

**MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
SUMY STATE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE
BACHELOR PROJECT**



On the topic: Computers Simulation of System Recognition of Affine relation between convex contour images.

**Head of Department
Student group: IN-65AH
Supervisor:**

**Dovbysh A.S.
Felista. G.P
Avramenko Victor Vasylovych**

SUMY 2020

**MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
SUMY STATE UNIVERSITY**

FACULTY OF ELECTRONICS AND COMPUTER SCIENCE

Approved _____
Head of department Dovbysh A.S.
“ ____ ” _____ 2020

Task to bachelor project

Student of fourth course, group IN- 65AH specialty "Informatics" Felista Gasper Paradiso.

Topic: Computers Simulation of System Recognition of Affine relation between convex contour images.

Approved by order of the Sumy State University
№ _____ of _____ 2020

Date of Task
Bachelor work supervisor
Received task to be performed

“ ____ ” _____ 2020.
Avramenko Victor Vasylovych
Felista.G.P

ABSTRACT

Note: 30 pages, 14 figures, 2 tables, 19 sources literature, 1 app.

Object of study- *is an affine relation between convex contour images.*

Purpose of the work- *is to develop an algorithm and a computer program for computer modeling of operational recognition of the affine connection between two convex contour images.*

Research methods-*is for solving the problem is based on the use of disproportionality functions.*

Results-*an algorithm has been developed and a computer program has been created for modeling the operation of the system for detecting affine dependence between two convex planar contours.*

Keywords: n-order derivative disproportionality function, n-order value disproportionality function, integral disproportionality function, affine relation, contour images, recognition of affine relation.

CONTENT

- INTRODUCTION.....4
- 1 LITERATURE REVIEW.....5
 - 1.1 RECOGNITION OF IMAGES.....5
 - 1.1.1 How the image is recognized.....5
 - 1.1.2 Importance of image recognition.....7
 - 1.2. AFFINE TRANSFORMATION.....8
 - 1.2.1 The relevance of recognizing.....11
 - 1.2.2 Normalization of the image13
 - 1.3. DISPROPORTIONALITY FUNCTION14
 - 1.3.1 Research issue statement.....15
 - 1.3.2 Solution of the issue.....16
 - 1.3.3 N-order derivative disproportionality.....17
 - 1.3.4 Disproportionality for parametric form of functions.....17
 - 1.3.5 Value disproportionality functions18
 - 1.3.6 Applications of disproportionality functions.....19
 - 1.4 COMPUTER MODELING.....19
 - 1.4.1 How it works.....19
 - 1.4.2 The pros of computer modeling.....18
 - 1.4.3 The cons of computer modeling.....19
- 2. SOLVING PROBLEM.....19
 - 2.1 Problem statement.....20
 - 2.2 Mathematical statement of the problem.....20
 - 2.3 Choice of method for solving the problem.....21
 - 2.4 Process of solving the problem.....21
 - 2.5 Algorithm for solving the problem.....21
- 3 INSTRUCTIONS FOR USER.....24
- 4. CONTROL TESTING.....24
- CONCLUSION.....26
- REFERENCES27
- APPENDIX.....28

Introduction

Image recognition is an important practical task. In many cases, it comes down to recognition of contour images. One of the common recognition methods is to compare the contour of the analyzed object with a reference contour. However, in robotics, as well as in the processing of images obtained as a result of aerial photography, often observed objects differ from the reference by angle of rotation, scale and displacement. That is, the recognizable image is the result of an affine conversion of the standard one. One of the effective methods for recognizing objects distorted as a result of affine transformation is the normalization method [1]. In this paper, we solve the problem of operational recognition of the presence of an affine connection between two contour images using disproportionality functions.

Disproportionality functions are a measure of the discrepancy between two numerical functions. They allow at a single point to quantify how much one function is unlike the other and this estimate does not depend on the scale of their representation. In my project I am going to use the function of disproportionality to recognize affine relation between contour images.

Using computer simulation, the operability of the proposed method is shown.

Computers simulation of system recognition of affine relation between contour images.

1. LITERATURE REVIEW

1.1 Recognition of images

Image recognition is the process of recognizing image as a particular object using pixel. This applies to the phrase “what you see is what you get” which is commonly abbreviated as WYSWYG, since human brains can easily recognize images. Example for a human being it’s not hard to recognize a part of a rat or chicken but it’s quite difficult for a computer to emulate. Let’s say I want to train a model to be able to detect one class from another and I have several images, I can do all this and solve the problem by using machine learning. Machine learning is like a solution to the problem. We use image classification in order to label objects. Image detection is used to detect objects. The combination of image detection and image classification plus recognition is where you get image recognition [2]. One of the examples of image recognition is optical character recognition or optical character reader (OCR) which is used to convert images from handwritten or typed form to machine-encoded text. It can be used in business cards, bank statements, passport documents etc. The figure(A) below shows a clear picture of how image recognition looks like.

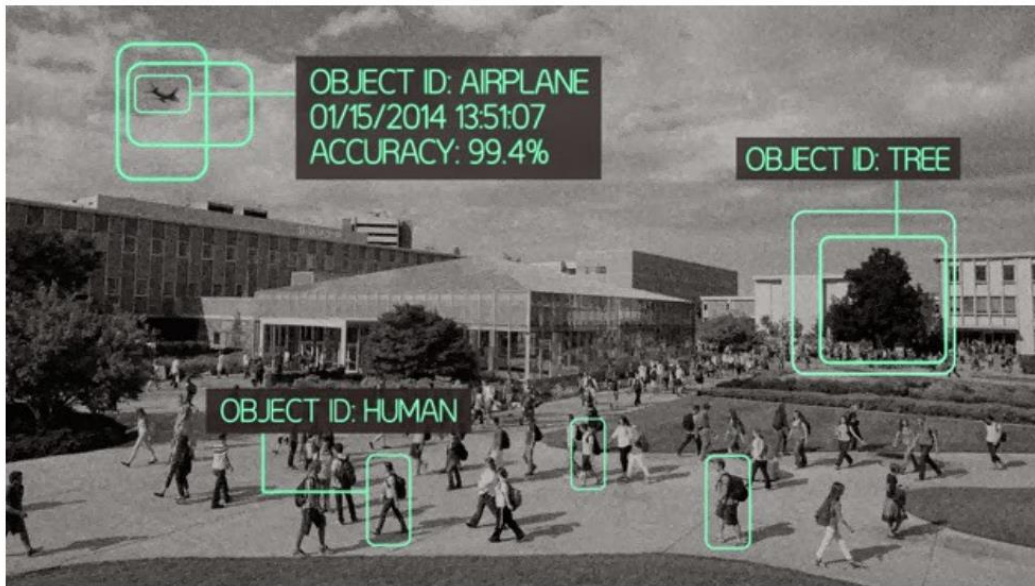
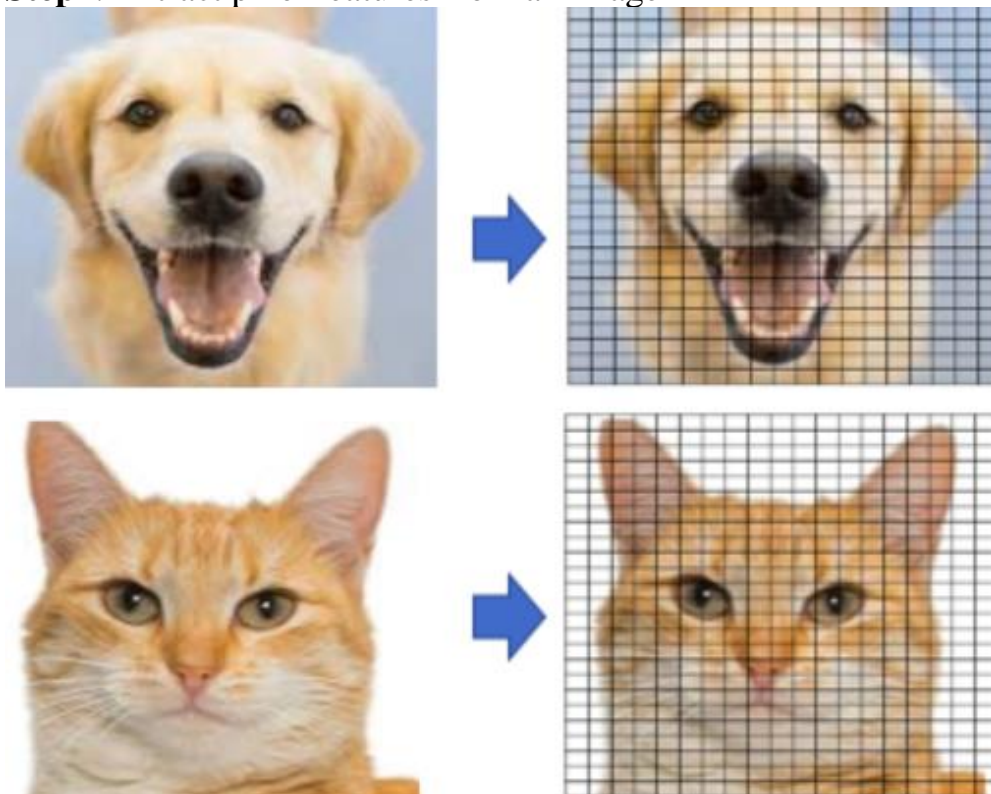


