

申请上海交通大学博士学位论文

可视媒体编辑与重用关键技术研究

论文作者 桂彦

学 号 0080339026

指导教师 马利庄教授

专 业 计算机应用技术

答辩日期 2012年09月

Submitted in total fulfilment of the requirements for the degree of Doctor
in Application of Computer Science

Research on the Key Technologies of Visual Media Editing and Reusing

YAN GUI

Supervisor:

Prof. LIZHUANG MA

DEPART OF COMPUTER SCIENCE AND TECHNIQUE, SCHOOL OF ELECTRONIC,
INFORMATION AND ELECTRICAL ENGINEERING
SHANGHAI JIAO TONG UNIVERSITY
SHANGHAI, P.R.CHINA

Sept., 2012

可视媒体编辑与重用关键技术研究

摘要

近年来，随着互联网的快速发展，网络中具有丰富的可视媒体资源。目前，这些海量的可视媒体已经广泛应用到虚拟现实、影视制作、计算机游戏、动画生成等多个领域中。将现有可视媒体素材经编辑与处理后广泛重用于影视动画制作中，这不仅能提高可视媒体素材的利用率和使用水平，而且还在很大程度上降低了制作成本。可视媒体素材的高效重用是以研究可视媒体的编辑工具集为出发点，相应的，对可视媒体素材的深度开发、及时处理和有效利用，一方面大大增加了可视媒体素材的数量，为影视动画制作提供更加丰富的可视媒体资源；另一方面通过对可视媒体素材的复用处理，可以提高影视动画创作与生成的效率。

围绕可视媒体编辑与重用技术中的若干热点问题，本文通过分析纹理的内在结构和利用图像/视频中的纹理特征对2D纹理合成、流体动画模拟和灰度图像彩色化等方面展开了相应地研究，并实现了相关算法。论文工作取得了以下创新性研究成果：

(1) 提出了一种基于特征描述的快速纹理合成方法。基于块的纹理合成技术是从给定的样本纹理中生成一幅新的纹理，但是这种方法存在的问题是由于不精确的相似性度量导致块重叠区域出现结构的误匹配。在本文中，我们提出一种拓展的度量机制测量结构的相似性，该方法在样本块采样过程中能够找到最佳匹配块。我们首先引入层的概念通过给每个像素赋权值描述像素的关联性，然后构建方向场捕获纹理特征内在的方向和布局。实验结果显示，该方法能够显著地减轻已合成纹理中结构特征的不连续。

(2) 提出了一种基于基元分布的周期纹理合成和设计方法。由于人的视觉敏感于结构和周期性特征，因此，我们通过使用基元的分布进行纹理合成。给定一样本纹理，纹理分析过程是分割纹理成单个的基元，以及在样本纹理中检测每一个基元以至于通过建立连通性分析基元之间的相邻关系；纹理合成过程是在用户指定的画布上通过重组已分割的基元产生一个新的、更大的纹理图

像，即基于已构建的基元间的相邻关系合成新的2D基元分布。实验结果显示，本文提出的方法擅长于合成近似规则的纹理，并能很好地保持其潜在的规律性。

(3) 提出了一种基于样本视频编辑的单幅图像的流体动画合成。通过全局地传递样本视频的流体运动，我们从单幅静止图像中合成连续的流体动画序列。给定一幅具有似流场景的目标图像，alpha遮片用来表示图中的似流体区域。我们需要调整用户给定的样本视频，从而产生符合这个感兴趣区域的流体运动。基于构建的流体视频库，我们通过变形视频帧的光流场来获得目标图像的流场信息，此时，该视频帧与目标图像具有非常相似的场景。由此，我们自动地分配流场的方向向量和速度值。实验结果已经证明，本文提出的方法能够在合成的动画中很好地保持较大尺度的流体特征，且能够生成高质量的连续流体动画。

(4) 提出了一种基于可并行图像分割的灰度图像彩色化及相关应用。给定待上色的灰度图像，我们首先采用监督分类算法建立每一像素点与已分割样本彩色图像中各区域的对应关系，从而为各像素点分配颜色。结合基于空间关联的滤波，我们仅保留具有较高置信度值的像素点的颜色值，这个即为灰度图像的调色板。此外，我们采用基于纹理的可变水平集图像分割方法并行地抠取图像中各具有相似特征的区域，从而快速、精确地确定各区域的边界。根据这些分割的区域，我们可以采用多种上色方法进行颜色传递。我们还将本文提出的可并行图像分割方法应用于图像中具有重复场景元素的抠取，并使用替换、重排等操作对图像进行编辑。

关键词： 2D纹理分析与合成 视频纹理 流体动画合成 灰度图像彩色化 图像编辑

Research on the Key Technologies of Visual Media Editing and Reusing

ABSTRACT

In recent years, with fast developing of the internet, there are abundant resources of image/video. At present, these vast amounts of images/videos have been widely used in the fields of virtual reality, film making, computer games, animation creation. It was widely reused in film, television and animation production by editing and processing the existing image/video materials, which not only improves the level and utilization of image/video, but also greatly reduces the costs. The efficient reusing of image/video is based on the research for constructing the image/video's editing toolsets. Accordingly, the deeply developing, rapidly processing, effectively using of image/video materials thus are meaningful. On the one hand, it can increase the amount of image/video resources for film, television and animation production. On the other hand, it improves the efficiency of film, television and animation production by the reusing the image/video resources.

This dissertation focuses on some issues of image/video editing and reusing, and we study some key issues about 2D texture synthesis, fluid animation simulation, and colorization by analyzing the inherent structure of the texture and using the texture feature of image/video. And these related algorithms were implemented. The main contributions are summarized as follows:

(1) A fast texture synthesis method based on feature description is proposed. Patch-based texture synthesis techniques from a small sample texture generate a new texture image efficiently, but one significant problem is that these existing texture synthesis algorithms suffer from structure mismatching across patch boundaries due to inaccurate similarity measure. We propose a desired distance metric to measure structural similarity when especially handling textures with salient features, which has the ability to find best matches during the patch-based sampling. To do this, we introduce

the concept of "layers" to describe the correlation of pixels through computing each pixel weight, and construct the orientation fields to capture the inherent direction and variation of texture features through computing the dominant local orientation angle and the degree of anisotropy of each pixel. Our experimental results demonstrate that the proposed method can significantly alleviate structure discontinuities in synthesized textures.

(2) A texture synthesis and design method is proposed. Since human perception is sensitive to structure and periodicity, we thus perform texture synthesis by taking into account the distribution of texels. Given a sample texture, the analysis procedure consists in segmenting texture into individual texels, and detecting each texel in order to analyze their neighborhood relationships by constructing connectivity. Then the synthesis process consists in reproducing a new large texture directly on a user-specified canvas by recomposing segmented texels, which synthesizes two-dimensional texel arrangements based on the previously constructed neighborhood relationships of texels. Results show that the method especially deals with the near-regular textures, which well preserves underlying structural regularity.

(3) A method for fluid animation from a single image by editing video examples is proposed. We perform synthesizing animation from a single image by transferring fluid motion of a video example globally. Given a target image of a fluid scene, an alpha matte is required to extract the fluid region. Our method need to adjust a user-specified video example for producing the fluid motion suitable for the extracted fluid region. Employing the fluid video database, the flow field of the target image is obtained by warping the optical flow of a video frame that has a visually similar scene with the target image according to their scene correspondences, which assigns fluid orientation and speed automatically. The results demonstrate that our method is successful in preserving large fluid features in the synthesized animations.

(4) A method for grayscale image colorization by using parallel image segmentation and the related applications is proposed. Given a grayscale image to colorize, we construct the relationships between pixels in the input images and the regions extracted from a segmented example image by using a robust supervised classification scheme, assigning color values for these pixels. Combined with spatial filtering with

the spatial coherence, the colors of pixels with highest confidence values are retained. This is provided as color palette to the grayscale image. In addition, the regions with similar features in input image can be segmented in parallel by using texture based variational level set method. It can fast and accurately determine the boundaries of the regions. With the segmented regions, various colorization techniques can be applied to replace colors. We also apply this parallel image segmentation for repeated scene elements, and edit images by using replacement and rearrangement operations.

KEY WORDS: 2D texture analysis and synthesis, video texture, fluid animation, grayscale image colorization, image editing

目 录

摘要	i
ABSTRACT	iii
目录	vii
插图索引	xv
表格索引	xv
主要符号对照表	xvii
第一章 绪论	1
1.1 研究背景与研究意义	1
1.2 相关技术研究现状综述	2
1.2.1 二维纹理合成技术	2
1.2.2 视频纹理合成技术	11
1.2.3 图像/视频彩色化技术	16
1.3 研究内容和拟解决的关键性问题	20
1.3.1 研究内容	20
1.3.2 拟解决的关键性问题	21
1.4 本文创新工作和结构安排	22
第二章 基于特征描述的纹理快速合成	25
2.1 引言	25
2.2 基于块的纹理合成算法框架	26
2.3 纹理特征描述	29

2.3.1	基于层间关联的判别模型	29
2.3.2	基于方向场的特征匹配	32
2.3.3	基于纹理特征描述的距离公式定义	34
2.4	基于结构变形的纹理子块拼接	36
2.5	实验结果与分析	39
2.6	本章小结	44
第三章	基于基元分布的周期纹理合成与设计	45
3.1	引言	45
3.2	周期纹理分析与建模	47
3.2.1	基元的提取	47
3.2.2	基元检测、定位及分类	52
3.2.3	基元空间分布建模	56
3.3	基于基元分布的纹理合成	57
3.3.1	纹理合成算法概述	57
3.3.2	基元分布的扩张	59
3.3.3	最佳匹配基元	60
3.3.4	基元的放置	62
3.3.5	种子的局部增长	63
3.4	基于基元的纹理设计	65
3.4.1	基于简笔画的基元分布生成	65
3.4.2	基于薄板样条的基元变形	67
3.5	实验结果与分析	69
3.5.1	纹理合成结果与分析	69
3.5.2	纹理设计结果与分析	72
3.6	本章小结	74
第四章	基于样本视频编辑的单幅图像的流体动画合成	75
4.1	引言	75

4.2	基元样本视频的流体动画合成系统	76
4.2.1	现有流体动画合成系统分析	76
4.2.2	流体动画合成系统概述	77
4.3	样本视频的调整	79
4.3.1	流体区域的抽取	79
4.3.2	基于细缝的视频缩放	81
4.4	样本视频的分析	82
4.5	流场的估计	85
4.5.1	流体视频库的构建	85
4.5.2	基于SIFT流的运动传递	85
4.6	流体动画合成	87
4.6.1	平均图像的估计	87
4.6.2	残差序列的合成	89
4.6.3	视频重构	90
4.7	实验结果与分析	91
4.8	本章小结	94
第五章	基于可并行图像分割的灰度图像彩色化和相关应用	95
5.1	引言	95
5.2	基于样图的调色板的生成	96
5.2.1	基于伽伯小波变化的纹理特征分析	96
5.2.2	特征空间和分类	97
5.2.3	调色板的生成	99
5.3	基于并行可变水平集的图像分割	100
5.3.1	传统水平集 (Level Set) 方法	101
5.3.2	初始轮廓的放置	102
5.3.3	可变水平集表达式	103
5.3.4	边界检测函数定义	105
5.3.5	数值求解框架	106

5.4 颜色传播	107
5.5 实验结果与分析	108
5.5.1 灰度图像彩色化实验结果与分析	108
5.5.2 具有重复场景元素的图像抠取与编辑	110
5.6 本章小结	113
第六章 总结与展望	115
6.1 论文工作总结	115
6.2 未来工作展望	116
参考文献	119
致谢	133
攻读学位论文期间发表的学术论文目录	135

表格索引

2-1 多种纹理合成方法性能比较	39
4-1 流体动画合成的相关信息	94

插图索引

1–1 纹理和图像的差异	6
2–1 基于样图的纹理合成方法的缺陷	26
2–2 这基于块采样的纹理合成过程	27
2–3 基于块的采样策略	28
2–4 纹理子块边界区域的层次划分	30
2–5 合不同距离公式下的合成结果比较	32
2–6 编织纹理方向场的构建	33
2–7 参数 λ 对合成结果的影响	35
2–8 结构变形	38
2–9 与基于像素点的纹理合成和的比较（一）	40
2–10 与基于像素点的纹理合成和的比较（二）	41
2–11 与基于块采样的纹理合成方法和规则纹理合成方法的比较	42
2–12 与图像缝合纹理合成方法和基于图割的纹理合成方法的比较	42
2–13 与基于外观空间的纹理合成方法和基于特征匹配的纹理合成方法的比较	43
2–14 纹理合成结果	43
2–15 失败的例子	44
3–1 基于基元分布的纹理分析与合成	46
3–2 自动的纹理图像分割	49
3–3 交互的基元抽取	51
3–4 形状上下文描述子	53
3–5 基元检测和分类	56
3–6 基元间连通关系的构建	57
3–7 基元间连通关系的扩张	60

3-8 邻域比较过程	61
3-9 种子间冲突检测	64
3-10 基于简笔画的基元分布生成	66
3-11 第二类简笔画及其扩张	67
3-12 细薄板样条映射	68
3-13 本章方法与基于图割的、图像缝合的和近似规则纹理合成方法的比较	69
3-14 本章方法与基于特征的纹理合成技术的比较	70
3-15 样本基元分布对纹理合成的影响	71
3-16 更多纹理合成结果	71
3-17 纹理设计结果	72
3-18 与多种经典纹理合成方法的比较	73
3-19 与基于基元分布的纹理合成方法的比较	74
4-1 现有流体动画合成系统	77
4-2 全局特征保持的流体动画合成系统	78
4-3 目标图像的抠图结果	80
4-4 基于图割的细缝缩放	82
4-5 样本视频的调整	82
4-6 平均光流场和残差的抽取	84
4-7 部分流体视频	85
4-8 流体运动的自动传递	87
4-9 平均图像的估计	88
4-10 基于直方图匹配的外观调整	91
4-11 与现有流体动画合成系统的比较	92
4-12 单幅图像的流体动画合成	92
4-13 具有不连通流体区域的单幅图像的流体动画合成	93
5-1 基于相似光照值和纹理模式统计的KNN匹配	99
5-2 自动生成的调色板	100

5-3 区域分割和彩色化结果	107
5-4 与现有彩色化方法的比较	108
5-5 基于纹理特征分类的灰度图像彩色化	109
5-6 基于并行分割的灰度图像彩色化	110
5-7 更多彩色化结果	110
5-8 重复场景元素抠取	111
5-9 更多重复场景元素抠取结果	111
5-10 与现有交互式图割方法的比较	112
5-11 重复场景元素的编辑	113

主要符号对照表

I	图像
N_1, N_2	邻域
p, q	像素点
B_i, P	纹理子块
M, N	纹理子块大小
ψ	经验直方图
$p(\cdot)$	概率
$d(\cdot), D(\cdot)$	距离函数
$\delta(\cdot)$	狄拉克函数
ω	权值系数
θ	角度
$F(\theta)$	幅度
ρ	连贯度
f	特征点
$g(\cdot)$	特征点集
L_2	2-范式
TSE	总平方误差
q_n	颜色量化值
C_n	簇
g	图
$E(\cdot)$	能量函数
F, B	前景和背景
$\sum \omega$	二阶矩
A	线性变换矩阵
h_i	极对数直方图

k_1, k_2	特征值
δ_E	变化误差
δ_I	颜色值差异
X, Y	位置坐标
e^{ref}	参照基元
e^{tar}	目标基元
$\omega(e^{ref})$	参照基元的邻域
$\omega(e^{tar})$	目标基元的邻域
Ω	区域
(yl, yr, xl, yr)	尺度参数
(Nly, Nlx, Nry, Nrx)	区域参数
ov	空隙参数
S_{ref}, S_{tar}	偏移值
TPS	细薄板样条
$\phi(\cdot)$	径向基函数
α_i, β_i	未知参数
α	alpha遮片
w	窗口
$\omega := (u, v)^T$	光流场
R_i, R'_i	残差序列
D_i, D'_i	差异序列
$W(\cdot)$	变形函数
\vec{x}, \mathbf{x}	像素点位置
$\vec{\nu}$	流场向量
F_i	视频序列
$g(x, y)$	二维伽伯函数
$G(u, v)$	二维伽伯函数的傅立叶变换
$\mathbb{W}_{m,n}(u, v)$	伽伯小波变换
$\mu_{m,n}$	均值

$\sigma_{m,n}$	标准差
T	纹理特征向量
S_W	类内分离矩阵
S_B	类间分离矩阵
$conf(\cdot)$	置信度
$\phi(t, x, y)$	水平集函数
$sing(\cdot)$	符号函数
τ	迭代步长
YUV	YUV颜色空间

第一章 绪论

1.1 研究背景与研究意义

视觉是人类认知世界最重要的手段。通过视觉，人类感知外界物体的大小、明暗、颜色、动静，至少有80%以上的外界信息经视觉获得，包括图像、视频、数字几何等，且称之为可视媒体。近年来，随着互联网的快速发展，网络中提供了丰富的可视媒体资源，并且这些信息呈现几何级增长趋势。针对网络可视媒体素材规模十分庞大的特点，充分挖掘这些海量素材信息并将这些信息应用到虚拟现实、影视特效、计算机游戏、动画生成等多个领域中，有助于推动我国影视动漫、互动娱乐等数字内容产业的发展。

为了在影视动画创作与设计的制作过程中提高对可视媒体素材的使用水平和利用率，将现有可视媒体素材经编辑与处理后广泛重用于影视动画制作中，这在改善影视动画制作环境的同时，也很大程度上降低了制作的成本。目前，国家科技部和文化部非常重视和关注可视媒体素材的编辑与重用技术的研究，同时也得到了学术界和产业界的重视，如每年的重大国际会议（Siggraph, Siggraph Asia, Eurographics, Pacific Graphics等）上都有相关的成果发表与展示，这些成果也被成功地应用于制作影视动画、动漫、游戏等的大型商业软件中（如Maya, 3dsMax, Photoshop等），由此带来的理论和技术上的突破将对海量可视媒体素材的高效利用产生深远影响。

可视媒体素材的高效重用是以研究可视媒体的编辑工具集为出发点，包括了以研究图像/视频为主的合成技术、分割技术、抠图技术及彩色化技术等的可视媒体编辑工具。相应的，对可视媒体素材的深度开发、及时处理和有效利用，一方面大大增加了可视媒体素材的数量，为影视动画制作提供更加丰富的可视媒体资源；另一方面通过对可视媒体素材的复用处理，不仅可以实现如战争场面、自然景观、历史面貌等高质量影视动画场景虚拟再现的效果，而且还可以提高影视动画创作与生成的效率。纵观可视媒体编辑与重用技术在影视动画制作中的相关应用，在这些工作中如下的几个方面非常具有代表性：

纹理合成相关技术已被广泛应用到三维动画制作中，特别是用于绘制大规模复杂场景，创作人员经常使用纹理在粗糙的几何模型上表达表面几何细节和

光照细节，从而达到增强模型表面的视觉效果。自上世纪末，研究人员提出了使用纹理合成技术来解决自动生成真实感绘制所需的大量纹理的问题。该技术基于给定的小区域纹理样本，按照表面的几何形状，拼合生成整个曲面的相似纹理。其主要的优点在于：避免繁琐的手工纹理绘制，保证纹理的真实性，以及表现几何模型无法体现的细节。除了大规模复杂场景生成外，该技术还在真实感和非真实感绘制、虚拟现实，以及游戏等领域具有广泛的应用前景。

自然景物模拟一直是计算机图形学和虚拟现实领域的研究热点。海浪、瀑布、烟雾、火焰、雨雪、扬尘等现象在真实世界中是频繁出现的，因此在电脑游戏、数字电影、虚拟漫游以及科学计算可视化场景中，都需要绘制出高视觉质量的这类特殊效果。当前，Maya、3dsMax等专业的建模软件都针对这类特效提供相应的建模功能。另一方面，通过采用基于视频纹理的合成方法，它对一段长度有限的视频进行分析，将原始视频中的某些片断重新组合，得到无限连续的动态效果，合成效率高。如基于视频纹理的火焰生成，可以逼真地模拟火焰的模糊边界、火焰产生的烟雾效果、物体爆炸时的情形等。

可视媒体编辑工具中必然涉及到颜色处理技术。近年来，可视媒体的颜色处理技术已渗透到相关的软硬件系统设计中，其中专业级的如Adobe Photoshop、CoralDraw、GIMP、Smoke等；普通级的如Windows操作系统中的Paint工具等。其中相关彩色化技术是当前计算机图形学与计算机视觉领域的研究热点例如：将彩色化技术应用于黑白动画片和漫画上色，可以快速实现对轮廓清晰的动态变化的前景层和相对较静态的背景层的着色。彩色化技术在推动漫发展的同时，也在其领域中起着越来越关键的作用。

本文得到了国家重点基础研究发展计划（973计划）“可视媒体的智能交互与处理的理论与方法（2006CB303100）”和国家863目标导向类项目“交互式高效动画创作技术与系统（2009AA01Z334）”的资助。本文结合可视媒体素材高效编辑和重用的思想，拟对纹理/视频纹理合成、流体动画合成、灰度图像彩色化其中的关键技术进行研究，希望能为广大用户提供具有较强功能、性能优越、使用方便的可视媒体编辑工具。

1.2 相关技术研究现状综述

1.2.1 二维纹理合成技术

在计算机图形学中，纹理是增强物体表面细节的重要手段。根据表面纹理

生成方法的不同，这里主要探究二维纹理合成技术，主要包括过程纹理合成 [1–12] 和基于样图的纹理合成 [13]这两类关键技术的研究发展现状。

1.2.1.1 过程纹理合成

过程纹理合成(**Procedural Texture Synthesis**, PTS)通过对纹理的物理生成过程进行模拟来生成新的纹理。过程纹理中应用最广的是以Perlin noise [2, 3]为代表的分形噪声技术。Perlin从噪声带中构建纹理模式，可以通过噪声函数在空间的任意位置产生一个数值，该数值可以直接映射为颜色值，也可以用来对规则的图案进行空间扰动。该方法简单实用，无需任何的存储量或预算算，可以广泛用于生成图形学中的各种视觉效果，如云、火、水、大理石、木纹、岩石等。然而，利用Perlin noise生成纹理容易导致走样和细节丢失问题。Cook等人通过对Perlin noise是由一组多分辨率的带限函数构成的观察，提出使用小波噪声（Wavelet Noise）函数 [4]来避免上述问题。小波噪声是近似于带限函数的噪声，它能够很好的保持细节并且最大程度地抑制走样的产生。

基于反应扩散方程(Reaction-Diffusion Equations)，Witkin和Kass [5]通过允许各相异性和空间非均匀的扩散改进传统的反应扩散系统，从而模拟具有斑点和条带状的动物皮毛。反应扩散是一个在模型表面以不等速率进行两个或更多物质扩散，并相互反应后形成如斑点及条带状模式的过程。Turk [6]则证明了使用反应扩散系统模拟纹理模式可以直接在多边形表面进行，避免映射纹理坐标到复杂曲面的过程，实现无缝的拼接。相同的，Walter等人 [7, 8]通过构建符合生物学模式的图案生成模型来生成哺乳动物的皮毛纹理。

Fleischer等人 [9]在基于反应扩散系统的基础上，结合粒子系统和动力模型的特性，提出利用生物学细胞模拟过程直接在模型表面上产生几何纹理，如鳞状、刺状和羽毛状的纹理模式，且称之为细胞纹理（Cellular Texture） [10]。该方法采用3D细胞粒子来构建模型表面的几何纹理，能够表现比传统的纹理映射技术更加丰富的几何细节，而且比由手工制作的几何模型更加便利。细胞粒子模拟器通常计算与几何纹理相关的位置、方向、颜色、大小等属性值，这些信息将作为形状和外观的参数用于绘制最终的视觉效果。

Miyata [11]提出了一种仅需要给定少量参数的纹理生成器来模拟石头墙纹理。该纹理生成器描述的石头墙模式是由“节点&连接（node & link）”所构成的网格状模型。其中，节点用来描述位置和连接数据，而由节点包围的多边形

封闭区域表示石块占据的空间。除模拟石墙纹理以外，该方法还可以生成人行道和台阶等纹理模式。石墙的另一显著特征是受自然环境的影响，如风蚀和矿物质的分解等，而发生外观的变化。Dorsey等人 [12]提出一种方法建模和绘制石头外观及形状的变化。该方法利用风化模型来模拟湿度流动、传输和物质分解等现象，还可以控制表面材质的腐蚀程度。为了绘制出半透明、色素沉着等视觉效果，采用蒙特卡罗光线跟踪方法模拟石头内部的光线散射。该方法能够捕获广泛的时间依赖的石头外观效果，因此能够有效地建模大理石、花岗岩和砂岩等表面外观的变化。

1.2.1.2 基于样图的纹理合成

近十年来，基于样本的纹理合成已经取得了显著的进步。其目标在于：根据用户给定的样本纹理，自动生成一张具有任意尺寸的新纹理，并且与样本纹理具有相似的外观特征。与过程纹理合成相比，基于样本的纹理合成只需一个样本纹理作为输入，避免了繁琐的参数调试，使用起来更为方便。按照所选纹理模型的不同，基于样本的纹理合成可以分为参数化的纹理合成和非参数化的纹理合成。由于非参数的纹理合成不仅适用于各种类型的纹理，从随机纹理到结构纹理，且能产生高质量的输出纹理结果，因而受到越来越多研究人员的关注。

参数化的纹理合成方法（**Parametric Texture Synthesis**）利用概率统计模型建模纹理，模型的参数由给定样本纹理的统计特征信息计算得到。在合成时，通过对参数确定的纹理模型进行重采样来生成新的纹理。Heeger和Bergen [14]提出了一种基于金字塔的纹理分析和合成方法。在纹理分析过程中，通过采用多尺度多方向的线性滤波直方图对输入纹理进行分析和表示，即构建样本纹理和噪声图像的多级可控向金字塔（**Steerable Pyramid**），此时，纹理分解为一组空间频率子带（**sub-band**）。在纹理合成过程中，通过采用直方图匹配来更改噪声纹理可控向金字塔的各个子带，使其和样本纹理具有相同的直方图，最后根据更改过的噪声纹理控向金字塔重建出合成结果。该方法能够很好地处理随机纹理，但是对结构性强的纹理不能获得高质量的输出结果。

De Bonet [15]也提出了一种基于多分辨率的线性滤波器的纹理合成方法。该方法首先建立样本纹理的拉普拉斯金字塔（**Laplacian Pyramid**），把低分辨率各层金字塔的滤波响应作为相似性约束条件。在合成时，采用自顶（低分辨率）向下（高分辨率）的顺序对样本纹理金字塔的各层分别进行采样，在较低

分辨率的样本纹理中选择相似的纹理特征，然后在较高分辨率的合成纹理中保持这些结构信息。与Heeger等人 [14]的工作相比，该方法考虑了金字塔中不同分辨率之间的相关性，能够很好地捕获样本纹理的结构特征。

Portilla和Simoncelli [16, 17]也提出一种基于多分辨率框架的纹理合成方法，通过小波变换后计算的统计信息来参数化纹理的统计模型，对样本纹理构建多尺度多方向的可控向金字塔。在采样过程中，同样按照自顶（低分辨率）向下（高分辨率）的顺序更改初始噪声纹理的可控向金字塔，并在每一层金字塔的低通和带通频率中添加统计约束，同时重构低分辨率的噪声纹理。最后，根据更改的噪声纹理的可控向金字塔生成新的纹理。由于该方法考虑了多尺度多方向可控金字塔之间的关联性，使得复杂的采样过程无法保证合成算法的有效收敛。

非参数化的纹理合成技术则采用马尔科夫随机场 [18] (Markov Random Field, MRF) 建模纹理是一个局部的、静态的随机过程。也就是说，纹理中的每一像素点与其空间上相邻的像素点具有相同的特性。我们用一个实验来说明马尔科夫模型隐含的这些特性（参见图1.1）。给定一幅图片，仅允许观察者在一个可移动的窗口内观察图像。当窗口移动时，观察者可以看到图像中不同的部分。如果说图像具有静态的特性，那么在移动窗口下的图像区域其外观看起来总是相似的；如果说图像具有局部的特性，那么每个像素点的颜色值仅依赖于其相邻的像素点，与图像中的其他区域内的像素点无关。基于MRF模型，我们只需根据局部邻域信息直接对样本纹理进行采样，就可以合成新的纹理。邻域的大小是用户指定的唯一参数，且与纹理模式中的特征大小成比例。由于这些局部的、静态的假设，在样本纹理和合成纹理之间的局部邻域相似性能够保证具有相似的外观。

Efros和Leung [19]最先提出使用非参数方法进行纹理合成，该方法是一种基于像素的纹理合成方法 (**Pixel-based Texture Synthesis**)。在合成过程中，首先从给定的样本纹理中任选小块区域作为初始种子区域 (seed region)，然后以从内到外的方式逐像素点扩充种子区域，从而生成新的纹理。通过采用邻域搜索过程确定每一次输出像素点的候选集，然后从候选集中选取任一邻域的中心像素点作为当前最佳匹配像素点。输出纹理的合成质量与所选邻域的大小有关。此外，合成过程中，由于邻域形状每次搜索时都可能发生改变，很难对搜索过程进行加速，因此合成速度很慢。

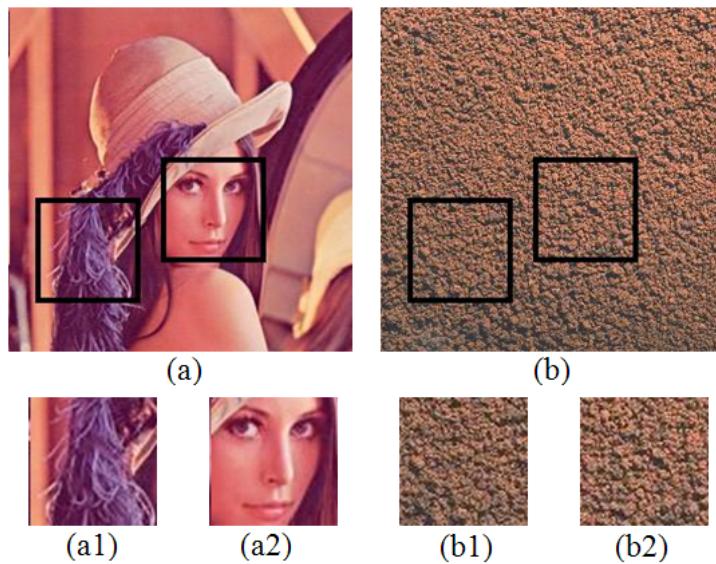


图 1-1 纹理和图像的差异 [13]

Fig 1-1 How textures differ from image [13]

Wei和Levoy [20]采用固定邻域大小的搜索方法，并以扫描线的顺序进行纹理合成。使用固定L形邻域的优势在于：一方面，可以使用多分辨率框架进行纹理合成，使得它能够用较小的邻域捕获大尺度的纹理特征；另一方面，可以使用树结构向量量化（Tree-Structured Vector Quantization, TSVQ）加速像素点搜索过程，对邻域进行聚类。使用这些方法加快了纹理合成算法的速度，极大地提高了算法的效率。然而，使用TSVQ加速会缩小搜索空间又或引起搜索不到最佳匹配，从而导致数据丢失，降低了纹理合成的质量。

Ashikhmin [21]通过改进Wei和Levoy [20]提出的纹理合成方法，对自然纹理，如花、鹅卵石、树叶、灌木等，获得了更好的合成效果。基于空间相关性（Spatial Coherence）理论：在纹理合成过程中，在样本纹理中汇集的像素在拷贝到输出纹理时也趋向于汇集的，而不是散落在输出纹理的任意位置上。该方法根据待合成像素点其邻域内每一像素点在样本纹理中的位置确定其候选集，最佳匹配的像素点即为候选集中具有最相似邻域的像素点。该方法利用空间关联性缩小了像素搜索空间，提高了算法的合成效率。然而，该方法不能很好地处理具有水平结构特征的纹理。借鉴相同的思路，Zelinka和Garland [22]提出的基于“跳跃图（Jump Map）”的纹理合成方法，以及Tong等人 [23]在提出的基于双向纹理函数的曲面纹理合成方法中描述了一种更适用的K-相关

(k-coherence) 搜索机制，加快了纹理合成速率。

Hertzmann等人 [24] 把基于像素的纹理合成方法推广用于图像编辑，提出了一个被称之为“图像类推 (Image Analogies)”的框架，可以处理如图像滤波、超分辨率、纹理传递等任务。图像类推的目标是：给定一组训练图像 A 和 A' ，以及一幅目标图像 B ，以类似于图像 A 和 A' 之间的滤波关系合成与目标图像 B 对应的滤波图像 B' 。在合成时，该方法基于YIQ的光照-颜色空间的特征向量进行相似度匹配，能够更好地描述和捕获训练图像中的特征，从而对大部分纹理都能取得较好的效果。然而，该方法需要对每个像素点进行两次最佳匹配搜索，合成速率较慢。

为了提高基于像素的纹理合成方法的合成效率，利用GPU图形加速技术进行纹理合成的方法受到了研究者们的关注。Lefebvre等人 [25] 提出了一种并行可控的纹理合成方法，该方法在输出纹理中存储的是像素在样本纹理中的坐标值，而不是像素的颜色值。该方法在GPU上依次执行上采样、抖动、修正三个步骤，通过多遍渲染计算出新纹理中每个像素的坐标值。该方法还利用PCA对每个邻域进行降维，以及使用k-coherence方法进行邻域搜索，因此提升了纹理合成的效率，可以达到实时的合成速度。然而，基于像素的合成方法很难捕获大尺度的纹理结构信息，因而该类方法不能很好地处理具有显著结构的纹理。

基于块的纹理合成 (**Patch-based Texture Synthesis**) 在合成过程中每次输出一个纹理子块而不是一个像素点，因而该类方法不仅是在合成质量上还是合成效率上都有很大的提升。Chaos mosai [26] 是较早的一种基于块的纹理合成方法。该方法首先从样本纹理中选取一定数量的且形状规则的矩形小块，然后将这些纹理子块粘贴到合成纹理的任意位置上，通过采用猫映射(Cat Map)来重新分布这些纹理块，以及在纹理块之间的过渡区域处进行模糊处理 (Blurring)，使得纹理块之间产生平滑、自然的过渡，最终生成新的纹理。然而，该方法不能很好处理具有水平方向和垂直方向结构特征的砖墙纹理，在合成结果中通常会出现明显的不连续、杂乱结构特征。Praun等人 [27] 通过使用不规则形状的纹理子块来改进Chaos mosai方法，然而，这增加了算法的复杂性以及降低了算法的合成速度。除此之外，该方法仍然存在结构特征不对齐的问题。

Efros和Freeman [28] 提出了一种被称之为“图像缝合 (Image Quilting)”的纹理合成方法。该方法以纹理块的边界区域为邻域，通过比较待合成纹理中当前邻域与样本纹理块的邻域之间的相似程度，以光栅扫描顺序每次拷贝一个

最佳匹配样本纹理块到待合成的纹理中。然后，采用动态规划算法（Dynamic Programming）计算纹理块之间重叠区域上的最小误差边界割（Minimum Error Boundary Cut, MEBC），从而对相邻的纹理块进行缝合。Liang等人 [29]则采用羽化方法（Feathering）[30]合并纹理块之间的重叠区域，其主要的原因是能够在纹理块之间产生更平滑的颜色变化。另一方面，该方法采用优化的KD-树[31, 32]和主元分析（PCA）方法[33]加速了基于块搜索的采样过程，这使得算法可进行实时的纹理合成。虽然采用该类方法可以避免在使用基于像素的纹理合成方法时产生的具有大面积颜色块的垃圾区域，但是这些方法在合成过程中容易产生结构不连续和纹理块重拷贝的现象。

Kwatra等人 [34]提出一种基于图割（Graph-Cut）的方法能够把新的纹理块缝合到已合成的纹理区域。该方法对纹理块的形状和大小没有限制，所选择的纹理块与预先计算的误差区域完全重合后，通过采用基元最大流/最小割（Max-flow/Min-cut）[35, 36]的图割方法计算划分重叠区域的最优缝（seam），实现对添加纹理块的合并。此外，该方法使用快速傅立叶变换（Fast Fourier Transforms, FFT）加速块搜索的过程。区别于动态规划方法计算最小代价割，基于图割的方法适用于任意维数上的合成，即除二维的图像纹理合成外，还可以应用到三维空间上进行视频纹理合成（Video Texture Synthesis）。该方法在算法效率上优于前两种基于块的纹理合成方法。

此外，Kwatra等人 [37]将合成纹理作为一个整体，提出了一个基于全局优化的纹理合成方法。该方法通过定义衡量样本纹理和输出纹理之间相似性的全局能量函数，将纹理合成问题转化为一个能量优化问题，通过使用期望最大算法（Expectation Maximization, EM）迭代地对纹理能量函数进行优化，从而合成新纹理。该方法提出的能量函数形式简单，通过添加各种约束来改变能量函数可以取得不同的合成效果。此外，该能量函数还能够捕获大尺度的纹理结构信息，因而对大部分纹理都能产生很好的结果。

另一类则是基于拼接的纹理合成方法（**Tile-based Texture Synthesis**），该类方法通过对样本纹理直接拼接来产生新的纹理，是生成大量纹理最为简单而快速的方法。在该类方法中，直接对样本纹理进行拼接，结果中可能产生明显的结合缝；当样本纹理被多次使用时，结果中则会出现严重的周期性重复。针对这个问题，研究者们试图找到一组可以相互拼接的样本纹理块进行组合，使得合成结果中可以明显减少拼接块的周期性重复，且能提高算法的合成速度。

在许多已有的拼接方法 [38–41] 中，Grunbaum和Shephard [42] 提出使用16个拼接块就可以铺满整个平面，同时保证生成的结果没有周期性重复。Culik等人 [43] 进一步证明发现，仅使用13个拼接块就能够产生绝对的非周期性模式。

Wang Tiles [44, 45] 是图形学中应用最广的一种纹理拼接技术，它是由一组同大小的正方形组成的拼接块集合，且每条边都分配一种颜色进行标记。这些拼接块根据边的颜色进行排列，要求在相邻拼接块之间的邻接边具有相互匹配的颜色。使用Wang Tiles不仅能生成大面积的复杂纹理模式，而且还能有效地避免产生拼接块的周期性重复。Stam [46] 是最早提出使用Wang Tiles进行纹理合成，他使用16个纹理拼接块来生成无明显重复的新纹理。然而，该方法没有给出一般纹理的Wang Tiles构建方法，以及没有讨论如何快速进行拼接块的组合。Neyret和Cani [47] 使用三角形纹理块直接在模型表面上进行拼接，从而生成没有周期性重复的纹理图案。尽管该方法合成速度很快，但在拼接过程中需要旋转三角形纹理块，从而不适合方向性强的纹理。Cohen等人 [48] 提出了一个简单的随机系统，通过使用一组从样本纹理中自动构建的Wang Tiles来生成非周期性的新纹理。该方法从样本纹理中随机取出4个菱形纹理块，把它们分别放置到拼接块的4条边上，然后使用Image Quilting [28] 方法把4个菱形纹理块缝合起来生成最终的拼接块。然而，由于Wang Tiles技术没有考虑对对角位置的拼接块进行约束，因此在拼接块的角上会出现不连续的现象。针对这个问题，Lagae等人 [49] 在拼接块的角上进行颜色标记，从而通过引入对角相邻约束对对角拼接块进行有效组合。

上述基于像素的纹理合成方法和基于块的纹理合成方法都是使用马尔科夫随机模型建模纹理，通过比较邻域之间的相似性对样本纹理进行采样从而产生新的纹理。除此之外，纹理也可以用其特征进行描述。近年来，基于特征的纹理合成方法 (**Feature-based Texture Synthesis**) 已经成为研究热点并成功地用于建模具有显著结构特征的纹理。Zhang等人 [50] 提出一种基于纹元模板 (Texton Masks) 的方法能够直接在模型表面上进行纹理合成。该方法使用纹元模板来建模局部纹理的变化，包括描述纹理元素的尺度、方向、颜色和形状等。此外，基于纹元模板的合成算法还能够有效地保持纹理元素的完整性。Wu和Yu [51] 通过同时考虑纹理的几何特征和颜色提出一种新的纹理合成算法。该方法通过特征匹配和结构对齐进行结构的相似性度量，以及采用拉普拉斯滤波方法抽取样本纹理的结构图 (Feature Map) 来引导新纹理特征图的合成，从而在合成结果中尽可能的减少产生不连续的结构特征。Lefebvre等人 [52] 则

将样本纹理描述为一个低维的欧式外观空间（Appearance Space），即由空间邻域、纹理特征和径向变换组成的外观特征向量。该方法通过使用该特征空间能够用于非等轴纹理合成、表面纹理合成以及纹理对流等，极大地提高了合成结果的质量。Dischler等人 [53]提出了一种结构化的纹理合成方法，侧重于用结构的分布（Arrangement Map）描述纹理模式。该方法通过抽取纹理特征构建相对于样本纹理的多边形网格（Texture Mesh），然后在几何空间中进行纹理合成，从而可以达到实时的、帧率的合成速率。很明显，上述这些方法依赖于抽取纹理特征来保持结构的连续性。然而，当样本纹理的特征是低对比度又或是模糊不清时，通常很难提取到这些纹理特征。

利用上述纹理合成方法用于合成近似规则的纹理（Near-regular Textures）时，合成结果并不能很好地保持结构的规律性。Liu等人 [54]提出一种近似规则纹理合成（**Near-Regular Texture Synthesis**）框架，即通过构建近似规则纹理与其相对应的规则分量之间的变形场（Deformation Field），可将近似规则纹理的合成过程看成是对规则纹理的一次变形。该方法采用用户辅助的方式抽取近似规则纹理中潜在的弯曲网格（Lattice）来描述变形场的三个分量，包括几何变形场、光照变形场和颜色变形场。另一方面，构建的变形场可以用于指导合成过程中纹理的形变，同时也被看成是纹理且可以进行相应的纹理合成操作。因此，该方法可以通过控制变形场来合成具有不同视觉外观效果的近似规则纹理。这之后，Liu等人 [55]又提出了一种针对几何结构规则而颜色分布不规则的近似规则纹理的合成算法。根据近似规则纹理具有的平移对称特性，该方法可以找到纹理模式潜在的网格状结构并定位每一纹理拼接块。由此，样本纹理被划分成具有多边形形状的多个纹理子块。该方法通过重复拼接这些纹理块来保持样本纹理中所具有的结构规律性和颜色多变性。此外，该方法采用动态规划和图像混合技术处理纹理块之间的重叠区域进行纹理块的缝合。可以看出，该类方法对具有近似规则/规则结构特征的纹理能产生高质量的合成结果。

近年来，随着基于样本纹理合成技术的深入研究，研究者们根据纹理是由纹理元素（Texture Elements）按照一定随机性规则通过不断重复而构成的特性提出了基于纹理元素分布的纹理合成新方法。Dichier等人 [56]利用颜色量化方法将构成纹理的基本元素提取出来，称之为纹理粒（Texture Particle），通过分析纹理粒子在样本纹理中的空间分布，然后根据该空间分布重新组织这些纹理粒子进行纹理合成。文献 [57]和文献 [58]也使用了类似的方法，先分析样本纹理中元素的空间分布，然后参照样本纹理中纹理元素的空间分布重新组织这些

纹理元素，从而生成新的纹理。

1.2.1.3 存在的核心问题

通过概括现有二维纹理合成技术中的相关重要方法，这些方法在进行纹理合成时存在如下几个核心问题：1) 过程纹理合成方法已不再是纹理合成的主流方法，主要原因是这类方法仅适用于合成某些特定的纹理类型，如动物的皮毛、石墙等。此外，由于这类方法是通过模拟纹理的生成模式来产生新的纹理，因此，针对某类纹理构建的纹理模型是不通用的，降低了合成的效率。2) 虽然基于样本的纹理合成技术已经获得了长足的发展，但是这类技术中已提出的诸多纹理合成方法仍然无法产生高质量的输出结果。也就是说，在使用这些方法处理结构显著的纹理时，这些方法仍然无法避免纹理结构的无匹配，从而在合成的纹理中通常出现不连续结构特征的瑕疵。3) 尽管现有部分基于纹理元素的方法提出了建模纹理的新思路，但是这类方法通常没有考虑不同类型纹理元素之间的空间相邻关系，以及在重组纹理元素没有考虑其位置、形状对纹理合成的影响。

