Artificial Intelligence and Robotics Research 人工智能与机器人研究, 2019, 8(4), 197-207
Published Online November 2019 in Hans. http://www.hanspub.org/journal/airr
https://doi.org/10.12677/airr.2019.84022

Development and Application of MATLAB (GUI) Image Segmentation System Platform Based on PSO-KMeans Algorithm

Xingzhi Chen, Wentao Le, Daiwen Wang, Feixiang Huang, Naiyao Liu

School of Science, Southwest University of Science and Technology, Mianyang Sichuan
Email: 1451025961@qq.com

Received: Oct. 21st, 2019; accepted: Nov. 7th, 2019; published: Nov. 14th, 2019

Abstract

Image segmentation is not only a classic difficult problem but also an important step in digital image processing. Finding a fast and efficient image segmentation method is very important. Based on the simplicity and easy operation of GUI design and development in MATALB, this paper is based on the theoretical research of PSO-KMeans image segmentation combination algorithm, and compared with the segmentation results and data of KMeans algorithm. Through the layout design of the image user interface related controls and the writing of the callback function program, a set of MATLAB (GUI) image segmentation system platform based on PSO-KMeans algorithm was developed, which realized the segmentation result, parameter output and data visualization in image segmentation. The simple operation solves the complicated problem in the image segmentation process. At the same time, the sample image in the image segmentation application is selected to segment the image. The results show that the image segmentation system has a friendly interface and simple operation, and accurately realizes image segmentation, and obtains the related data of segmentation and the comparison with the segmentation results of similar algorithms. This study improves the simplicity and visualization of image segmentation.

Keywords

PSO-KMeans Algorithm, Image Segmentation System, GUI, Result Comparison, Segmentation Application

基于PSO-KMeans算法的MATLAB(GUI)图像分割系统平台开发应用

陈兴志，乐文涛，王代文，黄飞翔，刘乃瑶

西南科技大学理学院，四川 绵阳

文章引用: 陈兴志, 乐文涛, 王代文, 黄飞翔, 刘乃瑶. 基于PSO-KMeans算法的MATLAB(GUI)图像分割系统平台开发应用[J]. 人工智能与机器人研究, 2019, 8(4): 197-207. DOI: 10.12677/airr.2019.84022

Email: 1451025961@qq.com

收稿日期：2019年10月21日；录用日期：2019年11月7日；发布日期：2019年11月14日

摘　要

图像分割是一个经典的难题，也是数字图像处理的重要一步。找到一种快速高效的图像分割方法具有十分重要的意义。本文利用MATALB中GUI设计开发的简便性和易操作等特性，以PSO-KMeans图像分割组合算法理论研究为基础，并与同类分割算法KMeans算法的分割结果和数据作比较。通过图像用户界面相关控件的布局设计及回调函数程序的编写，开发了一套基于PSO-KMeans算法的MATLAB(GUI)图像分割系统平台，实现了对图像分割中分割结果、参数输出、数据可视化的简洁操作，解决了图像分割过程中实现复杂的问题。同时，选取在图像分割应用中的示例图片对图像进行分割测试。结果表示，图像分割系统界面友好，操作简单，准确实现了图像的分割，并得到分割的相关数据以及与同类算法分割结果的比较。该研究提高了图像分割的简单性以及可视化性。

关键词

PSO-KMeans算法，图像分割系统，GUI，结果对比，分割应用

Copyright © 2019 by author(s) and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
http://creativecommons.org/licenses/by/4.0/
Open Access

