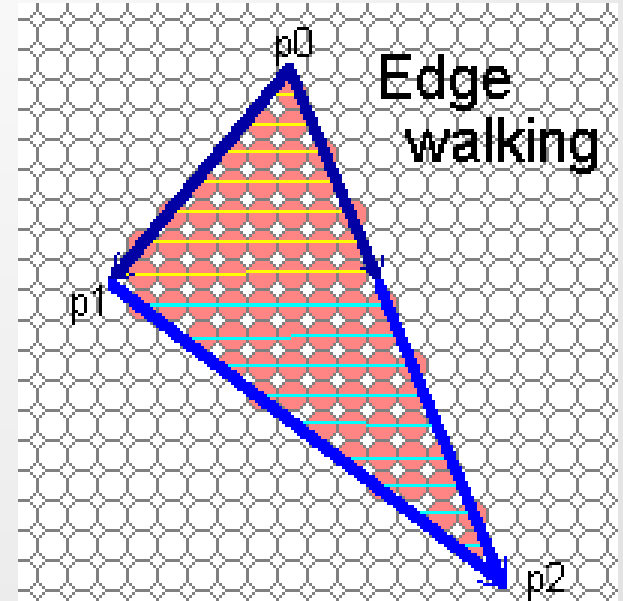
Giải thuật đường biên (Boundary – File algorithm)



```
• Giải_thuật_đường_biên ( x, y )
Color : biến màu
Begin
Color = Readpixel ( x, y );
If ( Color = màu tô ) or ( Color = màu đường biên )
Kết thúc vì chạm biên
hoặc chạm phần đã tô
Else
Putcolor(x,y, mauto)
Giải_thuật_đường_biên ( x+1, y );
Giải_thuật_đường_biên ( x-1, y );
Giải_thuật_đường_biên ( x, y+1 );
Giải_thuật_đường_biên ( x, y-1 );
// Thực hiện lại giải thuật với các điểm lân cận
End.
```

Edge Walking

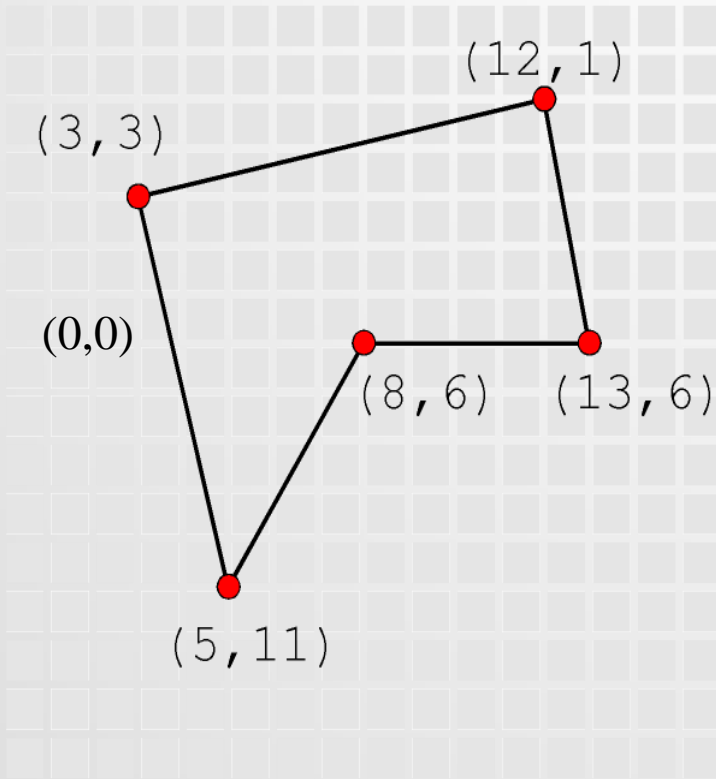
- Ý tưởng
 - Vẽ các cạnh theo chiều dọc
 - Tô các đường nằm bên trong miền theo mỗi đường ngang
 - Nội suy xuống các cạnh ở dưới
 - Tại mỗi đường ngang, nội suy màu sắc của cạnh theo các đường bên trong miền



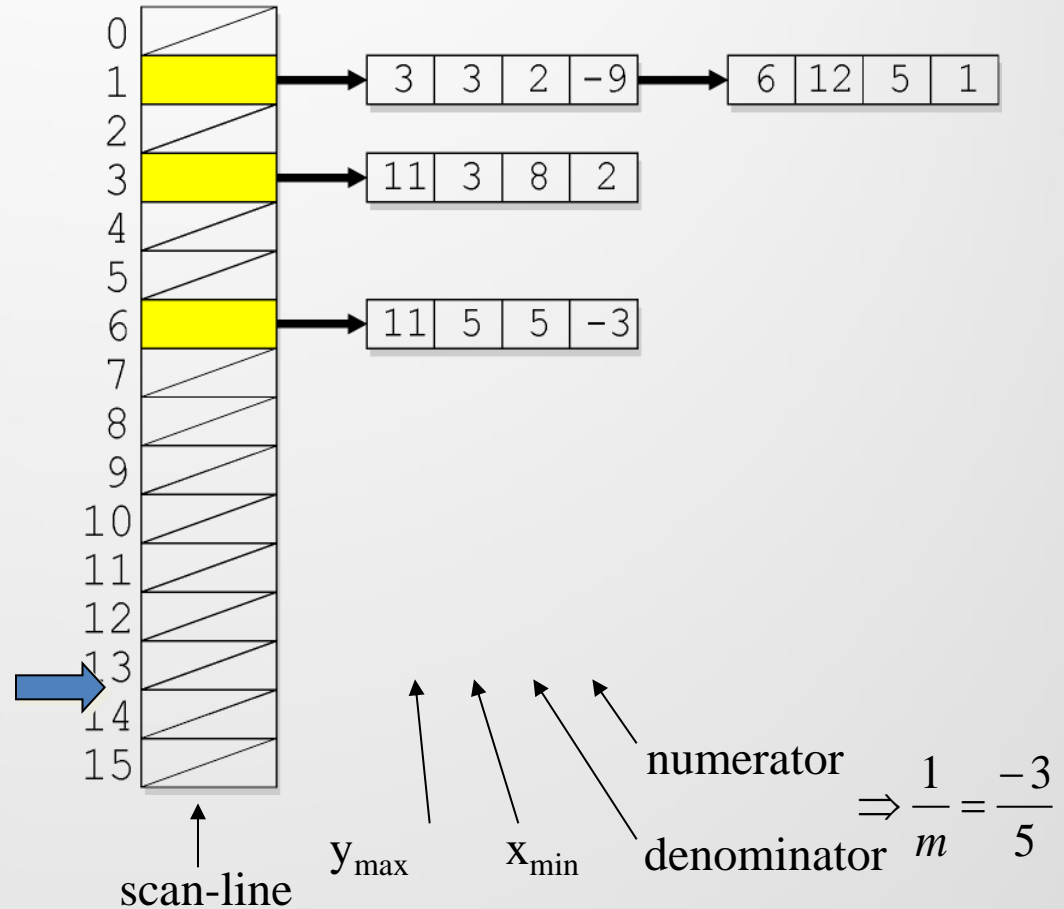
Giải thuật đường quét scan-line algorithm

- Giải thuật đường quét sử dụng *gắn kết cạnh* và các tính toán số nguyên tăng dần cho hiệu quả cao nhất
- Tạo bảng *edge table* (ET) tập của các cạnh đa giác theo thứ tự giá trị y_{min} của chúng
- Tạo bảng active edge table (AET) tập các cạnh giao với đoạn thẳng quét scan-line
- Trong tiến trình quét các cạnh sẽ chuyển từ ET ra AET.
- Các cạnh sẽ ở trong AET cho đến khi giá trị y_{max} của cạnh đạt tới = scanline
- Lúc nay cạnh sẽ bị loại ra khỏi AET.