1.2.2 视频纹理合成技术

近年来，研究者们开始尝试利用视频纹理来产生自然逼真的动态场景，如瀑布、火焰、烟雾等。目前，和视频纹理比较相近的媒体主要有时序纹理、时变纹理、动态纹理等，它们从不同的角度对视频纹理的特征进行了描述。通过定义某种准则描述样本视频中潜在的运动规律，从而合成与样本视频相似的动态场景。本文把近年来已发表的并与本文研究工作关系紧密的视频纹理合成研究工作归纳为两个主要方面的内容：动态纹理合成和基于视频纹理的流体动画合成。

1.2.2.1 动态纹理合成

动态纹理是按时间顺序排列的用于描述某种动态景观的一组图像序列，具有稳定的、运动形态重复的特征。采用时间序列分析方法合成动态纹理主要是采用参数模型对所观测的有序随机数据进行分析与处理，从而捕获动态纹理的时空特性。Szummer和Picard [59, 60]采用时空自回归模型（Spatio-temporal Autoregressive Model, STAR）建模动态纹理的图像序列。STAR模型要求动态

纹理数据呈多模态高斯分布且具有宽平稳特征。然而，由于图像序列中的非均匀光照，使得动态纹理数据是非平稳的，此时需要采用均值滤波平滑图像序列。另一方面，为简化模型参数的估计，需要假设相邻空间是一种半球或立方体结构。在确定了像素相邻空间结构后，该方法采用最小二乘（CLS）准确地估计模型参数。该方法使用STAR模型能够很好地合成水蒸气、沸腾的水和河流等动态纹理。然而，该方法不能很好地处理具有旋转的运动（如马桶中的水流），这主要是违背了平稳特性的假设。

Bar-Joseph等人 [61]基于多分辨率分析（Multi-Resolution Analysis, MRA）提出了一种统计学习算法用于合成时变纹理。在分析阶段，输入的样本时变纹理被看成是随机过程中产生的3D信号，且使用小波变换将3D信号构建成一颗MRA树。在构建MRA树时，采用由二维控向金字塔小波变换和一维小波滤波器变换的笛卡儿乘积组成的三维小波变换对时变纹理信号进行层次分解。在合成阶段，对样本MRA树中具有相似条件概率的路径进行学习和重采样，通过合并样本时变纹理的多棵MRA树构建新的随机MRA树。最后，对新构建的MRA树进行小波逆变换生成新的时变纹理。该方法对流动的瀑布、游走的鱼群、飘动的云彩、走动的人群等时变纹理进行了合成，合成的效果良好。

Doretto等人 [62]将动态纹理看作是时间轴上的平稳随机过程，通过采用线性动态系统（Linear Dynamic System, LDS）分析和合成动态纹理。该模型用一个状态转移方程和一个输出方程描述动态纹理的时间统计特性，通过采用最大似然估计及奇异值分解的近似方法求解模型的各个参数。该方法通过采用LDS模型对视频样本中重复出现的动态特性进行学习，得到它们的表示形式，并根据估计的模型参数合成有限或者无限长度的新视频。LDS模型还采用了线性PCA方法从高维的图像中构建低维的状态矢量，从而能够以帧速率实时合成动态纹理。然而，由于LDS模型的状态转移方程和输出方程均是线性的，在合成许多非线性的或复杂的动态纹理时可能出现抖动、结构特征模糊等问题。针对这个问题，Yuan等人 [63]提出了一种闭环的线性动态系统模型（Closed-loop LDS）用于建模动态纹理。在CLDS模型中，通过引入反馈系统控制视频纹理合成过程，从而降低LDS模型中出现的不稳定状态。尽管该方法在合成质量上有了明显地提高，但是使用CLDS模型仍然只能在线性空间中建模动态纹理的场景运动。因此，对于如风中摇摆的旗帜和树枝、波涛汹涌的浪花等动态纹理仍然无法产生高质量的输出结果。

Ruei-Sung Lin等人 [64]提出采用全局坐标的非线性流形（Globally Coordinated nonlinear Manifold）建模这些动态纹理中复杂场景的运动。该方法首先采用非线性降维（Nonlinear Dimensionality Reduction, NLDR）算法将高维的动态纹理数据用低维的参数形式（即流形）进行描述，然后通过求解NLDR来构建非线性流形中的坐标系。对于每一动态纹理，该方法将动态纹理中的场景运动信息在流形的全局坐标系中描述为一条低维的连续轨迹，通过对每一条学习的轨迹进行重采样并将其映射到图像空间中，从而合成具有复杂场景的动态纹理。

基于相同的思路，Liu等人 [65, 66]则提出同时采用外观模型和动态模型建模动态纹理中时序序列的外观变化和形状变化。在外观模型中，该方法使用局部线性PCA混合对非线性流形进行参数化；在动态模型中，该方法采用全局的坐标模型来构建每一混合分量的坐标与全局流形坐标之间的映射，使得克服运动的不连续问题。通过使用这两个模型，该方法能够很好地处理在形状和外观上具有较大变化的动态纹理，产生高质量的合成结果且可达到实时的合成速率。

Wang等人 [67, 68]提出一种生成模型建模动态纹理中物体的随机运动，如飘落的雪花、翻滚的波浪、舞动的草和飞翔的鸟群等。在合成过程中，该方法首先将视频序列中的每幅图像表示成一组伽柏基（Gabor bases）或傅立叶基（Fourier bases）在不同尺度和方向上的线性叠加。这种从原始图像到基描述的变换能够实现数据降维的目的。然后，采用自回归（AR）动态模型建模物体的运动以及物体之间的相互运动，从而捕获物体在运动过程中产生的结构变形和运动轨迹。该方法采用统计梯度算法来估计生成模型的参数，通过从学习的模型中合成与样本视频序列具有外观相似性的随机运动序列。然而，该方法忽略了物体运动的细节，如在合成飞翔的鸟群时并没有考虑鸟儿翅膀的运动。

区别于上述基于统计模型的视频纹理合成方法，Schodl等人 [69]提出了一种基于图像的视频纹理合成方法，即通过重复利用视频中的图像数据产生新的视频。该方法以基于 L_2 距离的马尔科夫链模型（Markov Chain Model）计算视频中任意两帧之间的相似性。在视频合成过程中，通过在给定的视频样本中找到足够相似的任意两帧，并在这两帧之间进行一次跳变，从而合成任意长度的视频。同时，为了消除跳转帧之间的视觉不连续，该方法采用颜色混合技术（blending）和变形技术（morphing）来产生自然的切换。该方法成功合成了钟

摆、蜡烛的火焰、飘扬的旗帜、瀑布、脸部运动等动态纹理，且能够进行实时合成。然而，对于任意两帧存在较大差异的动态纹理，如麦浪、海岸波浪等，合成效果不佳。基于相同的思路，Schodl等人 [70] 扩展上述方法用于分析和控制视频中随时间变化的前景物体的运动，通过重复利用相似的运动来合成新的视频动画。

除此之外，也有部分视频纹理合成方法是直接将二维纹理合成技术扩展到三维空间来构建动态纹理。如方法 [20] 中将原算法中的二维实体替换为相应的三维体数据，像素的邻域也由二维扩展到三维，合成结果以帧为单位进行输出。在TSVQ算法的加速下，每一帧的合成大约需要20秒。方法 [34] 则将视频序列看成是3D时空子块，通过采用图割方法计算两个3D时空子块之间的最优缝，实现时空子块之间的无缝拼接，避免了方法 [69] 中使用混合技术和变形技术导致的特征模糊。

1.2.2.2 基于视频纹理的流体动画合成

利用计算机模拟流体动画效果一直是计算机图形学和虚拟现实领域的研究热点之一。已有的模拟流体动画的方法主要分为两类：一类是基于物理的流体模拟方法；另外一类是基于视频的合成方法。基于物理的流体模拟方法 [71–73] 能够模拟和控制产生多种流体效果，如表现自然场景中水、烟和火等。然而，这类方法需要为特定的流体类型设置合适的参数，计算量大，难以达到实时要求。而基于视频合成的方法，由于直接处理视频中的图像数据，因而具有合成效率高的特点。针对本文的研究工作是以动态纹理为处理对象，即通过分析和编辑视频样本的流体运动信息来合成连续的流体动画效果，如下主要阐述现有基于视频的流体动画合成相关关键技术。

Sun等人 [74] 提出了一个由视频驱动动画的系统（Video-Input Driven Animation, VIDA）。该系统首先从给定的样本视频中抽取物理参数，如风速、水波高度等；然后利用这些参数合成真实场景中物体的运动效果。该方法将物理参数的抽取过程看成是一个逆谐波振荡问题（Inverse Harmonic Oscillation），即以观测物体的运动为振荡器的位移来估计振荡系统的参数，包括频率、阻尼系数以及引起这些振荡运动的外部推动力。该方法采用傅立叶级数展开方法来估计振荡系统参数。该方法能够很好地模拟风和水波这两种自然现象，然而，该方法受视频分辨率的约束，对于远距离的运动物体，由于不能很好地捕获运动物体的细节，从而无法进行精确的参数估计。

Bhat等人 [75]提出以流线来标记视频中连续的流体模式，通过分析和建模流线上纹理子块的运动，从而在其他给定的流线上合成相似的且任意长度的流体动画序列。该方法对自然现象，如瀑布、河流等，通常描述成沿着固定流线的粒子运动，这些粒子用纹理子块表示且随着运动发生外观的变化。在粒子动态特性分析中，该方法用一个表示时间-位置关系的矩阵描述流线上粒子的运动过程。已知当前帧中某一位置的粒子，通过沿着流线搜索下一帧中最匹配的粒子来确定新的输出帧中粒子的位置。最后，通过采用羽化方法处理纹理块之间的重叠区域来产生最终的流体动画序列。此外，使用该方法对样本视频进行编辑可以改变流体的外观。

Lin等人 [76]提出一个允许用户从有限的高分辨率图集中合成自然逼真的流体动画序列的系统。对于给定的一组图集，该系统首先构建图用于连接图集中相似的图像。然后，通过采用时间顺序恢复算法从构建的图中自动的寻找子图集中图像的排列顺序。这些顺序为动画序列的合成提供了参照的动态特性。在合成过程中，该方法利用二阶马尔科夫链采样在构建的图中选取图像来生成动画序列。为消除合成帧之间的视觉不连续，该系统还采用基于光流的双向插值方法来产生平滑的过渡。然而，对于有重叠的物体运动，该方法需要更多的图像序列来捕获物体在不同阶段的运动状态。

Chuang等人 [77]提出使用随机运动纹理（Stochastic Motion Texture）驱动单幅图像中似流的区域来合成流体动画。他们约束场景中存在被动的元素（passive elements），包括树、水、云、漂浮的船等，能够在自然力的作用下产生一定形式的运动。该方法首先采用图像抠图算法 [78]将图像划分成层次结构。为了模拟动态特性，该方法采用频谱方法合成随机运动纹理，这些运动纹理是具有时变特征的2D位移图。最终，通过将这些动态纹理添加到分割的图像区域中，并采用图像融合方法对各层进行无缝拼接，最终产生动画序列。虽然该方法通过使用基于物理的随机运动纹理生成方法来模拟和控制流体的运动，但运动的质量不如直接将动态视频中运动传递到图像中。

针对这个问题，Okabe等人 [79]将视频样本中的流运动信息直接添加到单幅静止图像中来生成连续的流体动画序列。在视频分析阶段，视频样本被分解为光流场和残差序列。其中光流场用于描述视频样本的全局动态信息，而残差序列则表示光流场不能捕获的非传递高频信息。在视频合成阶段，该系统需要为目标图像提供等价的信息，包括与目标图像对应的流场及残差序列。此外，

对于目标图像的各个子块，在样本视频的光流场和目标图像的流场中进行最佳匹配搜索，从而将样本视频的残差体块拷贝到目标图像中合成对应于目标图像的残差序列。然而，当图像中的场景较复杂时，流场的设计需要使用大量的人工交互来描述场景的细节，工作量繁琐。

Okabe等人 [80]在上述工作的基础上借助预先构建的视频库来驱动单幅静止的图像产生流体动画。也就是说，该方法致力于从视频库中选择与目标图像相似的样本视频，然后将其分配给目标图像的各子块，通过无缝拼接后产生最终的流体动画序列。该方法以较少的工作量来合成高质量的流体动画，如不再需要计算残差序列以及进行残差的传递。然而，该方法仍然存在如下不足：一方面，该方法很难保持目标图像的外观细节，这主要是由于在视频库中很难找到与目标图像具有非常相似外观的样本视频；另一方面，该方法需要划分样本视频且以局部的方式分配视频子块，由于丢失了全局信息而引起错误的分配，导致无法保持大尺度的流体特征结构。

1.2.2.3 存在的核心问题

通过概括现有视频纹理合成技术中的相关重要方法，这些方法在进行视频纹理合成以及通过编辑视频纹理进行流体动画模拟时仍存在以下几个核心问题：1) 对不具有时序平稳特性的视频纹理进行合成时，即视频的场景存在快速移动的物体，合成的新视频纹理中通常存在运动不连续、物体结构特征模糊等问题。2) 通过编辑视频纹理模拟流体运动时，现有方法通常依赖于辅助的流场信息来描述特定环境下流体的运动状态，这需要提供大量的用户交互。3) 而在传递视频纹理的运动信息过程中，由于采用的方法是通常局部的块的拷贝与拼接，因此在合成的流体动画中仅能保持时序上的连续性而不能有效地保持空间上的一致性，这将导致产生不连续的视觉效果。

1.2.3 图像/视频彩色化技术

灰度图像彩色化（Colorization）是Wilson Markle于1970年提出的一个术语，当时这种技术被用来为阿波罗登月计划的黑白视频着色 [81]。现有灰度图像彩色化的方法可以粗略的分为两类：一类是基于用户标记的彩色化方法，通常需要用户在待上色的灰度图像中指定颜色线条来获得理想的结果；另一类是基于样图的彩色化方法，通常需要提供彩色图像作为参照图像，通过传递颜色信息来实现彩色化的效果。

1.2.3.1 基于用户标记的彩色化技术

早期的彩色化过程通常需要将图像序列分割成多个区域并跟踪这些区域来添加颜色信息。由于分割和跟踪过程的不稳定性，因此，彩色化过程需要相当多的用户交互，是一个繁琐、费时且代价高的工作。针对这个问题，Levin等人[82]提出了一种交互式的彩色化方法。基于在时空域上具有相似灰度的相邻像素也应该具有相似颜色的假设，Levin等人把彩色化过程转化为最小二乘优化问题。上色之前，需要用户预先在待上色的区域内部标注少量的颜色信息。根据这些颜色约束，通过在时空域上自动传播指定的颜色信息到图像/视频中的其他像素点，从而产生彩色化的图像和视频。然而，对于具有复杂场景的图像和视频，该方法需要用户标注较多的颜色信息且很仔细地进行标记。此外，在区域的边界处容易引起明显的颜色溢出。为此，Huang等人[83]在方法[82]的基础上增加了自适应边界检测处理。通过抽取可靠的边界信息，该方法能够有效地阻止彩色化过程在物体边界处产生的颜色溢出。

上述方法最大的不足是需要付出较大的计算代价来获得高质量的彩色化结果，为此，Yatziv等人[84]提出一种基于测地线距离和色度混合的快速彩色化方法。由于灰度信息的几何特性能够描述图像的几何结构，认为两点之间的测地线距离越小其色度上也就越相似。该方法通过构建图来计算图像中任意一像素点到颜色条上像素点之间的测地线距离，然后采用Dijkstra算法求解这两像素点之间的最小代价路径。一旦确定了某待上色像素点的所有最小代价路径，以每条路径的测地线距离为权重，该方法通过色度混合计算为该像素点快速、准确地分配相似的颜色。相比于方法[82]中需要花几分钟的时间彩色化一幅图像，该方法仅需要不到一秒的时间，同时还能保持产生高质量的彩色化结果。

Qu等人[85]提出一种新的彩色化技术用于处理黑白漫画。通常，人们在黑白漫画中会大量使用线条、阴影线、半色调等模式来增强内容的视觉感，但这也使得利用现有基于灰度连续的彩色化方法不能很好地处理这些灰度不连续的模式。为此，Qu等人提出的方法中采用伽伯小波变换（Gabor Wavelet Transform）的特征统计方法来度量纹理模式的连续性。根据用户在感兴趣的区域内标注的颜色信息，通过采用基于纹理模式的水平集方法来进行区域的分割。使用该图割方法，初始的轮廓在容易产生纹理模式变化的区域边界处停止传播，同时也能很好地分割边界不封闭的区域及具有相似纹理模式的不连通区域。一旦完成了区域分割，该方法可采用多种彩色化技术对黑白漫画上色，包

括进行灰度连续区域防漏的彩色化、能够保持模式结构的模式连续区域的彩色化和影子模式区域的彩色化。

Luan等人 [86]提出一个交互式系统对具有复杂场景的自然图像进行彩色化。在该系统中，彩色化过程主要分为颜色标记和颜色映射两个阶段。在颜色标记阶段，该方法引入了一个同时考虑灰度连续和纹理相似约束的能量优化框架。通过分配颜色标记给图像中的每一像素点，将自然场景图像分割成多个连贯的区域。颜色标记过程仅需要用户在图像中提供少数颜色标记，极大地减少了用户输入。在颜色映射阶段，对于图像中的每一连贯的区域，该系统选择少数几个具有明显差异亮度值的像素点并分配相应的颜色，同时还提供实时反馈，使得更换颜色产生不同风格的彩色化效果。

1.2.3.2 基于样图的彩色化技术

基于用户标记的彩色化方法除了需要用户在灰度图像/视频中标注颜色线条外，还需要用户为每一颜色线条很仔细地选择理想的颜色。由此，该类彩色化方法需要大量的人工交互，使得产生自然、逼真的彩色化效果。为减轻用户的输入，另一类彩色化方法则通过从一幅彩色的参照图像中选取颜色信息，此时，参照图像与待上色的灰度图像具有相似的场景。

Welsh等人 [87]利用Reinhard等人 [88]的颜色转移思想提出了一种自动的灰度图像彩色化方法，通过匹配灰度图像与参照图像中的亮度分量来传递对应的颜色值。在该方法中，一旦灰度图像中的某个像素在参考图像中找到了匹配的像素时，在灰度像素的亮度值不变的情况下，将匹配像素的颜色信息传递该灰度像素。在颜色传递阶段，首先需要将灰度图像和参考图像均变换到 $l\alpha\beta$ 颜色空间，并使用跳跃采样（jittered sampling）在彩色图像中选择像素子集作为颜色样本；然后，对灰度图像的每个像素以扫描顺序进行处理，根据邻域统计量在彩色参照图像中选择与当前灰度像素的最佳匹配像素；最后，将匹配像素的颜色值（ α 和 β 通道）传递给灰度像素，最终产生彩色化的图像。

Chen等人 [89]结合采用灰度图像抠图技术和颜色转移技术对灰度图像进行彩色化处理。该方法首先采用贝叶斯抠图算法[]将具有复杂场景的灰度图像分割成不同区域以及抽取对应的遮片（ α 通道）。然后，该方法对抽取出的每一图像区域选择对应的参照颜色样本，通过采用Welsh等人[]的基于颜色转移的灰度图像彩色化方法对每一区域进行基于直方图平移和比例缩放的着色处理。最

后，通过结合已提取的 α 遮片，将着色后的各区域合成为最终的彩色图像。该方法的贡献在于将图像抠图技术运用到了灰度图像彩色化，从而能够很好地处理具有容易引起混淆光照分布和细微边界的自然图像。

Sykora等人 [90, 91]提出一种无监督的黑白卡通视频彩色化算法。预处理阶段，该方法以被分割的两幅图像为输入：第一幅图像为彩色参照图；第二幅图像则包含待着色的区域。根据第一幅图像的颜色分配情况，该算法通过基于特征颜色转移方法为第二幅图像的各个区域进行着色。在彩色化过程中，该算法首先从待处理图像中抽取轮廓线上的高曲率点以及交叉点为特征点，并采用后向映射的方法对这些特征在彩色参照图中进行基于块的结构相似性匹配。当待处理图像中的每一特征块在参照图中找到最佳匹配的像素块后，就将相应颜色块拷贝到待处理图像的对应位置。最后，采用非最大值压缩方法为每一区域分配颜色。当不同颜色被用于相似结构模式时，基于块的采样策略有可能会失效，该算法则采用概率松弛方法来进行结构匹配调整。

Irony等人 [92]提出了一种新的基于样图的灰度图像彩色化方法，即从已分割的样本图像中传递颜色信息。该方法首先将用户给定的彩色参照图像分割成多个不连通的区域，并为每一区域分配相应的标记。以彩色参照图像的光照通道为训练数据，并根据已分配的标记进行有监督的学习，从而为待上色图像中的每一像素点分配相应的颜色值。在该过程中，该方法采用离散余弦变化计算训练数据集中以每一像素点为中心的子块的纹理特征向量，并采用线性辨别分析（Linear Discriminant Analysis）降维方法构建表示彩色参照图像的低维特征空间。根据这个特征空间，该方法采用K最邻近邻域分类方法（K-Nearest-neighbors, KNN）为待上色的灰度图像中部分像素点分配相应的颜色。此时，以这些着色好的点作为约束，采用方法 [82]将颜色传播到其余的像素中，从而得到最终的彩色图像。

尽管彩色参照图像的使用有助于减轻彩色化过程的工作量，但是彩色化结果的质量很大程度地依赖于所选择的参照样本。对此，Liu等人 [93]提出一种基于多幅彩色参照图像的灰度图像彩色化方法，能够很稳定地处理待上色图像与参照图像之间具有不同光照条件下的彩色化问题。该方法首先从多幅彩色参照图像中恢复光照独立的反射图像，在这个计算过程中需要分解每一参照图像为反射分量（反射率）和光照分量（阴影）。按照相同的方式，该方法计算得到目标灰度图像的反射分量和光照分量。此时，彩色反射图像的颜色信息被传递

给灰度反射图像中具有较高置信度反射分量的像素点。这个部分着色的反射分量可等价为方法 [82] 中由用户标注的颜色线条，通过采用方法 [82] 将颜色传播到其余像素的反射分量，并通过添加目标图像的光照分量来产生最终的彩色化效果。

Chia 等人 [94] 提出了一个基于 web 图像数据为参照样本图像的彩色化系统，使得产生更加可靠、自然真实的彩色化结果。该方法中，用户需要使用待上色图像中的已分割的前景以及对应的语意标注信息作为系统输入。利用这些语意标注信息为分割的前景物体在互联网中搜索大量图像，通过进一步采用图像滤波方法来选择与灰度图像具有最佳匹配特征的图像为彩色参照图像。从这些滤波后的参照图像中，该系统采用基于图优化的方法传递对应前景和背景物体的颜色。由于使用滤波方法选择的参照图像与目标灰度图像具有最一致的特征空间分布，这使得颜色传递过程更加可靠。此外，通过替换不同的参照图像能够产生多样的彩色化结果。

1.2.3.3 存在的核心问题

通过概括现有灰度图像彩色化技术中的相关重要方法，这些方法在为灰度图像着色时仍存在如下几个核心问题：1) 对场景复杂灰度图像上色时，基于用户标注的彩色化技术通常需要进行大量的用户输入，即在图像中特定的区域内标记合适的颜色信息。此外，无丰富经验的用户很难为准确地分配合适的颜色，这有可能无法获得满意的彩色化效果。2) 全局优化的颜色传递方式容易在物体边界处产生颜色溢出、混合等问题，特别是对于模糊的或者分辨率低的边界轮廓。3) 结合图像分割的彩色化方法最终产生的上色效果通常依赖于已分割区域的质量，以及图像分割操作和上色过程通常是独立的，这极大了降低了上色过程的效率。4) 基于样图的彩色化技术通常需要用户预先选定与待上色图像具有相似的场景参照彩色图像，这种约束使得上色后的效果通常是单一的且依赖于参照彩色图像中是否具有丰富的颜色信息。

1.3 研究内容和拟解决的关键性问题

1.3.1 研究内容

可视媒体的编辑与重用一直是计算机图形、计算机视觉中的研究热点，本文结合实际承担的课题项目，研究内容主要涵盖三个方面的相关技术，包括二

维纹理合成与设计、视频纹理及流体动画合成、灰度图像彩色化。通过研究这些关键技术并解决其中存在的问题，力图构建有效的可视媒体编辑与重用工具集。

(1) 研究基于特征描述的纹理合成方法，包括研究纹理像素空间的相关性，用于描述连续的颜色空间分布；研究纹理方向场的构建方法，用于反映纹理结构特征的空间布局。在此基础上，通过结合纹理的相关性分析和纹理的方向场，研究设计度量纹理块之间相似性的距离公式；针对纹理块之间的重叠区域，研究基于结构变形的纹理块缝合方法，实现纹理块之间的平滑过渡。

(2) 研究基于基元分布的周期纹理合成和设计方法，包括研究周期性纹理的特征提取方法，分析该类纹理的基本组成单元（基元）的大小、形状、位置和类别；研究分析基元之间的空间相邻关系，即获取样本纹理的拓扑结构。根据参照的拓扑结构，研究拓扑结构的自增长过程，从而获得与样本拓扑结构具有相似外观的新的基元的分布；研究基元与基元的重组方法，主要包括基元变形和基元之间重叠区域的处理；根据用户定义的纹理生成模式，研究周期性纹理设计方法。

(3) 研究全局特征保持的单幅图像流体动画合成方法，包括通过抽取给定目标图像中的流体区域，研究动态视频缩放方法；通过构建场景的对应关系，研究流场自动生成方法；研究一种全局的流体运动自动传递方法，使得能够保持大尺度的流体特征以及保持时空连续性。

(4) 研究基于可并行水平集的灰度图像彩色化方法，包括参照给定的样本图像，研究基于连续纹理模式的颜色线条自动生成方法。同时考虑图像中灰度和纹理模式的连续性，并借助水平集具有拓扑结构分离与合并的特性，研究可并行的水平集图像分割方法，实现快速高效的灰度图像/视频彩色化。在此基础上，结合使用纹理特征模式相似性分析，研究图像/视频中重复场景元素的抠取方法及其相关应用。

1.3.2 拟解决的关键性问题

通过1.3.1节中本文主要研究内容（纹理合成、视频纹理和流体动画合成、灰度图像彩色化），我们总结这些关键技术研究中拟解决的问题如下：

(1) 对于处理具有显著结构的纹理，现有基于特征的纹理合成方法需要预先提取出样本纹理特征信息来保持纹理结构的连续性。但是，当纹理特征不

明显或分辨率低时，上述方法是失效的。为此，在不进行纹理特征提取操作的前提下如何描述纹理的结构布局；在纹理块的采样过程中如何获得准确的纹理相似性估计；如何在相邻纹理块之间保持结构的连续性以及产生自然、平滑的颜色传递。

(2) 对于处理近似规则纹理，现有纹理合成方法通常难以在合成结果中保持样本纹理潜在的结构规律性。虽然已有方法用于处理此类纹理，但是合成结果中在相邻纹理块之间存在明显的拼接痕迹。为此，如何在合成的纹理中保持样本纹理潜在的结构规律性以及如何保持纹理元素结构的完整性；在不采用基于马尔科夫模型的采样过程中，如何提高纹理的合成效率。

(3) 现有的方法根据给定的视频样本并结合设计的流场来驱动单幅图像产生连续的流动动画。然后，交互的流场设计需要大量的人工交互且受模糊的、低对比度的流体特征的影响。此外，局部的流体特征传递方式无法保持空间域上的一致性。为了解决这些问题，如何理解目标图像中的结构特征和流体的状态自动的产生相应的流场。同时，如何在合成的流体动画中保持时空连续性以及保持大尺度的流体结构特征。

(4) 现有灰度图像彩色化方法中，一类需要用户进行颜色标注，对于场景复杂的图像则需要大量人工交互；另一类中，虽然从参照样本图像中进行颜色传递，但对于参照图像与目标图像之间存在较大风格差异时，不能获得高质量的彩色化结果。为此，如何利用给定的参照图像自动的产生颜色标注信息，极大地减少用户输入；如何检测中具有灰度和纹理模式连续的区域，即实现并行的分割方法，提高彩色化过程的时间效率。

1.4 本文创新工作和结构安排

本文围绕可视媒体编辑与重用中的一些关键技术，针对所存在的一些关键问题，开展了一系列深入的研究工作。工作的主要创新点如下：

(1) 提出了一种基于特征描述的纹理合成方法。该方法的主要贡献在于通过分析纹理的相关性和构建纹理的方向场来描述纹理的结构特征，包括描述纹理结构布局的位置和方向信息。此外，区别于现有纹理合成方法中仅考虑颜色上的差异来定义纹理相似性的距离公式，该方法还同时考虑了纹理结构在位置和方向上的差异，由此能够很好地捕获纹理结构信息。该方法还使用基于结构变形的图像缝合方法来拼接相邻的纹理块，从而有效地解决了纹理结构和颜

色不连续问题。

(2) 提出了一种基于基元分布的周期性纹理合成与设计方法。该方法由纹理分析和纹理合成两个阶段组成。纹理分析过程通过检测及抽取每一纹理元素，不仅能够获取描述基元位置、大小、形状等特征信息，还能有效的获取基元之间的空间拓扑关系。纹理合成过程是基于样本纹理的基元间的相邻关系合成新的2D基元分布，能够达到实时的纹理合成速率。此外，该方法擅长处理近似规则的纹理，并能在合成结果中很好地保持其潜在的规律性。

(3) 提出了一种全局特征保持的流体动画合成方法。该方法主要参照行样本视频中的流体运动信息来驱动具有似流体区域的单幅静止图像，从而合成流体动画。基于构建的流体视频库，根据目标图像与视频库中最佳匹配视频帧之间的场景对应关系，该方法自动获得目标图像的流场，不再需要进行大量的人工交互。此外，该方法采用全局的流体特征的传递，能够在合成的流体动画中保持时空连续性以及保持大尺度的流体结构特征，从而产生高质量的合成结果。

(4) 提出了一种基于可并行水平集的灰度图像彩色化方法。通过分析参照图像中纹理特性，该方法能够自动产生颜色标注信息，从而极大地减轻了用户的输入。通过定义纹理尺度相似性的判别机制，该方法采用水平集图像分割方法对具有相似纹理特性的图像区域进行快速有效地分割。基于这一特性，该方法被用于实施并行的灰度图像彩色化过程，提高了彩色化过程的时间效率。此外，该图割方法还能广泛用于重复场景元素抽取、替换和编辑传递等应用中。

本文的结构安排如下：

第一章是论文的绪论，介绍了本文工作的研究背景，综述了可视媒体编辑与重用相关技术研究现状，讨论了本文的研究意义，介绍了本文的研究内容和拟解决的关键性问题，阐明了本文工作的创新之处。

第二章和第三章主要研究基于样图的纹理合成方法。其中，第二章提出和研究一种基于特征描述的纹理合成方法，主要用于处理具有显著结构特征的纹理；第三章提出和研究了基于基元分布的周期性纹理合成与设计框架，主要用于合成近似规则的纹理。

第四章提出和研究了视频纹理合成方法和基于视频样本的流体动画合成方法。

第五章提出和研究了基于可并行水平集的重复场景元素提取及灰度图像彩色化方法。

第六章是论文的总结和展望。本章概括了本文的主要研究内容，指出存在的一些问题，在此基础上，展望了下一步的研究工作。

第二章 基于特征描述的纹理快速合成

2.1 引言

纹理 (Texture) [95]是自然界中普遍存在的一种现象。通常意义上的纹理，是指由纹理元素 (elements/primitives) 按照一定随机性规则不断重复而组成，如花地、鹅卵石小路、树皮纹路等。纹理是决定物体外观的重要因素，而在实际应用中，通常需要丰富的、尺寸较大的纹理来增强几何模型表面的真实感。然而，受采样区域的局限性，并且由于光照、阴影、透视形变等原因，人们所能获取的通常为小块纹理，难以满足应用的需求。若将小块纹理映射到大的曲面上，它将导致映射后表面纹理模糊不清，或出现表面纹理接缝走样等问题。纹理合成是从给定的样本纹理中生成一幅任意大小且与样本纹理具有相似的外观的新纹理。在过去十年里，纹理合成技术已被证明是一个能自动再现样本纹理图像的有力工具，也因此得到了广泛的研究。

现有基于样图的纹理合成方法，主要包括基于像素的纹理合成方法、基于块的纹理合成方法和基于特征的纹理合成方法，总是采用马尔科夫随机场 (MRF) 来分析样本纹理的局部特征，并使用基于颜色 (RGB space) 空间的度量机制为合成新纹理选择最佳匹配。然而，使用这些方法产生的结果中时常包含明显的瑕疵，尤其是处理具有显著结构特征的纹理。例如：基于像素的纹理合成方法容易在合成结果中产生垃圾区域或纹理结构丢失 (参见图2-1(b)和(c))；而基于块的纹理合成方法则无法避免纹理结构的误匹配，从而在合成结果中产生断裂的纹理结构 (参见图2-1(d))。主要的原因是：在这些方法中通常用于相似性测量的基于RGB的距离公式，即颜色的平方差之和 (summed squared differences (SSD) of colors)，并不能很好地捕获纹理结构信息。由于不精确地相似性测量，使得这些方法在给定的样本纹理中不能搜索到合适的邻域。因此，一个能较好捕获结构信息的测量机制应该能够在对应的纹理特征之间估计到最小的累积距离。

在基于块的纹理合成框架下，本章中提出了一个理想的距离公式用于纹理结构的相似性测量，该距离公式能够在纹理子块的采样过程中找到最佳匹配，从而产生高质量的输出结果。此时，通过考虑纹理内在的特性，包括纹理特征

的位置和方向，来实现精确的相似性测量。

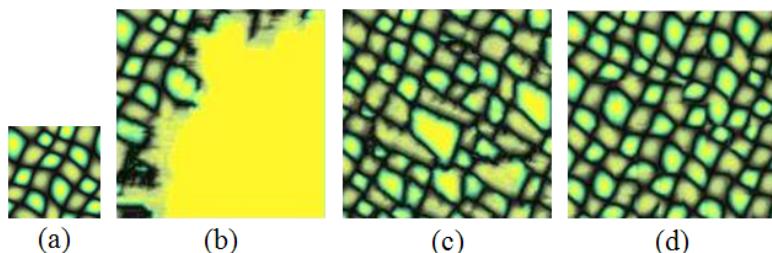


图 2-1 基于样图的纹理合成方法的缺陷

Fig 2-1 The drawbacks of example-based texture synthesis

为了实现以上目标，一方面，我们利用像素点之间的空间相关性 [96] 并引入“层”的概念（即将纹理子块的边界区域划分为相互关联的层次结构）来描述各层之间像素点的相关程度。为此，根据边界区域内各像素点到纹理块中心区域的距离来计算这些像素点的权重。当比较相邻两纹理子块的边界区域时，用于描述像素点连续分布的相关性有助于引导纹理结构相似性的测量。另一方面，我们的目标是捕获更多的结构信息来执行有效的特征匹配。由于纹理特征具有局部方向选择性 (local selectivity of orientation) 的特点，因此，可以利用纹理特征的方向来恢复纹理的结构布局。在本章提出的方法中，我们采用方向滤波并操作由此产生的梯度向量场来抽取纹理的方向场，用于描述变化较大的纹理特征。在此基础上，根据已构建的纹理方向场进一步增强相邻两纹理块之间相似性的度量。

受上述两个方面的启发，本章提出的方法通过利用纹理特征在层间和方向场之间的差异设计了一个能够选择最佳匹配的纹理结构相似性测量机制，极大地改善了基于块的纹理合成方法。

2.2 基于块的纹理合成算法框架

Liang 等人 [29] 提出了一种基于块的采样算法 (patch-based sampling algorithm)，通过将样本纹理 I_{in} 中的纹理子块作为构建输出纹理 I_{out} 的基本组成单元。为了在相邻纹理子块之间避免特征的误匹配，通常基于已合成纹理子块 $\{B_0, \dots, B_{k-1}\}$ 来选择当前的纹理子块 B_k 。该在合成过程中，算法以扫描线的顺序从左下角开始依次拷贝纹理子块到输出纹理 (参见图2-2) 中，从而产生

最终的输出结果。为简单起见，该算法仅使用大小为 $w_B \times w_B$ 的方块作为纹理子块。

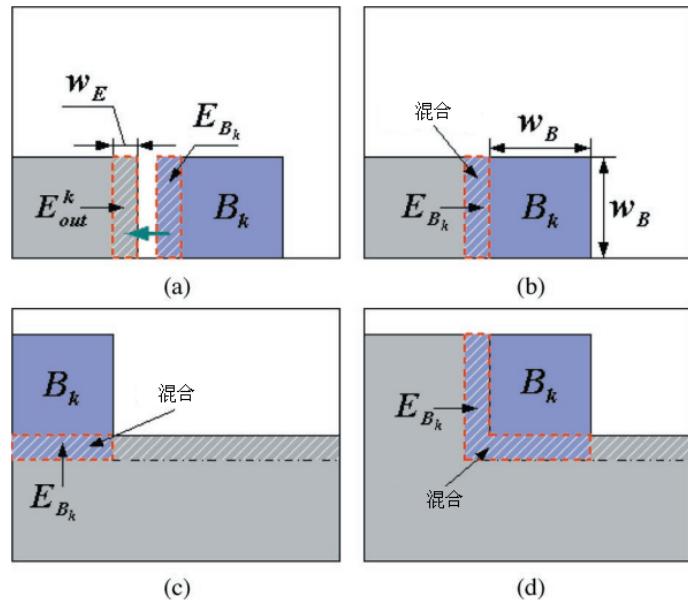


图 2-2 基于块采样的纹理合成过程 [29]

Fig 2-2 Texture Synthesis by patch-based sampling [29]

在纹理子块采样过程中，假设 I_{R_1} 和 I_{R_2} 是两个具有相同形状和大小的纹理子块，当它们之间的距离 $d(R_1, R_2)$ 小于一定阈值 δ 的时候，我们说这两个纹理子块是匹配的。通过定义纹理子块的边界区域 ∂R 为围绕区域 R 的宽度为 w_E 的带状区域（图2-3(b)中显示为阴影线的区域），该算法利用已知纹理子块的边界区域 $I_{\partial R}$ 来估计未知纹理子块 I_R 的条件概率密度（图2-3(a)）。基于MRF特性，该算法采用经验直方图来估计局部条件MRF密度 $p(I_R|I_{\partial R})$ 。也就是说，该算法直接在样本纹理中搜索到与已知 $I_{\partial R}$ 具有相似边界区域的所有纹理子块，通过从搜索的结果（用经验直方图 ψ 表示）中随机挑选一个元素作为合成的纹理子块 I_R 。其中，局部条件MRF密度由如下公式计算得到：

$$p(I_R|I_{\partial R}) = \sum_i \alpha_i \delta(I_R - I_{R^i}), \quad \sum_i \alpha_i = 1 \quad (2-1)$$

其中， I_{R^i} 是样本纹理中的一个纹理子块，其边界区域 $I_{\partial R^i}$ 匹配于边界区域 $I_{\partial R}$ ； $\delta(\cdot)$ 是狄拉克函数；权重 α_i 是归一化的相似度因子。

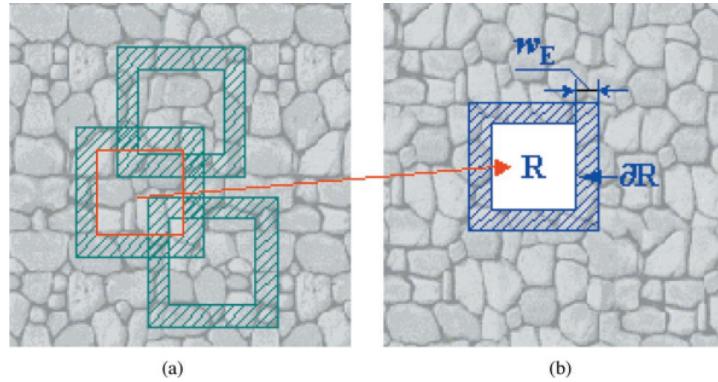


图 2-3 基于块的采样策略 [29]

Fig 2-3 A patch-based sampling strategy [29]

如图2-2(a)所示，仅有宽度为 w_E 的边界区域 E_{B_k} 与已合成纹理子块 $\{B_0, \dots, B_{k-1}\}$ 的边界区域 E_{out}^k 产生重叠。基于块的采样策略，边界区域 E_{B_k} 应该与边界区域 E_{out}^k 匹配。为保证输出纹理 I_{out} 的随机性，该算法需要生成纹理子块集 ψ_B ，即由一组与边界区域 E_{out}^k 匹配的纹理子块组成。约束样本纹理中的纹理子块 $B_{(x,y)}$ 是有效的（即每一纹理子块的左下角点坐标 (x, y) 需要在样本纹理中），纹理子块集可用如下公式计算得到：

$$\psi_B = \{d(E_{B_{(x,y)}}, E_{out}^k) < d_{max}, B_{(x,y)} \in I_{in}\} \quad (2-2)$$

其中， $d(E_{B_{(x,y)}}, E_{out}^k)$ 为基于RGB空间的距离公式，用于度量相邻两边界区域的相似程度：

$$d(E_{B_k}, E_{out}^k) = [\frac{1}{A} \sum_{i=1}^A (p_{B_k}^i - p_{out}^i)^2]^{1/2} \quad (2-3)$$

其中， A 表示边界区域内像素的总数； $p_{B_k}^i$ 和 p_{out}^i 是各自边界区域 E_{B_k} 和 E_{out}^k 内第 i 个像素点的颜色值或灰度值。 d_{max} 定义为边界区域的距离容差：

$$d_{max} = \epsilon [\frac{1}{A} \sum_{i=1}^A (p_{out}^i)^2]^{1/2} \quad (2-4)$$

其中， $\epsilon \geq 0$ 表示边界区域之间的相对匹配误差，它控制着输出纹理与样本纹理之间的相似性以及输出纹理的质量。根据上述采样过程，该算法即可从纹理子块集 ψ_B 中随机选择一个纹理子块拷贝到输出纹理中来合成最终的新纹理，纹理的合成过程描述如下：

1. 从样本纹理中随机选择一个大小为 $w_B \times w_B$ 的纹理子块 B_0 , 并将该纹理子块拷贝至待合成纹理的左下角。此时, 设置 $k = 1$;
2. 构建纹理子块集 ψ_B , 其中每一纹理子块的边界区域匹配于已合成纹理子块的边界区域 E_{out}^k ;
3. 当纹理子块集 ψ_B 为空集时, 设置 $\psi_B = \{B_{min}\}$, 即选择具有最小距离的纹理子块 B_{min} 为当前输出。此时, 该纹理子块的边界区域与已合成纹理子块的边界区域 E_{out}^k 是最接近;
4. 从纹理子块集 ψ_B 中随机选择一个纹理子块 B_k 拷贝到输出纹理中, 并设置 $k = k + 1$;
5. 重复执行步骤2-4直至纹理子块铺满整个输出纹理;
6. 在相邻纹理子块的重叠区域进行混合处理 (blending)。该操作既可以在确定了所有纹理子块之后进行, 也可以在每找到一个纹理子块时进行。

2.3 纹理特征描述

本章节的主要工作是试图减轻纹理结构的误匹配。给定样本纹理, 其纹理特征的描述在于: 如何确定样本纹理中每一像素点的位置和方向。为进一步改进基于RGB颜色空间的测量机制, 我们将利用基于层关联的判别和基于方向场的特征匹配来更好地估计纹理子块之间的相似性。

2.3.1 基于层间关联的判别模型

基于块采样的纹理合成关键在于局部纹理相似性的匹配搜索, 而纹理子块之间相似性的度量和邻域搜索又是以像素点匹配的两个相互依赖的重要过程。以纹理子块的边界区域为邻域, 基于块采样的纹理合成通常使用 L_2 距离来度量两邻域 N_1 和 N_2 之间的相似性, 其中, L_2 距离是两邻域内各对应像素点RGB颜色值的平方差之和。然而, 这个距离公式可能产生不准确的相似性测量, 由于具有最小匹配误差的两邻域之间其对应像素点的颜色值可能完全不同。此时, 运用 L_2 距离进行像素点匹配仅考虑了邻域 N_1 和 N_2 之间的相关性, 而忽略了各邻域自身所具有的相关性, 且称之为邻域的自相关性。

