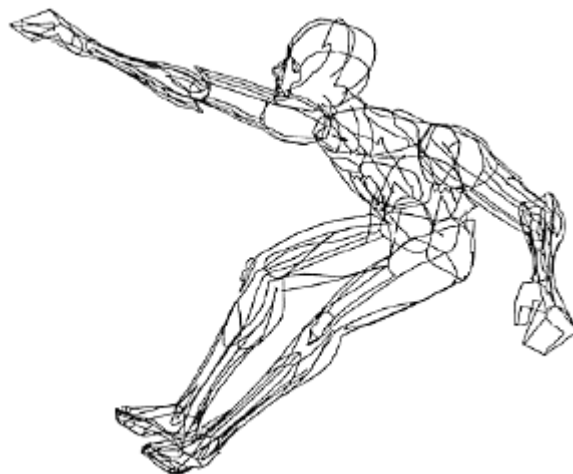


Tổng quan về đồ họa máy tính

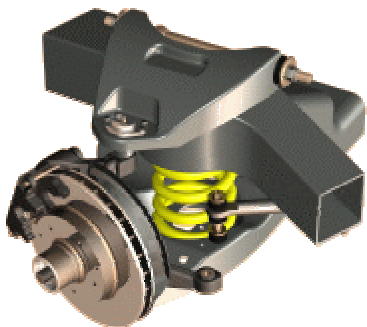
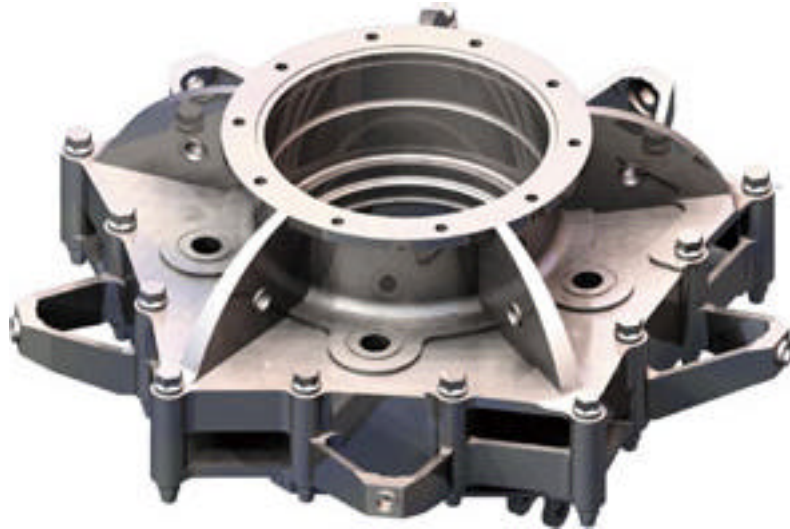
Khái niệm

- Đồ họa máy tính có thể được hiểu như là tất cả những gì liên quan đến việc tạo ra ảnh (image) bằng máy tính. Chúng bao gồm : tạo, lưu trữ, thao tác trên các mô hình (model) và các ảnh.
- Thuật ngữ đồ họa máy tính (computer graphics) do William Fetter đặt ra năm 1960 để mô tả một cách thiết kế mới khi đang làm việc tại hãng Boeing.
- Với cách này, anh ta đã tạo nhiều ảnh có thể sử dụng lại để có thể dễ dàng thiết kế buồng lái của phi công theo ý muốn.



