

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC NHA TRANG  
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC TẬP CƠ SỞ  
CÀI ĐẶT THUẬT TOÁN XÉN TỈA ĐA GIÁC  
BẰNG THUẬT TOÁN SUTHERLAND - HODGMAN**

**Giảng viên hướng dẫn: ThS. Đoàn Vũ Thịnh**

**Sinh viên thực hiện: Trần Đình Trí**

**Mã số sinh viên: 60131750**

**KHÁNH HÒA - 2021**

**TRƯỜNG ĐẠI HỌC NHA TRANG  
KHOA CÔNG NGHỆ THÔNG TIN  
BỘ MÔN KỸ THUẬT PHẦN MỀM**



**BÁO CÁO THỰC TẬP CƠ SỞ  
CÀI ĐẶT THUẬT TOÁN XÉN TỈA ĐA GIÁC  
BẰNG THUẬT TOÁN SUTHERLAND - HODGMAN**

Giảng viên hướng dẫn: ThS. Đoàn Vũ Thịnh

Sinh viên thực hiện: Trần Đình Trí

Mã số sinh viên: 60131750

Khánh Hòa, tháng 01/2021

**TRƯỜNG ĐẠI HỌC NHA TRANG**  
**Khoa: Công nghệ Thông tin**

**PHIẾU THEO DÕI TIẾN ĐỘ VÀ ĐÁNH GIÁ BÁO CÁO THỰC TẬP CƠ SỞ**  
**Tên đề tài: CÀI ĐẶT THUẬT TOÁN XÉN TỈA ĐA GIÁC BẰNG THUẬT TOÁN**  
**SUTHERLAND - HODGMAN**

**Giảng viên hướng dẫn:** ThS. Đoàn Vũ Thịnh

**Sinh viên được hướng dẫn:** Trần Đình Trí

**MSSV:** 60131750

**Khóa:** 60

**Ngành:** Công nghệ Thông tin

Lần	Ngày	Nội dung	Nhận xét của GVHD
<b>1</b>	<b>7/12/2020</b>	Nhận đề tài hướng dẫn và định hướng giải quyết vấn đề. Sinh viên trình bày kế hoạch thực hiện.	Sinh viên và GVHD trao đổi nội dung của đề tài. Phân chia công việc theo từng thời gian sao cho phù hợp với yêu cầu.
<b>2</b>	<b>14/12/2020</b>	Sinh viên trình bày việc mô phỏng thuật toán chính dựa trên kiến thức đã được học ở môn kỹ thuật đồ họa và các kiến thức thu nhận được từ Internet để minh họa bài toán đa dạng nhất có thể.	Sinh viên hiểu được vấn đề cần phải thực hiện và có giải pháp cho từng vấn đề cụ thể. Tuy nhiên một vấn đề hoàn toàn mới là sử dụng thư viện của chuột thì chưa làm lần nào nên cần đầu tư nhiều thời gian hơn.
<b>3</b>	<b>21/12/2020</b>	Sinh viên hoàn thiện các thuật toán đã đề ra với dữ liệu đầu vào được nhập từ bàn phím. Trình bày thuật toán với các trường hợp sai và chỉ ra được hướng khắc phục cho các trường hợp đó.	Sinh viên hiểu nội dung của thuật toán khá chi tiết với các trường hợp hạn chế của thuật toán. Việc lập trình cũng hoàn thành ở mức độ nhập dữ liệu từ bàn phím nhưng vấn đề với chuột thì chưa thực hiện được.
<b>4</b>	<b>4/01/2021</b>	Sinh viên nộp bản thảo của báo cáo thực tập lần thứ 1 và tiến hành chỉnh sửa.	Báo cáo chỉ trình bày chung chung chưa đi vào cụ thể phân tích các yêu cầu của bài toán,

			hình ảnh, bảng biểu chưa trình bày rõ ràng. Cần hiệu chỉnh theo yêu cầu của GVHD.
<b>5</b>	<b>11/01/2021</b>	Sinh viên nộp bản thảo lần 2 và có minh họa với thư viện của chuột nhưng chưa thể kết nối với phần trước.	Báo cáo lần này đã khắc phục được các lỗi của lần trước, tuy nhiên phần phương pháp và kết quả chưa nổi bật, chưa có sự liên kết giữa các phần.
<b>6</b>	<b>18/1/2021</b>	Sinh viên nộp bản thảo lần cuối sau khi đã chỉnh sửa các yêu cầu như đã đề ra.	Sinh viên nghiêm túc chỉnh sửa báo cáo theo định hướng của GVHD.

**Nhận xét chung (sau khi sinh viên hoàn thành ĐA/KL):**

Sinh viên thực hiện tốt các yêu cầu của GVHD, trong quá trình thực hiện đề tài có sự liên hệ chặt chẽ với GV. Theo lịch hẹn Sinh viên đều có mặt để trình bày ý tưởng của các nội dung lần trước. Trong quá trình hoàn tất báo cáo đều nỗ lực không ngừng mặc dù đang cao điểm của đợt thi học kỳ nhưng SV vẫn dành thời gian không ít cho TTCS.

Về nội dung báo cáo đã thỏa mãn các yêu cầu của đề tài như trong đề cương. Về kết quả chương trình đã minh họa được thuật toán. Về các yêu cầu cao hơn như sử dụng thư viện của chuột hay minh họa các trường hợp nhược điểm của thuật toán thì chưa thực hiện thành công.

Về hình thức của báo cáo và sản phẩm, báo cáo trình bày rõ ràng các mục tiêu, phương pháp, kết quả và thảo luận cho sản phẩm. Còn về sản phẩm như đã trình bày có phần hạn chế.

Điểm hình thức: 8.0/10 Điểm nội dung: 8.0/10 **Điểm tổng kết: 8.0/10**

Đồng ý cho sinh viên: Được bảo vệ: ☒ Không được bảo vệ: ☐

Khánh Hòa, ngày 20 tháng 01 năm 2021

Cán bộ hướng dẫn

(Ký và ghi rõ họ tên)

## LỜI CẢM ƠN

Để có thể hoàn thành đợt thực tập lần này, chúng em xin chân thành cảm ơn đến quý thầy cô khoa Công nghệ Thông tin đã tạo điều kiện hỗ trợ và giúp đỡ chúng em trong quá trình học tập và nghiên cứu đề tài này.

Qua đây, nhóm xin chân thành cảm ơn thầy Đoàn Vũ Thịnh, người đã trực tiếp quan tâm và hướng dẫn chúng em hoàn thành tốt đợt thực tập trong thời gian qua.

Do kiến thức còn hạn chế và thời gian thực hiện còn ngắn nên bài báo cáo của chúng em còn nhiều thiếu sót, kính mong sự góp ý của quý thầy cô.

