



中山大學
SUN YAT-SEN UNIVERSITY

数字图像选做作业二

多方法图像填充对比介绍

姓名 杨佳鸣

学号 21304175

学院 数学学院

专业 统计学

2024 年 6 月 15 日

目录

1	总体介绍	1
2	方法介绍	2
2.1	Texture Synthesis by Non-parametric Sampling	2
2.1.1	核心细节步骤	2
2.1.2	方法点评	3
2.2	Image Completion using Planar Structure Guidance	3
2.2.1	图像补全步骤	3
2.2.2	损失函数	4
2.2.3	方法点评	4
2.3	其他方法	4
3	结果展示	5
3.1	砖块图	5
3.2	指纹图	5
3.3	简笔画	6
3.4	水果图	7
4	总结	8
A	代码	8

1 总体介绍

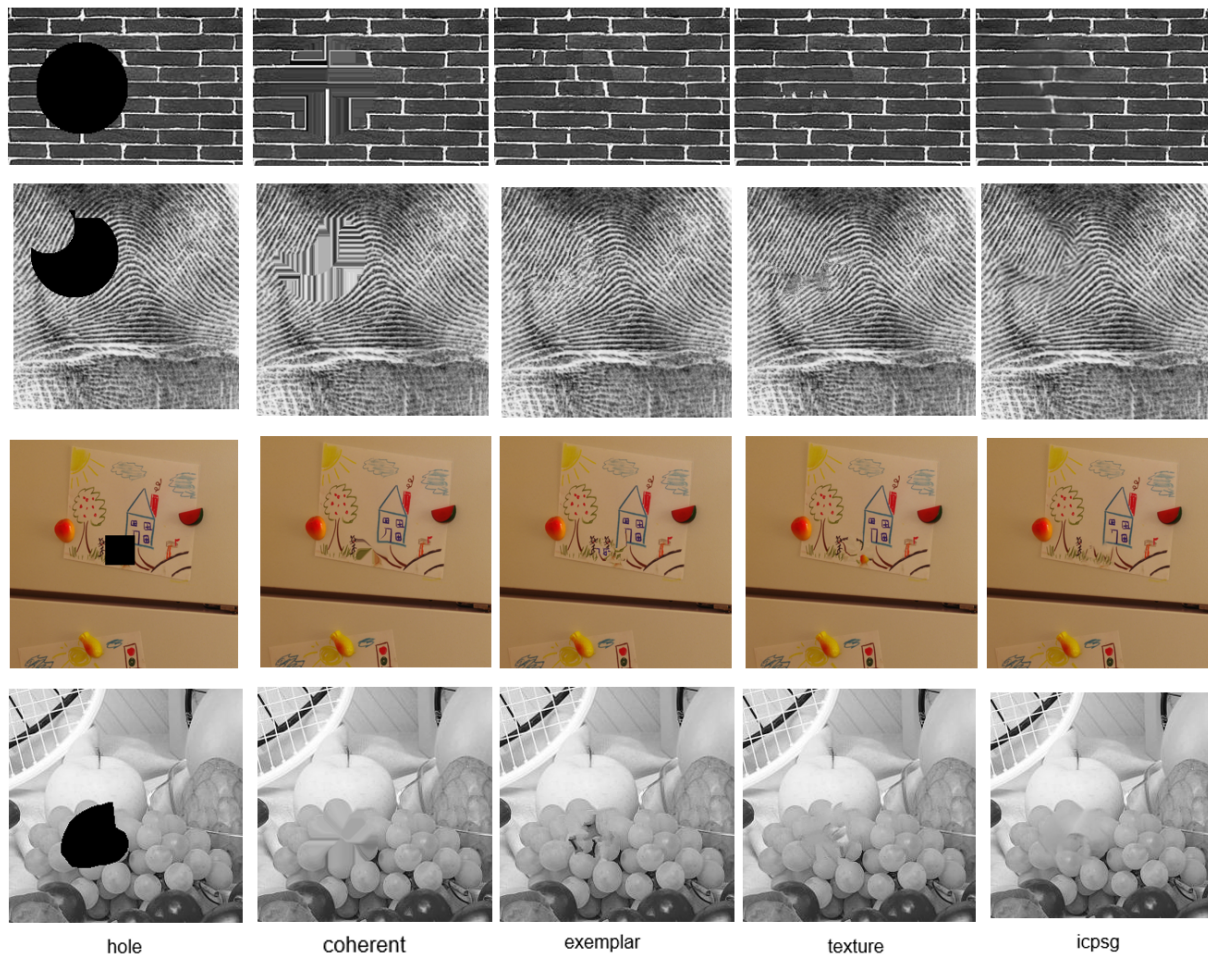


图 1: 效果总览

面对本次作业四张风格迥异的图像，我有一些不好的预感，相信一种方法并不能完美解决所有问题，故本次报告除了使用两种 MATLAB 中自带函数，还另外介绍了两种方法：“Texture Synthesis by Non-parametric Sampling”和“Image Completion using Planar Structure Guidance”。本次报告的方法全部是基于数学的方法。这篇报告将深入探讨这些方法在处理各种图像上的表现，一一对比它们的优点和不足。这是一个探索性的过程，我们希望通过这个过程，能够更深入地理解这些方法，掌握它们的运作原理，以及它们如何影响图像的处理结果。

2 方法介绍

2.1 Texture Synthesis by Non-parametric Sampling

该方法主要参考了 1999 年的一篇论文 [1]，初心是为了填充砖块图和指纹图，但具体代码实现与论文描述略有不同。方法的核心假设是把图像看作一个马尔可夫随机场 (MRF)，且假设整个图像是平稳的，也就是说，对于图像 (I) 中的一个像素 p ，当我们给定它周围的像素 $\omega(p)$ 时， $p \perp I \setminus \omega(p)$ 。粗略描述填充步骤如下：

1. 对于一个待填充的 pixel，用模板匹配的方式，搜索邻域与其邻域最接近的 N 个匹配点
2. 所有相似的邻域构成了一个无参数的概率分布函数 $P(p|\omega(p))$
3. 采用一个距离 d 来衡量待填充 pixel 的邻域与源邻域之间的相似度，用于挑选最相似的一些邻域。

这是一个粗略的思想介绍，下面我们介绍一下细节：

2.1.1 核心细节步骤

对于前面所讲的步骤，有两个问题要解决：

- 我们要填充的 pixel 是块状缺失的，对于缺失区域内部的点，其并无邻域信息，而对边界点，只有部分信息。
- 距离 d 如何定义的问题。

下面我们来解决上述问题：