Figure (A)

1.1.1 How the image is recognized

This how we really instruct a computer to differentiate images from one another. This process is similarly to machine learning modeling. The following process are one step on how it works.

Step1: Extract pixel features from an image











































Figure(B)

Features are extracted from the image. Images are made of “pixel” as shown in the figure above. Each pixel is represented by a number(numbers). In which these numbers are called color depth which is the maximum number of potential colors that can be used in an image.

This is when the RGB comes which combines red, green and blue colors in a pixel, and each color has a range from 0 to 255.The value of the RGB is (102,255,102) respectively [3].

Image resolution=1024 ×768 hence, the mega pixel is 0.78

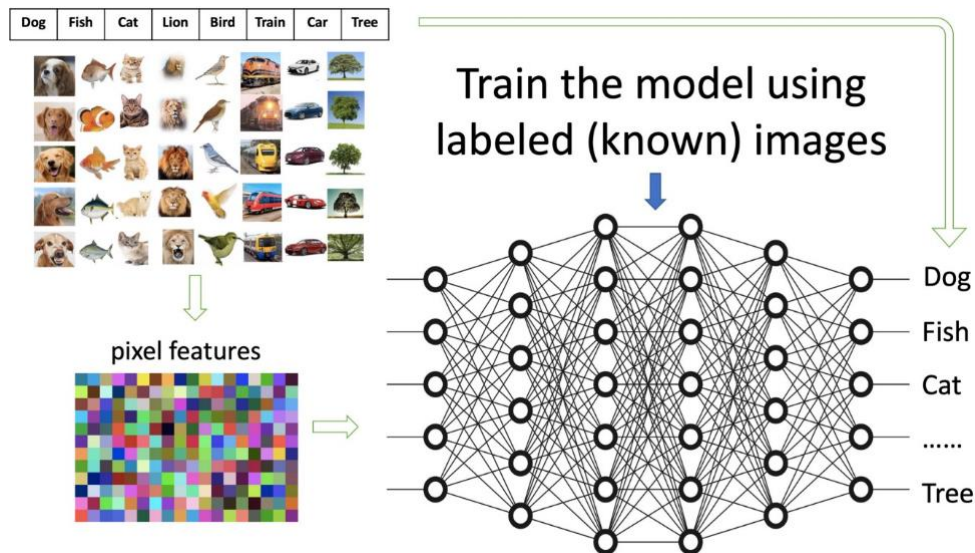
Step 2: Prepare labeled images to train the model

Dog	Fish	Cat	Lion	Bird	Train	Car	Tree
							
							
							
							
							

Figure(C)

In this process we give names to all the images we want to be recognized. The more image we use the better we train the model to be able to differentiate the image. We call this supervised machine learning.

Step 3: Train the model to be able to categorize images

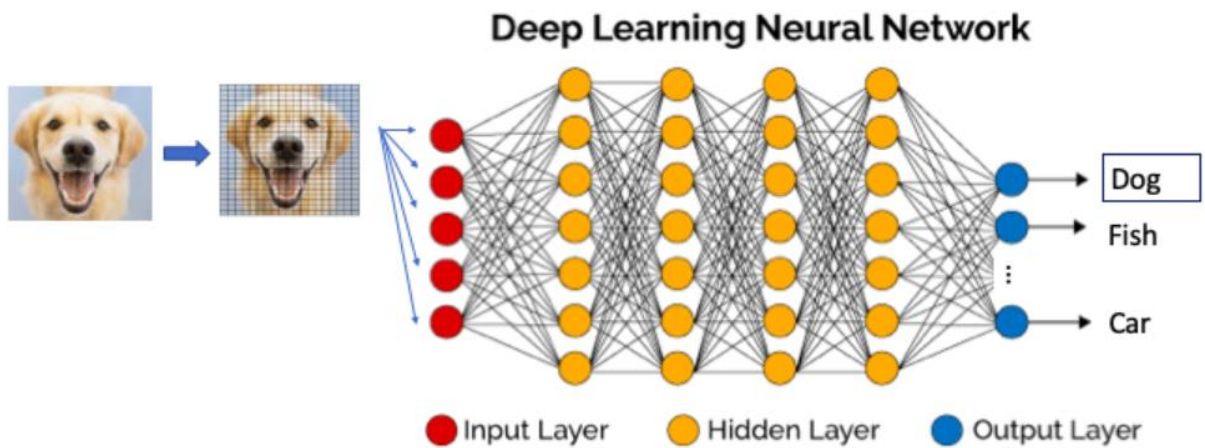


Figure(D)

The above figures show how we train the model with labeled images. The middle of the network is huge and is known as a giant filter. The real concept here is to be able to train the networks in such a way that when the image comes from the left input it will be able to peer with the label output to the right.

Step 4: Recognize a new image to be one of the categories.

After we have trained our model it can now be able to recognize an unknown image. As seen in the figure below that the model was able to recognize the image of a dog.



Figure(E)

1.1.2 Importance of image recognition

- It is very important when it comes to security system i.e. facial recognition.
- It is important in identifying images or objects instantly.
- It can also be used in hospitals to detect cancer through X-ray images.
- It can be used to track moving images.
- It can be used in gesture recognition and visual geolocation.

- It can be used in surveillance for close observation especially in criminal cases.
- It can also be used in E-commerce industry where by we use mobile applications to identify specific products

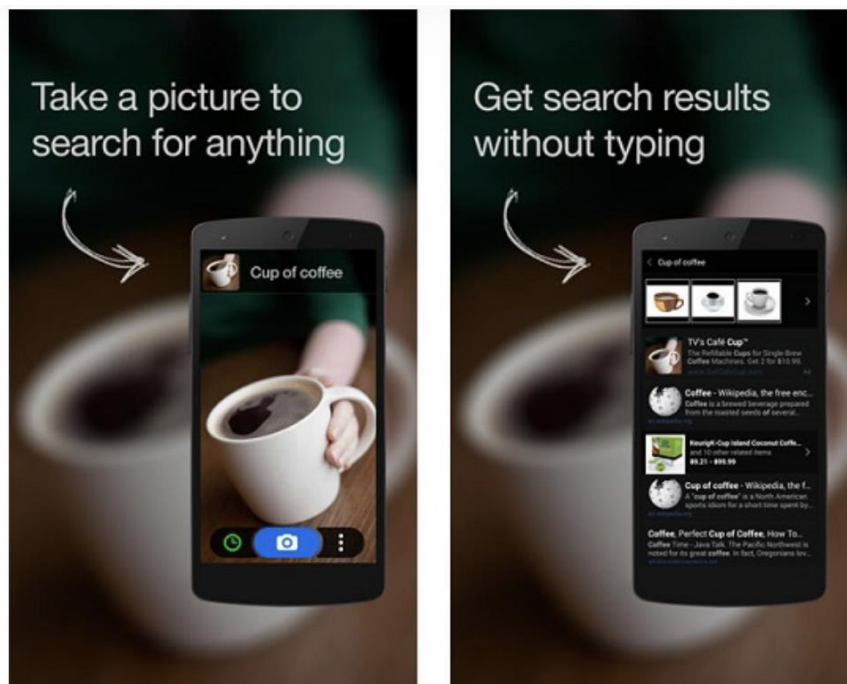


Figure (F)

For example, in the figure above a CamFind API is used. CamFind is normally used to identify objects like bags, cups, belts, glasses etcetera. and give the result of the purchase to the user. Developers can use the API to create their own phone electronic commerce application. Thus, this enables online shopping.

