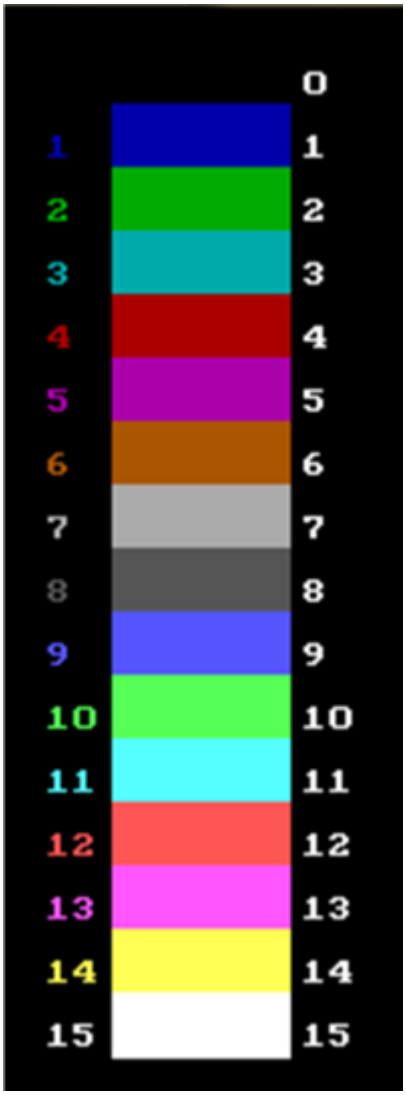
KĨ THUẬT ĐỒ HỌA

1.Đồ họa cơ sở

- Thư viện đồ họa C/C++: graphics.h

Đồ họa C sử dụng các chức năng Graphics.h hoặc WinBGIM (Windows 7) có thể được sử dụng để vẽ các dạng đồ họa khác nhau, hiển thị văn bản bằng các phông chữ khác nhau, thay đổi màu sắc và nhiều chức năng hơn nữa. Sử dụng các chức năng của graphics.h trong trình biên dịch C/C++, tacó thể tạo các chương trình đồ họa, hoạt ảnh, dự án và trò chơi. Có thể vẽ hình tròn, đường thẳng, hình chữ nhật và nhiều hình hình học khác. Có thể thay đổi màu sắc của chúng bằng cách sử dụng các chức năng có sẵn và tô cho đối tượng đồ họa.

Bảng màu:



- Khởi động chế độ đồ họa:
+ Trong TurboC, BorlandC: khai báo tự động tìm mode màn hình

```
int manhinh,mode;
manhinh = mode = 0; // tự động tìm mode màn hình
```

initgraph(&manhinh,&mode,"đường dẫn tới thư mục chứa file EGAVGA.BGI");
Trong TurboC file EGAVGA.BGI ở thư mục: "*\TC\BGI"
+ Trong Devcpp: trước khi lập trình đồ họa trên devcpp bạn phải cài thư viện đồ họa, hướng dẫn chi tiết trên google search
initwindow(int x,int y);// kích thước màn hình đồ họa dài rộng bao nhiêu
VD: initwindow(900,700);
- Kiểm tra xem có lỗi gì khi tạo màn hình đồ họa:
graphresult();
hàm trả lại 0: không có lỗi gì, từ 1 -> 18 tương ứng với các lỗi
- Xóa tất cả những hình đã vẽ trên màn hình đồ họa:
cleardevice();
- Kết thúc chế độ đồ họa:
closegraph();
- Các hàm thường dùng:

getmaxx()	Tọa độ x lớn nhất
getmaxy()	Tọa độ y lớn nhất
getx()	Vị trí x hiện tại của con trỏ
gety()	Vị trí y hiện tại của con trỏ
setbkcolor(<màu nền>)	Đặt màu nền: 0 -> 15
setcolor(<màu vẽ>)	Đặt màu vẽ: 0 -> 15
getbkcolor()	Màu nền hiện tại
getmaxcolor()	Số màu tối đa của màn hình
getcolor()	Màu vẽ hiện tại
putpixel(x,y,c)	Vẽ điểm có tọa độ (x,y) với màu c
getpixel(x,y)	Trả lại màu tại điểm (x,y)
line(x,y,x_1,y_1)	Vẽ đoạn thẳng đi qua (x,y) và (x_1,y_1)
lineto(x,y)	Vẽ đoạn thẳng đi qua vị trí hiện tại của con trỏ tới điểm (x,y)
linrel(x,y)	Giống với line(x,y)

