

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA



BÁO CÁO BÀI TẬP LỚN
MÔN : ĐẠI SỐ TUYẾN TÍNH
ĐỀ TÀI SỐ 11: PHÉP QUAY GIVEN (GIVEN ROTATION)
XỬ LÝ HÌNH ẢNH TRONG KHÔNG GIAN NHIỀU
CHIỀU

Giảng viên hướng dẫn: TS. NGUYỄN HỮU HIỆP
Nhóm báo cáo: 06

Sinh viên thực hiện:

Họ và tên	MSSV	Mã lớp
Lê Tuấn Kiệt	2013569	L14
Lý Đặng Anh Vũ	2313958	L14
Lý Thị Trúc Huỳnh	2311282	L14
Mạch Ngọc Thuận	2313349	L14
Lê Trọng Tín	2313452	L14
Mai Xuân Phúc	2312687	L14
Ngô Đức Thắng	2313184	L14
Ngô Hà Phương	2312744	L14

TP Hồ Chí Minh, 2024

Mục lục

Phần mở đầu	2
0.1 Đối tượng nghiên cứu	2
0.2 Phương pháp nghiên cứu	2
0.3 Mục tiêu nghiên cứu	2
1 Chương 1: Cơ sở lý thuyết phép quay Given	2
1.1 Cơ sở vector quay Given	2
1.2 Phép quay Given trong xử lý ảnh 2 chiều	3
2 Chương 2: Ứng dụng của phép quay Given trong không gian 2 chiều, 3 chiều, n chiều	5
2.1 Ý tưởng ứng dụng của phép quay trong xử lý hình ảnh	5
2.2 Mô tả sơ bộ phép quay trong không gian	6
2.3 Ứng dụng phép quay trong xử lý hình ảnh	7
2.4 Ứng dụng thực tiễn phép quay	7
3 Chương 3: Ứng dụng MATLAB xử lý hình ảnh trong không gian nhiều chiều bằng phép quay Given	9
3.1 Giới thiệu về phần mềm MATLAB	9
3.2 Cơ sở lý thuyết	9
3.2.1 Mở đầu	9
3.2.2 Xoay các pixel quanh tâm ảnh gốc	10
3.2.3 Trả về kết quả	11
3.3 Mã nguồn của chương trình	13
3.4 Giải thích ý nghĩa mã nguồn của chương trình	14
4 Lời cảm ơn	16
Tài liệu tham khảo	17

Phần mở đầu

0.1 Đối tượng nghiên cứu

- Phép quay given
- Chương trình xử lý hình ảnh trong không gian nhiều chiều bằng phép quay Given
- Ứng dụng của phép quay Given.

0.2 Phương pháp nghiên cứu

Tìm hiểu, phân tích lý thuyết của phép quay và thử nghiệm bằng cách viết chương trình MATLAB.

0.3 Mục tiêu nghiên cứu

Có sản phẩm là một phần mềm MATLAB có khả năng xử lý hình ảnh trong không gian n chiều.

1 Chương 1: Cơ sở lý thuyết phép quay Given

1.1 Cơ sở vector quay Given

Kỹ thuật quay Givens là một phương pháp để phân tích ma trận $[A]$ thành tích của ma trận $[Q]$ và ma trận $[R]$ bằng cách làm cho các phần tử lần lượt bằng 0 cho đến khi có được ma trận tam giác phải. Nếu ta có một điểm có tọa độ bất kỳ trong không gian, khi xoay nó theo một không gian con được xác định bởi vector độc lập tuyến tính thì khi đó ta đang thực hiện một phép quay Given. Phép quay Given thực chất là một ma trận quay có dạng:

$$[G_\theta] = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

(Phép quay trong không gian 2 chiều)

Đối với không gian 3 chiều thì ma trận sẽ được viết dưới dạng:

- Trục Ox:

$$X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}$$

- Trục Oy:

$$Y = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

- Trục Oz:

$$Z = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Toạ độ ảnh M' của điểm M qua phép quay Given được tính bằng cách:

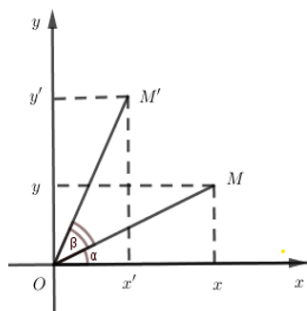
$$M' = [G_\theta] \cdot M$$

Ví dụ: Ảnh của ma trận $M = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ qua phép quay Given quanh Oy một góc θ :

$$M' = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

1.2 Phép quay Given trong xử lí ảnh 2 chiều

Cho điểm $M(x; y)$ là một điểm trong hệ trục Oxy và vectơ \vec{r} (có độ dài là r) hợp với trục Ox một góc α . Điểm M quay quanh gốc O , ngược chiều kim đồng hồ một góc β tạo ra ảnh $M'(x'; y')$.