As the above importance of image recognition, it is used in facial recognition and this in how it applies by drawing pixels around the eye area and so on [4].



Figure(G)

Image recognition tools are used to analyze and recognize images. They are cheap and save a lot of time. They can figure out countless images and restore data which is especially applicable to your business in a fast way [5].

Examples of tools that can be used in image recognition.

- Price comparison this uses google shopper to find nearby stores and buy what you are interested.
- Drive less cars are used to recognize vehicles, road signs and pedestrians using a computer, reading road signs and traffic lights.
- Image search in google used to search for various type of images.
- Shazam of the television using any video or image from your phone to be able to find the image or video you want.

1.2 Affine transformation

The affine transformation is any transformation that maintain collinearity that is to say a set of parallel lines remains parallel even after affine transformation and the midpoint of a line remain the same after transformation [6].

Methods of affine transformation are such as rotations, translations, reflections, dilations, shears etc. For example, a triangle is affine since it can be transformed into any other using affine transformation.

The following figures show affine transformation

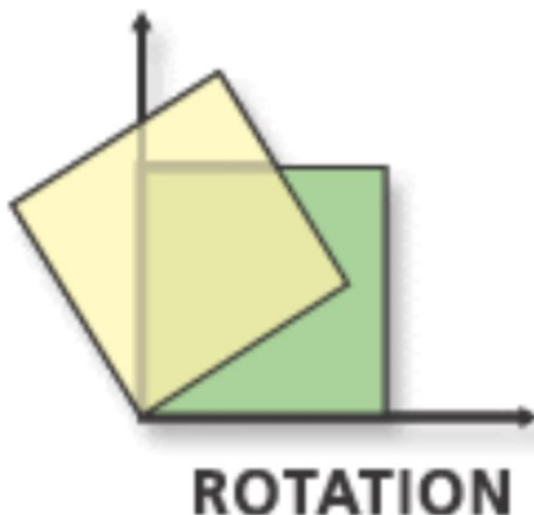


Figure (H)

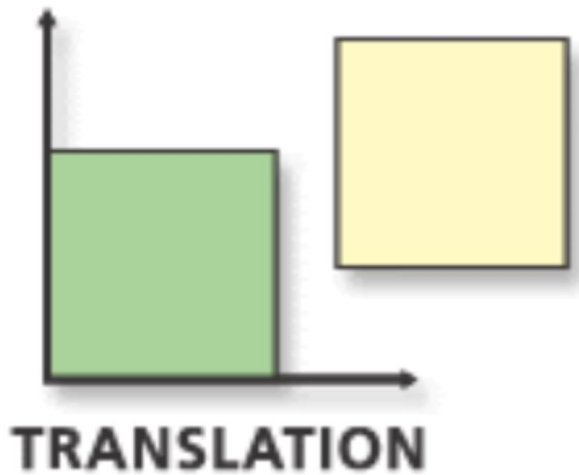


Figure (I)

An example combining rotation and expansion is the rotation-enlargement transformation[7].

$$\begin{aligned} \begin{bmatrix} x \\ y \end{bmatrix} &= s \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} \\ &= s \begin{bmatrix} \cos \alpha (x - x_0) + \sin \alpha (y - y_0) \\ -\sin \alpha (x - x_0) + \cos \alpha (y - y_0) \end{bmatrix}. \end{aligned}$$

Separating the equations, this can be also written as:

$$\begin{aligned} x' &= (s \cos \alpha) x + (s \sin \alpha) y - s (x_0 \cos \alpha + y_0 \sin \alpha) \\ y' &= (-s \sin \alpha) x + (s \cos \alpha) y + s (x_0 \sin \alpha + y_0 \cos \alpha) \end{aligned}$$

$$\begin{aligned} x' &= ax - by + c \\ y' &= bx + ay + d, \end{aligned}$$

where

$$\begin{aligned} a &= s \cos \alpha \\ b &= -s \sin \alpha \end{aligned}$$

The scale factor s is then defined by

$$S = \sqrt{a^2 + b^2},$$

and the rotation angle by

$$\alpha = \tan^{-1}\left(-\frac{b}{a}\right).$$

An affine transformation of \mathbb{R}^n is a map $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ of the form

$$F(\mathbf{p}) = \mathbf{A}\mathbf{p} + \mathbf{q}$$

for all $\mathbf{p} \in \mathbb{R}^n$, where \mathbf{A} is a linear transformation of \mathbb{R}^n . If $\det(\mathbf{A}) > 0$, the transformation is orientation-preserving; if $\det(\mathbf{A}) < 0$, it is orientation-reversing [8].



Figure(J)

Using the affine transformation is going to be able to identify different images and objects in different angles.

1.2.1 The relevance of recognizing the presence of an affine transformation

- It's important to recognize affine transformation because it can be used to represent the distortion in an image.
- It is also commonly useful in satellite imaging.
- It is used to remove errors for geometric deformations which occur with camera angles.
- It helps to rearrange pixels on the image plane.
- Affine transformation also helps to maintain the form of a vector space.

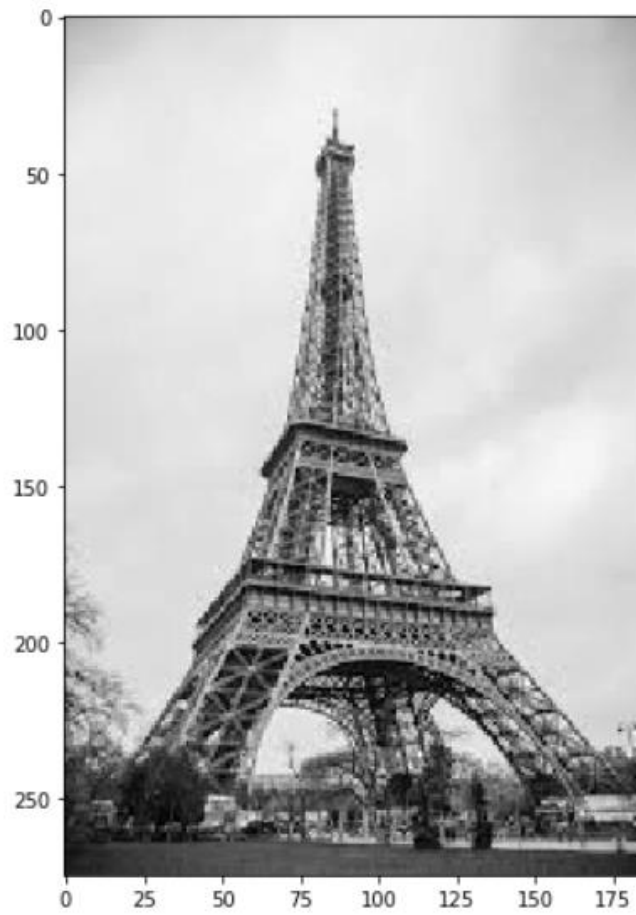


Figure (K)