Một số ứng dụng của đồ họa máy tính

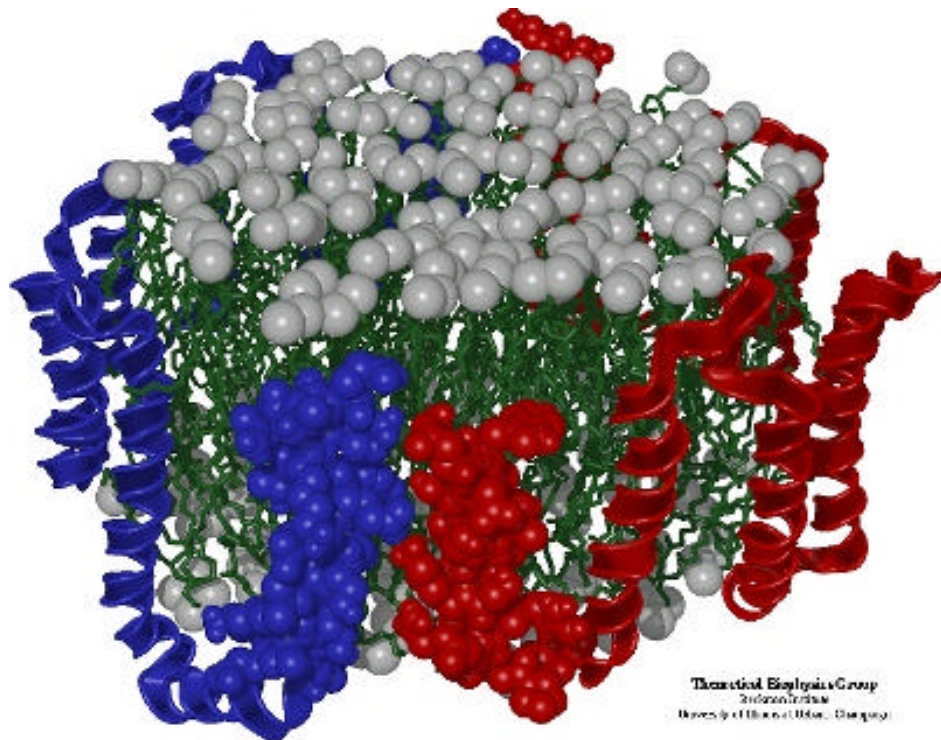
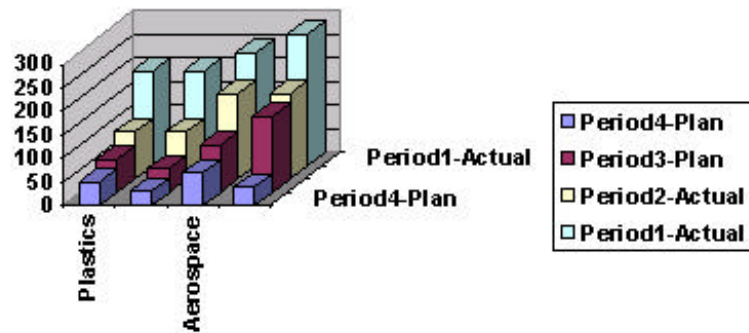
- Hỗ trợ thiết kế (CAD - Computer Aided Design)



Gồm hai bước chính

- ◆ Phác thảo của phần khung(wireframe outline) mà từ đó có thể thấy được toàn bộ hình dạng và các thành phần bên trong của các đối tượng. Sử dụng kỹ thuật này, người thiết kế sẽ dễ dàng nhận thấy ngay các thay đổi của đối tượng khi tiến hành hiệu chỉnh các chi tiết hay thay đổi góc nhìn,
- ◆ Kết hợp các mô hình chiếu sáng, tô màu và tạo bóng bề mặt để tạo ra kết quả cuối cùng rất gần với thế giới thực.

