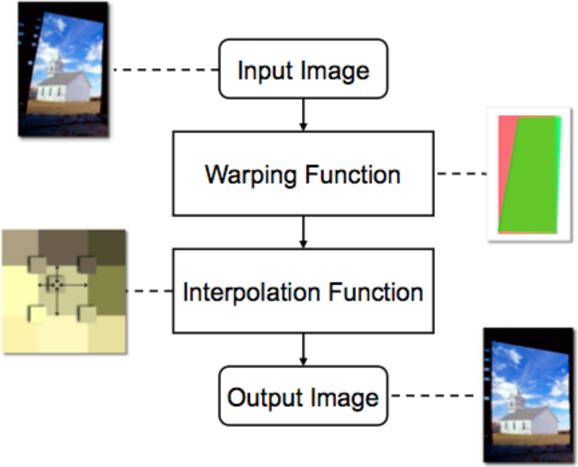
Xử lý ảnh - *warping* ảnh và xoay ảnh

Đỗ Thanh Hà

Bộ môn Tin học
Khoa Toán - Cơ - Tin học
Trường Đại học Khoa học Tự nhiên

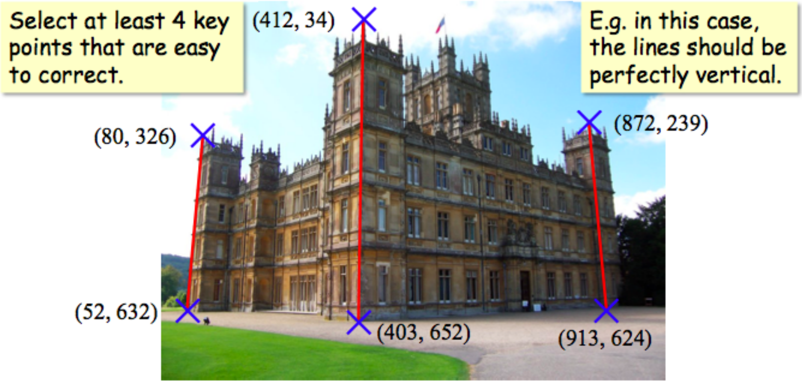
Ánh xạ hình học



Ánh xạ hình học

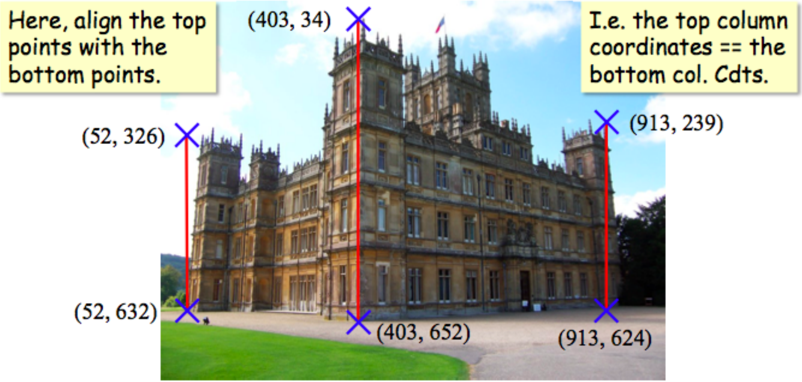
1. Giả sử có ảnh input $\mathbf{I}$
2. Tính kích thước $R_{out} \times C_{out} \times B$ và khởi tạo ảnh đầu ra $\mathbf{J}$
3. Tạo ánh xạ (hàm *warping* Φ) như sau
   - Khởi tạo một mảng Φ có kích thước $R_{out} \times C_{out} \times 2$
   - Với mỗi vị trí pixel (r, c) trong $\mathbf{J}$ tìm vị trí pixel giá trị thực (r_f, c_f) trong $\mathbf{I}$
   - Đặt $\Phi(r, c, 1) = r_f$ và $\Phi(r, c, 2) = c_f$
4. Tạo một hàm nội suy Θ, hàm này sẽ sinh ra giá trị pixel từ các giá trị của $\mathbf{I}$ trong miền hàng xóm $\mathfrak{N}(r_f, c_f)$
5. Sau đó đặt $\mathbf{J} = \Theta\{\mathbf{I}; \mathfrak{N}(r_f, c_f)\}$