rectangle(x_1,y_2,x_2,y_2)	Vẽ hình chữ nhật rỗng
bar(x_1,y_2,x_2,y_2)	Vẽ hình chữ nhật đặc
bar3d(x_1,y_2,x_2,y_2,h,top)	h là chiều cao, top = 1 có nắp
setlinestyle(kiểu đường, mẫu tô, độ đậm)	+ kiểu đường từ 0 -> 4 0: đường đặc 1: đường chấm 2: đường gạch 3: đường gạch dài 4: đường tự tạo + mẫu tô: chỉ có tác dụng khi kiểu đường là 4, ta dùng 2byte để định nghĩa + độ đạm là 1 hoặc 3 1: nét vẽ bình thường 3: nét vẽ đậm
setfillstyle(mẫu tô, màu tô)	+ mẫu tô: từ 0 -> 12 + màu tô: từ 0 -> 15

setfillpattern(mẫu tô, màu tô)	Định nghĩa mẫu tô
getfillsettings(struct fillsettingstype *info)	Lấy mẫu tô hiện tại
getfillpattern(mẫu tô)	Trả lại mẫu tô hiện tại
drawpoly(mảng số nguyên chứa tọa độ các điểm,số cặp điểm)	Vẽ đa giác rỗng
fillpoly(mảng số nguyên chứa tọa độ các điểm,số cặp điểm)	Vẽ đa giác đặc
arc(x, y, góc đầu, góc cuối, bán kính)	Vẽ cung tròn có tâm (x, y) với các góc và bán kính tương ứng
circle(x, y, bán kính)	Vẽ đường tròn có tâm tại (x, y)
pieslice(x, y, góc đầu, góc cuối, bán kính)	Vẽ hình quạt tròn đặc với mẫu hiện tại
ellipse(x, y, góc đầu, góc cuối, a, b)	Vẽ cung elip với tâm, các góc và các bán kính theo hoàng độ và tung độ tương ứng
fillellipse(x, y, a, b)	Vẽ hình elip đặc
sector(x, y, góc đầu, góc cuối, a, b)	Vẽ hình quạt elip
floodfill(x, y, c)	Tô màu một hình kín chứa điểm x, y và màu c, màu c phải trùng với setfillstyle(mẫu tô, c);

- Viết văn bản trong màn hình đồ họa
outtextxy(x, y, s): viết văn bản tại vị trí (x,y)
moveto(x, y): chuyển con trỏ chuột đến vị trí (x,y)
- Điều chỉnh Font chữ, hướng cỡ chữ
settextstyle(Font, hướng, cỡ chữ)
+ Font
DEFAULT_FONT 0
SMALL_FONT 1
TRIPLEX_FONT 2
SANS_SERIF_FONT 3
GOTHIC_FONT 4
+ Hướng
HOIRIZ_DIR 0 //nằm hàng ngang
VERT_DIR 1 //nằm theo đường thẳng đứng
+ Cỡ chữ bắt đầu từ 1
- Điều chỉnh cách viết
settextjustify(theo hướng ngang, theo hướng dọc)
+ Theo hướng nằm ngang:
+ Theo hướng nằm ngang:

LEFT_TEXT = 0 : Viết từ trái sang phải.

CENTER_TEXT = 1 : Viết từ vị trí con trỏ sang hai bên.

RIGHT_TEXE = 2 : Viết từ phải sang trái.

+ Theo hướng thẳng đứng:

BOTTOM_TEXT = 0 : Viết từ dưới lên.

CENTER_TEXT = 1 : Viết từ vị trí con trỏ lên trên và xuống dưới.

TOP_TEXT = 2 : Viết từ trên xuống

2. Kỹ thuật tô màu, tô bóng hình học

- Các hệ màu cơ bản của đồ họa:

+ Hệ màu RGB:

Hệ màu RGB là từ viết tắt trong tiếng Anh và có nghĩa :

R: Red (màu đỏ)

G: Green (màu xanh lá cây)

B: Blue (màu xanh lam)

Đây là ba màu gốc trong các mô hình ánh sáng bổ sung. Từ ba màu cơ bản này từ cách thay đổi tỉ lệ giữa các màu RGB để tạo ra vô số các màu sắc khác nhau và cách tổng hợp từ 3 màu RGB này gọi là màu cộng (các màu sinh ra từ 03 màu này sẽ sáng hơn màu gốc – additive color). Hệ màu RGB mô tả màu sắc trong một mô hình gọi là "không gian màu". Không gian này được minh họa bằng một khối lập phương với các trục chính R, G, B.