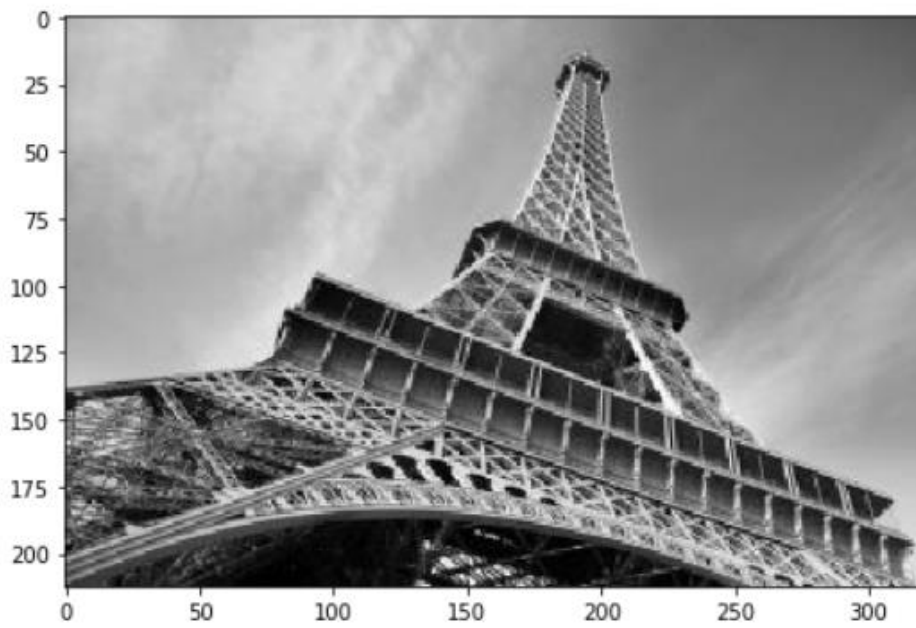


Figure (L)

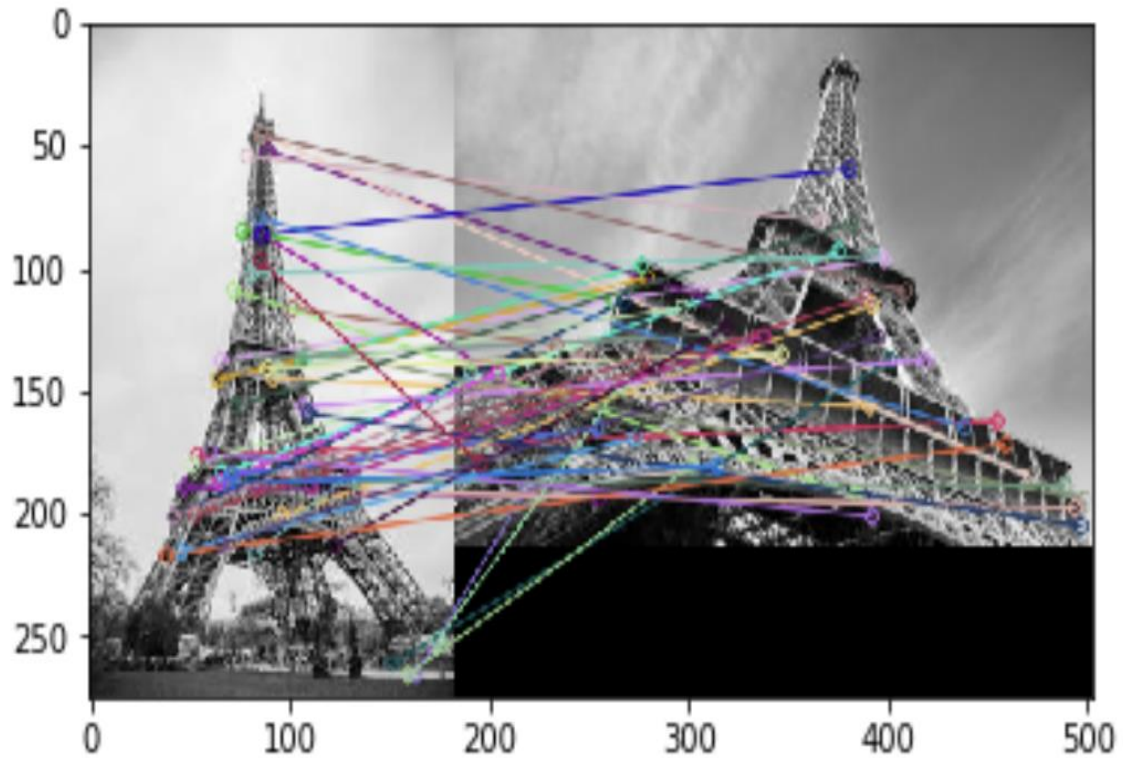


Figure (M)

Well above we have figure K and figure L they are images of the same object but taken from different angles. These images are related by linear transformation. Therefore, to recognize the relationship between these two images is by identifying the presence of linear transformation between them as we can see in figure M.

1.2.2 Normalization of the image with respect to the affine transformation

The affine transform

$$\begin{aligned} x' &= a_0 + a_1x + a_2y, \\ y' &= b_0 + b_1x + b_2y \end{aligned} \quad (1)$$

can be decomposed into 6 simple one-parameter transforms and one nonparameter

Horizontal translation :

$$u = x - x_0$$

$$u = y$$

Vertical translation:

$$u = x$$

$$v = y - y_0$$

Scaling :

$$u = \omega x$$

$$v = \omega y$$

First rotation :

$$u = x \cos \alpha - y \sin \alpha$$

$$v = x \sin \alpha + y \cos \alpha$$

Stretching :

$$u = \delta x$$

$$v = \frac{1}{\delta} y$$

Second rotation :

$$u = x \cos \rho - y \sin \rho$$

$$v = x \sin \rho + y \cos \rho$$

Mirror reflection :

$$u = x$$

$$v = \pm y \quad (2)$$

The above mathematical expression show that the ordering of one-parameter transforms can be changed, but the stretching must be between rotations [9]

1.3 Disproportionality function

The numerical function can be determined in several ways. For example, let us consider the technical diagnostics problem of a large class of substitutionary which is given as [10]

$$y = k(t)x \quad (1)$$

Here:

x and y are the input and output parameters, respectively;

t is time;

$k(t)$ changes slowly depending on $x(t)$.

Example of these object are converters, amplifiers and other sensors.

The static characteristic in the general case becomes

$$y = k(x,t)x + b(t), \quad (2)$$

$b(t) \neq 0.$

Where

In this case the deterioration should be detected and analyzed. If so, it will be easy to compare (1) and (2) but it will depend on $k(t)$ in (1) since it changes in time

If the value of x from (2) is not proportional to y , then the comparison of the values in (1) and (2) functions is excluded because $k(t)$ is unknown .

1.3.1 RESEARCH ISSUE STATEMENT

Assume there is a set X of real numbers and set T of real ordered numbers and set Y of real numerical functions[11]

Where by,

$$y = f(x,t), \quad (3)$$

where $x \in X, t \in T$.

Note that Y must be zero if the relationship between X and Y is proportional for $x \in X, t \in T$. [12]

1.3.2 SOLUTION OF THE ISSUE

The solution to the problem is as follows.

Given, $y = f(x), x \in X$.

The derivative of $y = f(x)$ is $\frac{dy}{dx}$

The limit of the ratio $\frac{\Delta y}{\Delta x}$ for $\Delta x \rightarrow 0$.

Thus, if the condition

$$\frac{y}{x} = \frac{dy}{dx}, \quad (4)$$

is carried for a given $x \in X$, then we can assume that at this point the relationship between x and y is proportional[13].

For $y = cx$,

as $c = \text{const}$.

The disproportionality the first-order derivative with respect to x .

The function $y = f(x)$

$$\text{The first order derivative @ } d_x^{(1)}y = \frac{y}{x} - \frac{dy}{dx} \quad (5)$$

@ is the disproportionality,

d is the derivative.

Proof 1. With respect to the first order function, $y = f(x,t)$, where $x \in X, t \in T$.

Thus, the disproportionality with respect to the first-order derivative of the function with respect to x for a given value of t has the form

$$@ d_x^{(1)} y_t = \frac{f(x,t)}{x} - f'_x(x,t) \quad (6)$$

This can be taken as the preference function.