Warping tuyến tính của ảnh



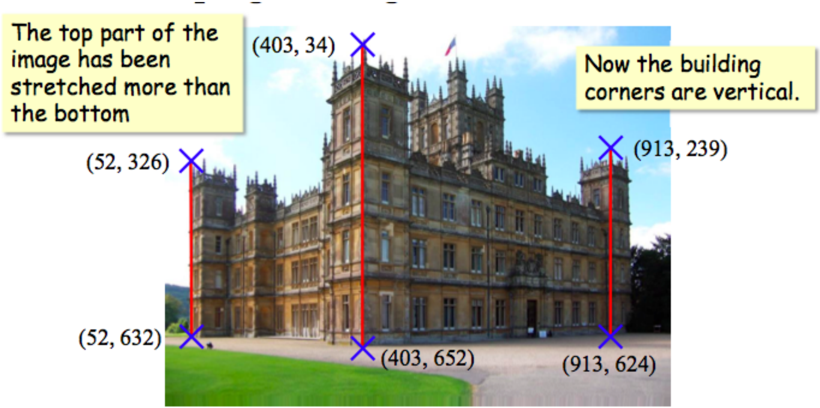
Lựa chọn các điểm để correction

Warping tuyến tính của ảnh



Các điểm correction mục tiêu

Warping tuyến tính của ảnh



Kết quả

Cho tập $\mathbf{X}$ các điểm trong ảnh $\mathbf{I}$ và tập $\mathbf{Y}$ các điểm mục tiêu; cần tìm $\mathbf{H}$ sao cho $\mathbf{Y} = \mathbf{HX}$

Trong đó

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_p \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \dots & x_p \\ y_1 & y_2 & \dots & y_p \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \dots & \mathbf{y}_p \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & \dots & u_p \\ v_1 & v_2 & \dots & v_p \end{bmatrix} \text{ và } \mathbf{H} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix}$$

$$\mathbf{H}^{-1} \begin{bmatrix} c_\mathbf{J} \\ r_\mathbf{J} \end{bmatrix} = \begin{bmatrix} c_\mathbf{I} \\ r_\mathbf{I} \end{bmatrix} \text{ khi đó } \mathbf{J}(r_\mathbf{J}, c_\mathbf{J}) = \mathbf{I}\left(\mathbf{H}^{-1} \begin{bmatrix} c_\mathbf{J} \\ r_\mathbf{J} \end{bmatrix}\right) \text{ là ảnh wraped}$$

Warping tuyến tính của ảnh - Thực hiện như thế nào

Tuy nhiên, công thức ở slide trước làm việc chưa thực tốt vì nó là phép biến đổi 2D. Chúng ta cần một phép biến đổi 3D affine, chiều thứ 3 cho phép chúng ta mô hình các phép chiếu tổng quát hơn. Do đó người ta thường sử dụng toạ độ đồng nhất

Warping tuyến tính của ảnh - Toạ độ đồng nhất

- Gán giá trị 1 cho chiều thứ 3 của mỗi vị trí pixel input

$$\mathbf{x}_i = \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

- $\mathbf{H}$ là ma trận kích thước 3×3
- Vị trí pixel được ánh xạ cũng là 3D. Gọi thành phần thứ 3 là k_i và viết 2 thành phần khác như là tỉ lệ của k_i

$$\mathbf{y}_i = \begin{bmatrix} k_i u_i \\ k_i v_i \\ k_i \end{bmatrix}$$

Warping tuyến tính của ảnh - Toạ độ đồng nhất

- Vị trí pixel trong dạng đồng nhất được viết dưới dạng:

$$\mathbf{X} = \begin{bmatrix} x_1 & x_2 & \dots & x_p \\ y_1 & y_2 & \dots & y_p \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