1. 引言

图像分割是图像识别和计算机视觉至关重要的预处理[1]。找出一种合理的算法会使此项技术更加广泛应用各个方面的研究。目前国内外对于图像分割已经有了一些研究成果，其中，K 均值算法是最常用的图像分割算法。也有使用 PSO 算法对图像进行分割的。但是无论是 K 均值算法还是 PSO 算法，其作为单一的图像分割算法还是存在一定的不足。PSO 算法是一种随机搜索算法[2]，K-Means 算法是针对图像分割的有效方法[3]。针对 K-means 算法的不足，把 PSO 算法引入到 K-Means 聚类算法中，以提高 K-Means 算法的效率和结果精度[4]。但是这一过程比较复杂，因此，有必要开发一款界面友好，操作简单的图像分割系统。在近年来的研究中，有不少学者利用 GUI 开发系统设计。李磊等人[5]利用 MATLAB 跟 GUI 设计了图像处理系统，李念念等人[6]利用 GUI 设计了信号与系统分析软件，刘艳华等人[7]设计了一个数字调制系统的 MATLAB GUI 用户界面。黄冉等人[8]开发了基于 MATALB(GUI)的灰色预测系统。但是对于图像分割系统 GUI 的开发和研究还很少。因此，本文基于 PSO-KMeans 算法为理论基础，运用 MATLAB (GUI)开发出了基于 PSO-KMeans 算法的图像分割系统，并选取原始灰度图像对系统进行测试，分割结果良好，并且可以提供分割的相关参数以及分割数据的图表可视化，提高了图像分割的简单性以及可视化性。

2. 基于 PSO-KMeans 算法的图像分割方法

设在 D 维的像素矩阵搜索空间中，有 N 个像素矩阵元素组成一个群落，其中第 i 个像素矩阵元素表示为 D 维的向量：$x_i=(x_{i1},x_{i2},\cdots,x_{in}),i=1,2,\cdots N$。第 i 个像素矩阵元素的“飞行”速度也是 D 维的向量，

记为 $v_i=\left(v_{i1},v_{i2},\cdots,v_{iD}\right),i=1,2,\cdots N$。第 i 个像素矩阵元素迄今为止搜索到的最优位置称为个体极值，记为 $p_i=\left(p_{i1},p_{i2},\cdots,p_{iD}\right),i=1,2,\cdots N$。整个像素矩阵群迄今为止搜索到的最优位置为全局极值，记为 $g_{best}=\left(p_{g1},p_{g2},\cdots p_{gD}\right)$。在找到这两个最优值时，像素矩阵元素根据如下的公式来更新自己的速度和位置：$v_{id}^{(k+1)}=w_{id}^{(k)}+c_1 rand_1^{(k)}\left(p_{bestid}^{(k)}-x_{id}^{(k)}\right)+c_1 rand_2^{(k)}\left(p_{bestid}^{(k)}-x_{id}^{(k)}\right),x_{id}^{(k+1)}=x_{id}^{(k)}+v_{id}^{(k+1)}$ [9]。

通过 PSO 算法将 N 个对象分为 K 个簇，并将其通过迄今为止搜索得到的最优值回代到 K-Means 算法中作为初始聚类分割中心和初始分割类 K，K-Means 算法是以 K 为参数对图像分割，使簇内相似度较高，簇间较低。相似度用欧式距离表示。PSO-KMeans 组合算法的算法思想和流程图如下图 1 所示：