Ta có:
$$\begin{cases} x = r \cos \alpha \\ y = r \sin \alpha \end{cases}$$

$$\Rightarrow \begin{cases} x' = r \cos(\alpha + \beta) = r \cos \alpha \cos \beta - r \sin \alpha \sin \beta = x \cos \beta - y \sin \beta \\ y' = r \sin(\alpha + \beta) = r \sin \alpha \cos \beta + r \cos \alpha \sin \beta = y \cos \beta + x \sin \beta \end{cases}$$

Ví dụ: Bài tập 2.7.9 trang 67 - Giáo trình Đại số tuyến tính

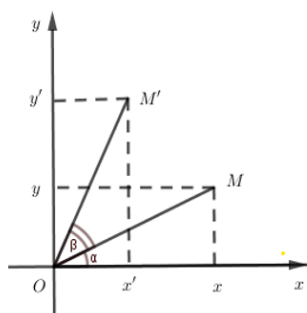
a) Tìm ma trận phép quay trong hệ trục Oxy quay quanh gốc O , ngược chiều kim đồng hồ một góc α .

b) Tìm và biểu diễn ảnh của hình chữ nhật $ABCD$ qua phép quay với góc $\alpha = \frac{\pi}{2}$. Biết rằng $A(1, 1)$, $B(5, 1)$, $C(5, 3)$ và $D(1, 3)$.

Giải

a) Cho $M(x, y)$ là một điểm trong hệ trục tọa độ Oxy . Đặt $r = OM$ và β là góc giữa \overrightarrow{OM} và chiều dương Ox .

$$\text{Ta có: } \begin{cases} x = r \cos \beta \\ y = r \sin \beta \end{cases}$$



Gọi $M_1(x_1, y_1)$ là ảnh của M qua phép quay quanh gốc O ngược chiều kim đồng hồ góc α .

$$\text{Khi đó: } \begin{cases} x_1 = r \cos(\alpha + \beta) = r \cos \alpha \cos \beta - r \sin \alpha \sin \beta = x \cos \alpha - y \sin \alpha \\ y_1 = r \sin(\alpha + \beta) = r \sin \alpha \cos \beta + r \cos \alpha \sin \beta = x \sin \alpha + y \cos \alpha \end{cases}$$

$$\Rightarrow \begin{cases} x_1 = x \cos \alpha - y \sin \alpha \\ y_1 = x \sin \alpha + y \cos \alpha \end{cases}$$

Ở dạng ma trận:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\Leftrightarrow [M_1] = R[M]$$

Ma trận $R = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix}$ được gọi là ma trận của phép quay quanh gốc O trong hệ Oxy ngược chiều kim đồng hồ một góc α .

b) Ma trận biểu diễn hình chữ nhật ABCD:

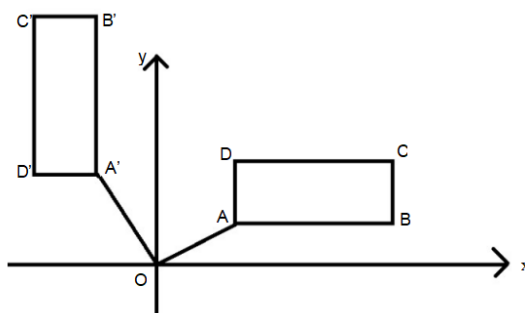
$$M = \begin{bmatrix} 1 & 5 & 5 & 1 \\ 1 & 1 & 3 & 3 \end{bmatrix}$$

Ma trận của phép quay quanh gốc O trong hệ trục tọa độ Oxy với góc quay $\alpha = \frac{\pi}{2}$ là:

$$R = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Ảnh của hình chữ nhật ABCD là:

$$R[M] = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 5 & 5 & 1 \\ 1 & 1 & 3 & 3 \end{bmatrix} = \begin{bmatrix} -1 & -1 & -3 & -3 \\ 1 & 5 & 5 & 1 \end{bmatrix}$$



2 Chương 2: Ứng dụng của phép quay Given trong không gian 2 chiều, 3 chiều, n chiều

2.1 Ý tưởng ứng dụng của phép quay trong xử lý hình ảnh

- Phép quay là một phép biến đổi hình học trong đó một đối tượng được xoay xung quanh một điểm hoặc một trục cố định một góc nhất định.
- Phép quay có thể được mô tả bằng một ma trận quay.

