

# Exploring Features in a Bayesian Framework for Material Recognition

## 摘要:

我们对从一张图像中鉴定物体的材料（比如玻璃，金属，织物等）感兴趣。但不像其他视觉识别工作，我们很难找到很好的，可靠的特征来把这些材料区分开来。我们的方法是用一大堆低级，中级图像特征来描述材料外观的各种属性。我们提出一个增强的主题模型(aLDA)，在贝叶斯生成模型下把这些特征结合起来，并通过训练学习使这种结合达到最优。

## 引言:

材质识别是视觉识别的一个重要方面，我们经常在日常生活中碰到各种材料并通常由他们的外观来判断。比如说，判断一个冰面是否可以行走，判断农产品是否新鲜，材料的质量经常会对我们的决策产生影响。因此，很有必要建立一个能从图像中识别材质属性的视觉识别系统。

这个问题通常在反射评价的背景下完成。物体表面的外观取决于几个因素：光照，不同尺度下的几何形状，物体表面反映的特征属性，通常可以由双向反射率分布函数（BRDF）和它的变体表示。

本文我们主要研究识别材料类别，比如玻璃，金属，织物等，而不是描述反馈特征。金属的各种反馈属性通常是与其它类别相互联系的，本文我们主要研究这种关系。必须指出，仅仅知道材料表面的反馈属性对材料分类还远远不够。比如说，透明的材料我们不能分辨出它是玻璃的，塑料的还是蜡的。

由于一种材料的外观可以有各种各样的，要提取能很好分辨出这些材料的特征就很难了。我们的方法是用一些低级，中级特征来表示材料外观的不同属性。加上一些得到验证过的特征，颜色，jet 和 SIFT，我们还引入一些新特征，比如说边缘曲率特征，边缘 HOG,垂直 HOG。我们把图像转化成 BOW 然后用 LDA 来对这些词汇的分布建模。

选择正确的图像库很重要，很多现存的材料/纹理图像库没有获取现实中各种各样的材料，CURET 只是个试题数据库，KTH-TIPS2 每个种类的样本太少。本文中，我们使用自己创建的 FMD 图像库，它有 10 个材料种类：织物，植物，玻璃，皮革，金属，纸，塑料，石头，水和木材。每个类别我们有 100 张图像。所有的图像都被归一化成 512\*384 大小，且都被标记好了类别。这些图像的外观差异性很大，尽管材质识别是个很困难的工作，我们的方法还是取得了很好的效果。

## 相关工作:

图像的材质识别与对象识别有很大不同，尽管对象的识别有时候也是个材料种类，一个给定的物体可以由不同的材料组成，不同的物体也可以由同一种材料组成。因此，对象识别的一些好的方法比如说形状上下文，目标探测和类标传递并不适合与材质识别。实际上，很多对象识别系统基于材料不变特征且经常忽略材料信息。

材质识别与纹理识别相关但也有不同点，纹理也可以作为材料外观的重要组成部分，比如木材的纹理就跟光滑的金属纹理不一样。但是，不同的材料也可以有相同的纹理，因此，纹理识别并不能很好地运用于材质识别。

材质识别也和 BRDF（双向反射率分布函数）估计不一样，计算机图像学领域经常抓取现实材料的外观。材料的视觉外观，比如木材和皮肤，已经被建模成 BRDF 或相关的图像表示比如 BTF 和 BSSRDF。如果 BRDF 已知，材质识别会变得很简单，但实际上，我们基本上不可能从一张单一的图像建立它的 BRDF。很多低级特征也被利用于鉴定材质，[Use of image-based information in judgments of surface reflectance properties] 发现图像的亮度直方图和我们对物体表面的反射率相关，

图像库的选择对视觉识别也很重要，CURET数据库包含61个在205中视角，光照条件下拍摄的纹理样本，已经成为3-D纹理分类算法的标准。

## 材质识别的特征：

定义能分辨不同材料种类的特征对我们的材质识别系统非常关键。由于我们不知道什么特征最适合材质识别，我们尝试了很多特征，有些来自目标与纹理识别，有些专门为材质识别发展。一旦相机和目标固定了，图像将取决于：1，表面的BRDF（反射，吸收和透射）；2，物体表面的结构；3，目标形状；4光照影响。我们的特征将包含以上所有因素。

