基于视频的动态纹理合成

文庆福
2011013239 thssvince@163.com
清华大学软件学院11班

2014年 6月 13日

目录

1 背景介绍

动态纹理是指描述某种动态景观的具有时间相关重复特征的图像序列，比如海浪、瀑布、火焰、烟雾等。它在视频制作、虚拟仿真、虚拟漫游中具有重要的应用价值。动态纹理的合成一直是计算机图形学的主要目标和难点之一。

此次作业的主要目标就是针对已给出一组 150 帧的动态纹理数据，建立一个状态空间模型，并合成 2000 帧的新动态纹理帧。

希望通过此次作业，对以下几个方面的知识有所学习：状态空间模型的建立，系统分析相关结论的综合应用，MATLAB 的矩阵计算、图像操作、视频操作等的用法。

2 算法描述

根据老师已经给出的状态空间模型：

$$\begin{cases} x(k+1) = Ax(k) + Bv(k) \\ y(k) = cx(k) \end{cases} \tag{1}$$

因此我们只要依据原有的 150 帧图像计算出这个模型，就可以利用模型进行预报合成新的图像了。

从已给的源代码中可以看出，我们就用$y(k)$表示第 k 帧的图像，图像是大小是$row * col$，那么$y(k)$就是一个长度为$row * col$的一维向量。这里考虑比较简单的情况，即图像是灰度图（若不是灰度图则先转化为灰度图）。

首先，将所有图片数据读入后，我们可以得到$Y = [y(1), y(2), \ldots, y(T)]$。由$Y = CX$，利用矩阵奇异值分解可以计算出$C$和$X$。通过对$Y$进行奇异值分解，我们可以得到$Y = U\Sigma V^T$，其中$U$的前$n$列作为$C$，则$X = \Sigma(1:n, 1:n)V(:, 1, n)^T$。

接下来根据老师提供的方法计算：

$$A = [x(2), y(3), \ldots, x(T)][x(1), y(2), \ldots, x(T-1)]^{-1} \tag{2}$$

$$Q = \frac{1}{T-1}\sum_{t=1}^{T-1} \hat{v}(t)\hat{v}^T(t) \tag{3}$$

$$Q = U_Q \Sigma_Q {U_Q}^T \tag{4}$$

$$Q = BB^T \tag{5}$$

其中(2)式中$\hat{v}(t) = x(t+1) - Ax(t)$，视为估计的误差。根据以上几个式子就可以求出$AB$，这样整个模型就计算出来了。

3 实验过程

根据以上算法，在原有的 main.m 的基础上编写了代码，包括了main.m、readImage.m 以及 train.m 三个文件。

其中主函数在 main.m 中，读入数据，训练模型，合成新的图片，导出视频。

readImage.m 中的函数 function [Y, row, col] = readImage(imagedir) 用于读入图片数据，其中 imagedir 就是输入图片存放的路径，Y 是训练数据，row 和 col 分别是图片的行数和列数。

train.m 中的函数 function [Ahat, Bhat, Chat, x0, Ymean] = train(Y, n, nv) 用来计算模型参数。其中 Y 是训练数据，n、nv 是 x 和 v 的维数。$Ahat, Bhat, Chat$ 是训练结果，$x0$ 是模型输入的起始 x，Ymean是 y 的均值。

下面通过对不同的输入数据进行测试，同时不断调节 n 和 nv 以达到临界稳定。倘若系统趋于稳定，则最后 y 将收敛到某个值，那么生成新的纹理帧就不具有动态性；如果系统发散，则生成的 y 有可能使得两帧过渡不自然，也很可能得到有纯黑纯白图像组成的动态纹理。显然上述两种情况都不是我们期望的。因此只有临界稳定的系统才能持续生成我们需要的动态纹理。

通过大量实验发现，当 n 选取过小时，x 的维度比较小，生成的 y 容易不稳定，而当 n 过大时，生成的 y 容易导致相邻两帧连贯性差。nv 决定了白噪声的维数，而白噪声主要可以促进系统平滑过渡，因此选出合适的 nv 也是很有必要的。因此在调解的过程中，我们可以先选择调节 n, 然后再微调 nv。

最终我选择的输入数据是 see-i-far-2, 这组数据训练出来的模型比较容易临界稳定，我只是经过一两次调节就达到比较合适的效果了。最终选取的 $n = 50, nv = 48$。我猜想，其中愿意可能是这组训练数据中的颜色差异并不大，数据本省波动比较小，不太容易发散。

4 实验总结

状态空间模型中的 x 维数 n 和白噪声的维数 nv 这两个量作为参数，经实际调整发现对生成结果的影响很大，n 对能不能生成期望结果起决定性作用，而虽然 nv 影响较小，但选择合适的对提升生成结果的真实感、连贯性都颇有帮助。

本次实验的一些方法和分析方式让我深受启发，我们是按照老师给出的模型和算法进行合成，能够得到比较实用的结果，而且这个模型其实非常简单，如果换用其它复杂一点的模型不知道是否还能得到这样的效果。其实我们有时候在做算法的过程中，往往一味的为了追求算法的完备性而将算法搞得非常复杂，其实有时候简单就是最好的。我想这样一种思路对我们解决实际的工程问题应该会大有帮助。