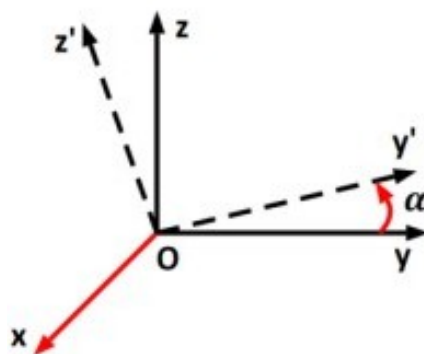
Đặc điểm của phép quay:

- Một phép quay bất kỳ là một sự chuyển động của một không gian nhất định nhưng bảo toàn ít nhất một điểm.
- Chẳng hạn, nó có thể mô tả chuyển động của một vật rắn quanh một điểm cố định.
- Phép quay có thể có dấu (giống với dấu của góc): phép quay theo chiều kim đồng hồ được biểu diễn bởi một số âm còn phép quay ngược chiều kim đồng hồ được biểu diễn bởi một số dương.

2.2 Mô tả sơ bộ phép quay trong không gian

Các Phép Quay Cơ Bản Quanh Các Trục Tọa Độ Ox, Oy, Oz:

- Phép quay quanh trục Ox: Nếu ta giữ nguyên trục Ox và quay mặt phẳng Oyz quanh trục Ox một góc α , ta thu được hệ trục tọa độ 3 chiều mới Ox'y'z' như trong hình:



Ma trận tương ứng với phép quay này là: $R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}$

- Phép quay quanh trục Oy: Tương tự như góc α , nếu ta giữ nguyên trục Oy và quay mặt phẳng Oxz quanh trục Oy một góc β , ta thu được hệ trục tọa độ 3 chiều mới Ox'yz'.

Ma trận tương ứng với phép quay này là: $R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}$

- Phép quay quanh trục Oz: Nếu ta giữ nguyên trục Oz và quay mặt phẳng Oxy quanh trục Oz một góc γ , ta thu được hệ trục tọa độ 3 chiều mới Ox'y'z'.

Ma trận tương ứng với phép quay này là: $R_z(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$

- Phép quay Given n chiều: Ví dụ cho n chiều $x_1, x_2, x_3, \dots, x_n$. Không làm mất tính tổng quát, ta cho x_1 và x_2 thay còn $n - 2$ chiều còn lại x_3, \dots, x_n vẫn giữ nguyên.

Ma trận tương ứng với phép quay này là: $\begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

2.3 Ứng dụng phép quay trong xử lý hình ảnh

Phép quay là một phép biến đổi hình học trong đó có một đối tượng được xoay xung quanh một điểm hoặc một trục cố định một góc nhất định. Phép quay Given có thể được mô tả bằng một ma trận quay, là một phương pháp để phân tích ma trận $[A]$ thành tích của ma trận $[Q]$ và ma trận $[R]$ bằng cách làm cho các phần tử lần lượt bằng 0 cho đến khi có được ma trận tam giác phải. Một phép quay bất kỳ là một sự chuyển động của một không gian nhất định nhưng bảo toàn ít nhất một điểm.

Trong Không gian 2 chiều: Phép quay được sử dụng để xoay các hình ảnh, đối tượng hoặc vật thể trong không gian 2D, tạo ra các hiệu ứng chuyển động, tạo ra các góc nhìn khác nhau cho đối tượng. Được sử dụng để thực hiện các phép biến đổi hình học trên các hình ảnh hoặc khung hình, chỉnh sửa hình ảnh, xoay hoặc cắt video, thực hiện các hiệu ứng đặc biệt.

Trong không gian 3 chiều: Phép quay được sử dụng để mô phỏng và định vị vị trí của các đối tượng trong môi trường 3D. Thực hiện các phép biến đổi trên các dữ liệu hình ảnh và 3D như xoay và cắt khung hình.

Trong không gian n chiều: Phép quay có thể được sử dụng để xoay một đối tượng n chiều, hoặc để thay đổi góc nhìn của một camera n chiều. Phép quay cũng có thể được sử dụng để tạo hiệu ứng xoay n chiều, hoặc để thực hiện phép biến đổi trên ảnh n chiều.

Thực hiện ứng dụng phép quay trong xử lý hình ảnh:

- Xây dựng công thức chuyển đổi quan sát
- Xây dựng công cụ 3D
- Áp dụng bộ công cụ 3D để mô phỏng hình lập phương đơn vị
- Cho phép nhìn vật thể từ các góc độ khác nhau: từ phía trước, từ phía sau, từ trên xuống, từ dưới lên, ...