Edge Table (ET)

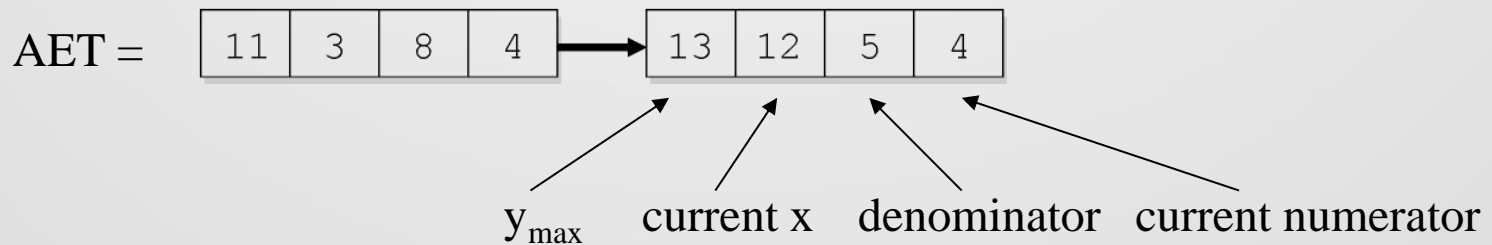
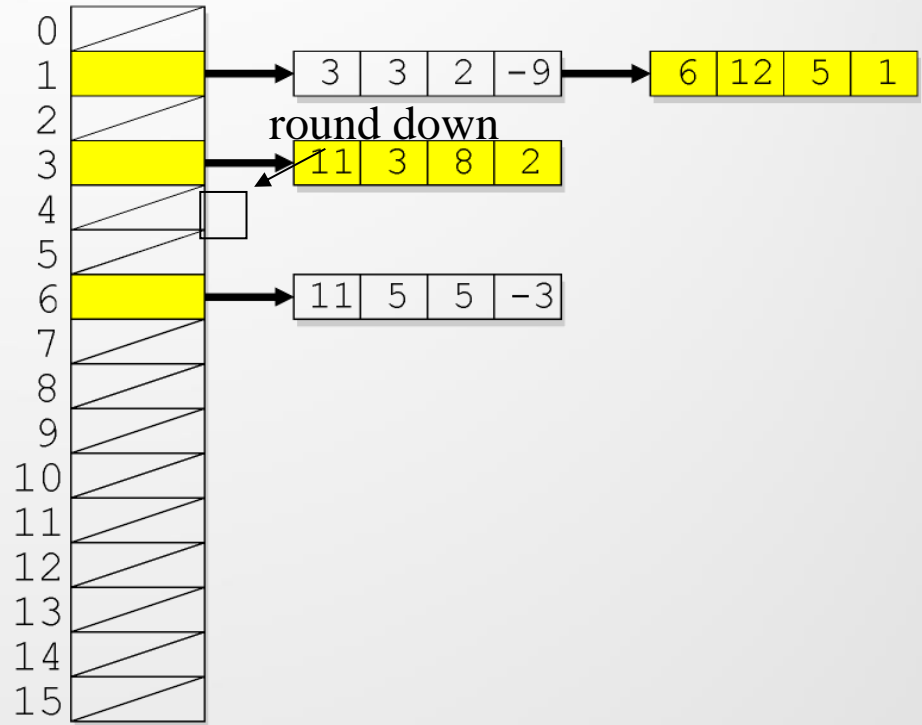
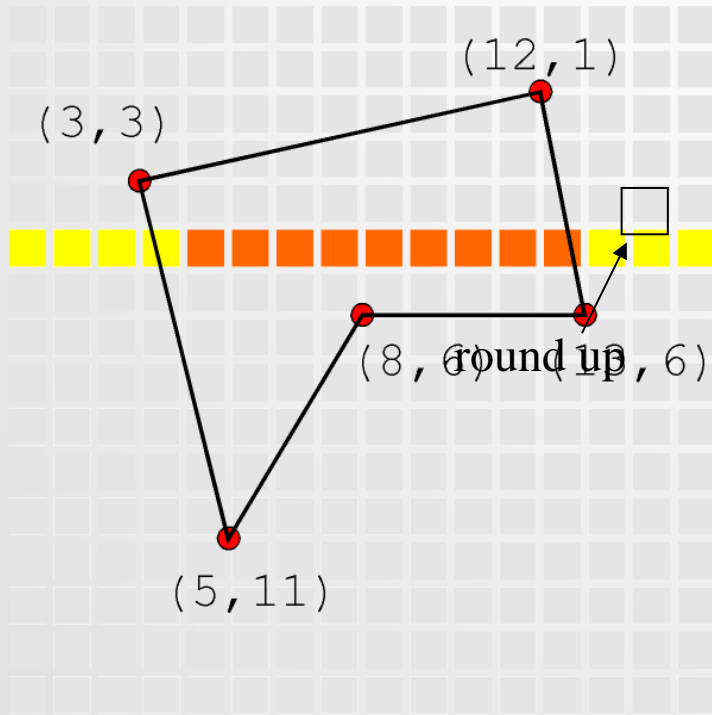


(15,15)