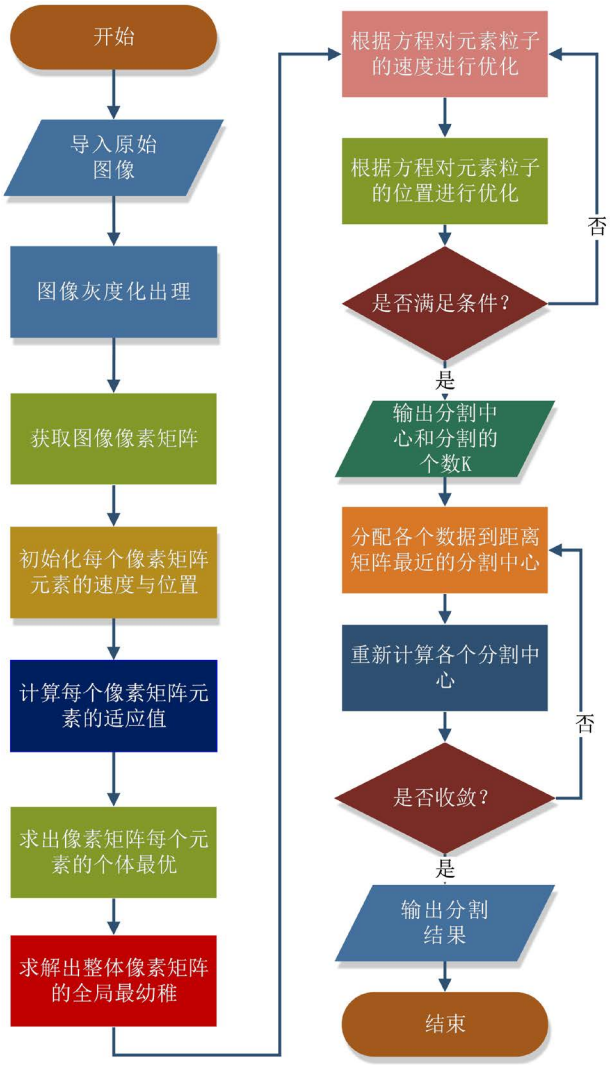

Figure 1. PSO-KMeans combination algorithm idea
图 1. PSO-KMeans 组合算法思想

2. 基于 PSO-KMeans 算法的 MATLAB(GUI)图像分割系统平台开发

本文通过对 PSO-KMeans 图像分割组合算法进行研究，并通过与 KMeans 图像分割算法的分割结果和数据进行比较，更加体现基于 PSO-KMeans 分割算法的高效性和价值，最后基于 MATLAB(GUI)理论对图像分割系统平台进行设计，对图像分割系统的运行界面进行设计和对回调函数程序进行了编写，对

图像分割系统的 Fengesystem 主界面和 Fengeresult (分割结果)、Fengeparameter (参数输出)和 Fengedata (数据可视化)三个子界面进行了设计和代码的编写。

2.1. Fengesystem 分割系统主界面设计

新建一个图形用户界面，打开 GUIDE 中自带的空白的输出编辑器，添加相应控件：选取 1 个 Static Text 静态文本块，4 个 Pushbutton 按钮分别放置在界面的合适位置，点击保存并命名该图形界面为 Fengesystem。双击各个控件，利用属性查看器查看其属性，并设置参数。设置 FontSize 的值并更改 String 和 Tag。修改了各项属性后的分割系统 Fengesystem 主界面如图 2 所示。

Figure 2. Split system Fengesystem main interface
图 2. 分割系统 Fengesystem 主界面

回调函数编写

利用 MATLAB 生成的两个 Fengesystem.m 文件和 Fengesystem.fig 文件，保存到相应文件路径下，点击并打开 Fengesystem.m 文件，对回调函数的代码进行编写：

1) 在“分割结果”按钮的回调函数 functionFengeresult_Callback 下编写相应代码如下：

run Fengeresult;

2) 在“参数输出”按钮的回调函数 functionFengeparameter_Callback 下编写相应代码如下：

run Fengeparameter;

3) 在“数据可视化”按钮的回调函数 functionFengedata_Callback 下编写相应代码如下：

run Fengedata;

4) 在“退出”按钮的回调函数 functionExit_Callback 下编写相应代码如下：

run quit;

5) 保存并运行 Fengesystem 主界面的 GUI 功能界面。

2.2. Fengeresult 分割结果子界面设计

新建一个图形用户界面，打开 GUIDE 中自带的空白的输出编辑器，添加相应控件：4 个 StaticText 静态文本块，1 个 EditText 编辑文本块，2 个 Pushbutton 按钮，3 个 Axes 绘图块。击保存并命名该图形界面为 Fengeresult。双击各个控件，利用属性查看器查看其属性，并设置参数。设置 FontSize 的值并更改 String 和 Tag。修改属性后的分割系统 Fengeresult 分割结果界面如图 3 所示：

