

一种基于随机采样的贴片图 像自动生成算法

用
户
手
册

1.设计说明

1.1 概述

本算法程序包括多个计算模块，主要用于通过对输入设计样本进行解析统计，使用贴片图片生成新的设计方案，

本算法程序由多个 m 文件、一个约束 txt 文件、多个设计样本 txt 文件和多个贴片 jpg 文件组成。可使用 Matlab 程序直接进行运行计算。
(如图 1)

design	文件夹	
facade	文件夹	
newDesigns	文件夹	
.gitignore	GITIGNORE 文件	1 KB
analysisFactor.m	Notepad++ Doc...	4 KB
analysisFactorNS.m	Notepad++ Doc...	5 KB
bestDesign.jpg	JPEG 图像	192 KB
blockSampleSAT.m	Notepad++ Doc...	7 KB
blockSampleSATNS.m	Notepad++ Doc...	7 KB
choseBestDesign.m	Notepad++ Doc...	1 KB
constrain.txt	Notepad++ Doc...	1 KB
dataReader.m	Notepad++ Doc...	1 KB
factorFun.m	Notepad++ Doc...	2 KB
factorFunNS.m	Notepad++ Doc...	2 KB
factorsFun.m	Notepad++ Doc...	1 KB
factorsFunNS.m	Notepad++ Doc...	1 KB
klDivergence.m	Notepad++ Doc...	2 KB
klPreprocee.m	Notepad++ Doc...	2 KB
LICENSE	文件	18 KB
mergeFact.m	Notepad++ Doc...	1 KB
patReplace.m	Notepad++ Doc...	1 KB
patSymAnaly.m	Notepad++ Doc...	1 KB
patSymEachBox.m	Notepad++ Doc...	1 KB
README.md	MD 文件	1 KB
rebuildByMat.m	Notepad++ Doc...	2 KB
symAnaly.m	Notepad++ Doc...	1 KB
symEachTile.m	Notepad++ Doc...	2 KB

图 1 该程序的文件组成（部分）

1.2 编程语言

程序采用 Matlab 编程语言，编程环境是 PC 计算机和 Win7 操作系统，可以在绝大多数版本的 Matlab 上运行。

1.3 运行环境

软件可以运行在大部分普通 PC 计算机上，使用 Windows 系列操作系统。

2. 操作步骤

2.1 修改默认参数

多数默认参数位于 synTiles.m 文件中（如图 2），iteraN 为迭代次数，默认为 30 次，迭代次数设置过小可能会导致无法收敛至全局最优解。tileDir 为贴片图片读取目录，贴片图片须按正整数顺序命名（如图 3）。designDir 为设计样本的读取目录，newDesignDir 为设计方案输出目录，constrPath 为约束文本文件路径。

在 synBlockSS.m 中可以设置 newDesignDir 参数，此参数需与 synTiles.m 中的 newDesignDir 保持一致。pWlakOrig、pFlipOrig、pSmoothOrig、TOrg 为初始迭代参数，不建议用户修改（如图 4）。

```

1  %The Matlab implementation of the paper Synthesis of Tiled Patterns using Factors Graphs
2  %
3  %by Yaoyi Li
4  %Jan 10, 2015
5
6  close all;
7  clear;
8  clc;
9
10 rand('seed', 0);
11
12 iteraN = 30;
13
14 tileDir = './facade';
15 designDir = './design';
16 newDesignDir = './newDesigns';
17 constrPath = './constrain.txt';
18
19 if ~exist(newDesignDir, 'dir')
20     mkdir(newDesignDir);
21 end
22
23 [tileSets, numTile, inPatt, numDesign, constr] = dataReader(tileDir, designDir, constrPath);
24
25
26 %[G, blocks] = analysisFactor(inPatt{1});
27 G = {};
28 blocks = {};
29
30 for cntIn = 2:numDesign
31     %[newG, newBlocks] = analysisFactor(inPatt{cntIn});
32

```

图 2 参数位置 (synTiles.m)