## MỤC LỤC

LỜI CẢM ƠN.....	iii
DANH MỤC HÌNH .....	v
TÓM TẮT.....	vi
PHẦN 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1. Thuật toán xén tia.....	1
1.2. Thuật toán xén tia đoạn thẳng .....	1
1.2.1 Thuật toán Cohen-Sutherland .....	2
1.2.2 Thuật toán Cyrus-beck.....	4
1.3 Thuật toán xén tia đa giác .....	5
PHẦN 2. PHƯƠNG PHÁP NGHIÊN CỨU .....	10
2.1. Đoạn thẳng nằm trên biên trái cửa sổ cắt .....	10
2.2. Đoạn thẳng nằm trên biên phải cửa sổ xén .....	11
2.3 Đoạn thẳng nằm trên biên trên cửa sổ cắt .....	12
2.4 Đoạn thẳng nằm trên biên dưới cửa sổ cắt.....	13
2.5 Đoạn thẳng nằm ngoài cửa sổ xén .....	14
2.6 Đoạn thẳng nằm trong cửa sổ xén.....	14
PHẦN 3. KẾT QUẢ ĐẠT ĐƯỢC .....	15
THẢO LUẬN .....	16
TÀI LIỆU THAM KHẢO .....	17

## DANH MỤC HÌNH

Hình 1.1. Ví dụ minh họa cho thuật toán xén tia đa giác.....	1
Hình 1.2. Các đoạn thẳng nằm trong hoặc nằm ngoài .....	1
Hình 1.3. Các đoạn thẳng cắt cửa sổ.....	2
Hình 1.4. Mã vùng quy định vị trí tương đối của vùng so với cửa sổ .....	3
Hình 1.5. Mã vùng được xác định theo 9 vùng.....	3
Hình 1.6. Cửa sổ cắt tia đoạn thẳng .....	5
Hình 2.1. Nhập tọa độ đa giác và cửa sổ cắt.....	10
Hình 2.2. Đoạn thẳng nằm trên biên trái cửa sổ cắt (TH1).....	10
Hình 2.3. Đoạn thẳng nằm trên biên trái cửa sổ cắt (TH2).....	11
Hình 2.4. Đoạn thẳng nằm trên biên phải cửa sổ cắt (TH1) .....	11
Hình 2.5. Đoạn thẳng nằm trên biên phải cửa sổ cắt (TH2) .....	12
Hình 2.6. Đoạn thẳng nằm trên biên trên cửa sổ cắt (TH1).....	12
Hình 2.7. Đoạn thẳng nằm trên biên trên cửa sổ cắt (TH2).....	13
Hình 2.8. Đoạn thẳng nằm trên biên dưới cửa sổ cắt (TH1).....	13
Hình 2.9. Đoạn thẳng nằm trên biên dưới cửa sổ cắt (TH2).....	13
Hình 2.10. Đoạn thẳng nằm ngoài cửa sổ cắt .....	14
Hình 2.11. Đoạn thẳng nằm trong cửa sổ cắt.....	14
Hình 3.3. Minh họa trường hợp đa giác nằm hoàn toàn bên trong cửa sổ cắt.....	15
Hình 4.1. Hạn chế của thuật toán Sutherland-Hodgman .....	16

## TÓM TẮT

Đồ họa máy tính là một lĩnh vực của khoa học máy tính nghiên cứu về cơ sở toán học, các thuật toán cũng như các kỹ thuật để cho phép tạo, hiển thị và điều khiển hình ảnh trên màn hình máy tính. Đồ họa máy tính có liên quan ít nhiều đến một số lĩnh vực như đại số, hình học giải tích, hình học họa hình, quang học,... và kỹ thuật máy tính, đặc biệt là chế tạo phần cứng (các loại màn hình, các thiết bị xuất, nhập, các vi mạch đồ họa). Kỹ thuật xén tia đa giác là một trong số rất nhiều thuật toán quan trọng của ngành đồ họa máy tính và các phần mềm thương mại hay các thiết bị xử lý đồ họa đều vận dụng các thuật toán này. Thuật toán xén tia đa giác với giải thuật Sutherland –Hodgman được cài đặt trong lần thực tập lần này giúp hiểu rõ hơn về một trong những thuật toán cơ bản của ngành đồ họa máy tính. Quy trình thực hiện được trải qua các bước từ thiết kế giao diện, cài đặt thuật toán, hiển thị kết quả đầu ra trên màn hình hiển thị đều được thực hiện trên môi trường C++ thông qua ứng dụng DevC/C++ có kết hợp với thư viện graphics.h. Kết quả của việc cài đặt thuật toán đáp ứng được các yêu cầu đặt ra của đợt thực tập cơ sở lần này.

Toàn bộ mã nguồn của báo cáo được tải lên theo địa chỉ:

<https://github.com/thinhdoanvu/ThuctapCoSo2020/tree/main/TranDinhTri/>

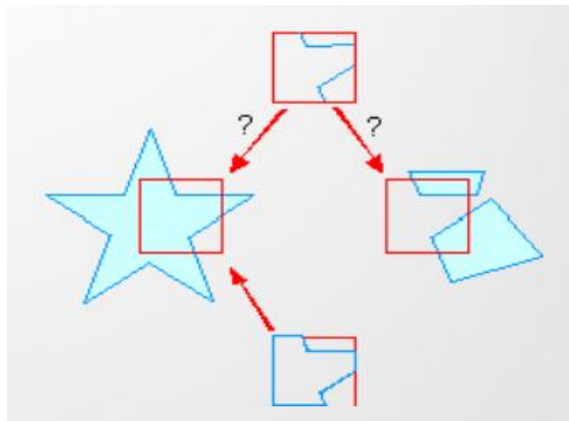


## PHẦN 1. GIỚI THIỆU ĐỀ TÀI

Cùng với sự phát triển kinh tế hiện nay là sự phát triển mạnh mẽ của ngành công nghệ thông tin. Các ứng dụng của công nghệ thông tin ngày càng nhiều và có mặt hầu hết trong các lĩnh vực cuộc sống. Với đề tài: **“Cài đặt thuật toán xén tia đa giác bằng thuật toán Sutherland-Hogman”**, ý nghĩa của thuật toán xén hình trong công nghệ thông tin là không hề nhỏ, đặc biệt đối với màn hình máy tính thì nó có ý nghĩa vô cùng quan trọng.

### 1.1. Thuật toán xén tia

Xén tia là tiến trình xác định các điểm của một đối tượng nằm trong hay ngoài cửa sổ hiển thị. Nằm trong được hiển thị, nằm ngoài loại bỏ. Việc loại từng điểm ảnh của đối tượng thường chậm nhất là khi đối tượng mà phần lớn nằm ngoài cửa sổ hiển thị.

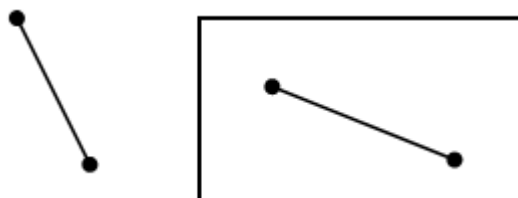


Hình 1.1. Ví dụ minh họa cho thuật toán xén tia đa giác

Nhưng trước hết muốn xén tia một đa giác thì chúng ta nên tìm hiểu cách xén tia một đoạn thẳng bởi vì xén tia đoạn thẳng là nền tảng để chúng ta đi xén tia một đa giác.