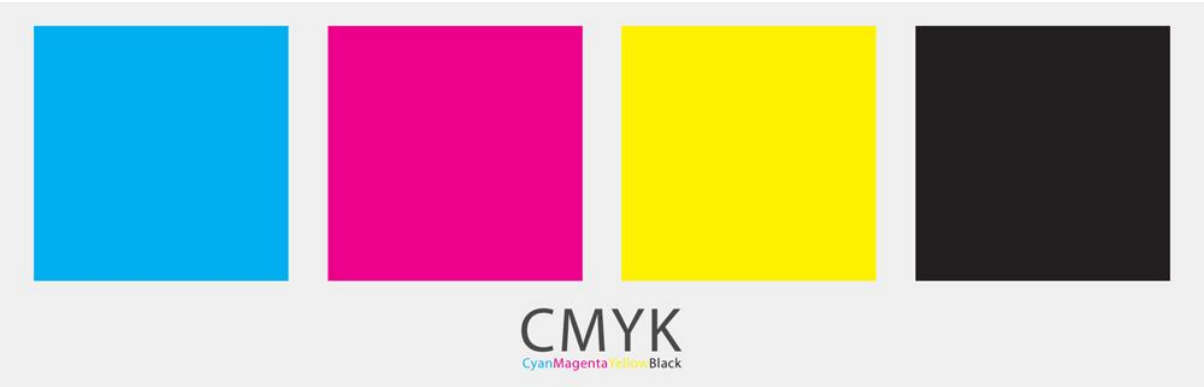
Hệ màu RGB là chế độ hiển thị màu sắc tự nhiên của màn hình CRT, màn hình LCD và màn hình plasma. Máy ảnh và máy quét cũng có thể sử dụng chế độ RGB. Hệ màu RGB là hệ màu là tốt nhất cho thiết kế : thiết kế website, hình ảnh kỹ thuật số, thiết kế các tài liệu quảng cáo trực tuyến,…

Bất lợi:

- Các giá trị R,G,B của một màu là khác nhau đối với các màn hình khác nhau: Nghĩa là các giá trị R,G,B của một màu trên màn hình màu này sẽ không sinh ra đúng màu đó trên một màn hình khác.

- Sự mô tả các màu trong thế giới thực đối với không gian RGB còn nhiều hạn chế bởi vì không gian RGB không hoàn toàn phù hợp với sự cảm nhận màu sắc của con người. Hai điểm phân biệt trong không gian RGB, với mắt người có thể hoặc không thể là thể hiện của hai màu khác nhau.

+ Hệ màu CYMK (hệ màu trừ)



Hệ màu CYMK là từ viết tắt trong tiếng Anh và có nghĩa :

C – Cyan là màu lục lam

M – Magenta là màu đỏ tươi (màu hồng đỏ)

Y – Yellow là màu vàng

K – Keyline/Black là màu đen

CMYK là một mô hình màu trong đó tất cả các màu được mô tả như là một hỗn hợp của các quá trình hòa trộn của bốn màu sắc. CMYK là mô hình màu tiêu chuẩn được sử dụng trong in offset cho các tài liệu đầy đủ màu sắc. Vì in ấn sử dụng các loại mực của bốn màu cơ bản, nó thường được gọi là in bốn màu.

Nguyên lý làm việc của CMYK là trên cơ sở hấp thụ ánh sáng. Màu mà chúng ta nhìn thấy là từ phần của ánh sáng không bị hấp thụ. Trong CMYK hồng sẫm cộng với vàng sẽ cho màu đỏ, cánh sen cộng với xanh lơ cho màu xanh lam, xanh lơ cộng với vàng sinh ra màu xanh lá cây và tổ hợp của các màu xanh lơ, cánh sen và vàng tạo ra màu đen.

+ Hệ màu PANTONE

Màu Pantone được xác định là một loạt các con số thay vì tên (ngoại trừ với màu sắc sử dụng trong thời trang), do đó ta sẽ nghe các tham chiếu PANTONE 2985 C thay vì màu xanh da trời.