- Visualization

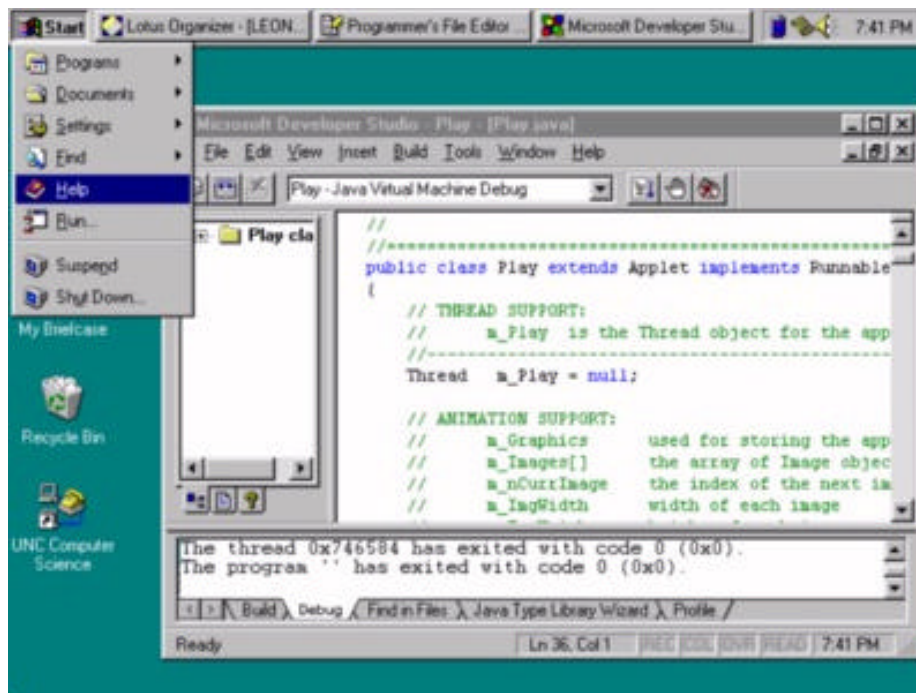


- ◆ Dùng phát sinh các biểu đồ, đồ thị, ... trong việc minh họa mối quan hệ giữa nhiều đối tượng với nhau.
- ◆ Tóm lược các dữ liệu về tài chính, thống kê, kinh tế, khoa học, toán học, ... giúp cho việc nghiên cứu, quản lí, ... một cách có hiệu quả.

- Giải trí



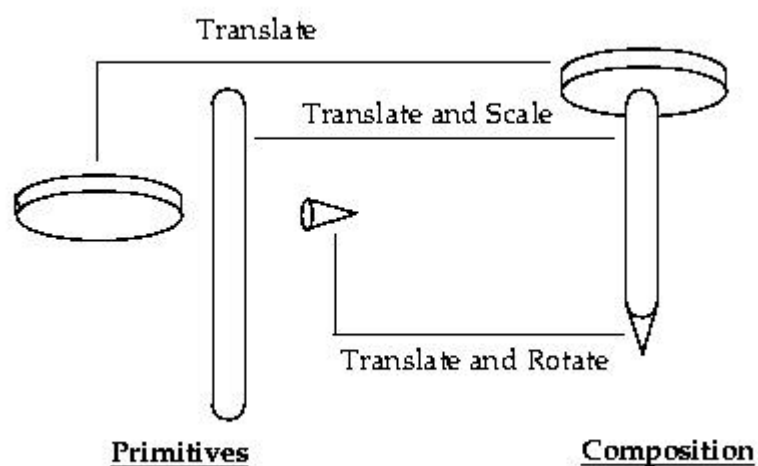
- Tạo giao diện



Tổng quan về một hệ đồ họa

- Các thành phần phần cứng
 - ◆ Thiết bị hiển thị : màn hình, máy in, ...
 - ◆ Thiết bị nhập : bàn phím, chuột, ...
- Các công cụ phần mềm
 - ◆ Công cụ ứng dụng (application package) : Được thiết kế cho các người sử dụng để tạo ra các hình ảnh mà không cần quan tâm tới các thao tác bên trong hoạt động như thế nào. Ví dụ : AutoCAD, Adobe Photoshop, 3D Studio, ...
 - ◆ Công cụ lập trình (programming package) : Cung cấp một tập các hàm đồ họa có thể được dùng trong các ngôn ngữ lập trình cấp cao như C, Pascal, ... Ví dụ : GRAPH.TPU, GRAPHICS.LIB, Open GL, ...
- Các chuẩn phần mềm
 - ◆ Ra đời để đáp ứng tính tương thích : Nếu các phần mềm được thiết kế với các hàm đồ họa chuẩn chúng có thể dùng được cho nhiều hệ phần cứng và môi trường làm việc khác nhau.
 - ◆ GKS (Graphics Kernel System) là chuẩn ra đời đầu tiên cho việc phát triển các phần mềm đồ họa. Ban đầu GKS được thiết kế chỉ dùng cho tập các công cụ đồ họa hai chiều, sau đó mới được mở rộng ra cho đồ họa ba chiều.
 - ◆ Các hàm của GKS thực sự chỉ là các mô tả trừu tượng, độc lập với bất kì ngôn ngữ lập trình nào. Để cài đặt một chuẩn đồ họa cho ngôn ngữ cụ thể nào, các cú pháp tương ứng sẽ được xác định và cụ thể hóa.