通常, 对于样本纹理中的任一像素点 p , 像素点 q 与像素点 p 具有极强的空间相关性当且仅当他们之间的距离是彼此靠近的。在样本纹理中, 虽然像素点

的分布在几何结构上是连续的，但在颜色空间上却是以离散形式分布的。因此，我们需要通过空间相关性分析来描述连续的像素点分布。已知纹理子块由中心区域 R 和边界区域 ∂R 组成（图2-4(a)），根据边界区域内每一像素点到中心区域的距离将边界区域划分成相互关联的层次结构（图2-4(b))。需要指出的是：纹理子块的中心区域可以看成是一个像素点，是因为被边界区域包围的中心区域是不可分的整体。为计算这个距离，我们采样8-邻域距离（棋盘距离）函数 $D_8(p, q) = \max(|i - h|, |j - k|)$ ，其中， q 表示中心像素点； p 表示像素点 q 的邻域内的一个像素点； (i, j) 和 (h, k) 表示这两个像素点的坐标。可以看出，由8-邻域距离计算得到的区域形状与纹理子块的形状相符，即方块区域（图2-4(c))。

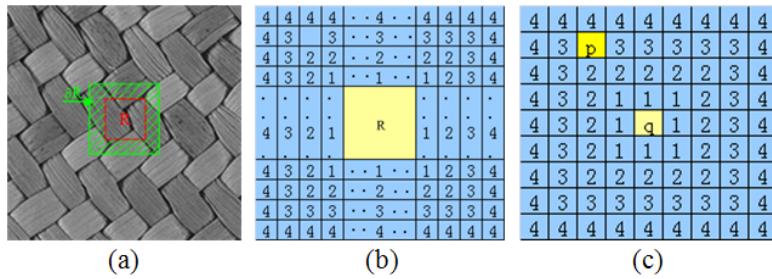


图 2-4 纹理子块边界区域的层次划分

Fig 2-4 The layer partition for the boundary region of texture patch

基于上述空间相关性分析和层次划分，纹理子块边界区域的层间相关性定义如下：1) 当某一层逐渐偏离中心区域时，其相关程度将逐渐减弱；2) 相对于中心区域，某一层内所有像素点具有相同的相关性。这个层间相关性表明边界区域内的每一像素点对于中心区域的贡献各不相同，由此，可以通过计算这些像素点的权值来进一步量化层间的差异。由于每一像素点的权值依赖于该像素点所属的层次以及在这一层中像素点的总数，因此，该权值由层间权重 $wL(\cdot)$ 和层内权重 $wIL(\cdot)$ 两部分组成。

假设纹理子块的边界区域被划分为 M 层，即表示为边界区域的宽度。横跨每一层 L 的累积代价 $Cost = \sum^M L$ 用于描述纹理子块的边界区域与中心区域的空间相关性。对于边界区域内的任一像素点 p ，其层间权重由如下公式计算得到：

$$wL(p_L) = \frac{M - L + 1}{Cost(M)}, \quad L = 1 : M \quad (2-5)$$

$$\sum_{L=1}^M wL(p_L) = 1 \quad (2-6)$$

其中，公式(2-5)反映边界区域内每一层相对于中心区域的关联程度；为计算层内权重，根据当前 L 层中像素的总数计算每一像素点 p 的贡献值。假定层 L 中像素点的总数为 $Num(L)$ ，该像素点的层内权重为像素总数的倒数：

$$wIL(p_L^i) = \frac{1}{Num(L)}, \quad L = 1 : M, \quad i = 1 : Num(L) \quad (2-7)$$

$$\sum_{i=1}^{Num(L)} wIL(p_L^i) = 1 \quad (2-8)$$

其中， p_L^i 表示第 L 层中的第 i 个像素点。一旦获得了层间权重 $wL(p_L)$ 和层内权重 $wIL(p_L^i)$ ，边界区域内的每一像素点 p 的权值可定义：

$$\omega(p_L^i) = wL(p_L) \times wIL(p_L^i) \quad (2-9)$$

使用该公式即可计算得到纹理子块边界区域内所有像素点的权值。

通过考虑边界区域 B_{In} 和 B_{Out} 自身的层间相关性，重新定义如下距离公式来度量相邻两纹理子块之间的差异：

$$\begin{aligned} D_c(B_{In}, B_{Out}) &= \sum_{i=1}^A (\omega(p_{In}^i) \cdot p_{In}^i - \omega(p_{Out}^i) \cdot p_{Out}^i)^2 \\ &\approx \sum_{i=1}^A sprt(\omega(p^i) \cdot (p_{In}^i - p_{Out}^i)^2) \end{aligned} \quad (2-10)$$

其中， A 表示边界区域内像素的总数； p_{In}^i 和 p_{Out}^i 表示边界区域 B_{In} 和 B_{Out} 内第 i 个像素点的颜色值或灰度值； $\omega(p_{In}^i)$ 和 $\omega(p_{Out}^i)$ 为它们的权值系数，并用 $\omega(p^i)$ 表示由于相同的取值。在公式(2-10)中，我们使用 $\omega(p^i)$ 替代 $\omega^2(p^i)$ ，是因为当边界区域足够宽时 $\omega^2(p^i)$ 的取值接近于零值。此外，我们使用开方操作 $sprt(\cdot)$ 来求解近似的估计。如图2-5中所示的例子，编织纹理(图2-5(a))中各纹理元素由于具有相近的颜色和光照，当使用基于RGB颜色空间的距离公式进行纹理子块之间的相似性度量时，在合成的结果中容易产生明显的结构断裂的纹理特征，参见图2-5(b)中蓝色椭圆标记的区域。其主要原因是像素点的RGB值仅能捕获影响纹理特征变化的颜色信息。图2-5(c)给出在引入层间关联后获得的输出结果，此时，结果中极大地减轻了纹理结构的不连续现象。由此可看出，空间相关性的使用能够在合成结果中有效地保持样本纹理的结构特征。

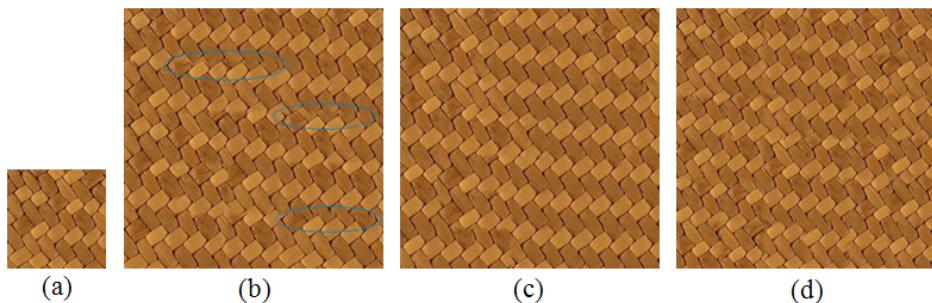


图 2-5 不同距离公式下的合成结果比较

Fig 2-5 The comparison of the synthesis results using different distance metric

2.3.2 基于方向场的特征匹配

在多数情况下，现有纹理合成方法都是利用纹理特征所固有的方向特性来执行有效的特征匹配。例如，Wu和Yu [51]提出的纹理合成方法中主要关注纹理子块边界区域内特征像素点的正切方向。一致的正切方向能够保证几何的连续性，使得在合成结果中能够再次生成平滑的纹理特征。然而，该方法最大的局限在于很难抽取出模糊的或分辨率低的纹理特征。与基于特征抽取的纹理合成技术不同，本章方法则利用Rao等人 [97, 98]提出的纹理流场分析方法来构建样本纹理的方向场，其主要优势是能够估计纹理中各像素点的局部方向。

纹理方向场的估计依赖于高斯梯度的计算。当纹理图像是灰度图像时，通常可采用有限差分来计算纹理图像中各像素点的梯度。然而，对于多数彩色纹理图像，通过分别计算RGB颜色通道的梯度并合并这三个独立分量得到的合成梯度可能导致错误的结果，主要原因是梯度的方向不一定是相同的。因此，我们采用多通道图像的梯度求解方法 [99]在RGB颜色空间上直接计算纹理图像的梯度，从而获得纹理中各像素点的方向角 θ 和幅度 $F(\theta)$ 。

Rao等人定义纹理的方向场由图像角度 (image angle) 和图像连贯度 (image coherence) 组成，其中图像的角度用于表示纹理中每一像素点的局部主导方向 (dominant local orientation)；而图像连贯度则表示每一像素点的各向异性程度。对于给定的样本纹理，其方向场的构建过程概括如下：

1. 采用高斯滤波平滑样本纹理，去除纹理中的噪声。同时，采用方法 [99]计算平滑后纹理图像的梯度，包括方向角 θ 和幅度 $F(\theta)$ ；
2. 估计样本纹理中每一像素点的局部主导方向角 $\hat{\theta}$ 。为了计算该主导方向角，通常需要结合使用当前像素点邻域内的梯度。由于在纹理特征密集的

区域，梯度方向角往往互会相互抵消。因此，以当前像素点 (x, y) 为中心的邻域 $N \times N$ 内的主导方向角 $\hat{\theta}$ 可由如下公式计算得到：

$$\hat{\theta} = \tan^{-1}\left(\frac{\sum_{i=x-N/2}^{i=x+N/2} \sum_{j=y-N/2}^{j=y+N/2} F^2(\theta_{ij}) \sin 2\theta_{ij}}{\sum_{i=x-N/2}^{i=x+N/2} \sum_{j=y-N/2}^{j=y+N/2} F^2(\theta_{ij}) \cos 2\theta_{ij}}\right) \quad (2-11)$$

其中， i 和 j 为邻域内各像素点的位置索引； θ 和 $F(\theta)$ 为像素点在位置的方向角和幅度。如果已估计的主导方向角 $\hat{\theta}$ 是公式(2-11)的一个解，则 $\hat{\theta} + \pi/2$ 也是。这两个相隔 90° 的值意味着像素点在位置 (x, y) 上具有两个相互正交的方向。

3. 计算连贯度值。为了估计每一像素点的连贯度，我们仍然考虑像素点的邻域。此时，邻域的使用确保了像素点在邻域中心处具有最高的连贯度。再次利用先前已计算的梯度方向角和幅度，连贯度的估计可由如下公式定义：

$$\rho = \frac{\sum_{(i,j) \in N} \|F(\theta_{ij}) \cos(\theta_{ij} - \hat{\theta}_{xy})\|}{\sum_{(i,j) \in N} F(\theta_{ij})} \quad (2-12)$$

其中， $F(\theta_{ij}) \cos(\theta_{ij} - \hat{\theta}_{xy})$ 表示邻域内各像素点 (i, j) 以方向角为 θ_{ij} 的梯度幅度 $F(\theta_{ij})$ 在像素点 (x, y) 处以主导方向为 $\hat{\theta}_{xy}$ 上的投影。

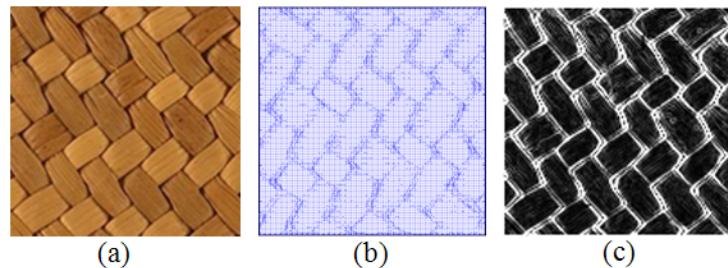


图 2-6 编织纹理方向场的构建
Fig 2-6 The constructed orientation field for the given fabric texture

根据上述关键步骤，当指定大小的邻域以扫描线的顺序贯穿整个纹理图像时，可计算得到每一像素点的主导方向角和连贯度值。图2-6中给出了编织纹理（图2-6(a)）的方向场，包括角度图（图2-6(b)）和连贯度图（图2-6(c)）。需要指出的是，纹理方向场的构建受邻域大小的影响。一方面，具有尺度较小的邻域并不能很好地捕获纹理特征，使得该过程不能产生正确的纹理方向场；另一方面，具有尺度较大的邻域将会检测出许多细微的纹理特征，容易在纹理合成过程中干扰特征的匹配。此外，使用较大尺度的邻域会增加相应的计算时

间，由于邻域需要在纹理图像中以扫描线顺序逐像素点移动。为了平衡纹理方向场的构建效率，本章方法中指定大小为 $N = 11$ 的邻域用于处理本章节提供的所有样本纹理，特别是结构显著的样本纹理。

一旦样本纹理由其对应的方向场进行表示，我们将使用该方向场作为输入对象并用于相邻纹理子块边界区域之间的特征匹配。由于已构建的方向场并不是一个向量场，样本纹理中各像素点的主导方向角在符号上是有二义性的 [99]，因此，将两个离散方向场直接相减来计算它们之间的相似性距离是不正确的。假定每一像素点的方向场可以表示成方向的线段，其中，线段的长度为该像素点的连贯度，方向为局部主导方向角。一种有效的测量方式是计算由一对方向线段组成的三角形区域的面积，这两个方向线段来自于对应位置上的两个方向场。当这两个线段的方向彼此靠近时，所组成区域的面积通常接近于零值。如果这两个线段的方向截然相反时，所计算的面积取值同样为零。当在纹理子块之间进行特征匹配时，这个面积因此可用来估计对应方向场之间的差异，通过累加边界区域所有方向场的差异来度量纹理特征之间的相似性程度：

$$D_o(of_{In}, of_{Out}) = \frac{1}{A} \sum_{i=1}^A \left(\frac{1}{2} \rho_{In}^i \rho_{Out}^i |\sin(\hat{\theta}_{In}^i - \hat{\theta}_{Out}^i)| \right) \quad (2-13)$$

其中， of_{In} 和 of_{Out} 分别为从边界区域 B_{In} 和 B_{Out} 构建得到的方向场； A 表示边界区域内像素的总数； ρ_{In}^i 和 ρ_{Out}^i 表示方向场 of_{In} 和 of_{Out} 内第 i 个像素点的连贯度；同时， $\hat{\theta}_{In}^i$ 和 $\hat{\theta}_{Out}^i$ 为主导方向。

通过构建给定样本纹理的方向场，图2-5(d)中显示的合成结果中比使用基于RGB颜色空间的距离公式下产生的合成结果（图2-5(b)）能更好地保持结构的规律性。主要原因是被用于特征匹配的纹理方向信息能够在样本纹理中找到具有相似结构布局的纹理子块。然而，即使在本章方法中使用基于纹理方向的距离公式进行相似性度量，在合成结果中仍然无法完全消除不连续的纹理结构。一方面，采样高斯滤波虽然能够消除纹理图像中的噪声，但这模糊了样本纹理中主要的结构特征。另一方面，相似的纹理特征在统计上并不是完全相同的，使得无法避免纹理特征的不对齐。

2.3.3 基于纹理特征描述的距离公式定义

为在纹理合成过程中进行最佳匹配纹理子块的选择，本章方法通过结合上述层间相关性和纹理方向场来设计一个新的距离公式，用于获得准确的相似性

测量。对于纹理子块边界区域内的各像素点，相关性分析有助于在颜色空间上描述一个连续的像素点分布；以及方向场的构建在描述纹理特征布局上是有用的。为产生最终的合成结果，从给定样本纹理中选取的每一最佳匹配纹理子块 P 需要满足如下要求：

$$P = \min dist(B_P, B_{Out}) \quad (2-14)$$

$$dist(B_P, B_{Out}) = D_c(B_P, B_{Out}) + \lambda D_o(of_p, of_{Out}) \quad (2-15)$$

其中， B_P 和 B_{Out} 分别表示样本纹理中纹理子块的边界区域和当前已合成纹理子块的边界区域；能量项 $D_c(B_P, B_{Out})$ 用于测量边界区域在颜色上的差异，能量项 $D_o(of_p, of_{Out})$ 用于估计边界区域在方向场上的差异，他们分别用公式(2-10)和公式(2-13)计算得到。参数 λ 是一个用来调节这两个能量项之间相对重要性的正比例系数。在本章方法的实现中，当 $\lambda = 0.13$ 时，公式(2-15)可以达到一种平衡，使得产生高质量的合成结果。图2-7显示参数 λ 设置不同值时对合成结果的影响。当设置 $\lambda = 0.6$ 时，在合成的结果中虽然能够很好的保持样本纹理的结构特征，但是它并没有很好地保持结构的规律性，参见图2-7(b)中用蓝色标记的椭圆区域。比较设置 $\lambda = 0.2$ 和 $\lambda = 0.13$ 时产生的合成结果(图2-7(c)和(d))，前者结果中显示明显的结构误匹配，由于细微的特征不对齐容易导致纹理结构的不连续；后者则显示一个较高质量的合成结果，能够与样本纹理具有非常相似的外观。图2-5(c)中显示的是 $\lambda = 0$ 时的合成结果。

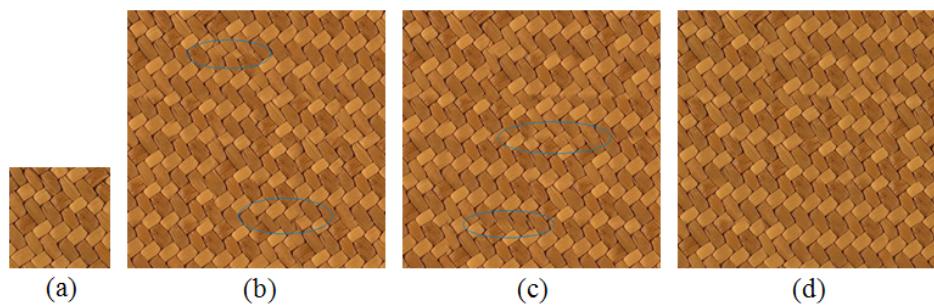


图 2-7 参数 λ 对合成结果的影响
Fig 2-7 The effect of different relative important weight λ

基于块采样框架下的纹理合成算法，其中需要由用户预先指定的主要两个参数，包括纹理子块的大小和边界区域的宽度，很大程度上影响了算法的合成效率。一个具有合理大小的纹理子块，不仅能很好地捕获样本纹理的结构信

息，而且还能减少合成过程的运行时间。而一个具有适合宽度的边界区域能保证在样本纹理中找到唯一匹配的纹理子块，使得在纹理子块之间避免了误匹配的纹理特征。在本章进行的所有实验中，所选纹理子块的尺寸应该足够大，但也不能超过样本纹理大小的一半。同时，边界区域的宽度应该设置为纹理子块大小的 $10 - 15\%$ 。例如，当指定纹理子块大小为 64×64 时，对于随机纹理，边界的宽度通常为6像素宽度；而对于结构显著的纹理则设置为10像素宽度。

此外，纹理子块集的生成是基于块采样过程中的关键计算步骤，然而，由于穷尽的搜索，使得这个采样过程是非常耗时的。如距离度量公式(2-15)的计算复杂度为 $O(n^2)$ ，其中， n 为样本纹理中的总像素点数目。为了加快纹理合成的速率，在本章方法中将采用离散傅立叶变换 [100, 101] (Fast Fourier Transforms, FFT) 加速基于层间关联的相似性测量部分，此时，公式(2-10)被改写为如下形式：

$$D_c(B_{In}, B_{Out}) = (\omega(p^i))^2 \cdot [\sum_{i=1}^A (p_{In}^i)^2 + \sum_{i=1}^A (p_{Out}^i)^2 - 2 \sum_{i=1}^A p_{In}^i \cdot p_{Out}^i] \quad (2-16)$$

其中，前两项为纹理子块边界区域 B_{In} 和 B_{Out} 内所有像素点的平方和，能够以 $O(n)$ 时间代价进行有效计算；第三项为两个边界区域的卷积，在使用基于FFT的搜索下的计算时间为 $O(n \log(n))$ 。由于公式(2-13)不具有平方差的形式，从而我们采样主元分析方法(PCA)通过减少搜索空间的维数来实现基于方向场匹配部分的加速过程。对于大小为 $M \times N$ 的样本纹理，对应的纹理方向场包含 $n = M \times N$ 个数据点 $\{x_1, \dots, x_n\}$ ，每一数据点是一个 $d = M' \times N'$ 维的向量，其中， M' 和 N' 为纹理子块大小。PCA计算这个数据点集的特征值和特征向量。由最大特征值对应的特征向量组成的子空间能够反映源数据点集的主要变化特征，若将高维数据点集映射到这个低维的子空间，此时，每一数据点的维数 $d' \ll d$ 将大大降低。使用PCA降维不仅能够保留源高维数据97%的主要特征，保证相似性测量的有效性，而且还能加速特征匹配过程。

2.4 基于结构变形的纹理子块拼接

在合成过程中，每次输出的最佳匹配纹理子块由于局部光照的差异通常存在强度的不一致，因此，当使用现有技术 [28, 29, 34] 进行纹理子块的拼接时，如羽化、动态规划、图割等技术，在处理后的重叠区域内会产生纹理结构的断

裂和颜色的不连续。为了消除这些瑕疵，本章方法采用Jia等人 [102, 103]提出的基于结构变形的图像缝合方法来拼接纹理子块，其处理过程概括如下：

1. 计算纹理子块重叠区域内的最优划分。在重叠区域内，定义任意相邻像素点 p 和 q 的梯度对齐代价 $S(p, q)$ 为在RGB颜色通道的计算值的总和：

$$S(p, q) = \sum_{rgb} (1 - \beta) S_m + \beta S_d \quad (2-17)$$

其中， β 是一个平衡代价项 S_m 和 S_d 相对影响的权值系数； S_m 用于测量相邻像素点之间的梯度平滑性：

$$S_m(p, q) = \|\nabla I_S(p)\| + \|\nabla I_S(q)\| + \|\nabla I_T(p)\| + \|\nabla I_T(q)\| \quad (2-18)$$

其中， ∇ 为梯度算子； I_S 和 I_T 为具有重叠区域的两相邻纹理子块。由此，通过考虑梯度的平滑性，从而可以避免在断裂的边界处进行划分。 S_d 用于测量相邻像素点之间的梯度相似性：

$$\begin{aligned} S_d = & \|\nabla_x I_S(p) - \nabla_x I_T(p)\| + \|\nabla_x I_S(q) - \nabla_x I_T(q)\| + \\ & \|\nabla_y I_S(p) - \nabla_y I_T(p)\| + \|\nabla_y I_S(q) - \nabla_y I_T(q)\| \end{aligned} \quad (2-19)$$

其中，每一项表示重叠区域内各对应像素点位置上梯度级的相似性，使得相似性判断在梯度域上进行。因此， S_d 促使划分的曲线穿过重叠区域内具有相似梯度的像素点位置。该步骤通过采样图割方法求解代价函数 $S(p, q)$ ，从而在重叠区域上产生一条的缝合割，分别用 $\partial I'_S$ 和 $\partial I'_T$ 表示。

2. 特征点的检测与匹配。分别在缝合割 $\partial I'_S$ 和 $\partial I'_T$ 上选择梯度幅度大于一定阈值的所有像素点为1D特征点，并记录各像素点的方向强度，标记为 g_S 和 g_T 。通过构建这两个特征点集的一一映射关系： $f_s^i \rightarrow f_t^i$ 来进行特征点的匹配，这个问题可以表示为：

$$\begin{aligned} E' = \min \sum_{0 \leq i < m} (g_T(i) - g_S(k_i))^2 \\ s.t. \quad 0 \leq k_0 < \dots < k_{m-1} < n \end{aligned} \quad (2-20)$$

其中， m 和 n 分别为特征点集 g_S 和 g_T 的像素点总数。采用动态规划算法进行求解，通过最小化上述能量公式来匹配缝合割中第 a 个特征点和缝合割中第 b 个特征点。

3. 计算变形向量。根据缝合割上的每一匹配特征点对 $(f_t^{i_1}, f_s^{i_2})$, 构建其对应的变形向量:

$$V(f_t^i) = \{V_x(f_t^{i_1}), V_y(f_t^{i_1}), V_{\|\nabla\|}(f_t^{i_1})\} \quad (2-21)$$

其中, V_x 和 V_y 为特征点 $f_t^{i_1}$ 距离特征点 $f_s^{i_2}$ 在图像域上 x 和 y 方向的偏移分量。 $V_{\|\nabla\|}$ 测量梯度域上的强度差异:

$$V_{\|\nabla\|}(f_t^{i_1}) = \|\nabla I_t(f_t^{i_1})\| - \|\nabla I_s(f_s^{i_2})\| \quad (2-22)$$

因此, 变形向量由图像空间的几何差异和梯度域上的强度差异组成。通过将计算的变形向量从缝合割处传播到纹理子块的内部, 以保持重叠区域内纹理结构和颜色的平滑传递。

4. 变形传播。为了平滑地从缝合割 $\partial I'_t$ 上稀疏的特征点传递变形到纹理子块 I_T , 定义变形区域 S_t 为重叠区域划分后属于纹理子块 I_T 的那部分区域。在这个变形区域上, 通过采用共轭梯度方法 (conjugate gradients) 求解如下最小能量公式:

$$V^* = \arg \min_V \int \int_{p \in S_t} \|\nabla V\|^2 dp \quad (2-23)$$

$$\Delta V = 0 \quad \text{with} \quad V|_{\partial S_t} = V^*|_{\partial S_t} \quad (2-24)$$

由此, 可计算获得变形区域内每一像素点的变形向量, 从而构建变形梯度图 (参见图2-8(e))。最后, 通过对变形梯度图进行泊松方程求解来重构最终的拼接结果。

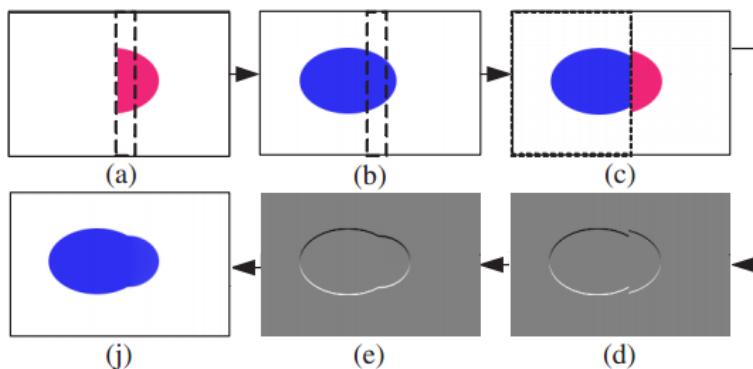


图 2-8 结构变形 [102]

Fig 2-8 Structure deformation [102]

图2-8给出了两纹理子块的拼接流程图，在重叠区域内将实现结构特征和颜色的平滑传递。需要指出的是，我们在找到所有纹理子块后再采用上述操作进行纹理子块的拼接。对于该方法的详细说明可参见文献 [102, 103]。

2.5 实验结果与分析

本章方法在AMD Phenom(tm) 8400 2.10 GHz三核处理器及2GB内存空间的硬件环境中，通过使用未优化的Matlab代码进行所有纹理合成实验。本章方法能够很好地处理多种类型的纹理，从随机纹理到具有显著结构的纹理。由于纹理子块的采样过程需要花费大量的时间，对于大小为 128×128 的样本纹理，合成大小为 300×300 的输出结果所耗时间约为5-7分钟。因此，本章方法采用基于FFT搜索和PCA降维技术来加速纹理合成过程，使得合成时间减少到1分钟左右。表格1给出了本章方法与现有纹理合成方法（图2-9到图2-13中给出的所有方法）在合成质量和速度等方面的比较。

表 2-1 多种纹理合成方法性能比较
Table 2-1 Performance comparison for many texture synthesis methods

方法	方法种类	合成质量	合成速度	纹理类型
文献[20]	基于像素点的	较差	一般	所有类型
文献[21]	基于像素点的	一般	快速	自然纹理
文献[22]	基于像素点的	较好	实时	所有类型
文献[25]	基于像素点的	较好	快速	所有类型
文献[29]	基于块的	一般	实时	所有类型
文献[54]	基于块的	高质量	快速	近似规则纹理
文献[28]	基于块的	一般	中等	所有类型
文献[34]	基于块的	高质量	快速	所有类型
文献[52]	基于块的	较好	快速	所有类型
文献[51]	基于块的	高质量	一般	结构显著的纹理
本章方法	基于块的	高质量	快速	所有类型

图2-9和图2-10给出的是使用基于像素点的纹理合成方法产生的结果。方法 [20]对于由不规则形状和大小物体随机排列组成的纹理（图2-9中第一列）处理

相对较差，输出质量较差的结果（图2-9中第二列）。这主要是因为在该方法中所使用的L-形邻域的大小严重影响算法的性能，使得在结果中经常产生大片的垃圾区域。Ashikhmin等人 [21]提出的方法利用空间一致性来加速纹理合成过程，然而，局部的邻域搜索容易导致结果中（图2-9中第三列）出现不规律和模糊的纹理结构。基于跳跃图的纹理合成算法 [22]和并行可控的纹理合成算法 [25]在质量上和速度上已经取得了很大的进步（参见图2-10）。然而，这些方法无法保证结果中纹理结构是相互匹配的（图2-10中第二、三列），导致产生断裂的纹理特征。显然，本章方法能够产生具有更好合成质量的结果（图2-9和图2-10中的第四列），保持了与样本纹理具有相似的外观特征。这主要是由于本章方法属于基于块的纹理合成方法，继承了这类方法所具有的可扩展性和高效性等优势，有效地改善了纹理合成的质量和速度。

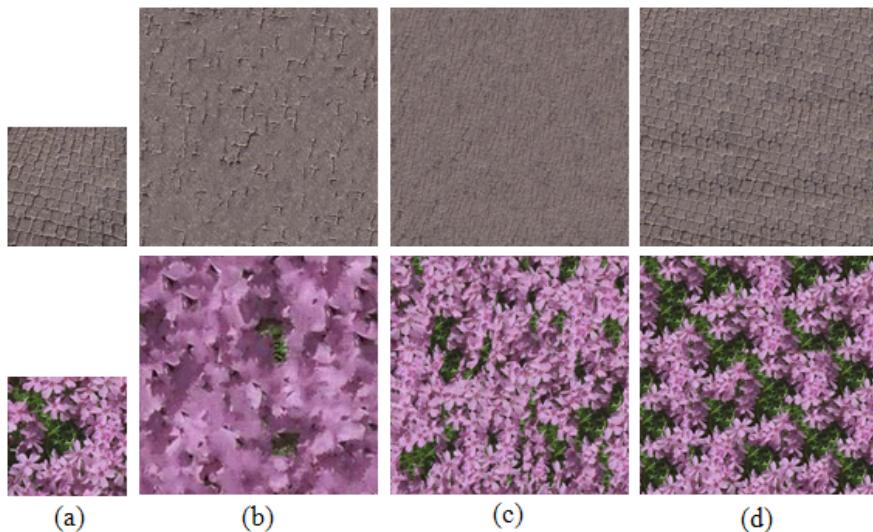


图 2-9 与基于像素的纹理合成 [20] 和 [21] 的比较

Fig 2-9 Comparison with the WL algorithm [20] and the Ashikhmin's algorithm [21]

图2-11中，对于给定的样本（图2-11中第一列），使用本章方法能够产生高质量的结果（图2-11中第四列）。传统的基于块采样的纹理合成算法 [29]未能产生理想的结果，由于样本纹理中不易辨别的颜色信息容易产生混淆。在第二列给出的结果中显示明显的视觉瑕疵，如合成第一个样例时产生不连续纹理结构以及合成第二个样例时产生垃圾区域。采用近似规则纹理合成方法 [54]能够产生高质量的结果（图2-11中第三列），然而，当纹理拼接块不能准确地粘贴在一起时，合成第二个样例的结果中显示断裂的纹理结构（石头的阴影），由于纹

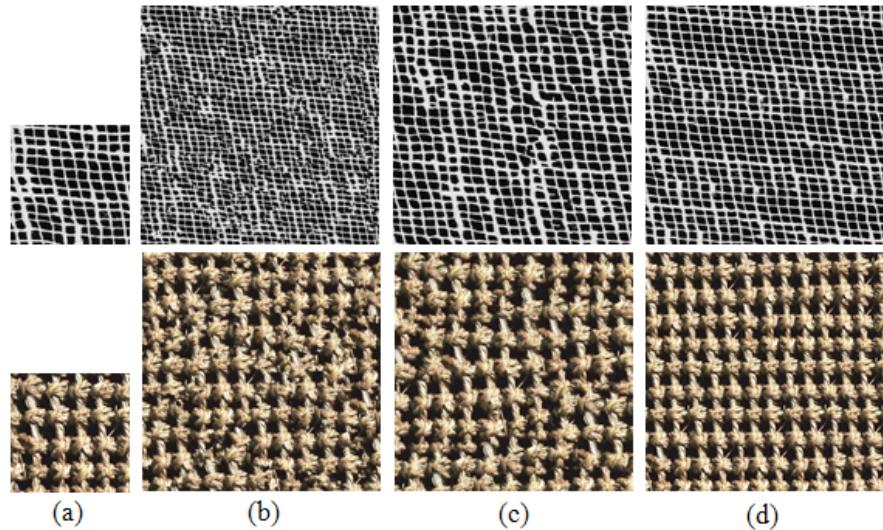


图 2-10 与基于像素点的纹理合成 [22]和 [25]的比较

Fig 2-10 Comparison with the Jump Map based texture synthesis algorithm [22] and the parallel controllable texture synthesis algorithm [25]

理拼接块不能保持全局的纹理特征。本章方法在比较相邻纹理子块的差异时，尝试利用像素的相关性和纹理的方向场信息来获得准确的相似性度量。此外，本章方法能够保持样本纹理潜在的结构规律性。

图2-12给出了本章方法与现两种基于块的纹理合成方法的比较，包括图像缝合纹理合成方法 [28]和基于图割的纹理合成方法 [34]，比较使用不同纹理子块拼接技术下的合成质量。这些方法一个共同的问题是在相邻纹理子块的边界区域内通常会出现断裂的纹理结构（图2-12第二、三列）。由于基于RGB的相似性度量不能找到满足合成需求的最佳匹配纹理子块，使得采用如动态规划和图割的技术拼接纹理子块仍然存在结构特征和颜色的不对齐。而在本章方法中，通过使用基于结构变形与变形传播的方法处理纹理子块的重叠区域，该方法有助于产生纹理特征的平滑传递，因而能够产生比这两种方法更好的结果。

图2-13给出了与基于外观空间的纹理合成方法 [52]和基于特征匹配的纹理合成方法 [51]的比较，其中第二、三列是这些方法产生的结果。它们利用纹理特征来解决纹理结构不连续问题。然而，由于方法 [52]中使用二义性的符号特征距离场进行特征匹配，因此未能产生平滑的过渡。此外，虽然方法 [51]通过引入局部变形来对齐纹理特征，但是结果中仍然显示不自然的纹理结构扭曲痕迹。本章方法能够产生较好的结果（图2-13第四列），主要原因是基于层间相

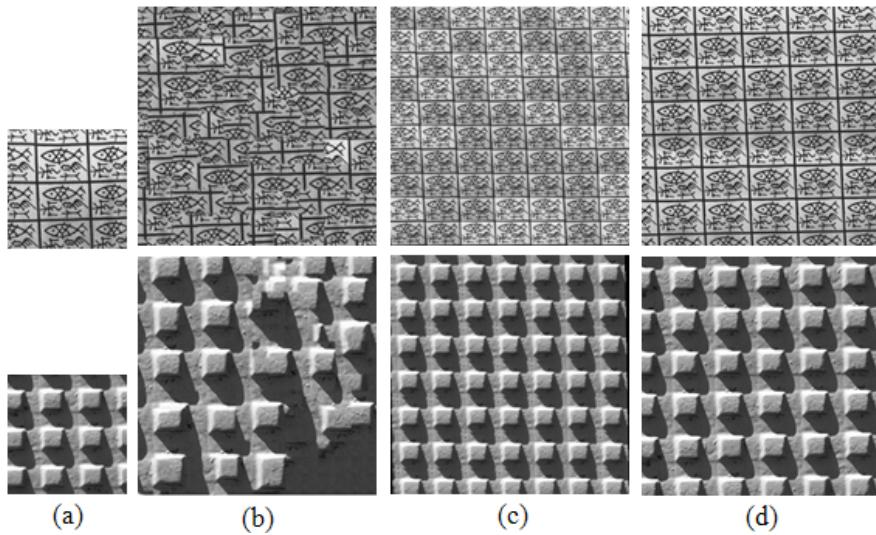


图 2-11 与基于块采样的纹理合成方法 [29]和规则纹理合成方法 [54]的比较

Fig 2-11 Comparison with patch-based sampling algorithm [29] and near-regular texture synthesis [54]

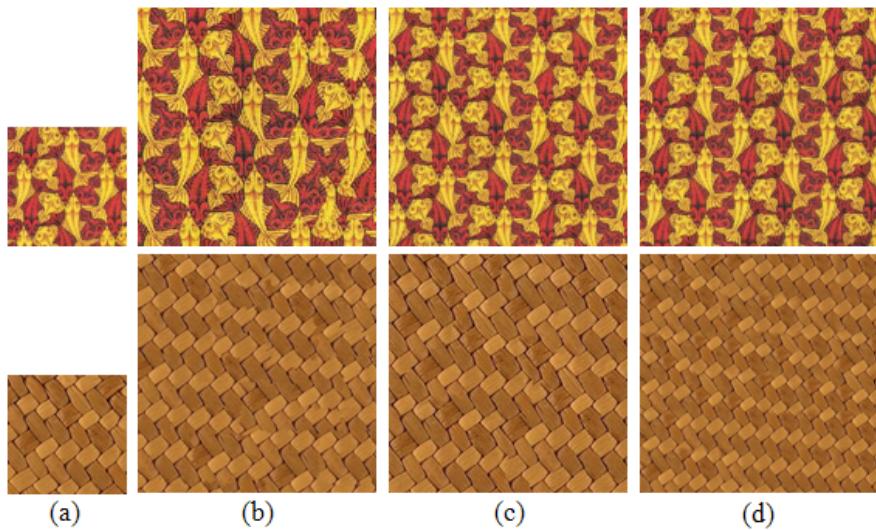


图 2-12 与图像缝合纹理合成方法 [28]和基于图割的纹理合成方法 [34]的比较

Fig 2-12 Comparison with image quilting algorithm [28] and the graph cuts algorithm [34]

关性的颜色差异度量能够识别显著的纹理结构，而且基于方向场的距离度量能够保持结构的连续性。图2-14给出了更多的输出结果，其中，左边对应样本纹理，右边为合成的纹理。这里给出的都是具有显著结构的样本纹理，本章方法

能够较好地捕获结构信息从而产生令人满意的结果。

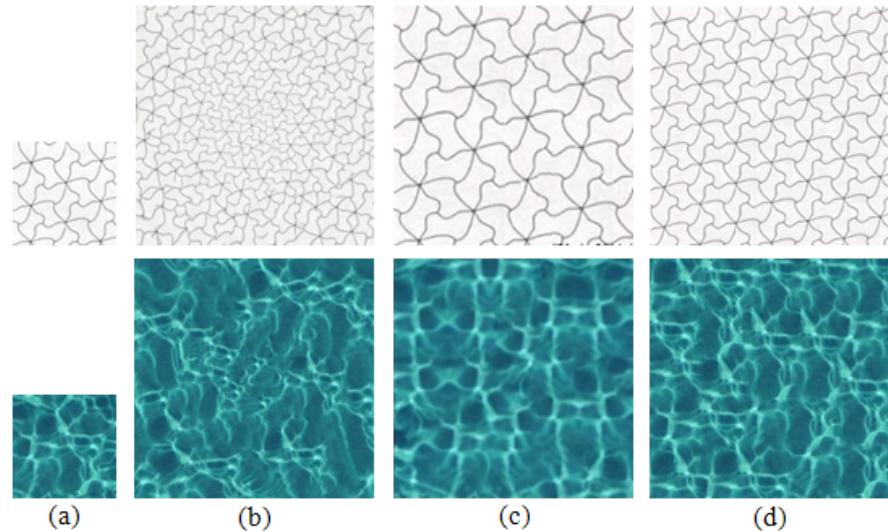


图 2-13 与基于外观空间的纹理合成方法 [52]和基于特征匹配的纹理合成方法 [51]的比较
Fig 2-13 Comparison with appearance-space texture synthesis [52] and the feature matching synthesis technique [51]



图 2-14 纹理合成结果
Fig 2-14 More results for texture synthesis

2.6 本章小结

在本章方法中，我们提出基于特征描述的纹理合成方法来生成新纹理，主要改进了现有纹理合成方法中使用基于RGB距离公式的相似性测量。层间相关性的和纹理方向场的使用能够很好地捕获样本纹理中的结构信息。另外，采用结构变形及变形传播方法拼接相邻纹理子块，使得在纹理子块的边界区域处能够产生平滑的传递。通过与现有几种经典的纹理合成方法进行比较，给出的实验结果显示本章方法不仅能够减轻纹理结构的不连续，而且还能很好地处理具有显著结构特征的纹理，产生高质量的合成结果。

然而，本章方法的不足之处在于：1) 合成的结果缺乏随机性，由于在合成过程中每次仅选择一个具有最小匹配误差的纹理子块作为当前输出。如图2-15中第二列所示合成结果虽然没能保持样本纹理的结构布局，但纹理结构是匹配的。2) 本章方法不能在输出结果中完全消除不连续的纹理结构，即使采用了结构变形及变形传播技术对齐误匹配的纹理特征（图2-15中第四列）。当纹理子块的重叠区域内仅能检测到有限数量的特征点时，主要受边界区域宽度的约束，此时，变形操作是失效而不能产生正确的结构匹配结果。此外，结果中还会显示严重的纹理子块重复拷贝，由于无法有效地估计合理的纹理子块大小和边界区域的宽度，这仍然是基于块的纹理合成方法中一个待解决的问题。

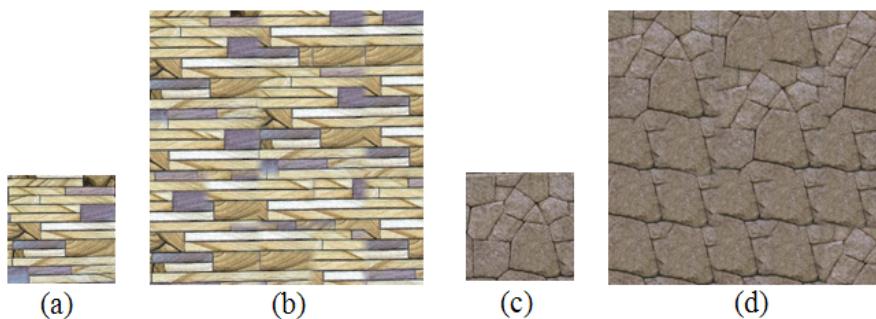


图 2-15 失败的例子
Fig 2-15 The failed example

第三章 基于基元分布的周期纹理合成与设计

3.1 引言

在自然界中，近似规则的纹理是普遍存在的，如建筑物、墙纸、地板、编织物、窗格、蜂窝、动物的皮毛和树叶等。这种近似规则模式的可变外观背后的共性是趋于极强的周期性和对称性，且伴随着一定随机信号和噪声。因此，近似规则纹理可以看成是规则纹理在统计特征上的变形，也就是说，其中每一局部纹理区域在形状、大小、颜色和光照上具有各自的变化。现有纹理合成方法 [19–21, 28, 29, 34]都是基于局部邻域的统计方法，且利用样本纹理中的如颜色、结构特征等信息产生新的纹理。然而，使用这些纹理合成方法处理近似规则的纹理，我们发现合成的纹理不能很好地保持结构的规律性。关键问题是这些纹理合成方法忽略了近似规则纹理潜在的结构特性，尽管目前已有方法 [54, 55]做出了一些尝试。

本章提出的方法受这类纹理基本概念的启发，即近似规则纹理是由纹理元素（称之为“基元”）在空间上不断重复出现而形成的。这些基元的2D空间排列（基元的分布）可以看成是一种近似周期性的模式。区别于传统的纹理合成方法，我们通过利用基元的分布提出了一种新的近似规则纹理合成方法。观念上，这个算法将以基元的分布作为输入对象，并且能够产生一个新的、更大的分布，用于指示合成纹理中基元的放置位置。因此，这需要我们特别关注纹理的周期性模式。图3-1给出了本章提出的纹理合成方法的整个流程图。

总体来说，本章方法中的纹理合成过程分为以下两个主要阶段：

分析阶段：从给定的一个有限大小的样本纹理中提取基元以及基元的分布。样本纹理中，基元是具有相似视觉特征的区域。已抽取的基元分布通常用于描述基元之间的空间相邻关系。

合成阶段：利用一个有效的生成过程从给定的基元分布和已分割的基元中产生新的纹理图像。合成的分布与从样本纹理中抽取的基元分布具有相似的外观特征，这直接决定了合成纹理的质量。