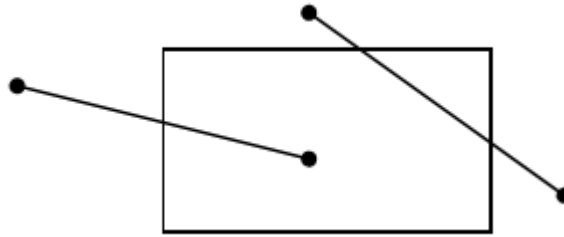
### 1.2. Thuật toán xén tia đoạn thẳng

Các đoạn thẳng không cắt cửa sổ thị: nằm trong hoàn toàn, hoặc ngoài hoàn toàn.



Hình 1.2. Các đoạn thẳng nằm trong hoặc nằm ngoài

Nếu đoạn thẳng đi qua 2 cạnh bất kỳ của cửa sổ cắt thì phân thành 2 đoạn: phần nằm trong và phần nằm ngoài cửa sổ cắt.



Hình 1.3. Các đoạn thẳng cắt cửa sổ

Các đoạn thẳng có thể rơi vào các trường hợp sau:

**TH1:** Hiển thị (visible): cả hai đầu cuối của đoạn thẳng đều nằm bên trong cửa sổ.

**TH2:** Không hiển thị (invisible): đoạn thẳng xác định nằm ngoài cửa sổ. Điều này xảy ra khi đoạn thẳng từ  $(x_1, y_1)$  đến  $(x_2, y_2)$  thỏa mãn bất kỳ một trong bốn bất đẳng thức sau:

$$\begin{aligned} x_1, x_2 > x_{\max} \quad y_1, y_2 > y_{\max} \\ x_1, x_2 < x_{\min} \quad y_1, y_2 < y_{\min} \end{aligned}$$

Thao tác xén hình là một trong những thao tác cơ bản của quá trình hiển thị đối tượng. Ý tưởng chung của thao tác xén tia đoạn thẳng đó là phép toán tìm giao điểm giữa đoạn thẳng với biên. Với các đoạn thẳng đặt biệt như nằm trong hoàn toàn hoặc nằm ngoài hoàn toàn cửa sổ (Hình 1.3, 1.4). Đối với các đoạn thẳng cắt cửa sổ thì phải tìm giao điểm.

Người ta thường sử dụng phương trình tham số của đoạn thẳng trong việc tìm giao điểm của đoạn thẳng với cửa sổ.

$$\begin{aligned} X &= x_1 + t(x_2 - x_1) = x_1 + tDx_1, \quad Dx_1 = x_2 - x_1 \\ Y &= y_1 + t(y_2 - y_1) = y_1 + tDy_1, \quad Dy_1 = y_2 - y_1 \quad 0 \leq t \leq 1 \end{aligned}$$

### 1.2.1 Thuật toán Cohen-Sutherland

Đây là một trong những thuật toán ra đời sớm nhất. Bằng cách kéo dài các biên của cửa sổ, người ta chia mặt phẳng thành chín vùng gồm cửa sổ và 8 vùng xung quanh.

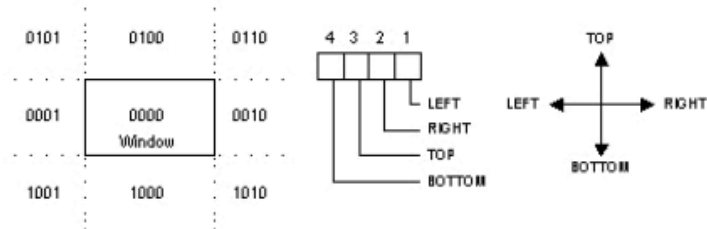
**Khái niệm mã vùng:** Mã vùng là một con số 4 bit nhị phân sẽ được gán cho mỗi vùng để mô tả vị trí tương đối của vùng đó so với cửa sổ. Bằng cách đánh số từ 1 đến 4 theo thứ tự từ phải sang trái, các bit theo mã vùng được dùng theo quy ước sau để chỉ một trong bốn vị trí tương đối của vùng so với cửa sổ bao gồm: trái, phải, trên, dưới.

**Bit 1:** trái (LEFT)

**Bit 2:** phải (RIGHT)

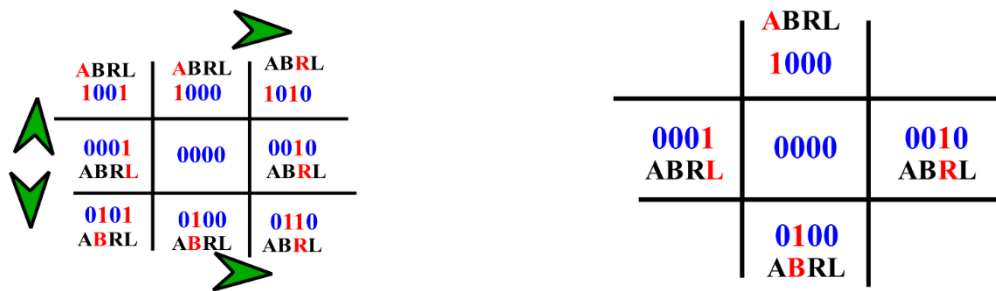
**Bit 3:** trên (TOP)

**Bit 4:** dưới (BOTTOM)



Hình 1.4. Mã vùng quy định vị trí tương đối của vùng so với cửa sổ

Mã vùng được xác định theo 9 vùng của mặt phẳng mà các điểm cuối nằm vào đó. Một bit được cài đặt true (1) hoặc false (0).

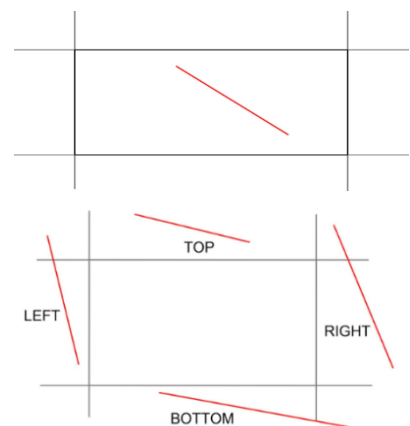


Hình 1.5. Mã vùng được xác định theo 9 vùng

### Các trường hợp xén:

**Trường hợp 1:** Đoạn thẳng có 2 điểm hoàn toàn nằm trong cửa sổ nên không cần xén.

**Trường hợp 2:** Đoạn thẳng có 2 điểm cùng nằm ngoài về một phía ngoài của cửa sổ và sẽ bị xén mất.



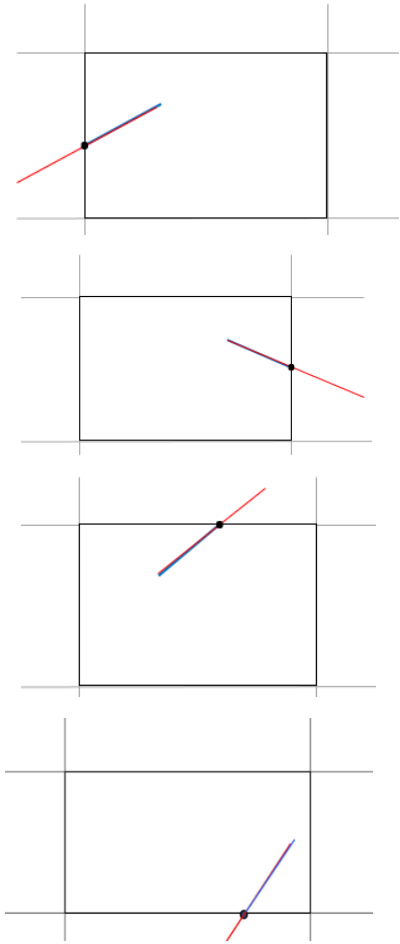
**Trường hợp 3: Nếu đoạn thẳng cắt biên cửa sổ bên trái thì tọa độ tại điểm đó được xác định như sau:**