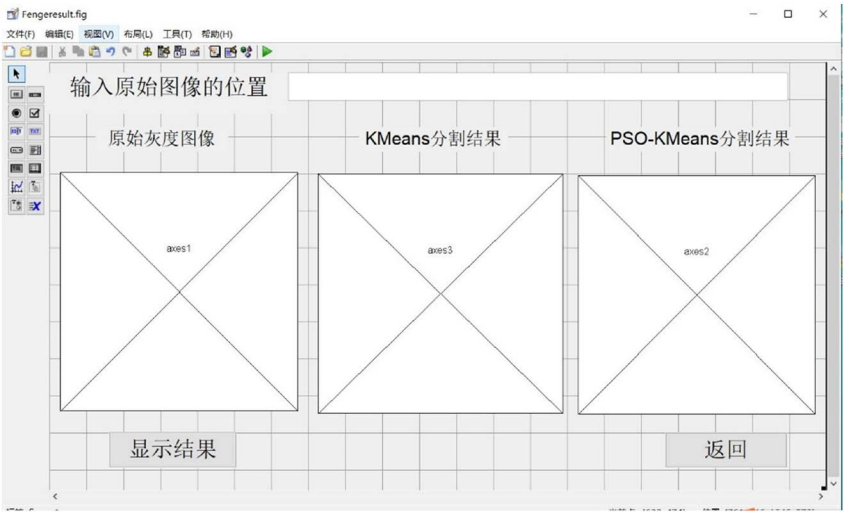

Figure 3. Initial segmentation result interface after modifying attributes
图 3. 修改属性后的初始分割结果界面

回调函数编写

利用 MATLAB 生成的两个生成 Fengeresult.m 和 Fengeresult.fig 文件，保存到相应文件路径下，点击并打开 Fengeresult.m 文件，找到相应的回调函数位置，对回调函数的代码进行编写：

1) 在“输入原始图像位置”的动态文本框的回调函数 function data_Callback 处编写如下代码：

I=imread(get(handles.edit1, ‘string’));

2) 在“显示结果”按钮的回调函数 function result_Callback 处编写如下代码：

```
J=rgb2gray(I);
[a,b]=size(J);
[p,x]=imhist(J,256);
L=x';
LP=p'/(a*b);
n=256;
c1=2;
c2=2;
wmax=0.9;
wmin=0.4;
G=10;
M=15;
X=min(L)+fix((max(L)-min(L))*rand(1,M));
V=min(L)+(max(L)-min(L))*rand(1,M);
m=0;
for i=1:1:n
m=m+L(i)*LP(i);
end
pbest=zeros(M,2);
gbest1=0;
gbest2=0;
gbest3=0;
GG=0;
for k=1:1:G
w(k)=wmax-(wmax-wmin)*k/G;
for i=1:1:M
t=length(find(X(i)>=L));
r=0;
end
kk=1:1:G;
gbest1
k = 2;
[m, n, p] = size(K);
X = reshape(double(K), m*n, p);
rng('default');
sey1 = X(X>gbest1);
aver1 = ceil(mean(sey1(:)));
[ro_1,co_1]=find(X==aver1);
ZB1 = [ro_1,co_1];
[Hshu1,Lshu1] = size(ZB1);
center1=ZB1(randperm(Hshu1, 1),:);
sey2 = X(X<gbest1);
aver2 = ceil(mean(sey2(:)));
[ro_3,co_3]=find(X==aver2);
ZB2 = [ro_3,co_3];
[Hshu2,Lshu2] = size(ZB2);
center2= ZB2(randperm(Hshu2, 1),:);
C=[X(center1(1,1), : );X(center2(1,1), : )];
J_prev = inf;
iter = 0;
JJ = [];
tol = 1e-10;
[m, n, p] = size(J);
k=2;
while true
iter = iter+1;
```