一旦样本纹理表示成基元的分布，合成过程将不再使用该样本纹理。任意样本纹理分析后，其输出的基元和基元分布可被重复用于多次合成任务。因

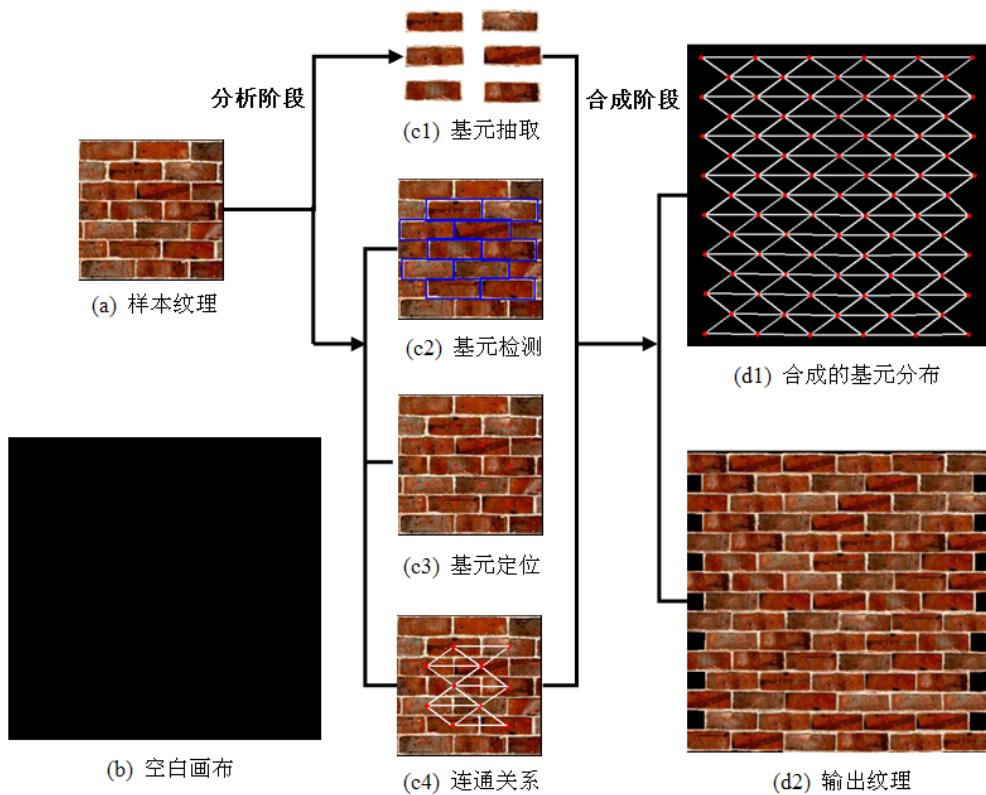


图 3-1 基于基元分布的纹理分析与合成

Fig 3-1 An overview of our texture analysis and synthesis based on the distribution of texels

此，这些分析结果可建立一个数据库，且能够根据每次合成任务任意选择已抽取的基元和基元分布生成新纹理图像。纹理合成过程始于给定初始种子位置以及放置在该位置上的基元，通过依次放置新的基元逐渐向外扩展，从而生成新的、全局的基元分布及纹理图像。其中，每一次输出的基元是样本基元分布中与已合成基元具有最相似邻域的基元。此时，基元的邻域由基元不再是像素点组成。然而，使用本章方法必须满足的前提条件是样本纹理中的每个基元是可辨别的。此外，本章方法还提出了一种基于基元的纹理设计方法，即参照用户给定的简笔画（即初始基元分布），在不局限于用户提供的基元分布的情况下能够产生多样的纹理合成结果。

由上可以看出，本章方法利用基元分布进行纹理合成的优势包括：1) 能够对已抽取的基元分布进行如平移、缩放及旋转等变换，从而引入各种变化；2) 在合成阶段可以提供用户更多的可控性操作。

3.2 周期纹理分析与建模

纹理分析阶段的主要工作包括：分割样本纹理、检测纹理基元和分析基元的空间2D排列。分割纹理是将样本纹理分解成单个的、独立的区域，这些区域就是构成纹理的主要元素，且称之为基元（Texel）。基元检测的目的是为了精确定位样本纹理中基元的位置。在确定所有基元的位置之后，我们通过分析基元之间的空间相邻关系来获得样本纹理的拓扑结构。与现有纹理合成方法不同，本章方法利用基元的分布进行纹理建模。

3.2.1 基元的提取

分割的目的是识别样本纹理中具有相似视觉特征的区域。其中所有划分的区域具有如下属性：一些区域为表示主要纹理元素的基元；而另一些则是由相近颜色形成的同质区域（homogenous zones），即纹理背景模式（the background pattern）。需要注意的是，我们并不使用已抽取的基元来分析样本纹理的拓扑结构，而是用于合成阶段产生新纹理。

在本章方法中，我们采用两类不同的图像分割技术进行基元抽取，包括自动的图像聚类方法和交互的前景对象抠取方法，从而尽可能地处理大范围的由基元组成的纹理图像。基于纹理特征可根据其颜色进行辨别的观察，此时，样本纹理中的基元区域与其他区域形成鲜明的对比。由此，我们采用颜色量化（color quantization）[104]方法在RGB颜色空间上检测这些基元。颜色量化处理主要考虑两个方面内容：树结构调色板（即从样本纹理图像中可获得的颜色集合）的设计和样本纹理图像中每一像素点的颜色到调色板中某一颜色的映射。

已知由给定样本纹理中所有像素点组成的集合 S 以及每一像素点的颜色值 x_s ，调色板设计算法的目标是将集合 S 划分成不相交的 M 个簇，其中 M 为预先定义的调色板的大小。该算法约束集合 S 的划分为二叉树结构，此时，树的每一叶子节点为集合 S 的一个子集，如叶节点 n 表示由像素点子集组成的簇 C_n 。使用该算法生成二叉树结构的关键在于分裂节点以及决定哪个节点在分裂之后能够使得纹理图像和量化后纹理图像之间的总平方误差（the total squared difference, TSE）最小。总平方误差定义为：

$$TSE = \sum_n \sum_{s \in C_n} \|x_s - q_n\|^2 \quad (3-1)$$

其中， q_n 是量化的值，用于表示簇 C_n 的颜色。在决定分裂哪一树节点及分裂的顺序之前，该算法需要定义如下三个簇特征：

$$\begin{cases} R_n = \sum_{s \in C_n} x_s x_s^T \\ m_n = \sum_{s \in C_n} x_s \\ N_n = |C_n| \end{cases} \quad (3-2)$$

其中， R_n 是 3×3 的矩阵； m_n 是一个三分量的向量；一个簇的量化值 $q_n = m_n/N_n$ 为该簇的均值。同时，该算法给出簇的协方差定义：

$$\tilde{R}_n = R_n - \frac{1}{N_n} m_n m_n^T \quad (3-3)$$

一个节点的划分等价于选择两个新的量化级 q_{2n} 和 q_{2n+1} ，即选择一个能够最佳分裂簇颜色的平面，且该平面正交于簇的最大变化方向并穿过簇的均值。此时，该算法使用这个平面分裂当前簇。更具体地说，每一次节点的分裂需要确定一个单位向量 e ，使得能够最大化如下表达式：

$$\sum_{s \in C_n} ((x_s - q_n)^T e)^2 = e^T \tilde{R}_n e \quad (3-4)$$

该等式的解就是对应于 \tilde{R} 的最大特征值 λ_n 的特征向量 e_n 。一旦确定了这个主要特征向量，当前簇 C_n 中的像素点可以按照如下条件分裂为两个新的簇 C_{2n} 和 C_{2n+1} ：

$$\begin{cases} C_{2n} = \{s \in C_n : e_n^T x_s \leq e_n^T q_n\} \\ C_{2n+1} = \{s \in C_n : e_n^T x_s > e_n^T q_n\} \end{cases} \quad (3-5)$$

由上可以看出，最大特征值 λ_n 可以用来作为节点分裂的测量标准。本章方法采用上述颜色量化算法进行基元自动抽取的步骤为：

1. 采用高斯滤波平滑样本纹理去除图像中的噪声；
2. 设置 $C_1 = S$ ，并计算簇的统计特征 R_1 ， m_1 和 N_1 ；
3. 迭代如下步骤 $M - 1$ 次：
 - 3.1 找到具有最大特征值 λ_n 的叶子节点 n ；
 - 3.2 使用公式 (3-5) 产生两个新的节点 $2n$ 和 $2n + 1$ ；
 - 3.3 使用公式 (3-2) 为每一新节点计算相应的簇统计特征。

4. 在步骤3结束后，将每个簇内的像素点的颜色替换为所在簇的量化颜色。

图3-2给出使用颜色量化方法实现的基元自动抽取，第一行为样本纹理，第二行为实验结果。对于基元容易辨别的纹理图像，该方法通常能够获得足够好的分割结果，参见图3-2 (a) 和 (b)。在这两个例子中，纹理中的基元和纹理背景具有较明显的差异。然而，对于图3-2 (c) 中给定的样本纹理，这种自动的颜色聚类方法并不能很好地分割出单个独立的基元。此时，由于基元中存在细微光照和颜色变化的区域是很难用肉眼进行区分，因此，基元被进一步分割成多个子块区域。另外，我们在分割之前通常会使用平滑操作去除纹理图像中的噪声，然而，平滑操作可能模糊纹理中的主要纹理特征。事实上，对于结构性强的纹理，我们需要获得基元的边界信息，使得这些重要的纹理特征能够在合成的纹理中得到恢复。也就是说，我们需要抽取出完整的基元，包括基元的整个区域及其边界轮廓。

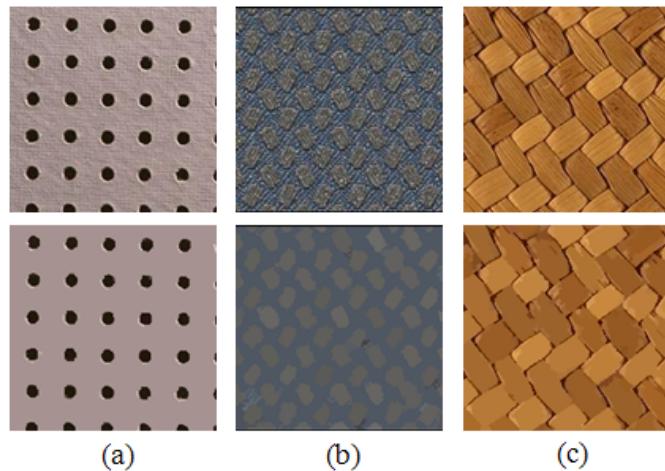


图 3-2 自动的纹理图像分割

Fig 3-2 Automatic texture segmentation using color quantization

针对上述颜色量化方法存在的不足，第二种方法是采用交互式图像抠取方法 [105–107]进行基元抽取。其中，Lazy Snapping和GrabCut这两种图割方法仅需要提供少量的用户交互，即标注样本纹理中绝对的基元和背景区域，来获得令人满意的分割结果。此时，基元的抽取可以看成是一个二值标签问题，也就是说，用‘0’标记纹理背景和‘1’标记基元。假定样本纹理为图 $g = \langle v, \varepsilon \rangle$ ，其中 v 是所有像素点的集合，且称之为节点集； ε 是连接相邻节点的边的集

合。通过最小化吉布斯能量函数 $E(X)$ (Gibbs energy) [108] 可获得纹理图像中各像素点的标签 $X = x_i$:

$$E(X) = \sum_{i \in v} E_1(x_i) + \sum_{(i,j) \in \varepsilon} E_2(x_i, x_j) \quad (3-6)$$

其中, $E_1(x_i)$ 是数据能量项, 表示节点 i 的标记是 x_i 时的代价; $E_2(x_i, x_j)$ 是先验能量项, 表示相邻节点 i 和 j 的标记是 x_i 和 x_j 时的代价。

约定用户标注的基元和背景区域所覆盖的像素点分别为前景种子集 F 和背景种子集 B 。在公式 (3-6) 中, E_1 判定图中任一节点的颜色相似性, 用于指示该节点 (像素点) 属于基元还是背景区域。为了计算 E_1 , Lazy Snapping 方法采用 k-means 聚簇方法统计种子集 F 和 B 中像素点的颜色信息。此时, 已标注前景和背景中各簇的均值颜色分别表示为 $\{K_n^F\}$ 和 $\{K_m^B\}$, 其中 n 和 m 分别为种子集 F 和 B 聚类后簇的总数。对于样本纹理中未标注的任一像素点, 它的颜色 $C(i)$ 到前景簇的最短距离定义为 $d_i^F = \min_n \|C(i) - K_n^F\|$, 同理, $d_i^B = \min_m \|C(i) - K_m^B\|$ 为到背景簇的最短距离。因此, $E_1(x_i)$ 可由如下公式定义:

$$\begin{cases} E_1(x_i = 1) = 0 & E_1(x_i = 0) = \infty \\ E_1(x_i = 1) = \infty & E_1(x_i = 0) = 0 \\ E_1(x_i = 1) = \frac{d_i^F}{d_i^F + d_i^B} & E_1(x_i = 0) = \frac{d_i^B}{d_i^F + d_i^B} \end{cases} \quad (3-7)$$

其中, $U = v \setminus \{F \cup B\}$ 是样本纹理中未标注的区域。在公式 (3-7) 中, 前两项等式确保种子集 F 和 B 中各像素点的标签与用户的标注是一致的; 而第三项等式则保证节点的标签为具有相似颜色信息前景或背景的标签。 E_2 表示沿基元边界梯度的能量, 通常定义为相邻节点之间梯度的函数:

$$E_2(x_i, x_j) = |x_i - x_j| \cdot g(C_{ij}) \quad (3-8)$$

其中, $g(\xi) = \frac{1}{\xi+1}$ 和 $C_{ij} = \|C(i) - C(j)\|^2$ 为相邻两像素点之间的颜色差异; $|x_i - x_j|$ 用于捕获基元边界处的梯度信息。根据用户的输入, 该方法采用基于 Boykov 和 Jolly 提出的最大流算法 (the max-flow algorithm) [109] 抽取样本纹理中的所有基元, 并同时获得他们的边界信息。此外, 该方法是一个从粗到精 (coarse-to-fine) 的过程, 即在分割的结果上根据用户提供的新的前景和背景标注信息及再次使用优化方法 [109] 可逐步获得准确的基元边界。然而, 对于具有许多细小结构特征的纹理, 该方法则需要大量的用户交互。与 Lazy Snapping 图

割方法不同，GrabCut方法采用高斯混合模型定义公式（3-6）中各能量项，详细的定义可参见文献 [106]。这种方法的主要特点在于仅需要用户提供包围基元的最小矩形框标记基元即可进行基元抽取，且该方法采用方法 [109]进行的迭代优化过程能够极大地减少用户的交互。

本章方法结合上述Lazy Snapping和GrabCut这两种图割方法的优点进行样本纹理的基元抽取，整个过程描述如下：

1. 在样本纹理中交互地用矩形框选择当前待抽取的基元。采样GrabCut图割方法抽取出该基元区域；
2. 如果步骤1的输出结果不能满足分割的质量，则采用Lazy Snapping图割方法进行迭代地优化直至获得满意的分割结果；
3. 重复执行步骤1和步骤2，直到抽取出样本纹理中的所有基元。

由上可以看出，同时使用Lazy Snapping和GrabCut图割方法进行基元提取仍然需要大量的人工交互。然而，值得特别注意的是仅使用GrabCut图割方法进行基元提取不能立即获得满意的分割结果。如图3-3中每一分割结果的顶行所示，分割的基元区域内除包含当前抽取的基元外，还包含了其他基元的部分区域。因此，我们不得不继续使用Lazy Snapping图割方法对分割结果进行细微的完善。最终的分割结果如下一行所示。此时，为避免负载，我们在图3-3中仅显示部分已抽取的基元。

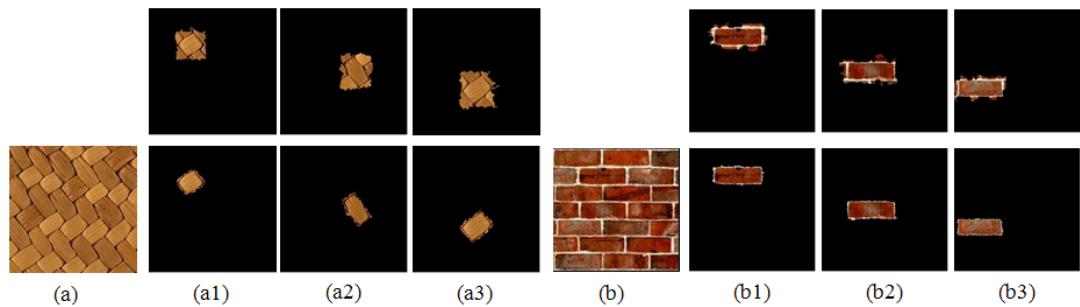


图 3-3 交互的基元抽取

Fig 3-3 Segmenting sample textures into texels interactively

此外，一些商业的抠图工具，如Adobe Photoshop软件中提供的磁性套索工具（Magnetic Lasso）和Microsoft PhotoDraw软件中提供的边检测剪刀工具（edge-detecting scissor），都能用于抽取基元的区域及其边界。虽然这些软件需

要进行繁琐的用户手工交互，但是它们能很好地捕获样本纹理中细小的或模糊的结构特征。事实上，在合成过程中当使用泊松差值方法放置基元时，我们仅需要勾勒出每一个基元，此时，可省去基元抽取操作。详细的实现细节将在第3.3.4小节中讨论。在本章方法中，我们允许使用上述三种方式进行基元抽取。

3.2.2 基元检测、定位及分类

这个阶段的主要目标是找出样本纹理中所有的基元，并确定它们在纹理中的位置。Leung和Malik [110]已经提出了一种基于仿射变形的相似性度量方法来检测图像中重复的场景元素，同时该方法也适用于样本纹理中基元的检测。由于纹理指的是具有有限模式的一类特殊图像，因此，我们需要辨别样本纹理中“完整的”和“不完整的”基元。如果某一基元中大部分像素点位于纹理图像的边界或其边界是不封闭的，则该基元是不完整的。我们称所有完整的基元为真基元（待检测的），而不完整的基元为伪基元。本章方法将样本纹理划分成相互重叠的子块，从而用于构建检测所有真基元的候选基元集。这个过程被看成是一个必不可少的初始化操作。此时，我们在每一个子块内计算二阶矩（second moment matrix）：

$$\sum w = \sum_{i=1}^M \sum_{j=1}^N \nabla I(i, j) \nabla I^T(i, j) \quad (3-9)$$

其中， $\nabla I(i, j)$ 是梯度向量； M 和 N 表示预先定义的块大小。二阶矩通常反映子块内的空间亮度变化，其特征值 k_1 和 k_2 （约定 $k_1 > k_2$ ）表示在两个主元方向的能量幅度。若子块内呈现2D的结构， k_1 和 k_2 则具有较大幅度；如子块内呈现细微变化， k_1 和 k_2 的值则很小。通过使用这两个特征值，我们可以进一步量化子块内的空间亮度变化。事实上，上述过程构建的候选基元集中存在许多因错误检测而产生的伪基元。为了排除这些伪基元，方法 [110]通过采用基于仿射变形的块匹配方法进一步确定所有的真基元。给定任意两个子块和，它们之间具有较好的对应关系当且仅当以下能量公式达到最小：

$$E(x) = \sum [P_1(x) - P_2(Ax + d)]^2 \quad (3-10)$$

$$\{A, d\} = \arg \min_{\{A, d\}} E(x) \quad (3-11)$$

其中, $P_1(x)$ 和 $P_2(x)$ 为待匹配的两个块中像素点在位置 x 处的颜色(灰度)值; A 是一个数组, d 是一个向量, 表示块 $P_1(x)$ 和 $P_2(x)$ 之间的线性空间变换。使用上述线性估计, 我们通过求解一组线性方程组来估计增量 ΔA 和 Δd 。此时, $\Delta A = A - I$, I 为单位矩阵; 及 $\Delta d = d$ 。

虽然方法 [110]能够检测出样本纹理中所有的真基元, 但是该方法不能对检测出来的基元进行自动地分类。也就是说, 样本纹理中可能包含不同种类的基元, 它们在形状、大小、颜色等方面具有一定的差异。然而, 分析不同类别基元之间的相关性在合成过程中是非常重要的。因此, 在检测所有真基元之后, 结合基元抽取获得的各基元边界信息, 本章方法采用形状上下文描述子(shape context) [111, 112]分类基元。任意两基元的差异可根据他们的形状进行测量, 而基元的形状可由一组从基元边界上选取的采样轮廓点表示(参见图3-4 (b) 中绿色的点)。给定由 n 个轮廓点组成的点集 $P = \{p_1, \dots, p_n\}$, 考虑产生于形状上的任意一轮廓点到其他轮廓点的一组向量, 这 $n - 1$ 个向量表示相对于参照轮廓点的整个形状的构造。更具体地, 对于形状上某一轮廓点 p_i , 我们计算剩余 $n - 1$ 个点的相对坐标的直方图 h_i :

$$h_i = \#\{q \neq p_i : (q - p_i) \in bin(k)\} \quad (3-12)$$

这个直方图定义为轮廓点 p_i 的形状上下文描述子, 用于描述该轮廓点相对于形状上其他所有轮廓点在空间直方图中的分布。如图3-4 (c) 所示, 直方图的桶状结构(bins)在极-对数空间(log-polar space)中均匀分布, 使得该形状描述子更敏感于邻近位置的轮廓点而不是那些距离较远的轮廓点。

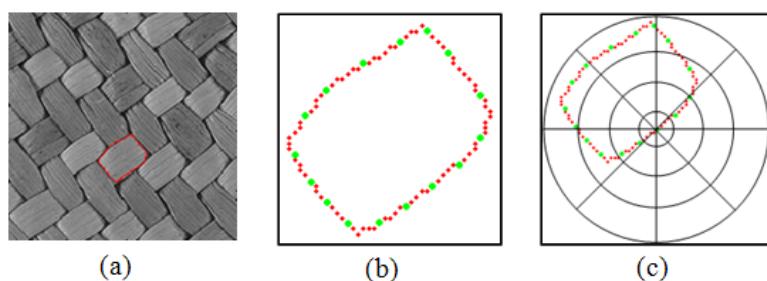


图 3-4 形状上下文描述子

Fig 3-4 Shape context descriptor

为方便后续计算, 我们给每个轮廓点 q_j 附加一个单位切向量, 即为该轮廓点的方向。在每个桶状结构中, 我们累加落入该桶状结构中所有轮廓点的切向

量。此时，某一轮廓点 p_i 的直方图 h_i 改写为：

$$\hat{h}_i^k = \sum_{q_j \in Q} t_j, \quad Q = \{q_j \neq p_i : (q_j - p_i) \in bin(k)\} \quad (3-13)$$

当比较两个轮廓点之间的形状描述子时， K 维的桶状直方图可以转化为 $2K$ 维的特征向量 \hat{v}_i ，我们使用 L_2 范式度量它们之间的差异：

$$\hat{v}_i = \langle h_i^{1,x}, h_i^{1,y}, h_i^{2,x}, h_i^{2,y}, \dots, h_i^{K,x}, h_i^{K,y} \rangle \quad (3-14)$$

$$d_{SC}(\hat{h}_i, \hat{h}_j) = \|\hat{v}_i - \hat{v}_j\|_2 \quad (3-15)$$

其中， $h_i^{1,x}$ 和 $h_i^{1,y}$ 分别是 \hat{h}_i^k 在 x 和 y 上的分量。公式(3-14)称之为量化的形状上下文描述子。使用该描述子进行基元分类的好处是它具有平移、缩放和旋转不变性，甚至在细微几何变形、遮挡等情况下运算也是稳健的。

如上所述，对给定样本纹理中基元的检测和分类过程描述如下：

1. 构建一个基元子集。其中每个元素是不同类别基元中的任意一个基元，这些元素在各类基元中具有显著的特征。我们用包围该基元的最小矩形框进行标记，并记录该矩形框的大小。子集中每个基元的类别号(class id)为它们在子集中的索引。

2. 构建候选基元集。根据基元子集中基元的大小将样本纹理划分成多个相互重叠的子块，并使用公式(3-9)计算每个子块的二阶矩 $\sum w$ 及其特征值 k_1 和 k_2 。某子块内的基元是可能的候选真基元当且仅当定义的能量比率 $r = k_1/k_2$ 满足如下条件等式：

$$W(i, j) = \begin{cases} \text{true}, & \text{if } r \leq T \\ \text{false}, & \text{if } r > T \end{cases} \quad (3-16)$$

其中， $W(i, j)$ 表示左下坐标 (i, j) 在样本纹理中的子块； T 是用户定义的阈值，不同的样本纹理具有不同的阈值。通常情况下，其取值范围在3到5之间。该步骤直到计算了样本纹理中所有子块的二阶矩时结束。

3. 检测所有候选基元中的真基元并确定它们在样本纹理中的位置。使用仿射变形估计(公式(3-10)和(3-11))测量空间相邻的两子块之间的对应关系，我们说两子块是真基元当他们之间的相似度 E_{sim} 小于或等于用户定义的阈值 τ 。给定两个子块 $P_1(x)$ 和 $P_2(x)$ ，相似度由如下公式计算得到：

$$E_{sim} = \frac{\delta_E}{\delta_I} \quad (3-17)$$

$$\delta_E = \sqrt{\sum (E(x) - \bar{E}(x))^2} \quad (3-18)$$

$$\bar{E}(x) = \sum [P_2(x) - P_1(A'x + d')]^2$$

$$\delta_I = \sqrt{\sum (P_1(x) - P_2(x))^2} \quad (3-19)$$

其中， δ_E 是两子块 $P_1(x)$ 和 $P_2(x)$ 之间的变化误差； δ_I 是两子块之间的颜色值（灰度）差异。阈值 τ 的取值范围定义为 $[0, 1]$ 。该步骤的输出是子块集合且对应样本纹理中的所有真基元。子块的中心即为基元在样本纹理中的位置。

4. 分类所有已检测的真基元。将每一真基元与已构建基元子集中的基元逐一比较，并计算它们之间的形状差异。已知两基元的轮廓点集 S_k 和 S_l ，我们同时考虑轮廓点集 S_k 到 S_l 的距离和轮廓点集 S_l 到 S_k 的距离来测量两基元之间的差异。度量公式定义如下：

$$D(S_k, S_l) = \frac{1}{N_k} \sum_{i \in N_k} \|v_i - v_{j^*}\|_2 + \frac{1}{N_l} \sum_{i \in N_l} \|v_j - v_{i^*}\|_2 \quad (3-20)$$

其中， N_k (N_l) 表示集合 S_k (S_l) 中轮廓点的总数； v_i (v_j) 是集合 S_k (S_l) 中第 i (j) 个轮廓点的形状上下文特征向量；集合 S_l 中第 j^* 个轮廓点与集合 S_k 中第 i^* 个轮廓点对应，此时， $j^* = \arg \min_j \|v_i - v_{j^*}\|_2$ 。同理， $i^* = \arg \min_i \|v_j - v_{i^*}\|_2$ 。通过确定具有最小差异的基元子集中基元，当前所选基元与该基元具有相同类别。

如果样本纹理中的所有基元仅有一类，那么可省去步骤4。步骤1中的操作是为了给基元检测过程提供先验信息，即子块的初始大小，使得划分的子块尽可能地包围每个真基元。此外，通过使用仿射变形模型，我们可以迭代地改变子块的大小直到匹配误差（公式（3-17））停止减小或达到最大迭代次数，由此确定每个基元的最小包围盒。这有助于获得基元在样本纹理中的准确位置。然而，对上述过程的强约束条件是样本纹理中的基元是容易辨别的。在某些情况下，基元检测的结果中可能存在伪基元或包围真基元的子块的大小并不是最小包围盒。由于方法 [110]本身并不具备自动纠错能力，因此，我们不得不进行手动调整来精确定位样本纹理中每一基元。上述过程的执行时间取决于样本纹理中基元的总数目。基于这一点，我们不排除通过手动地拖拽最小包围盒来选择每一个真基元。图3-5给出部分基元检测和分类的结果：第二列显示的是已检测出的所有真基元；最后一列显示它们在样本纹理中的位置。其中，我们使用不同的颜色表示基元的类别。

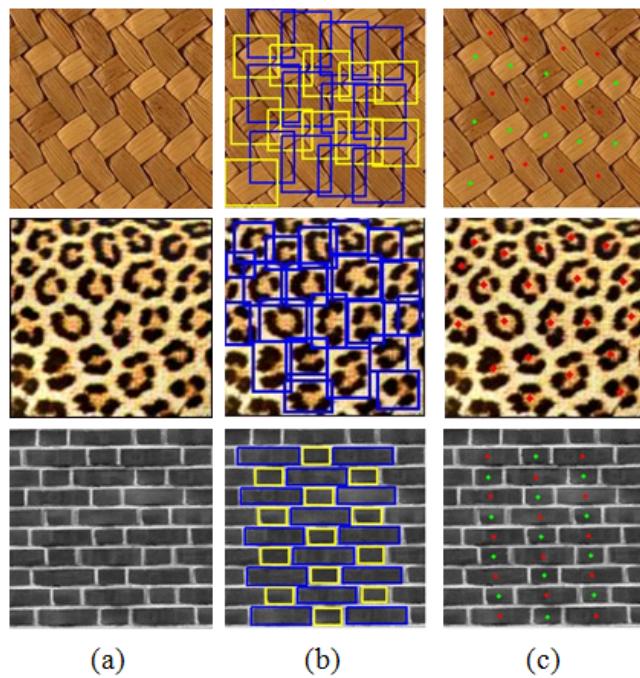


图 3-5 基元检测和分类

Fig 3-5 Texels detection and classification

3.2.3 基元空间分布建模

与现有基于像素的或基于块的纹理合成方法不同，这些方法都是在图像空间中进行操作，而本章方法则考虑每一基元的邻域特征，即获取与每一基元最邻近的所有基元的集合。在定位样本纹理中所有基元之后，我们需要分析基元的空间分布（或空间排列）。为了抽取这种邻域特征，这个阶段的主要任务是构建每一个基元与其相邻基元之间的空间相邻关系（该基元与其相邻基元可能具有相同的类别，也有可能具有不同的类别），称之为基元之间的连通关系或样本纹理的拓扑结构。

在样本纹理中，我们用基元的位置信息代替每一个基元，这些位置构成一个离散的点集。已知这个离散点集，我们采用三角剖分（Delaunay triangulation）方法来建立基元之间的连通关系。为了避免产生倾斜边（图3-6中第二列第一行结果中所示的虚线边），我们约束三角剖分过程中产生的每一三角形的最大内角的角度必须小于120度。当基元的分布是规则分布或者近似规则分布时，即使上述的约束条件是满足的，三角剖分可能构建不理想的基元连通关

系（图3-6中第二列最后二行结果所示）。此时，将无法保持基元分布的潜在规律性。为了解决这个问题，在使用三角剖分构建基元的连通关系之前，我们首先将可能引起二义性的三角边（连接两相邻基元的边）标记出来，用于阻止三角剖分过程中这些边的产生，然后即可自动地构建基元之间的连通关系。图3-6的第三列给出部分样本纹理的基元连通关系结果。一旦构建了基元之间的连通关系，我们将记录每个基元的相邻基元、三角网格中边的长度和每个三角形三内角角度等数据。这些数据和已抽取的样本纹理的拓扑结构将在后续纹理合成阶段中使用。

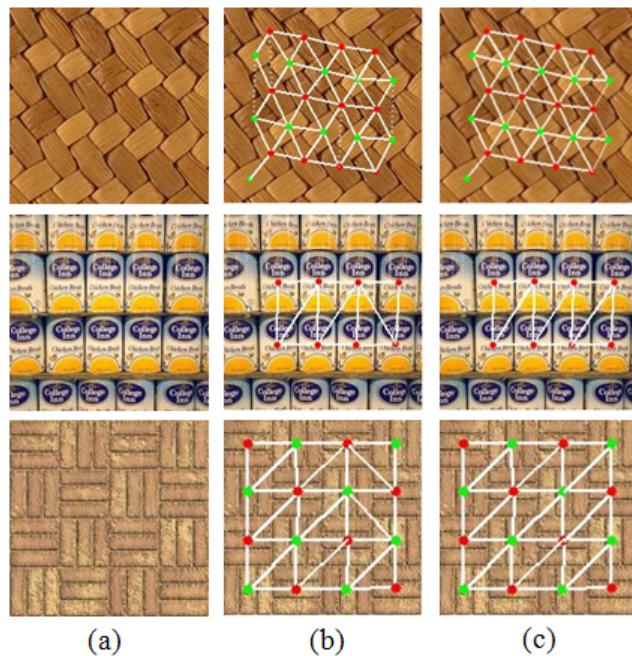


图 3-6 基元间连通关系的构建

Fig 3-6 Construction of connectivity among texels

3.3 基于基元分布的纹理合成

3.3.1 纹理合成算法概述

纹理合成过程起始于在待合成纹理中放置单个基元在用户指定的初始位置上，然后通过邻域比较（neighborhood comparisons）逐步向外扩张基元，从而合成新的纹理图像。需要注意的是，后续文中基元的位置统称为“种子”。本

章方法在进行纹理合成时还需考虑如下两个方面：1) 在已构建的基元连通关系中，如果每一真基元不具备足够多的相邻基元，那么这种情况将影响种子增长。2) 由于已抽取的基元可能具有不同的形状、大小及方向，因此在放置基元到待合成纹理中，基元之间势必会产生重叠区域或空洞。

给定样本纹理、已抽取的基元集合、构建的基元连通关系（用于合成过程中的参照连通图）、用户指定大小的画布（可能为空白图像或具有纹理背景的图像）以及最大种子数目，纹理合成阶段的整个流程如下：

1. 在参照连通图中，逐一检查每个基元的邻域总数。如果多数基元没有足够多的相邻基元，我们则根据已知的基元连通关系再次扩张现有基元分布。同时，需更新基元之间的连通关系。此时，每一基元相邻基元是充足的。否则，直接跳转到下一步。

2. 首先在给定的参照连通图中选取具有最多相邻基元的基元，并指定画布的中心位置为待合成纹理的初始种子。然后，将所选基元放置在初始种子上。添加该基元的所有相邻基元的位置到画布中（按照它们在参照连通图中的位置进行添加），并分别记录这些种子的序号（sub-id）和类别号（class-id）。最后，在这些种子之间构建初始的连通关系，且称之为目标连通图。

3. 在画布中，选择一个与初始种子之间具有最短距离的种子作为当前的目标种子。一般情况下，可能存在多个这样的候选种子，此时我们选择具有最高优先级的种子（种子的优先级根据种子的类别确定）。

4. 从已抽取的基元集中找到一个最佳基元替换当前所选的目标种子。所选的最佳基元是其邻域与当前所选目标种子具有最相似邻域的基元。然后，将所选最佳基元放置在当前所选目标种子上。

5. 对照当前所选最佳基元的邻域与目标种子的邻域，其中不存在的部分将作为新的候选种子。如果这些候选种子允许添加到画布中（例如还未超出允许的最大种子数目），则添加这些候选种子作为新的种子，并添加这些新种子与目标种子之间的边。此时，在所有种子之间重新构建连通关系，更新目标连通图。

6. 重复步骤3到步骤5，直到所有种子上均放置了基元。最后输出新的纹理图像。

上述合成过程中的实现细节将在后续章节中进行讨论。

3.3.2 基元分布的扩张

样本纹理通常仅由有限数量的基元组成，包括真基元和伪基元。在合成过程中，我们通常去除所有伪基元，且仅建立所有真基元之间的连通关系。丢弃所有伪基元的原因是它们在样本纹理中的位置是不准确的，而且它们缺失的部分区域也使得我们无法估计其真实大小。因此，在所构建的基元连通关系中，某些真基元可能没有足够多的相邻基元，这使得该连通关系无法提供一个信息丰富的（information rich）周期性模式（periodic pattern），它能够反映样本纹理中基元的排列规则。为此，在进行纹理合成之前，我们将辨别信息丰富的和信息匮乏的（information poor）基元连通关系。

已知一个基元及其邻域，假定邻域中的每个基元看成是一个节点，相邻两个节点由一条边连接。邻域中的所有基元组成一个环形结构（ring shape）当且仅当边的总数等于节点的总数。在已构建的基元连通关系中，若每类基元中如果存在一个或者多个基元，其邻域中的基元形成上述的环形结构，我们说该连通关系是信息丰富的。否则，构建的连通关系是信息匮乏的。为了避免产生信息匮乏的基元连通关系，我们利用参照连通图中已知的基元相邻关系来重构一个新的连通关系，且与参照连通图保持一定的外观相似性。此时，在样本纹理中可恢复伪基元与真基元之间的空间相邻关系。

重构基元连通关系的思路是：在样本纹理中，首先确定需要添加哪些新的基元，并计算这些基元的位置；然后利用三角剖分方法在已有的基元和新添加的基元之间建立相应的边。这个过程称为“连通关系的扩张”。事实上，这个扩张过程是半自动的过程，因为需要用户事先提供每一个新基元的先验知识。该先验知识定义为一个五元组 (r_1, r_2, t_1, t_2, C) ，其中， r_1 和 r_2 表示两个源始基元的序号； t_1 和 t_2 表示两个目标基元的序号（一端基元的信息是已知 t_1 ，而另一端基元的信息是未知 t_2 ，即为待添加的新基元，并用‘0’进行标记）； C 表示新基元的类别。假设源始基元对与目标基元对是一一对应的，我们根据它们之间的相对位置关系来估计目标基元对中新添加基元的位置。已知 r_1 ， r_2 和 t_1 是参照连通图中已存在的基元， t_2 是待添加的新的基元（参见图3-7（c）），相对位置关系定义如下：

$$\begin{cases} |X_{r_1} - X_{r_2}| = |X_{t_1} - x_{t_2}| \\ |Y_{r_1} - Y_{r_2}| = |Y_{t_1} - y_{t_2}| \end{cases} \quad (3-21)$$

其中， (X_i, Y_i) 表示已存在基元的位置坐标。 (x, y) 是新添加基元的位置坐标，

可通过求解上述线性方程组计算得到。同理，给定其他待添加新基元的先验知识，它们的位置信息即可自动获得。图3-7中给出信息缺乏的基元间连通关系的扩张结果，其中图3-7 (b) 中每个基元的邻域均不构成环形结构，图3-7 (d) 为扩张后重构的基元连通关系。需要指出的是，从样本纹理中抽取的基元仍然是那些真基元。

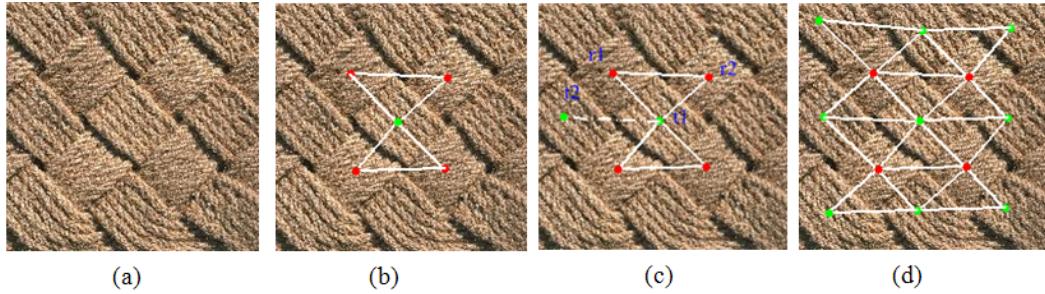


图 3-7 基元间连通关系的扩张

Fig 3-7 Extending connectivity from the arrangement of texels

3.3.3 最佳匹配基元

在待合成纹理中，假定 e^{tar} 是当前所选种子以及 $\omega(e^{tar})$ 是其所有相邻种子组成的邻域（其中部分种子可能已被基元覆盖）。在本章方法中，我们采用邻域比较方法（neighborhood comparisons）从已抽取的基元集合中找到与 e^{tar} 匹配的最佳基元 e^{ref} ，并将它放置在当前所选的种子上。需要指出的是，样本纹理中的所有基元在邻域比较过程中也看成种子。此时，从给定的参照连通图中抽取的最佳基元的邻域 $\omega(e^{ref})$ 相似于邻域 $\omega(e^{tar})$ 。不同于现有的基于局部邻域比较的纹理合成方法，本章方法中所用的“邻域”由基元而不是像素点组成。两个邻域 $\omega(e^{ref})$ 和 $\omega(e^{tar})$ 是相似的指的是这两个邻域的相对基元在位置和类别上的差异是最小的，而不再是它们在像素点颜色上的最小差异。

当进行位置之间的匹配时，我们允许邻域 $\omega(e^{ref})$ 中所有种子分布在以 e^{ref} 为中心的局部坐标系中，同理标记 $\omega(e^{tar})$ 和 e^{tar} 。中心种子及其所有相邻种子的分布如图3-8 (a) 和 (b) 所示。局部坐标系分为四个象限，根据相对位置关系可以确定邻域中每一个种子的象限。如图3-8 (a) 所示，种子 e_1^{ref} 属于第一象限是由于 e_1^{ref} 的x-轴坐标大于 e^{ref} 的，以及 e_1^{ref} 的y-轴坐标大于 e^{ref} 的。如果某一象限中有多个种子，我们使用象限角 θ （定义为x-轴与

边(e_i^{ref}, e^{ref})之间的夹角) 进一步估计这些种子顺序(象限角是具有方向的角)。在这种方式下, 我们分别逆时针地排序邻域 $\omega(e^{ref})$ 和 $\omega(e^{tar})$ 中的各种子 $e_i^{ref} \in \omega(e^{ref})$ 和 $e_j^{tar} \in \omega(e^{tar})$ 。将邻域 $\omega(e^{tar})$ 中的每个种子依次与邻域 $\omega(e^{ref})$ 中的各种子进行匹配, 以 e_j^{tar} 和 e_i^{ref} ($j = 1, i = 1$) 作为的初始种子对, 递增地匹配后续种子对 e_{j+1}^{tar} 和 e_{i+1}^{ref} , 即可建立邻域 $\omega(e^{tar})$ 和 $\omega(e^{ref})$ 之间的一个匹配 f 。依次更换初始种子对的索引 $j = 1$ 和 $i = 2 : n$, 可建立候选匹配集 F ($f_i \in F, i = 1 : n$), 其大小取决于邻域 $\omega(e^{tar})$ 中种子的总数目 n 。

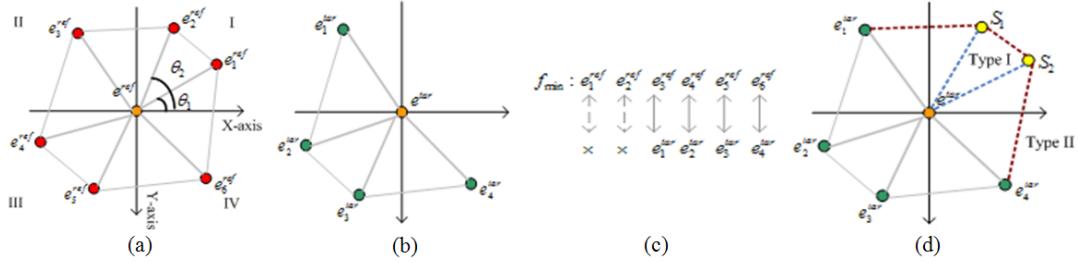


图 3-8 邻域比较过程

Fig 3-8 The neighborhood comparison procedure

一旦获得了匹配集 F , 基于邻域 $\omega(e^{tar})$ 和 $\omega(e^{ref})$ 之间差异的误差函数定义如下:

$$Error(\omega(e^{ref}), \omega(e^{tar})) = \min_{f \in F} err(\omega(e^{ref}), \omega(e^{tar})) \quad (3-22)$$

$$\begin{aligned} err(\omega(e^{ref}), \omega(e^{tar})) &= \\ &\sum_{i,j \in F} w_1 q(e_i^{ref}, e_j^{tar}) + w_2 \theta(e_i^{ref}, e_j^{tar}) + w_3 L(e_i^{ref}, e_j^{tar}) \\ &+ w_4 Cid(e_i^{ref}, e_j^{tar}) + w_5 subid(e_i^{ref}, e_j^{tar}) \end{aligned} \quad (3-23)$$