$$\begin{cases} x_{P'} = x_{wmin} \\ y_{P'} = y_{P1} + (y_{P2} - y_{P1}) * \left( \frac{x_{wmin} - x_{P1}}{x_{P2} - x_{P1}} \right) \end{cases}$$

**Trường hợp 4: Nếu đoạn thẳng cắt biên cửa sổ bên phải thì tọa độ tại điểm đó được xác định như sau:**

**Trường hợp 5: Nếu đoạn thẳng cắt biên cửa sổ bên trên thì tọa độ tại điểm đó được xác định như sau:**

**Trường hợp 6: Nếu đoạn thẳng cắt biên cửa sổ bên dưới thì tọa độ tại điểm đó được xác định như sau:**



**Xác định chiều đoạn thẳng:**

**TH1: đoạn thẳng cắt biên bên trái:** Nếu  $x_{P1} \leq x_{wmin}$  và  $x_{P2} \geq x_{wmin}$  thì đoạn thẳng có chiều từ ngoài vào trong. Ký hiệu OUT2IN.

**TH2: đoạn thẳng cắt biên bên phải:** Nếu  $x_{P1} > x_{wmax}$  và  $x_{P2} \leq x_{wmax}$  thì đoạn thẳng có chiều từ ngoài vào trong. Ký hiệu OUT2IN.

**TH3: đoạn thẳng cắt biên bên trên:** Nếu  $y_{P1} > y_{wmax}$  và  $y_{P2} \leq y_{wmax}$  thì đoạn thẳng có chiều từ ngoài vào trong. Ký hiệu OUT2IN.

**TH4: đoạn thẳng cắt biên bên dưới:** Nếu  $y_{P1} < y_{wmin}$  và  $y_{P2} \geq y_{wmin}$  thì đoạn thẳng có chiều từ ngoài vào trong. Ký hiệu OUT2IN.

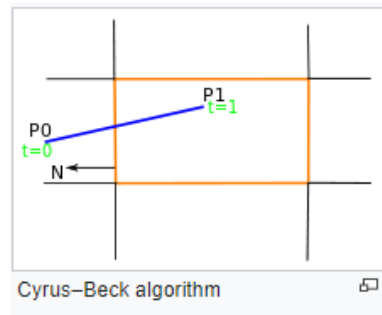
### 1.2.2 Thuật toán Cyrus-beck

Thuật toán Cyrus-Beck là thuật toán cắt dòng tổng quát. Nó được thiết kế để hiệu quả hơn thuật toán Cohen-Sutherland, sử dụng cắt lặp đi lặp lại. Cyrus-Beck là một thuật toán tổng quát và có thể được sử dụng với cửa sổ cắt đa giác lồi, không giống như Cohen-Sutherland, chỉ có thể được sử dụng trên một khu vực cắt hình chữ nhật.

Ở đây phương trình tham số của một đường trong mặt phẳng:

$$\mathbf{p}(t) = t\mathbf{p}_1 + (1 - t)\mathbf{p}_0$$

where  $0 \leq t \leq 1$ .



Hình 1.6. Cửa sổ cắt tia đoạn thẳng

Ngoài ra còn có nhiều thuật toán để xén tia đa giác khác nhau như: Nicholl-Lee-Nicholl, Fast Clipping,  $O(\lg N)$ , Skala, See also, references,... Các thuật toán xén tia này cũng được cải thiện đáng kể về độ chính xác và thời gian xử lý. Xén tia đa giác là một trong những phần mở rộng của thuật toán xén tia đoạn thẳng. Có thể xem tại đường link: (<https://en.wikipedia.org/>).

### 1.3 Thuật toán xén tia đa giác

**Ivan Sutherland và Gary W. Hodgman (1974)** đề xuất thuật toán xén tia đa giác bằng cách lần lượt mở rộng từng đoạn thẳng của *đa giác lõi* và chỉ chọn các đỉnh từ đa giác cần xén nằm ở phía có thể nhìn thấy. Hạn chế của thuật toán này là nếu đa giác cần xén là đa giác lõm, đa giác mới có thể có các cạnh trùng nhau.

Qui ước xác định chiều các điểm của đa giác theo chiều kim đồng hồ như sau:

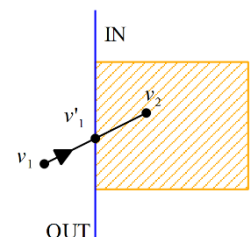
**Theo qui ước:** một đa giác với các đỉnh  $P_1, \dots, P_N$  (các cạnh là  $P_{i-1}P_i$  và  $P_NP_1$ ) được gọi là theo hướng dương nếu các hình theo thứ tự đã cho tạo thành mạch ngược chiều kim đồng hồ.

Nếu bàn tay dọc theo bất kỳ cạnh  $P_{i-1}P_i$  hoặc  $P_NP_1$  cũng chỉ về bên trong đa giác.

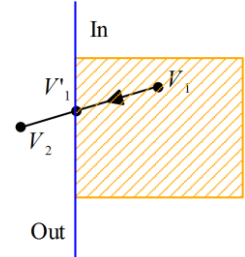
**Quy tắc khác:** Tính tổng các cạnh của đa giác:  $(x_2 - x_1)(y_2 + y_1)$ . Nếu kết quả cho số dương thì chiều của đa giác thuận chiều kim đồng hồ và ngược lại.

#### Các trường hợp trong giải thuật hogman:

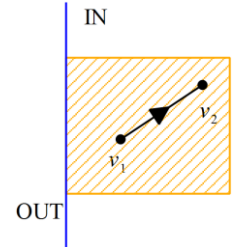
**TH1:** Nếu đỉnh  $v_1$  nằm bên ngoài và đỉnh  $v_2$  nằm bên trong của cạnh và có chiều từ trái sang phải thì giao điểm  $v'_1$  và  $v_2$  được lưu lại.