- **H** là ma trận kích thước 3×3: $\mathbf{Y} = \mathbf{HX}$
- Vị trí pixel được ánh xạ trong toạ độ đồng nhất có dạng như sau, trong đó mỗi k_i thường là khác nhau

$$\mathbf{Y} = \begin{bmatrix} k_1 u_1 & k_2 u_2 & \dots & k_p u_p \\ k_1 v_1 & k_2 v_2 & \dots & k_p v_p \\ k_1 & k_2 & \dots & k_p \end{bmatrix}$$

- Mỗi vector $\mathbf{x}_i$ và $\mathbf{y}_i$ là các thành phần của phép biến đổi

$$\begin{bmatrix} k_i u_i \\ k_i v_i \\ k_i \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

Warping tuyến tính của ảnh - Đạo hàm

- Với mỗi vị trí pixel, có 3 phương trình 3 ẩn

$$k_i u_i = h_{11} x_i + h_{12} y_i + h_{13}$$
$$k_i v_i = h_{21} v_i + h_{22} v_i + h_{23}$$
$$k_i = h_{31} x_i + h_{32} y_i + h_{33}$$

- Chia phương trình thứ nhất và thứ hai cho phương trình thứ 3

$$u_i = \frac{h_{11} x_i + h_{12} y_i + h_{13}}{h_{31} x_i + h_{32} y_i + h_{33}}$$
$$v_i = \frac{h_{21} v_i + h_{22} v_i + h_{23}}{h_{31} x_i + h_{32} y_i + h_{33}}$$

- Chia cả tử số và mẫu số cho h_{33}. So đó gán nhãn lại các hệ số

$$u_i = \frac{h_{11} x_i + h_{12} y_i + h_{13}}{h_{31} x_i + h_{32} y_i + 1}$$
$$v_i = \frac{h_{21} v_i + h_{22} v_i + h_{23}}{h_{31} x_i + h_{32} y_i + 1}$$

- Nhân cả hai với mẫu số bên phải

$$
\begin{aligned}
(h_{31}x_i + h_{32}y_i + 1)u_i &= h_{11}x_i + h_{12}y_i + h_{13} \\
(h_{31}x_i + h_{32}y_i + 1)v_i &= h_{21}x_i + h_{22}y_i + h_{23}
\end{aligned}
$$

- Trừ cả hai vế cho vế phải

$$
\begin{aligned}
-h_{11}x_i - h_{12}y_i - h_{13} + h_{31}x_iu_i + h_{32}y_iu_i + u_i &= 0 \\
-h_{21}x_i - h_{22}y_i - h_{23} + h_{31}x_iv_i + h_{32}y_iv_i + v_i &= 0
\end{aligned}
$$

- Viết dưới dạng phương trình ma trận

$$
\begin{bmatrix}
-x_i & -y_i & -1 & 0 & 0 & 0 & x_iu_i & y_iu_i & u_i \\
0 & 0 & 0 & -x_i & -y_i & -1 & x_iv_i & y_iv_i & v_i
\end{bmatrix} \mathbf{h} = 0
$$

- với ma trận **H** được viết dưới dạng vector **h**

$$
\mathbf{h} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & 1 \end{bmatrix}^T
$$

Warping tuyến tính của ảnh - Đạo hàm

- Tập hợp $p \geq 4$ các vị trí pixel, ánh xạ chúng như mong muốn và tạo ra ma trận $\mathbf{A}$

$$\mathbf{A} = \begin{bmatrix} -x_1 & -y_1 & -1 & 0 & 0 & 0 & x_1 u_1 & y_1 u_1 & u_1 \\ 0 & 0 & 0 & -x_1 & -y_1 & -1 & x_1 v_1 & y_1 v_1 & v_1 \\ -x_2 & -y_2 & -1 & 0 & 0 & 0 & x_2 u_2 & y_2 u_2 & u_2 \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & x_2 v_2 & y_2 v_2 & v_2 \\ \ldots\ldots\ldots & & & & & & & & \\ -x_p & -y_p & -1 & 0 & 0 & 0 & x_p u_p & y_p u_p & u_p \\ 0 & 0 & 0 & -x_p & -y_p & -1 & x_p v_p & y_p v_p & v_p \end{bmatrix}$$

