

灰度图像着色算法

摘要:

我们将介绍一种为灰度图像着色的普适技术，通过将颜色信息由原始的彩色图像映射到作为目标的灰度图像。虽然为灰度图像添加色度值的一般问题是没有一种精确客观的解决方法，本文所用的方法试图找到对人工劳动要求最少的着色方法。我们通过图像间亮度和纹理信息的匹配来使源图像中的整体色彩映射到目标图像中，而不是人工在调色板中选择 RGB 色彩涂到一个个分立的色块中去。我们选择的方法是：只映射色度信息，保持目标图像中原始的亮度信息。进一步地，我们允许用户用矩形框标注两幅图像间匹配的区域，以此可以增强着色程序的效果。我们将展示在亮度和纹理具有足够的可区分性时，这种简单的技术在许多图像和视频中都有很好的应用。这些生成的图像证明了这种技术在诸多应用领域的潜力和实用价值。

关键词：图像处理，颜色，纹理合成，视频

1 简介

为了增加某些图像（例如老的黑白照片，以前的电影或是科学插图）的视觉感染力，我们可以为这些灰度图像着色。除此之外，利用色度和亮度的变化，可以提高我们对某些科学图像中所含信息的感知。

为灰度图像着色的任务涉及对一张仅含一维信息（亮度或灰度）图像进行三维像素值（RGB）的分配。由于不同颜色可能有着相同的亮度值，只是在色度或饱和度上有差异，使得对灰度图像的着色问题没有所谓的“正确”方法。正因为这些模糊性，人的参与在着色过程中起着非常重要的作用。即使在这种能够将亮度值自动映射到颜色值的“伪彩色”过程中，对颜色表选择也是取决于人的决定。

因为大多数在电影业中使用的着色软件都是有专利所有权的，所以描述着色方法的详细技术文档通常不会公开。然而，还是有一些网上的文章描述了这类软件的使用：需要使用者细致指明每一个色块的颜色。例如，据描述，有一个软件包的使用方法是这样的：图像首先被分割成多边形，这样使用者可以对每一块区域分别着色，很像是彩色画图本；然后系统将会追踪这些多边形在不同帧中的位置并相应的着色，以此来减少需要使用者手工着色的帧数。另一种方法是先用照片编辑软件手动地，或者自动地在整幅场景中选择若干成分，然后，通过覆盖一个不透明度低的图层，为这些成分涂上从调色板中选定的颜色。

有关在信息可视化中使用颜色的应用还有很多。例如，Gonzalez and Wintz[1987]描述了一种将机场里 X 光检查仪所获得的行李的灰度图像转化为伪彩色图像的简单方法。这种方法通过对每种颜色通道的分开转换，实现了将密度与炸药相近的物体标注成亮橙色，其他物品涂成蓝色系的。进一步说，颜色可以被添加到一些作为说明图例和用于教育目的的科学图像上。在医学领域，通过对一些只有灰度图像的影像——例如磁共振成像(MRI)，X 光透视和电脑断层扫描(CT)——着色，可以加强它们的展示效果。

伪彩色是一种为 X 光透视、磁共振成像、扫描式电子显微镜以及其他不存在颜色信息的图像形式进行着色的通用技术。Pratt [1991]将这种方法描述为一种“图像增强”技术，因为它可以被用于“增强图像细节的可分辨性”。伪彩色的最基本形式可以表示为某种变换 T ，使得 $c(x, y) = T(f(x, y))$ ，其中 $f(x, y)$ 表示原始灰度图像， $c(x, y)$ 是最终得到的表示 RGB 通道的颜色值的颜色向量。这种方法的一个简化了的例子是对(原始灰度图像)数据应用某张任意的颜色表，为每种灰度值赋予某种全局、单一的颜色向量。这种方法的优势在于没

有引入多余信息，因此它不会改变原始数据的信息内容。例如，在一幅“伪彩色化”的 MRI 图像中，存在一种密度值和颜色值的一一对应，虽然颜色的选择是任意的。然而，如果使用一张不随着亮度单调递增的颜色表，伪彩色图像有可能会引入感知上的扭曲。已有研究表明，使人们感觉“自然”的脸部图像和颜色表对于亮度的正相关性之间有很强的关联性。[Rogowitz and Kalvin 2001]

我们对于将颜色由一幅图像映射到另一幅图像的想法受到了 Reinhard 等人[2001]的启发——他们实现了两幅彩色图像之间的颜色映射。在他们的工作中，颜色通过一种简单但是惊人成功的步骤从一幅彩色图像映射到了另一幅彩色图像。基本的方法是对两幅图像的颜色值在三维空间中的分布进行匹配，然后将目标图像的颜色值空间分布映射为源图像颜色值的空间分布。进一步地，可以引入选择框来匹配两幅图像的相似区域。