- Các thành phần của công cụ lập trình
 - ◆ Tập các công cụ tạo ra các đối tượng đồ họa cơ sở như điểm, đoạn thẳng, đường cong, vùng tô, kí tự, ...
 - ◆ Tập các công cụ thay đổi thuộc tính của các đối tượng cơ sở kể trên như màu sắc, kiểu đường, kiểu chữ, mẫu tô...
 - ◆ Tập các công cụ thực hiện các phép biến đổi hình học dùng để thay đổi kích thước, vị trí, hướng, ...
 - ◆ Tập các công cụ biến đổi hệ quan sát dùng để xác định vị trí quan sát của các đối tượng và vị trí trên thiết bị hiển thị đối tượng.
 - ◆ Tập các công cụ nhập liệu : các ứng dụng đồ họa có thể sử dụng nhiều loại thiết bị nhập khác nhau như chuột, bàn phím, bút vẽ, bảng, ... để điều khiển và xử lí dòng dữ liệu nhập.
 - ◆ Tập các công cụ chứa các thao tác dùng cho quản lí và điều khiển như khởi tạo và đóng chế độ đồ họa, xóa toàn bộ màn hình, ...



Hai mô hình cơ bản của ứng dụng đồ họa

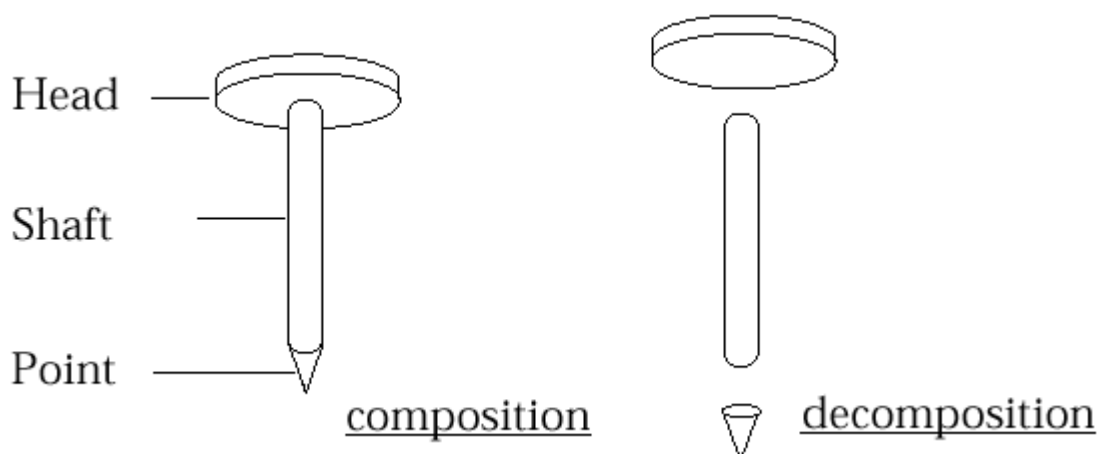
- Ứng dụng đồ họa dựa trên mẫu số hóa (sampled-based graphics)
 - ◆ Các pixel (điểm ảnh) được tạo ra bởi thao tác số hóa ảnh bằng cách sử dụng các chương trình vẽ dựa trên mẫu số hóa hay máy quét.
 - ◆ Các ứng dụng thuộc dạng này gồm : PaintBrush, Adobe Photoshop, ...
- Ứng dụng đồ họa dựa trên đặc trưng hình học (geometry-based graphics)
 - ◆ Dùng các đặc trưng hình học và các thuộc tính để mô tả đối tượng. Sau đó các đối tượng sẽ được số hóa để phục vụ cho hiển thị.
 - ◆ Các ứng dụng thuộc dạng này : Adobe Illustrator, AutoCAD, ...