1.3.3 N-ORDER DERIVATIVE DISPROPORTIONALITY

The n-order derivative must be equal to zero for

$$y = cx^n \quad (8)$$

Where,

$$\frac{y}{x^n} = \frac{1}{n!} \frac{d^n y}{dx^n} \quad (9)$$

Proof 2. The disproportionality with respect to the derivative of the n -order of a function $y = f(x)$ with respect to x is

$$@ d_x^{(n)} y = \frac{y}{x^n} - \frac{1}{n!} \frac{d^n x}{dx^n}, \quad (10)$$

And for the function $y = f(x,t)$ for a fixed t the n-order is

$$@ d_x^{(n)} y_t = \frac{f(x,t)}{x^n} - \frac{1}{n!} \frac{\partial^n x}{\partial x^n}, \quad (11)$$

In both (10) and in (11) n is an integer greater than zero[14].

Example. Let $y = k(t)x^3$. For a fixed t , $k(t)=c= \text{const}$.

Then, in accordance with (11), for

$$n=1 \quad @ d_x^{(1)} y_t = -2cx^2,$$

$$n=2 \quad @ d_x^{(2)} y_t = -2cx,$$

$$n=3 \quad @ d_x^{(3)} y_t = 0.$$

1.3.4 DERIVATIVE DISPROPORTIONALITY IF A FUNCTION IS SPECIFIED IN A PARAMETRIC FORM

$$\text{Let } x = \varphi(t), \quad y = \psi(t), \quad (14)$$

where $t \in [T_1, T_2]$ and $t = \Phi(x)$.

for $y = \Psi[\Phi(x)]$

Applying $\frac{d^n y}{dx^n}$ in parametric dependence of y on x .

for $n=1$

$$@ d_x^{(1)} y = @_{\phi(t)}^{(1)} \psi(t) = \frac{y}{x} - \frac{y_t'}{x_t'} = \frac{\psi(t)}{\phi(t)} - \frac{\psi_t'(t)}{\phi_t'(t)} \quad (15)$$

Is equal to zero varying in t for $\psi(t) = c(\Phi(x))$, where c is constant.

1.3.5 VALUE DISPROPORTIONALITY FUNCTIONS

The n -order value disproportionality with respect to x^n is as follows $@ v_x^{(n)} y$.

$$@ v_x^{(n)} y = y - \frac{x^n}{n!} \frac{d^n y}{dx^n} \quad (16)$$

for $n=1$

$$@ v_x^{(1)} y = y - x \frac{dy}{dx} \quad (17)$$

For $y = f(x, t)$ $x \in X, t \in T$ the value disproportionality of x^n , where n is an integer greater than zero, is determined for a fixed t [15].

$$@ v_x^{(n)} y_t = y_t - \frac{x^n}{n!} \frac{d^n y_t}{dx^n} \quad (18)$$

If the functions are specified in a parametric form in accordance with (15)

$$@ v_x^{(1)} y = @ v_{\phi(t)}^{(1)} \psi(t) = \psi(t) - \phi_t(t) \frac{\psi_t'(t)}{\phi_t'(t)} \quad (19)$$

***Proof.* Let us verify that the disproportionality with respect to the n -order derivative is not zero. In accordance with (11)**

$$@ d_x^{(n)} y_t = \frac{1}{x^n} [k_{n-1}(t)x^{n-1} + k_{n-2}(t)x^{n-2} \dots + k_1(t)x]. \quad (21)$$

By hypothesis, at least one of $k_{n-1}(t), k_{n-2}, \dots, k_1(t)$ is not equal to zero. Consequently, the disproportionality (21) is also not equal to zero.

Now we find a sequence $@(n)@ d_x^{(1)} y_t$ of n first-order disproportionalities (6) for (20).

For convenience, each disproportionality in this sequence will be denoted by Z with the corresponding index.[16]

$$Z_1 = @ d_x^{(1)} y_t = -[(n-1)k_n(t)x^{n-1} + (n-2)k_{n-1}(t)x^{n-2} + \dots + 2k_3(t)x^2 + k_2(t)x]$$

$$Z_2 = @ d_x^{(1)} Z_1 = [(n-1)(n-2)k_n(t)x^{n-2} + (n-2)(n-3)k_{n-1}(t)x^{n-3} + \dots + 6k_4(t)x^2 + 2k_3(t)x]$$

$$Z_i = @ d_x^{(1)} Z_{i-1} = (-1)^i \sum_{j=i}^n k_j(t)x^{j-i} \prod_{m=1}^i (j-m)$$

$$Z_{n-1} = @ d_x^{(1)} Z_{n-1} = (-1)^{n-1} (n-1)! k_n(t)x$$

$$Z_n = @ d_x^{(1)} Z_{n-1} = 0 [17].$$

This result was required to be obtained.

1.3.6 APPLICATION OF DISPROPORTIONALITY FUNCTIONS

- i. It enables us to obtain the information required to figure the causes of deterioration of the condition of the object.
- ii. It can be used as a numerical measure of the discrepancy between images.
- iii. It can be used to solve technical diagnostic problems and enable pattern recognition.

1.4 Computer modeling

Computer modeling is the process of writing a computer program in a mathematical way for a physical system or biological system. Computer modeling is used in various ways such as Aerospace, Architecture, Astronomy, Biology, Cognitive, Disaster, Ecology. It is also used in fields such as earthquake and models of buildings [18][19].

1.4.1 How it works

- Computer Modeling can be applied to recognize the face of unknown murder victim.
- It is applied when an accident occurs to calculate the speed of the vehicle before it happened. This is when the Newton's law of motion is applied which states that "The body in a uniform state of motion will remain in that state of motion unless acted upon by an external force".
- It also works by improving cancer treatment. Finding the cure and see how various patient respond to different drugs.
- In predicting the health code violations i.e., it helps to know the cause of the disease.
- It is applied in earthquake research.
- It responds to deadly pandemics. For example, a pandemic disease such as influenza in the 1918 occurred in this case the use of computer modeling was required to discover the right cure or vaccine for it.

1.4.2 Pros of computer modeling

- i. It is able to make changes quickly and see results.
- ii. It is able to repeat tests couple of times.
- iii. It helps to predict what will happen in the system.
- iv. They can be safer and cheaper compared to the real world.
- v. It answers a lot of “what if” questions.
- vi. It has a great understanding of the process.
- vii. It helps and enable the scientists to perform a lot of experiments using computer.
- viii. It is used to find unforeseen problems.

1.4.3 Cons of computer modeling

- i. The cost of the model can be expensive.
- ii. The cost of running various models can be high.
- iii. It takes a lot of time.
- iv. People reactions might affect the model because it can be unrealistic as people tend to react differently.
- v. Mistakes can be made while programming which might cause damage or loss.

2. SOLVING PROBLEM

2.1 Problem statement

It's necessary to develop an algorithm and a computer program for modelling the recognition system of affine connection between two contour images. Both images are convex.

From the outset to create a contour convex image. Then to create the affine transformation of this source image.

The computer program must recognize an event, when the affine connection between these images will be destroyed.

2.2 Mathematical statement of the problem

The plane convex contour image is given:

$$z=r(\theta),$$

Where r-is radius -vector;

θ -angle.

In a rectangular coordinate system x 0 y

$$X= r \cos(\theta)$$

$$Y= r \sin(\theta)$$

$$r = \sqrt{x^2 + y^2}$$

We set the coefficients a_{11} , a_{12} , a_{21} , a_{22} for the affine transformation of this image. Then in the rectangular coordinate system, the X and Y coordinates are defined as:

$$X = a_{11}x + a_{12}y$$

$$Y = a_{21}x + a_{22}y$$

Accordingly, the radius vector R and the angle θ are determined by the formulas:

$$R = \sqrt{X^2 + Y^2}$$

$$\theta = \arctg \frac{y}{x}$$

when creating the affine image on a certain interval of angle θ_1 to θ_2 , the affine connection between images will disappear.

It's necessary to develop the algorithm and the computer program to recognize this disappearance.

2.3 Choice of method for solving the problem

When solving the problem, it is necessary to control the proportional relationship between the areas of the source and converted images.

In this case, the proportionality coefficient is unknown. This can be done using the disproportionality functions. Specifically, to solve the problem, the integral disproportionality function with respect to the derivative of the first order was chosen.