在我们的研究中，灰度图像被表示为一维分布，因此两幅图像只有亮度通道可以进行匹配。因为一个单一的亮度值可以完全表示出一幅图像的不同部分，所以我们用相邻像素的统计值来指导匹配过程。一旦一个像素点匹配了，颜色信息就被映射过去，同时保留原始图像的亮度值。在困难的情况下，可以使用一些选择框来帮助完成原始图像和目标图像间的匹配过程。当源图像和目标图像间的颜色映射完成之后，最终的颜色将通过使用 L2 距离匹配灰度图像中各像素与选择框内目标图像中的像素来分配。因此，每一个像素的匹配结果是由将它和同幅图像中其他像素点的匹配结果所决定。我们发现，这种简单的算法对许多不同类型的图像都有很好的效果。进一步地，这些方法可以很容易地被拓展到视频应用。这时，着色过程首先被应用于视频中的某一帧。然后，场景中的其他帧将利用原始帧的选择框进行颜色分配。我们发现只要场景中的物体没有发生剧烈的变化，这种着色过程的效果出人意料的好。

2 颜色映射算法

在这一部分，我们将描述颜色映射的一般算法；之后，基本思想将延伸到使用选择框。颜色映射的一般流程包括几个简单的步骤。首先，将各幅图像投影到 $l\alpha\beta$ 色彩空间。我们采用随机撒点抽样在彩色图像中选择一个像素子集作为样本。之后，我们按扫描线顺序检查灰度图像中的每一个像素点，选出与其相邻统计值最佳匹配的彩色样本。是否最佳匹配由亮度值和其相邻统计值的加权平均值所决定。然后，最佳匹配点的色度值 ($\alpha\beta$) 被映射到灰度图像，从而生成最终的图像。使用选择框的颜色映射算法包含相同的全局图像匹配步骤，不过它的应用范围仅限于源选择框和目标选择框之间。然后，目标选择框内的彩色像素点被作为源像素，通过纹理合成的方法将颜色信息映射到剩余的非彩色像素点。下文将给出更具体的算法细节及讨论。

2.1 全局图像匹配

彩色（源）和灰度（目标）图像都被从 RGB 空间映射非相关的 $l\alpha\beta$ 色彩空间以便进行进一步分析。 $l\alpha\beta$ 色彩空间的三个坐标轴间具有最小的相关性。这种色彩空间提供了三个非相关的主通道，包括一个无色的亮度通道 (l)以及两个大约相当于黄-蓝和红-绿两组互补通道的彩色通道 α 和 β 。因此，发生在一个彩色通道中的变化能尽可能小地影响其他通道的值。我们在这种算法中选用 $l\alpha\beta$ 色彩空间是因为它能为彩色图像提供一个非相关的无色通道。这就允许我们有选择的通过彩色通道 α 和 β 将彩色图像映射到灰度图像，同时不引起交叉通道干扰。我们直接沿用了 Reinhard 等人的映射算法。

为了将色度值从源映射到目标，灰度图像中的每一个像素点都必须与彩色图像中的某个像素完成匹配。这种匹配比较是基于亮度值和该像素点的相邻统计值。亮度值取决于 $l\alpha\beta$

色彩空间中的 I 通道。考虑到两幅图像间亮度的全局差异，我们根据源图像和目标图像间亮度概率分布的线性比例关系对亮度值进行了重新分配。这有助于在两幅图像的亮度值域内获得更好的相关性，同时不改变目标图像的亮度值。

相邻统计值是预先计算好的，由像素点邻近点的亮度值的标准差组成。我们发现，将邻近区域设为 5×5 对大多数图像都有很好的效果。对一些有问题的图像我们会把邻近区域设置的更大。

由于大多数可见的像素点间的显著变化都是由亮度差异造成的，所以我们可以限制所使用的作为源像素点的样本数量而仍然可以包含图像中足够大的颜色范围。这允许我们减少对灰度图像中每个像素点进行比较的次数，也就缩短了计算时间。我们发现随机撒点抽取大约 200 个点就足够了。然后，按扫描线顺序检查灰度图像中的每个像素点，基于其亮度值（占 50%）和相邻统计值（50%）的加权平均值选出与其最佳匹配的彩色样本点。我们曾经在最佳匹配的选择策略中加入了邻近点的平均值，并且尝试了不同的权重，不过没有发现结果有显著的不同。一旦找到了最佳匹配像素点， α 和 β 通道的值将被映射到目标像素点，同时原始的亮度值得以保留。

这种自动的整体匹配算法对于处理两张有着相似的颜色区域和亮度值的图像有相当好的表现。然而，当目标图像中的区域亮度与有着合适画面结构的源图像不相近时，这种算法就得不到正确结果了。（图略）