其中, 如果 e_i^{ref} 和 e_j^{tar} 所在象限不同, 则 $q(e_i^{ref}, e_j^{tar})$ 的值为1, 否则为0。 $\theta(e_i^{ref}, e_j^{tar}) = |\theta_i^{ref} - \theta_j^{tar}|$ 为象限角 θ_i^{ref} 和 θ_j^{tar} 的绝对差异。 $L(e_i^{ref}, e_j^{tar}) = |L_i^{ref} - L_j^{tar}|$ 是边(e_i^{ref}, e_j^{ref})和边(e_i^{tar}, e_j^{tar})在长度上的绝对差异。上述三项用于测量邻域 $\omega(e^{tar})$ 和 $\omega(e^{ref})$ 中各种子在位置之间的差异。如果 e_i^{ref} 和 e_j^{tar} 的类别属性不同, $Cid(e_i^{ref}, e_j^{tar})$ 的取值为1, 否则取值为0。这一项用于测量基元之间在类别属性上的差异。如果 e_i^{ref} 和 e_j^{tar} 的序号相同, 那么用于避免基元重复出现的测量项 $subid(e_i^{ref}, e_j^{tar})$ 取值为1, 否则取值为0。公式 (3-23) 中的权重系数的取值为: $w_1 = 50$, $w_2 = 1$, $w_3 = 1$, $w_4 = 50$ 和 $w_5 \in [20, 1000]$ 。当然, 对于不同的

样本纹理，这些权重的取值可以不同。在与参照连通图中所有种子的邻域匹配之后，具有最小误差的种子 e^{ref} 即为我们当前所选的种子 e^{tar} 的最佳基元。需要指出的是，种子之间的匹配仅在同类基元中进行。此外，我们约束所选邻域 $\omega(e^{ref})$ 中的种子数目大于邻域 $\omega(e^{tar})$ 的，从而能够产生新的种子。然而，我们发现参照连通图中的环形结构可能被重复使用，主要是由于样本纹理中有限的基元数目使得本章算法总是选择具有环形结构的基元进行邻域比较。具体实现中，在找到最佳基元之后，我们可以从与目标种子具有相同类别的基元子集中随机地选择一个基元作为最佳基元，但需要约束最初确定的最佳基元的序号以及已合成基元的分布保持不变。本章算法在同类基元子集中允许随机地选择基元（不同的基元具有不同的颜色和亮度），这使得合成的纹理图像具有更自然的外观。

3.3.4 基元的放置

基元放置的主要工作是将所选最佳基元替换为当前所选目标种子。本章处理的近似规则纹理分为两大类：一类由基元和纹理背景组成；另一类仅由基元组成。为了合成这两类纹理，我们采用两种不同的方式将所选最佳基元放置到画布（待合成纹理）的某一区域 Ω_t 中。该区域定义为以当前所选种子为中心且与所选最佳基元同大小的区域。

在合成第一类样本纹理之前，本章方法需要提供一张足够大的且与样本纹理具有类似纹理背景模式的画布。由于样本纹理中的纹理背景既可能是随机纹理，也可能是结构纹理，因此，我们首先从样本纹理中剪裁出一小块纹理背景区域，然后采用图像拼接的纹理合成方法 [28]合成该画布。将基元放置到画布中时，我们采用泊松插值(Poisson Interpolation)方法 [113]实现基元与纹理背景图像的无缝融合。与基元抽取下获得基元的精确边界不同，此时，仅需要用户在样本纹理中标记出当前所选最佳基元的区域 Ω_r 。该方法参照区域 Ω_r 在区域 Ω_t 中确定待放置最佳基元的边界，根据这个边界条件，通过求解泊松方程将样本中标记的基元区域插值到画布中。本章方法采用泊松插值方法的优势在于：该方法能够避免在基元与纹理背景之间产生不连续的颜色传递，使得合成纹理中基元到纹理背景的过渡是自然平滑的。

根据第二类样本纹理具有的特征以及提供的空白画布，我们允许直接拷贝所选最佳基元到画布中。由于基元的形状和大小并不是完全一致的，当逐一地

添加基元到画布中时，基元之间可能产生重叠区域或者空洞。基元之间的过分重叠可能模糊基元的边界，导致重要纹理特征的丢失；而大的空洞则无法保证产生连续的纹理特征。因此，我们处理基元间产生的重叠区域和空洞的方式是：已知用户约束的基元间重叠区域的最大宽度为 W_{ov} ，当前所选基元与已合成基元间产生的重叠区域的宽度为 E_{ov} ，如果 E_{ov} 远大于 W_{ov} ，为避免过分重叠则缩小当前所选基元；反之，为避免空洞的产生则放大当前所选基元。此时，基元的大小（初始缩放因子 $S = 1$ ）以增量为 $\Delta S = 0.05$ 进行反复调整，直到基元间重叠区域的宽度 E_{ov} 被限定在 $[0, W_{ov}]$ 之内或者达到了用户给定的最小/最大缩放系数，这样即可避免产生过分重叠和缩小空洞的尺寸。

将已缩放的所选最佳基元拷贝到画布中时，如果基元之间仍然有重叠，我们则采用羽化方法（feathering）[30]对该基元与画布中已合成的相邻基元进行拼接处理。混合的重叠区域内像素点的值可由公式 $p_i = \alpha p_{new}^i + (1 - \alpha)p_{old}^i$ 计算得到，其中 p_{new}^i 表示当前所选最佳基元的重叠区域内的像素， p_{old}^i 表示已合成基元的重叠区域内的像素， $0 < \alpha < 1$ 的值取决于重叠区域内各像素到其边界的距离。此外，本章方法也可以使用基于图割的拼接方法[34]处理基元间的重叠区域，从而更好地进行纹理特征的传递。而对纹理合成后画布中仍然存在的空洞，本章方法则使用基于样例的图像修补算法[114]对其进行填充。类似于该方法的修补过程，我们首先在样本纹理中搜索与画布中带修补子块 ψ_q 最相似的子块 ψ_p ，此时，两子块之间相对位置的已知区域部分的距离最小。相似性度量公式 $d(\psi_p, \psi_q)$ 定义为两子块中已知区域内像素点颜色的平方差之和。然后将子块 ψ_p 对应于子块 ψ_q 的带修补区域中的像素逐一地拷贝到子块 ψ_q 的未知区域，从而实现空洞的填充。

3.3.5 种子的局部增长

在样本基元集中找到最佳匹配基元并用该基元替换当前所选目标种子之后，接下来的主要工作是在画布中添加新的种子，其目的是为了给后续选择的最佳匹配基元提供相应的位置，使得基元能够铺满整张画布。通过计算公式(3-22)，我们可建立邻域 $\omega(e^{ref})$ 和 $\omega(e^{tar})$ 中各种子之间的一一对应关系，由此可知，邻域 $\omega(e^{tar})$ 中哪些种子是邻域 $\omega(e^{ref})$ 中不存在的，并将这些种子被作为待添加的候选新种子。如图3-8 (c) 所示，邻域 $\omega(e^{tar})$ 和邻域 $\omega(e^{ref})$ 中形成种子点对的那些种子表示已经存在于画布中，而用'×'标记且对应于邻域 $\omega(e^{ref})$ 中剩余部分的种子即为候选种子。如果这些子没有超出画布的范围，或在添加

了这些种子之后种子总数没有超出用户指定的最大种子总数，那么这些候选种子可作为新种子添加到画布中。根据新种子与中心种子（当前所选目标种子）的相对位置关系添加新种子时，我们需要构建连接新种子与中心种子之间的边，且称为I型边（图3-8（d）），此时，更新的邻域 $\omega'(e^{tar})$ 由已有的种子及添加的新种子组成。然后再构建连接邻域 $\omega'(e^{tar})$ 内所有种子与中心种子之间的边，且称为II型边。

为了避免产生密集或稀疏的种子分布，或者出现边冲突的情况，本章在种子的局部增长过程中引入了冲突检测机制，如图3-9所示。其中图3-9（a）为样本中的部分基元分布，红色的点将作为新种子添加到画布中；图3-9（b-f）为局部种子增长过程中可能出现的五种冲突。此外，图3-9中所示红色和黄色的点表示种子，而椭圆形区域为已添加的基元。当添加某一新种子到画布中时，我们首先检测该种子的邻近区域，如果发现某一种子或基元与当前新添加种子的距离为 l^{short} （定义为参照连通图中最短边的长度的一半），则删除该新种子并与监测的种子和基元构建一条连接边（3-9（b））。如果添加的新种子与已存在种子之间的夹角小于源始角度（3-9（a））的一半时，则需要删除该新种子及构建的边（3-9（c））。接下来，我们需要检测新种子的边的冲出，即检测构建的I型边是否与已合成分布中的边相交。如果被检测到，则除该新种子及构建的边（3-9（d））。当构建连接种子间的II型边时，我们检测每条边的长度。如果某条边长于 l^{long} （定义为源始边长的1.5倍），则在中间位置追加一个额外的种子并构建连接种子的边（3-9（e））。如果检测到存在某种子点在构建的三角形中，则删除之间构建的II型边并构建连接该种子的I、II型边（3-9（f））。

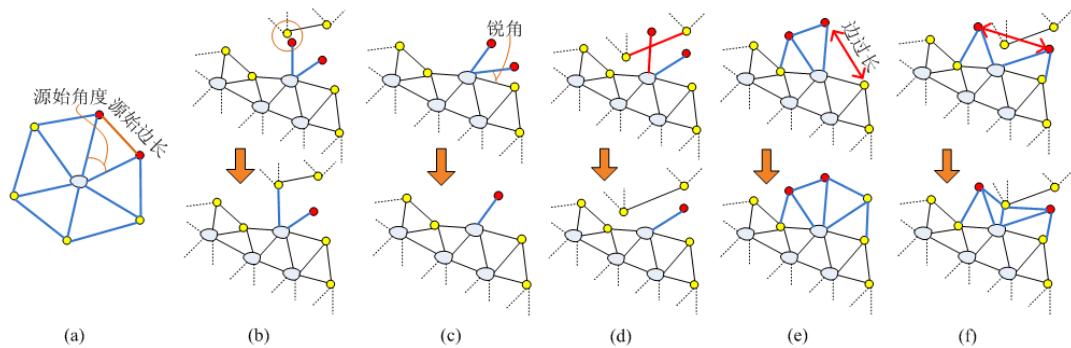


图 3-9 种子间冲突检测
Fig 3-9 Collision detection among seeds

3.4 基于基元的纹理设计

在3.3节中讨论的基于样本基元分布的纹理合成算法，一方面，用户极少提供手工操作干预这个全自动的合成过程；另一方面，由于上述算法是参照基元分布进行纹理合成的，这使得合成的结果通常是单一的。而在本章节中，我们在不参照样本纹理的基元分布的情况下通过控制基元的位置和形状来进行纹理设计，从而能够输出多种多样的纹理图像。

3.4.1 基于简笔画的基元分布生成

在纹理设计过程中，用户首先提供一幅简笔画（a freehand sketch），然后我们的目标是扩张它来生成基元的位置分布，即基元的空间排列（arrangement of texels）。简笔画的本质是初始的基元分布，我们称之为生成模型（generating model）。其中，每个基元用一个点表示，同时用户也可以给每个基元画一个形状轮廓。在此，我们提供两种不同类型的简笔画，包括规则的基元排列和随机的基元排列。

第一类简笔画（图3-10 (a)）可以分解为两个组成部分：左右模式（图3-10 (b)），用于引导水平方向上的基元扩张；上下模式（图3-10 (c)），用于引导垂直方向上的基元扩张。更具体地，我们按照这两种定义的生成模式进行基元分布的扩张，即向初始基元分布中逐一添加新的基元。此时，根据当前待添加基元的大小以及相邻已添加基元的相对位置关系，我们计算该基元的位置。假设简笔画（图3-10 (a)）中的第一个基元的位置为 (Cx, Cy) ，通过计算该基元中心位置到其四周的最短距离定义尺度参数 (yl, yr, xl, xr) ；以及根据基元的位置和尺度参数定义区域参数 (Nly, Nlx, Nry, Nrx) ，从而获得该基元的左上角和右下角的坐标信息。如果根据左右模式添加新的基元，计算该基元的尺度参数 (yl', yr', xl', xr') ，此时，该基元的位置 (Py, Px) 计算如下：

$$\begin{cases} Py = Cy + yr - yr' = Nry - yr' + 1 \\ Px = Cx + xr - 1 + ov + xl' = Nrx + ov + xl' \end{cases} \quad (3-24)$$

其中， ov 为一个像素宽度的空隙。按照同样的计算方式，如果按照上下模式添加新的基元，则该基元的位置计算如下：

$$\begin{cases} Py = Nry + ov + yl' \\ Px = Nlx + xl' + 1 \end{cases} \quad (3-25)$$

在获得新添加基元的位置之后，我们需要计算并记录该基元的区域参数，这将用于后续新添加基元位置的计算。不失一般性，所有新添加基元的位置坐标都可以按照公式（3-24）和（3-25）计算得到。如图3-10（d）所示，我们交替地首先执行左右模式然后上下模式扩张基元分布，其中，我们简单地使用矩形框表示基元的形状。第一类简笔画的多样性在于通过排列基元来改变基元的布局，即通过旋转改变基元的方向、通过缩放改变基元的大小及变换不同类别基元的先后顺序。此外，如果提供的简笔画中有多类基元，则根据其中各基元类别的排列顺序来确定新添加基元的类别。

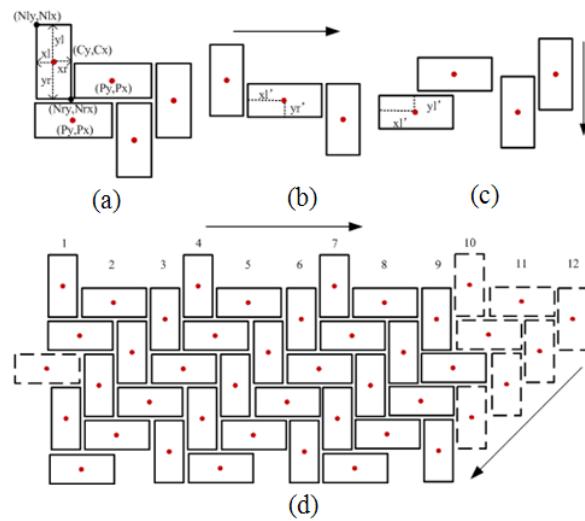


图 3-10 基于简笔画的基元分布生成

Fig 3-10 Creating texels arrangement from the user-provided freehand sketch

第二类简笔画（图3-11）则由一组点集的随机分布组成，且不再需要用户为每个点添加表示基元形状的轮廓，这主要是由于基元间彼此不相邻接的特征。该类简笔画的扩张仅需要参照样本基元分布对初始基元分布进行调整。给定一组离散点集，我们首先采用洛依德方法（Lloyd's method）[57, 58]计算点集的Voronoi图。此时，每个点位于其Voronoi区域的中心位置，由此产生基元的初始分布。由于该初始基元分布趋于规则的点的分布，因此，扩张过程需要进一步调整分布中各基元的位置，使得最终的基元分布具有样本基元分布的随机排列特征。此时，基元位置的调整如下：给定参照样本基元分布 D_{ref} 和初始基元分布 D_{tar} ，对于 D_{tar} 中任一基元 e^{tar} 满足其邻域 $\omega(e^{tar})$ 中所有基元形成一个环形结构：

1. 采用邻域比较方法（公式（3-22）），即逐一计算当前所选目标基元 e^{tar} 的邻域和参照样本基元分布 D_{ref} 中每一基元 e^{ref} 的邻域之间的相似程度，从而选出与 e^{tar} 最匹配的基元 e^{ref} ，此时这两个基元的邻域最相似。
2. 计算相对于基元 e^{ref} 的偏移 S_{ref} ，即基元 e^{ref} 与其邻域 $\omega(e^{ref})$ 质心 e_C 之间的距离。
3. 计算当前所选目标基元 e^{tar} 的偏移值 $S_{tar} = (S_{ref}A_{tar})/(nA_{ref})$ ，并根据该偏移调整基元 e^{tar} 的位置，其中 A_{tar} 和 A_{ref} 分别表示邻域 $\omega(e^{tar})$ 和邻域 $\omega(e^{ref})$ 所形成环形区域的面积， n 是邻域 $\omega(e^{tar})$ 中基元的总数。

重复步骤1到步骤3，直到调整了基元分布 D_{tar} 中所有符合要求的基元的位置（图3-11（b））。此时，基元的扩张过程结束，从而获得最终的基元位置分布（图3-11（c））。可以看出，第二类基元分布的多样性很大程度地依赖于初始离散点集中点的总数，通过增加或者减少点集的数目来调整基元分布的稠密程度。

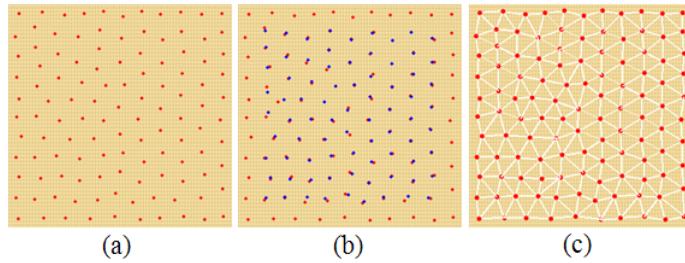


图 3-11 第二类简笔画及其扩张

Fig 3-11 Illustration for the second type of the freehand sketch and its expansion

3.4.2 基于薄板样条的基元变形

在3.3.4节中已经讨论，将基元放置到画布中相应位置上时，基元之间可能产生过分重叠和空洞。这种现象在合成基元间相互邻接的纹理模式时尤其明显。在本章节的纹理设计过程中，简笔画中各基元的形状可由用户自由给定，即基元的形状可能与从样本中抽取的基元的形状不同。由此，如果我们继续采用线性变换操作（如缩放、选择、平移等）处理已抽取基元，那么也只是改变它们的大小，而无法调整基元的形状。

为了解决这个问题，我们采用薄板样条技术（Thin Plate Spline, TPS）[115]变形样本纹理中基元的形状。当使用该技术变形基元的形状时，该方法首

先需要提供参照基元的轮廓点集与待变形基元的轮廓点集之间相应的个轮廓点对。其中，TPS映射技术（TPS mapping）的定义如下：

$$\begin{aligned}x' &= f_x(x, y) = a_1 + a_x x + a_y y + \sum_{i=1}^n \alpha_i \phi(\|(x_i, y_i) - (x, y)\|) \\y' &= f_y(x, y) = b_1 + b_x x + b_y y + \sum_{i=1}^n \beta_i \phi(\|(x_i, y_i) - (x, y)\|)\end{aligned}\quad (3-26)$$

其中， (x, y) 是参照基元中的像素点， (x', y') 是对应待变形基元中的像素点； $\phi(r) = r^2 \log r$ 是径向基函数； $\|\cdot\|$ 是欧式距离算子； $a_1, a_x, a_y, \alpha_i, b_1, b_x, b_y, \beta_i$ 为待确定的参数。为了确定这8个未知参数，该方法还需要提供两个额外的约束：

$$\begin{aligned}\sum \alpha_i &= \sum \alpha_i x_i = \sum \alpha_i y_i = 0 \\ \sum \beta_i &= \sum \beta_i x_i = \sum \beta_i y_i = 0\end{aligned}\quad (3-27)$$

在求解上述两个线性方程组之后，该变形方法即可根据 $f_x(x, y)$ 和 $f_y(x, y)$ 之间的映射关系变形基元的内部区域。假设以简笔画中某一基元的形状为参照轮廓，我们根据公式（3-20）可计算获得样本纹理中每一基元的边界轮廓与已知参照轮廓之间轮廓点对的一一对应映射关系。此时，我们利用TPS映射技术变形已抽取基元到具有参照轮廓的区域。图3-12给出一个样本基元变形的例子，其中图3-12（a）给出参照基元的轮廓点（红色小圈）和待变形基元的轮廓点（黑色十字叉），以及轮廓点对之间的一一对应关系；图3-12（c）为样本基元（图3-12（b））的变形结果。TPS映射技术的使用不仅能够保持基元的形状轮廓，还能在变形的基元内部产生平滑、连续的内容。

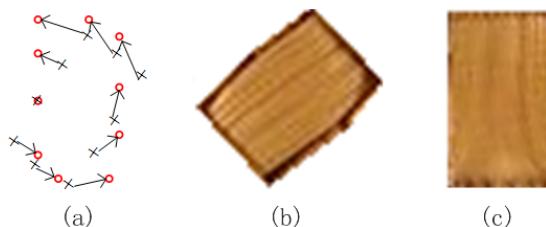


图 3-12 细薄板样条映射

Fig 3-12 Thin plate spline mapping

3.5 实验结果与分析

3.5.1 纹理合成结果与分析

图3-13给出本章方法与近似规则纹理合成方法 [54]、基于图割的 [34]和图像缝合的纹理合成方法 [28]的比较。我们发现使用方法 [34]和 [28]合成的结果不能很好地保持这种规律性特征。当基于邻域的搜索没有找到理想的候选集时，在这些方法中使用的图割技术和动态规划技术将无法产生纹理子块间的平滑传递（图3-13中第二、三列）。方法 [54]通过检测样本纹理中潜在的网格结构，从而能够在合成纹理中重建结构的规律性。然而，当样本纹理被裁剪成一组拼接子块集时，由于拼接块破坏了纹理元素的完整性，因此，纹理特征的扭曲和错位在合成结果中是无法避免的（图3-13中第四列）。而在本章的方法中，我们从样本纹理中抽取出完整的基元，并在合成过程中试图保持基元间的全局拓扑结构。因而，本章合成方法能产生较高质量的纹理并保持结构的规律性。

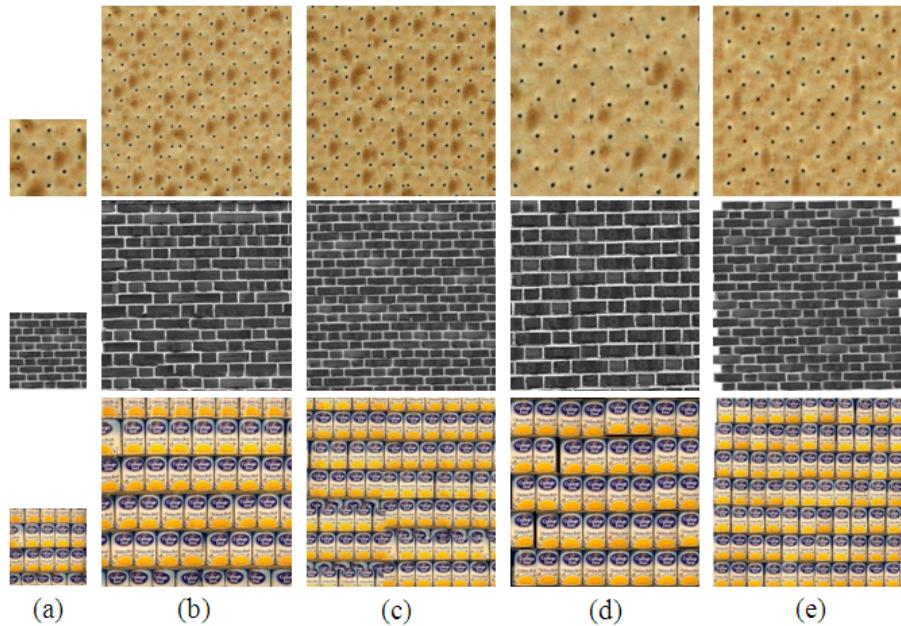


图 3-13 本章方法与基于图割的、图像缝合的和近似规则纹理合成方法的比较

Fig 3-13 Comparison of our method with Graph-cut, image quilting and near-regular texture synthesis

图3-14给出本章提出的纹理合成方法和现有基于特征的纹理合成方法的比较，其中第一行是样本纹理，中间一行是使用现有纹理合成方法获得的结果

(包括基于特征匹配的纹理合成方法 [51]，基于外表空间的纹理合成方法 [52] 和结构化的纹理合成方法 [53])，最后一行是本章方法产生的结果。在本章方法中，我们通过拼贴基元来重建纹理结构。由于在合成过程中仅使用完整的基元，因此输出结果中不会出现破裂的纹理特征。相较于结构化的纹理合成方法，本章方法通过检测和定位样本纹理中的基元来分析基元的空间排列。此外，本章方法合成速度快，例如仅需要3-5秒即可合成基元总数是样本纹理中六倍的新纹理图像。

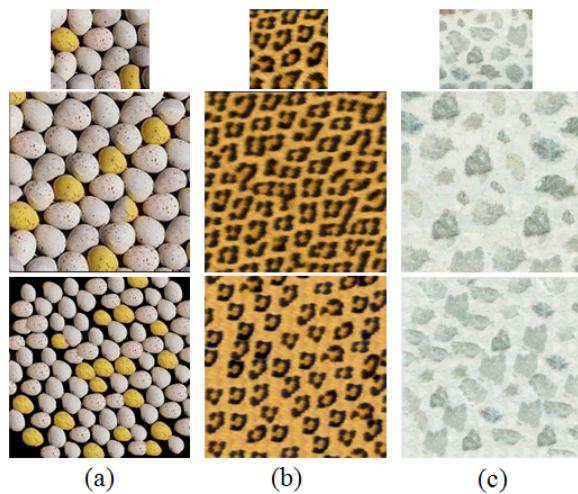


图 3-14 本章方法与基于特征的纹理合成技术的比较

Fig 3-14 Comparison with feature-based texture synthesis techniques

图3-15中讨论样本基元分布对纹理合成的影响。以第一列给出的样本纹理为例，该样本纹理具有三类不同的基元，其中包含两个黄色、三个白色的和七个黑色的基元。当我们确定画布的中心为基元的初始位置时，我们发现合成的分布中并没有完全被种子点覆盖，而且黑色的种子数远远多于其他两类基元的数目，其主要原因是样本纹理中仅有黑色的基元具有足够多的相邻基元。对于这种情况，我们的方法将在样本纹理中添加新的基元，从而产生信息丰富的基元分布。在图3-15的其余三例中，我们通过参照扩张的基元分布，从而获得了满意的结果。

图3-16给出了更多纹理合成结果，其中第一列为样本纹理，第二列为参照基元分布，最后两列分别为最终的合成结果和产生的新的基元分布。给出的这些样本纹理从上到下可以分为：规则纹理、近似规则纹理和不规则纹理。对于

前两类的纹理，本章方法能够合成理想的结果。而对于不规则纹理，由于种子局部增长过程中的误差累积，合成的基元分布往往偏离样本纹理的基元分布。但是，合成的结果仍然能够保持纹理特征的连续性和基元的整体性。

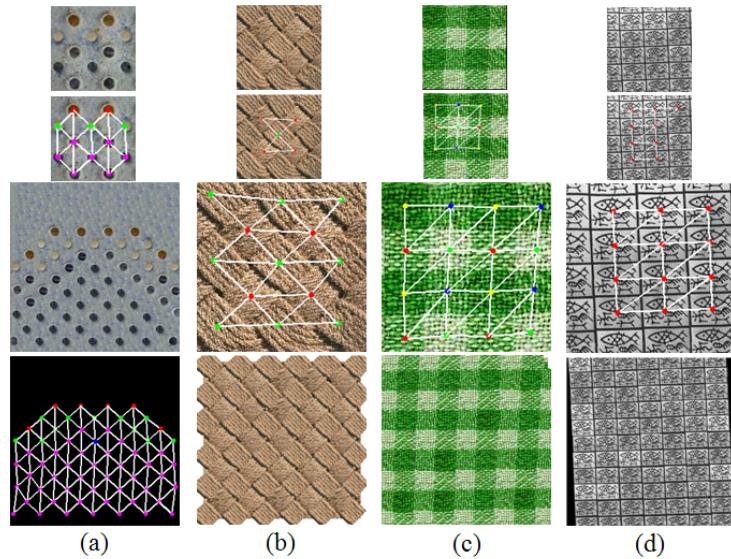


图 3-15 样本基元分布对纹理合成的影响

Fig 3-15 The effect of arrangements of texels on the synthesized textures

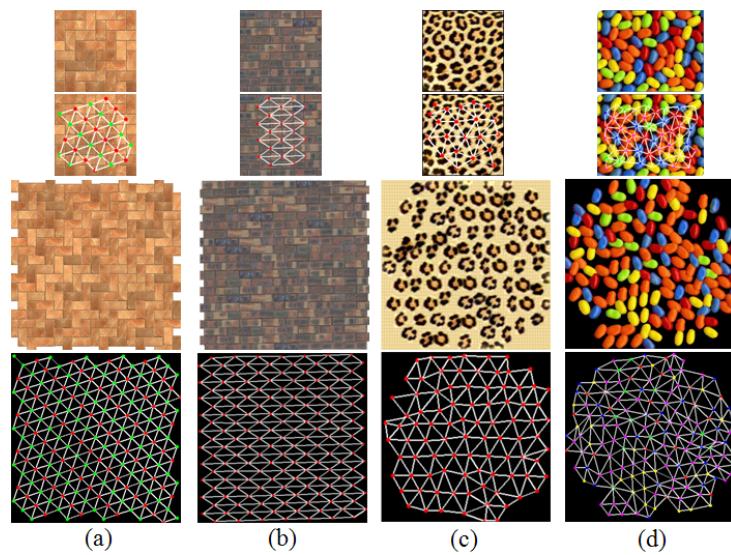


图 3-16 更多纹理合成结果

Fig 3-16 More results for texture synthesis

3.5.2 纹理设计结果与分析

图3-17中给出本章的纹理设计方法能够从给定的样本纹理中产生多种多样的输出结果。在前两个例子中，如果简笔画中基元的初始位置与样本中基元的排列是一致的，那么我们可以获得与样本纹理具有相似外观的结果（图3-17中前两行的第二列）。否则，设计的输出结果将产生不同的外观（图3-17中前两行的第三到第六列）。而在后两个例子中，本章方法根据初始简笔画中给定的基元总数来控制输出结果中基元的稠密程度，此时，图3-17中后两行所示结果的基元数目是不断增加的。此外，本章方法还可以通过替换基元的分布（图3-17的第三行第六列）或操控基元的尺度和方向（图3-17的第四行第五、六列）来产生多样的结果。

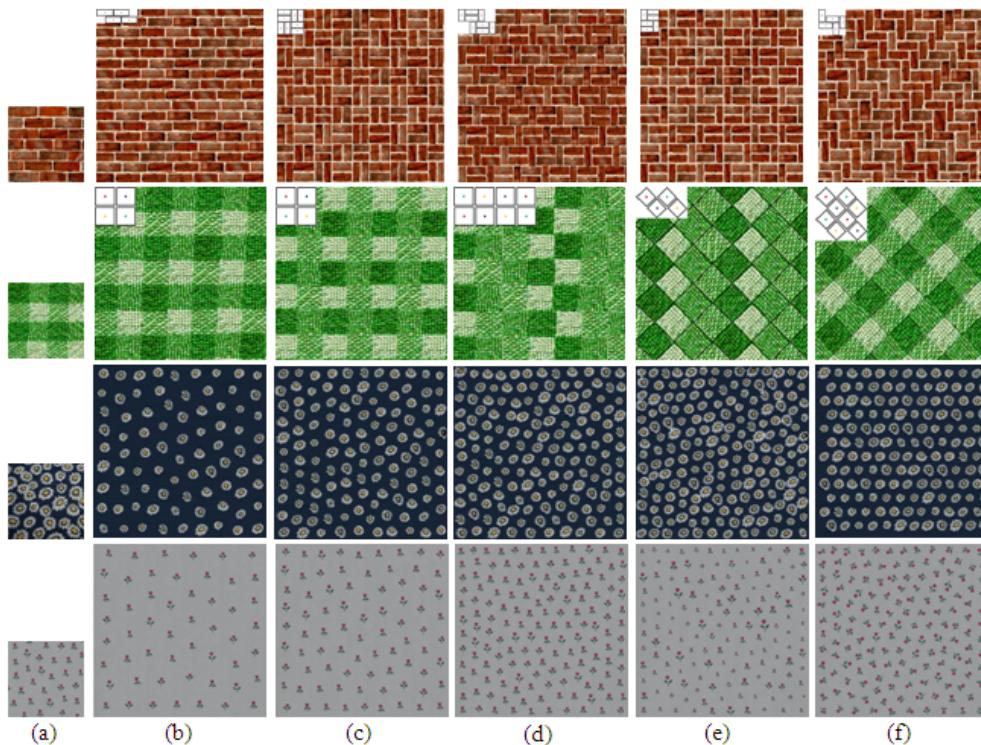


图 3-17 纹理设计结果
Fig 3-17 Results for texture design

在图3-18所示结果中，我们给出本章纹理设计方法和现有能产生较高质量结果的纹理合成方法 [22, 28, 29, 34, 52, 54] 的比较。对于图3-18 (a) 所示由倾斜基元组成的编织纹理，用于比较的这些纹理合成算法都不能产生很好的合成

结果（图3-18（c-i）），合成结果不能够很好地保持结构的规律性和连续性。例如，使用基于块的纹理合成方法 [28, 29, 34, 54]产生的结果（图3-18（e-h））虽然优于基于像素的纹理合成方法的 [22]（图3-18（i）），但是合成的结果中仍然存在断裂的纹理结构。而在使用基于特征的纹理合成方法 [52]产生的结果（图3-18（d））中，虽然能够保持结构的连续性，但是无法保持结构的规律性。而在使用本章提出的纹理合成方法得到的结果中（图3-18（c）），主要存在的问题是基元之间容易产生大的空洞，使得合成结果中出现多余的纹理区域。然而，本章交互可控的纹理设计方法能够根据提供的初始简笔画进行纹理合成，并在基元放置的过程中采用TPS映射技术控制基元的形状（图3-18（b1））。然后，通过对设计的结果进行任意的旋转，我们即可获得最终的纹理合成结果（图3-18（b2））。可以看到，该合成结果能够保持结构规律性和连续性。

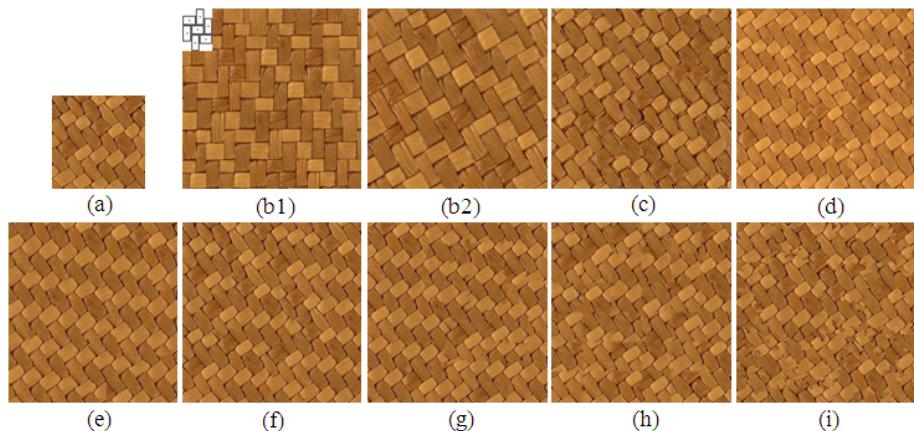


图 3-18 与多种经典纹理合成方法的比较

Fig 3-18 Comparison with some classical existing texture synthesis techniques

最后，我们比较本章中提出的纹理合成方法和纹理设计方法。对于图3-19中给出这两个例子，基于样本基元分布的纹理合成方法使用邻域比较进行基元位置的扩张来产生全局的基元随机分布。由于种子局部增长过程中的误差累加，使得该方法很难无限地产生新的基于的位置，因此，我们无法产生最优的基元随机分布（图3-19（a4）和（b4））。这些新产生的基元的分布很大程度地偏离了样本纹理的基元分布，使得在合成的结果中容易导致基元间过分重叠（图3-19（a3））或者出现不断重复的基元（图3-19（b3））。而基于简笔画的纹理设计方法则采用Lloyd方法进行分布扩张并控制合成分布中基元的总数目，从而较好地产生这种基元的随机分布（图3-19（a6）和（b6））。此外，纹理设

计方法还允许在基元放置过程中随机的选择样本基元，最终合成高质量的新纹理图像（图3-19（a5）和（b5））。

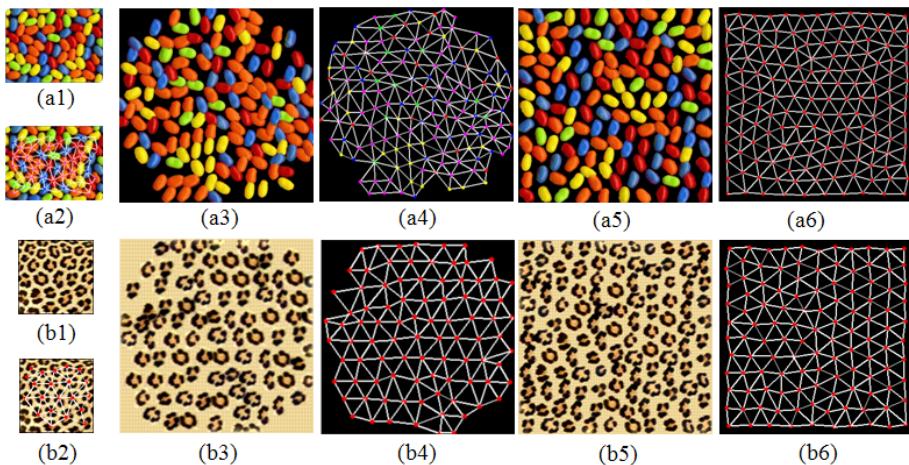


图 3-19 与基于基元分布的纹理合成方法的比较

Fig 3-19 A comparison with the texels distribution based texture synthesis

3.6 本章小结

在本章方法中，我们通过分析样本纹理的基元连通关系来合成新的纹理图像。该方法的关键是参照基元的分布获得基元在合成纹理中的位置。一方面，我们采用交互式图割方法进行完整基元抽取，从而保留重要的基元边界轮廓信息。另一方面，我们使用基于邻域比较方法来产生新的基元分布模式，使得能够很好地保持纹理的结构规律性。然而，该方法的主要局限在于需要分割样本纹理为单个的基元，这需要样本纹理中的基元是清晰可辨别的。而另一个局限是：当放置具有不规则形状的基元时，合成纹理中容易产生基元间过分重叠和大的空洞，即使我们引入了基元缩放机制。

针对这个问题，我们在本章所提出的纹理设计方法中引入了TPS映射技术来变形基元的形状，从而尽可能地保持纹理结构的连续性。此外，该纹理设计方法根据用提供的简笔画来生成基元的分布，而不再是参照样本纹理的基元分布。因此，该纹理设计方法能够合成多种多样的输出结果。

第四章 基于样本视频编辑的单幅图像的流体动画合成

4.1 引言

当观望一幅自然场景图像时，场景中可辨别的元素（又或称为似流体的元素），如水、烟雾、火焰、瀑布等，通常使得我们想象图像中的场景是运动的。在计算机图形学和计算机视觉领域中，研究者们提出了许多方法 [75, 77–80] 用于探讨如何驱动此类静止的图像，从而合成具有动态特性的流体动画。目前，具有似流体区域的图像已被广泛应用于数字图像产业。例如，在卡通动画或动作影片中，这类图像通常用作移动角色的背景。同时，如果这些背景也是动态的，那么这会使整个场景更具戏剧性或更加令人印象深刻。然而，如何从单幅静止的图像中生成连续的流体动画仍然是一个巨大的挑战。

合成流体动画的方法主要分为两大类：一类是基于物理的模拟；另一类是直接对现有视频进行编辑。前者方法 [71–73] 能够模拟多种流体运动，然而，由于这些方法是计算量大的，而且需要针对特定的流体类型设置相应的参数，因此，即使是经验丰富的设计者也必须花费相当长的时间，通过不断试验和纠错来取得令人满意的结果。后者方法则是拷贝或粘贴现有视频帧到静止图像。然而，可获得的视频并不总是与当前所选图像的外观匹配。因此，视频纹理合成方法 [62, 69] 则成为了流体模拟的另一种可能性。但是这些方法并没能成功地从给定的静止图像中合成连续的流体动画，并同时保持整体外观。

在本章方法中，我们设计了一个能够从具有似流体区域的静止图像中合成连续流体动画的系统。给定一幅待驱动的目标图像，用户需要为该系统提供如下输入：1) 选择具有特定流体运动的样本视频；2) 构建包含典型流体运动的视频库；3) 自动估计对应于目标图像的流场信息；4) 传递样本视频的高频流体特征到目标图像。通过使用该系统，我们能够快速地给目标图像中的流体区域添加合理的运动。

我们约束合成的流体动画具有平稳的时序特征，即假设流体动画中某一固定位置点的速率在时间轴上是恒定不变的。基于这个假设，该系统首先将样本视频（本文中所用的样本视频均为视频纹理）分解为三个分量，包括平均图像、平均流场和残差序列。为了抽取高质量的残差序列，我们采用基于大

位移的光流估计算法 [116] 提取样本视频的流体运动，特别是当样本视频具有剧烈变化的动态特征。然后，基于构建的流体视频库，我们使用 SIFT (Scale Invariant Feature Transform) 流算法 [117] 传递与目标图像具有相似场景的视频帧的光流场，使得该系统能够自动地从单幅图像中估计目标图像的流场信息。在获得目标图像的平均图像和等价的残差序列之后，该系统通过计算相应的差异序列来合成流体动画。在这个过程中，根据目标图像中流体区域的大小，我们需要采样基于细缝的视频缩放方法 [118] 对样本视频的大小进行调整，并且需要根据已估计的目标图像的流场信息来变形样本视频的残差序列。最后，该系统采用直方图匹配方法 [119] 保持目标图像的源始外观特征。可以看出，本章提出的流体动画合成系统具有如下三个显著特征：高精度的光流计算、流场的自动估计和时空一致的残差序列。

需要指出的是，本章中所提系统的设计基于如下三个假设：1) 能够容易地想象静止图像中流体区域的大致运动；2) 能够找到与静止图像中流体区域具有相似运动和场景的样本视频；3) 能够根据估计的流场信息驱动静止图像中流体区域从而产生流体动画。

4.2 基元样本视频的流体动画合成系统

4.2.1 现有流体动画合成系统分析

Okabe等人 [79] 提出的流体动画合成系统主要分为两个部分，包括视频分析和视频合成。图4-1给出了该系统的整个流程。视频分析阶段首先分解样本视频（图4-1（a））为所有视频帧的平均图像（图4-1（b））和在原视频帧与平均图像之间的差异图像序列（图4-1（c））。差异图像序列又进一步分解为平均流场（图4-1（d））和残差序列（图4-1（e））。平均流场是光流序列的平均；残差序列仅包含样本视频中非平移的高频信息，这是平均光流场不能捕获的。

在视频合成阶段，根据样本视频分析后获得的分解信息，该系统提供目标图像（图4-1（j））相应的等价信息。该过程的两主要工作是流场的设计（flow field design）和残差的传递（transfer of residuals）。该系统允许用户交互地进行流场的设计，通过使用绘图画刷描绘目标图像中流体的运动，并为画刷线条覆盖的各像素点分配方向向量和速度值（图4-1（f））。此外，该系统自动地传递样本视频的残差序列给目标图像。对于目标图像中的每一子块区域，该系统在样本视频和目标图像的流场之间进行最佳匹配搜索。然后，该系统通过

拷贝及拼接残差序列体块（由所有残差子块组成）获得目标图像的残差序列（图4-1（g））。根据已设计的流场，该系统采用运动模糊技术获得目标图像的平均图像（图4-1（h））。最后，该系统结合已估计的平均图像和合成的差异序列，从而产生具有动态特征的动画序列（图4-1（i））。为了保持目标图像的源始外观特征，该系统对获得的动画序列进行直方图匹配操作获得最终的流体动画（图4-1（k））。详细的实现细节可参见文献 [79]。

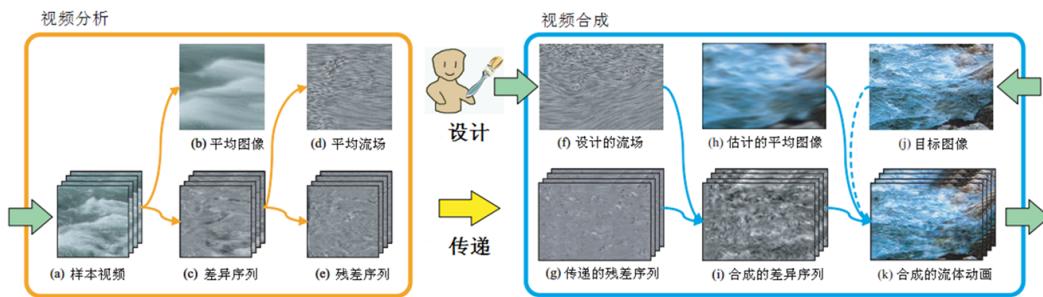


图 4-1 现有流体动画合成系统 [79]
Fig 4-1 Existing fluid animation system [79]