2.4 Process solving of the problem

The affine transformation property is used, which consist in the presence of the proportional relationship between the areas of the original and converted images:

$$S = Ks,$$

Where s is the area of original image:

S is the area of converted image;

$$K = a_{11} a_{22} - a_{12} a_{21}$$

It's is known that the area of the curved sector is calculated by the formula:

$$F = \frac{1}{2} \int_{\alpha}^{\beta} R^2(\phi) d\phi$$

Where F -is area;

ϕ -angle,

R -radius vector.

Correspondingly, the areas of fragments of the original and transformed images at intervals of angle changes are described by the expressions

$$\Delta S = \frac{1}{2} \int_{\phi - \Delta\phi}^{\phi} \zeta^2(\phi) d\phi$$

$$\Delta S' = \frac{1}{2} \int_{\theta - \Delta\theta}^{\theta} R^2(\theta) d\theta$$

Let's describe both images as one -dimension arrays: $\phi_0, \dots, \phi_{N-1}, \dots, \theta_0, \dots, \theta_{N-1}$.

Here N -is a number of elements. Correspondingly will be arrays S_0, \dots, S_{N-1} of the origin image and S'_0, \dots, S'_{N-1} of the converted one. The elements of the area array are defined as:

$$S_i = S_{i-1} + 0.5r_i^2 (\phi - \phi_{i-1})$$

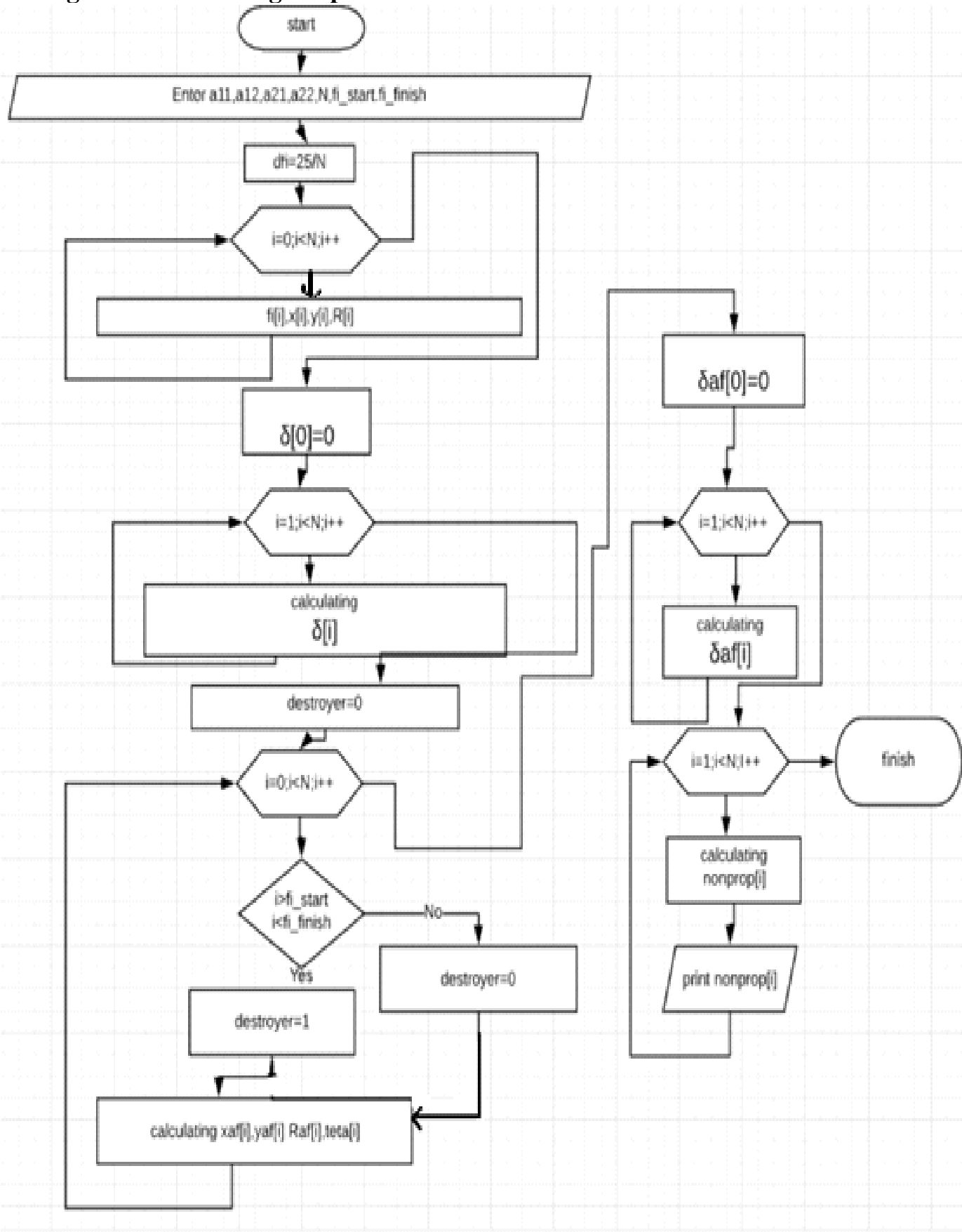
$$S'_i = S'_{i-1} + 0.5R_i^2 (\theta_i - \theta_{i-1})$$

As i increases from 0 to N , it is required to control the proportional relationship between the current areas s_i and S_i of images. Since the areas are represented discretely, their derivative doesn't exist. There to solve this problem it is necessary to use integral disproportionality functions with respect to the first order derivative

$$I_i = @I_S^{(1)} S'_i = \frac{S'_{i-1} + S'_i}{S_{i-1} + S_i} - \frac{S'_i}{S_i}$$

In the presence of the proportional connection, the disproportionality is zero. However, when using integral disproportionality, the calculations are approximate. Therefore, it is necessary to compare the obtained disproportionality with a certain ε close to zero. If $I_i \leq \varepsilon$, then it should be assumed that a proportional connection is disproportionate and this means that for a given angle there is no affine connection.

2.5 Algorithm for solving the problem



In Table 1 there a list of identifiers that are used for describing algorithm

Table 1

VARIABLES	IDENTIFIERS	EXPLANATION
N	N	Number of elements
r[N], R[N]	R[N],Raf[N]	Radius vectors
ϕ [N], θ [N]	Fi[N], teta[N]	Angles
X, Y	xaf, yaf	Affine coordinates
x, y	x ,y	Rectangular coordinates
ϕ_1, ϕ_2	Fi_start,fi_finish	Start and Finish angles
a ₁₁ ,a ₁₂ ,a ₂₁ ,a ₂₂	a ₁₁ ,a ₁₂ ,a ₂₁ ,a ₂₂	Coefficient
s[N],S[N]	s[N],Saf[N]	Areas
	x0,y0	Starting point coordinates
I[N]	Nonprop[N]	Disproportionality

3. Instructions for user

The program is written in Borland C ++

Source file felista.cpp 2216 bytes

work file felista.exe 33810 bytes

Data is entered for the test mode automatically; for the operating mode, input from the keyboard.

Results are output into two files: result.dan, res_exl.dan.

The first for reading and the second for plotting with using Excel.

4. Control Testing

Initial data for testing: a₁₁=0,5, a₁₂=15, a₂₁=0,2, a₂₂=3, x₀=10, y₀=15, R₀=10.

The pitch of the change of the polar angle is 0.0628.