2.2 选择框

为了允许使用者在颜色映射的过程中发挥更大的主观能动性，也为了获得更好的结果，选择框被使用在了两幅图像的相应区域。图 2a,b,d 演示了基本想法。第一步是使用上文描述的颜色映射的一般算法，不过现在只用在对应的选择框的范围内。这允许使用者能够选择在源图像和目标图像间的某些区域进行颜色映射。我们认为，可以期望这样的个性化选择框对着色结果有好处，因为这样应该会减少相同选择框中不同颜色区域间亮度级的重叠。如同在全局算法中一样，我们也进行了亮度值的重新分配，不过只是在选择框的范围内。同样的，我们使用随机撒点抽样，在一个选择框中大约抽取 50 个像素点。第二步和纹理合成算法很相似[Efros and Leung 1999; Efros and Freeman 2001]，通过使用 L2 距离寻找纹理匹配。我们定义误差距离 E 是灰度图像中的邻近点 N_g 和着色了的选择框中的邻近点 N_s 之间的 L2 距离：

$$E(N_g, N_s) = \sum_{p \in N} [I(p) - S(p)]^2$$

其中 I 代表灰度图像， S 是被选择的彩色图像的亮度通道， p 代表邻近的像素点。

注意，在这个阶段，我们不再寻找彩色图像中的纹理匹配，而只是寻找已着色的选择框内的匹配。这种方法的优点是在第一步中我们可以有选择的指定颜色映射的范围，从而防止虽然具有相似的相邻统计值但是位于图像中错误位置的像素点干扰了目标选择框中的颜色。它也允许使用者将图像任何部位的颜色映射到指定区域，即使这两个相应区域在纹理和亮度值域上都有很大的差异。其次，由于我们可以认为同一幅图像内部比两幅不同图像之间具有更好的纹理连续性，我们能够期望与彩色目标选择框里的图像在纹理上相似的像素点在颜色上也与它们相似。

2.3 视频

对视频的着色可以使用上文介绍的着色算法自动完成。为了给场景中的每一帧着色，

我们首先将颜色从一张源彩色图像映射到某一幅目标帧。然后视频序列中的每一帧都可以使用目标帧中相同的选择框进行着色。如果我们能使用这些算法成功地为单独的一帧着色，那么也能类似地为其他与之有着相同元素的帧着色。我们提供了三段视频记录作为视频处理的样本。

3 结果

图 3 展示了算法被应用于各个图像领域所获得的结果。图 3a-c 展示了为植物，人脸和风景照片着色的结果。这种技术在亮度能被清晰聚类或者各个区域有明显的纹理特征的图片中有很好的表现。然而一般来说，当前的技术在对于人脸的处理上效果并不理想。虽然选择框里的颜色映射完成的很好，但是 L2 距离并不总是能充分度量出皮肤和嘴唇的差别，有时候连衣服和发色都不能区分。

图 3d-f 展示了这种算法在不同类型的科学数据上的应用。虽然这种技术并不是针对使用医学图像的临床诊断而设计，但是它仍然可能用于解剖图或是增强科学展示效果。

虽然我们展示了这种算法在许多图像领域的成功应用，我们并没有说这种技术对于大多数图像都能有很好的效果。显然，当我们只考虑某个像素点的一个很小的邻近区域时，通常是不可能判断这个邻近区域是属于哪种纹理的。然而，通过使用高分辨率的图像和更大的邻近区域，我们可以获得更好的结果。进一步地，我们相信如果使用更好的纹理分类方法，以算法的复杂化和计算时间开销的增大为代价，上文提供的这种基础着色算法能够适用于更多的图像。

在使用奔腾 III 900MHz 的 CPU 的环境下，以最优化的 MATLAB 代码实现的这种算法，处理一张图像所需的运行时间大约在 15 秒到 4 分钟之间。运行时间的差异由用于比较的样本数目，选择框的数量，邻近区域的大小和图像的尺寸所决定。大部分图像可以在一分钟内得到相当好的着色结果。

4 总结

这篇论文中，我们用公式给出了一种新的，普适的，快速且对用户友好的方法来解决为灰度图像着色的问题。虽然完成这项任务的标准方法是通过一个全局调色板设定像素点的颜色值，但是我们的技术允许使用者首先选择一幅合适的图像，其次将这幅图像的颜色整体映射到待处理的灰度图像上。我们有意不要求对图像进行标注或引入空间信息，以此保持基本技术的简单性和一般性。我们的技术通过引入少量使用者的指导，可以适用于更广泛的图像。在这种模式下，使用者首先将想要的颜色从彩色图像中指定的一组样本区域内映射到灰度图像相应的一组样本区域中。然后，在第二步也是最后一步着色过程中，利用已着色的区域，通过纹理合成的方法，为灰度图像中其余的像素点着色。目前，我们使用 L2 距离来评估图像内的纹理相似性。将来，我们相信通过使用更复杂的纹理相似性评估方法，这个技术能得到很大的提升。

鉴于网络图像搜索引擎复杂性的增加和可被用于简单定位合适的彩色图像的集中的、可索引的图像集合的出现，我们采用一张彩色示例图像来为灰度图像着色的技术已引起了特别的关注。最终，我们也可以利用基本纹理样本数据库来完成现在着色算法中需要使用者指导的初步颜色映射。