### 1. 颜色和纹理

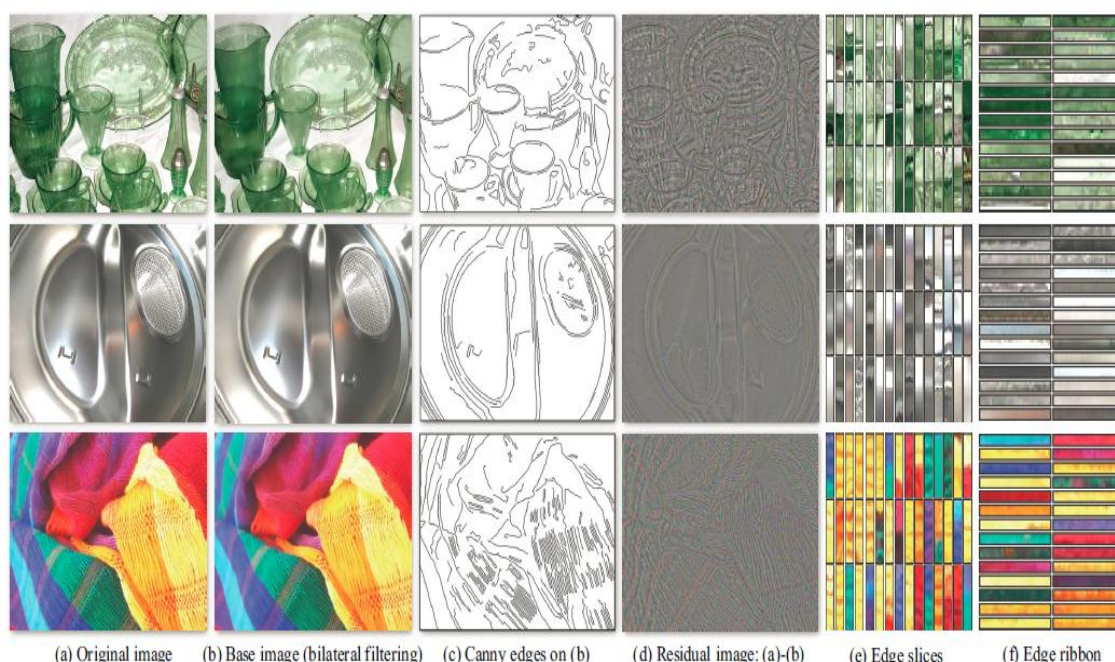
颜色是物体表面的重要特征，也对材质识别有很大帮助：木材物体一般是棕色，叶子一般是绿色，织物和塑料一般充满了显眼的色彩，石头一般是不透明的。我们从RGB图像中提取一个3\*3的块作为我们的颜色特征。

纹理特征也能对区分材料起很大作用。比如，石头和木材有很显著的纹理来区分他们。我们用两种特征来标识纹理：第一种通过多尺度，多方向的Gabor 滤波来包含图像的滤波响应，通常被叫做jet。另一种是SIFT，SIFT特征已经广泛应用于场景与对象识别。

### 2. 微纹理

两个有共同的BRDF（双向反射分布函数）的表面也会因不同的表面结构而看起来不同，比如说一个光滑，一个粗糙。现实中我们通常用触摸的方式来分辨，然而，我们的视觉识别系统能处理触觉的输入。比如我们可以看到皮革上的褶皱，玻璃物体的光滑表面。

我们使用[Two-scale tone management for photographic look]中使用的方法来提取物体表面的信息，使用双边滤波来处理图片然后进行进一步分析。



如上图所示，我们用玻璃，金属，织物3种材料为例。(a) 是图像的原图，我们使用双边滤波得到(b)，图像的滤波图像。在此基础上使用canny边缘探测得到(c) 和边缘图，我们提取边缘的曲率作为特征。我们用(b) 减去(a)，得到图像(d)，它显示了材料表面的细微结构，我们提取它的micro-jet 和micro-micro-SIFT作为特征。(e) 和(f)是edge-slice 和edge-ribbon的图像，我们将在之后介绍。同时，从(e)和(f)中我们也提取出HOG特征。