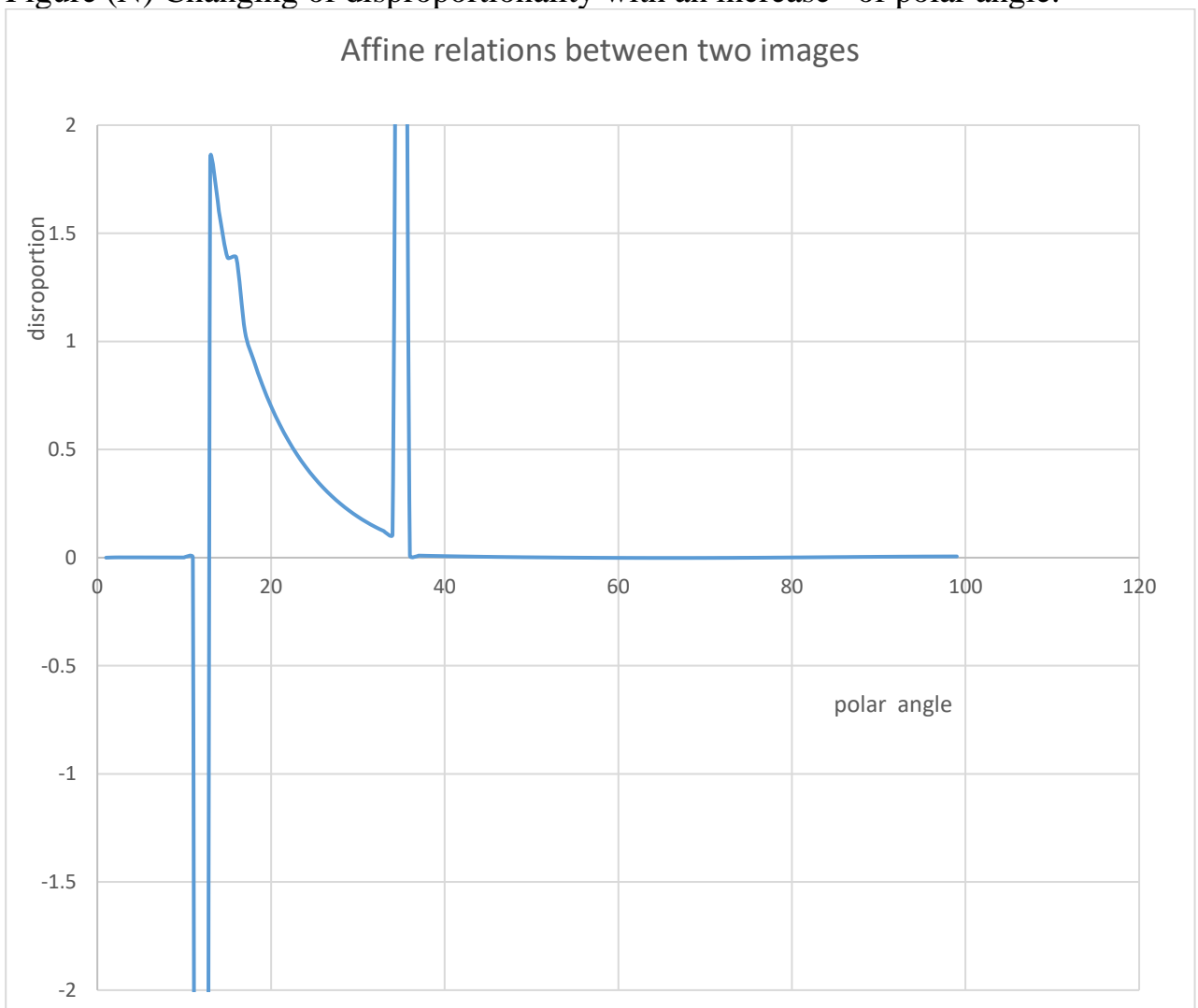
It is envisaged that when the angle changes from 0 to 0.6908 and from 2.198 to 6.28, the relationship between the contour images is affine. In the interval from 0.6908 till 2.198, this connection is destroyed. The results are shown in Table 2 and Fig.N

Table 2

№	nonprop	№	nonprop	№	nonprop	№	nonprop	№	nonprop
1	0	21	0,615938	41	0,006444	61	-0,00104	81	0,001299
2	0,001207	22	0,541719	42	0,005769	62	-0,00114	82	0,001617
3	0,001356	23	0,476743	43	0,005138	63	-0,00122	83	0,00194
4	0,001356	24	0,419596	44	0,004547	64	-0,00128	84	0,002265
5	0,00131	25	0,369128	45	0,003994	65	-0,00132	85	0,00259

6	0,001244	26	0,324403	46	0,003475	66	-0,00133	86	0,002911
7	0,001169	27	0,284652	47	0,002989	67	-0,00131	87	0,003225
8	0,00109	28	0,24925	48	0,002534	68	-0,00127	88	0,003531
9	0,00101	29	0,217686	49	0,002108	69	-0,00121	89	0,003824
10	0,000929	30	0,189548	50	0,001709	70	-0,00112	90	0,004103
11	0,000849	31	0,164515	51	0,001337	71	-0,00101	91	0,004367
12	-20,5896	32	0,142352	52	0,000991	72	-0,00088	92	0,004612
13	1,83876	33	0,122921	53	0,00067	73	-0,00072	93	0,004839
14	1,60196	34	0,106208	54	0,000374	74	-0,00053	94	0,005045
15	1,38754	35	6,81173	55	0,000101	75	-0,00033	95	0,00523
16	1,38754	36	0,010642	56	-0,00015	76	-0,0001	96	0,005393
17	1,04711	37	0,009673	57	-0,00037	77	0,000145	97	0,005534
18	0,913734	38	0,008776	58	-0,00057	78	0,000411	98	0,005653
19	0,799518	39	0,007943	59	-0,00075	79	0,000694	99	0,005751

Figure (N) Changing of disproportionality with an increase of polar angle.



Control of Affine linkage between two images.

The results show that in the intervals where there is an affine relationship between the images, the disproportionality is equal to zero. In the interval from 11 to 35, the disproportionality is not equal to zero. This indicates that there is no affinity bond in this interval.

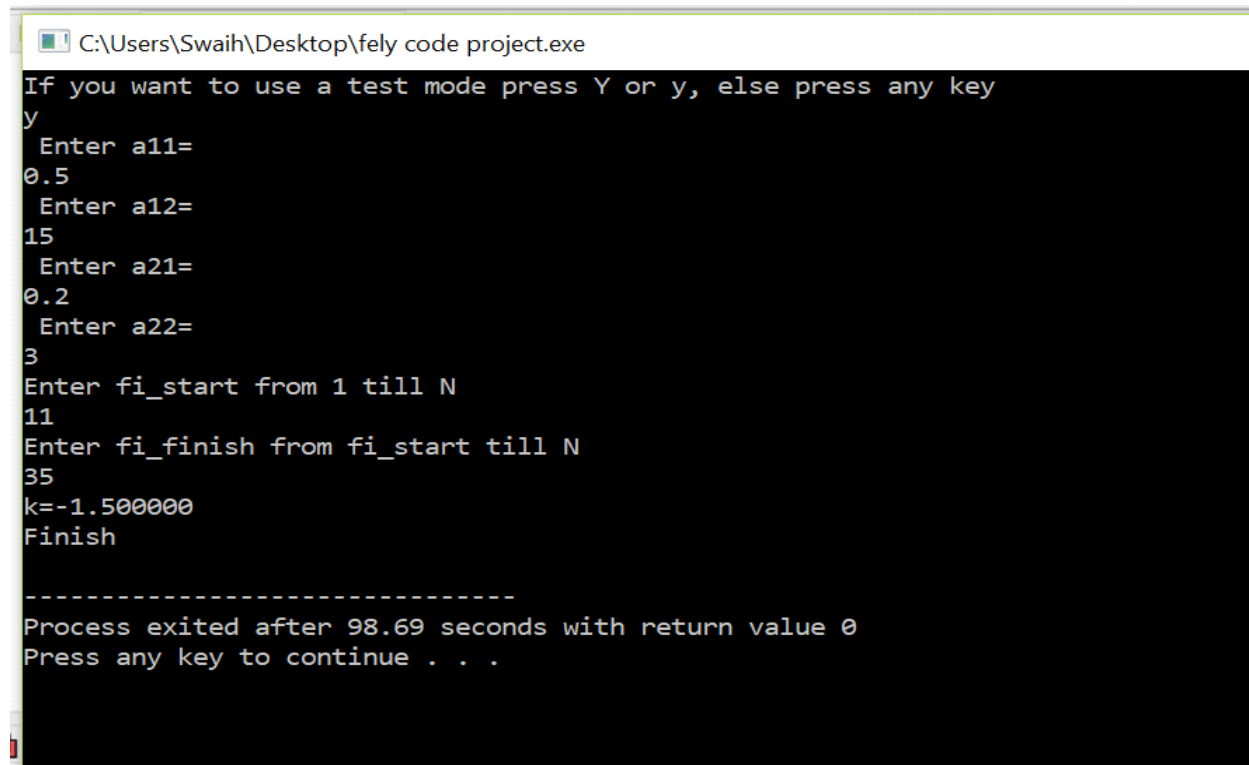
Thus, it is confirmed that the proposed algorithm and the computer program work correct

Conclusion

An algorithm and a computer program have been developed for recognizing the affine relationship between the contours of convex flat images. A feature of the proposed method is the ability to detect affinity linkage disturbances at one point.