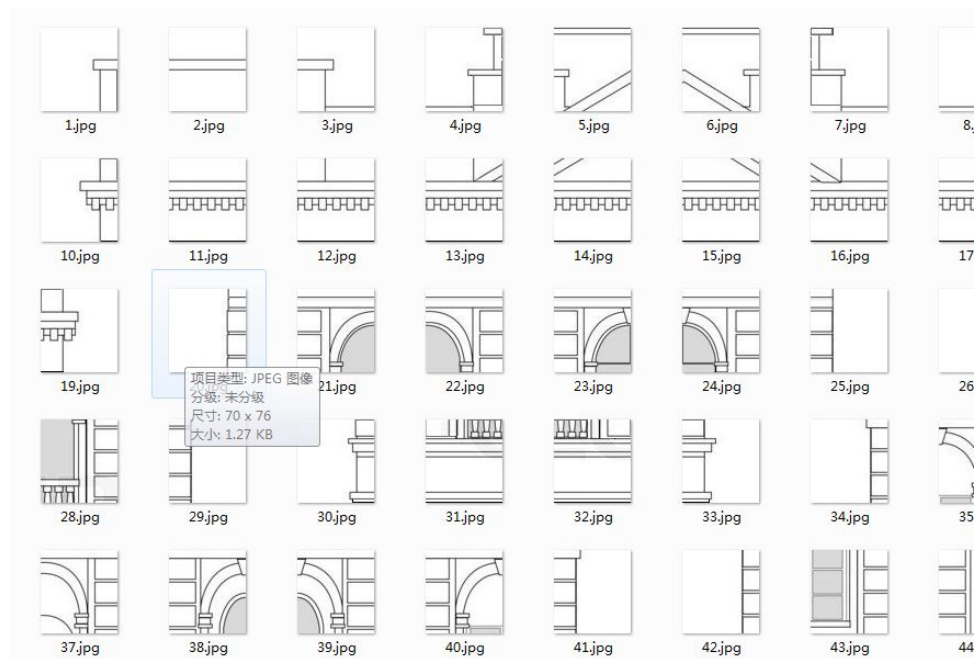


图 3 贴片图片文件

```

% 0          cell of patterns
%
%by Yaoyi Li
%Dec 19, 2014

newDesignDir = './newDesigns';

numTile = numel(tileSets);

designTileNum = 2*sum(sum(constr));

pWlakOrig = 0.8;
pFlipOrig = 0.1;
pSmoothOrig = 0;
TOrig = 0.25;

pWlak = pWlakOrig;
pFlip = pFlipOrig;
pSmooth = pSmoothOrig;
T = TOrig;

[M, N] = size(constr);

```

图 4 参数位置 (synBlockSS.m)

2.2 约束及设计样本设置。

约束文件由一个数字矩阵构成，1 表示该位置有填充贴片，0 表示该位置为空白，外部由 0 表示边界，且约束文件应具备可满足性，对于默认的实验样本可满足性表示为，约束矩阵横向宽度为偶数，纵向高度为 $3n+2$ （图 5）。

设计样本文件应命名为 **design*.txt**，*表示设计样本序号。设计样本文件同样由一个数字矩阵构成，非零数字表示填充在该位置的贴片序号，0 表示该位置为空白，外部由 0 表示边界（图 6）。

```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

图 5 约束文件

```

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 1, 2, 2, 2, 2, 2, 2, 9, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 10, 18, 11, 18, 11, 11, 18, 19, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 20, 21, 22, 23, 24, 21, 22, 25, 0, 0, 0, 0, 0
0, 1, 2, 2, 3, 4, 27, 28, 27, 28, 27, 28, 7, 4, 7, 8, 9, 0
0, 10, 18, 11, 18, 11, 31, 32, 31, 32, 31, 32, 11, 18, 11, 18, 19, 0
0, 34, 40, 35, 40, 35, 40, 35, 38, 39, 40, 35, 40, 35, 38, 39, 41, 0
0, 42, 48, 43, 48, 43, 48, 43, 44, 45, 48, 43, 48, 43, 44, 45, 49, 0
0, 50, 51, 52, 51, 52, 51, 52, 55, 56, 51, 52, 51, 52, 55, 56, 57, 0
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

```

图 6 设计样本文件

2.3 执行算法。

在 Matlab 中将工作目录设置为当前目录，输入 synTiles 运行，即可执行该算法。Matlab 在执行完数据预处理后将显示 preprocessing finished，对于每次迭代将显示迭代序号，并对局部最优解和全局最优解进行提示。执行过程中在设计样本目录下会输出每次迭代产生的设计方案（如图 7），最优结果 bestDesign.jpg 将输出在当前目录。

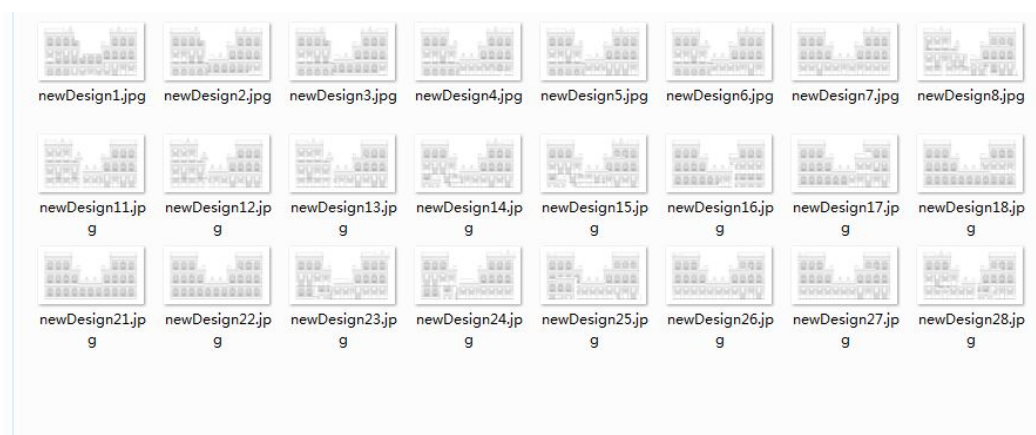


图 7 每次迭代的中间结果