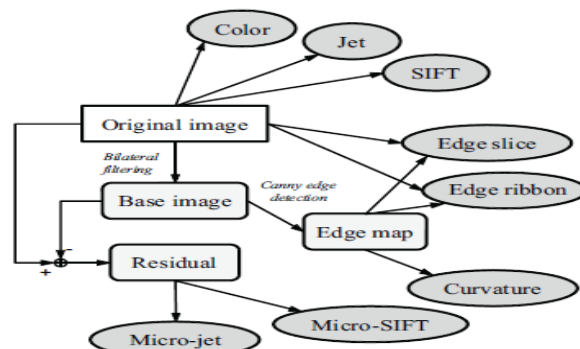
### 3. 大致形状

尽管一种材料可以有各种各样的形状，但物体表面的大致形状有时候跟它的材料还是相关的，比如织物和玻璃有长的，弯曲的边，金属有直的线条和锋利的角。物体表面的大致形状可以由图像的边缘图获得。我们对图像进行Canny边缘探测，剔除些短的边来获得图像的边缘图。为描绘不同材料种类的边缘图的多样性，我们在3种不同尺度下计算边缘图的曲率（curvature）作为特征。

### 4. 反射特征

光泽度和透明度对材质识别很重要。玻璃和水是透明的，石头是不透明的。这些特征经常在图像的边缘显示为特别的剧烈转变。我们用边缘方向的HOG特征表示这种特征。我们沿着边缘主方向截取一段固定大小的像素片段(slice)，计算每个像素的梯度，把这个片段(slice)分成6个cell，把这些方向梯度量化成12个bin。这种特征叫做edge-slice，我们也用同样的方法来表示图像的切线方向的变化，这种特征叫做edge-ribbon。

通过以上方法我们获取了许多对材质识别有用的特征color, SIFT, jet, micro-SIFT, micro-jet, curvature, edge-slice 和 edge-ribbon，这些特征生成的流程图如下图所示。其中，颜色，jet，SIFT直接由原图获得，其余的是由base图像和边缘图获得。由于我们不知道哪些特征会对我们的系统作用大，我们用一个贝叶斯学习模型来得到这些特征的最好结合。



## 贝叶斯计算模型：

我们把以上得到的特征转化成视觉词汇，并扩展LDA模型来选择最好的特征，然后学习每个类别的特征分布来进行材质识别。

### 特征量化与连接：

我们用k-means算法将每个特征聚类生成词典，然后把每张图像映射成视觉词汇。假设我们有m个特征，因此就有m个对应的特征词典，每个词典有 $V_i$ 个视觉词汇。由于特征是分开来进行量化的，因此图像的特征集可以表示为： $\{w_1^1, w_2^1, w_3^1, \dots, w_{N_1}^1\}, \{w_1^2, w_2^2, w_3^2, \dots, w_{N_2}^2\}, \dots, \{w_1^m, w_2^m, w_3^m, \dots, w_{N_m}^m\}$ ,

### LDA:

LDA是为单词的层次结构建模而发明的，[Latent Dirichlet Allocation]和 [A bayesian hierarchical model for learning natural scene categories]中详细阐述了它的详细过程。

### 训练:

我们通常会假设每个类别会同概率出现。但我们是各自独立地训练每个类别的LDA模型，仅仅是 $\beta$ 相同，训练过程可能不会在有限的迭代内聚合。因此，概率密度函数（pdfs）应该以同一比较为基础。我们设计以下贪心算法，通过最大化识别概率来学习 $\lambda$ 。

### aLDA:

我们使用全部的特征并不能取得很好的效果。我们设计了以下的贪心算法来找到最优的特征子集。

主要的思想是每次选择最好的，能获得最大识别效果的特征。当增加更多的特征会降低识别的正确率。由于我们随机将训练集H分割成L 和E ， L 用于参数学习，E用于交叉验证。当 $\mathbb{D}$  学习完一次了，我们用整个训练集H 重新学习 $\mathbb{D}$  的参数。