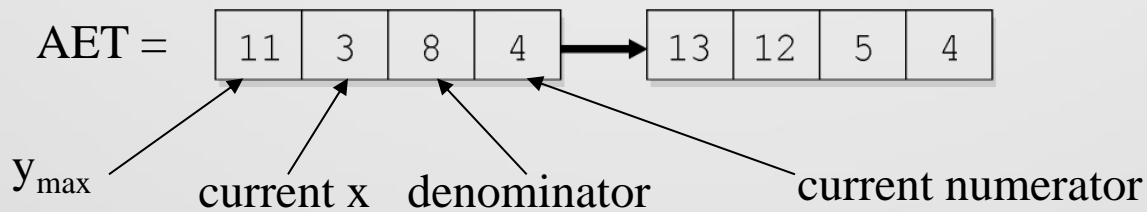
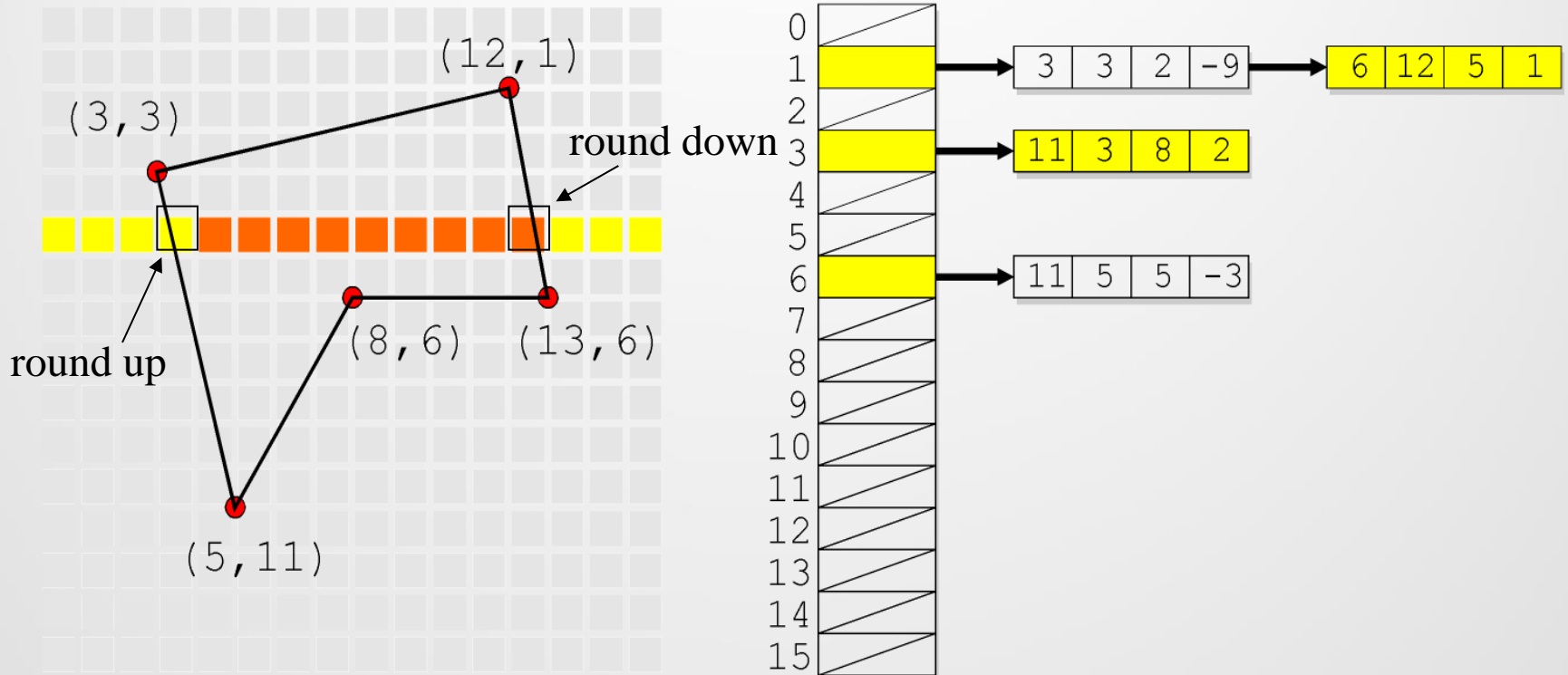


Note: line (8,6) \rightarrow (13,6) has been deleted according to the scan rules

Active Edge Table (AET)



Active Edge Table (AET)



SCAN-LINE ALGORITHM

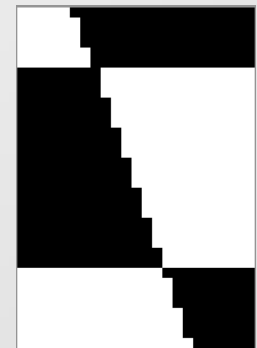
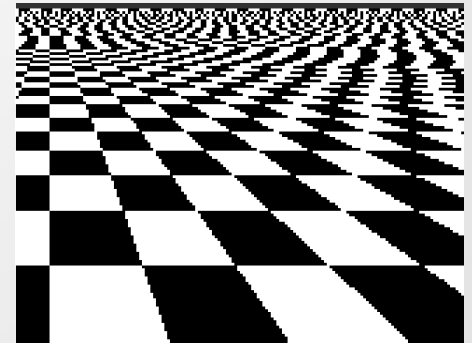
```
y = y of first non empty entry in ET
AET = null
repeat
  move all ET entries in slot y to AET
  sort AET entries according to  $x_{\min}$ 
  fill spans using pairs of AET entries
  for all AET members
    if  $y_{\max} = y$  then remove from AET
  y = y+1
  for all AET members
    update numerator
    if numerator > denominator
      numerator = numerator -
denominator
      x = x+1
until AET and ET empty
```

3

GIẢI THUẬT KHỬ RĂNG CỬA

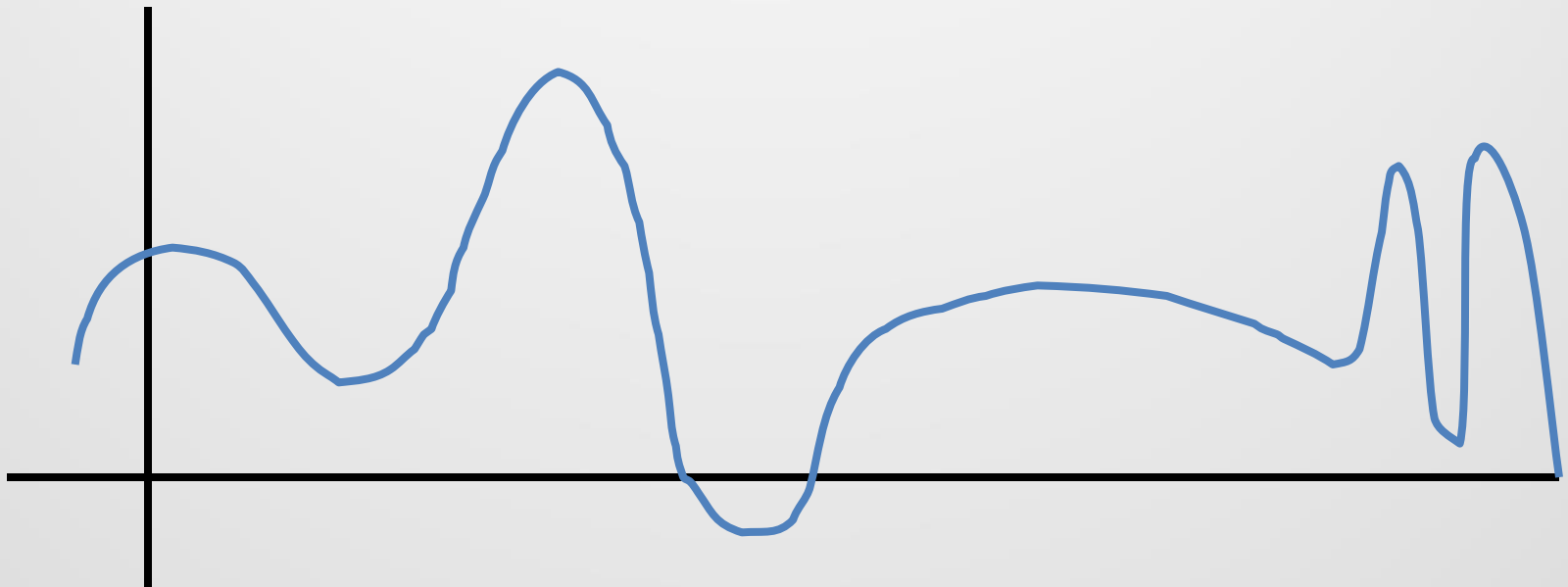
Hiệu ứng răng cưa

- Aliasing: signal processing term with very specific meaning
- Aliasing: computer graphics term for any unwanted visual artifact
- Antialiasing: computer graphics term for avoiding unwanted artifacts