对于第一个问题，我们的策略是**逐步扩张**。即从待填充的边缘像素开始，逐步往内填充。对于待填充块的边界（寻找边界时用了**形态学中的膨胀方法**），只有部分像素点是已知的，我们将其记为 $\omega'(p)$ ，故我们在寻找匹配块的时候，只用这部分信息去计算距离 d ，当边缘被填好时，已填好的边缘又可作为相邻内部未填好的 pixel 邻域信息。

对于第二个问题，我们首先想到的肯定是均方误差，但这样的度量会使得无论是像素中心 p ，还是窗口边缘，误差权重都是一样的，而我们希望要保持纹理的局部结构，故在均方误差的基础上，我们对误差进行卷积操作。故度量 d 可以表达为下式：

$$d(A, B) = \|\text{mask} \circ (A - B)^2 * G\|_2 \quad (1)$$

其中 A, B 为两个模板， mask 是遮住待填充 pixel 邻域中缺失的部分， G 为高斯核， \circ 是哈达玛乘积， $*$ 是卷积。

于是我们选择的最优匹配可以表示为

$$\omega_{best} = \operatorname{argmin}_{\omega} d(\omega(p), \omega) \quad (2)$$

但是为了避免一些纯边缘复制，或者图像过于整齐而导致失真，我们用了一些技巧，来放松匹配模块的选取：

$$d(\omega(p), \omega) < (1 + \epsilon)d(\omega(p), \omega_{best}) \quad (3)$$

这里 $\epsilon = 0.1$ ，之后我们从满足条件的这些模板中随机抽取一个作为匹配模板，并取其中心元素来填充待填充像素。

2.1.2 方法点评

理论上来说，这种方法对图片本身的规律性和平稳性有较高的要求，能够很好地解决规律性纹理的填充问题。然而，这种方法的一个显著缺点是运行速度非常慢。尽管如此，我们会在后续看到，这种方法在填充简笔画图像上展现出了惊人的效果。

2.2 Image Completion using Planar Structure Guidance

这个方法是 JiaBin Huang 在 2014siggraph 的一篇文章 [2]，据说是 Image Completion 传统算法的收官之作¹。该方法基于很多图形学和数学的知识，难度很高，本次报告只介绍一般性的流程。该方法在本次作业中对风格不一的四张图均能做到不错的填补。

2.2.1 图像补全步骤

1. 首先先将图像中的不同 plane 的参数求解出来，这里涉及到的技术包括 line segment extraction, vanishing point 估计和根据 vanishing point 分组等。
2. 根据 plane 的参数，对 plane 进行 projective transform 和 affine rectified
3. 对 plane 进行 rectified 以后，找到图像中的 regularity(regularity 可以理解为 plane 中有规则的物体)。
4. 利用检测到的 regularity 的位置作为指导，即在对已知区域进行采样的时候，尽可能选取 regularity 位置的 patch 作为候选。
5. 构建损失函数进行优化。