- Giải tìm $\mathbf{h}$ sao cho

$$\mathbf{h} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & 1 \end{bmatrix}^T$$
$$\mathbf{Ah} = 0$$

- Nếu định nghĩa **A** ở slide trước, giải thu được **H** mà kết quả sai, ta sử dụng dạng khác của ma trận **A** như sau

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & -x_1 & -y_1 & -1 & x_1 v_1 & y_1 v_1 & v_1 \\ x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1 u_1 & -y_1 u_1 & -u_1 \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & x_2 v_2 & y_2 v_2 & v_2 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2 u_2 & -y_2 u_2 & u_2 \\ \ldots\ldots. & & & & & & & & \\ 0 & 0 & 0 & -x_p & -y_p & -1 & x_p v_p & y_p v_p & v_p \\ x_p & y_p & 1 & 0 & 0 & 0 & -x_p u_p & -y_p u_p & -u_p \end{bmatrix}$$

- Giải tìm **h** sao cho

$$\mathbf{h} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & 1 \end{bmatrix}^T$$
$$\mathbf{Ah} = 0$$

Warping tuyến tính của ảnh - Đạo hàm

- Để tìm **h** tính sigular value decomposition (SVD) của **A**

$$svd(\mathbf{A}) = \mathbf{USV}^T$$

- **S** là ma trận đường chéo

$$\mathbf{S} = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ .. & \dots & & \dots \\ 0 & 0 & \dots & \sigma_p \end{bmatrix}$$

- Viết ma trận **V** dưới dạng các cột của nó

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \dots. & \mathbf{v}_p \end{bmatrix}$$

- Tìm σ_k: $k = arg\min\{\sigma_1, \sigma_2, ..., \sigma_p\}$
- Khi đó vector **h** được cho bởi vectro cột thứ k, $\mathbf{v}_k$

$$\mathbf{h} = \mathbf{v}_k$$

Warping tuyến tính của ảnh - Đạo hàm

- Nếu $\mathbf{v}_k$ là vector của $\mathbf{V}$ mà tương ứng với giá trị singular nhỏ nhất thì $\mathbf{H}$ có dạng

$$\mathbf{H} = \begin{bmatrix} v_{1k} & v_{2k} & v_{3k} \\ v_{4k} & v_{5k} & v_{6k} \\ v_{7k} & v_{8k} & v_{9k} \end{bmatrix}$$

- Ánh xạ $(r_{\mathbf{I}}, c_{\mathbf{I}})$ thành $(r_{\mathbf{J}}, c_{\mathbf{J}})$ thông qua $\mathbf{H}$ sao cho

$$\mathbf{J}(r_{\mathbf{J}}, c_{\mathbf{J}}) = \mathbf{I}(r_{\mathbf{I}}, c_{\mathbf{I}})$$

- Nhưng ta muốn quét ảnh output $\mathbf{J}$ và tại mỗi vị trí pixel $(r_{\mathbf{J}}, c_{\mathbf{J}})$ lấy giá trị từ $\mathbf{I}$ tại vị trí $(r_{\mathbf{I}}, c_{\mathbf{I}})$, do đó ta thực hiện

$$\mathbf{J}(r_{\mathbf{J}}, c_{\mathbf{J}}) = \mathbf{I}\left(N\left\{\mathbf{H}^{-1}\begin{bmatrix} c_{\mathbf{J}} \\ r_{\mathbf{J}} \\ 1 \end{bmatrix}\right\}\right)$$

Warping tuyến tính của ảnh - Remapping

- Ánh xạ ngược của $(r_{\mathbf{J}}, c_{\mathbf{J}}, 1)$ qua $\mathbf{H}^{-1}$ là $(kr_{\mathbf{J}}, kc_{\mathbf{J}}, k)$

$$\mathbf{H}^{-1}\begin{bmatrix} c_{\mathbf{J}} \\ r_{\mathbf{J}} \\ 1 \end{bmatrix} = \begin{bmatrix} k_{(r,c)}c_{\mathbf{J}} \\ k_{(r,c)}r_{\mathbf{J}} \\ k_{(r,c)} \end{bmatrix}$$