Continued

```
s=0;
for j=1:1:t
r=r+LP(j);
s=s+L(j)*LP(j);
end
W0(i)=r;
W1(i)=1-r;
U0(i)=s/r;
U1(i)=(m-s)/(1-r);
end
for i0=1:1:M
BB(i0)=W0(i0)*W1(i0)*((U1(i0)-U0(i0))^2);
end
for i=1:1:M
if pbest(i,2)<BB(i)
pbest(i,2)=BB(i);
pbest(i,1)=X(i);
end
end
[MAX,CC]=max(BB);
[MIN,DD]=min(BB);
if MAX>=gbest2
gbest2 = MAX;
gbest1 = X(CC);
end
if MIN>=gbest3
gbest3 = MIN;
end
GG(k)=gbest3;
fori=1:1:M
V(i)=round(w(k)*V(i)+c1*rand*(pbest(i,1)-x(i))+c2*rand*(gbest1
-x(i)));
X(i)=V(i)+X(i);
end
end
K = 0;
K = J;
sey1 = K(K>gbest1);
sey2 = K(K<gbest1);
average1 = mean(sey1(:));
average2 = mean(sey2(:));
%SPG=abs(average1-average2)/(average1+average2)
for i=1:1:a
for j=1:1:b
if J(i,j)>gbest1
J(i,j)=250;
else
J(i,j)=0;
end
end
dist = sum(X.^2, 2)*ones(1, k) + (sum(C.^2, 2)*ones(1, m*n))' -
2*X*C';
[~, label] = min(dist, [], 2) ;
for i = 1:k,
C(i, :) = mean(X(label == i , :));
End
J_cur = sum(sum((X - C(label, :)).^2 , 2));
JJ = [JJ, J_cur];
display(sprintf('#迭代次数: %03d, 目标函数值: %f', iter, J_cur));
if norm(J_cur-J_prev, 'fro') <tol,
break;
end
J_prev = J_cur;
end 结束
I_seg = reshape(C(label, :), m, n, p);
I=rgb2gray(I); %灰度化处理
axes(handles.axes1);
subplot(1, 2, 1), imshow(I, [])
axes(handles.axes2);
subplot(1, 2, 2), imshow(uint8(I_seg), [])X = reshape(double(I),
m*n, p);
rng('default');
SB = randperm(m*n, k);
C = X(SB, :);
J_prev = inf;
iter = 0;
J = [];
tol = 1e-2;
while true,
iter = iter + 1;
dist = sum(X.^2, 2)*ones(1, k) + (sum(C.^2, 2)*ones(1, m*n))' -
2*X*C';
[~, label] = min(dist, [], 2) ;
for i = 1:k,
C(i, :) = mean(X(label == i , :));
end
J_cur = sum(sum((X - C(label, :)).^2, 2));   %目标函数计算
J = [J, J_cur];
display(sprintf('#迭代次数: %03d, 目标函数值: %f', iter, J_cur));
if norm(J_cur-J_prev, 'fro') <tol,
break;
end
J_prev = J_cur;
end
I_Seg = reshape(C(label, :), m, n, p);
axes(handles.axes3);
imshow(uint8(I_Seg), [])
```

3) 在“返回”按钮的回调函数 function exit_Callback 处编写如下代码：

close Fengeresult;

FengeSystem;

4) 保存并运行 Fengeresult 分割结果子界面。

2.3. Fengeparameter 参数输出子界面设计

新建一个图形用户界面，打开 GUIDE 中自带的空白的输出编辑器，添加相应控件：选取 4 个

Pushbutton 按钮，16 个 Edit Text 静态文本块，16 个 Edit Text 编辑文本块，10 个 Panel 面板块，点击保存并命名该图形界面为 Fengeparameter。双击各个控件，利用属性查看器查看其属性，并设置参数。设置 FontSize 的值并更改 String 和 Tag。修改属性后的分割系统 Fengeparameter 参数输出界面如图 4 所示：

Figure 4. Parameter output sub-interface
图 4. 参数输出子界面

回调函数编写

利用 MATLAB 生成的两个生成 Fengeparameter.m 和 Fengeparameter.fig 文件，保存到相应文件路径下，点击并打开 Fengeparameter.m 文件，找到相应的回调函数位置，对回调函数的代码进行编写：

1) 在“输出参数”按钮的回调函数 function paramete_Callback 处编写如下代码：

```
t1 = clock;
center1 = ZB1(randperm(Hshu1, 1),:);
str1=num2str(center1) ;
set(handles.edit3, 'string',str1);
center2 = ZB2(randperm(Hshu2, 1),:);
str2=num2str(center2);
set(handles.edit4, 'string',str2);
set(handles.edit7, 'string',iter);
set(handles.edit8, 'string',J_cur);
t2 = clock;      %计时器
t = etime(t2,t1)
set(handles.edit5, 'string',t);
t11 = clock;
t22 = clock;      %计时器
tt = etime(t22,t11)
set(handles.edit18, 'string',tt);
average1 = sum(C(C>=gbest1));
average2 = sum(C(C<gbest1));
SPGG = abs(average1-average2)/(average1+average2)
set(handles.edit19, 'string',SPGG);
set(handles.edit20, 'string',iter);
set(handles.edit21, 'string',J_cur);
```