这里有很多名词需要理解，下面简单解释一下这些名词：

- plane: 可以理解为一张图像中不同物体的 2D 投影平面。
- line segment extraction: 在图像的已知区域中检测边缘并拟合线段。
- vanishing point: 在透视投影中，平行线在远处相交的点称为消失点。在图像中，消失点通常表示物体的朝向或视角。

¹网友说的

- projective transform: 投影变换是一种将三维空间中的点映射到二维图像平面上的变换。它可以用于描述相机的成像过程，以及将三维物体投影到二维图像上。
- affine rectified: 是一种将图像从透视投影转换为仿射投影的变换。它可以用于消除图像中的透视失真，使得图像中的平行线在仿射投影中仍然保持平行。

2.2.2 损失函数

下面主要介绍一下损失函数：

$$\min_{\{\mathbf{t}_i, \mathbf{s}_i, m_i\}} \sum_{i \in \Omega} E_{\text{color}}(\mathbf{s}_i, \mathbf{t}_i, m_i) + E_{\text{guide}}(\mathbf{s}_i, \mathbf{t}_i, m_i) \quad (4)$$

，如上展示，损失函数主要为两部分，颜色损失和引导损失。颜色损失顾名思义就是在 RGB 空间中的匹配损失，而引导损失则分为三部分：

$$E_{\text{guide}}(\mathbf{s}_i, \mathbf{t}_i, m_i) = \lambda_1 E_{\text{plane}}(\mathbf{s}_i, \mathbf{t}_i, m_i) + \lambda_2 E_{\text{direction}}(\mathbf{s}_i, \mathbf{t}_i, m_i) + \lambda_3 E_{\text{proximity}}(\mathbf{s}_i, \mathbf{t}_i), \quad (5)$$

这涉及了许多图形学的知识，具体细节可以参考 [2]。

这个损失函数的好处在于，用户可以根据图片的特点，对惩罚系数进行调整。实验表明，不同的惩罚参数，填充的效果非常不一样，在设置惩罚参数时，需要对图像本身有一定的理解且需要进行一定尝试。

2.2.3 方法点评

虽然这种方法的复杂度较高，并且理解起来有一定的难度，需要对数学和图形学都有一定的理解，但实际应用效果却相当出色。只要我们合理设定损失分配，这种方法在处理这次作业中的四张图像时，展现出了不错的效果。因此，尽管理解和应用这种方法有一定难度，但它的潜力是无可置疑的。

2.3 其他方法

除此之外，我们还使用了课件中的方法加以对比。

- "inpaintExemplar": 基于块匹配的填充算法。
- "inpaintcoherent": 基于相关性系数传输理论衍生出的方法，本质上是一种非局部匹配。

为了之后叙述简洁，我们将这以上四种方法简称列表如1

全称	简称
inpaintExemplar	Exemplar
inpaintCoherent	Coherent
Texture Synthesis by Non-parametric Sampling	TSNS
Image Completion using Planar Structure Guidance	ICPSG

表 1: 方法简称对照表

3 结果展示

由于嘉哥给的四张图片风格各异，每张图片在不同方法下的填充效果各异，故我们分图片来对结果进行分析描述。

3.1 砖块图

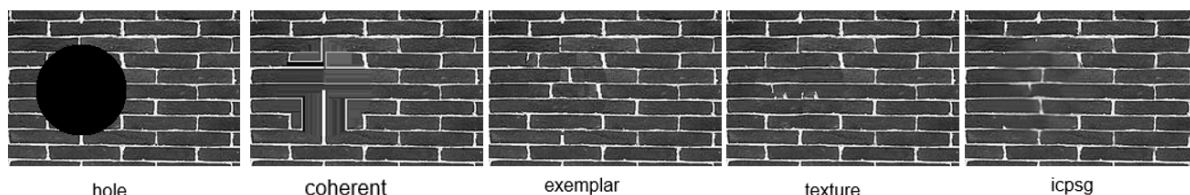


图 2: 砖块图总览

粗略来看：由于 Coherent 方法是基于相关性传输理论的填充方法，它很难捕捉到砖块的整体结构，故表现不佳；exemplar 方法很好的捕捉了砖块的块结构，整体上更像是砖块；基于随机采样的纹理填充 texture，表现也不错，目测与 exemplar 不分伯仲；而最后的 icpsg，虽然也能捕获块结构，但是在颜色上有一些失真，原因可能是损失函数的惩罚比重没有设置好，应当基于颜色损失更多的比重。

