

# 图像技术在直播中的应用（上）——美颜技术

2016年作为视频直播元年，无论从资本层面不断高涨的估值，到平台主播各种天文数字的报酬，再到像“局座”这样的主流人士争相上直播，直播的社会热度可见一斑。而各大直播平台在经历了直播概念从无到有的阶段后，如何做出差异化，如何解决在野蛮生长期产生的各种涉黄问题，成为了几乎所有平台“成长的烦恼”。抛开政策、内容问题不说，单就技术层面，基于图像技术的创新成为解决这些难题最为可行的办法。作为Tu料的首个分享，我们首先就选择了直播这个热门话题，就图像技术在直播中的应用做一个入门的介绍。该分享系列整理自涂图CTO在架构师沙龙上的演讲内容。

## 一、技术框架

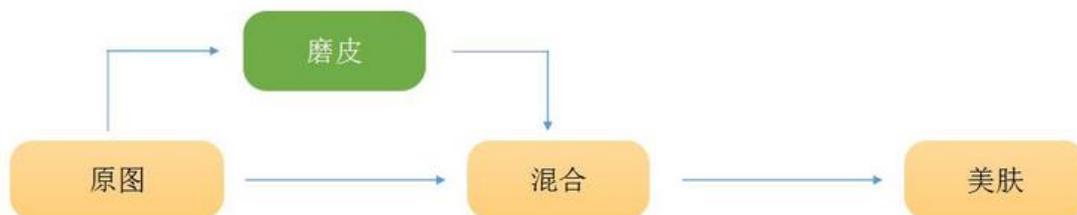
爱美之心，人皆有之。早期的图片美颜教育了市场，到了直播时代，美颜同样成为直播平台的标配。就目前来说，直播美颜用的主流技术是OpenGL ES。它的好处首先是直接在GPU上运行的，所以性能高、功耗小，用在直播上比较划算。第二，它是跨平台的，iOS和安卓都支持，美颜效果能够直接在这两个平台上达到跨平台的效果。另外OpenGL ES有一个优势，就是有大量现成的开源库。比如像GPUImage，谷歌的grafika，还有基于安卓的一些具有实践意义的库，都是很常用的。网上也有一些比较热心的开发者把自己的美颜的算法直接开源，包括一整套解决方案，从采集到处理到美颜处理，到最后输出一个源编码，都有相关的解决方案。

- ◆ 跨平台（Android、iOS）
- ◆ 性能高，功耗小
- ◆ 现成的开源库  
GPUImage、Google grafika .....

## 二、美颜原理——混合不能少

市面上大部分美颜产品的一般原理都是相似的。摄像头采集画面，通过一定的方式处理，最后输出一张美颜后的图片。

具体的说，原图首先经过磨皮处理，也就是把痘、斑这些消除掉。然后把经过磨皮的图片与原图进行混合。混合这个步骤是不可缺少的，因为如果只用磨皮后的图，很容易丢失细节。而且把两张图混合，还可以通过调整两个图的混合权重，来控制磨皮的程度，达成不同级别的磨皮效果。当然最后一步也很关键，就是美肤，比如把皮肤色调得白一点、红嫩一点，或者一些特殊的需求都可以实现。基本上大部分的美颜，都是这样的流程。



### 三、磨皮算法——去痘就是降噪

本质上说，一张图就是一个二维的数据。如果相临的两个区域灰度值相差比较大，这就意味着存在着噪点。比如脸上有个痘，这个痘自然产生一个灰度值的变化，从抽象意义上说就是一个噪点。所以美颜磨皮的算法，核心是去噪。去噪有很多的办法，网上有各种各样的算法，包括现成的论文。但是不管产用什么算法，美颜的去噪算法都要保持一个特点，那就既要是边界保持，同时还要做到平滑，也就是要滤波。

滤波算法中比较常见的是双边滤波，它的优点在于很高效，因此非常适合移动平台。还有一些其他的算法比较复杂一点，也可以达到那样的效果，但是在移动应用上效率不高。虽然GPU是并行运算的，很适合这种运算，但GPU的能力是有一定范围的，超过这个范围也很耗电。包括双边滤波在内，一种滤波算法也有各种各样实现，考虑在移动平台上运行，可以做一下特别的优化，比如在精度计算里，适当降低精度，达到效果与效率的平衡。

磨皮的核心是去噪，即平滑噪点和周围像素的亮度差

边界保持类平滑滤波算法

- 双边滤波 (高效算法, 适合移动平台)
- K近邻
- ...



灰度图

### 四、肤色调整——检测才是难题

在做好磨皮后，最后一个流程就是肤色调整。调整肤色本身的手法已经非常成熟，所以更难的其实在于肤色检测。为什么要做肤色检测？有些早期的直播美颜是没有这个功能的，所以他们就是以美化的肤色为基础，简单粗暴的把整个图像按这个色彩全部处理掉了，这就造成了整体偏色，效果还不如不做。因此，在处理图像前，必须先进行肤色检测，在图像的所有像素点中找到对应肤色范围的像素再进行处理。



直播中的肤色检测特殊的地方在于颜色空间的转换。因为跟图像处理相关的颜色空间主要有三种：RGB、YUV、HSV，而这三种色彩空间在直播中都要用到。

RGB是最常见的色彩空间，我们日常用的显示设备就是基于RGB空间，这里不多解释。

YUV是一种比较传统的颜色空间，最早是应用在电视信号的传播里面的，目前多用在直播的数据采样、传输的过程。这是因为人眼对亮度 (Y) 远比色度(U、V)更敏感，所以YUV比起RGB更容易被压缩，这样就更容易节省带宽进行传输。