2)在“返回”按钮的回调函数 function exit_Callback 处编写如下代码：

close Fengeparameter;

FengeSystem;

3) 保存并运行 Fengeparameter 参数输出子界面。

2.4. 数据可视化子界面的创建

新建一个图形用户界面，打开 GUIDE 中自带的空白的输出编辑器，添加相应控件：选取 4 个 Pushbutton 按钮，6 个 Static Text 静态文本编辑块，4 个 Axes 绘图块，2 个绘图表模块。点击保存并命名

该图形界面为 Fengedata。双击各个控件，利用属性查看器查看其属性，并设置参数。设置 FontSize 的值并更改 String 和 Tag。修改属性后的分割系统 Fengedata 数据可视化子界面如图 5 所示：

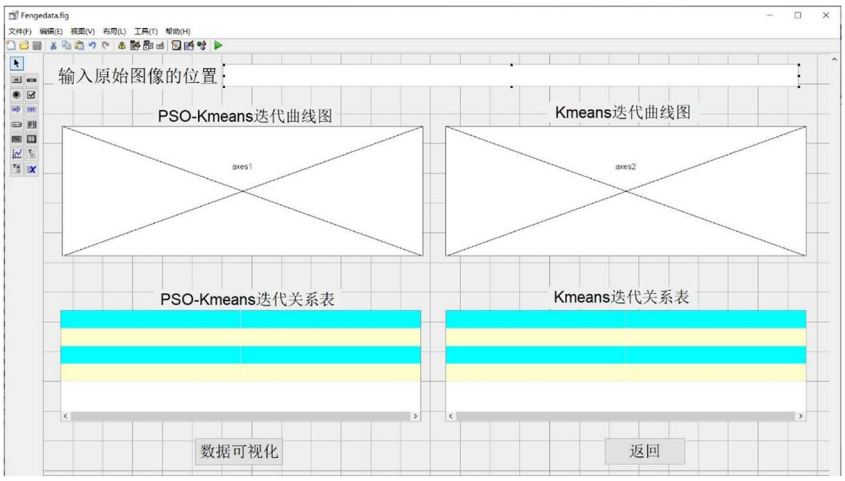

Figure 5. Data visualization interface
图 5. 数据可视化界面

回调函数编写

利用 MATLAB 生成的两个生成 Fengedata.m 和 Fengedata.fig 文件，保存到相应文件路径下，点击并打开 Fengedata.m 文件，找到相应的回调函数位置，对回调函数的代码进行编写：

1) 在“数据可视化”按钮的回调函数 function data_Callback 处编写如下代码：

```
Data = cell(iter,2);
for i=1:iter
Data{i,1}=i
End
for j=1:iter
Data{j,2}=J_cur
set(handles.table1, 'ColumnName', {'迭代次数','目标函数值'}, 'data',Data)
Data = cell(iter,2);
for i=1:iter
Data{i,1}=i
end
for j=1:iter
Data{j,2}=J_cur
end
set(handles.table2, 'ColumnName', {'迭代次数','目标函数值'}, 'data',Data)
axes(handles.axes2);
plot(1:length(J),J,'*b-'), xlabel('迭代次数'),ylabel('目标函数')
run table2;
```

2) 在“返回”按钮的回调函数 function exit_Callback 处编写如下代码：

close Fengedata;

FengeSystem;

3) 保存并运行 Fengedata 数据可视化子界面。

3. 基于 PSO-KMeans 算法的 MATLAB(GUI)图像分割系统实例应用

在基于 PSO-KMeans 算法的 MATLAB(GUI)图像分割系统平台开发的基础上，针对于原始未分割图像“lena.jpg”，选取原始灰度图像作为分割原图像，将基于 PSO-KMeans 算法的 MATLAB(GUI)图像分割系统应用到该图像的分割上，分别利用 KMeans 图像分割算法和于 PSO-KMeans 算法的图像分割算法对“lena.jpg”进行分割，并对结果进行比较。对基于 PSO-KMeans 算法的 MATLAB(GUI)图像分割系统进行应用测试。