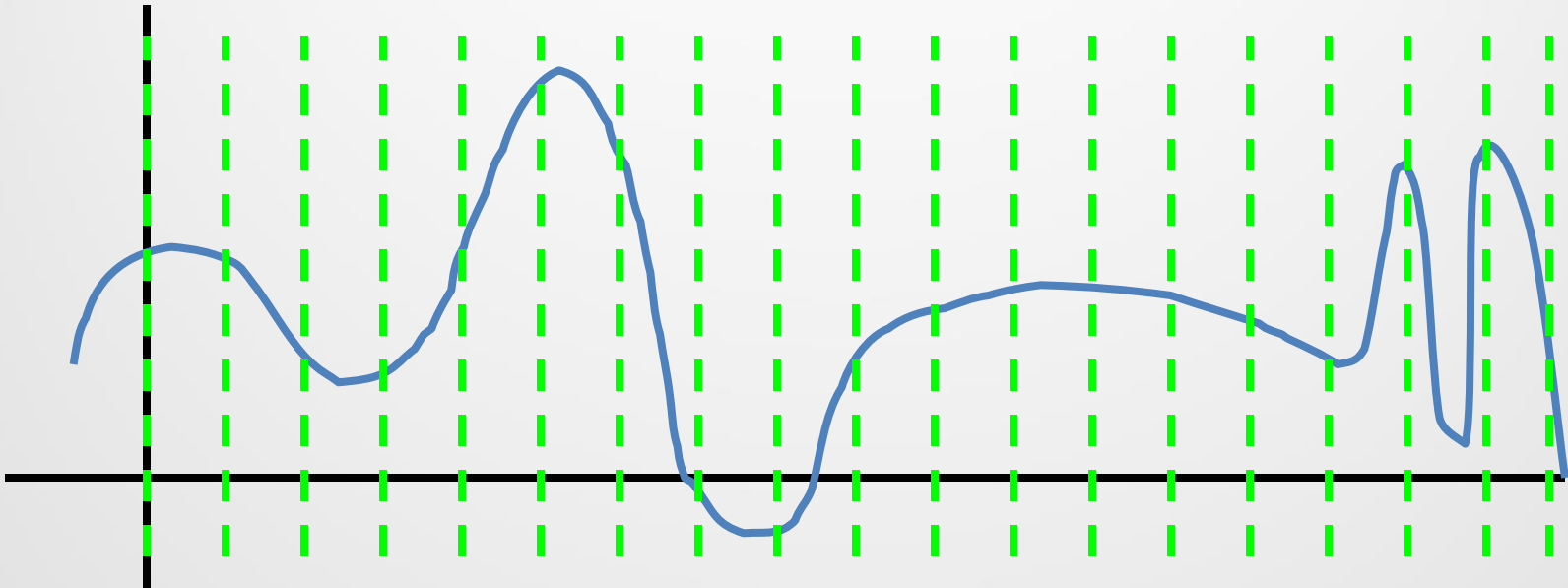


Xử lý tín hiệu

- Rời rạc hóa: Bằng cách lấy mẫu theo chu kỳ nhất định
- Ví dụ với đường sau đây

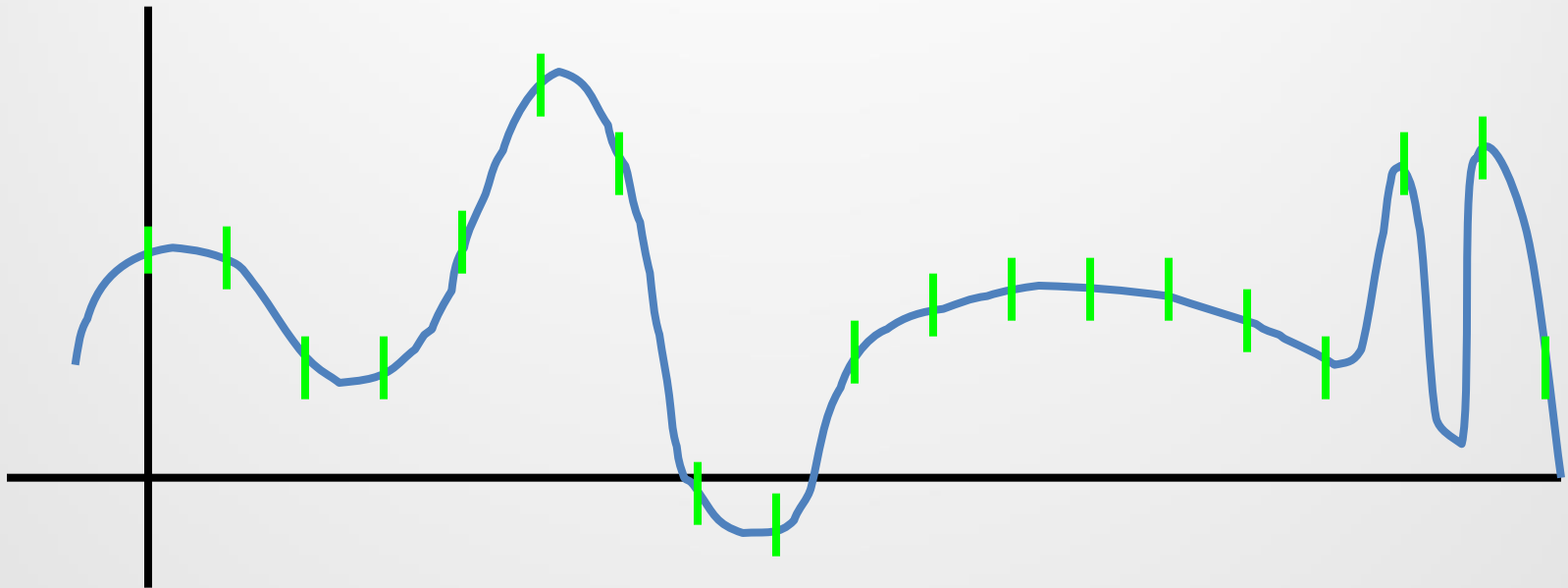


Xử lý tín hiệu



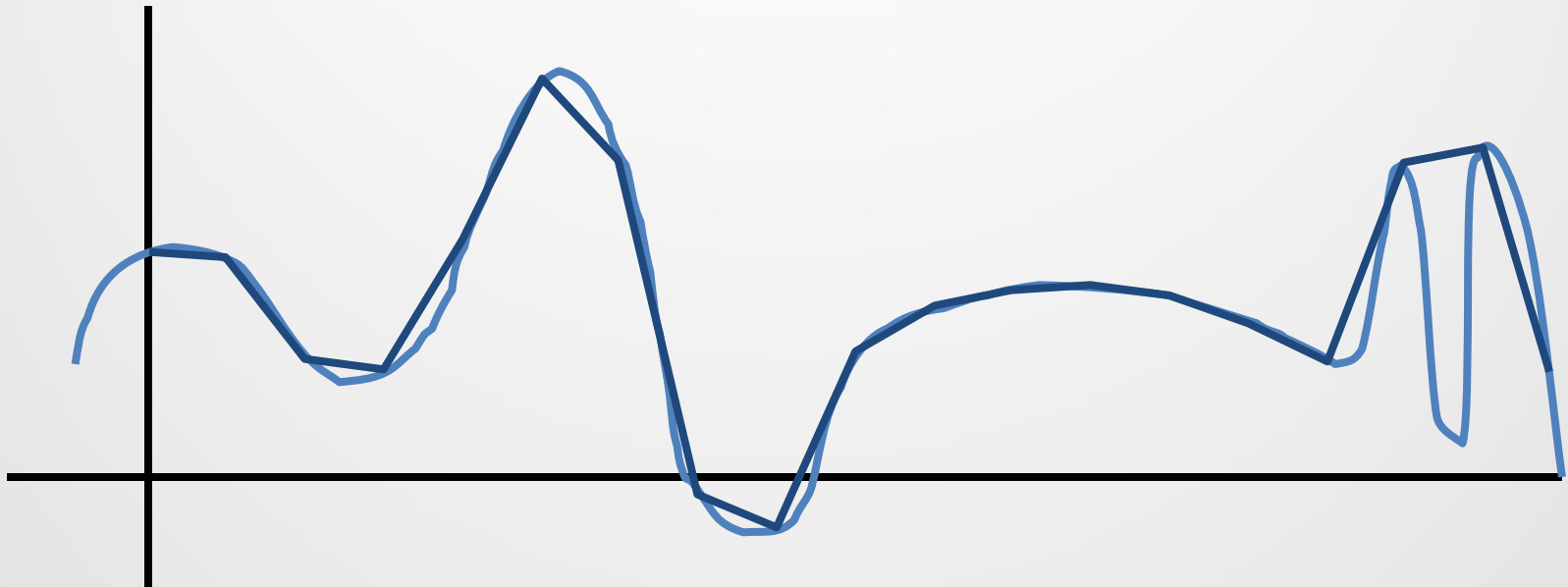
- Lấy mẫu

Xử lý tín hiệu



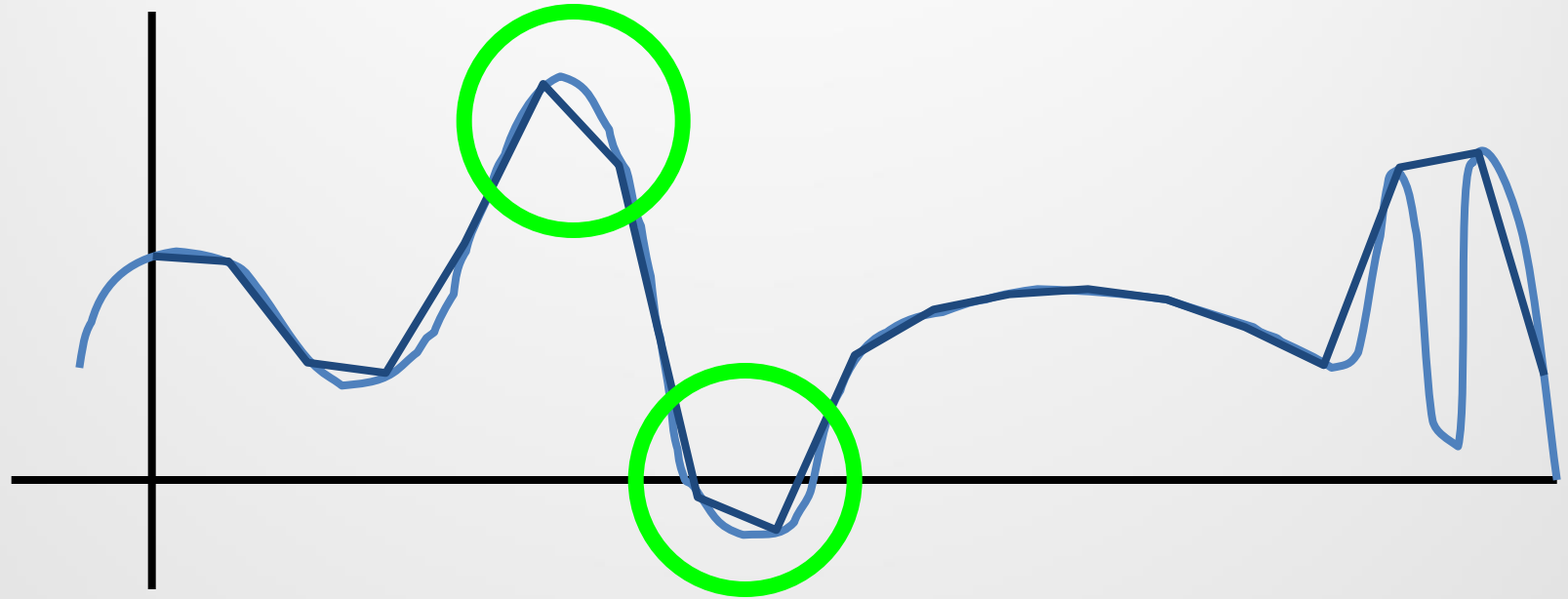