2.4 Ứng dụng thực tiễn phép quay

Một sản phẩm là một phần mềm có khả năng xử lý hình ảnh trong không gian n chiều có thể là một ứng dụng hữu ích trong nhiều lĩnh vực, ví dụ như trong khoa học dữ liệu: phần mềm có thể được sử dụng để phân tích, khai thác và trực quan hóa dữ liệu n chiều như dữ liệu học thời gian, dữ liệu không gian, dữ liệu mạng, dữ liệu đa phương tiện, y học, v.v... Với những ứng dụng thực tế này, phép quay trong không gian đóng vai trò quan trọng trong các ngành công nghiệp và khoa học, góp phần đưa ra các giải pháp và thiết kế hiệu quả.

1. **Trong Đồ họa máy tính và Thiết kế đồ họa:** Phép quay được sử dụng để biến đổi vị trí các đối tượng trong không gian 3D, xoay các hình ảnh, đối tượng hoặc vật thể trong không gian 2D, tạo ra các hiệu ứng

chuyển động, tạo ra các góc nhìn khác nhau cho đối tượng. Được sử dụng để thực hiện các phép biến đổi hình học trên các hình ảnh hoặc khung hình, chỉnh sửa hình ảnh, xoay hoặc cắt video, giúp tạo ra, hiển thị, chỉnh sửa hiệu ứng chuyển động, thao tác với các hình ảnh n chiều, ví dụ như ảnh 3D, ảnh 4D, ảnh VR, ảnh AR, ảnh hologram,... Trong thiết kế đồ họa, phép quay được sử dụng để tạo ra các tác phẩm nghệ thuật có tính đối xứng, hài hòa và thẩm mỹ..

2. **Trong Robot học và Điều khiển robot:** Phép quay được sử dụng để điều khiển vị trí và hướng đi của robot trong không gian 3D, nó có thể được áp dụng để xoay các bộ phận của robot, như cánh tay hay chân, giúp robot thực hiện các nhiệm vụ cụ thể như di chuyển, quét môi trường.
3. **Trong Hàng không vũ trụ:** Phép quay được sử dụng để điều khiển vị trí và hướng đi của tàu vũ trụ, thiên thạch, và thiết bị khác trong không gian, đồng thời giúp ổn định và điều chỉnh hành trình di chuyển.
4. **Trong Công nghiệp:** Phép quay được áp dụng trong việc thiết kế các bộ phận và điều khiển các máy móc, thiết bị có khả năng xoay quanh một trục cố định hoặc một điểm cố định như bánh răng, tuốc bin,... và các quy trình sản xuất để tạo ra các sản phẩm chính xác và đáng tin cậy. Trong khoa học máy: phần mềm có thể được sử dụng để huấn luyện, kiểm tra và tối ưu hóa các mô hình học máy n chiều, ví dụ như mạng nơ-ron nhân tạo, máy vector hỗ trợ, phân cụm, phân loại, hồi quy,...
5. **Trong Y học:** Trong lĩnh vực y tế, phép quay có thể được sử dụng để mô phỏng và điều tra các cấu trúc và chuyển động trong cơ thể con người, ví dụ: máy chụp cắt lớp vi tính (CT) sử dụng phép quay để quét xuyên qua cơ thể người bằng tia X và tạo ra các hình ảnh ba chiều của các cơ quan, giúp bác sĩ có thể chẩn đoán, điều trị và theo dõi các bệnh lý của các cơ quan, cũng có thể được sử dụng để hỗ trợ trong quá trình phẫu thuật và điều trị. Trong thị giác máy tính: phần mềm có thể được sử dụng để xử lý, nhận dạng các hình ảnh n chiều như ảnh chụp X-quang, ảnh MRI, ảnh chụp CT, ảnh độ sâu, ảnh nhiệt,...

Ngoài ra, phép quay Given vẫn còn rất nhiều ứng dụng hữu ích khác trong các lĩnh vực: thiết kế cơ khí và xây dựng, thiết kế sản phẩm và mô hình hóa 3D, mô phỏng thiết kế môi trường, thực tế ảo và thực tế tăng cường, trí tuệ nhân tạo, trò chơi điện tử và các ứng dụng giải trí,..

3 Chương 3: Ứng dụng MATLAB xử lý hình ảnh trong không gian nhiều chiều bằng phép quay Given

3.1 Giới thiệu về phần mềm MATLAB

MATLAB – Matrix Laboratory là một môi trường tính toán số và ngôn ngữ lập trình mạnh mẽ, phần ảnh mục tiêu ban đầu của nó là cung cấp một công cụ tương tác cho các tính toán ma trận. Các tính năng nổi bật: Tính toán ma trận, visualization, toolboxes, giao diện dễ sử dụng, hỗ trợ lập trình.