Màu Pantone là màu pha, hay màu thứ 5. Nói một cách dễ hiểu hơn, màu Pantone là màu được nhà sản xuất pha sẵn, khác với việc nhà in pha trộn các màu CMYK là 4 màu cơ bản trong in ấn để tạo ra những màu chúng ta mong muốn. Màu Pantone có sắc độ tươi tắn rất nổi

bật, làm tăng tính thẩm mỹ cho sản phẩm in ấn. Khi đặt cạnh những ấn phẩm in ấn được in offset với 4 màu cơ bản, sắc độ của màu Pantone bao giờ cũng nổi bật hơn hẳn.

- Các thuật toán tô màu(quét dòng, loang):

+ Thuật toán tô màu Scanline (quét dòng)

Thuật toán tô màu Scanline (hay còn gọi là thuật toán tô màu theo dòng quét) được áp dụng để tô màu các đa giác lồi, lõm hay đa giác tự cắt.

Với mỗi dòng quét, ta sẽ xác định phần giao của đa giác và dòng quét, rồi tô màu các pixel thuộc đoạn giao đó. Để xác định các đoạn giao, ta tiến hành việc tìm giao điểm của dòng quét với các cạnh của đa giác, sau đó các giao điểm này sẽ được sắp theo thứ tự tăng dần của hoành độ giao điểm. Các đoạn giao chính là các đoạn thẳng được giới hạn bởi từng cặp giao điểm một.

Ý tưởng:

- Tìm ymin, ymax lần lượt là giá trị nhỏ nhất, lớn nhất của tập các tung độ của các đỉnh của đa giác đã cho.
- Ứng với mỗi dòng quét y=k, k thay đổi từ ymin đến ymax, lặp :Tìm tất cả các hoành độ giao điểm của dòng quét y=k với các cạnh của đa giác. Sắp xếp các hoành độ giao điểm theo thứ tự tăng dần : x0, x1, … Tô màu các đoạn thẳng y=k trên đường thẳng lần lượt được giới hạn bởi các cặp (x0, x1), (x2, x3), ….(x2k, x2k+1).
- Để hạn chế số cạnh cần tìm giao điểm ứng với dòng quét, ta áp dụng công thức hệ số góc sau:

$$x_{k+1} = x_k + 1/m$$

trong đó: m là hệ số góc của cạnh; x_{k+1} , x_k lần lượt là hoành độ giao điểm của một cạnh nào đó với dòng quét y=k và y=k+1.

- Giải quyết trường hợp số giao điểm đi qua đỉnh đơn điệu thì ta tính số giao điểm là 1; đi qua đỉnh cực trị thì tính số giao điểm là 0 (hoặc 2).

+ Thuật toán tô màu Scanline (loang)

Thuật toán tô màu loang (hay còn gọi là tô màu theo đường biên, tô lân cận). Khác với thuật toán tô màu dựa theo dòng quét, đường biên của vùng tô màu ở thuật toán tô loang được xác định bởi tập các đỉnh của 1 đa giác, đường biên trong thuật toán được mô tả bằng một giá trị duy nhất, đó là màu của tất cả các điểm thuộc về đường biên (nói ngắn gọn là chúng ta sẽ tô đường biên một màu riêng). Bắt đầu từ 1 điểm nằm bên trong vùng tô, ta sẽ kiểm tra các điểm lân cận của nó đã được tô màu hay có phải điểm biên hay không. Nếu không phải là điểm đã tô và không phải là điểm biên ta sẽ tô màu nó. Lặp lại cho tới khi nào không còn tô được điểm nào nữa thì dừng.

Thuật toán đệ quy:

- Bước 1: Kẻ vùng biên cần tô.
- Bước 2: Xác định một điểm (x,y) bên trong vùng cần tô.
- Bước 3: Tô điểm (x,y) sau đó tô loang những điểm lân cận.

Ưu điểm:

- Có thể tô vùng có hình dạng bất kỳ.
- Cài đặt thuật toán dễ dàng.

Khuyết điểm:

- Không thể tô các vùng có kích thước lớn do tràn bộ nhớ khi sử dụng đệ quy.
- Việc gọi đệ quy, thuật toán cho 4 điểm lân cận của điểm hiện hành không quan tâm tới một trong bốn điểm đó đã được xét ở bước trước hay chưa (tức là sẽ có những điểm bị xét đi xét lại nhiều lần). Điều này khiến tốc độ chậm, không hiệu quả.

Thuật toán khử đệ quy:

Áp dụng hàng đợi Queue, BFS

Bước 1: Cất điểm hạt giống đầu tiên vào kho.