- và cần được chuẩn hoá như sau

$$N\left\{ \begin{bmatrix} k_{(r,c)}c_{\mathbf{J}} \\ k_{(r,c)}r_{\mathbf{J}} \\ k_{(r,c)} \end{bmatrix} \right\} = \frac{1}{k_{(r,c)}} \begin{bmatrix} k_{(r,c)}c_{\mathbf{I}} \\ k_{(r,c)}r_{\mathbf{I}} \end{bmatrix} = \begin{bmatrix} c_{\mathbf{I}} \\ r_{\mathbf{I}} \end{bmatrix}$$

- Do đó mỗi pixel tại vị trí $(r_{\mathbf{J}}, c_{\mathbf{J}})$ trong ảnh wraped

$$\mathbf{J}(r_{\mathbf{J}}, c_{\mathbf{J}}) = \mathbf{I}\left(N\left\{ \mathbf{H}^{-1}\begin{bmatrix} c_{\mathbf{J}} \\ r_{\mathbf{J}} \\ 1 \end{bmatrix} \right\} \right)$$

Cong tuyến tính của ảnh - Các bước

1. Chọn ít nhất 4 pixels trong $\mathbf{I}$
2. Chọn các vị trí đích bằng cách thay thế các giá trị của vị trí được lựa chọn
3. Xây dựng từ cặp các vị trí, ma trận $\mathbf{A}$ như mô tả trong slide 13
4. Tính SVD của $\mathbf{A} = \mathbf{USV}^T$
5. Lựa chọn vector $\mathbf{v}_k$ mà tương ứng với giá trị singular nhỏ nhất
6. Xây dựng $\mathbf{H}$ từ $\mathbf{v}_k$
7. Tính $\mathbf{H}^{-1}$
8. Tạo ra ảnh output $\mathbf{J}$
9. Với mỗi vị trí $(r_\mathbf{J}, c_\mathbf{J})$ trong $\mathbf{J}$, chọn $(r_\mathbf{I}, c_\mathbf{I})$ trong $\mathbf{I}$ sử dụng phương trình ở slide trước
10. Do $(r_\mathbf{I}, c_\mathbf{I})$ là các phân số, nội suy trên hàng xóm của $(r_\mathbf{I}, c_\mathbf{I})$ trong $\mathbf{I}$ để tính $\mathbf{J}(r_\mathbf{J}, c_\mathbf{J})$

Xoay ảnh

Xoay ảnh

- Kích thước ảnh

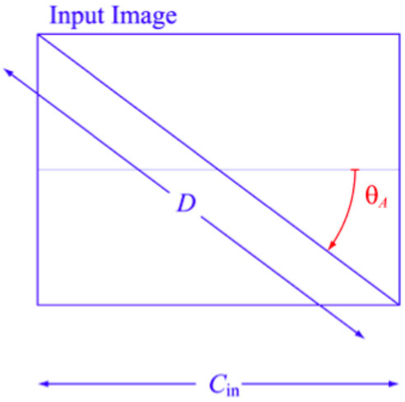
$$[R_{in}, C_{in}, B] = \text{size}(\mathbf{I})$$