然而，在上述视频分析阶段中，Okabe等人使用Lucas-Kanade方法 [120]计算得到的光流不能产生平滑的速度场，特别是处理具有剧烈变化运动特征的样本视频。由于使用该方法计算的光流是稀疏的，因为仅能获得特征点上的速率。不精确的光流估计可能影响残差序列生成的质量。而在视频合成阶段中，虽然交互的流场设计能够很好地处理多种似流场景的静止图像，但是该操作对处理具有细小的或模糊不清的似流体场景时是失效的，这需要提供繁琐的用户交互。此外，该系统基于块的采样策略自动传递样本视频的残差序列，从而获得目标图像等价的残差序列。然而，由于采样的各子块具有独立的流体特征和不一致的颜色和光照，因此，在拼接这些子块时容易产生不连续的视觉特征。也就是说，尽管传递的残差序列能够保持时间的连续性，但在外观空间上是不平滑的。另一方面，由于穷尽的块搜索方式，该视频合成过程是耗时的。

4.2.2 流体动画合成系统概述

为了克服上述存在的问题，我们对文献 [79]中提出的流体动画合成系统进行了改进。此时，本章中提出的系统由四部分组成，包括视频调整、视频分析、运动估计和视频合成。给定目标图像（图4-2（a）），我们首先检测其中

感兴趣的流体区域，这需要提供相对于这些区域的alpha遮片（图4-2（b））。然后，根据已检测流体区域的大小，我们对所选样本视频（图4-2（c））的尺寸进行缩放调整，使得调整后的样本视频（图4-2（d））和目标图像中的流体区域具有相同的大小。执行这种有效的样本视频缩放操作是为了完整地保持其主要的流体特征。

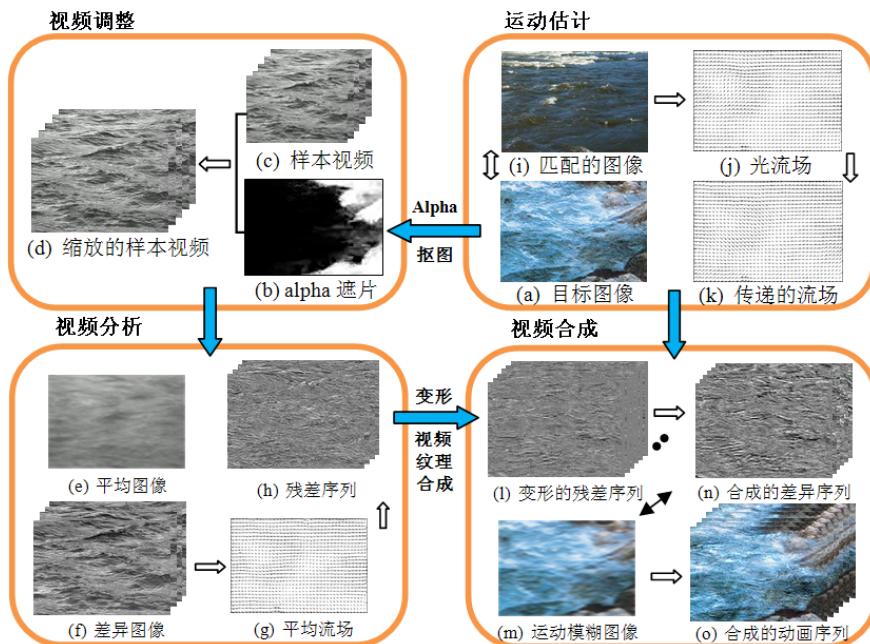


图 4-2 全局特征保持的流体动画合成系统

Fig 4-2 Preserving global features of fluid animation system

已知缩放的样本视频，该视频分析阶段与系统 [79] 中的具有相同的功能，即获得平均图像（图4-2（e））和差异图像序列（图4-2（f））。为了抽取平均流场（图4-2（e）），其中光流的计算是采用基于大位移的运动估计方法。残差序列（图4-2（h））是在差异图像和根据平均流场变形的差异图像之间计算得到的差图像。在这个阶段，平均图像的抽取使得能够很容易地替换成流体动画的整体外观；平均流场是流体动画的一个重要特征，它的抽取是产生流体动画的辅助因子；而残差序列的抽取则是用于传递样本视频中的流体特征。

为了获得与目标图像相对应的平均流场，本章中的系统利用构建的流体视频库能够从单幅图像中自动地进行运动场估计。根据SIFT流算法建立的场景对应关系，我们能够从视频库中找到与目标图像具有最相似场景的匹配图像（即

视频帧) (图4-2 (i))。此外, 根据已计算的SIFT流, 我们变形所选匹配图像的光流场 (图4-2 (j)) 来获得目标图像的流场信息 (图4-2 (k))。使用这种根据运动先验信息进行运动传递的优势在于能够避免使用手工描绘目标图像中模糊的或低分辨率的流体特征, 极大地减少了用户的输入。

已知样本视频的残差序列和目标图像的平均流场, 该系统首先采用视频纹理技术 [69]以部分残差序列为样本合成任意长度的新的残差序列, 并根据传递的流场进行变形, 从而产生目标图像对应的残差序列 (图4-2 (l))。这个过程是为了能够合成无限长度的流体动画序列, 以及能够在合成功画序列中保持时空的一致性。最后, 本章的系统也是结合已估计的目标图像的平均图像 (图4-2 (m)) 和合成的差异图像序列 (图4-2 (n)), 从而生成连续的流体动画序列。我们在合成的流体动画序列和目标图像之间通过执行直方图匹配操作, 从而获得最终流体动画的外观 (图4-2 (o))。

本章中提出的流体动画合成系统和文献 [79]中所给系统的主要差异体现在: 1) 我们的系统则使用构建的流体视频库自动地获得目标图像对应的流场信息; 2) 我们的系统则根据目标图像中流体区域的大小对所选样本视频进行缩放调整, 并采样全局地方式传递残差序列到目标图像。这使得在合成的流体动画中能够保留大尺度的流体特征。上述合成连续流体动画的实现细节将在后续章节中进行讨论。

4.3 样本视频的调整

4.3.1 流体区域的抽取

目标图像中的场景不仅包括待抽取的似流体区域, 而且包括静止的物体 (如岩石、树、建筑物等)。我们的目标是驱动这些似流体的区域产生动画序列, 但同时需要保持静止的物体在合成的结果中是始终如一的。基于有限的用户输入, 我们采用交互的图像抠图方法 [121]抽取目标图像中的似流体区域 (即前景 F)、静止物体 (即背景 B) 以及前景的透明度 (alpha matte)。此时, 目标图像 I 中的每一像素点的颜色值可以看成是前景和背景颜色的线性组合:

$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i \quad (4-1)$$

其中, α_i 是第*i*个像素点的前景透明度。基于围绕每一像素点的小窗口内前景和背景的颜色是局部平滑的假设, 公式 (4-1) 可以改写成 α 为目标图像 I 的线性

函数：

$$\alpha_i \approx \sum_c a^c I_i^c + b, \quad \forall i \in w \quad (4-2)$$

其中， $a = \frac{1}{F-B}$ 和 $a = -\frac{B}{F-B}$ ； w 为局部窗口，通常设置为 3×3 的大小；为三颜色通道的累加。通过使用这个颜色线性模型，为计算目标图像 alpha 遮片的代价函数定义为：

$$J(\alpha, a, b) = \sum_{j \in I} \left(\sum_{i \in w_j} (\alpha_i - \sum_c \alpha_j^c I_i^c - b)^2 + \epsilon \sum_c \alpha_j^{c^2} \right) \quad (4-3)$$

其中， ϵ 为正则化项，通常设置 $\epsilon = [0.1^5, 0.1^7]$ 用于保持数值求解过程的稳定性，从而避免产生过分平滑的 alpha 遮片。由此， α 即为这个代价函数的全局最优解，可以通过求解一个稀疏线性方程组获得：

$$J(\alpha) = \alpha^T L \alpha \quad (4-4)$$

$$L_{ij} = \sum_{k|(i,j) \in w_k} \left(\delta_{ij} - \frac{1}{|w_{ij}|} \left(1 + (I_i - \mu_k)(\Sigma_k + \frac{\epsilon}{|w_k|} I_3)^{-1} (I_j - \mu_k) \right) \right) \quad (4-5)$$

其中， L 是一个 $N \times N$ 数组，它的第 (i, j) 个元素可以公式 (4-5) 计算得到； Σ_k 是一个 3×3 的协方差矩阵； μ_k 是在局部窗口上的一个 3×1 的均值向量； I_3 是一个 3×3 的单位矩阵。

图4-3给出了使用上述抠图方法获得的结果。我们只要需要在目标图像中使用少数颜色线条标记绝对的前景和背景（图4-3 (b)），即可抽取出alpha遮片（图4-3 (c)），静止的区域（图4-3 (d)）和似流体的区域（图4-3 (e)）。其中，alpha遮片即为目标图像中感兴趣的流体区域，可用于测量似流体区域的大小。

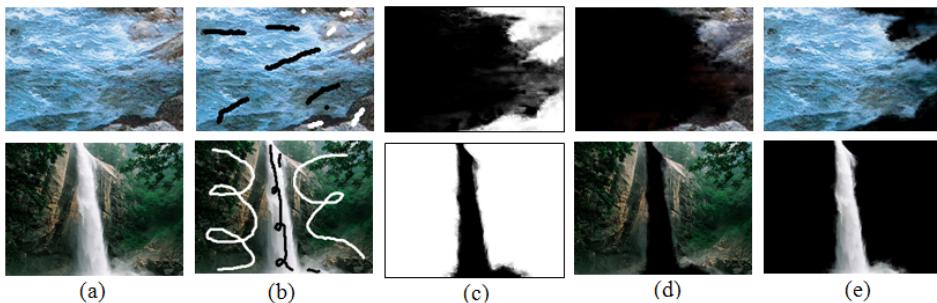


图 4-3 目标图像的抠图结果

Fig 4-3 Preserving global features of fluid animation system

4.3.2 基于细缝的视频缩放

明显地，用户所选样本视频和给定目标图像的大小通常不可能完全相同。（我们约定目标图像中流体区域的大小为已抽取alpha遮片中包围黑色区域的最小包围盒的大小。）这也是为什么现有流体动画合成系统 [79] 需要进行残差的自动传递，使得传递的残差序列相对于目标图像是满足要求的。然而，这种残差传递方式可能导致合成的动画序列中容易出现断裂的流体特征和颜色的不一致。为了在传递的残差序列中保持时空的一致性，根据目标图像中的流体区域的大小，我们采用基于图割的细缝缩放方法 [118] 调整所选样本视频的大小。样本视频缩放过程概括如下：

1. 纵横比的改变。已知样本视频的大小是 $n \times m$ 和目标图像的大小是 $n' \times m'$ ，如果相对应长和宽的差 $h = n - n'$ 和 $w = m - m'$ 的取值之一不为零，则对样本视频进行缩放。否则，跳出该缩放操作。
2. 构建图。样本视频可以看成一个3D时空体，我们将其中所有平面构建一个3D网格立体图。此时，每个像素点代表图中的节点；图的边为每个点的与其4-邻域点的连接。接着，我们连接源端节点到所有视频帧的最左一列节点和末端节点到视频帧的最右一列节点。需要指出的是，当前构建的图用于产生垂直的细缝；相同的构建过程并进行适当的旋转可来产生水平的细缝。
3. 边权的计算。为了避免在移除细缝过程中引入锯齿状的瑕疵，方法 [118] 约束上述图中建立的边是有方向的，且添加两对角节点之间的边（图4-4 (a)）。其中，向左的边的权值为 $+LR = |I(i, j + 1) - I(i, j - 1)|$ ，向右的边的权值为无穷大；向上的边的权值为 $+LU = |I(i - 1, j) - I(i, j - 1)|$ ，向下的边的权值为 $-LU = |I(i + 1, j) - I(i, j - 1)|$ ；而两对角边的权值也为无穷大。其中， $I(i, j)$ 表示像素点 (i, j) 的颜色值。
4. 计算最优缝。在计算3D图中所有边的权值之后，我们采用基于图割的方法获得3D图的一个划分，从而生成三维域上的2D流形曲面。此时，流形定义为各样本视频帧中的一条细缝（图4-4 (b)）。

通过移除所有已计算的垂直方向和水平方向的细缝，我们可以获得缩放后的样本视频。如图4-5所示，第一行给出从样本视频中随机选取的视频帧（图4-5 (a1-a5)）；第二行为使用方法 [118] 进行缩放之后得到的结果（图4-5 (b1-b5)），这些结果能够很好地保留样本视频中显著的流体特征。然而，如最后一行结果所示，当采用上述方法放大样本视频时，由于需要重复地进行像素

点的复制，这使得放大的结果可能产生模糊的流体特征（图4-5（c3-c4）），即使垂直方向的放大能够产生较好的缩放效果。因此，当处理大面积纹理区域时，传统的线性插值缩放方法是有用的。此外，当目标图像中的流体区域由多个不连通的区域组成时，我们可以采用图像剪裁方法进行处理。

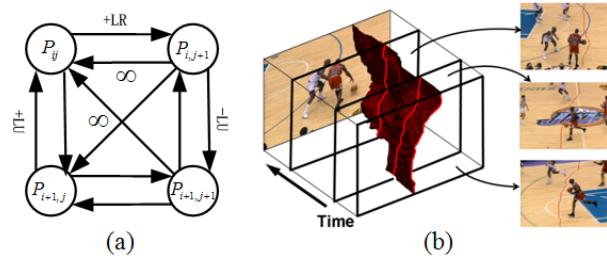


图 4-4 基于图割的细缝缩放 [118]

Fig 4-4 Graph-cut based seam carving resizing [118]

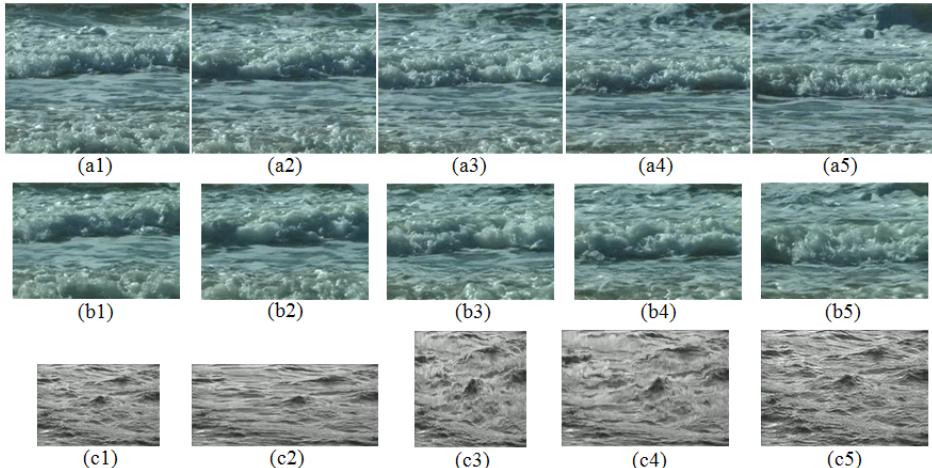


图 4-5 样本视频的调整

Fig 4-5 Adjustment of video examples

4.4 样本视频的分析

视频分析阶段主要是从给定的样本视频中抽取平均流场和残差序列，这些都依赖于光流的计算。我们利用Brox等人提出的大偏移光流算法 [116]对样本视频中的运动信息进行估计。该方法基于从粗到精的变形框架构建了

一个可变的模型，从而能够用于处理具有剧烈变化动态特征的样本视频。已知 $I_1, I_2 : (\Omega \subset R^2)$ 是相邻的两视频帧，我们通过最小化如下能量公式可计算得到这两帧之间的光流场 $w := (u, v)^T$ ：

$$\begin{aligned} E(w) = & E_{color}(w) + \gamma E_{grad}(w) + \alpha E_{smooth}(w) \\ & + \beta E_{match}(w, w_1) + E_{desc}(w_1) \end{aligned} \quad (4-6)$$

其中， $E_{color}(w)$ 的定义是基于两视频帧中相应像素点的颜色值是连续的假设：

$$E_{color}(w) = \int_{\Omega} \Psi(|I_2(\mathbf{x} + w(\mathbf{x})) - I_1(\mathbf{x})|^2) d\mathbf{x} \quad (4-7)$$

函数 $\Psi(s^2) = \sqrt{s^2 + \epsilon^2}$ 是一个拉普拉斯函数， ϵ 设置值为 0.01； \mathbf{x} 表示图像空间中像素点的位置向量。由于光照的影响，仅使用颜色的匹配是不稳定的。因此，这需要在公式 (4-7) 的基础上添加梯度约束，即保持光照的不变性：

$$E_{grad}(w) = \int_{\Omega} \Psi(|\nabla I_2(\mathbf{x} + w(\mathbf{x})) - \nabla I_1(\mathbf{x})|^2) d\mathbf{x} \quad (4-8)$$

$E_{smooth}(w)$ 的定义是基于平滑约束的假设，用于补偿光流的整体变化：

$$E_{smooth}(w) = \int_{\Omega} \Psi(|\nabla u(\mathbf{x}) - \nabla v(\mathbf{x})|^2) d\mathbf{x} \quad (4-9)$$

这三个能量项组成了光流估计的一般形式，能够用于处理各种变形、不连续的运动、遮挡和大的偏移。然而，由于仅优化前三项组成的能量公式可能导致二义的解，因此在公式 (4-6) 中增加了描述性特征匹配的能量项：

$$E_{match}(w) = \int \delta(\mathbf{x}) \rho(\mathbf{x}) (w_1(\mathbf{x}) - w(\mathbf{x})) d\mathbf{x} \quad (4-10)$$

其中， $w_1(\mathbf{x})$ 表示在位置处进行描述性特征匹配的相应向量；当这个向量可获得时， $\delta_i(\mathbf{x})$ 的值为 1；否则，值为 0。 $\rho_i(\mathbf{x})$ 是匹配权重系数。 $E_{desc}(w_1)$ 是定义在两匹配帧中稀疏特征向量场 $f_1(\mathbf{x})$ 和 $f_2(\mathbf{x})$ 之间的又一能量项：

$$E_{desc}(w_1) = \int \delta(\mathbf{x}) |f_2(\mathbf{x} + w_1(\mathbf{x})) - f_1(\mathbf{x})|^2 d\mathbf{x} \quad (4-11)$$

在公式 (4-10) 和公式 (4-11) 中添加辅助变量的目的是在不连续的运动处仍然能够获得精确的光流估计。图 4-6 的第三行给出了光流计算的结果，上述方

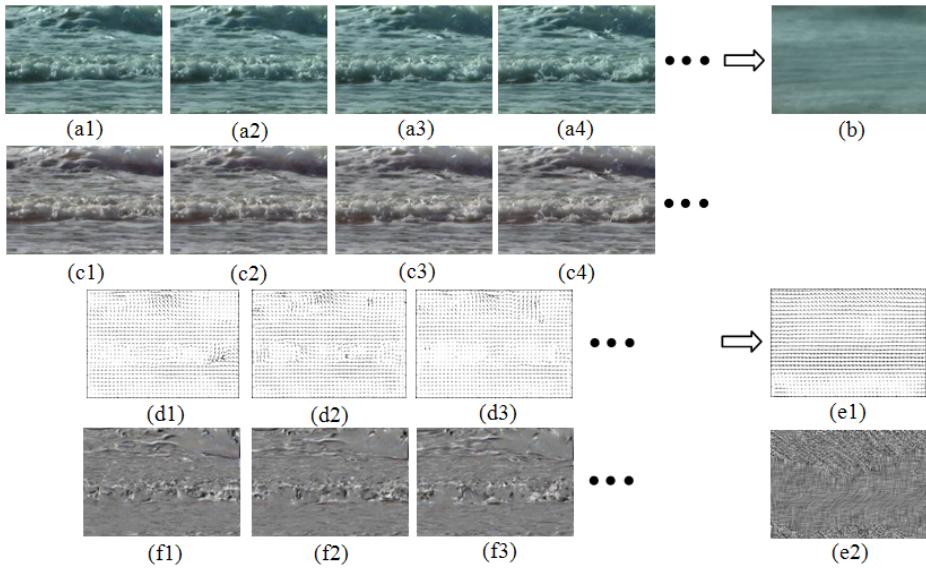


图 4-6 平均光流场和残差序列的抽取

Fig 4-6 Extraction for average flow field and residuals

法能够从给定的样本视频中准确地进行稠密的光流场的估计。此时，所选样本视频（图4-6（a1-a4））显示的是剧烈的海浪场景。

在确定了光流计算的方法之后，从已缩放的样本视频中抽取平均流场和残差序列的过程描述如下：

1. 计算平均图像和差异序列。对样本视频中所有帧的平均定义为 $A = \frac{1}{N} \sum_{i=1}^N F_i$ ，其中 N 为视频帧的总数目； F_i 表示样本视频中的一帧（图4-6（b））。然后，定义差异图像为 $D_i = F_i - A$ （图4-6（c1-c4））。
2. 构建平均光流场。以相邻两差异图像为输入，并采用方法 [116] 计算它们的光流场（图4-6（d1-d3））。此时，平均流场 $\vec{\nu}(\vec{x})$ 定义为所有光流场的平均（图4-6（e1）），其中 $\vec{x} = (x, y)$ 为像素点的位置。
3. 计算残差序列。残差序列定义为在差异序列的第 $i+1$ 帧和差异序列的第 i 帧之间的差图像 R_i （图4-6（f1-f3）），可由如下公式计算得到：

$$R_i = D_{i+1} - W(D_i) \quad (4-12)$$

$$W(D_i) = D_{i-1}(\vec{x} - \vec{\nu}(\vec{x})) \quad (4-13)$$

其中， $W(\cdot)$ 是基于后向映射 $p(\vec{x}, i) = p(\vec{x} - \vec{\nu}(\vec{x}), i-1)$ 的变形函数。此时，可根据已构建的平均流场 $\vec{\nu}(\vec{x})$ 变形差异图像 D_i 。

4.5 流场的估计

4.5.1 流体视频库的构建

为了执行有效的最佳匹配搜索，我们需要准备尽可能多的流体视频，这是因为资源丰富的流体视频库有利于获得高准确率的数据驱动的场景匹配。尽管大量的视频数据能从网络上获得，但是我们发现从中很难找到成百上千的且符合我们要求的流体视频数据。因此，我们选择流体视频的标准是：1) 流体区域是视频的主要特征；2) 流体视频具有足够清晰的画质；3) 除流体区域外，视频中的其他物体无明显的动态特征。我们通过从商业数字媒体素材库购买、网络下载或自行拍摄等多种方式收集了上百个流体视频，其中包括瀑布、河流、火焰和烟雾等多种类型的流体场景。流体视频库中所有视频的分辨率在 320×240 和 640×480 之间，且具有 25fps 的帧率和10-30秒的播放长度。在大多数情况下，我们发现收集的流体视频通常与所选目标图像具有不同的大小。为了使场景匹配是可行的，我们根据所选目标图像的尺寸对每一流体视频进行缩放和剪裁。由此，这些操作也增量了流体视频的数量。图4-7给出视频库中部分流体视频的场景截图。



图 4-7 部分流体视频

Fig 4-7 Fluid videos in the database

4.5.2 基于SIFT流的运动传递

在目标图像和最佳匹配的图像之间，我们需要找到一个密集的场景对应。观念上，如果流体视频库是足够大的（几乎包含了各种可能的场景），那么当前所选最佳匹配图像可能在视觉上极相似于目标图像。区别于光流是建立相邻

视频帧之间时域上的位置对应，由于SIFT流是建立图像之间空间上的场景对齐 (scene alignment)，因此本章方法可根据使用SIFT流构建的场景对应来传递不同场景图像的运动信息。此时，我们将流体视频库中的所有视频帧作为选取最佳匹配图像的候选集。

为了执行一个快速的图像搜索，我们首先采用SIFT关键点检测方法 [122] 提取每一视频帧的SIFT特征点，并将这些特征点表示为128维的特征向量（即SIFT描述子）。此时，我们通过计算各SIFT描述子之间的差异来测量图像的相似性。对于给定的目标图像，我们首先从流体视频库中选出10幅具有较相似场景的视频帧；然后采用SIFT流算法计算目标图像与所选每一邻近视频帧之间的场景对应并根据相应的对齐分数重新排序这10幅图像；最后选择具有最大对齐分数的视频帧即为当前所选最佳匹配图像。根据该视频帧所在视频中前后相邻的视频帧，我们计算该视频帧的光流场。值得一提的是，这些具有相似场景的视频帧可能从多个视频中获得，又或可能来自于同一个视频。

尽管最佳匹配图像与目标图像具有非常相似的场景，但是这些场景在颜色值和梯度上仍具有明显的差异。由于SIFT描述子具有视觉不变性和亮度独立性的特点，因此，匹配SIFT描述子能够在场景中不同物体之间构建稠密的对应。此外，SIFT流算法允许一幅图像中的像素点匹配另一幅图像中的其他像素点，也就是说，像素点的偏移可能比图像本身还要大。已知目标图像 I_1 和最佳匹配图像 I_2 ，场景间的对应 (SIFT流， SIFT flow) 可以通过求解如下代价函数获得：

$$E(w') = \sum_p \|s_1(p) - s_2(p + w')\|_1 + \frac{1}{\sigma^2} \sum_p (u^2(p) + v^2(p)) \\ \sum_{(p,q) \in \epsilon} \min(\alpha |u(p) - u(q)|, d) + \min(\alpha |v(p) - v(q)|, d) \quad (4-14)$$

其中， $w'(p) = (u(p), v(p))$ 是在像素点位置 $p = (x, y)$ 处的偏移向量； $s_i(p)$ 是图像 I_i 中在位置 P 处抽取得到的SIFT描述子； ϵ 是像素点的空间邻域，通常为4-邻域结构。该代价函数的前两项类似于光流计算中的基于颜色连续及平滑约束的能量项；而第三项是规则化项，用于建模像素点间不连续的偏移场。

在获得目标图像和最佳匹配图像的场景对应之后，我们根据这个已构建的SIFT流变形最佳匹配图像的光流场，即可实现从单幅图像中进行运动估计。然而，这个运动传递过程可能使得目标图像中静止的区域具有非零值的流

向量。因此，在流体动画合成过程中，我们将使用已抽取的静止物体替换由这些非零值流向量产生的动画区域，从而保持场景中静止的物体始终不变。如图4-8所示，本章系统根据获得的目标图像（图4-8（a1））和最佳匹配图像（图4-8（b1））各自的SIFT特征点集（图4-8（a2）和图4-8（b2））及其SIFT描述子（图4-8（a3）和图4-8（b3）），通过使用SIFT流算法建立这两个图像的场景对应关系（图4-8（c2））。由此，该系统能够自动地传递最佳匹配图的运动信息（图4-8（c1）），从而获得了相对于目标图像的流场信息（图4-8（c2）），记为 $\vec{v}'(\vec{x})$ 。

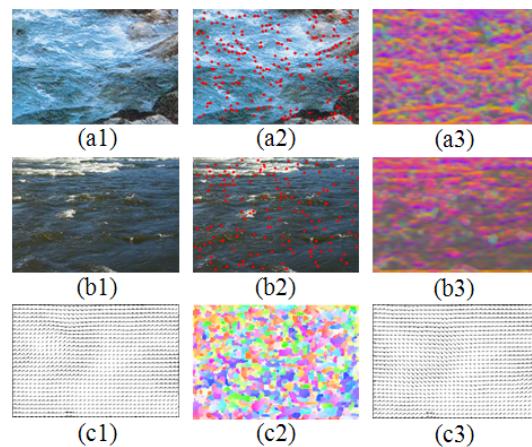


图 4-8 流体运动的自动传递

Fig 4-8 Automatic transfer of fluid motion

4.6 流体动画合成

作为产生连续流体动画的最后一个阶段，其主要任务是获得对应于目标图像的等价信息，包括平均图像、残差序列和差异图像。

4.6.1 平均图像的估计

单幅目标图像的平均图像可以看成是一幅具有运动模糊的图像。本章系统采用Brostow等人 [123]提出的方法自动地进行运动模糊的模拟。根据4.5节中已估计的目标图像的流场信息 $\vec{v}'(\vec{x})$ ，这些运动向量能够说明目标图像中各像素点将如何移动到新的位置。建模像素点随时间偏移的路径为函数 $L_i(x, y, t)$ ，它

是以像素为单位的时间参数函数，并在时间周期 $t = t_i$ 到 $t = t_{i+1}$ 内是有效的。也就是说，目标图像的模糊图像可通过插值图像中待模糊的各像素点的路径计算得到。在这个过程中，用于模拟快门速度（shutter speed）的参数 s 能够控制运动模糊的效果。

假定运动模糊的绘制等同于路径 $L_i(x, y, t)$ 在时间区间（以速度按下快门的起始时间）上的积分，分为前半段和后半段时间。首先分析发生在后半段时间的运动模糊，此时，产生的模糊图像记为 $I_{Aft}(t_i)$ 。已知相对的时间区间是从 t_i 到 $t_i + s/2$ 和像素点的颜色值 c ，当目标图像中的某一位置 (x, y) 上的像素点有可能落入给定目标像素点的边界 $[a, b]$ 且时间周期为 $a \leq t \leq b$ 时，其偏移路径的长度 $w_{(x,y)}$ 可计算如下：

$$w_{(x,y)} = \frac{\int_a^b (\sqrt{(\frac{dx}{dt})^2 + (\frac{dy}{dt})^2}) dt}{s/2} \quad (4-15)$$

其中，位置坐标 x 和 y 可以看成是时间 t 的参数函数，即该像素点的运动向量。此时，位于 (x, y) 处的目标像素点的颜色值定义为 $c w_{(x,y)}$ 。通过这种计算方式，每一目标像素点累加经过它的所有移动像素点的更新的颜色值的值，从而获得该时间间隔上的模糊图像。与计算模糊图像 $I_{Aft}(t_i)$ 相同，我们可获得目标图像从时间区间 t_i 到 $t_i - s/2$ 的运动模糊图像 $I_{Bef}(t_i)$ 。需要指出的是，该计算过程需要用到从目标图像的反向流场信息中获得路径函数 $L_{i-1}(t)$ 。

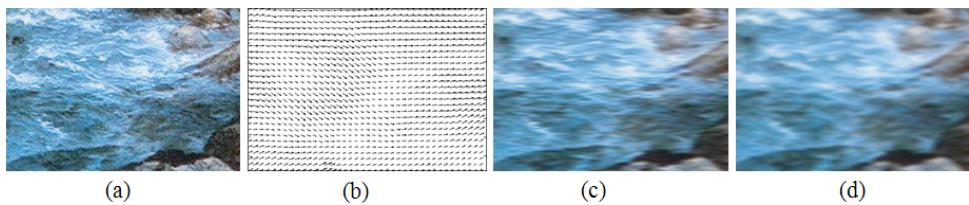


图 4-9 平均图像的估计

Fig 4-9 Approximation of the average image

此时模糊图像 $I_{Aft}(t_i)$ 和 $I_{Bef}(t_i)$ 可以看成是目标图像的两种曝光形式，通过平均对这两个模糊图像即可获得目标图像的运动模糊图像。图4-9给出根据目标图像的流场信息（图4-9（b））进行的运动模糊结果（图4-9（c））。然而，由于沿流场信息产出的运动模糊容易产生锋利的边等高频信息，这在合成的流体动画中容易出现印象视觉的瑕疵，因此，我们将采用高斯滤波操作来获得更平滑的结果（图4-9（d）），这个结果即被用作目标图像对应的平均图像 A' 。

4.6.2 残差序列的合成

给定已缩放样本视频的残差序列 R_i 和目标图像的流场信息 $\vec{v}(\vec{x})$ ，我们的目标是获得目标图像的残差序列。由于所选样本视频通常是有限持续时间的一视频片段，因此，在进行视频分析后我们仅能获得有限长度残差序列。合成流体动画的持续时间很大程度上取决于样本视频的持续时间。为此，我们以已抽取的残差序列为样本，通过采用视频纹理合成方法 [69] 获得任意长度的且具有相似外观特征的残差序列。本章改进的系统在这个阶段使用该合成技术可以避免通过残差序列重构差异图像时不断积累的高频特征出现在合成的动画序列中。

本章系统首先建立连接相似残差图像的图，其中每一残差图像是一个节点；连接两阶段的边的权值是相应残差图像的相似性度量。然后，在图中寻找以每个节点为起始点的极值路径。这些路径是非常重要的，是因为每条路径的两个端点对应残差序列中两个具有截然不同外观的高频信息，并且中间节点对应的这两个极端外观的过渡序列。此时，新的残差图像序列的选取过程可以看出是图的二阶马尔科夫链采样问题。给定前一帧残差 R_{n-1} 和当前帧残差 R_n ，下一帧残差可由如下计算得到：

$$w_{s,k} = \text{Dist}(I(s, k-1), R_{n-1}) + \text{Dist}(I(s, k), R_n) \quad (4-16)$$

其中， σ 是第 s 条路径的第 k 个节点（残差图像）； $\text{Dist}(\cdot)$ 的值是两残差图像的距离； $w_{s,k}$ 的分布决定了选择 $I(s, k+1)$ 作为下一帧残差的概率：

$$P(I(s, k+1)|R_{n-1}, R_n) \sim \exp(-w_{s,k}/(\sigma \sum_{s,k} w_{s,k})) \quad (4-17)$$

其中， σ 是用户可控的参数。上述公式 (4-16) 和公式 (4-17) 可以说明如果第 s 条路径的第 $k-1$ 和 k 个残差图像近似于给定的残差帧 R_{n-1} 和 R_n ，那么极有可能选择第 s 条路径的第 $k+1$ 个残差图像作为下一帧残差 R_{n+1} 。在合成的新残差序列中，如果任意两相邻残差图像之间具有明显的闪动，我们则采用基于光流的双向插值方法生成这两残差图像之间的过渡残差序列，使得产生平滑连续的残差序列。

由于合成的残差序列中流体特征的运动需要与目标图像中潜在的流体运动模式一致，因此，我们进一步根据目标图像的流场变形合成的残差序列：

$$R'_i = W'(R_i) = R_i(\vec{x} - \vec{v}'(\vec{x})) \quad (4-18)$$

其中，是基于后向映射的变形函数。此时，可根据目标图像的平均流场变形合成的残差序列图像。这个变形结果即为对应于目标图像的残差序列。

4.6.3 视频重构

根据已估计流场的变形函数 $W'(\cdot)$ 和变形的残差序列 R'_i ，通过使用 $W'(\cdot)$ 的复合函数 W^n 和生命时间参数 τ ，我们重构目标图像的差异序列 D'_i ：

$$W^n = W' \circ W'^{n-1}(R'_k), n = i - k - 1 \quad (4-19)$$

$$D'_i = \sum_{k=i-\tau}^{i-1} W'^{i-k-1}(R'_k) \quad (4-20)$$

在获得重构的差异图像之后，我们需要进一步调整 D'_i 的对比度和亮度：

$$D''_i = (D'_i - \langle D' \rangle) \cdot \alpha + \beta \quad (4-21)$$

其中， $\langle D' \rangle$ 是当前重构差异序列 D' 的所有像素点的均值； α 和 β 是对比度和亮度参数。定义 C 为目标图像和估计的平均图像之间的差图像，由于差异序列 D''_i 应该与差图像 C 具有相同的对比度和亮度，因此 β 亮度参数 C 即为差图像的所有像素点的平均值；对比度参数为 C 和 D' 的标准差的比值，即定义为 $\alpha = \sigma_C / \sigma_{D'}$ 。然后，通过结合估计的平均图像 A' 和合成的差异序列 D''_i ，我们可以获得流体动画序列 $F'_i = A' + D''_i$ 。当样本视频有帧 N ，合成的动画序列的长度为帧 $N - \tau$ 。这是由于变形的残差序列有 $N - 1$ 帧，因而合成的流体动画序列的长度是从帧 F'_τ 到帧 F'_{N-1} 。

由于当前合成的动画序列中具有模糊的特征结构并与目标图像的外观特征不同，因此，我们使用的是Heeger等人提出的基于金字塔的纹理合成算法[119]，从而恢复每一帧损失的对比差异和源始外观。通过使用这个方法，目标图像和每一动画帧分解为多级的可控向金字塔。然后，在各子带对应的金字塔系数直方图之间进行直方图匹配。最后，根据已更改的动画帧对应的控向金字塔，重建的动画帧能够保持目标图像的源始外观特征。如图4-10所示，给定目标图像（图4-10（a））和合成的一幅动画帧（图4-10（b）），通过使用该方法可获得的外观改善的动画帧（图4-10（c））。

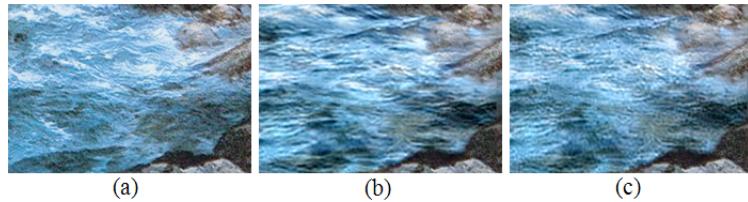


图 4-10 基于直方图匹配的外观调整

Fig 4-10 Original appearance adjustment using histogram matching

4.7 实验结果与分析

图4-11给出本章改进的系统和现有流体动画合成系统的比较。给定目标图像（图4-11（a））和样本视频（图4-11（b）），现有流体动画系统通过拷贝样本视频的残差子块来生成目标图像的残差序列。如图（图4-11（c1-c5））所示，这些传递的残差图像中通常显现严重的块的重复。当在流体动画合成过程中使用这些残差序列，那么在合成的动画序列中容易出现显著的视觉不连续的外观特征（图4-11（e1-e6））。与本章改进的系统进行比较，我们的方法首先根据目标图像中流体区域的大小调整所选的样本视频；然后，我们根据目标图像的流场信息变形已缩放样本视频的残差序列，从而获得目标图像的残差序列（图4-11（d1-d5））。这种全局地方法能够保持样本视频中流体特征的完整性。因此，我们的系统通过使用这种残差序列能够在合成的流体动画在很好地保持大尺度的流体特征（图4-11（f1-f6））。

在图4-12中，我们使用本章改进的系统处理包含多种类型场景的单幅图像。图中第一列（图4-12（a））为具有三种场景的目标图像，分布包括河流、火焰和烟雾；第二列（图4-12（b））是选定的样本视频；第三列（图4-12（c））是根据给定的目标特征在构建的流体视频种子找到的具有相似场景的视频；最后四列（图4-12（d1-d4））给出的是部分合成的流体动画序列。如这些给出的例子所示，场景中由似流体的区域和静止的物体共同组成。此时，通过采用图像抠图方法从目标图像中抽取alpha遮片是有用的，这使得在合成的流体动画中能够恢复这些静止的物体。此外，特别是对于火焰和烟雾的流体动画合成，即使估计的目标图像的平均图像具有静态的形状轮廓，而在合成的动画序列中立体区域的形状是发生变化的。其中关键在于我们在进行动态流体特征融合时使用了已抽取的alpha遮片。

如图4-13所示，给定的目标图像（图4-13（a））中包含多个不连通的流体

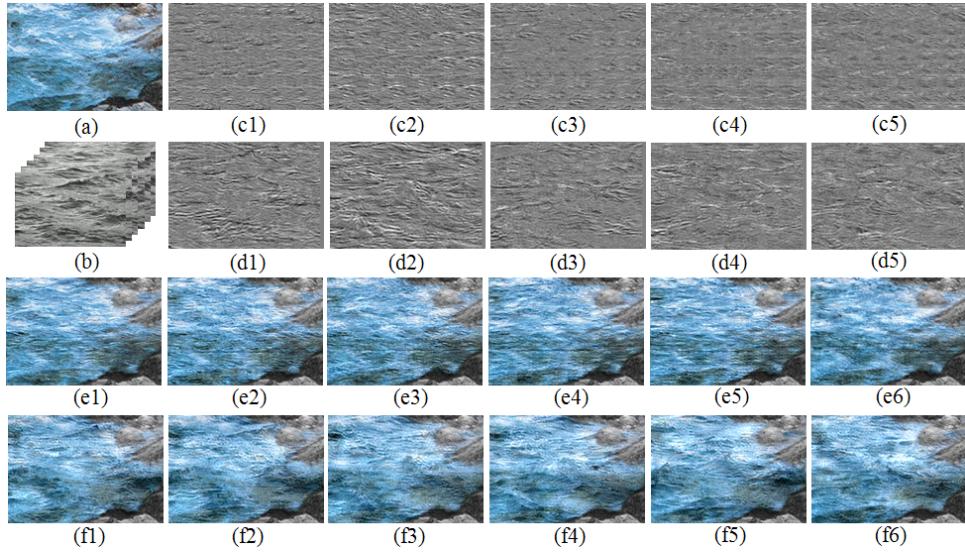


图 4-11 与现有流体动画合成系统的比较 [79]

Fig 4-11 A comparison with the existing flow animation system [79]

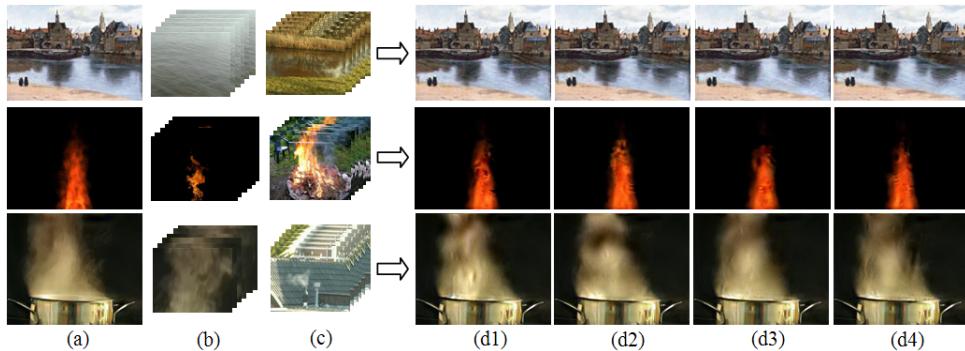


图 4-12 单幅图像的流体动画合成

Fig 4-12 The synthesized flow animations from various types of fluid images

区域，分别用黄色和蓝色的矩形框标记。对应选择的样本视频（图4-13（b）），以第一幅图像（桂林山水场景）为例，其中似流体区域分别是飘动的云和流动河水。对于给定的样本视频，我们采用基元图割的细缝缩放方法调整这些视频的大小。而在第二幅图像中，由于所选样本视频中包含影响流体动画合成的其他干扰物体，如瀑布场景样本视频中的岩石和河流场景样本视频中的鸭子等，由此，我们采用图像裁剪的方法获得仅由流体部分组成的场景视频。基于从构建的视频库中选择的流体视频（图4-13（c）），本章改进的系统首先分别对这些

不连通的流体区域进行动画合成，然后通过组合这些分离的视频帧来产生最终的连续流体动画（图4-13（d1-d4））。

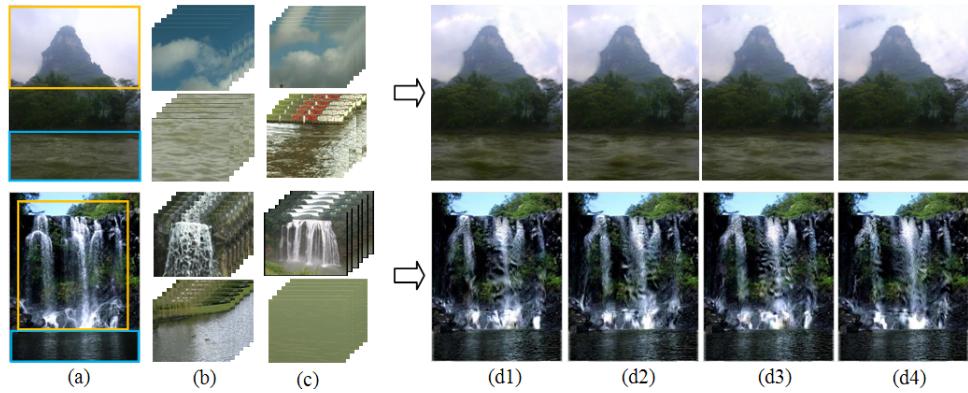


图 4-13 具有不连通流体区域的单幅图像的流体动画合成

Fig 4-13 Flow animation synthesis from single images with unconnected fluid regions