The operation of the algorithm and the computer program is tested on a test example. The algorithm and computer program can be used in artificial vision systems as well as in the analysis of images obtained by aerial photography.

RESULTS



```
C:\Users\Swaih\Desktop\fely code project.exe
If you want to use a test mode press Y or y, else press any key
y
Enter a11=
0.5
Enter a12=
15
Enter a21=
0.2
Enter a22=
3
Enter fi_start from 1 till N
11
Enter fi_finish from fi_start till N
35
k=-1.500000
Finish

-----
Process exited after 98.69 seconds with return value 0
Press any key to continue . . .
```

An analysis of the results shows that the algorithm and the computer program work correctly

References

- [1]. S. C. Pei and C. N. Lin, "Image normalization for pattern recognition," *Image and Vision Computing*, vol. 13, pp. 711–723, 1995.
- [2]. <https://medium.com/ai-techsystems/image-detection-recognition-and-image-classification-with-machine-learning-92226ea5f595>
- [3]. Dr. Dataman-towards data science image recognition
- [4]. <https://marutitech.com/working-image-recognition/>
- [5]. <https://www.talkwalker.com/blog/best-image-recognition-tools>
- [6]. <https://www.google.com.ua/amp/s/www.learnopencv.com/image-recognition-and-object-detection-part1/amp/>
- [7]. <https://medium.com/ai-techsystems/image-detection-recognition-and-image-classification-with-machine-learning-92226ea5f595>
- [8]. Croft, H. T.; Falconer, K. J.; and Guy, R. K. *Unsolved Problems in Geometry*. New York: Springer-Verlag, p. 3, 1991.
- [9]. Gray, A. *Modern Differential Geometry of Curves and Surfaces with Mathematica*, 2nd ed. Boca Raton, FL: CRC Press, p. 130, 1997.
- [10]. Zwillinger, D. (Ed.). "Affine Transformations." §4.3.2 in *CRC Standard Mathematical Tables and Formulae*. Boca Raton, FL: CRC Press, pp. 265-266, 1995.
- [11]. D. Shen and H. H. S. Ip, "Generalized affine invariant image normalization," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, pp. 431–440, 1997.
- [12]. Avramenko V.V. *Harakteristiki neproporcional'nosti chislovyh funkcij.*- Dep. V GNTB Ukrainy 19.01.98, №59-Uk98. unpublished.
- [13]. Shkurin G. P. *Spravochnik po elektro- i elektronno-izmeritel'nyh priboram* — M., 1982
- [14]. Poltorak V.P., Dorogoj Ja.Ju. *Sistema raspoznavaniya obrazov na baze nechetkogo nejronnogo klassifikatora.* <http://aaecs.org/poltorak-vp-dorogoi-yayu-sistema-raspoznavaniya-obrazov-na-baze-nechetkogo-neironnogo-klassifikatora.html>.
- [15]. Krasil'nikov N.N. *Cifrovaja obrabotka 2-D i 3-D izobrazhenij : uchebn. posobie.*- SPb: BHV – Peterburg, 2011.- 608s.
- [16]. Avramenko V.V. , Slepushko N.Ju. *Raspoznavanie jetalonnih signalov pri nepolnoj informacii o harakteristikah pomeh*//*Visnik SumDU. Serija Tehnichni nauki.* №4 2009, s.13-19.
- [17]. Avramenko V.V. *Harakteristiki neproporcional'nostej i ih primenenie pri reshenii zadach diagnostiki*/ V.V. Avramenko // *Vestnik SumGU.* – 2000. – №16.
- [18]. <https://www.nibib.nih.gov/science-education/science-topics/computational-modeling>.
- [19]. <https://www.encyclopedia.com/social-sciences-and-law/law/crime-and-law-enforcement/computer-modeling>.

Appendix

The code of the program `felista.cpp` in the algorithmic language Borland C ++ is attached.

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#define N 100

int main()
{
double R[N],fi[N],dfi,R0=10;
double s[N];
double Raf[N],teta[N],saf[N];
double a11=0.5,a12=15,a21=0.2,a22=3,k;
int fi_start=11,fi_finish=35;
double x[N],y[N],x0=10,y0=15;
double xaf[N],yaf[N],nonprop[N];
int i,destroyer=0;
char nch;
FILE*p,*p1;
p=fopen("result.dan","w");
p1=fopen("res_exl.dan","w");
puts("If you want to use a test mode press Y or y, else press any key");
fflush(stdin);
scanf("%c",&nch);
if(nch!='Y' && nch!='y')
{
puts(" Enter a11=");
scanf("%lg",&a11);
puts(" Enter a12=");
scanf("%lg",&a12);
puts(" Enter a21=");
scanf("%lg",&a21);
puts(" Enter a22=");
scanf("%lg",&a22);
puts("Enter fi_start from 1 till N");
scanf("%i",&fi_start);
puts("Enter fi_finish from fi_start till N");
scanf("%i",&fi_finish);
}
}
```

```

k=a11*a22-a12*a21;

dfi=2*M_PI/N;
printf("k=%lf\n",k);// Source Image Generation
for(i=0;i<N;i++)
{
fi[i]=i*dfi;
x[i]=x0+R0*cos(fi[i]);
y[i]=y0+R0*sin(fi[i]);
R[i]=sqrt(pow(x[i],2)+pow(y[i],2));
//fprintf(p,"fi[%i]=%lf x=%lg y=%lg R[%i]=%lf\n",i,fi[i],x[i],y[i],i,R[i]);
}
// Calculating elementary areas for the original image
s[0]=0;
for(i=1;i<N;i++)
{
s[i]=s[i-1]+0.5*pow(R[i],2)*(fi[i]-fi[i-1]);
//fprintf(p,"fi[%i]=%lf R[%i]=%lf s[%i]=%lf\n",i,fi[i],i,R[i],i,s[i]);
}
// Affine Image Generation
for(i=0;i<N;i++)
{
if(i>fi_start && i<fi_finish )
destroyer=1;
else
destroyer=0;
xaf[i]=a11*x[i]+a11*destroyer*(x[11]+sqrt(fabs(x[i]-x[11]))) +a12*y[i];
yaf[i]=a21*x[i]+a22*y[i]+a22*destroyer*(y[11]+sqrt(fabs(y[i]-y[11])));
//fprintf(p1,"%lg %lg\n",xaf[i],yaf[i]);
Raf[i]=sqrt(pow(xaf[i],2)+pow(yaf[i],2));
teta[i]=atan(yaf[i]/xaf[i]);
//fprintf(p," Raf[%i]=%lf teta[%i]=%lf\n",i,Raf[i],i,teta[i]);
}
//Calculating elementary areas for the affine image
saf[0]=0;
for(i=1;i<N;i++)
{
saf[i]=saf[i-1]+0.5*pow(Raf[i],2)*(teta[i]-teta[i-1]);
//fprintf(p2,"%lf %lg\n",s[i],saf[i]);
}
// Calculating disproportion function for elementary areas
for(i=1;i<N;i++)
{
nonprop[i]=(saf[i-1]+saf[i])/(s[i-1]+s[i])-saf[i]/s[i];
}

```

```
fprintf(p," nonprop[%i]=%lg \n",i,nonprop[i]);  
fprintf(p1," %i  %lg \n",i,nonprop[i]);  
}  
puts("Finish");
```