Ứng dụng của MATLAB trong bài tập lớn trong xử lý hình ảnh trong không gian nhiều chiều:

- Xoay ảnh n chiều
- Ma trận xoay n chiều: Trong không gian n chiều, ma trận xoay có kích thước $n \times n$ và được sử dụng để xoay các vector trong không gian đó. MATLAB cho phép bạn tạo và áp dụng các ma trận này.

1. Xử lý dữ liệu n chiều: MATLAB hỗ trợ xử lý các tập dữ liệu n chiều dưới dạng mảng n chiều.

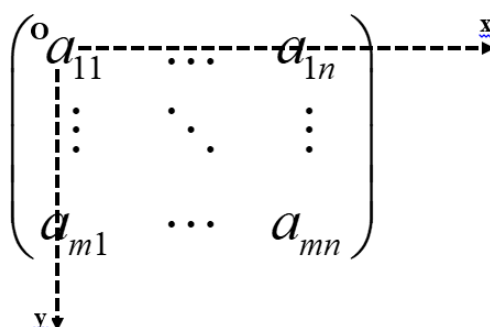
Ngoài ra còn có những chức năng khác như:

2. Trực quan hóa ảnh n chiều Công cụ trực quan hóa: MATLAB cung cấp các công cụ để trực quan hóa dữ liệu trong không gian cao hơn 3 chiều. Bạn có thể sử dụng các hàm như slice, volshow, và scatter3 để hiển thị các lát cắt hoặc chiếu các dữ liệu n chiều xuống không gian 2D hoặc 3D.

3. Giao diện người dùng tương tác Xây dựng GUI: MATLAB cho phép xây dựng giao diện người dùng để nhập các tham số xoay và hiển thị kết quả một cách trực quan. Bạn có thể sử dụng uifigure, uibutton, uislider, và uiaxes để tạo giao diện.

3.2 Cơ sở lý thuyết

3.2.1 Mở đầu

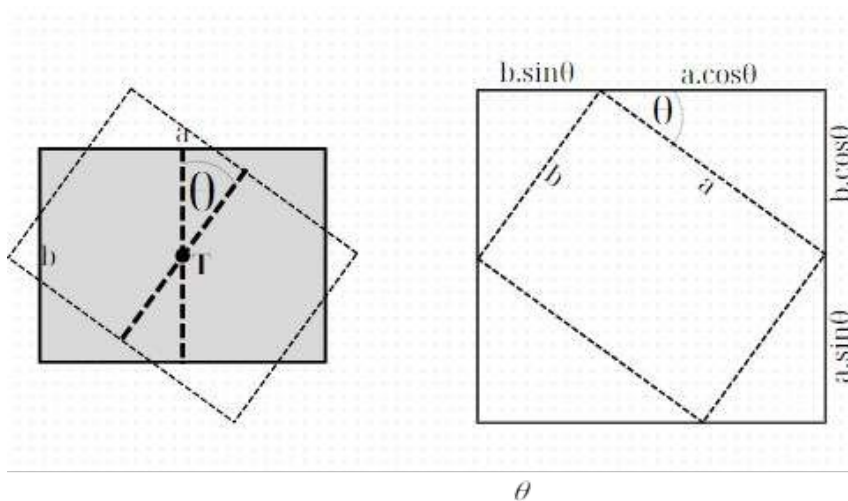
$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$


Mọi tính toán trong hệ toạ độ tưởng tượng này (dời hình, xoay hình, ...) giống như trong mặt phẳng Oxy bình thường, chỉ khác: phép xoay với ma trận given làm ảnh quay cùng chiều kim đồng hồ (do trục toạ độ bị lật xuống) và toạ độ mỗi điểm (x, y) thoả điều kiện sau:

$$(x, y) \in N^*, \quad (x, y) \leq (n, m) \quad (1)$$

Vì chỉ số các phần tử luôn là số tự nhiên khác không và chúng không được vượt quá số pixel của ảnh gốc.

Sau khi xoay ảnh, kích thước ma trận ban đầu thay đổi nên trước đó ta cần tạo ra một ma trận 0 để chứa các điểm ảnh mới (tạm gọi là “khung”).