1) 第一步：打开基于 PSO-KMeans 算法的 MATLAB(GUI)图像分割系统，并运行 Fengesystem 分割系统主界面函数文件 Fengesystem.m，点击“分割结果”按钮，在 Fengeresult 分割结果子界面的“输入原始图像位置”输入框内输入原始灰度图像分割图片位置，点击“显示结果”按钮。得到分割结果如图 6 所示：

Figure 6. Image segmentation results
图 6. 图像分割结果

从图中的分割结果可以看出，PSO-KMeans 图像分割算法在基于 PSO-KMeans 算法的 MATLAB (GUI)图像分割系统快速而精准的对原始图像进行了分割。并且基于 PSO-KMeans 算法的 MATLAB (GUI)图像分割系统界面简洁，操作简单。

2) 第二步：在第一步的基础之上，点击返回到 Fengesystem 分割系统主界面，点击“参数输出”按钮，在 Fengeparameter 参数输出子界面的“输入原始图像位置”输入框内输入原始灰度图像分割图片位置，点击“输出参数”按钮。原始灰度图像在被分割之后，得到相关图像分割参数结果如图 7 所示，目标函数迭代值由表 1 和表 2 所示：

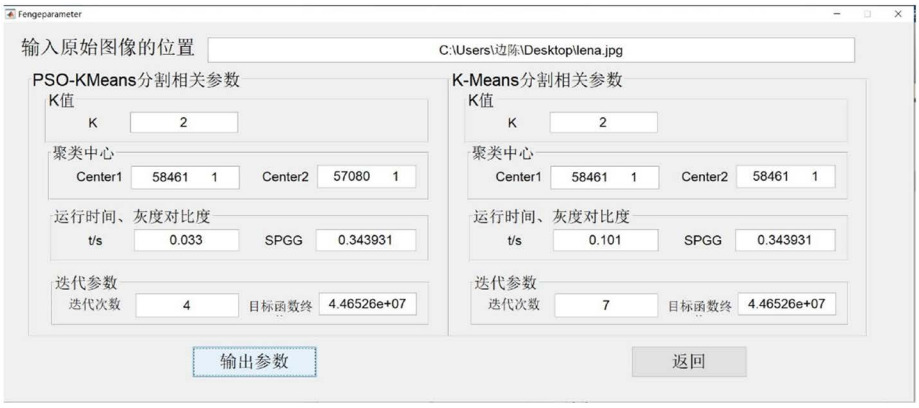

Figure 7. Parameter output interface
图 7. 参数输出界面

从图 7 和表 1、表 2 可以看出，在对基于 PSO-KMeans 算法的 MATLAB (GUI)图像分割系统的简单操作以后，可以更加直观的得到相关分割参数。经过 PSO 算法寻优以后，PSO-KMeans 分割算法对图像分割是 K 值被选取为 2，两个聚类中心点选取为(58461, 1)和(57080, 1)。分割时间为 0.033 s，灰色对比度 SPGG 为 0.343931，迭代次数为 4 次。目标函数的迭代值为 44652618.064838。

Table 1. PSO-KMeans algorithm iteration parameters
表 1. PSO-KMeans 算法迭代参数

迭代次数	目标函数值
001	44845644.023540
002	44678959.824275
003	44652618.064838
004	44652618.064838

Table 2. K-Means algorithm iteration parameters
表 2. K-Means 算法迭代参数

迭代次数	目标函数值
001	49486071.618034
002	46681505.860586
003	45371445.491741
004	44845644.023540
005	44678959.824275
006	44652618.064838
007	44652618.064838

3) 第三步：在第二步的基础之上，点击返回到 Fengesystem 分割系统主界面，点击“数据可视化”按钮，在 Fengedata 数据可视化子界面的“输入原始图像位置”输入框内输入原始灰度图像分割图片位置，点击“数据可视化”按钮。原始灰度图像在被分割之后，得到相关图像分割的数据可视化图表结果如图 8 所示：