Ứng dụng đồ họa dựa trên mẫu số hóa

- Các đối tượng đồ họa được tạo ra bởi lưới các pixel rời rạc.
- Các pixel này có một mô tả về tọa độ để xác định vị trí và giá trị mẫu (sample values), thông thường là độ sáng hay màu sắc.
- Các pixel này có thể được tạo ra bằng các chương trình vẽ, máy quét, ...
- Khi một ảnh được xác định bởi tập các pixel, chúng có thể có các thao tác :
 - ◆ Biên tập ảnh (image editing) : cắt, dán các vùng trên ảnh, sử dụng các công cụ tô màu để hiệu chỉnh, ...
 - ◆ Xử lý ảnh (image processing) : sử dụng các thuật toán để thay đổi ảnh mà không có sự can thiệp của người dùng, bao gồm : làm nhòe ảnh (blurring), làm nét ảnh (sharpening), dò đường biên (edge-detection), cân chỉnh màu sắc, ..
- Một số thuận lợi
 - ◆ Dễ dàng thay đổi ảnh bằng cách thay đổi màu sắc hay vị trí của các pixel, ví dụ như lấy ảnh âm bản, ...
 - ◆ Có thể di chuyển các vùng ảnh từ nơi này sang nơi khác dễ dàng.
- Một số bất lợi
 - ◆ Không thể xem xét đối tượng từ các góc nhìn khác nhau.
 - ◆ Hiệu chỉnh về thuộc tính hình học, kích thước phức tạp.

Ứng dụng đồ họa dựa trên đặc trưng hình học

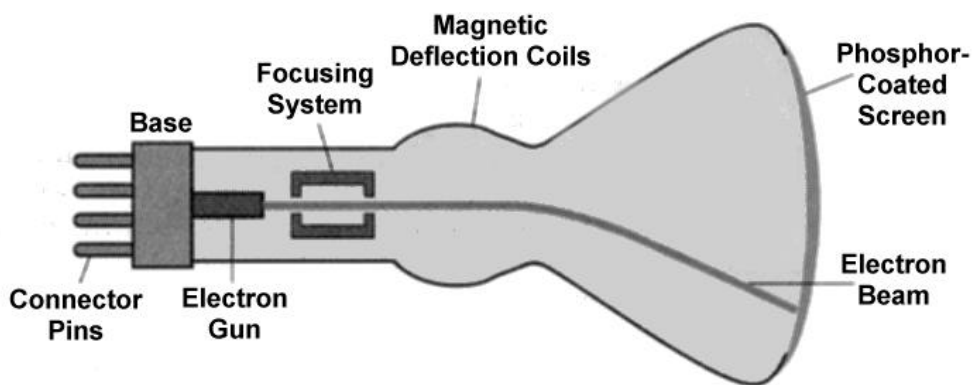
- Các đối tượng đồ họa cơ sở như đoạn thẳng, đa giác, ... được lưu trữ bằng các mô hình (model) và các thuộc tính (attribute) của chúng.
 - ◆ Các mô hình thực chất là các mô tả toán học, ví dụ đoạn thẳng được mô hình bằng hai điểm đầu, cuối, ...
 - ◆ Các thuộc tính được dùng để mô tả cách mà các đối tượng được hiển thị ví dụ như màu sắc, độ dày, ..
- Các ảnh được tạo bởi tập các pixel thông qua việc số hóa các đặc trưng hình học phục vụ cho mỗi yêu cầu hiển thị. Các ảnh có thể khác nhau tùy vào mỗi yêu cầu hiển thị khác nhau, nhưng đều xuất phát từ một mô hình.
- Người dùng không thao tác trực tiếp với từng pixel của ứng dụng dạng này mà thao tác trên các thành phần hình học của đối tượng, sau đó số hóa lại rồi mới hiển thị.



Thiết bị hiển thị : Màn hình

Cấu tạo của CRT

- Một chùm các tia điện tử (tia âm cực) phát ra từ một súng điện tử, vượt qua các hệ thống hội tụ (focusing) và dẫn hướng (deflection) sẽ hướng tới các vị trí xác định trên màn hình được phủ một lớp phosphor.