- Kết quả thu được

Xử lý tín hiệu



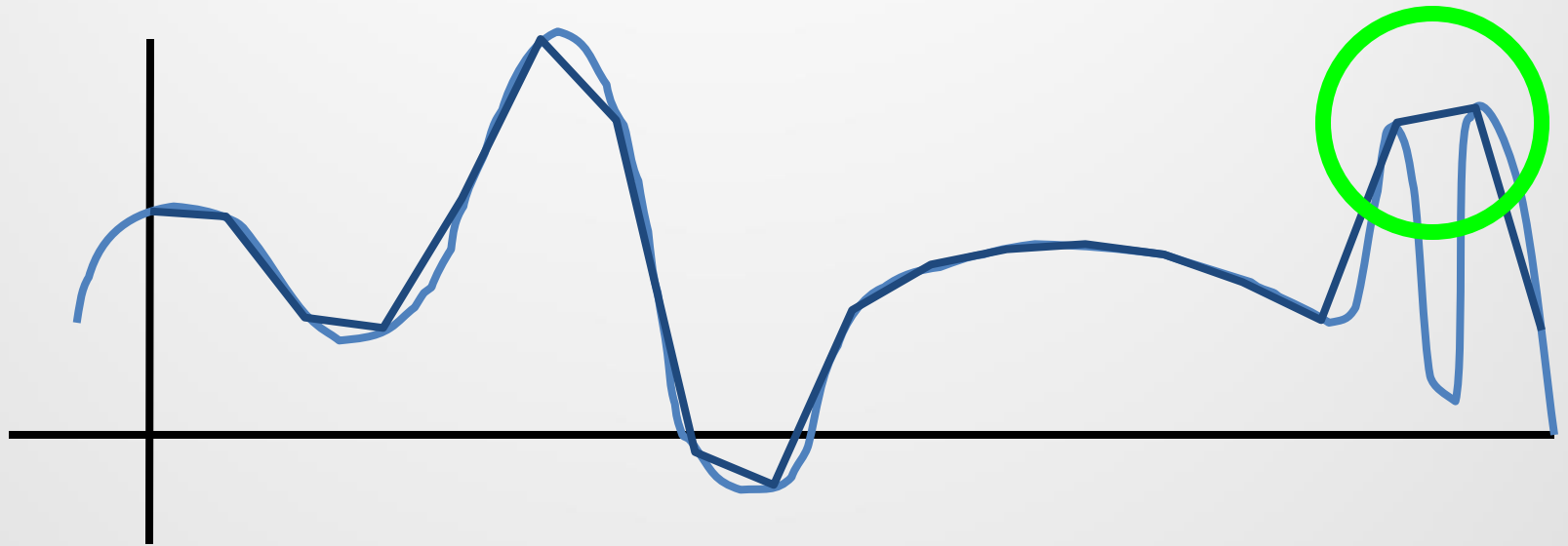
- Kết nối các giá trị thu được

Xử lý tín hiệu



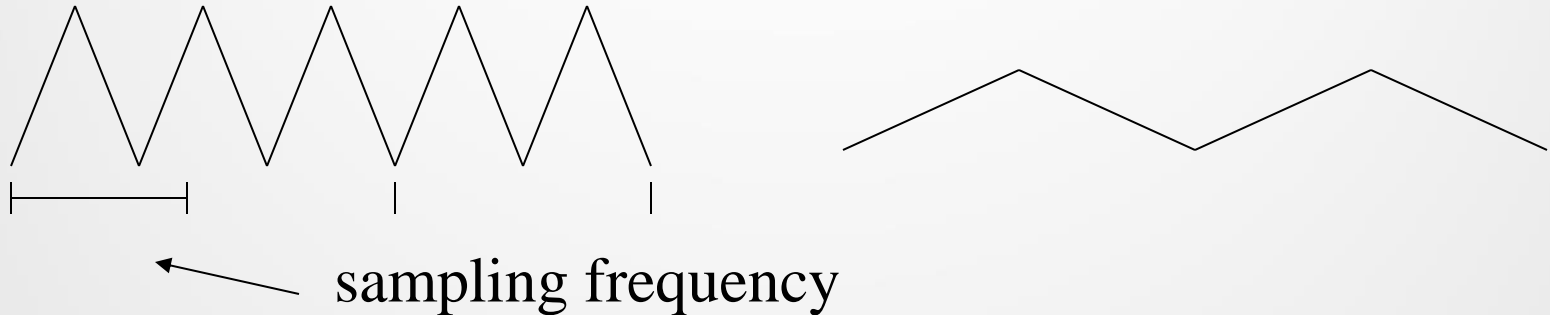
- Một số đoạn bị gãy khúc

Xử lý tín hiệu



- Một số đoạn bị thất thoát dữ liệu

Antialiasing