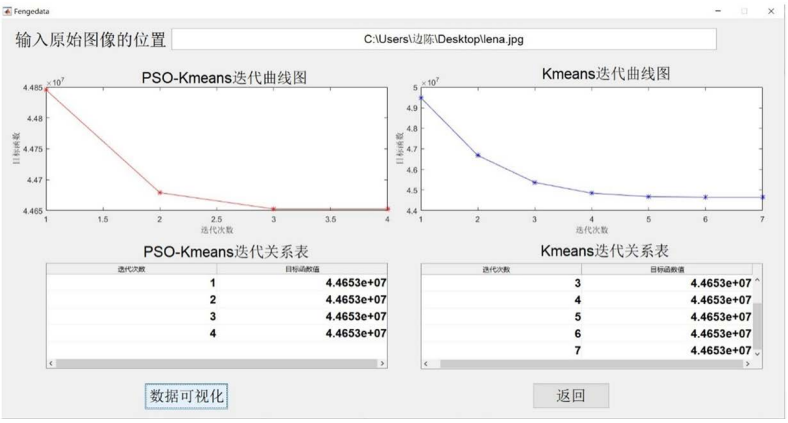

Figure 8. Data visualization interface
图 8. 数据可视化界面

从图 8 可以看出，在对基于 PSO-KMeans 算法的 MATLAB(GUI)图像分割系统的简单操作以后，可以更加直观的得到图像分割的目标函数迭代曲线图和迭代关系表。算法只经过了 4 次迭代就达到了最优值。

4. 总结

本文利用 MATLAB 在图形用户界面开发上的简洁和快捷性以及可视化界面开发方面的优势为基础，通过对 PSO-KMeans 图像分割组合算法理论的研究和实践，开发了基于 PSO-KMeans 算法的 MATLAB

(GUI)图像分割系统平台，该平台不仅可以实现对图像的精准分割，还可以得到分割过程中的产生的相关参数以及图形分割过程中的图表数据可视化展示。并且可以和同类分割算法 KMeans 算法进行分割结果和数据的直观比较，更能体现出新创算法的优越性和高效性。基于 PSO-KMeans 算法的 MATLAB (GUI)图像分割系统平台简单易操作，分割结果和参数的展示简洁明了易读。并且对图像分割的相关结果进行了图表可视化的展示和与同类分割算法进行比较，进一步提高了图像分割的精准性和可视化性。使得图像分割的方法更加简单。未来通过对其他分割算法的研究和对代码的编写，以及对图像分割系统界面的进一步优化设计。本系统将实现多种算法对图像进行分割。进一步提高图像分割的便捷性和分割效率。

基金项目

西南科技大学大学生创新基金项目(项目编号：CX19-061)。

参考文献

[1] 安超, 李磊民, 黄玉清. 一种改进的基于 SLIC 的自适应 GrabCut 算法[J]. 自动化仪表, 2017, 38(10): 17-20.

[2] 李炳宇, 萧蕴诗, 汪镭. PSO 算法在工程优化问题中的应用[J]. 计算机工程与应用, 2004, 40(18): 74-76.

[3] 王爱莲, 伍伟丽, 陈俊杰. 基于 K-means 聚类算法的图像分割方法比较及改进[J]. 太原理工大学学报, 2014, 45(3): 372-375.

[4] 张世勇. 基于混合 PSO 的 K-means 算法及并行化研究[D]: [硕士学位论文]. 重庆: 重庆大学, 2007.

[5] 李磊. 基于 MATLAB GUI 的数字图像处理系统设计[D]: [硕士学位论文]. 成都: 成都理工大学, 2012.

[6] 李念念, 张红梅. 基于 MATLAB GUI 的信号与系统分析软件开发[J]. 工业控制计算机, 2011, 24(3): 19-21.

[7] 刘艳华. 数字调制系统的 GUI 用户界面设计[J]. 科技视界, 2018(2): 150-151.

[8] 黄冉, 杨本硕, 杨德平, 等. 基于 MATLAB/GUI 灰色预测系统开发及应用[J]. 青岛大学学报(工程技术版), 2014, 29(3): 32-37.

[9] 谢秀华, 李陶深. 一种基于改进 PSO 的 K-means 优化聚类算法[J]. 计算机技术与发展, 2014(2): 34-38.