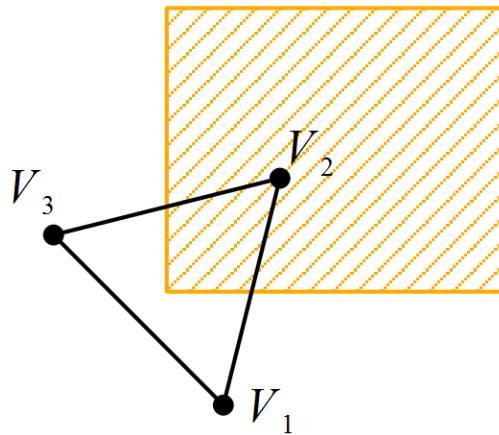
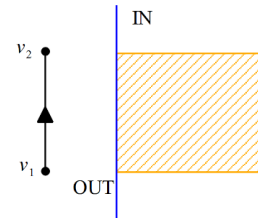
**TH2:** Nếu đỉnh  $v_1$  nằm bên trong và đỉnh  $v_2$  nằm bên ngoài của cạnh và có chiều từ phải sang trái thì giao điểm  $v'_1$  và  $v_1$  được lưu lại.



**TH3:** Nếu cả 2 đỉnh  $v_1$  và  $v_2$  cùng nằm về phía bên trong của cạnh thì cả 2 đỉnh  $v_1$  và  $v_2$  đều được lưu lại.

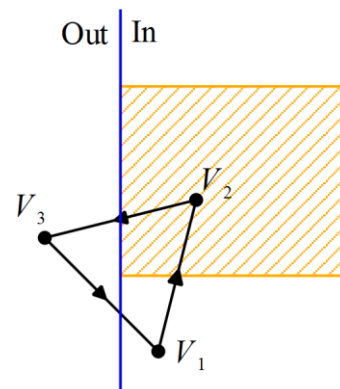


**TH4:** Nếu cả 2 đỉnh  $v_1$  và  $v_2$  cùng nằm về phía bên ngoài của cạnh thì cả 2 đỉnh  $v_1$  và  $v_2$  không được lưu lại.



**Ví dụ minh họa:** Cho đa giác dương sau và cửa sổ cắt, hãy xác định các đỉnh cắt:

**Bước 0:** Áp dụng quy tắc bàn tay trái cho đa giác dương để xác định chiều của các cạnh.

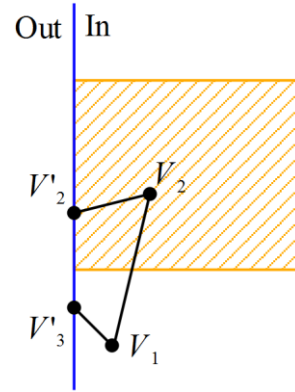


**Bước 1: Cắt bên trái**

Xét cạnh  $V_1-V_2$  (In-In):  $V_2$ .

Xét cạnh  $V_2-V_3$  (In-Out):  $V'_2$ .

Xét cạnh  $V_3-V_1$  (Out-In):  $V_3-V'_3$ .

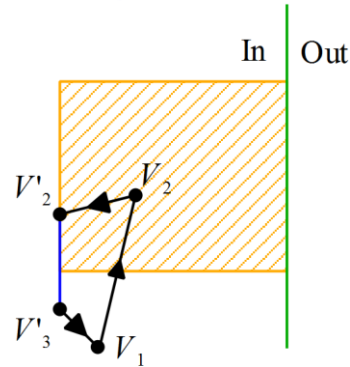
**Bước 2. Cắt bên phải**

Xét cạnh  $V_1-V_2$  (In-In):  $V_2$ .

Xét cạnh  $V_2-V'_2$  (In-In):  $V'_2$ .

Xét cạnh  $V'_2-V'_3$  (In-In):  $V'_3$ .

Xét cạnh  $V'_3-V_1$  (In-In):  $V_1$ .

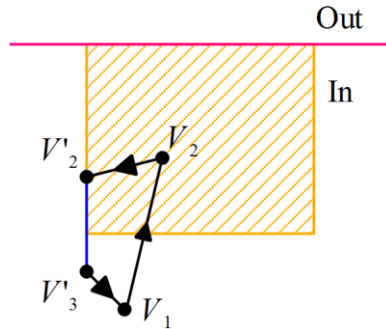
**Bước 3. Cắt bên trên**

Xét cạnh  $V_1-V_2$  (In-In):  $V_2$ .

Xét cạnh  $V_2-V'_2$  (In-In):  $V'_2$ .

Xét cạnh  $V'_2-V'_3$  (In-In):  $V'_3$ .

Xét cạnh  $V'_3-V_1$  (Out-Out): NULL.

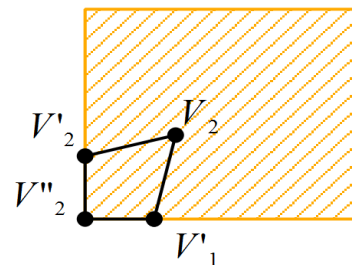
**Bước 4. Cắt bên dưới**

Xét cạnh  $V_1-V_2$  (Out-In):  $V'_1V_2$ .

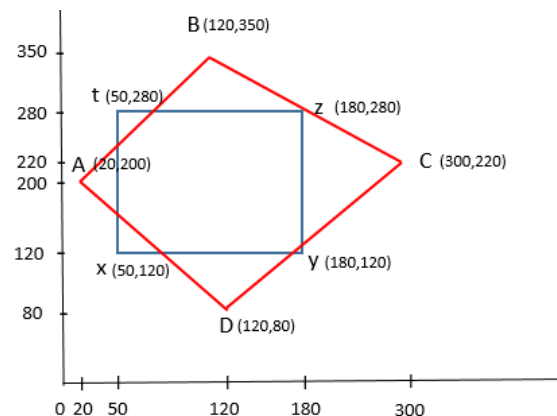
Xét cạnh  $V_2-V'_2$  (In-In):  $V'_2$ .

Xét cạnh  $V'_2-V'_3$  (In-Out):  $V''_2$ .

Xét cạnh  $V'_3-V_1$  (Out-Out): NULL.

**Ví dụ 2:**

Cho đa giác ABCD với tọa độ lần lượt là (20,200), (120,350), (300,220), (120,80) và cửa sổ cắt xyzt với tọa độ lần lượt là (50,120), (180,120), (180,280), (50,280).



**Bước 1: Ta xét cạnh bên trái****Cạnh AB:** nằm ngoài cửa sổ theo chiều Out-In

Tìm giao điểm A' thuộc đường thẳng AB

$$\left\{ \begin{array}{l} x_{A'} = 50 \\ y_{A'} = y_A + (y_B - y_A) * \left( \frac{x_{A'} - x_A}{x_B - x_A} \right) = 200 + (350 - 200) * \frac{50 - 20}{120 - 20} = 245 \end{array} \right\}$$

Vậy A'(50,245)

**Cạnh DA:** nằm ngoài cửa sổ theo chiều In-Out

Tìm giao điểm A'' thuộc đường thẳng DA

$$\left\{ \begin{array}{l} x_{A''} = 50 \\ y_{A''} = y_D + (y_A - y_D) * \left( \frac{x_{A''} - x_D}{x_A - x_D} \right) = 80 + (200 - 80) * \frac{50 - 120}{20 - 120} = 164 \end{array} \right\}$$

Vậy A''(50,164)