Bước 2: Lặp nếu kho không rỗng

- Lấy điểm hạt giống.
- Tô điểm hạt giống, sau đó tô loang sang 2 bên.
- Bổ sung những điểm hạt giống mới vào kho từ dòng trên và dòng dưới.

Tiêu chuẩn để làm điểm hạt giống: điểm này chưa được tô màu và không phải điểm biên.

- Thuật toán tô bóng:

Tô bóng Phong (tiếng Anh: Phong shading) là thuật ngữ dùng để chỉ một tập hợp các kỹ thuật trong Đồ họa 3D bao gồm: một mô hình phản xạ ánh sáng từ các bề mặt và một phương pháp dùng để ước lượng màu sắc của điểm ảnh bằng cách nội suy véc-tơ trực giao bề mặt. Các mô hình và phương pháp trên được phát triển bởi Bùi Tường Phong tại Đại học Utah và được công bố trong luận án Tiến sĩ của ông vào năm 1973

Tô bóng Gouraud , được đặt theo tên của Henri Gouraud , là một phương pháp nội suy được sử dụng trong đồ họa máy tính để tạo ra bóng đổ liên tục của các bề mặt được biểu diễn bằng các mắt lưới đa giác . Trong thực tế, tính năng đổ bóng Gouraud thường được sử dụng nhất để đạt được ánh sáng liên tục trên các mắt lưới Tam giác bằng cách tính toán ánh sáng ở các góc của mỗi hình tam giác và nội suy tuyến tính các màu thu được cho mỗi pixel được bao phủ bởi hình tam giác. Gouraud lần đầu tiên công bố kỹ thuật này vào năm 1971.

3. Phép biến hình (Trong báo cáo)

4. OpenGL (Trong báo cáo)

5. Đường con và mặt cong (Trong báo cáo)

6. Ứng dụng đồ họa sử dụng công cụ Java (Trong báo cáo và Projects)

7. Đồ họa kỹ xảo 3D

B1. Tiền sản xuất (Pre-production): Chuẩn bị chi tiết cho khung hình, phác họa đối tượng

B2. Dựng hình (Modeling): Dựng hình 2D là bước đầu, kế tiếp là sử dụng nó để chuyển sang phiên bản 3D. Có 2 hướng làm việc là: dựng hình đa giác (polygonal modeling) và điêu khắc kỹ thuật số (digital sculpting).

B3. Chất liệu và Bề mặt (Shading & Texturing): Bề mặt và màu sắc được thêm vào bằng cách áp hình ảnh 2d vào đối tượng 3d, hoặc được tô trực tiếp vào mô hình như vẽ lên bảng vẽ.

B4. Ánh sáng (Lighting): Đổ bóng đối tượng, điều chỉnh nguồn sáng.

B5. Diễn hoạt (Animation): Mô hình 3D được điều khiển bởi "xương" (rigs), Các xương có khả năng điều khiển tay, chân, mặt, và tư thế của nhân vật.

B6. Kết xuất hình ảnh (Rendering) và xử lý hậu kì (post-production):

- Hoàn thiện ánh sáng: Bóng đổ và phản chiếu phải được tính toán.
- Hiệu ứng đặc biệt: nhất là với những hiệu ứng như mờ do chiều sâu không gian (depth of field blurring), khói, sương mù, và các vụ nổ.
- Xử lý hậu kì: Nếu độ sáng, màu sắc và độ tương phản cần chỉnh sửa, chúng sẽ được thực hiện vào lúc này.

8. Tăng tốc xử lý đồ họa (Áp dụng với Java)

	Process	Thread
Khái niệm	Một chương trình đang chạy được gọi là một **process**.	Một chương trình chạy có thể có nhiều **thread**, Cho phép chương trình đó chạy trên nhiều luồng một cách "đồng thời".
Không gian địa chỉ	Mỗi **process** có một không gian địa chỉ riêng biệt.	Tất cả **thread** thuộc một **process** chia sẻ không gian địa chỉ với nhau, hợp chúng lại thành một tiến trình.
Đa nhiệm	Đa nhiệm dựa trên process cho phép máy tính chạy 2 hoặc nhiều hơn 2 chương trình đồng thời.	Đa nhiệm dựa trên thread cho phép một chương trình chạy trên 2 hoặc nhiều luồng đồng thời.
Giao tiếp	Giao tiếp giữa 2 tiến trình là tốn kém và bị giới hạn.	Giao tiếp giữa 2 thread ít tốn kém hơn so với tiến trình.
Thành phần	Một tiến trình có: không gian địa chỉ, biến global, xử lí tín hiệu, những tiến trình con, thông tin tính toán.	Một thread có: thanh ghi, trạng thái, stack, bộ đếm chương trình.
Điều khiển	Đa nhiệm dựa trên process không thuộc quyền kiểm soát của Java.	Đa nhiệm dựa trên thread phụ thuộc quyền kiểm soát của Java.
Ví dụ	Khi chạy một ứng dụng Java thì đó được gọi là một tiến trình.	Một ứng dụng đếm từ trong 1000 file, có sử dụng 4 luồng để chạy đồng thời.