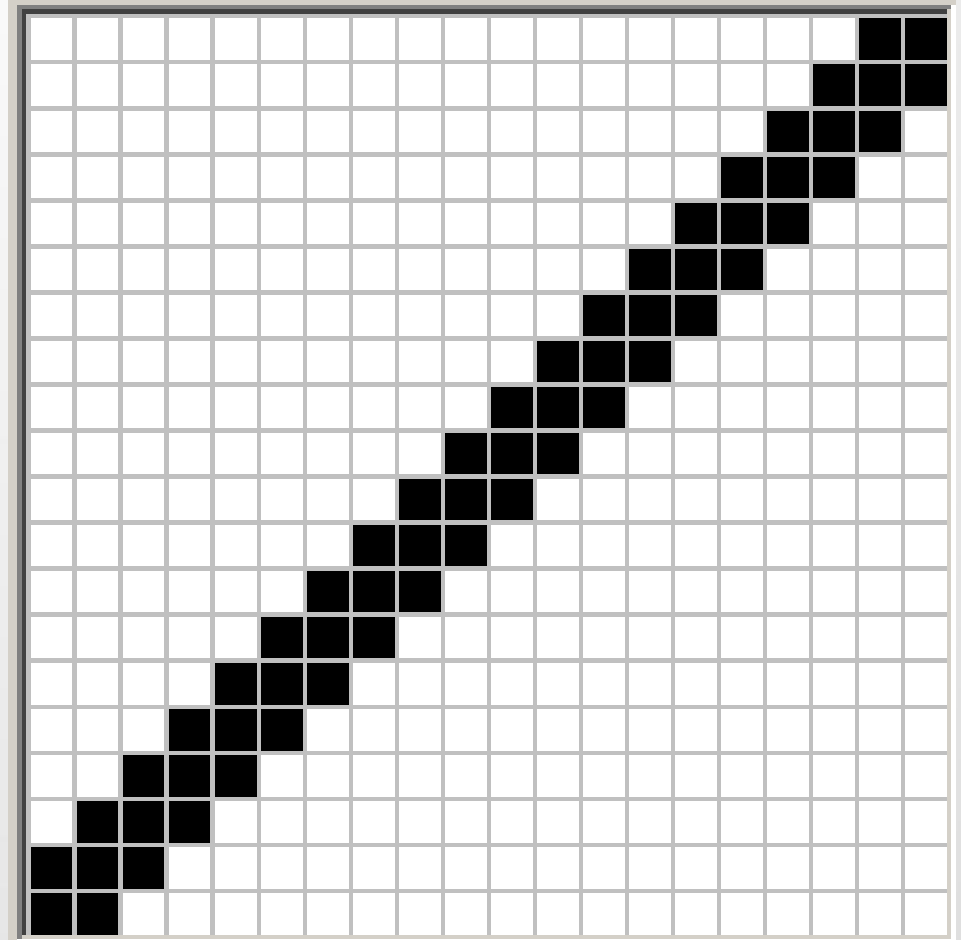
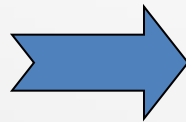
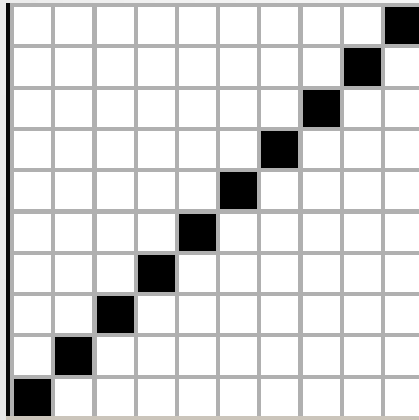


- Méo thông tin trong quá trình lấy mẫu tần số thấp
- Một số trường hợp bị méo thông tin với hiệu ứng bậc thang – staircase effect
- Việc làm giảm hiệu ứng méo thông tin bằng phương pháp bù trừ