**Bước 2: Ta xét cạnh bên phải****Cạnh BC:** nằm ngoài hoàn toàn cửa sổ nên ta không tạo ra điểm mới cho đa giác sau khi xén tỉa**Cạnh CD:** nằm ngoài cửa sổ và chiều Out-In

Tìm giao điểm C' thuộc đường thẳng CD

$$\left\{ \begin{array}{l} x_{C'} = 180 \\ y_{C'} = y_C + (y_D - y_C) * \left( \frac{x_{C'} - x_C}{x_D - x_C} \right) = 220 + (80 - 220) * \frac{180 - 300}{120 - 300} = \frac{380}{3} \end{array} \right\}$$

Vậy C'(180,  $\frac{380}{3}$ )**Bước 3: Ta xét cạnh bên trên****Cạnh AB:** nằm ngoài cửa sổ chiều In-Out

Tìm giao điểm B' thuộc đường thẳng AB

$$\left\{ \begin{array}{l} y_{B'} = 280 \\ x_{B'} = x_A + (y_{B'} - y_A) * \frac{x_B - x_A}{y_B - y_A} = 20 + (280 - 200) * \frac{120 - 20}{350 - 200Q} = \frac{220}{3} \end{array} \right\}$$

Vậy B'( $\frac{220}{3}$ , 280)**Cạnh BC:** nằm ngoài hoàn toàn cửa sổ cắt nên sẽ không tạo điểm mới**Bước 4: Xét cạnh bên dưới****Cạnh DA:** nằm ngoài cửa sổ và chiều Out-In

Tìm giao điểm D' thuộc đường thẳng DA

$$\left\{ \begin{array}{l} y_{D'} = 120 \\ x_{D'} = x_D + (y_{D'} - y_D) * \frac{x_A - x_D}{y_A - y_D} = 120 + (120 - 80) * \frac{20 - 120}{200 - 80} = \frac{260}{3} \end{array} \right\}$$



Vậy  $D'(\frac{260}{3}, 120)$

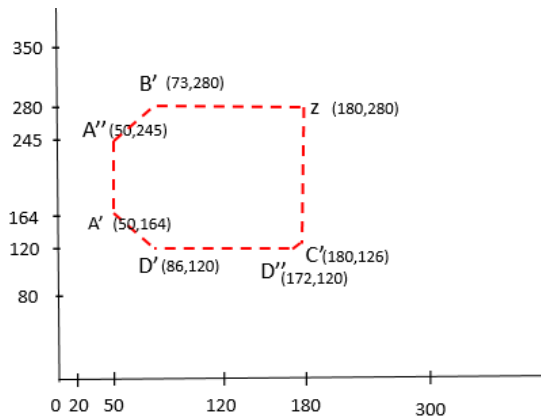
**Cạnh CD:** nằm ngoài cửa sổ và chiều In-Out

Tìm giao điểm  $D''$  thuộc đường thẳng CD

$$\begin{cases} y_{D''} = 120 \\ x_{D''} = x_C + (y_{D''} - y_C) * \frac{x_D - x_C}{y_D - y_C} = 300 + (120 - 220) * \frac{120 - 300}{80 - 220} = \frac{1200}{7} \end{cases}$$

Vậy  $D''(\frac{1200}{7}, 120)$

Sau khi thực hiện xong ta được đa giác mới  $A''B'zC'D''D'A'$  có tọa độ lần lượt là  $(50,245)$ ,  $(73,280)$ ,  $(180,280)$ ,  $(180,126)$ ,  $(172,120)$ ,  $(86,120)$ ,  $(50,164)$



**Giải thuật Vatti** (1992) được đề xuất bởi Bala R. Vatti, thuật toán giải quyết hạn chế của các loại đa giác có thể được sử dụng làm đối tượng hoặc clip. Ngay cả các đa giác phức tạp (tự giao nhau) và đa giác lõm có thể được xử lý ([https://en.wikipedia.org/wiki/Vatti\\_clipping\\_algorithm](https://en.wikipedia.org/wiki/Vatti_clipping_algorithm)).

## PHẦN 2. PHƯƠNG PHÁP NGHIÊN CỨU

Để thực hiện các thao tác xén tia đa giác thì trước hết cần nhập dữ liệu các tọa độ điểm của cạnh cần vẽ và cửa sổ cắt.

```
void nhapdulieu()
{
    int i;
    printf("Nhap so canh cua da giac = "); //nhap so canh cua da giac can xen
    scanf("%d",&pointsize);

    for (i=0;i<pointsize;i++)
    {
        printf("Toa do x cho canh %d = ",i);
        scanf("%d",&xc[i]);
        printf("Toa do y cho canh %d = ",i);
        scanf("%d",&yc[i]);
    }

    printf("Nhap so canh cua cua so clipping = "); //nhap so canh cua cua so cat
    scanf("%d",&>windowsize);

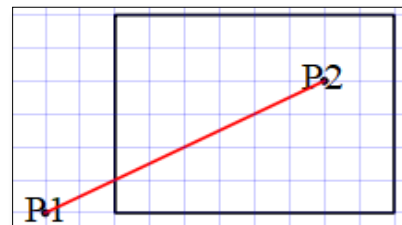
    for (i=0;i<windowsize;i++)
    {
        printf("Toa do x cho canh window %d = ",i);
        scanf("%d",&wx[i]);
        printf("Toa do y cho canh window %d = ",i);
        scanf("%d",&wy[i]);
    }
}
```

Hình 2.1. Nhập tọa độ đa giác và cửa sổ cắt

Sau khi nhập dữ liệu tọa độ các điểm có thể rơi vào những trường hợp như sau:

### 2.1. Đoạn thẳng nằm trên biên trái cửa sổ cắt

**Trường hợp 1:** Đoạn P1P2 nằm trên biên trái cửa sổ xén có chiều từ ngoài vào trong.

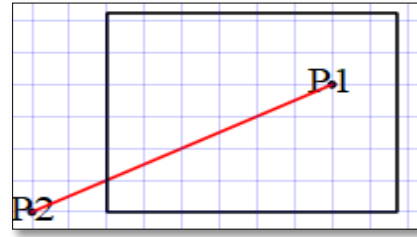


Hình 2.2. Đoạn thẳng nằm trên biên trái cửa sổ cắt (TH1)

Minh họa thuật toán:

```
//toa do x cua diem moi cat duoc
xnew = xmin;
//toa do y cua diem moi cat duoc
ynew = ya + (float)(yb-ya)*(float)(xmin-xa)/(float)(xb-xa);
//gan gia tri toa do
xc_tam[j]=xnew;
yc_tam[j]=ynew;
j++;
xc_tam[j]=xb;
yc_tam[j]=yb;
j++;
break;
```

**Trường hợp 2:** Đoạn P1P2 nằm trên biên trái cửa sổ nên có chiều từ trong ra ngoài.



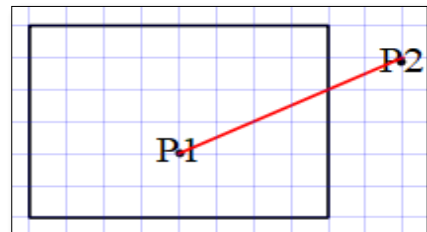
Hình 2.3. Đoạn thẳng nằm trên biên trái cửa sổ cắt (TH2)