Giả sử cần xoay hình chữ nhật axb một góc θ cùng chiều kim đồng hồ. Nếu “khung” là $a'xb'$, ta có:

$$\begin{cases} a' = a|\cos \theta| + b|\sin \theta| \\ b' = a|\sin \theta| + b|\cos \theta| \end{cases} \Leftrightarrow \begin{pmatrix} a' \\ b' \end{pmatrix} = \begin{pmatrix} |\cos \theta| & |\sin \theta| \\ |\sin \theta| & |\cos \theta| \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix}$$

Dấu “trị tuyệt đối” để biểu thức luôn đúng cho cả $\theta > 90^\circ$. Ngoài ra cần làm tròn a' và b' để thoả (1).

3.2.2 Xoay các pixel quanh tâm ảnh gốc

Tọa độ tâm T trong hệ Oxy:

$$\begin{cases} (X, Y) = \left(\frac{a}{2}, \frac{b}{2}\right) \text{ nếu } (x, y) \text{ chia hết cho } 2 \\ (X, Y) = \left(\frac{a+1}{2}, \frac{b+1}{2}\right) \text{ nếu } (x, y) \text{ không chia hết cho } 2 \end{cases}$$

Tọa độ x_T, y_T của mỗi pixel so với tâm T này:

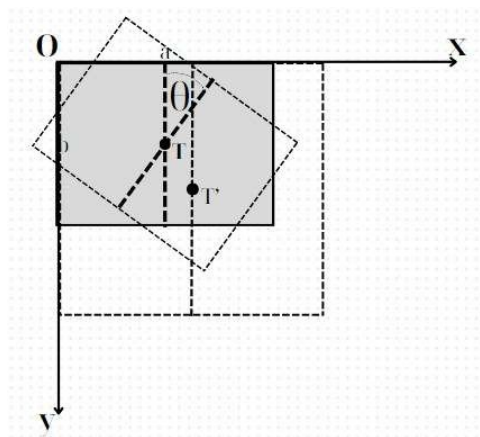
$$\begin{cases} x_T = x - X \\ y_T = y - Y \end{cases}$$

Với (x,y) là tọa độ trong hệ Oxy ban đầu

Sau khi xoay góc θ , tọa độ mới là (x'_T, y'_T) so với tâm T:

$$\begin{pmatrix} x'_T \\ y'_T \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x_T \\ y_T \end{pmatrix}$$

Cuối cùng, đưa về tọa độ trong hệ Oxy:



Gọi T' là tâm "khung" trong hệ Oxy , tính toán tương tự như tâm T chỉ thay a, b thành a', b' . Trước hết cần đưa (x'_T, y'_T) vào trong khung bằng phép toán $(x'_T, y'_T) + T' - T$ ($T' - T$ là khoảng cách đại số giữa T' và T). Sau đó đổi về tọa độ trong hệ Oxy bằng cách cộng thêm T vào. Vậy tọa độ (x', y') có được qua phép xoay là:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x'_T \\ y'_T \end{pmatrix} + T' - T + T = \begin{pmatrix} x'_T \\ y'_T \end{pmatrix} + T'$$

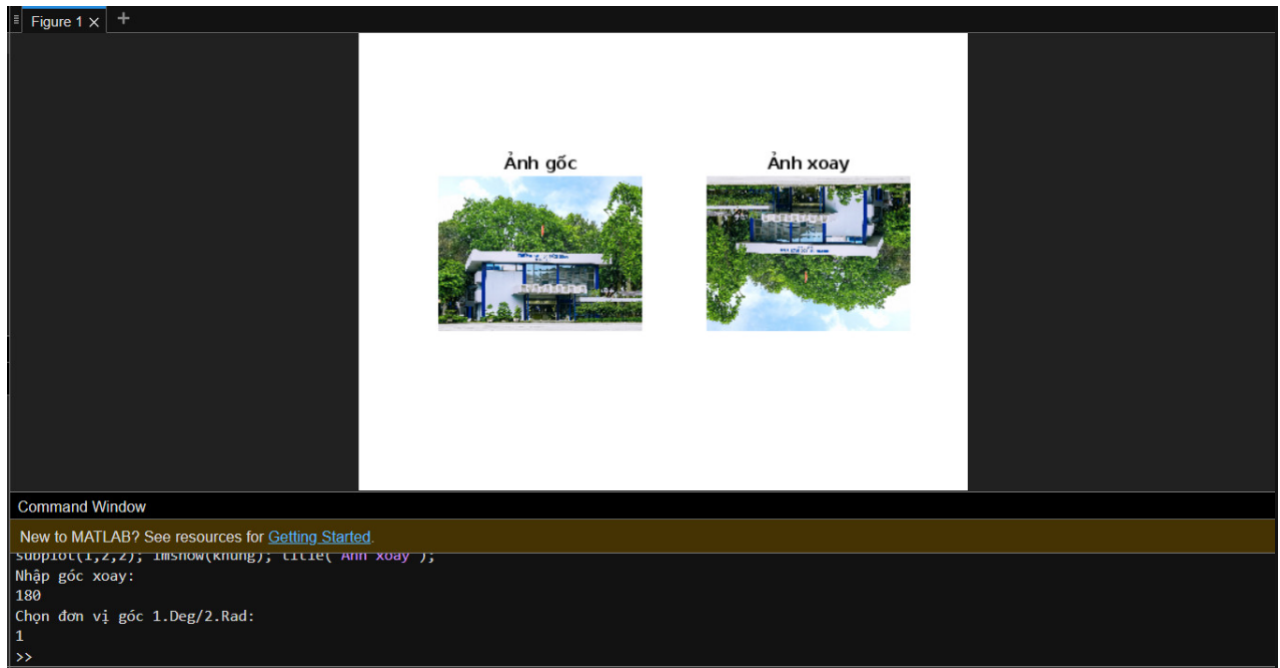
Lưu ý: (x', y') phải được làm tròn để thỏa (1)