- Góc quay

$$\theta_A = \tan^{-1}\left[\frac{R_{in}}{C_{in}}\right]$$

- Chiều dài đường chéo

$$D = \sqrt{R_{in}^2 + C_{in}^2}$$



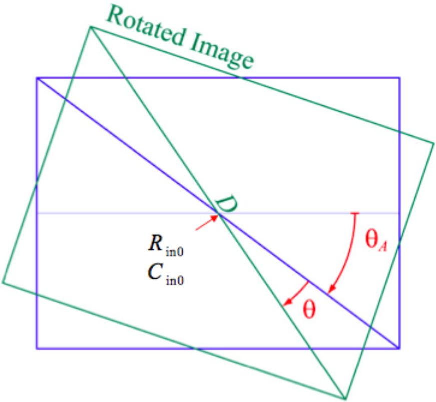
- góc quay θ
- ma trận quay

$$\mathbf{P}(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

- biến đổi ảnh ảnh đầu vào thành ảnh đầu ra

$$\begin{bmatrix} r \\ c \end{bmatrix} = \mathbf{P}(\theta) \begin{bmatrix} r_{in} - R_{in0} \\ c_{in} - C_{in0} \end{bmatrix} + \begin{bmatrix} R_{out0} \\ C_{out0} \end{bmatrix}$$

trong đó



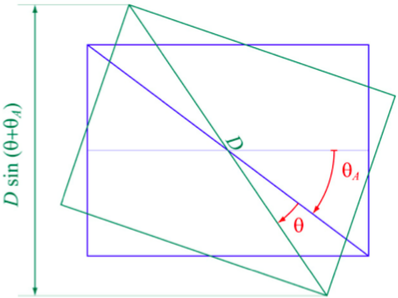
$(R_{in0}, C_{in0}) = (\frac{1}{2}R_{in}+1, \frac{1}{2}C_{in}+1)$; (R_{out0}, C_{out0}) là tâm của ảnh ouput

Xoay ảnh

- Tính chiều của ảnh đầu ra: số các hàng

  $$R_{out} = \text{round}(|D\sin(\theta + \theta_A)|)$$

  nếu $0^0 < \theta \leq 90^0$

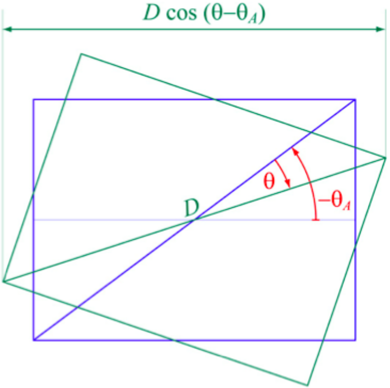


Xoay ảnh

- Tính chiều của ảnh đầu ra: số các cột

  $C_{out} = \text{round}(|D\cos(\theta - \theta_A)|)$

  nếu $0^0 < \theta \leq 90^0$



Xoay ảnh

Chiều của ảnh đầu ra phụ thuộc vảo giá trị của θ như sau

- Nếu $0^0 \leq \theta < 90^0$

$$R_{out} = \text{round}(|D\sin(\theta + \theta_A)|)$$

$$C_{out} = \text{round}(|D\cos(\theta - \theta_A)|)$$

- Nếu $90^0 \leq \theta < 180^0$

$$R_{out} = \text{round}(|D\cos(\theta - 90 - \theta_A)|)$$

$$C_{out} = \text{round}(|D\sin(\theta - 90 + \theta_A)|)$$

- Nếu $-90^0 \leq \theta < 0^0$

$$R_{out} = \text{round}(|D\sin(\theta - \theta_A)|)$$

$$C_{out} = \text{round}(|D\cos(\theta + \theta_A)|)$$

- Nếu $-180^0 \leq \theta < -90^0$

$$R_{out} = \text{round}(|D\cos(\theta + 90 + \theta_A)|)$$

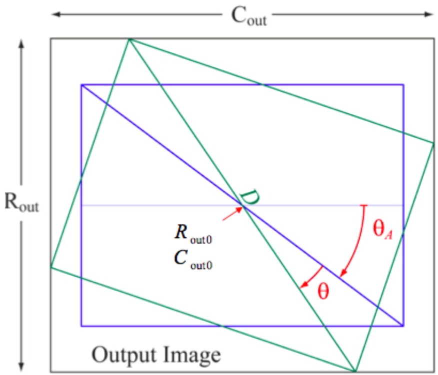
$$C_{out} = \text{round}(|D\sin(\theta + 90 - \theta_A)|)$$