Minh họa thuật toán :

```
//toạ độ x của điểm mới cắt được
xnew = xmin;
//toạ độ y của điểm mới cắt được
ynew = ya + (float)(yb-ya)*(float)(xmin-xa)/(float)(xb-xa);
//gán giá trị toạ độ
xc_tam[j]=xnew;
yc_tam[j]=ynew;
j++;
break;
```

## 2.2. Đoạn thẳng nằm trên biên phải cửa sổ nên

**Trường hợp 1:** Đoạn P1P2 nằm trên biên phải cửa sổ nên có chiều từ trong ra ngoài

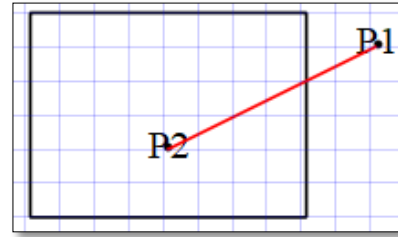


Hình 2.4. Đoạn thẳng nằm trên biên phải cửa sổ cắt (TH1)

Minh họa thuật toán :

```
//toạ độ x của điểm mới cắt
xnew = xmax;
//toạ độ y của điểm mới cắt
ynew = ya + (float)(yb-ya)*(float)(xmax-xa)/(float)(xb-xa);
//gán giá trị
xc_tam[j]=xnew;
yc_tam[j]=ynew;
j++;
break;
```

**Trường hợp 2:** Đoạn thẳng P1P2 nằm trên biên phải cửa sổ nên có chiều từ trong ra ngoài.



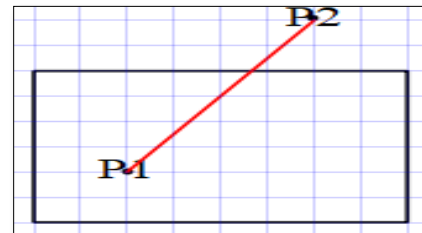
Hình 2.5. Đoạn thẳng nằm trên biên phải cửa sổ cắt (TH2)

Minh họa thuật toán :

```
//toa do x của điểm mới cắt
xnew = xmax;
//toa do y của điểm mới cắt
ynew = ya + (float)(yb-ya)*(float)(xmax-xa)/(float)(xb-xa);
//gan giá trị
xc_tam[j]=xnew;
yc_tam[j]=ynew;
j++;
break;
```

### 2.3 Đoạn thẳng nằm trên biên trên cửa sổ cắt

**Trường hợp 1:** Đoạn thẳng P1P2 nằm trên biên trên cửa sổ nên có chiều từ trong ra ngoài

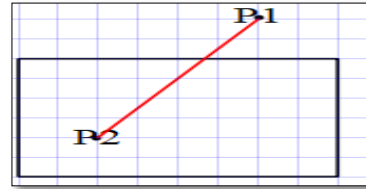


Hình 2.6. Đoạn thẳng nằm trên biên trên cửa sổ cắt (TH1)

Minh họa thuật toán :

```
xnew = xmax;
ynew = ya + (float)(yb-ya)*(float)(xmax-xa)/(float)(xb-xa);
xc_tam[j]=xnew;
yc_tam[j]=ynew;
j++;
break;
```

**Trường hợp 2:** đoạn thẳng P1P2 nằm trên biên trên của sổ xén có chiều từ ngoài vào trong



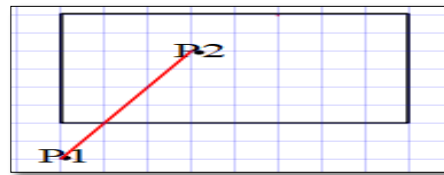
Hình 2.7. Đoạn thẳng nằm trên biên trên của sổ cắt (TH2)

Minh họa thuật toán :

```
//toa do x cua diem moi cat
xnew = xwmax;
//toa do y cua diem moi cat
ynew = ya + (float)(yb-ya)*(float)(xwmax-xa)/(float)(xb-xa);
//gan toa do
xc_tam[j]=xnew;
yc_tam[j]=ynew;
j++;
xc_tam[j]=xb;
yc_tam[j]=yb;
j++;
break;
```

#### 2.4 Đoạn thẳng nằm trên biên dưới của sổ cắt

**Trường hợp 1:** Đoạn thẳng P1P2 nằm trên biên dưới của sổ xén có chiều từ ngoài vào trong

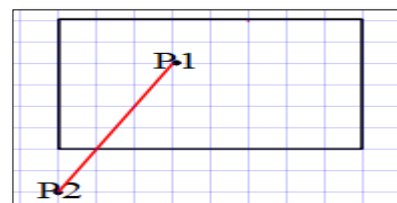


Hình 2.8. Đoạn thẳng nằm trên biên dưới của sổ cắt (TH1)

Minh họa thuật toán :

```
xnew = xa + (float)(ywmax-ya)*(xb-xa)/(yb-ya);
ynew = ywmax;
xc_tam[j]=xnew;
yc_tam[j]=ynew;
j++;
xc_tam[j]=xb;
yc_tam[j]=yb;
j++;
break;
```

**Trường hợp 2:** Đoạn thẳng P1P2 nằm trên biên dưới của sổ xén có chiều từ trong ra ngoài



Hình 2.9. Đoạn thẳng nằm trên biên dưới của sổ cắt (TH2)

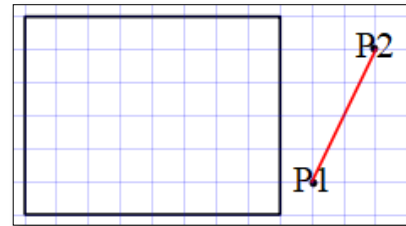
Minh họa thuật toán :

```
xnew = xa + (float)(ymax-ya)*(xb-xa)/(yb-ya);
ynew = ymax;
xc_tam[j]=xnew;
yc_tam[j]=ynew;
j++;
break;
```

## 2.5 Đoạn thẳng nằm ngoài cửa sổ xén

Minh họa thuật toán:

```
break; //(không có điểm mới được tạo ra)
```

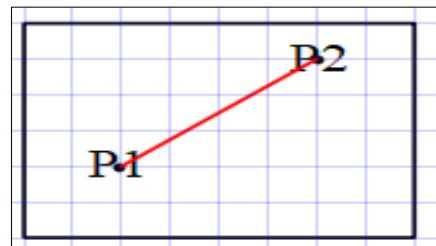


Hình 2.10. Đoạn thẳng nằm ngoài cửa sổ cắt

## 2.6 Đoạn thẳng nằm trong cửa sổ xén

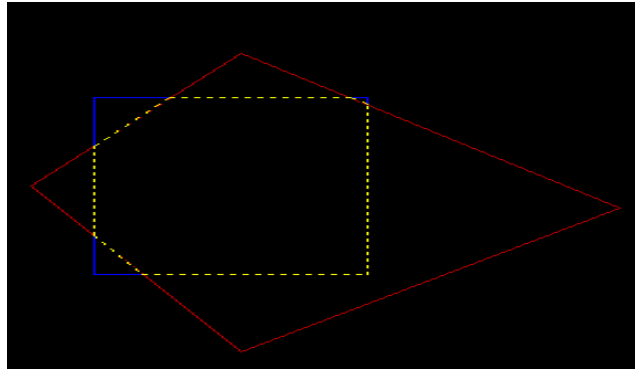
Minh họa thuật toán:

```
xnew = xb;
ynew = yb;
xc_tam[j]=xnew;
yc_tam[j]=ynew;
j++;
break;
```