3.2.3 Trả về kết quả

Gán dữ liệu pixel ban đầu vào các vị trí (x', y') rồi dùng lệnh `imread()` ta được kết quả:

Nhập góc xoay:
180
Chọn đơn vị góc 1.Deg/2.Rad:
1S

Hình 1: Yêu cầu người dùng chọn góc xoay và đơn vị góc



Hình 2: Kết quả hiện ra trên màn hình



Hình 3: Ảnh trước và sau khi xoay 1 góc 180°

3.3 Mã nguồn của chương trình

Dưới đây là phần mã nguồn của chương trình:

```
% Chọn tệp hình ảnh
file = uigetfile('*.jpg;*.png;*.jpeg;*.gif','Chọn_anh_de_xoay');
while file == 0
    file = uigetfile('*.jpg;*.png;*.jpeg;*.gif','Ban_van_chua_chon_anh');
end
% Đọc hình ảnh
anh = imread(file);
% Nhập góc xoay và đơn vị góc
angle = input('Nhập_góc_xoay: ');
donvigoc = input('Chọn_don_vi_góc 1.Deg/2.Rad: ');
while donvigoc ~= 1 && donvigoc ~= 2
    donvigoc = input('Chọn_don_vi_góc 1.Deg/2.Rad: ');
end
% Chuyển đổi góc sang radian nếu cần
if donvigoc == 1
    theta = deg2rad(angle);
else
    theta = angle;
end
% Kích thước ảnh ban đầu
[d0, n0, ~] = size(anh);
% Toa do tam ban dau
b = ceil(d0 / 2);
a = ceil(n0 / 2);
O = [a; b];
% Kích thước ảnh sau khi xoay
B = [abs(cos(theta)) abs(sin(theta)); abs(sin(theta)) abs(cos(theta))];
S = round(B * [n0; d0]);
% Toa do tam anh sau khi xoay
as = ceil(S(1,1) / 2);
bs = ceil(S(2,1) / 2);
Os = [as; bs];
% Tạo khung ảnh mới
khung = zeros(S(2,1), S(1,1), size(anh, 3), 'uint8');
% Ma tran phép quay
G = [cos(theta) -sin(theta); sin(theta) cos(theta)];
% Xoay tung pixel của ảnh
for d = 1:d0
    for n = 1:n0
        % Toa do moi pixel so voi O
        X0 = [n; d] - O;
        % Toa do sau khi xoay
        X = round(G * X0) + Os;
```

```

        % Gan gia tri vao khung anh moi
        if X(1,1) > 0 && X(1,1) <= S(1,1) && X(2,1) > 0 &&
            X(2,1) <= S(2,1)
            khung(X(2,1), X(1,1),: ) = anh(d, n, :);
        end
    end
end
% Hien thi anh goc va anh da xoay
subplot(1,2,1); imshow(anh); title('Anh_goc');
subplot(1,2,2); imshow(khung); title('Anh_xoay');

```

3.4 Giải thích ý nghĩa mã nguồn của chương trình

Trong chương trình này, chúng ta sẽ thực hiện xoay ảnh trong không gian 2D để có thể dễ dàng hình dung được quá trình thuật toán được thực hiện và nhận diện kết quả.

Đoạn mã MATLAB dưới đây được sử dụng để xoay một hình ảnh 2D theo một góc nhập từ người dùng. Hãy cùng phân tích từng phần của đoạn mã để hiểu rõ cách hoạt động của nó:

Trong chương trình này, chúng ta sẽ thực hiện xoay ảnh trong không gian 2D để có thể dễ dàng hình dung được quá trình thuật toán được thực hiện và nhận diện kết quả.