Xoay ảnh

- Khởi tạo ảnh output với chiều (R_{out}, C_{out})
- và tâm của ảnh đầu ra

$$R_{out0} = \left\lfloor \frac{1}{2} R_{out} \right\rfloor + 1$$

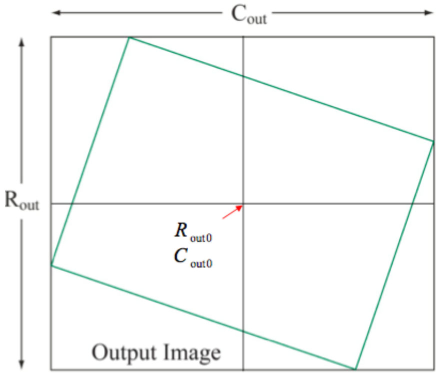
$$C_{out0} = \left\lfloor \frac{1}{2} C_{out} \right\rfloor + 1$$



$$\mathbf{P}^{-1}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \mathbf{P}(-\theta)$$

Làm ngược lại: với mỗi vị trí đầu ra (r, c) lựa chọn vị trí đầu vào (r_f, c_f) bằng cách quay (r, c) xung quanh tâm của ảnh một góc $-\theta$

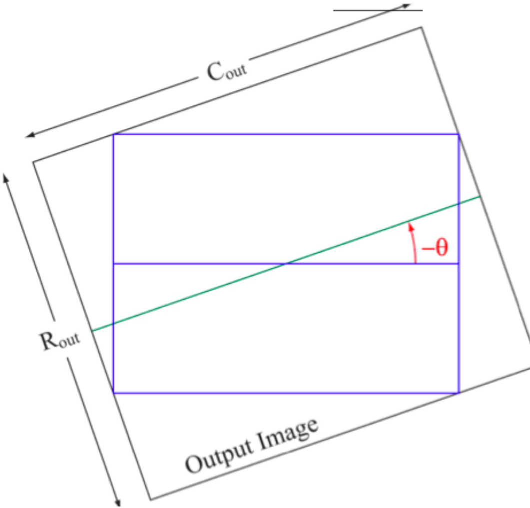
$$\Phi(r, c, :) = \begin{bmatrix} r_f \\ c_f \end{bmatrix} = \mathbf{P}^{-1}(\theta) \begin{bmatrix} r - R_{out0} \\ c - C_{out0} \end{bmatrix} + \begin{bmatrix} R_{in0} \\ C_{in0} \end{bmatrix}$$



Xoay ảnh

Xoay ảnh input một góc θ tương ứng với xoay ảnh output một góc $-\theta$

$$\Phi(r,c,:) = \mathbf{P}^{-1}(\theta)\begin{bmatrix} r - R_{out0} \\ c - C_{out0} \end{bmatrix} + \begin{bmatrix} R_{in0} \\ C_{in0} \end{bmatrix}$$



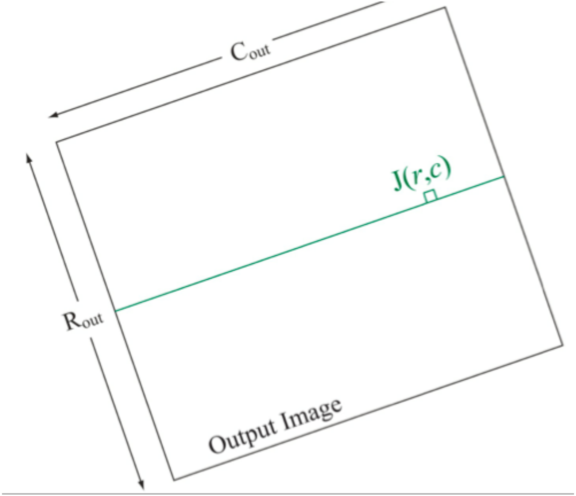
Xoay ảnh

Sau khi xoay ảnh input một góc $-\theta$, $\mathbf{J}(r, c)$ gần như cùng vị trí với $\mathbf{I}(r_f, c_f)$

$$\Phi(r, c, :) = \begin{bmatrix} r_f \\ c_f \end{bmatrix}$$

$$= \mathbf{P}^{-1}(\theta) \begin{bmatrix} r - R_{out0} \\ c - C_{out0} \end{bmatrix}$$