<p>Input: dictionary pool <math>\{D_1, \cdots, D_m\}</math>, training set <math>H</math></p> <ul style="list-style-type: none"> <li>• Initialize: <math>\mathbb{D} = \emptyset</math>, recognition rate <math>r = 0</math></li> <li>• Randomly split <math>H = L \cup E</math> <ul style="list-style-type: none"> <li>for <math>l = 1</math> to <math>m</math> <ul style="list-style-type: none"> <li>for <math>D_i \notin \mathbb{D}</math> <ul style="list-style-type: none"> <li>• Augment dictionary <math>\mathbb{D}' = \mathbb{D} \cup \{D_i\}</math></li> <li>• Concatenate words according to <math>\mathbb{D}'</math> using Eqn. (2)</li> <li>• Train LDA on <math>L</math> for each category (sharing <math>\beta</math>)</li> <li>• Learn prior <math>\lambda</math> using Eqn. (5) and (6)</li> <li>• <math>r_i =</math> recognition rate on <math>E</math> using Eqn. (3)</li> </ul> </li> </ul> </li> <li>end</li> <li>if <math>\max r_i &gt; r</math> <ul style="list-style-type: none"> <li>• <math>j = \arg \max_i r_i, \mathbb{D} = \mathbb{D} \cup \{D_j\}, r = r_j</math></li> </ul> </li> <li>else <ul style="list-style-type: none"> <li>break</li> </ul> </li> <li>end</li> </ul> </li> <li>end</li> <li>• Train LDA and learn prior <math>\lambda</math> on <math>H</math></li> <li>• <math>r =</math> recognition rate on <math>H</math></li> </ul> <p>Output: <math>\mathbb{D}, r</math></p>
--

aLDA贪心算法过程

## 附录：

### LDA 方法介绍：

LDA 方法使生成的文档可以包含多个主题，该模型使用下面方法生成 1 个文档：

Choose parameter  $\theta \sim p(\theta)$ ;

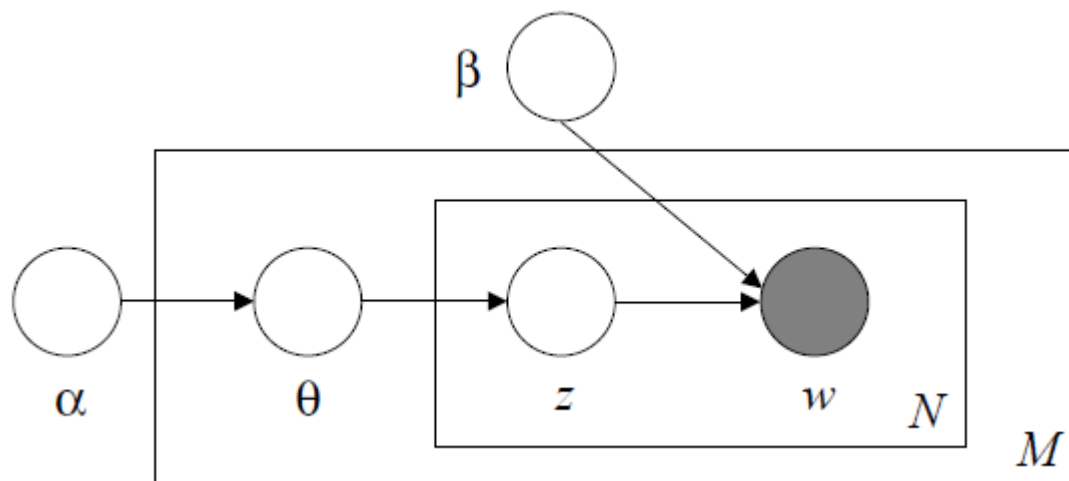
For each of the N words  $w(n)$ :

    Choose a topic  $z(n) \sim p(z|\theta)$ ;

    Choose a word  $w(n) \sim p(w|z)$ ;

其中  $\theta$  是一个主题向量，向量的每一列表示每个主题在文档出现的概率，该向量为非负归一化向量； $p(\theta)$ 是  $\theta$  的分布，具体为 Dirichlet 分布，即分布的分布；N 和  $w\_n$  同上； $z\_n$  表示选择的主题， $p(z|\theta)$ 表示给定  $\theta$  时主题  $z$  的概率分布，具体为  $\theta$  的值，即  $p(z=i|\theta)= \theta\_i$ ； $p(w|z)$ 同上。