Trong multi-threading có tồn tại 2 khái niệm là Concurrency (đồng thời) và Parallelism (song song).

- Concurrency nghĩa là những tác vụ có thể bắt đầu, chạy, và hoàn thành trong những khoảng thời gian chồng chéo lên nhau mà không theo thứ tự nào cả.
- Còn Parallelism là nhiều tác vụ hoặc một phần của tác vụ chạy đồng thời tại cùng một thời điểm.

9. Kỹ thuật đồ họa Fractal

Đồ họa fractal là một vật thể hình học thường có hình dạng gấp khúc trên mọi tỷ lệ phóng đại, và có thể được tách ra thành từng phần: mỗi phần trông giống như hình tổng thể, nhưng ở tỷ lệ phóng đại nhỏ hơn. Như vậy phân dạng có vô tận các chi tiết, các chi tiết này có thể có cấu trúc tự đồng dạng ở các tỷ lệ phóng đại khác nhau. Nhiều trường hợp, có thể tạo ra phân dạng bằng việc lặp lại một mẫu toán học, theo phép hồi quy. Đường cong Fractal không thể được mô tả như đường hai chiều thông thường, mặt Fractal không thể mô tả như mặt 3 chiều mà đối tượng Fractal có thêm chiều hữu tỷ. Mặc dù các đối tượng Fractal trong từng trường hợp cụ thể chỉ chứa một số hữu hạn chi tiết, nhưng nó chứa đựng bản chất cho phép mô tả vô hạn chi tiết, tức là tại một thời điểm xác định thì là hữu hạn, nhưng xét về tổng thể là vô hạn vì bản chất Fractal cho phép phóng đại lên một mức độ bất kỳ một chi tiết tùy ý.

Các thực thể Fractal có 3 đặc tính quan trọng:

- Tự tương tự (self-similarity)
- Tự tương tựđa phần (statistical self-similarity)
- Tự Affine

10. Công nghệ nhận dạng hình ảnh đối tượng đồ họa, công nghệ Deep face phát triển cho các ứng dụng đồ họa 3D

OpenCV (Open Computer Vision) là một thư viện mã nguồn mở chuyên dùng để xử lý các vấn đề liên quan đến thị giác máy tính hay còn gọi theo tên thông dụng khác là xử lý hình ảnh hoặc nhận dạng hình ảnh. Như đã nói là một thư viện rất đồ sộ và có thể được chia làm 4 phần chính:

- CxCore: Chứa các cấu trúc cơ bản như điểm, đường, dãy, mặt, ma trận… và các thao tác cấp thấp liên quan.
- CV: Chứa hầu hết các thao tác liên quan đến việc xử lý ảnh ở cấp thấp như lọc ảnh, trích biên, phân vùng, tìm contour, biến đổi Fourier…
- HighGUI: Các thao tác lên những file ảnh và file Video như đọc ảnh, hiển thị ảnh, chuyển đổi định dạng…
- CvCam: Làm việc với Camera.

Thư viện OpenCV bao gồm một số tính năng nổi bật như:

- Bộ công cụ hỗ trợ 2D và 3D
- Nhận diện khuôn mặt
- Nhận diện cử chỉ
- Nhận dạng chuyển động, đối tượng, hành vi,…
- Tương tác giữa con người và máy tính

- Điều khiển Robot
- Hỗ trợ thực tế tăng cường

Gói EmguCV của OpenCV

Khái niệm:

- Là một cross platform .NET, một thư viện xử lý hình ảnh mạnh mẽ dành riêng cho các ngôn ngữ .NET, cho phép gọi được chức năng của OpenCV là từ .NET.
- Tương thích ngôn ngữ như: C#, VB, VC ++, Iron Python...
- Wrapper có thể được biên dịch bởi Visual Studio, Xamarin Studio và Unity.
- Có thể chạy trên Windows, Linux, Mac OS X, iOS, Android và Windows Phone.