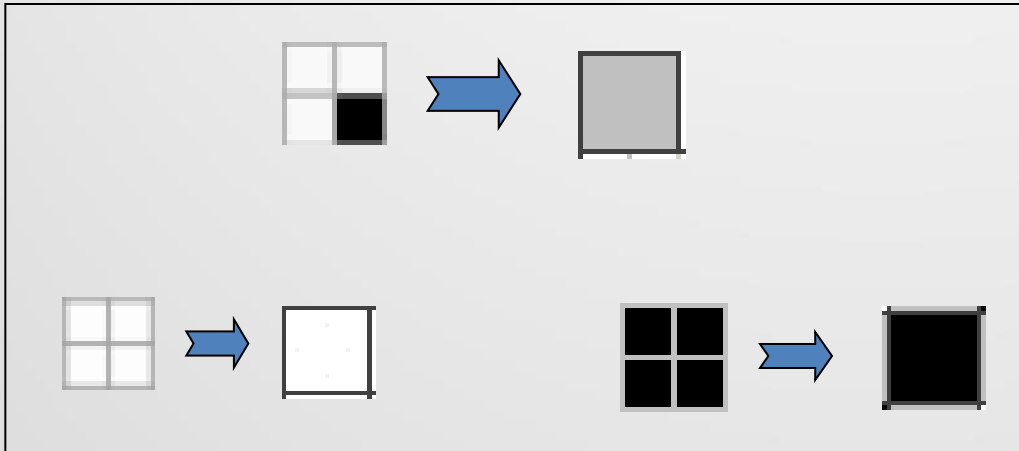
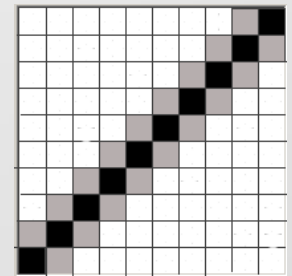
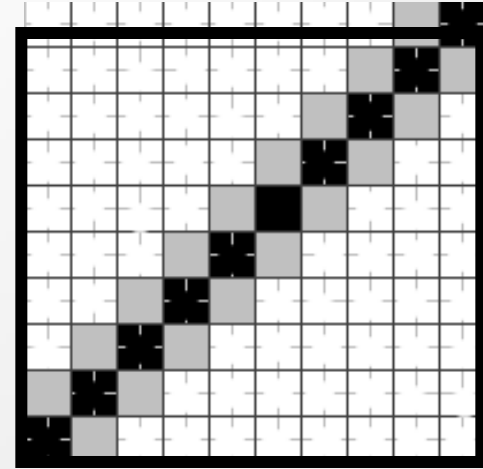
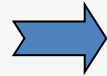
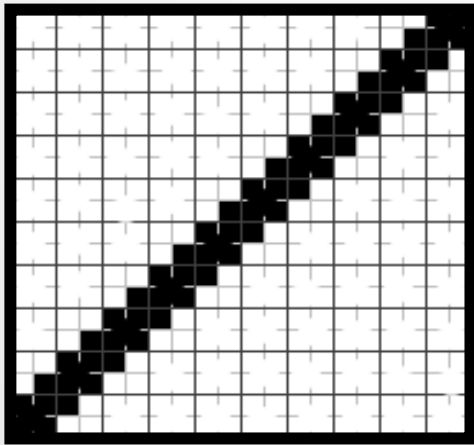
PHƯƠNG PHÁP KHỬ HIỆU ỨNG RẰNG CỬA

1. Cố định tín hiệu bằng phương pháp lọc-prefiltering:
 - Giảm độ rộng dải tần tín hiệu bởi bộ lọc thấp hơn trước khi lấy mẫu.
 - Chất lượng cao nhất nhưng không thực tiễn
2. Cố định mẫu bằng siêu mẫu supersampling:
 - Dùng nhiều mẫu hơn để tăng tần số
 - Đơn giản và được sử dụng rộng rãi
3. Cố định mẫu bằng phương pháp mẫu bất kỳ
 - Mẫu ngẫu nhiên nhưng không đồng nhất
 - Tương đối đơn giản, thường được kết hợp với phương pháp siêu mẫu

PHƯƠNG PHÁP SIÊU MẪU



Antialiasing by
supersampling



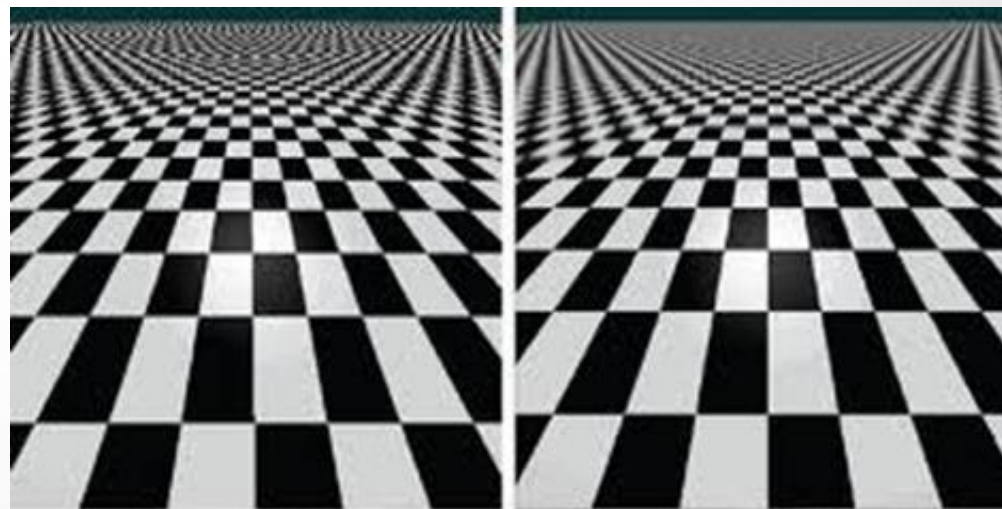
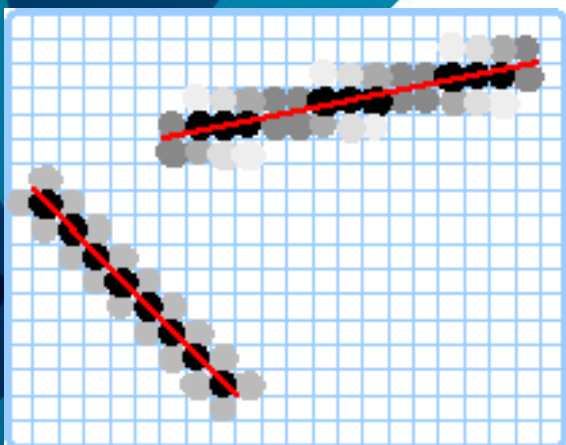
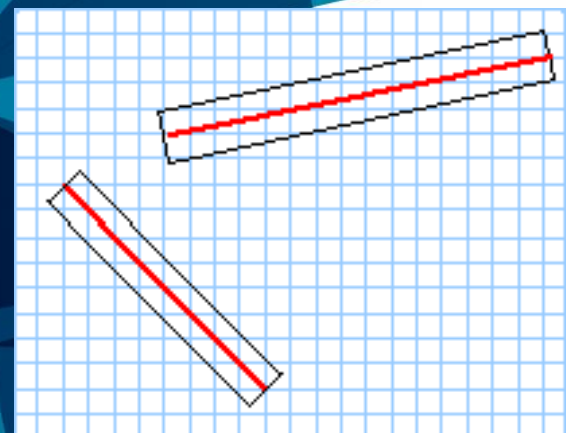
VÍ DỤ