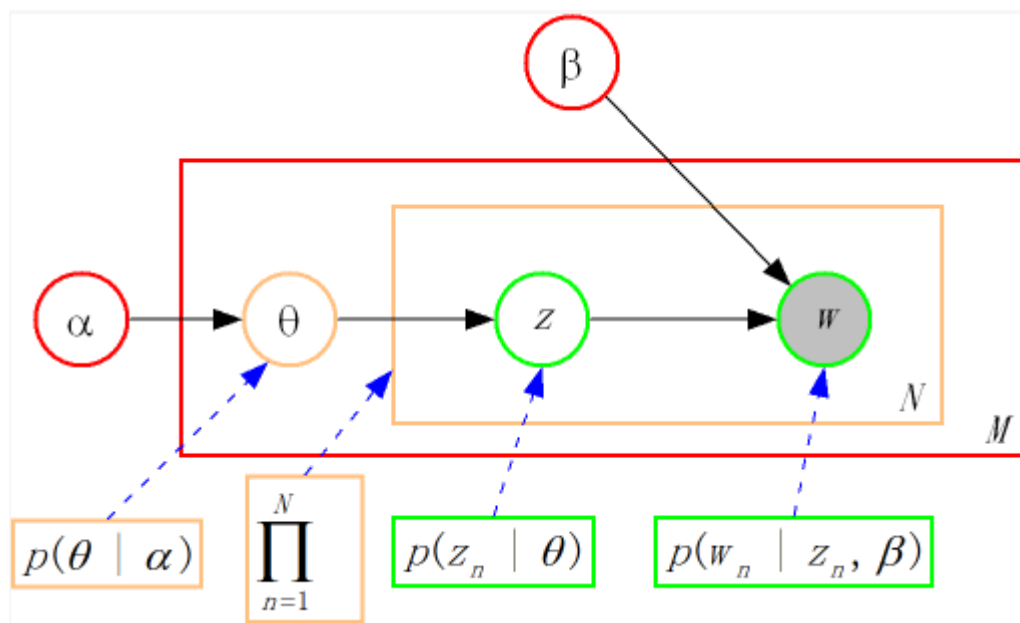
这种方法首先选定一个主题向量  $\theta$ ，确定每个主题被选择的概率。然后在生成每个单词的时候，从主题分布向量  $\theta$  中选择一个主题  $z$ ，按主题  $z$  的单词概率分布生成一个单词。其图模型如下图所示：



从上图可知 LDA 的联合概率为：

$$p(\theta, \mathbf{z}, \mathbf{w} | \alpha, \beta) = p(\theta | \alpha) \prod_{n=1}^N p(z_n | \theta) p(w_n | z_n, \beta)$$

把上面的式子对应到图上，可以大致按下图理解：



从上图可以看出，LDA 的三个表示层被三种颜色表示出来：

1. corpus-level（红色）： $\alpha$  和  $\beta$  表示语料级别的参数，也就是每个文档都一样，因此生成过程只采样一次。
2. document-level（橙色）： $\theta$  是文档级别的变量，每个文档对应一个  $\theta$ ，也就是每个文档产生各个主题  $z$  的概率是不同的，所有生成每个文档采样一次  $\theta$ 。
3. word-level（绿色）： $z$  和  $w$  都是单词级别变量， $z$  由  $\theta$  生成， $w$  由  $z$  和  $\beta$  共同生成，一个单词  $w$  对应一个主题  $z$ 。

通过上面对 LDA 生成模型的讨论，可以知道 **LDA 模型主要是从给定的输入语料中学习训练两个控制参数  $\alpha$  和  $\beta$** ，学习出了这两个控制参数就确定了模型，便可以用来生成文档。其中  $\alpha$  和  $\beta$  分别对应以下各个信息：

**$\alpha$ :** 分布  $p(\theta)$  需要一个向量参数，即 Dirichlet 分布的参数，用于生成一个主题  $\theta$  向量；

**$\beta$ :** 各个主题对应的单词概率分布矩阵  $p(w|z)$ 。

把  $w$  当做观察变量， $\theta$  和  $z$  当做隐藏变量，就可以通过 EM 算法学习出  $\alpha$  和  $\beta$ ，求解过程中遇到后验概率  $p(\theta, z|w)$  无法直接求解，需要找一个似然函数下界来近似求解，原文使用基于分解（factorization）假设的变分法（variational inference）进行计算，用到了 EM 算法。每次 E-step 输入  $\alpha$  和  $\beta$ ，计算似然函数，M-step 最大化这个似然函数，算出  $\alpha$  和  $\beta$ ，不断迭代直到收敛。