Đoạn mã MATLAB dưới đây được sử dụng để xoay một hình ảnh 2D theo một góc nhập từ người dùng. Hãy cùng phân tích từng phần của đoạn mã để hiểu rõ cách hoạt động của nó:

- **uigetfile**: Hiện thị hộp thoại để người dùng chọn một tệp hình ảnh có định dạng .jpg, .png, .jpeg, hoặc .gif.
- **while file == 0**: Kiểm tra nếu người dùng chưa chọn tệp nào (**file == 0**), hộp thoại sẽ tiếp tục hiện lên cho đến khi người dùng chọn một tệp hợp lệ.
- **imread**: Đọc tệp hình ảnh được chọn vào biến **anh**.
- **input**: Nhập góc xoay và đơn vị góc từ người dùng.
- **while donvigoc ~= 1 && donvigoc ~= 2**: Kiểm tra nếu đơn vị góc không phải là 1 (độ) hoặc 2 (radian), yêu cầu người dùng nhập lại cho đến khi giá trị hợp lệ.
- **deg2rad**: Nếu đơn vị góc là độ, chuyển đổi góc từ độ sang radian. Nếu đơn vị góc đã là radian, giữ nguyên giá trị.
- **size**: Lấy kích thước của ảnh. **d0** là chiều cao và **n0** là chiều rộng của ảnh.
- **ceil**: Lấy tọa độ tâm của ảnh ban đầu bằng cách chia chiều cao và chiều rộng cho 2 và làm tròn lên. 0 là tọa độ tâm của ảnh.

- Tính toán kích thước của ảnh sau khi xoay bằng cách sử dụng ma trận quay. B là ma trận chứa các giá trị tuyệt đối của \cos và \sin của góc xoay để tính kích thước mới.
- **round**: Làm tròn giá trị kích thước mới S .
- Tính tọa độ tâm $O_s = [a_s; b_s]$ của ảnh mới sau khi xoay.
- **zeros**: Tạo một khung ảnh mới với kích thước đã tính toán và các giá trị ban đầu bằng 0.
- **size(anh, 3)**: Giữ nguyên số lượng kênh màu.
- Tạo ma trận quay 2D dựa trên góc đã chuyển đổi θ .
- **for**: Duyệt qua từng pixel của ảnh gốc. Tọa độ X_0 là tọa độ của pixel so với tâm ban đầu 0.
- Tính tọa độ mới X của pixel sau khi xoay bằng cách nhân ma trận quay G với tọa độ ban đầu X_0 và dịch chuyển đến tọa độ tâm mới O_s .
- **round**: Làm tròn tọa độ mới.
- **if**: Kiểm tra nếu tọa độ mới nằm trong giới hạn của khung ảnh mới, gán giá trị của pixel đó vào khung ảnh mới.
- **subplot**: Tạo cửa sổ đồ họa với hai ô để hiển thị hai ảnh cạnh nhau.
- **imshow**: Hiển thị ảnh gốc và ảnh đã xoay.
- **title**: Đặt tiêu đề cho các ảnh hiển thị.

4 Lời cảm ơn

Để kết lại Báo cáo Bài tập lớn lần này, chúng em xin gửi lời cảm ơn chân thành nhất đến giảng viên hướng dẫn – Thầy Nguyễn Hữu Hiệp. Thầy đã để lại ấn tượng sâu sắc trong lòng chúng em bằng phương pháp giảng dạy riêng của mình, luôn luôn mang lại cho lớp một không khí thoải mái khi học tập. Nhóm chúng em xin chân thành cảm ơn Thầy vì đã tận tình truyền đạt kiến thức quý báu cho chúng em cũng như giải đáp những thắc mắc của chúng em trong những buổi học vừa qua.

Bài báo cáo lần này, nhóm chúng em đã rất nghiêm túc tìm hiểu đề tài, đầu tư thời gian để . Tuy nhiên, do vốn kiến thức của chúng em vẫn còn nhiều hạn chế nên khó tránh khỏi còn tồn tại những thiếu sót. Vì thế, nhóm chúng em rất mong nhận được sự góp ý từ thầy để chúng có thể cải thiện cũng như giúp cho bài cáo của chúng em thêm phần hoàn thiện.

Chúng em xin chân thành cảm ơn!

Tài liệu

- [1] Đặng Văn Vinh. *Giáo trình Đại số tuyến tính*. Đại học Quốc gia TP. Hồ Chí Minh, 2020.
- [2] Carl D. Meyer. *Matrix Analysis and Applied Linear Algebra*.
- [3] David C. Lay. *Linear Algebra and Its Applications*.