表格1统计了图4-11到图4-13中各样例在流体动画合成中的相关信息。如表格1所示，第二列给出的是已合成流体动画的分辨率；从第三、四列中可以看出，通常目标图像中的似流体区域和所选样本视频的大小是不相同的。本章系统需要花费3-5分钟时间进行样本视频的调整，由于基于3D图割的细缝缩放过程是非常耗时的。然而，若采用传统的线性缩放和图像裁剪方法，这将较少视频调整过程的计算量。用于场景匹配的时间主要取决于视频库中视频帧的总数，如本系统需要花费大约10分钟的时间用于河流场景的匹配搜索和需要花费大约3-7分钟的时间用于火焰、烟雾等场景的匹配搜索。这主要是因为流体视频库中有242个水、河流和瀑布的场景视频、110个火焰场景视频、55个烟雾场景视频和72个云场景视频。由于我们会预先抽取目标图像和各视频帧的SIFT特征和各视频帧的光流场，因此从单幅图像中自动进行运动估计的过程是快速的。参数 τ 是视频重构的重要参数，它的取值控制并影响差异图像生成的质量。当处理剧烈运动变化的视频时，我们通常设置 τ 较大的值；而的 τ 取值是足够小的，当处理的视频呈现动态平稳的特征。最后一列给出已合成流体动画的各结果的总帧数，即合成视频的长度。

本章改进的流体动画合成系统中仍存在如下不足：1) 我们需要手动地选取适合目标图像的样本视频。如果使用不合适的样本视频进行流体动画合成，那么在合成的结果将可能不符合实际的流体运动；2) 尽管自动的流场估计能

够很大程度地减少用户的交互，但是我们无法在传递的流场过程中进行流体运动方向和速度的控制；3) 我们改进的系统还无法实时地处理高清晰的单幅图像和样本视频。此外，当传递样本视频的流体特征时，由于这些特征覆盖了目标图像中的结构特征，因此合成的流体动画无法保持目标图像的外观特征。

表 4-1 流体动画合成的相关信息

Table 4-1 The related information for the synthesis of flow animations

场 景	分 辨 率	流体 区域	样 本 视 频	视 频 缩 放	场 景 匹 配	运 动 传 递	τ	帧 数
I(河流)	176×254	176×254	112×168	5	15	40	5	59
II(河流)	208×298	109×298	288×325	2	16	10	5	150
III(火焰)	240×320	170×90	240×320	1	7	10	12	88
IV(烟雾)	202×247	156×160	288×352	3	5	12	7	96
V(云 & 河流)	384×288 112×288	187×288 288×352	288×352	3 5	3 12	15 10	5 5	146 146
VI(瀑布 & 河流)	400×310 75×310	275×245 288×352	288×352	1	5	20	12	100 100

4.8 本章小结

本章改进的流体动画系统能够从单幅流体动画中合成连续的流体动画。基于从目标图像中抽取的流体区域的alpha遮片，本章方法采用细缝的视频缩放方法调整所选样本视频。因此，我们的系统能够全局地传递样本视频的流体特征，使得合成的流体动画能够保持较大尺度的流体结构。另一方面，基于构建的流体视频库，本章系统通过执行最佳匹配搜索过程，从而为目标图像选择具有相似场景的匹配视频帧。通过自动地传递视频帧的运动来获得目标图像的流场信息，这个过程能够极大地减少用户交互。然而，如何保持目标图像的源始外观特征仍然是我们面临的主要挑战。此外，我们也将继续构建一个资源丰富的流体视频库，使得该视频库尽可能的包含具有多种流体动态场景的视频。

第五章 基于可并行图像分割的灰度图像彩色化和相关应用

5.1 引言

近年来，对经典黑白电影和灰度照片彩色化的需求已经推动了大量的相关研究工作。在一类彩色化方法 [82, 85] 中，用户需要在灰度图像中指定许多颜色线条（scribbles），这类技术采用颜色全局优化过程将这些用户给定的信息传递到图像中的其他区域。然而，添加颜色线条的工作是繁琐的，特别是处理具有复杂细节的图像。另一类彩色化方法 [87, 92, 94] 是使用具有相似场景的彩色图像作为参照样本，通过传递参照彩色图像的颜色信息实现彩色化的效果。然而，尽管样本彩色图像的使用能够极大地减少用户的交互，但是彩色化结果的质量却很大程度地依赖于选择的样本图像。更详细地，当参照样本图像与目标灰度图像具有不同的光照条件时，这将导致一些明显的彩色化误差。

在设计灰度图像彩色化方法时，我们需要考虑以下两个问题：1) 如何自动生成灰度图像的调色板，即无需用户在图像中进行颜色标注；2) 根据这些颜色线条，如何准确地传递颜色信息到待上色物体的各个区域。目前，现有彩色化方法已经针对上述问题提出了各自的解决方案。例如：基于纹理的相似性度量，方法 [92] 为灰度图像的每一像素点在给定的样本彩色图像中找到最佳匹配，然后仅分配颜色给具有较高置信度的像素点。这个结果即为自动获得的调色板。然而，该方法采用颜色优化过程 [82] 进行颜色传递，这无法避免在具有模糊或低分辨率的物体边界轮廓处发生颜色混合或溢出等瑕疵；方法 [85] 利用基于纹理的水平集图像分割方法在纹理模式连续的区域上进行轮廓传播并同时传递颜色信息。该方法虽然能准确地传递颜色到图像中各区域的边界，但是该方法需要用户事先提供图像中各区域的颜色线条。

在本章中，通过结合上述两种彩色化方法的优势，我们提出了一种混合的灰度图像彩色化方法。类似于彩色化方法 [92]，本章提出的方法需要根据给定的样本彩色图像获得对应于灰度图像的调色板。这个过程的关键在于准确地分类灰度图像的各像素点并分配合适的颜色。由此，这需要我们对参照图像的各区域和对灰度图像的各像素之间的对应关系进行分析。更具体地，我们首先采

用自动分割或手工标注的方法识别参照图像中的各不同区域；然后，我们采用基于伽伯小波变换的纹理特征统计方法构建参照图像的特征空间，并采用有监督学习算法分类灰度图像的像素。此时，根据这个构建的特征空间，我们可建立灰度图像中各像素点与参照图像中各区域的一一映射关系。最后，根据基于图像空间的投票机制，该过程执行空间一致的颜色传递，从而自动地产生最终的调色板。

另一方面，根据获得的灰度图像的调色板，本章提出的方法自动地在具有连续纹理模式的区域内传递颜色信息。在这个过程中，同样基于纹理特征的相似性，我们利用可变的水平集方法在具有相似纹理模式的区域内进行曲线传播，从而实现图像区域的并行分割。这个传播能够在呈现突变纹理模式的区域边界处停止，即使区域的边界是模糊的或低分辨率的。在完成区域分割之后，我们采用多种方式 [85]对这些区域进行上色，同时保持源纹理模式的结构特征。本章方法还将上述可并行水平集图像分割方法应用于具有重复场景元素的图像抠取，并对提取的内容进行替换、重排等编辑处理。

5.2 基于样图的调色板的生成

这个阶段的主要任务是参照给定的样本彩色图像分类灰度图像中的每一像素点并为其中具有较高置信度的像素点分配合适颜色，从而获得输入图像的调色板。

5.2.1 基于伽伯小波变化的纹理特征分析

为了有效地分类灰度图像的各像素点，我们使用纹理作为主要的图像特征。本章节算法采用基于伽伯滤波（gabor filter）的多分辨率表示统计图像的纹理特征。文献 [124] 中已经证明伽伯表示是一种最佳的纹理特征描述方式，它能够在空间域和频率域上最小化联合二维的不确定性。这些滤波通常可以用于多方向和多尺度的边的检测，而对图像区域内的这些细微特征的统计即为图像中潜在的纹理信息。

二维的伽伯函数 $g(x, y)$ 和它的傅立叶变换 $G(u, v)$ 可以用如下等式表示：

$$g(x, y) = \left(\frac{1}{2\pi\sigma_x\sigma_y} \right) \exp \left[-\frac{1}{2} \left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right) + 2\pi j w_x \right] \quad (5-1)$$

$$G(u, v) = \exp\left\{-\frac{1}{2}\left[\frac{(u - w)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\} \quad (5-2)$$

其中， $\sigma_u = 1/2\pi\sigma_x$ 和 $\sigma_v = 1/2\pi\sigma_y$ 。一类自相似滤波函数 $g(x, y)$ （伽伯小波，gabor wavelet）可以通过膨胀和旋转小波系数获得：

$$g_{m,n}(x, y) = a^{-m}G(x', y') \quad (5-3)$$

给定 $a > 1$, m 和 n 是整数：

$$x' = a^{-m}(x \cos \theta + y \sin \theta) \quad (5-4)$$

$$y' = a^{-m}(-x \sin \theta + y \cos \theta) \quad (5-5)$$

其中， $\theta = n\pi/N$; N 是方向的总个数；尺度因子 a^{-m} 确保能量不依赖于 m 。

给定输入图像 $I(x, y)$, 它的伽伯小波变换可定义如下：

$$\mathbb{W}_{m,n}(u, v) = \int_{\Omega} I(x, y) G_{m,n}^*(u - x, v - y) dx dy \quad (5-6)$$

其中， $g_{m,n}$ 由上述公式 (5-1) 到公式 (5-5) 计算得到；上标*表示复共轭；下标 m 和 n 各自表示尺度和方向的索引。然后，我们计算伽伯小波变换系数的均值 $\mu_{m,n}$ 和标准差 $\sigma_{m,n}$:

$$\mu_{m,n} = \int \int |\mathbb{W}_{m,n}(x, y)| dx dy \quad (5-7)$$

$$\sigma_{m,n} = \sqrt{\int \int |\mathbb{W}_{m,n}(x, y) - \mu_{m,n}|^2 dx dy} \quad (5-8)$$

纹理模式的统计特征定义为由多尺度和多方向的 $\mu_{m,n}$ 和 $\sigma_{m,n}$ 组成的特征向量：

$$T = [\mu_{0,0} \sigma_{0,0} \mu_{0,1} \dots \mu_{M-1,N-1} \sigma_{M-1,N-1}] \quad (5-9)$$

其中， M 和 N 分布表示尺度和方向的总数。

5.2.2 特征空间和分类

本章算法采用监督学习算法分类灰度图像的像素点，其中以参照图像的亮度通道图像及其对应的分割图像作为学习算法的训练数据。对样本彩色图像的分割是指将该图像划分成多个不连通的区域，并且为每一区域分配唯一的标

记。这些分割的区域不需要铺满整幅图像，但是每一区域的颜色和纹理应该是统一的。我们可以采用自动图像分割算法或手动标记方法获得这些区域的划分。

根据提供的训练数据，我们首要的任务是构建一个特征空间和一个分类器。这些已分割区域中的任一像素点都可以定义为带标记的特征向量，它对应特征空间的一个点。在5.2.1节中，我们指出分类依赖于区别纹理特征的差异，这需要我们为这些像素点附加描述纹理模式的特征向量。我们统计以每个像素点为中心的 $K \times K$ 邻域上的伽伯小波变换系数，这些系数的均值和标准差即为这些像素点对应的纹理特征向量。该计算过程产生一个 MN 维的带标记的特征空间（ M 和 N 分布表示尺度和方向的总数），特征空间中的每一点对应训练数据集中的一个像素点。这个特征空间的大小与划分区域内像素点的总数有关。此外，我们根据样本彩色图像中纹理模式的尺度大小经验地设置 k 的取值。

将样本彩色图像的亮度图像表示成带标记的特征空间之后，我们需要分类灰度图像中各像素点，即为这些像素点找到最佳匹配的区域。一般来说，训练数据集中的像素点在特征空间中可能没有形成精确分离的簇。因此，我们采用线性辨别分析算法（Linear Discriminant Analysis, LDA）获得高维特征空间的投影矩阵 W ，从而投影高维特征空间到低维的子空间。已知高维特征空间 $\mathbf{x} = \{x^1, x^2, \dots, x^n\}$ ， n 为特征空间中点的总数；这些特征点可分为 $C = [\omega_1, \dots, \omega_C]$ 类，我们首先计算类内分离矩阵 S_W ：

$$S_W = \sum_{i=1}^C S_i \quad (5-10)$$

其中， $S_i = \sum_{\mathbf{x} \in \omega_i} (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T$ 和每类特征向量的均值 $\mu_i = \frac{1}{n_i} \sum_{\mathbf{x} \in \omega_i} \mathbf{x}$ ， n_i 为第 i 类特征点的总个数。然后，我们计算类间分离矩阵 S_B ：

$$S_B = \sum_{i=1}^C n_i (\mu_i - \mu)(\mu_i - \mu)^T \quad (5-11)$$

其中， $\mu = \frac{1}{n} \sum_{\forall \mathbf{x}} \mathbf{x} = \frac{1}{n} \sum_{\mathbf{x} \in \omega_i} n_i \mu_i$ 。此时，具有最大聚簇信息的投影矩阵 W 可近似定义为矩阵 $S_W^{-1} S_B$ 的最大特征值对应的特征向量。

对于灰度图像中的每一像素点，相同地计算该点的特征向量，并根据计算得到的投影矩阵进行降维。当分类灰度图像中各像素点时，我们定义两像素点之间的距离为：

$$D(p, q) = \|sT(p) - sT(q)\|_2 \quad (5-12)$$

其中, $T(\cdot)$ 是对应像素点的特征向量, s 为投影操作。最后, 该算法使用KNN (k nearest neighbor) 分类器在降维的特征空间中为每一像素点选取前 K 个最邻近的特征向量, 其中出现最多次的标记即为当前像素点的匹配区域。参照分割的样本彩色图像, 图5-1给出灰度图像的分类结果。其中图5-1 (a) 为样本彩色图像, 5-1 (b) 为对应的分割结果, 5-1 (c) 为灰度图像, 5-1 (d) 为分类的结果。



图 5-1 基于相似光照值和纹理模式统计的KNN匹配

Fig 5-1 Knn matching based on similar luminance and texture pattern statistics

5.2.3 调色板的生成

上节中阐述的KNN分类器虽然能获得较好的分类结果, 但是在给出的结果中(如图5-1 (d) 所示)仍然存在许多错误分类的像素点。对于灰度图像中的任一像素点 p 的邻域 $N(p)$, 即围绕像素点 p 的 $k \times k$ 大小的区域, 其中邻域内各像素点可能与当前所选像素点具有不同的标记。为了纠正这种情况, 我们使用邻域内 $N(p)$ 中占主导地位的标记替换为像素点 p 的标记。此时, 占主导地位的标记具有最高置信度 $conf(p, l)$, 并定义为如下公式:

$$conf(p, l) = \frac{\sum_{q \in N(p, l)} W_q}{\sum_{r \in N(p)} W_r} \quad (5-13)$$

其中, $N(p, l)$ 为邻域 $N(p)$ 内具有标记为 l 的一组像素点集; W_q 权重取决于像素点 q 和它的最佳匹配 M_q 之间的距离 $D(p, M_q)$ 。 M_q 是像素点 q 在特征空间中的最邻近点, 并与该像素点具有相同的标记。权重可由如下公式计算得到:

$$W_q = \frac{exp(-D(q, M_q))}{\sum_{r \in N(q)} exp(-D(r, M_r))} \quad (5-14)$$

由上可知, 置信度 $conf(p, l)$ 的值是较高的当该邻域 $N(p)$ 内的所有或大多数像素点的标记为 l 。而在区域边界处或低分辨率区域内, 它的取值是较低的。通过采用上述滤波方式, 该算法能够进一步改善分类结果, 如图5-1 (e) 所示。

在分类灰度图像的像素点之后，该算法需要传递样本彩色图像 L 的颜色信息给灰度图像 I 中的各像素点。我们在YUV颜色空间上进行颜色传递，其中 Y 是亮度通道， U 和 V 是色度通道。 $C(p)$ 定义为像素点 p 的色度坐标，该像素点的颜色值为其邻域内具有相同标记的像素点的颜色值的加权平均：

$$C(p) = \sum_{q \in N(p,l)} W_q C(M_q(p)) \quad (5-15)$$

其中， M_q 为像素点 $q \in I$ 在样本彩色图像 L 中的最佳匹配； $M_q(p)$ 是一个像素点，它在图像 L 中相对于最佳匹配 M_q 的位置与像素点 p 相对于像素点 q 的位置相同。也就是说，我们依次检查 $N(p,l)$ 中的每一个像素点，而这些像素点都能在图像 L 中为像素点 p 找到一个候选的颜色值。由此，该像素点的颜色值为这些候选颜色值的加权平均。事实上，该算法通过这种方式可以为灰度图像中的每一像素点都分配一个颜色值。然而，由于一些区域内仍存在错误的分类，因此我们仅为置信度 $conf(p,l)$ 大于0.5（归一化的值）的像素点传递颜色信息。由此，通过上述分类和颜色传递两个步骤，该算法能够自动生成灰度图像对应的调色板。如图5-2所示，第二列显示灰度图像中置信度大于0.5的各像素点用红色标记，最后一列给出灰度图像对应的的调色板。

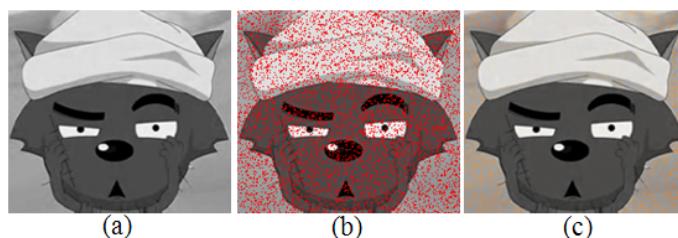


图 5-2 自动生成的调色板

Fig 5-2 Automatic color palette generation

5.3 基于并行可变水平集的图像分割

在灰度图像彩色化过程中，图像区域的分割是为了确定区域的边界轮廓，使得颜色信息的传播能够在区域边界处停止，避免发生颜色溢出、混合等瑕疵。在本章节中，基于纹理模式的相似性，我们改进传统的水平集方法，使得能够执行并行的图像分割。这能够极大地加快了彩色化的速率。

5.3.1 传统水平集 (Level Set) 方法

水平集方法 [125] 是由 Osher 和 Sethian 两人最早提出的零曲面方法，能够用于捕获动态的界面和形状。在图像处理和计算机视觉的应用中，Caselles [126] 和 Malladi [127] 等人在以动态轮廓模型 (active contour model) [126] 为基础的情况下将水平集方法分用于图像分割。给定一幅图像，动态轮廓模型的基本思想是演变一条曲线用以检测图像中的物体，这条曲线按照法线方向进行移动并在物体的边界处停止。已知 Ω 为有界子区域， $\partial\Omega$ 是它的边界； u_0 为给定的图像，以及 $C(s)$ 为参数化的曲线，该动态轮廓模型表示如下：

$$J_1(C) = \alpha \int_0^1 |C'(s)|^2 ds + \beta \int_0^1 |C''(s)| ds - \lambda \int_0^1 |\nabla u_0 C(s)|^2 ds \quad (5-16)$$

其中， α 、 β 和 λ 是正值的参数。上述公式中的前两项（内部能量项）控制曲线的平滑性，而第三项（外部能量项）推动曲线朝图像中的物体移动。我们通过最小化上述能量公式定位曲线在具有最大梯度 ∇u_0 的点（这可以看出是一个边界检测算子），并保持曲线在物体边界处是平滑的。

使用水平集方法描述曲线演变过程，曲线 C 可用一个 lipschitz 函数 ϕ 隐式地表示为 $C = \{(x, y) | \phi(x, y) = 0\}$ 。由此，曲线的演化指的是水平集函数 $\phi(t, x, y)$ 在 t 时刻的零水平级 (zero level) 曲线。曲线的演化能够表示为：

$$\frac{\partial C(s, t)}{\partial t} = FN \quad (5-17)$$

其中， F 是速度函数，用于控制曲线的运动； N 是曲线的法线向量。假设水平集函数在曲线内部取小于零的值，在曲线外部的取值大于零，朝内的法线向量可表示为 $N = -\nabla\phi/|\nabla\phi|$ 。然后，曲线演变等式 (5-17) 转换为求解如下偏微分方程 (partial differential equation, PDE)：

$$\frac{\partial\phi}{\partial t} = F|\nabla\phi|, \quad \phi(0, x, y) = \phi_0(x, y) \quad (5-18)$$

其中，集合 $\{(x, y) | \phi_0(x, y) = 0\}$ 定义为初始轮廓。这种以水平集形式表达的动态轮廓模型通常称之为几何动态轮廓模型 (geometric active contour model)。水平集方法的一个优势是它能够表示曲线复杂的拓扑结构并处理拓扑的变化，如分裂和合并。此外，水平集方法的另一特征是它能够在一个规则的笛卡尔网格上进行离散的数值计算。

然而，水平集方法存在的一个显著问题是水平集函数在曲线演化过程中呈现不规则性。在传统的水平集方法中，这种不规则性容易导致数值的错误以及最终破坏曲线演变的稳定性。为了克服这个难点，文献 [128] 中通过引入重新初始化（reinitialization）数值纠正操作重建水平集函数的规则性，使得保持曲线演变过程是稳定的。我们在每次演化过程停止时执行重新初始化操作，此时退化的水平集函数转化为一个带符号的距离函数。重新初始化的一种标准方法是求解如下演变等式：

$$\frac{\partial \Psi}{\partial t} = \text{sign}(\phi_0)(1 - |\nabla \Psi|) \quad (5-19)$$

其中， ϕ_0 是初始的水平集函数， $\text{sign}(\cdot)$ 是符号函数。理想地，上述等式的稳定状态的解仍然是一个带符号的距离函数。尽管重新初始化操作能够保持水平集函数的规则性，但是它可能错误地移动零水平集远离期望的位置，例如，初始 ϕ_0 是不平滑的或者在界面的一边是非常陡峭的。

可以看出，一方面，重新初始化过程是非常复杂的、耗时的，且具有一定的副作用。另一方面，传统的水平集演变等式（公式 (5-17)）中定义的用于引导零水平集运动的速度函数并不能保持水平集函数是一个带符号的距离函数。而在实践中，我们又需要使用重新初始化操作驱使水平集函数是一个带符号的距离函数。这个理论上和实际上的不一致使得我们无法确定在什么时候以及如何执行重新初始化操作。因此，在使用水平集方法进行图像分割时，我们应尽可能地避免这个重新初始化操作。

5.3.2 初始轮廓的放置

本章中使用的水平集方法是一种局部的图像分割方法，它仅需要用户在待分割的物体上放置初始的轮廓（即初始的水平集函数），该轮廓不断演变并在物体的边界处停止。明显地，水平集方法也需要用户标记待分割的物体。对于给定的一幅图像（如图5-1 (c) 所示），我们可以看到图中待分割的灰太狼由不同子区域组成：帽子区域、眼睛区域、眉毛和鼻子、嘴巴区域、耳朵和脸颊区域，这些子区域将根据参照样本彩色图像（如图5-1 (a) 所示）绘制不同的颜色。当使用水平集方法进行图像区域分割时，我们需要提供大量的交互操作来分割这些子区域，特别是处理具有许多细小区域的图像。

针对这一问题，本章中提出的解决方案是在同类子区域中自动检测具有相似特征的像素点，从而在确定初始轮廓的放置位置。我们使用颜色、纹理信息

和像素点的标记信息等特征定义区域相似性距离公式：

$$D(p, q) = d_{texture}(T(p), T(q)) + d_{color}(C(p), C(q)) + \omega L_{Label}(N(p), N(q)) \quad (5-20)$$

其中， p 和 q 为图像中的任意两像素点。上述公式的前两项定义为基础项， $d_{texture}(T(p), T(q))$ 用于度量围绕像素点 p 和 q 的 $k \times k$ 邻域之间的纹理特征差异，为纹理特征向量。 $d_{color}(C(p), C(q))$ 用于计算两像素点之间的灰度值或颜色值的差异；而最后一项为增强项， $L_{Label}(N(p), N(q))$ 用于统计像素点 p 和 q 各自邻域内相对应位置上具有不同标记像素点的总数目， ω 为增强系数。当该项的取值较大时，说明当前待检测的像素点 q 位于区域的边界处；当该项取值较小时，则说明当前待检测的像素点 q 位于区域的中心位置。由此，我们说两个子区域是相似的当且仅当这两个子区域具有足够多的具有最小相似距离($\min D(p, q)$)的像素点对。此外，我们还增加度量像素点 p 和 q 之间的欧氏距离 d_{dist} ，使得尽量避免在同一区域内进行相似像素点的检测。

以检测图5-1 (c) 中眉毛、鼻子和嘴巴区域为例，已知灰度图像各像素点的分类结果和对应的调色板，详细的操作流程如下：

1. 首先在眉毛区域内选择一组像素点集，它们对应调色板图像中已经分配颜色的像素点；
2. 将这组点集中的每一像素点与具有相同的标记区域内的各像素点进行特征相似性度量。此时，任一像素点对的距离 $D(p, q)$ 小于用户给定的阈值 D_{Thres} 时，我们记录这些检测点的位置。需要注意的是，待检测的点也是调色板图像中已经分配颜色的像素点；
3. 在用户给定的参照像素点集和检测的像素点集中随机选择部分像素点，并以这些像素点的位置为中心放置大小为 $m \times n$ 的矩形框作为初始的轮廓。此时，我们根据检测点所在区域的大小确定 m 和 n 的具体取值。

5.3.3 可变水平集表达式

根据5.3.1节中讨论的传统水平集方法存在的问题，本章方法采用可变的水平集方法 [129] 驱使水平集函数在演变过程中始终接近于带符号的距离函数，从而完全消除代价昂贵的重新初始化过程。可变能量函数由内部能量项和外部能量项组成：

$$\varepsilon(\phi) = \mu P(\phi) + \varepsilon_m(\phi) \quad (5-21)$$

其中， $\mu > 0$ 用于控制水平集函数 ϕ 相对于符号距离函数的偏移程度。由于符号距离函数满足 $|\nabla\phi| = 1$ 的特点，因此，内部能量项定义如下：

$$p(\phi) = \int_{\Omega} \frac{1}{2}(|\nabla\phi| - 1)^2 dx dy \quad (5-22)$$

这个积分等式可以描述函数 ϕ 在图像域 $\Omega \subset \Re^2$ 上是如何近似于一个符号距离函数。这个等式在可变能量函数中具有重要的作用，它使得水平集函数在演变过程中自动地保持为符号距离函数。外部能量项 $\varepsilon_m(\phi)$ 用于推动零水平曲线的运动，由如下等式进行定义：

$$\varepsilon_m(\phi) = \varepsilon_{g,\lambda,\nu}(\phi) = \lambda L_g(\phi) + \nu A_g(\phi) \quad (5-23)$$

其中， $\lambda > 0$ 和 ν 是常数。而项 $L_g(\phi)$ 和 $A_g(\phi)$ 的定义分别由如下等式给出：

$$L_g(\phi) = \int_{\Omega} g\delta(\phi)|\nabla\phi| dx dy \quad (5-24)$$

$$A_g(\phi) = \int_{\Omega} gH(-\phi) dx dy \quad (5-25)$$

其中， δ 是狄拉克函数， H 是阶跃函数函数。能量项 $L_g(\phi)$ 的几何意义是描述水平函数的零水平曲线为可微的参数曲线，它可以计算零水平曲线的弧长。能量函数 $A_g(\phi)$ 计算区域 $\Omega_{\phi}^- = \{(x, y) | \phi(x, y) < 0\}$ 的面积，可用于加速曲线的演化。该能量项的系数 ν 可以取正数值或者负数值。当初始的轮廓放置在区域内部时，系数通常取负数值来加速轮廓的向外扩张；而当初始的轮廓放置在区域外部时，系数 ν 通常取正数值来加速轮廓的向内收缩。根据定义的边界检测函数 g ，该能量项驱动零水平曲线到区域的边界。我们将在下一章节详细讨论边界检测函数。

方法 [129] 计算可变能量函数的加托导数 (Gateaux derivative)，并满足如下等式：

$$\frac{\partial\phi}{\partial t} = -\frac{\partial\varepsilon}{\partial\phi} \quad (5-26)$$

此时，水平集函数的演变可以写成如下形式：

$$\frac{\partial\phi}{\partial t} = \mu[\Delta\phi - \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)] + \lambda\delta(\phi)\operatorname{div}\left(g\frac{\nabla\phi}{|\nabla\phi|}\right) + \nu g\delta(\phi) \quad (5-27)$$

其中，等式的第二、三项与边界检测函数有关，它们负责驱动零水平曲线到区域的边界；而第一项与散度率因子 $1 - 1/|\nabla\phi|$ 有关，如果 $|\nabla\phi| > 1$ ，则散度率是正数值并减少梯度 $\nabla\phi$ ；如果 $|\nabla\phi| < 1$ ，则散度率是负数值并增大梯度 $\nabla\phi$ ；由此，可以保持零水平曲线在演变过程中始终近似于符号距离函数。

5.3.4 边界检测函数定义

在使用上述可变水平集方法进行图像分割时，初始轮廓的演变过程受边界检测函数的影响，使得初始轮廓向区域的边界移动。边界检测函数 g 一般定义为一个正数值的且逐渐递增的函数，依赖于图像的梯度并使得满足如下条件：

$$\lim_{z \rightarrow \infty} = 0 \quad (5-28)$$

由此，边界检测函数 g 可以定义为如下形式：

$$g = \frac{1}{1 + |\nabla G_\sigma * I|} \quad (5-29)$$

其中， $G_\sigma * I$ 表示图像 I 和高斯核函数的卷积。由此可见，当曲线传播到具有最大梯度的像素点处时，该函数的取值为零。因此，初始轮廓能够在边界处停止，由于边界处的梯度是较大的。然而，由于离散的梯度是界限的以及边界检测函数在边的值并不为零，这使得传播的曲线肯穿过区域的边界。也就是说，如果图像中含有大量的噪声，在使用较强高斯滤波进行去噪的同时可能使得图像中边的特征变得模糊，那么轮廓不会在边界处停止传播。

为解决这一问题，假设图像中具有明显可辨的纹理特征，此时我们可以根据纹理模式的连续性 [85] 来定义上述函数，且称之为停止条件函数：

$$g' = \frac{1}{1 + |D(T_{contour}, T_{front})|} \quad (5-30)$$

其中， $T_{contour}$ 是具有相似特征区域上由所有初始轮廓所覆盖像素点的主要纹理模式特征向量。这里所指的主要模式特征确定如下：已知所有初始轮廓，首先计算轮廓上所有像素点的纹理特征（由5.2.1节中定义的公式计算得到），然后使用k-means聚类方法分类这些像素点并确定主要的一类簇，最后计算这个簇的平均纹理模式特征。由此，这个平均模式特征向量即定义为 $T_{contour}$ 。这样做的目的是排除其他纹理模式的干扰。 T_{front} 是图像中传播界面的各像素点的模式特征向量，按照同上的方式计算得到。函数 D 是距离函数，用于测量初始轮廓和传播界面之间的纹理模式的差异。此时，我们仅使用平方差的和作为距离度量。根据上述基于纹理模式连续性的停止条件函数，初始轮廓将在呈现纹理模式突变的地方停止传播，使得停止函数的取值趋于零。由此，即使图像中边的特征是模糊的或者是低分辨率的，只要是具有丰富的纹理信息，我们可使用上述定义的停止条件函数控制曲线的演变。

5.3.5 数值求解框架

在传统的水平集方法中，我们需要初始化水平集函数为符号距离函数，并且在每一次曲线演变中重新初始化曲线为符号距离函数，从而确保稳定的曲线传播过程。不过，这使得传统的水平集方法只能使用逆风框架（upwind scheme）逼近演变的曲线。而在本章使用的可变水平集方法中，该方法不仅可以完全消除重新初始化过程，而且还不需要将水平集函数初始化为符号距离函数。由此，我们将使用如下函数作为初始的水平集函数 ϕ_0 。已知 Ω_0 是图像域 Ω 上的一块子区域， $\partial\Omega_0$ 是由它的边界上的所有像素点组成的封闭曲线，初始的水平集函数 ϕ_0 定义为：

$$\phi_0 = \begin{cases} -\rho, & (x, y) \in \Omega_0 - \partial\Omega_0 \\ 0, & (x, y) \in \partial\Omega_0 \\ \rho, & \Omega - \Omega_0 \end{cases} \quad (5-31)$$

其中， $\rho > 0$ 是常数。这个定义说明水平集函数在区域 Ω_0 内部的取值是负整数，在区域边界 $\partial\Omega_0$ 上的取值等于零，而在区域 Ω_0 外部的取值是正整数。可以看出，这种形式的初始水平集函数可以从图像域 Ω 的任意区域 Ω_0 计算得到。在演变过程中，水平集函数 ϕ 在整个图像域中并不能保持为近似的符号距离函数。但是，在本章使用的可变水平集方法中，内部能量函数能够保持水平集函数 ϕ 近似为零水平集曲线。

由此，我们可以使用向前差分的框架估计时间偏导 $\frac{\partial\phi}{\partial t}$ ，公式（5-27）的近似估计可以写成：

$$\frac{\phi_{i,j}^{k+1} - \phi_{i,j}^k}{\tau} = L(\phi_{i,j}^k) \quad (5-32)$$

其中，是使用差分框架对公式（5-27）右边的估计式。上述差分形式可以写成如下迭代等式：

$$\phi_{i,j}^{k+1} = \phi_{i,j}^k + \tau L(\phi_{i,j}^k) \quad (5-33)$$

在迭代过程中，我们可以选择足够大的时间步长 τ ，使得尽可能地加快曲线的传播。然而，为了使得这个曲线演变过程是稳定的，时间步长 τ 和系数 μ 需要满足条件关系是 $\tau\mu < \frac{1}{4}$ 。虽然使用较大的时间步长能够加快曲线传播，但是这在定位边界时可能产生一定错误。为平衡时间步长的选择和边界的精确定位，在本章所有实验中我们通常选择 $\tau = [10, 30]$ 。图5-3给出区域分割的结果。其中

图5-3 (b) 和图5-3 (d) 表示在具有相似特征区域上放置初始轮廓；图5-3 (c) 和图5-3 (e) 为使用并行可变水平集方法进行区域分割的结果。

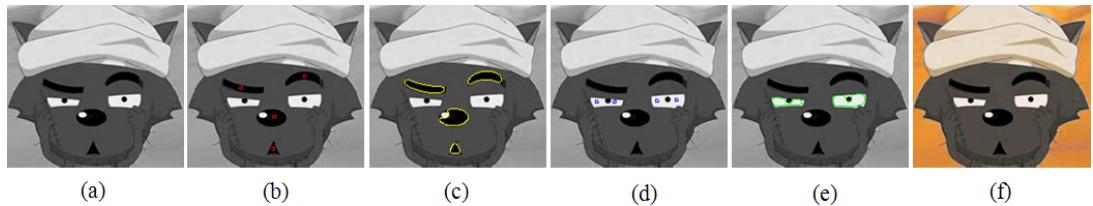


图 5-3 区域分割和彩色化结果

Fig 5-3 Results for region segmentation and colorization

5.4 颜色传播

在获得已分割的区域之后，我们使用多种上色方法 [85] 在这些分割的区域上进行颜色传递，自动生成的调色板提供了这些颜色信息。一种上色方式是保持源始图像中的纹理模式，此时，我们传递调色板中的已经分配颜色值的像素点的颜色信息到区域中其他待上色的像素点。我们约束在YUV 颜色空间上进行上色操作，这些待上色像素点的颜色值定义如下：

$$Y_{output}(x, y) = Y_{palette}(x, y) \otimes |1 - g'(x, y)|^2 \quad (5-34)$$

$$(U, V)_{output} = (U, V)_{palette} \quad (5-35)$$

其中， Y 是光照通道；下标 $output$ 和 $palette$ 分别对应输出和调色板中的颜色； \otimes 是卷积操作； $g'(x, y)$ 由公式 (5-30) 计算得到。由以上计算可以看出，调色板中的信息能够很好地传递到边界上。当图像中具有丰富的边的特征时，上述上色方法能够得到较好的效果。

另一种上色方式是在颜色传递过程中能够保持区域的底纹。为此，在进行颜色传递之前，我们首先在灰度图像中每一像素点的邻域内计算局部亮度：

$$s = f \otimes Y_{image} \quad (5-36)$$

其中， Y_{image} 是灰度图像， f 一个盒式滤波器。由此， Y 通道上待上色像素点的值可以通过已上色的像素点进行线性插值得到：

$$Y_{output} = sY_{palette}, \quad s \in [0, 1] \quad (5-37)$$

而 U 和 V 通道的值同公式(5-35)计算得到。此外，我们可以同时使用这两种方式进行上色，对图5-1(c)的上色效果如图5-3(f)所示。

5.5 实验结果与分析

5.5.1 灰度图像彩色化实验结果与分析

图5-4给出本章所提出的灰度图像彩色化方法和现有两种彩色化方法的比较。方法[92]和[85]中都充分利用了图像的纹理特征，如方法[92]通过分析给定参照图像和待上色灰度图像之间的纹理相似性，并构建灰度图像中每一像素点与参照图像中已分割区域的一一对应关系，从而能够有效地为这些像素点分配颜色。然而，该方法采用方法[82]中的全局优化方法将颜色信息传递给图像中剩余像素点，这容易导致产生大面积的颜色混合区域(图5-4(c))。而方法[85]采用基于纹理的水平集图像分割方法在纹理连续的区域内传递颜色信息。虽然该方法能够产生令人满意的彩色化效果(图5-4(d))，但是需要用户事先在每个区域上进行颜色标记。这不仅需要用户具有丰富的经验，而且需要大量的人工交互。本章提出的方法正是结合了这两种方法的优点：一方面，我们的方法通过自动获取调色板来减少用户交互；另一方面，我们通过确定图像中各区域的边界来避免彩色化过程中出现的颜色溢出和混色等瑕疵。



图 5-4 与现有彩色化方法的比较

Fig 5-4 Comparison with existing colorization methods

本章方法在通过纹理分析分类像素点的基础上，我们采用伽伯小波变换方法替换原方法[92]中使用的离散余弦变换方法进行纹理特征统计。明显地，参照已分割的样本彩色图像(图5-5(a))，我们手工交互地将具有相同颜色的或具有相似纹理模式的标记为同一区域。此时，用肉眼容易分辨的天空、白马、树木和草地均被标记(图5-5(b))。由于伽伯小波变换方法能够从多方向和多

尺度分析纹理结构的相关联性，因此，我们能够很好地描述每一局部小窗口内的纹理特征。基于纹理特性相似性的匹配，给定的灰度图像中的每一像素点都能在参照的划分区域内找到合适的标记（图5-5（d））。由此，根据该分类结果，本章方法自动获得灰度图像对应的调色板，这使得能够将给定样本彩色图像的颜色信息传递给灰度图像，从而获得满意的彩色化图像（图5-5（e））。

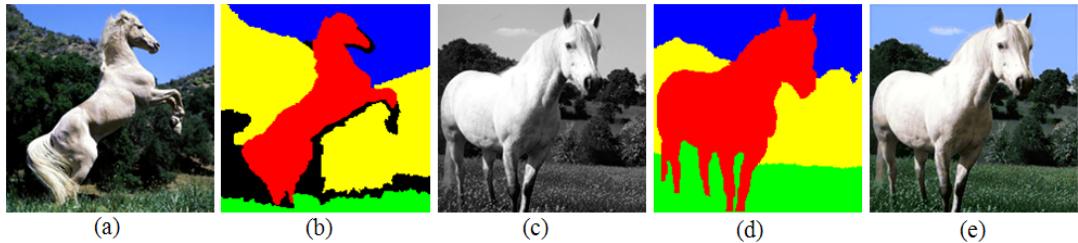


图 5-5 基于纹理特征分类的灰度图像彩色化

Fig 5-5 Colorization from classification based on texture feature

除上述改进之外，我们在本章提出的方法中还改进了基于纹理的水平集方法，使得能够并行地分割图像中的各区域，这些区域必须是具有相同纹理模式的区域。在图5-6给出的彩色化结果中，我们可以看到待上色灰度图像中的主要场景是由多朵郁金香花组成（图5-6（b））。若使用方法 [85]对这些重复的场景元素进行上色，那么这需要用户逐一地在各个物体上标记颜色信息。然而，这是工作量大的，特别是当图像中具有大量的重复场景元素或细小的区域。虽然方法 [85]中使用的水平集方法能够处理纹理模式连续的且是不连通的区域，但必须对这些区域逐一标记。针对这一问题，根据图像中各区域在纹理和颜色等特征上的相似性，我们自动地检测这些具有相同特征的区域。如图5-6（c）所示，具有相似纹理模式和颜色特征的郁金香花区域将被检测出来，各蓝色矩形框为放置的初始轮廓。通过利用水平集方法在拓扑结构方面的可变性，即轮廓传播是可分裂的和可合并的特性，我们采用可变水平集方法并行地对这些区域进行分割。图5-6（e）为最终获得的彩色化效果。

图5-7给出更多彩色化结果。这些待上色的灰度图像是从一段视频中随机选取的几帧，使用本章方法能够获得很好的彩色化效果。其中给出的柯蓝动画序列中包含许多灰度连续的区域，使用本章提出的并行图像分割算法能够快速的进行区域上色。而在另一组动画序列中，区域缺失是我们主要考虑的问题，即后几帧灰度图像中眼睛区域与样本彩色图像中的存在较大差异，但是使用本

章方法依然能够准确地进行传递颜色。

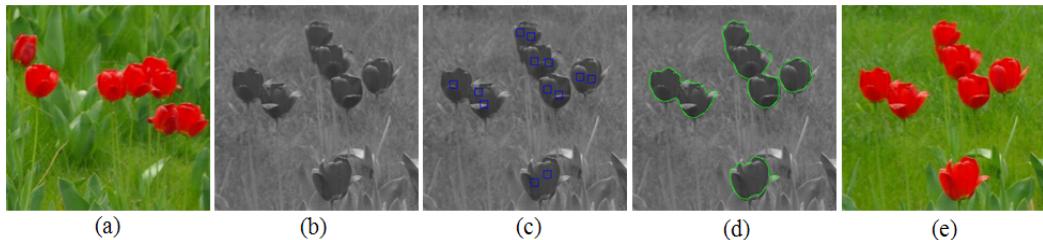


图 5-6 基于并行分割的灰度图像彩色化

Fig 5-6 Colorization based on parallel segmentation

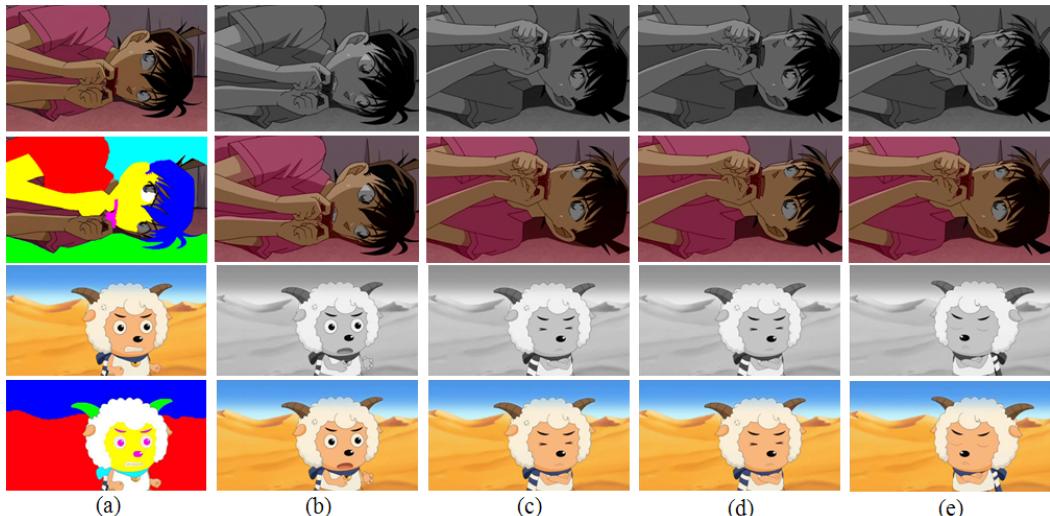


图 5-7 更多彩色化结果

Fig 5-7 More results for colorization

5.5.2 具有重复场景元素的图像抠取与编辑

本章中提出的并行可变水平集图像分割算法的一个重要应用是对图像中重复场景元素进行抽取。在现有图像分割方法中 [106, 107, 130, 131]，抽取这些重复元素（如图5-8（a）中所示的水果拼盘）通常需要进行乏味的和费力的用户交互，即逐一标记感兴趣的前景目标物体和背景。本章所采用的水平集方法是一种基于动态轮廓模型的局部分割方法，我们仅需要提供少量用户交互的即可抽取这些重复的场景元素。也就是说，我们只要在感兴趣的一个物体上进行

标记（如5-8（b）和5-8（c）中分别标记的猕猴桃片和橙瓣），其他剩余的猕猴桃片和橙瓣即可被全部抠取出来，不再需要考虑背景对前景的影响。其中关键是根据重复元素之间在颜色、纹理等图像特征上的相似性，我们通过建立的它们彼此之间的连接关系，从而实现图像的并行分割。需要指出的是，标记的区域应该覆盖了前景物体的所有显著特征，如猕猴桃片的果肉和果核、橙瓣的果肉和橙皮。