a

Alias

a

Anti-aliased

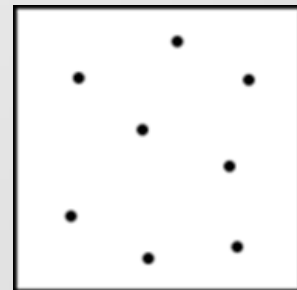
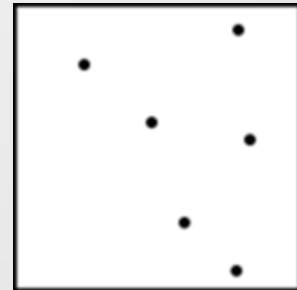
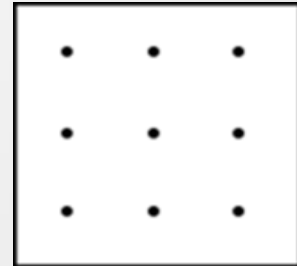


aliasing effects

anti-aliasing by over-sampling

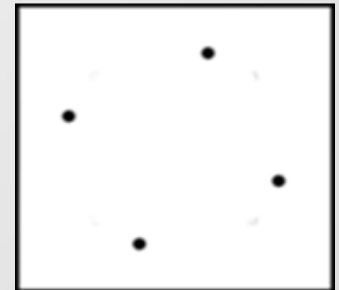
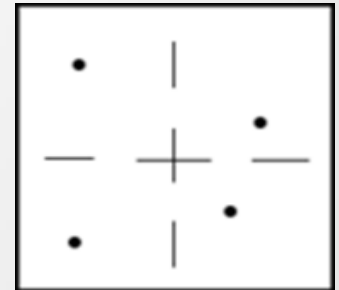
Supersampling patterns

- Grid : The simplest [algorithm](#). The pixel is split in several sub-pixels, and a sample is taken from the center of each. It is fast and easy to implement, although due to the regular nature of sampling, aliasing can still occur if a low number of sub-pixels is used.
- Random: Also known as stochastic sampling, it avoids the regularity of grid supersampling. However, due to the irregularity of the pattern, samples end up being unnecessary in some areas of the pixel and lacking in others.
- Poisson disc: Again an algorithm that places the samples randomly, but then checks that any two are not too close. The end result is even but random distribution of samples. Unfortunately, the computational time required for this algorithm is too great to justify its use in real-time rendering, unless the sampling itself is computationally expensive compared to the positioning the sample points or the sample points are not repositioned for every single pixel.



Supersampling patterns

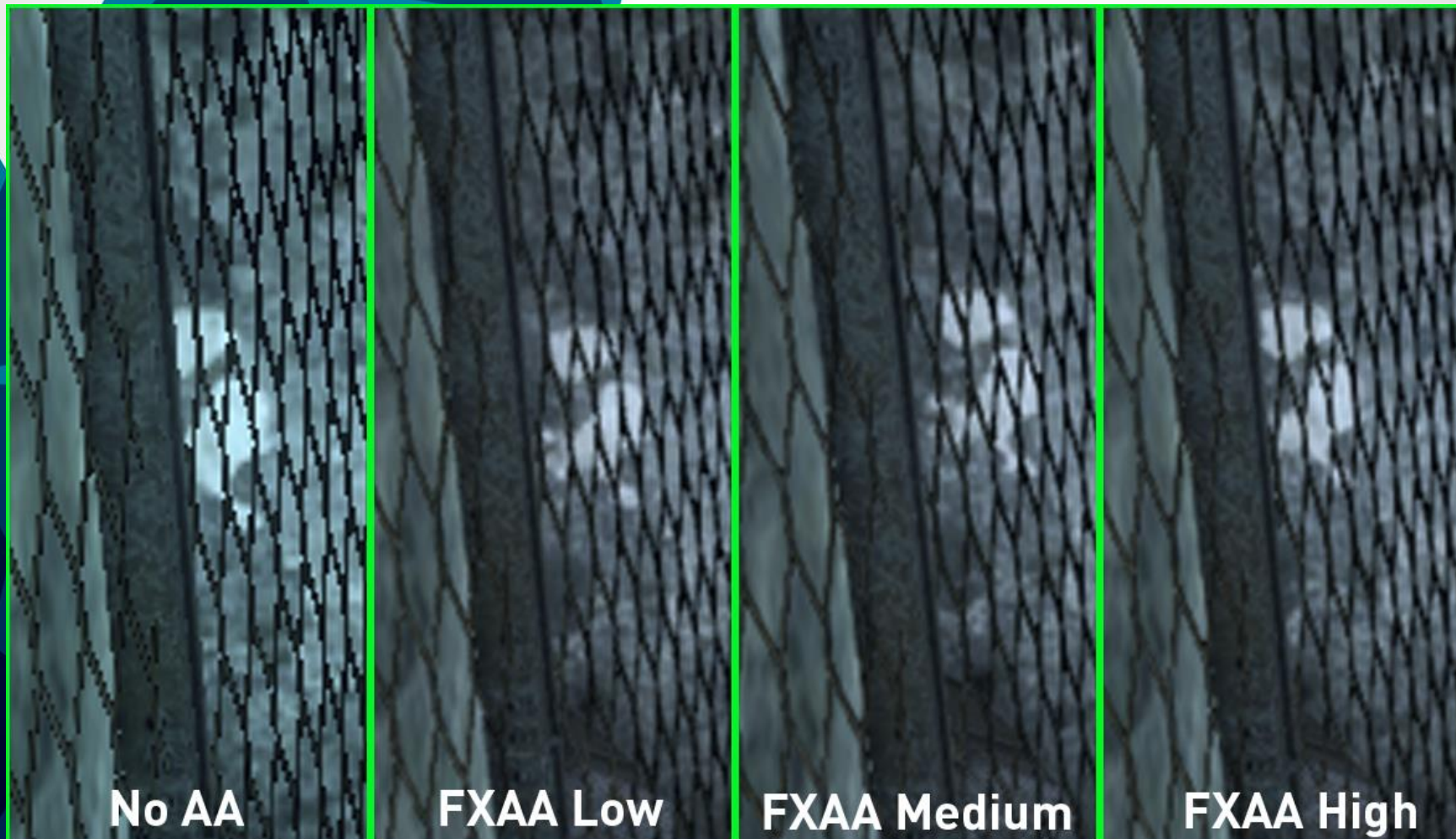
- Jittered: A modification of the grid algorithm to approximate the Poisson disc. A pixel is split into several sub-pixels, but a sample is not taken from the center of each, but from a random point within the sub-pixel. Congregation can still occur, but to a lesser degree.
- Rotated grid: A 2×2 grid layout is used but the sample pattern is rotated to avoid samples aligning on the horizontal or vertical axis greatly improving antialiasing quality for the most commonly encountered cases. For an optimal pattern, the rotation angle is $\arctan(1/2)$ (about 26.6 degrees) and the square is stretched by a factor of $\sqrt{5}/2$



VÍ DỤ



VÍ DỤ



VÍ DỤ

AA OFF



32xCSSAA ON