砖块图是一个结构规则，整体平稳的图片，故基于块状填充的 exemplar 是四种方法中最合理的，它充分利用了图片的规律性，能更好的捕获砖块大小，在合适的位置“画白线”。

3.2 指纹图

粗略来看，除了 Coherent 的结果比较离谱，其他三种方法都还不错，但是我们放大来仔细看，每一种方法都有自身的一些问题。

通过下图放大对比图4可以看到，基于块填充的 Exemplar 方法在凹槽处脉络不够清晰，非参抽样方法的 TSNS 也在凹槽处出现了崩溃，纹理直接消失。其实这两种方法都和块匹配有关，在匹配到这个部位时，源图像中没有比较相似的块供参考，故这两种方法都几乎在同一位置出现问题。

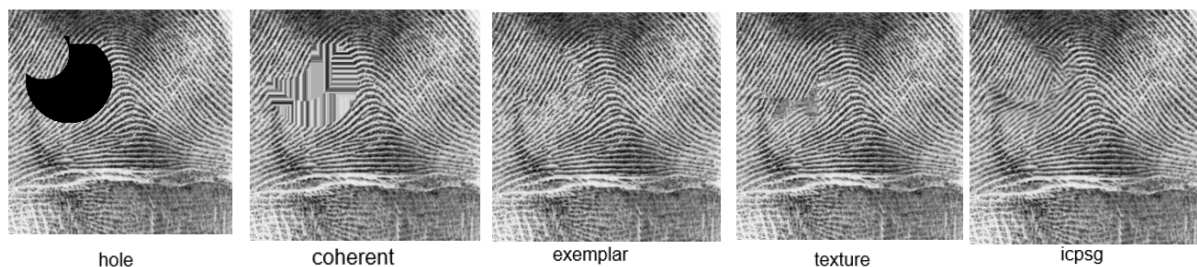


图 3: 指纹图总览

ICPSG 通过一系列的几何操作产生的图片，虽然在纹理方面基本上是连续的，但在纹理的变化上并不是很自然，存在一些纹理突变的情况，且整体纹路不够平缓。这可能是因为我提高了方向损失和邻近损失，确保了纹理的方向是正确的。但这种方法更倾向于检测直线，所以生成的纹理变化可能会更加尖锐。

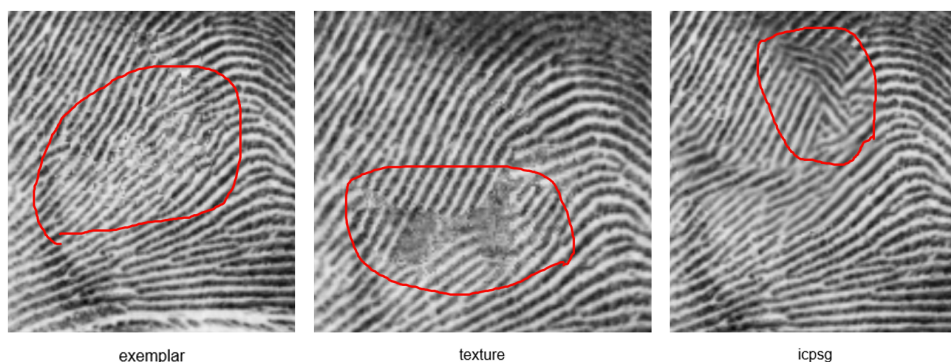


图 4: 放大对比图

3.3 简笔画



图 5: 简笔画总览

简笔画这张图的填补与前两张不同，前两张图片的源图像蕴含了丰富的信息，能够为目标区域提供足够的指导，而简笔画这张图的填充则不同，目标区域的信息很难通过源图像预测。相比于填充内容是否有语义一致性，我认为更应该关注算法对这张图的创

造性。²

对于简笔画这张图的填充，coherent 方法将房子的纹理补的很好，但是在主体内容上出现了大块的马赛克；至于 exemplar 方法，仿佛是随意的复制了左边的内容，使画面显得丰富，但细看下，却宛如孩童的涂鸦。然而，若是缺失的部分就是一个孩子天真无邪的涂鸦呢？故个中优劣，由读者评判。

其实这里最让人意外的，是基于随机采样的纹理填充 TSNS 方法，不光笔触连续，‘氧’字神来之笔，甚至还补了个桃子，真是意外之喜！唯一不足的一点就是房子的下沿没有补齐。但瑕不掩瑜，这个结果还是很不错的。