而基于HSV颜色空间才是用来做肤色检测的。因为如果用RGB来做肤色检测，需要检测R、G、B三个值是否同时满足肤色的颜色范围，YUV同理。而HSV三个值：色调 (H)、饱和度 (S)，明度 (V) 中，只有H是关乎肤色的，因此只需要对H进行考虑 (H值在25-50之间即可判断为肤色)，所需要的运算量自然比RGB少很多。

所以，在直播的不同阶段，要分别使用这三种色彩空间，要不停的把这三种色彩空间进行相互转换。

## 五、细节——在算法之外

美颜算法固然重要，但美更是一件很主观的事。算法写得非常优美和高效并不能保证美颜效果是最好的，所以用标准的算法处理后，还需要设计师依据自己的经验去进行调整。比如很多平台算法都大同小异，但是为什么最终出来的美颜效果让人感觉还是有差异，其实就是说里面有很多细节在，需要花时间优化，特别是用户的需求是什么，怎样更漂亮。

再举一个例子。很多平台在不同的光照条件下，比如白天、晚上，室内、室外，自然光、人工光，直播出来的美颜效果差异很大。这其中的原因可能就是算法中没有考虑光照因素，结果使很小的因素影响了效果。

所以，这就要求进行大量的测试，用技术手段结合人工去优化，才能保证最佳的美颜效果。还是那句话：细节是魔鬼。

## 六、性能——不服跑个分？

说到性能，iOS平台目前一般来说没什么问题，或者问题很少。比如 GPUImage是第三方的，算是iOS平台上很有历史的一个库。它实现了很多效果，比如刚才提到的一些算法，在GPUImage里面可以看到简单版的实现，包括怎么写脚本，怎么跑起来，怎么做双边滤波，里面有简单的实现，也可以有很好的效果。包括在做直播的时候，GPUImage可以作为很好的客户端扩展，唯一需要做的事情，就是加上一个推流；因为它包含的从客户端的采集、处理到每一帧的数据，无论是YUV，还是RGB，都可以输出来。所以iOS平台上相对来说问题少很多。

而安卓平台问题就比较大了。因为安卓本身的特点，厂商很多、设备很多、系统版本很多，因此相互间比较难兼容。

第一个是设备问题。比如一个美颜算法在不同机器上跑起来，即使是同一款GPU，性能也可能会差别很大。所以，为了保证一个脚本去适应不同的机器，有种办法是这样的：根据GPU的性能做了一个分级算法，如果评级比较高就采用最复杂的算法，如果性能评级比较低，就把美颜效果降低，保证在大部分环境下使用。

第二是版本的问题。比如，只有4.0以上版本才能够通过相机，直接从相机采集里面获取到一个纹理，这个叫GLTEXTUREEXTERNAL\_OES，相机直接把采集到的画面转给GPU，全部通过GPU加速。4.3以后可以做什么事情？从相机采集到处理到编码，走的是全GPU，这是效果最好的也是最快的，当然这对系统的兼容要求是最高的。因为有些厂商在实现的时候没有兼容这些东西，所以GPU加速很难做。

还有输出YUV。很多直播平台都要支持输出YUV。这些YUV的数据就涉及到CPU和GPU的转换过程。因为处理可能是在GPU里面完成的，GPU没法直接输出来，就需要从GPU到CPU的转换，这个目前来说还没有比较好的方案。安卓目前一些底层的GPU还没有开放，有的时候可以通过Graphics Buffer来实现，但是安卓并没有把这个开放出来。如果要有这个东西，唯一能做的是把安卓源代码拿出来，包括把源代码link到关键码里面去，这样才能达到比较好的效果。CPU到GPU的转换，是可以毫秒级别的，如果直接从GPU转换到CPU，可能好的设备也会花费20毫秒左右，这样导致的数据，你预测的是24帧，可能就会有掉帧的情况。在主流上可能影响不大，大部分情况下都是可以接受的，当然这最后也要看用户只有应用场景。

最后说说常被问到的一个问题：iOS平台、安卓平台都自带有人脸检测的API，为什么不用？

首先是系统频率低、速度慢。苹果可能有这样一个考虑，不要影响到相机API的正常使用，所以频率很低。检测一次可能3秒钟；不是检测一次需要3秒钟，而是3秒钟才给你一个数据，告诉你这张照片有没有人脸。而作为一个实用的产品，一秒钟假设24帧，起码要做十几次检测才可以达到实时的要求，要不然的话跟不上对帧率的要求。在安卓问题更严重，因为还要看设备，有些设备甚至就没有，厂商把这个设置就直接去掉了。另外一个特征点的问题。iOS上面有这些特征点，比如眼睛、嘴巴、鼻子这些，安卓上面是直接没有这些特征点的。

## 七、美颜2.0——从化妆到整容

以上的内容都属于美颜1.0的概念，而目前最新的美颜技术已经发展到了2.0概念。打个简单的比方，如果美颜1.0只是化妆的话，美颜2.0基本就能达到整容的效果——把眼睛变大，把圆脸变成瓜子脸。而实现这一效果的基础就是人脸识别。这很好理解，只有确定了有没有人脸，知道了五官在什么位置，我们才能把它们“整”的更漂亮。

关于人脸识别，就又是一个大问题了。限于篇幅，我们会在下一期详细展开这个问题说。