$$+ \begin{bmatrix} R_{in0} \\ C_{in0} \end{bmatrix}$$



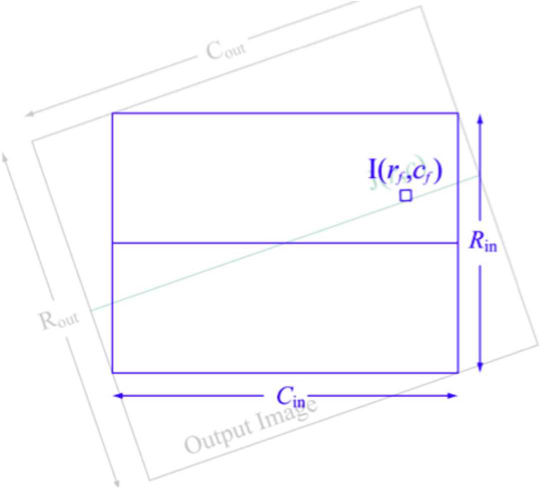
Xoay ảnh

Sau khi xoay ảnh input một góc $-\theta$, $\mathbf{J}(r, c)$ gần như cùng vị trí với $\mathbf{I}(r_f, c_f)$

$$\Phi(r, c, :) = \begin{bmatrix} r_f \\ c_f \end{bmatrix}$$

$$= \mathbf{P}^{-1}(\theta) \begin{bmatrix} r - R_{out0} \\ c - C_{out0} \end{bmatrix}$$

$$+ \begin{bmatrix} R_{in0} \\ C_{in0} \end{bmatrix}$$



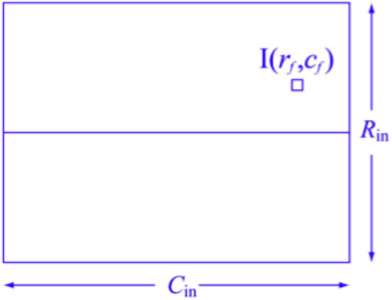
Xoay ảnh

Sau khi xoay ảnh input một góc $-\theta$, $\mathbf{J}(r,c)$ gần như cùng vị trí với $\mathbf{I}(r_f, c_f)$

$$\Phi(r,c,:) = \begin{bmatrix} r_f \\ c_f \end{bmatrix}$$

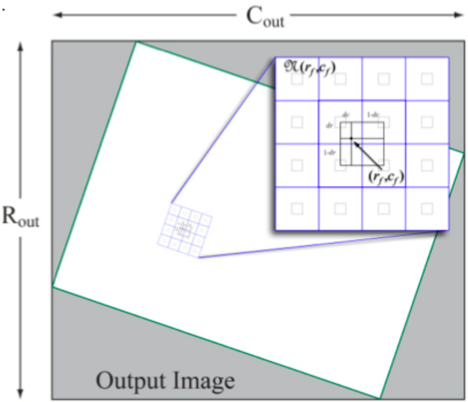
$$= \mathbf{P}^{-1}(\theta) \begin{bmatrix} r - R_{out0} \\ c - C_{out0} \end{bmatrix}$$

$$+ \begin{bmatrix} R_{in0} \\ C_{in0} \end{bmatrix}$$



Xoay ảnh

$\mathbf{J}(r, c) = \Theta\{\mathbf{I}; \mathfrak{N}(r_r, c_f)\}$
Nội suy:
Giá trị pixel output thường là một hàm của các giá trị trong hàng xóm
Nội suy bilinear: sử dụng hàng xóm 2×2; bicubic sử dụng 4×4
Hàng xóm gần nhất:
$\mathbf{J}(r, c) = \mathbf{J}(r_i, c_i)$ với
$(r_i, c_i) = \text{round}(r_f, c_f)$



Quay ảnh với nội suy - ví dụ

Ảnh ban đầu

Hàng xóm gần nhất

Bilinear

Bicubic

Ánh xạ ảnh vào khối cầu?