我觉得 ICPSG 的方法就像一个传统意义上的优秀学生，其对整体图像的补全非常合理，包括房屋的轮廓和部分草坪的延续。当然，也存在一些不足，比如“氧”字并未被完全补全，中间还有一些奇怪的笔触，可能是算法参考了嘴边的树根部分。这样的结果并不令人意外，因为该方法本质上就是尽可能地找到源图像的信息进行补全。

3.4 水果图

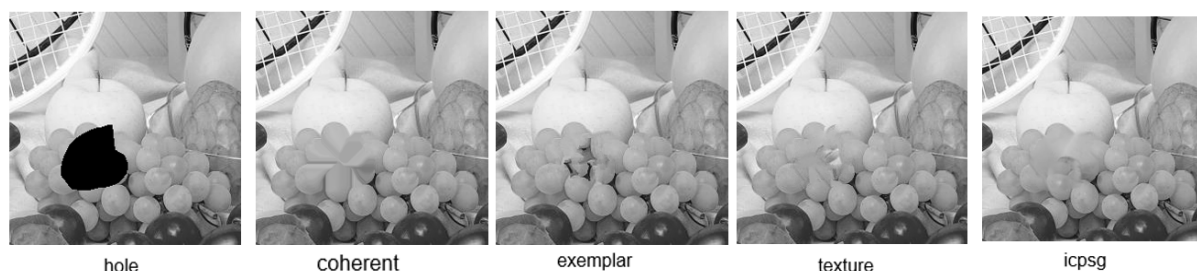


图 6: 水果图总览

水果这张图的补全难度是最高的。相比于前面三张图，水果图有立体感，且缺失部分是苹果和葡萄的交界处，这对算法理解边界又提出了很高的要求。

粗略来看，四个方法的生成结果风格各异。exemplar 方法和 TSNS 方法基本处于一种崩溃的状态。前者是生成了“烂葡萄”，后者则是胡乱涂鸦，蒙混过关；coherent 方法生成的葡萄有着明显的轮廓，已然非常不错，但是却无法感知葡萄的形状，而生成了“变异葡萄”；相比之下，ICPSG 方法令人意外的生成了一个完整的葡萄，但是在和苹果交界处的地方，它却没有生成葡萄间合理的轮廓和边界。

总得来说，基于块填充的方法不适用于该图片，只有 ICPSG 方法勉强过关。

²个人认为这个图像适合使用现在 AIGC 中比较火的“stable diffusion”去做语义引导的填充，但是由于一些不可抗原因，本次报告并未能使用该方法

4 总结

就方法上来讲，在机器学习理论中有个著名的定理，叫做“no free lunch(NFL)”，“没有免费午餐定理”，放在本文中的数学方法也是适用的。对于同一种方法，在不同图片上的表现是不同的；而达到同样的效果，不同的方法所需要的成本，计算量和复杂度也是不同的。³在本次报告中，coherent 的方法在前三张图中不如基于块填充的 EXemplar 和 TSNS 的方法，而在最后一张水果图上却超过了它们；TSNS 方法虽然在前三张图表现良好，甚至再简笔画的“创作”上令人惊喜，但它的计算速度非常慢；最后的 ICPSG 的方法，虽然每张图都生成了还不错的结果，但是该方法复杂度高，调参难度大。

就图片上来讲，四张图片风格不一，如果用同一种方法处理，对该方法的泛化能力无疑提出了极高的要求（这可能会让一些用机器学习的同学感到压力重重）。砖块图相对平稳，语义简单，非常适合块匹配填充的方法；而指纹图的纹理模式虽然简单，但并不具有周期性，每一部分都与众不同，因此，各种方法的填充结果都不尽完美；简笔画图片因为缺乏源图像的语义指导，更是对算法的创新能力提出了挑战；至于水果图，由于其具有 3D 结构和不同平面边界的生成，填补的难度相当大。

图片	Coherent	Exemplar	TSNS	ICPSG
砖块图	不好	良好	良好	良好
指纹图	不好	良好	良好	良好
简笔画	不好	及格	优秀	良好
水果图	及格	不好	不好	良好

表 2: 评价表

A 代码

所有代码均上传到 github 的个人 repo 中,地址为<https://github.com/yangCheston/hw0FImageInpaint.git>.

³不得不提出疑问，通用人工智能真的存在吗？

参考文献

- [1] A. A. Efros and T. K. Leung, “Texture synthesis by non-parametric sampling,” in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2, pp. 1033–1038, IEEE, 1999.
- [2] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, “Image completion using planar structure guidance,” *ACM Transactions on graphics (TOG)*, vol. 33, no. 4, pp. 1–10, 2014.