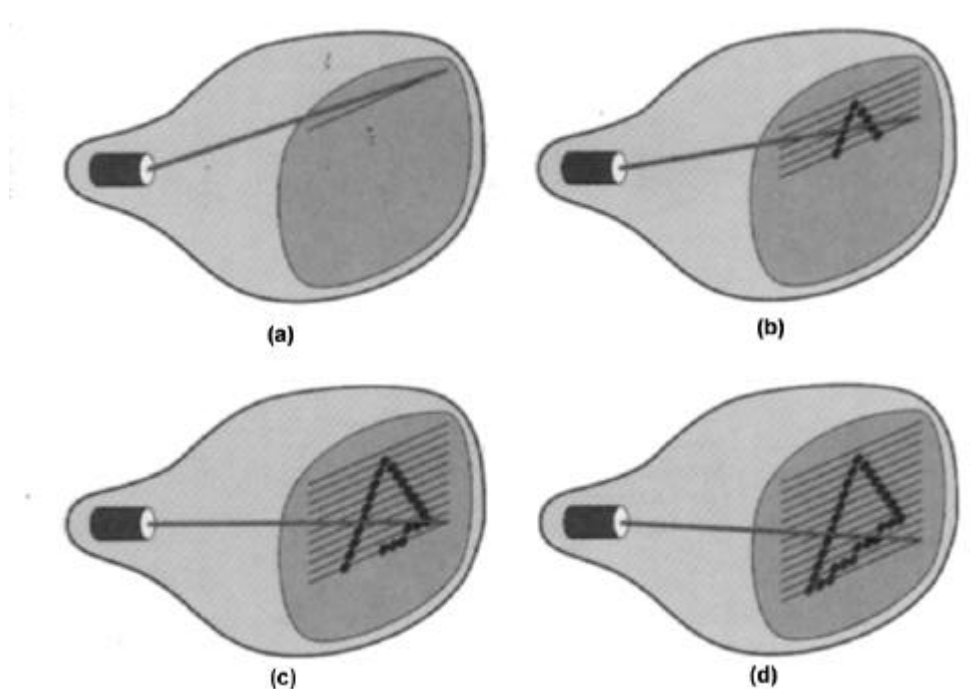


- Tại mỗi vị trí tương tác với tia điện tử, hạt phosphor sẽ phát ra một chấm sáng nhỏ. Vì ánh sáng phát ra bởi các hạt phosphor mờ dần rất nhanh nên cần phải có một cách nào đó để duy trì ảnh trên màn hình. Một trong các cách đó là lặp đi lặp lại nhiều lần việc vẽ lại ảnh thật nhanh bằng cách hướng các tia điện tử trở lại vị trí cũ. Kiểu hiển thị này gọi là refresh CRT.
- Có nhiều loại phosphor được dùng trong một CRT. Ngoài màu sắc ra, điểm khác nhau chính giữa các loại phosphor là “độ bền“ (persistent), đó là khoảng thời gian phát sáng sau khi tia CRT không còn tác động.

- Lớp phosphor có độ bền thấp cần tốc độ làm tươi cao hơn để giữ cho hình ảnh trên màn hình khỏi nhòe. Loại này thường rất tốt cho hoạt hình, rất cần thay đổi hình ảnh liên tục. Lớp phosphor có độ bền cao thường được dùng cho việc hiển thị các ảnh tĩnh, độ phức tạp cao. Mặc dù một số loại phosphor có độ bền lớn hơn 1 giây, tuy nhiên các màn hình đồ họa thường được xây dựng với độ bền dao động từ 10 đến 60 micro giây.
- Số lượng tối đa các điểm có thể hiển thị trên một CRT được gọi là độ phân giải (resolution).
- Kích thước vật lí của màn hình đồ họa được tính từ độ dài của đường chéo màn hình, thường dao động từ 12 đến 27 inch hoặc lớn hơn. Một màn hình CRT có thể được kết hợp với nhiều loại máy khác nhau, do đó số lượng các điểm trên màn hình có thể được vẽ thật sự còn tùy thuộc vào khả năng của hệ thống mà nó kết hợp vào.
- Tỉ số phương là tỉ lệ của các điểm dọc và các điểm ngang cần để phát sinh các đoạn thẳng có độ dài đơn vị theo cả hai hướng trên màn hình (trong một số trường hợp người ta thường dùng tỉ số phương như là tỉ số của các điểm theo chiều ngang so với các điểm theo chiều dọc). Với các màn hình có tỉ số phương khác 1, dễ dàng nhận thấy là các hình vuông hiển thị trên nó sẽ có dạng hình chữ nhật, các hình tròn sẽ có dạng hình ellipse.