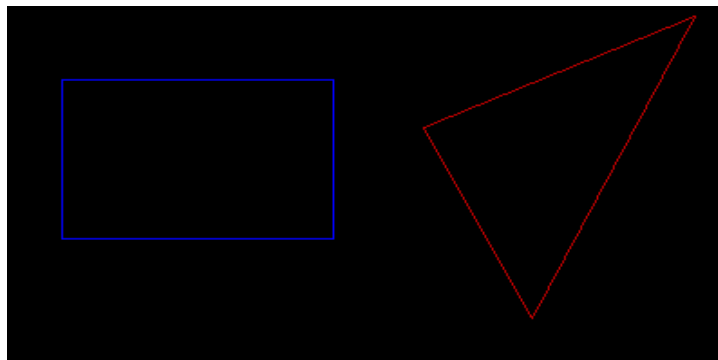
Hình 2.11. Đoạn thẳng nằm trong cửa sổ cắt

### PHẦN 3. KẾT QUẢ ĐẠT ĐƯỢC



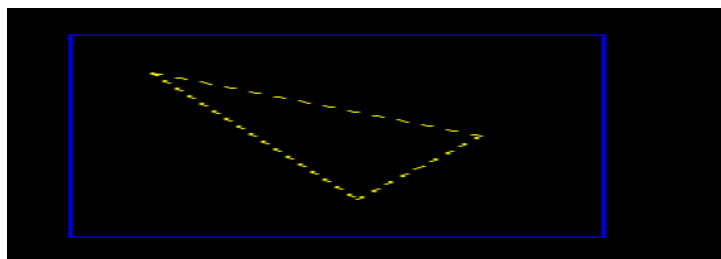
*Hình 3.1. Cửa sổ cắt cắt tất cả các cạnh của đa giác*

Hình 3.1 mô tả trường hợp đa giác giao biên cửa sổ cắt với hình màu xanh là cửa sổ cắt, hình màu đỏ là đa giác và hình màu vàng là kết quả sau khi cắt.



*Hình 3.2. Minh họa đa giác nằm hoàn toàn bên ngoài cửa sổ cắt*

Hình 3.2 mô tả trường hợp đa giác nằm ngoài cửa sổ cắt với hình màu xanh là cửa sổ cắt và hình màu đỏ là đa giác.



*Hình 3.3. Minh họa trường hợp đa giác nằm hoàn toàn bên trong cửa sổ cắt*

Hình 3.3 mô tả trường hợp đa giác nằm trong cửa sổ cắt với hình màu xanh là cửa sổ cắt và hình màu vàng vừa là đa giác vừa là kết quả sau khi cắt.

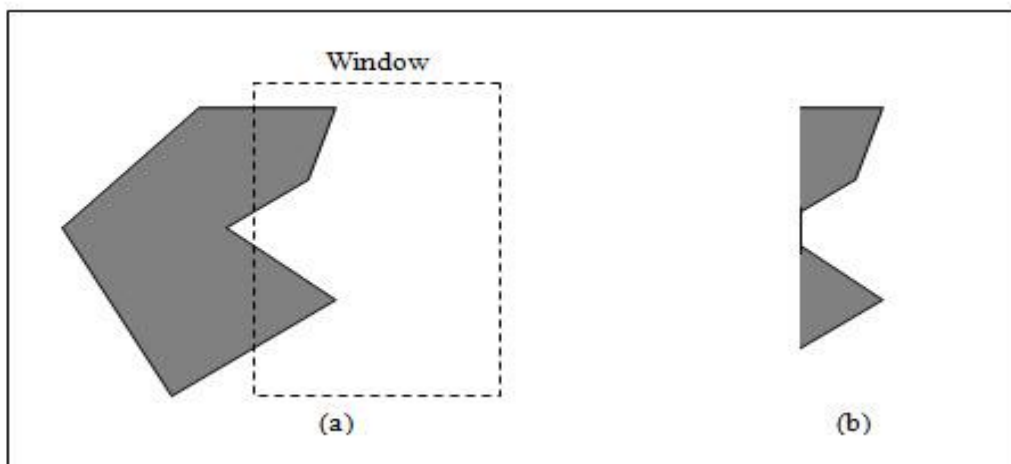
## THẢO LUẬN

Thuật toán xén tia đa giác **Sutherland-Hodgman** ra đời từ rất sớm và kế thừa những ưu điểm và cải tiến những hạn chế từ những thuật toán, giải thuật xén tia trước đó nên nó rất phổ biến và hữu ích để cắt đa giác

Bên cạnh những thuận lợi, ưu điểm mà thuật toán Sutherland-Hodgman đem lại trong việc xén tia thì nó cũng có những hạn chế, nhược điểm chưa được khắc phục như đòi hỏi một lượng bộ nhớ đáng kể. Kết quả sau khi thuật toán này thực hiện được lưu trữ trong bộ nhớ. Vì vậy, lãng phí bộ nhớ để lưu trữ đa giác trung gian

Và điểm hạn chế thứ hai chính là vấn đề của bài toán trên, đối với đa giác lõm thì thuật toán Sutherland-Hodgman, khi cắt có thể có các cạnh trùng nhau, chồng chéo nhau.

Trong ví dụ dưới đây ta thấy sau khi thực hiện việc cắt tia đa giác ở hình a) thì ta được hai đa giác tách biệt nối với nhau bằng một đoạn thẳng ở hình b) nhưng đoạn thẳng được nối ở giữa là dư thừa



Hình 4.1. Hạn chế của thuật toán Sutherland-Hodgman

Nguồn: <https://voer.edu.vn/>

Và thuật toán Weiler–Atherton ra đời và khắc phục được điểm hạn chế này bằng cách trả về một tập hợp các đa giác được chia, nhưng phức tạp hơn và đắt hơn về mặt tính toán và thuật toán này chưa được thực hiện trên máy tính. Trong tương lai thuật toán sẽ được cải thiện và thực hiện tối ưu hơn.



**TÀI LIỆU THAM KHẢO**

1. Nguyễn Quang Khánh, “Đồ họa máy tính”, 2015, NXB Khoa học Kỹ thuật
2. Vũ Hải Quân, “Đồ họa máy tính”, 2007, NXB Đại Học Quốc Gia
3. D. Hearn, M.P. Baker, “Computer Graphics, C version”, 1997, Prentice Hall
4. Đoàn Vũ Thịnh, “Bài giảng Kỹ thuật đồ họa”, 2019, Đại học Nha Trang