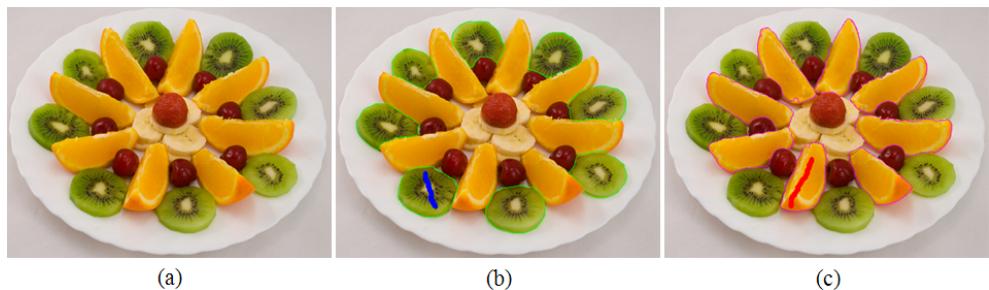


图 5-8 重复场景元素抠取
Fig 5-8 Image cutout for repeated scene elements

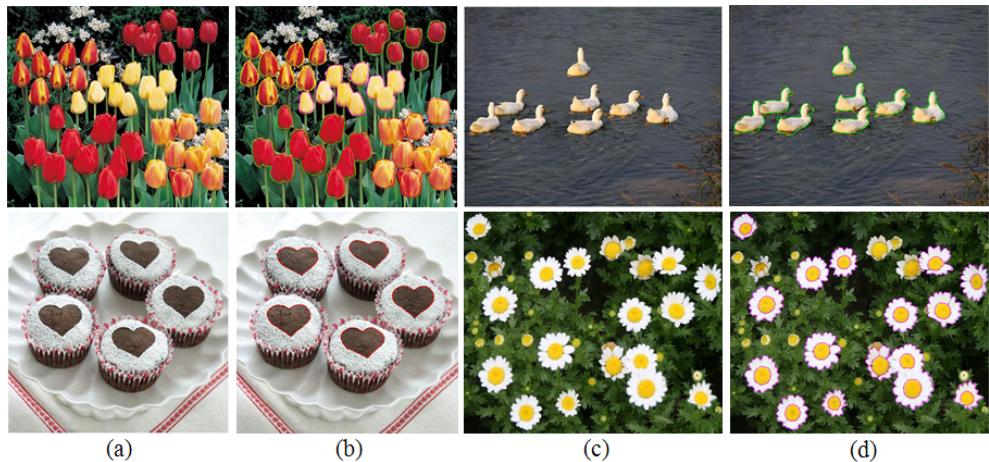


图 5-9 更多重复场景元素抠取结果
Fig 5-9 More image cutout results for repeated scene elements

图5-9给出了更多实验结果，其中以五个颜色各异的郁金香花束为例，相似特征内部和相似特征之间的重复场景元素能够同时进行分割；水鸭和蛋糕的例子相对比较简单，只要标记任一只鸭子和桃心区域，本章算法即可抠取出其

他相似物体；最后一个雏菊花的例子说明的问题是，我们仅抠取出每一朵花的花瓣，这种不同尺度下的分割是现有图像分割方法做不到的地方。

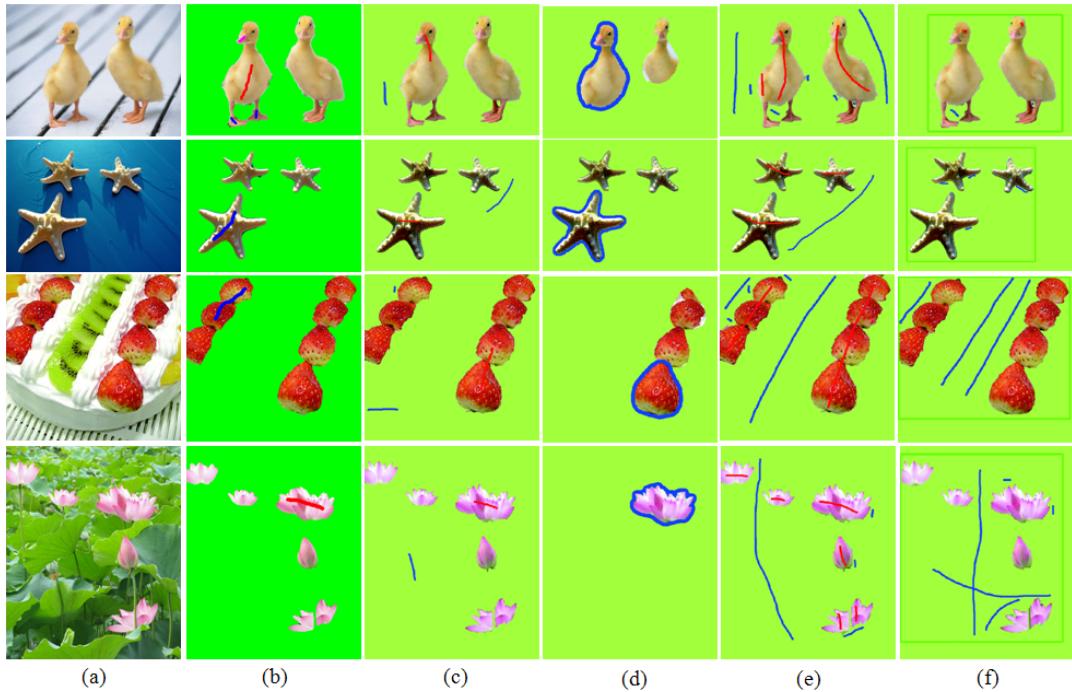


图 5-10 与现有交互式图割方法的比较

Fig 5-10 Comparison with existing image cutout methods

图5-10给出本章所提图像分割方法与现有几种图像分割方法的比较，包括重复场景元素抠取方法（RepSnapping [130]）、重复检测方法（RepFinder [131]）、交互式图像分割方法Lazy Snapping [106]和GrabCut [107]，与这些方法对应的后四列分割结果由文献 [130]提供。其中，Lazy Snapping和GrabCut两种方法是基于图割（graph cuts）的前景物体抠取方法。事实上，这些方法能够很好地处理对单个物体的抠取，而在处理多个重复场景元素的抠取时，则需要进行大量的前景和背景标记（如图5-10（e）和（f））。RepFinder方法分割的思路是根据形状特征的相似性来检测场景中的重复元素，因此，该方法不能很好地处理变形的物体。最近提出的RepSnapping方法则根据颜色的相似性，通过建立具有相同颜色像素点之间的连接关系改进图割方法，以此实现重复元素的并行抠取。然而，一方面该方法需要标记前景和背景；另一方面该方法不能很好地处理前景和背景颜色相似的物体。本章提出的方法的主要优势是仅需要在感兴趣的物体上进行标记，并且我们的方法不受物体颜色和形状的影响。但

是，场景中的物体在纹理模式上是容易辨别的。

在抠取出图像中的重复场景元素之后，我们可以采用替换、重排等方式进行图像编辑操作。图5-11给出这两种方式下的图像编辑结果，其中蛋糕图像中的桃心区域用草莓进行了替换；而水鸭的排列顺序可以进行相应调整。

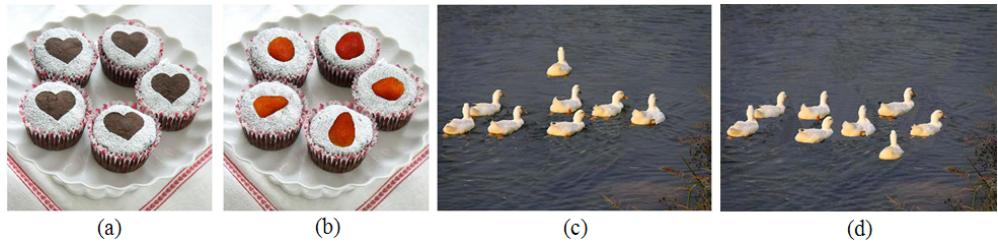


图 5-11 重复场景元素的编辑
Fig 5-11 Editing for repeated scene elements

5.6 本章小结

在本章方法中，我们提出了一种混合的灰度图像彩色化方法。一方面，我们借用基于样本彩色图像彩色化方法的颜色转移思路，自动生成待上色灰度图像的调色板，从而避免大量用户交互，即在图像中的区域上标记合适的颜色。另一方面，我们改进传统的水平集图像分割方法，通过事先检测出具有相似特征的区域，使得能够实现区域的并行分割，极大地加快了上色过程。然而，实现这两方面的关键是本章方法利用伽伯小波变换进行纹理特征分析。由此，灰度图像中各像素点能够在已分割的样本彩色图像中找到匹配的区域，从而能够准确地为这些像素点分配颜色信息。此外，可变的水平集方法能够在具有灰度连读或纹理连续的相似图像区域上进行分割，使得传播的曲线在区域边界处停止。此时，边界处具有最大梯度值或者纹理模式发生了突变。本章中给出的实验结果表明，这种混合的灰度图像彩色化方法能够获得令人满意的效果。

更重要的是，本章提出的可并行图像分割方法已经应用于进行图像中重复场景元素的抠取。由于并行可变水平集方法是一种局部的图像分割方法，通过结合基于颜色和纹理等特征的相似性检测，因此，我们仅需要标记一个感兴趣的前景目标物体即可同时抠取其余的相似物体。同时，该方法不再需要考虑背景对分割的影响。在本章提供的结果中，我们与多种现有图像分割方法进行了

比较，证明本章提出的方法是简单、有效的。此外，我们通过替换、重排等操作处理这些重复场景元素，从而对图像进行相应的编辑。

第六章 总结与展望

6.1 论文工作总结

本文对可视媒体编辑与重用的一些理论和应用问题进行了深入分析，在纹理合成与设计、流体动画合成以及灰度图像彩色化等方面进行了相关研究。针对现有工作的不足，本文提出了一些新的研究方法。概括起来，这些研究成果包括：

(1) 在基于块的纹理合成框架下，提出一种基于特征描述的新的二维纹理合成技术，特别用于合成具有显著结构的纹理。区别于基于特征的纹理合成方法，这些方法需要利用从样本纹理中抽取的特征进行纹理合成，本文提出的方法则通过引入层的概念和构建样本纹理的方向场来描述纹理特征的布局，使得能够自动地检测纹理特征的位置和方向，从而设计有效的纹理相似性的度量机制。然而，在不使用加速算法的情况下，由于穷尽的纹理子块的搜索，该方法是耗时的。此外，该方法没有提出一种合理的机制计算纹理子块的大小和边界区域的宽度，而这两个参数很大程度地影响纹理合成的效率。

(2) 由于现有基于样本的纹理合成方法不能很好地处理近似规则的纹理，在合成结果中不能保持样本纹理的结构规律性，因此，本文提出了一种基于基元分布的周期纹理合成与设计方法。该方法是一种图像和图形相结合的方法。一方面，通过检测样本纹理中基元（纹理元素）的位置，该方法构建基元间的空间相邻关系；另一方面，通过分析和重利用已抽取的样本纹理的拓扑结构，并以单个完整的基元为处理对象，使用该方法合成的纹理不仅能够恢复样本纹理中基元之间相邻接但不重叠的排列模式，而且还能保持纹理特征的连续性。此外，该方法通过定义基元的生成规则来获得多种多样的输出结果。然而，该方法的局限在于，当基元间的形状差异很大时，在合成的纹理中容易产生基元间过分重叠和大的空洞等瑕疵。

(3) 通过对给定的样本视频（视频纹理）进行编辑，本文提出了一种全局特征保持的流体动画合成方法，能够驱动单幅静止图像来获得连续的流体动画。该方法的主要贡献是：全局地传递样本视频中的流体运动特征给单幅图像，以及通过借助构建的流体视频库估计单幅静止图像的运动。全局的运动传

递方法能够使得在合成的流体动画中保持较大尺寸的流体特征；为单幅图像自动估计运动模式，这能够减轻大量且繁琐的用户交互。然而，由于传递的流体运动可能覆盖原单幅图像中的结构特征，由此，合成的流体动画很难保持图像的原始外观特征。此外，该方法需要由用户选定适合图像中似流体场景的样本视频，这使得合成流体动画的质量受所选样本视频的影响。

(4) 基于颜色转移的思路，本文提出的灰度图像彩色化方法通过传递参考样本彩色图像的颜色信息自动获得待上色灰度图像的调色板，使得尽可能地减轻用户输入。为了避免在图像边界处产生颜色混合或溢出等瑕疵，该方法通过采用基于纹理的可变水平集方法在亮度或纹理连续的区域内传播曲线，使得能够准确地找到区域的边界。该图像分割方法最大的特点是具有局部的、可并行的分割方式，由此，我们将该方法应用于抽取图像中重复的场景元素。此时，我们通过替换、重排等操作处理这些重复场景元素，从而编辑原始图像。

6.2 未来工作展望

基于本文的研究工作，我们将对其进行进一步的完善和改进，未来工作的主要方向如下：

(1) 针对现有基于样图的纹理合成方法中存在的问题，虽然本文提出了相应的解决方案，能够产生高质量的合成结果并极大地改善了合成的速度，但是本文中提出的两种纹理合成算法仍然停留在处理二维纹理图像。在未来的工作中，我们将致力于三维几何纹理的合成与设计方法的研究。我们主要研究：如何分析并利用几何纹理元素之间的空间拓扑结构；在不借助Maya、3Dmax等建模软件的情况下，如何快速地构建高精度的几何纹理元素模型；如何处理在堆积几何纹理元素时可能出现的遮挡、变形和重叠等问题。

(2) 在本文提出的单幅图像的流体动画合成方法中，流体动画合成的质量依赖于我们构建的流体视频库。由此，我们今后的部分工作是构建一个包含多种流体类型且场景丰富的大型视频库。此外，我们还将完善已提出方法中存在诸多问题：提出一种样本视频选择机制，使得能够根据提供的目标图像在视频库中自动选择合适的样本流体视频；提出一种能够保持目标图像原始外观的流体传递方法，即在传递流体运动的同时能够保留目标图像中的结构特征。更重要的是，提出的方法在合成流体动画时仍需要耗费大量的时间，因此，如何实时合成流体动画仍是一个挑战。

(3) 本文提出的灰度图像彩色化方法仅对图像进行上色。明显地，我们未来的研究工作是如何将现有方法扩展到视频上的彩色化。由于本文提出的方法需要事先获得调色板信息，因此，如何根据给定的一幅或多幅样本彩色图像获得一段视频对应的调色板是我们面临的一个巨大挑战。此外，在使用图割方法进行颜色传递的过程中，如何考虑视频的帧间关联性，即利用时空一致性保持颜色在视频帧间的传递是平滑、自然的。同时，我们也将进一步完善可并行图像分割算法，使得可变水平集图像分割算法更加稳定且能处理多纹理混合的前景物体抠取。

参考文献

- [1] EBERT D, MUSGRAVE K, PEACHEY D, et al. Texturing and Modeling: A Procedural Approach[M]. USA: Academic Press.
- [2] PERLIN K. An image synthesizer[C]//SIGGRAPH. San Francisco, California: ACM, 1985:287–296.
- [3] PERLIN K. Improving noise[J]. ACM Transactions on Graphics, 2002, 21(3):681–682.
- [4] COOK R L, DEROSSE T. Wavelet noise[J]. ACM Transactions on Graphics, 2005, 24(3):803–811.
- [5] WITKIN A, KASS M. Reaction-diffusion textures[C]//Computer Graphics (SIGGRAPH '91 Proceedings). Las Vegas, Nevada: ACM, 1991, 25:299–308.
- [6] TURK G. Generating textures for arbitrary surfaces using reaction-diffusion[C]//Computer Graphics (SIGGRAPH '91 Proceedings). Las Vegas, Nevada: ACM, 1991, 25:289–298.
- [7] WALTER M, FOURNIER A, REIMERS M. Clonal Mosaic Model for the Synthesis of Mammalian Coat Patterns[C]//Proceedings of Graphics Interface. Vancouver, BC, Canada: Canadian Human-Computer Communications Society, 1998:82–91.
- [8] WALTER M, BRAZIL U, FOURNIER A, et al. Integrating Shape and Pattern in Mammalian Models[C]//Proceedings of ACM SIGGRAPH 2001. Los Angeles, California, USA: ACM, 2001:317–326.
- [9] FLEISCHER K W, LAIDLAW D H, CURRIN B L, et al. Cellular Texture Generation[C]//Proceedings of ACM SIGGRAPH 1995. Los Angeles, California: ACM, 1995:239–248.

- [10] WORLEY S P. A Cellular Texturing Basis Function[C]//SIGGRAPH 96 Conference Proceedings. New Orleans, Louisiana: ACM, 1996:291–294.
- [11] MIYATA K. A Method of Generating Stone Wall Patterns[C]//Computer Graphics (SIGGRAPH '90 Proceedings), 24(4). Dallas, Texas: ACM, 1990:387–394.
- [12] DORSEY J, EDELMAN A, JENSEN H W, et al. Modeling and Rendering of Weathered Stone[C]//Proceedings of ACM SIGGRAPH 1999. Los Angeles, California: ACM, 1999:225–234.
- [13] WEI L Y, LEFEBVRE S, KWATRA V, et al. State of the Art in Example-based Texture Synthesis[C]//Eurographics 2009, State of the Art Report, EG-STAR. Murich Germany: Eurographics Association, 2009.
- [14] HEEGER D J, BERGEN J R. Pyramid-Based Texture Analysis/Synthesis[C]//Proceedings of ACM SIGGRAPH 1995. Los Angeles, California: ACM, 1995:229–238.
- [15] BONET J S D. Multiresolution Sampling Procedure for Analysis and Synthesis of Texture Images[C]//Proceedings of ACM SIGGRAPH 1998. Orlando, Florida: ACM, 1998:361–368.
- [16] SIMONCELLI E P, PORTILLA J. Texture Characterization via Joint Statistics of Wavelet Coefficient Magnitudes[C]//Proceedings of ACM SIGGRAPH 1998. Orlando, Florida: ACM, 1998:4–7.
- [17] PORTILLA J, SIMONCELLI E P. Texture Modeling and Synthesis using Joint Statistics of Complex Wavelet Coefficients[C]//IN IEEE WORKSHOP ON STATISTICAL AND COMPUTATIONAL THEORIES OF VISION. Fort Collins, Colorado: Kluwer Academic Publishers, 1999:22–30.
- [18] PAGET R. Strong Markov Random Field Model[J]. IEEE Trans. Pattern Anal. Mach. Intell, 2004, 26(3).
- [19] EFROS A, LEUNG T. Texture Synthesis by Non-parametric Sampling[C]//In International Conference on Computer Vision. Kerkyra, Corfu, Greece: IEEE Computer Society, 1999:1033–1038.

- [20] YI WEI L, LEVOY M. Fast Texture Synthesis using Tree-structured Vector Quantization[C]//SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques. New Orleans, Louisiana: ACM, 2000:479–488.
- [21] ASHIKHMİN M. Synthesizing Natural Textures[C]//In ACM Symposium on Interactive 3D Graphics. Chapel Hill, NC, USA: ACM, 2001:217–226.
- [22] ZELINKA S, GARLAND M. Towards real-time texture synthesis with the jump map[C]//Proceedings of the 13th Eurographics workshop on Rendering. Switzerland, Switzerland: Eurographics Association Aire-la-Ville, 2002:99–104.
- [23] TONG X, ZHANG J, LIU L, et al. Synthesis of bidirectional texture functions on arbitrary surfaces[C]//Proceedings of ACM SIGGRAPH 2002. San Antonio, Texas, USA: ACM, 2002:665–672.
- [24] HERTZMANN A, JACOBS C E, OLIVER N, et al. Image Analogies[C]//Proceedings of the 28th annual conference on Computer graphics and interactive techniques. Los Angeles, CA, USA: ACM, 2001:327–340.
- [25] LEFEBVRE S, HOPPE H. Parallel controllable texture synthesis[J]. ACM Trans. Graph., 2005, 24(3):777–786.
- [26] YING-QING XU H S, BAINING GUO. Chaos mosaic: Fast and memory efficient texture synthesis[R].[S.l.]: Microsoft, 2000.
- [27] PRAUN E, FINKELSTEIN A. Lapped textures[C]//Proceedings of SIGGRAPH 2000. New Orleans, Louisiana: ACM, 2000:465–470.
- [28] EFROS A, FREEMAN W. Image Quilting for Texture Synthesis and Transfer[C]//In SIGGRAPH'01. Los Angeles, California, USA: ACM, 2001:341–346.
- [29] LIANG L, LIU C, XU Y, et al. Real-time texture synthesis by patch-based sampling[J]. ACM Transactions on Graphics, 2001, 20(3):127–150.

- [30] SZELISKI R, SHUM H Y. Creating Full View Panoramic Image Mosaics and Environment Maps[C]//In SIGGRAPH'97. Los Angeles, California: ACM, 1997:251–258.
- [31] ARYA S, MOUNT D M, NETANYAHU N S, et al. An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions[C]//ACM-SIAM SYMPOSIUM ON DISCRETE ALGORITHMS. Arlington, Virginia: ACM/SIAM, 1994, 45:573–582.
- [32] MOUNT D M, MOUNT D M. ANN Programming Manual[R]. Maryland: Department of Computer Science, University of Maryland, College Park, Maryland, 1998.
- [33] JOLLIFFE I. Principal Component Analysis[M]. USA: Springer Verlag, 1986.
- [34] KWATRA V, SCHÖDL A, ESSA I, et al. Graphcut textures: image and video synthesis using graph cuts[J]. ACM Transactions on Graphics, 2003, 22(3):277–286.
- [35] FORD L R, FULKERSON D R. Flows in Networks[M]. USA: Princeton University Press, 1962.
- [36] SEDGEWICK R. Algorithms in C, Part 5: Graph Algorithms[M]. USA: Addison-Wesley, Reading, Massachusetts, 2001.
- [37] KWATRA V, ESSA I, BOBICK A, et al. Texture optimization for example-based synthesis[J]. ACM Trans. Graph., 2005, 24(3):795–802.
- [38] JARKKO K. A small aperiodic set of Wang tiles[J]. Discrete Mathematics, 1996, 160(1-3):259–264.
- [39] A. G. Aperiodic tiling, part 1[M]. USA: IEEE Computer Graphics and Applications 18, 4 (May), 1998.
- [40] KILTHAU S L, DREW M S, MOLLER T. Full Search Content Independent Block Matching Based on the Fast Fourier Transform[C]//In ICIP02, I:

- 669 – 672. Rochester, New York: IEEE Conference Publications, 2002:669–672.
- [41] SOLER C, CANI M P, ANGELIDIS A. Hierarchical pattern mapping[J]. ACM Transactions on Graphics, 2002, 21(3):673–680.
- [42] GRUNBAUM B, SHEPHARD G C. Tilings and patterns[M]. USA: W. H. Freeman and Company, 1987.
- [43] CULIK K, II. An aperiodic set of 13 Wang tiles[J]. Discrete Mathematics, 1996, 160:245–251.
- [44] WANG H. Proving theorems by pattern recognition II[J]. Bell Systems Technical Journal, 1961, 40:1–42.
- [45] WANG H. Games, logic, and computers[J]. Scientific American, 1965:98–106.
- [46] STAM J. Aperiodic Texture Mapping[R].[S.l.]: European Research Consortium for Informatics and Mathematics (ERCIM), 1997.
- [47] NEYRET F, PAULE CANI M. Pattern-Based Texturing Revisited[C]//Proceedings of SIGGRAPH 99. Los Angeles, California: ACM, 1999:235–242.
- [48] COHEN M F, SHADE J, HILLER S, et al. Wang Tiles for image and texture generation[C]//SIGGRAPH '03: ACM SIGGRAPH 2003 Papers. San Diego, California, USA: ACM, 2003:287–294.
- [49] LAGAE A, DUTRÉ P. An Alternative for Wang Tiles: Colored Edges versus Colored Corners[J]. ACM Transactions on Graphics, 2006, 25(4):1442–1459.
- [50] ZHANG J, UNIV T, ZHOU K, et al. Synthesis of progressively-variant textures on arbitrary surfaces[C]//InSIGGRAPH '03. San Diego, California, USA: ACM, 2003:295–302.
- [51] QING W, YIZHOU Y. Feature matching and deformation for texture synthesis[J]. ACM Trans. Graph., 2004, 23(3):364–367.

- [52] LEFEBVRE S, HOPPE H. Appearance-space texture synthesis[C]//SIGGRAPH '06: ACM SIGGRAPH 2006 Papers. Boston, MA, USA: ACM, 2006:541–548.
- [53] DISCHLER J M, ZARA F. Real-time structured texture synthesis and editing using image-mesh analogies[J]. Vis. Comput., 2006, 22(9):926–935.
- [54] LIU Y, LIN W C, HAYS J. Near-regular texture analysis and manipulation[C]//SIGGRAPH '04: ACM SIGGRAPH 2004 Papers. Los Angeles, California, USA: ACM, 2004:368–376.
- [55] LIU Y, TSIN Y. The Promise and The Perils of Near-regular Texture[J]. International Journal of Computer Vision, 2002, 62:1–2.
- [56] M. DISCHLER J, MARITAUD K, LÉVY B, et al. Texture Particles[J]. Computer Graphics Forum, 2002, 21:401–410.
- [57] BARLA P, BRESLAV S, THOLLOT J, et al. Stroke Pattern Analysis and Synthesis[C]//Computer Graphics Forum (Proc. of Eurographics 2006). Vienna, Austria: ACM, 2006, 25:663–671.
- [58] IJIRI T, MECH R, IGARASHI T, et al. An Example-based Procedural System for Element Arrangement[J]. Computer Graphics Forum, 2008, 27(2):429–436.
- [59] SZUMMER M, PICARD R W. Temporal Texture Modeling[C]//In IEEE International Conference on Image Processing. Lausanne, Switzerland: IEEE Conference Publications, 1996:823–826.
- [60] SZUMMER M. Temporal Texture Modeling[D]. USA: Massachusetts Institute of Technology, 1995.
- [61] BAR-JOSEPH Z, EL-YANIV R, LISCHINSKI D, et al. Texture Mixing and Texture Movie Synthesis using Statistical Learning[J]. IEEE Transactions on Visualization and Computer Graphics, 2001, 7:120–135.
- [62] DORETTO G, CHIUSO A, WU Y, et al. Dynamic textures.[J]. 2003, 51(2):91–109.

- [63] YUAN L, YEUNG SHUM H. Synthesizing dynamic texture with closed-loop linear dynamic system[C]//In Proc. European Conference on Computer Vision. Prague: Springer-Verlag, 2004:603–616.
- [64] SUNG LIN R, BIN LIU C, HSUAN YANG M, et al. Learning Nonlinear Manifolds from Time Series[C]//In Proc. ECCV. Graz, Austria: Springer-Verlag, 2006:239–250.
- [65] LIU C B, LIN R S, AHUJA N. Modeling Dynamic Textures Using Subspace Mixtures[C]//Proceedings of the 2005 IEEE International Conference on Multimedia and Expo, July 6-9. Amsterdam, The Netherlands: IEEE, 2005:1378–1381.
- [66] LIU C B, LIN R S, AHUJA N, et al. Dynamic Textures Synthesis as Non-linear Manifold Learning and Traversing[C]//The 17th British Machine Vision Conference. Edinburgh: BMVA Press, 2006:859–868.
- [67] WANG Y, CHUN ZHU S. Modeling textured motion: Particle, wave and sketch[C]//IEEE Int. Conf. on Computer Vision, ICCV’ 03. Nice, France: IEEE Computer Society 2003, 2003:213–220.
- [68] WANG Y, CHUN ZHU S. Analysis and Synthesis of Textured Motion: Particles and Waves[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26:1348–1363.
- [69] SCHODL A, SZELISKI R, SALESIN D, et al. Video textures.[C]//In Proc. of SIGGRAPH 2000. San Diego, California, USA: ACM, 2000:489–498.
- [70] SCHODL A, ESSA I A. Controlled animation of video sprites[C]//Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation. San Antonio, Texas, USA: ACM, 2002:121–127.
- [71] SHINYA M, AOKI M, TSUTSUGUCHI K, et al. Dynamic texture: physically based 2D animation.[C]//In Proc. ACM SIGGRAPH 1999. Los Angeles, California: ACM, 1999:239.

- [72] FOSTER N, METAXAS D. Practical animation of liquids[C]//In Proc. of SIGGRAPH 2001. Los Angeles, California, USA: ACM, 2001:23–30.
- [73] TREUILLE A, McNAMARA A, POPOVC C, et al. Keyframe Control of Smoke Simulations.[J]. 2003, 22(3):716–723.
- [74] SUN M, JEPSON A D, FIUME E. Video input driven animation (VIDA[C]//In International Conference on Computer Vision (ICCV. Nice, France: IEEE Computer Society 2003, 2003:96–103.
- [75] BHAT K S, SEITZ S M, HODGINS J K, et al. Flow-based Video Synthesis and Editing[J]. ACM Transactions on Graphics (SIGGRAPH 2004), 2004, 23(3):360–363.
- [76] LIN Z, WANG L, WANG Y, et al. High Resolution Animated Scenes from Stills[J]. IEEE Trans. Vis. Comput. Graph., 2007, 13(3):562–568.
- [77] CHUANG Y Y, GOLDMAN D B, CURLESS B, et al. Animating Pictures with Stochastic Motion Textures[C]//In Proc. of SIGGRAPH 2005. Los Angeles, California, USA: ACM, 2005:853–860.
- [78] CHUANG Y Y, CURLESS B, SALESIN D H, et al. A Bayesian Approach to Digital Matting[C]//Proceedings of IEEE CVPR 2001. Kauai, HI, USA: IEEE Computer Society, 2001, 2:264–271.
- [79] OKABE M, ICHI ANJKO K, IGARASHI T, et al. Animating Pictures of Fluid using Video Examples.[J]. Comput. Graph. Forum, 2009, 28(2):677–686.
- [80] OKABE M, ANJKO K, ONAI R. Creating Fluid Animation from a Single Image using Video Database[J]. Computer Graphics Forum, 2011, 30:1973–1982.
- [81] BURNS G. Colorization[C]//Museum of Broadcast Communications: Encyclopedia of Television. .[S.l.]: [s.n.] .
- [82] DANI A L, LISCHINSKI D, WEISS Y. Colorization using Optimization[J]. ACM Transactions on Graphics, 2004, 23:689–694.

- [83] YI-CHIN HUANG J C C S W W, YI-SHIN TUNG, WU J L. An adaptive edge detection based colorization algorithm and its applications[C]//ACM Multimedia. Hilton,Singapore: Addison-Wesley Longman Publishing Co., Inc., 2005:351–354.
- [84] YATZIV L, SAPIRO G. Fast image and video colorization using chrominance blending[J]. IEEE Transactions on Image Processing, 2006, 15(5):1120–1129.
- [85] QU Y, WONG T T, HENG P A. Manga Colorization[J]. ACM Transactions on Graphics (SIGGRAPH 2006 issue), 2006, 25(3):1214–1220.
- [86] LUAN Q, WEN F, COHEN-OR D, et al. Natural Image Colorization[C]//Rendering Techniques. Grenoble, France: Eurographics Association, 2007:309–320.
- [87] TOMIHISA WELSH M A, MUELLER K. Transferring color to greyscale images[J]. ACM Trans. Graph., 2002, 21(3):277–280.
- [88] ERIK REINHARD B G, MICHAEL ASHIKMIN, SHIRLEY P. Color Transfer between Images[J]. IEEE Computer Graphics and Applications, 2001, 21(5):34–41.
- [89] CHEN T, WANG Y, SCHILLINGS V, et al. Grayscale Image Matting And Colorization[C]//IN PROCEEDINGS OF ASIAN CONFERENCE ON COMPUTER VISION (ACCV 2004. Jeju, Korea: Elsevier Science Inc., 2004:1164–1169.
- [90] SYKORA D, BURIANEK J, ZARA J. Unsupervised colorization of black-and-white cartoons[C]//In Proc. NPAR (2004). Annecy, France: ACM 2004, 2004:121–127.
- [91] SÝKORA D, BURIÁNEK J, ŽÁRA J. Colorization of Black-and-White Cartoons[J]. Image and Vision Computing, 2005, 23(9):767–782.
- [92] IRONY R, COHEN-OR D, LISCHINSKI D. Colorization by Example[C]//Proceedings of Eurographics Symposium on Rendering 2005

- (EGSR'05, June 29–July 1, 2005, Konstanz, Germany). Konstanz,Germany: ACM siggraph, 2005:201–210.
- [93] XIAOPEI LIU Y Q T T W S L C S L, LIANG WAN, HENG P A. Intrinsic Colorization[J]. ACM Transactions on Graphics (SIGGRAPH Asia 2008 issue), 2008, 27(5):152:1–152:9.
- [94] CHIA A Y S, ZHUO S, GUPTA R K, et al. Semantic colorization with internet images[J]. ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia), 2011, 30(6).
- [95] TUCERYAN M, JAIN A K. The Handbook of Pattern Recognition and Computer Vision : Texture Analysis[M].[S.1.]: World Scientific Publishing Co., 1998.
- [96] 莫灿林, 陈 敏, 郭绍义. 纹理合成的自相关性判别法及其应用[J]. 计算机研究与发展, 2007, 45(2):337–341.
- [97] RAO A R, SCHUNCK B G. Computing oriented texture fields[J]. CVGIP: Graphical Model and Image Processing, 1991, 53(2):157–185.
- [98] RAO A R, JAIN R. Computerized Flow Field Analysis: Oriented Texture Fields[J]. IEEE Trans. Pattern Anal. Mach. Intell., 1992, 14(7):693–709.
- [99] DI ZENZO S. A Note on the Gradient of a Multi-Image[J]. Computer Vision, Graphics, and Image Processing, 1986, 33(1):116–125.
- [100] SOLER C, CANI M P, ANGELIDIS A. Hierarchical pattern mapping[J]. ACM Trans. Graph., 2002, 21(3):673–680.
- [101] KILTHAU S L, DREW M S, MOLLER T. Full Search Content Independent Block Matching Based on the Fast Fourier Transform[C]//In ICIP02, I: 669 – 672. Rochester, New York: IEEE, 2002:669–672.
- [102] JIA J, TANG C K. Eliminating Structure and Intensity Misalignment in Image Stitching[C]//ICCV. Beijing, China: IEEE Computer Society, 2005:1651–1658.
- [103] JIA J, TANG C K. Image Stitching Using Structure Deformation[J]. IEEE Trans. Pattern Anal. Mach. Intell., 2008, 30(4):617–631.

- [104] ORCHARD M, BOUMAN C. Color Quantization of Images[J]. IEEE Transactions on Signal Processing, 1991, 39:2677–2690.
- [105] MORTENSEN E N, BARRETT W A. Intelligent scissors for image composition[C]//SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques. Los Angeles, California: ACM, 1995:191–198.
- [106] ROTHER C, KOLMOGOROV V, BLAKE A. "GrabCut": interactive foreground extraction using iterated graph cuts[J]. ACM Trans. Graph., 2004, 23(3):309–314.
- [107] LI Y, SUN J, TANG C K, et al. Lazy snapping[J]. ACM Trans. Graph., 2004, 23(3):303–308.
- [108] BOYKOV Y, JOLLY M P. Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images[C]//ICCV. Vancouver, Canada: IEEE Computer Society, 2001:105–112.
- [109] BOYKOV Y, JOLLY M P. Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images[C]//ICCV. Vancouver, Canada: IEEE Computer Society, 2001:105–112.
- [110] LEUNG T K, MALIK J. Detecting, localizing and grouping repeated scene elements from an image[C]//ECCV '96: Proceedings of the 4th European Conference on Computer Vision-Volume I. Cambridge, UK: Jean Ponce, Andrew Zisserman, Martial Hebert (Eds.), 1996:546–555.
- [111] BELONGIE S, MALIK J, PUZICHA J. Shape Matching and Object Recognition Using Shape Contexts[J]. IEEE Trans. Pattern Anal. Mach. Intell., 2002, 24(4):509–522.
- [112] MORI G, BELONGIE S, MALIK J, et al. Efficient shape matching using shape contexts[J]. IEEE Trans. Pattern Analysis and Machine Intelligence, 2005, 27:1832–1837.

- [113] PÉREZ P, GANGNET M, BLAKE A. Poisson image editing[J]. ACM Trans. Graph., 2003, 22(3):313–318.
- [114] CRIMINISI A, PÉREZ P, TOYAMA K. Region filling and object removal by exemplar-based image inpainting[J]. IEEE Transactions on Image Processing, 2004, 13:1200–1212.
- [115] BOOKSTEIN F L. Principal Warps: Thin-Plate Splines and the Decomposition of Deformations[J]. IEEE Trans. Pattern Anal. Mach. Intell., 1989, 11(6):567 – 585.
- [116] BROX T, MALIK J. Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation[J]. 2011, 33(3):500–513.
- [117] LIU C, YUEN J, TORRALBA A, et al. SIFT Flow: Dense Correspondence across Different Scenes[C]//In Proceeding ECCV '08 Proceedings of the 10th European Conference on Computer Vision. Marseille, France: Springer, 2008:28–42.
- [118] RUBINSTEIN M, SHAMIR A, AVIDAN S. Improved seam carving for video retargeting.[J]. 2008, 27(3):1–9.
- [119] HEEGER D, BERGEN J. Pyramid-based texture analysis/synthesis[C]//In Proc. of SIGGRAPH 1995. Los Angeles, California: Kluwer Academic Publishers, 1995.
- [120] LUCAS B D, KANADE T. An Iterative Image Registration Technique with an Application to Stereo Vision (IJCAI)[C]//Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81). Vancouver, British Columbia, Canada: William Kaufmann 1981, 1981:674–679.
- [121] LEVIN A, LISCHINSKI D, WEISS Y. A Closed-Form Solution to Natural Image Matting.[J]. 2008, 30(2):228–242.
- [122] LOWE D. Distinctive image features from scale-invariant keypoints.[J]. 2004, 60(2):91–110.

- [123] BROSTOW G, ESSA I. Image-based motion blur for stop motion animation[C]//In Proc. of SIGGRAPH 2001. Los Angeles, California, USA: ACM, 2001:561–566.
- [124] MANJUNATH B S, MA W Y. Texture Features for Browsing and Retrieval of Image Data[J]. IEEE Trans. Pattern Anal. Mach. Intell., 1996, 18(8):837–842.
- [125] OSHER S, SETHIAN J A. Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations[J]. JOURNAL OF COMPUTATIONAL PHYSICS, 1988, 79(1):12–49.
- [126] V. CASELLES T C, F. CATTE, DIBOS F. A geometric model for active contours in image processing[J]. Numer. Math., 1993, 66(1):1–31.
- [127] MALLADI R, SETHIAN J A, VEMURI B C. Shape modeling with front propagation: A level set approach[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, 17:158–175.
- [128] SETHIAN J. Level Set Methods and Fast Marching Methods[M].[S.l.]: Cambridge, U.K.: Cambridge Univ. Press, 1999.
- [129] LI C, XU C, GUI C, et al. Level Set Evolution Without Re-initialization: A New Variational Formulation[C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). San Diego, CA, USA.: IEEE Computer Society 2005, 2005, 1:430–436.
- [130] HUANG H, ZHANG L, ZHANG H C. RepSnapping: efficient image cutout for repeated scene elements[J]. Computer Graphics Forum, 2011, 30:2059–2066.
- [131] CHENG M M, ZHANG F L, MITRA N J, et al. RepFinder: Finding Approximately Repeated Scene Elements for Image Editing[J]. ACM Transactions on Graphics, 2010, 29(4):83:1–8.

致 谢

本论文的研究和撰写工作是在我的导师马利庄教授的悉心指导下完成的，在论文即将完成之际，我谨向马老师致敬！诚挚地感谢他这四年的指导、教诲与关怀。他宽阔的视野、深厚的学术功底、严谨的治学态度、开阔活跃的思维、勇于开拓的精神深刻地影响了我，使我受益匪浅。同时，感谢盛斌老师给予我的许多帮助和指导。

非常感谢我的爱人王子卿，我能顺利完成学业，离不开他一直以来默默的支持与鼓励。从硕士阶段到博士阶段，这些年的风风雨雨、艰辛历程中，他与我同甘共苦、共同拼搏，没有他的理解与支持，也就没有我的今天，由衷感谢他。感谢我的爸爸和妈妈，在外面求学的这些年，感谢他们对我的关爱和理解，我的成长凝聚着他们的心血。感谢我的公公和婆婆等亲人，他们的关怀让我感到很温暖。祝福他们身体健康，事事顺意！

衷心感谢实验室与我同一项目组的师弟师妹们：尹超、钟珍、夏爱娟、李勇、眭博聪，他们与我共同的努力使有关的研究得以顺利进行。在此，特别感谢肖学中、褚娜、李灿林、陈敏刚、沈洋、谢志峰、林晓、刘先勇、丁守鸿等几位博士给予我的帮助。同时，也要感谢实验室其他所有师兄师弟和师姐师妹们，他们给了我不同方面的帮助，在此恕不一一列出他们的名字。

最后，感谢担任我论文评审工作的各位专家、教授，谢谢你们提出的宝贵意见和建议。

攻读学位论文期间发表的学术论文目录

- [1] **Yan Gui**, Lizhuang Ma, “Periodic Pattern of Texture Analysis and Synthesis based on Texels Distribution”, *The Visual Computer*, 2010, 26(6-8), 951-964. (SCI, UT ISI: 000278135800055)
- [2] **Yan Gui**, Mingang Chen, Lizhuang Ma, Zhihua Chen, “Texel Based Regular and Near-Regular Texture Characterization”, International Conference on Multimedia and Signal Processing, 2011, 1, 266-270. (EI, Accession number: 20113414259038)
- [3] **Yan Gui**, Mingang Chen, Zhifeng Xie, Lizhuang Ma, Zhihua Chen, “Texture Synthesis based on Feature Description”, *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, 2012, 6(3), 376-388. (SCI)
- [4] **Yan Gui**, Lizhuang Ma, Chao Yin Zhihua Chen, “Preserving Global Features of Fluid Animation from a Single Image using Video Examples”, *Journal of Zhejiang University-SCIENCE C (Computers & Electronics)*, accepted. (SCIE)
- [5] **Yan Gui**, Lizhuang Ma ‘Freehand Sketch-based Texture Design and Synthesis using Texel combinations’, Submitted to IET image processing. (SCI)
- [6] **Yan Gui**, Bin Sheng, Lizhuang Ma, “Parallel Image Cutout for Repeated Scene Elements using Variational Level Set”, Submitted to IEEE Transactions on Visualization and Computer Graphics (TVCG). (SCI)
- [7] Aijuan Xia, **Yan Gui**, Li Yao, Lizhuang Ma, Xiao Lin, “Exemplar-Based Object Removal in Video Using GMM”, International Conference on Multimedia and Signal Processing, 2011, 1, 366-370. (EI, Accession number: 20113414258982)
- [8] Chao Yin, **Yan Gui**, Zhifeng Xie, Lizhuang Ma, “Shape Context Based Video Texture Synthesis from Still Images”, International conference on computational and information sciences, 2011. (EI, Accession number: 20115114621578)

- [9] Zhen Zhong, **Yan Gui**, Lizhuang Ma, “An Automatic Image and Video Colorization Algorithm based on Pattern Continuity”, International Conference on Computational Intelligence and Software Engineering, 2011.
- [10] Zhifeng Xie, Rynson W.H. Lau, **Yan Gui**, Mingang Chen, Lizhuang Ma, “A-Gradient-domain-based Edge-preserving Sharpen Filter”, The Visual Computer, 2011, Published Online: 19 January 2012. (SCI)
- [11] Yong Li, Zhifeng Xie, Bin Sheng, **Yan Gui**, Lizhuang Ma, “Seamless Image Composition with Coherent Tone Adjustment”, The 10th International Conference on Virtual Reality Continuum and Its Applications in Industry, 2011. (EI, Accession number: 20120614739862)
- [12] 桂彦, 马利庄, “基于基元重组的纹理设计与合成方法 (申请号: 201110350537.7) ”, 2011. (专利)

上海交通大学
学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名： 桂彦

日期：2012 年 9 月 23 日

上海交通大学
学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海交通大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保 密 ，在 _____ 年解密后适用本授权书。

本学位论文属于

不保密

(请在以上方框内打“√”)

学位论文作者签名: 桂彦

指导教师签名: 孙立

日期: 2012 年 9 月 23 日

日期: 2012 年 9 月 23 日