Màn hình dạng điểm (raster - scan display):

- Chùm tia điện tử sẽ được quét ngang qua màn hình, mỗi lần một dòng và quét tuần tự từ trên xuống dưới. Sự bật tắt của các điểm sáng trên màn hình phụ thuộc vào cường độ của tia điện tử và đây chính là cơ sở của việc tạo ra hình ảnh trên màn hình.
- Mỗi điểm trên màn hình được gọi là một pixel. Các thông tin về hình ảnh hiển thị trên màn hình được lưu trữ trong một vùng bộ nhớ gọi là vùng đệm làm tươi (refresh buffer) hay là vùng đệm khung (frame buffer). Vùng bộ nhớ này lưu trữ tập các giá trị cường độ sáng của toàn bộ các điểm trên màn hình và luôn luôn tồn tại một song ánh giữa mỗi điểm trên màn hình và mỗi phần tử trong vùng này.
- Để thay đổi các hình ảnh cần hiển thị, các giá trị tương ứng với vị trí và độ sáng phải được đặt vào vùng đệm khung.



- Để tạo ra các ảnh đen trắng, đơn giản chỉ cần lưu thông tin của mỗi pixel bằng 1 bit (các giá trị 0, 1 sẽ tượng trưng cho việc tắt (tối), bật (sáng) pixel trên màn hình). Trong trường hợp ảnh nhiều màu, người ta cần nhiều bit hơn, nếu thông tin của mỗi pixel được lưu bằng b bit, thì ta có thể có 2^b giá trị màu phân biệt cho pixel đó.
- Trong các màn hình màu, người ta định nghĩa tập các màu làm việc trong một bảng tra (LookUp Table - LUT). Mỗi phần tử của LUT định nghĩa một bộ ba giá trị R (Red), G (Green), B (Blue) mô tả một màu nào đó. Khi cần sử dụng một màu, ta chỉ cần chỉ định số thứ tự (index) tương ứng của màu đó trong LUT. Bảng LUT có thể được thay đổi bởi các ứng dụng và người lập trình có thể can thiệp điều khiển. Với cách làm này chúng ta có thể tiết kiệm không gian lưu trữ cho mỗi phần tử trong vùng đệm khung.
- Số phần tử của LUT được xác định từ số lượng các bits/pixel. Nếu mỗi phần tử của vùng đệm khung dùng b bits để lưu thông tin của một pixel, thì bảng LUT có 2^b phần tử. Nếu $b=8$, LUT sẽ có $2^8=256$ phần tử, đó chính là số màu có thể được hiển thị cùng một lúc trên màn hình.
- Việc làm tươi trên màn hình dạng này được thực hiện ở tốc độ 60 đến 80 frame/giây. Đôi khi tốc độ làm tươi còn được biểu diễn bằng đơn vị Hertz (Hz – số chu kì/ giây), trong đó một chu kì tương ứng với một frame.

- Khi đạt đến cuối mỗi dòng quét, tia điện tử quay trở lại bên trái của màn hình để bắt đầu dòng quét kế tiếp. Việc quay trở lại phía trái màn hình sau khi làm tươi mỗi dòng quét được gọi là tia hồi ngang (horizontal retrace). Và tới cuối mỗi frame, tia điện tử (tia hồi dọc – vertical retrace) quay trở lại góc trên bên trái của màn hình để chuẩn bị bắt đầu frame kế tiếp.
- Trong một số màn hình, mỗi frame được hiển thị thành hai giai đoạn sử dụng kỹ thuật làm tươi đan xen nhau (interlaced refresh). Ở giai đoạn đầu tiên, tia quét sẽ quét một số dòng từ trên xuống dưới, sau tia hồi dọc, các dòng còn lại sẽ được quét. Việc đan xen các dòng quét này cho phép chúng ta thấy được toàn màn hình hiển thị chỉ trong một nửa thời gian so với dùng để quét tất cả các dòng một lần từ trên xuống dưới. Kỹ thuật này thường được dùng cho loại màn hình có tốc độ làm tươi thấp.

