

# 在iOS中如何正确的实现行间距与行高

2018-03-27 16:42:00

最近准备给 [VirtualView-iOS](#) 的文本元素新增一个 `lineHeight` 属性，以便和 [VirtualView-Android](#) 配合时能更精确的保证双平台的一致性。面向 Google 以及 Stack Overflow 编程了一会后发现，能查到的资料大部分是介绍如何实现 `lineSpacing` 属性，而不是 `lineHeight`。但是我就是因为 iOS 和 Android 的默认 `lineSpacing` 不一致所以才想实现个 `lineHeight` 啊！还是需要自己动手丰衣足食，顺带整理成文章造福后人。

## 关于行间距 `lineSpacing`

先贴出一张 iOS 中 UILabel 的默认排版样式：

MVVM是微软于2005年开发出的一种软件架构设计模式，主要是为了在WPF和Silverlight中更简单的对UI实现事件驱动编程。在WPF和Silverlight中，通过MVVM成功的实现了UI布局和数据逻辑的剥离。虽然WPF和Silverlight最后都没有推广开来，但是还是让大家看到了MVVM设计模式的优秀之处。

大家也都能看出来，默认的排版样式中，文本的行间距很小，显得文本十分挤。

这种时候，设计师就会提出行间距的需求，希望让文本展示得更美观。类似的标注就会像这样：

10pt

MVVM是微软于2005年开发出的一种软件架构设计模式，主要是为了在WPF和Silverlight中更简单的对UI实现事件驱动编程。在WPF和Silverlight中，通过MVVM成功的实现了UI布局和数据逻辑的剥离。虽然WPF和Silverlight最后都没有推广开来，但是还是让大家看到了MVVM设计模式的优秀之处。

通常来说既然设计师要求的是行间距，那么我们直接设置lineSpacing 就好。但是 UILabel 是没有这么一个直接暴露的属性的，想要修改 lineSpacing，我们需要借助 NSAttributedString 来实现，示意代码：

```
NSMutableParagraphStyle *paragraphStyle = [NSMutableParagraphStyle paragraphStyle];
paragraphStyle.lineSpacing = 10;
NSMutableDictionary *attributes = [NSMutableDictionary dictionary];
[attributes setObject:paragraphStyle forKey:NSParagraphStyle];
label.attributedText = [[NSAttributedString alloc] initWith
```

运行一下观察效果：

MVVM是微软于2005年开发出的一种软件架构设计模式，主要是为了在WPF和Silverlight中更简单的对UI实现事件驱动编程。在WPF和Silverlight中，通过MVVM成功的实现了UI布局和数据逻辑的剥离。虽然WPF和Silverlight最后都没有推广开来，但是还是让大家看到了MVVM设计模式的优秀之处。

虽然用我们的眼睛看上去好像没什么问题，但是设计师的火眼金睛一下就能看出来，和设计稿要求的有差距：

MVVM是微软于2005年开发出的一种软件架构设计模式，主要是为了在WPF和Silverlight中更简单的对UI实现事件驱动编程。在WPF和Silverlight中，通过MVVM成功的实现了UI布局和数据逻辑的剥离。虽然WPF和Silverlight最后都没有推广开来，但是还是让大家看到了MVVM设计模式的优秀之处。

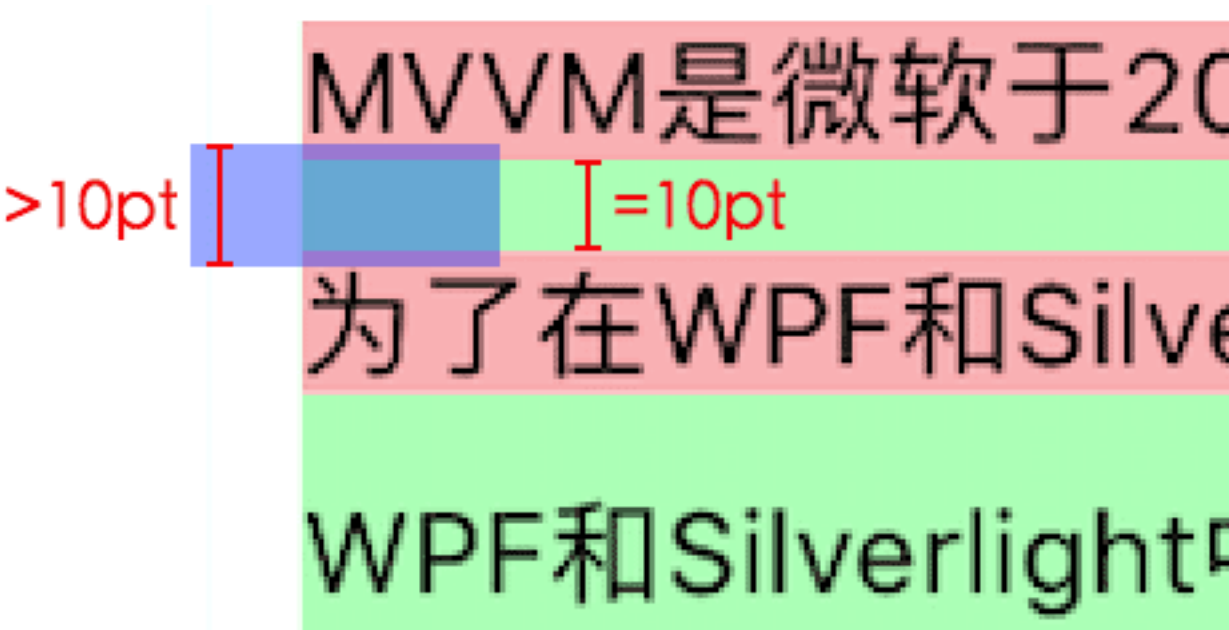
设计图（理想）

我们做出来的（现实）

怎么会成这样！？ 这跟说好的不一样对不对！？ 不要慌，我来细细解释下。

## 正确的实现行间距

先看示意图：



红色区域是默认绘制单行文本会占用的区域，可以看到文字的上下是有一些留白的（蓝色和红色重叠的部分）。设计师是想要蓝色区域高度为 10pt，而我们直接设置 lineSpacing 会将两行红色区域中间的绿色区域高度设置为 10pt，这就是问题的根源了。

那么这个红色的区域高度是多少呢？ 答案是

`label.font.lineHeight`，它是使用指定字体绘制单行文本的原始行高。

知道了原因后问题就好解决了，我们需要在设置 `lineSpacing` 时，减去这个系统的自带边距：

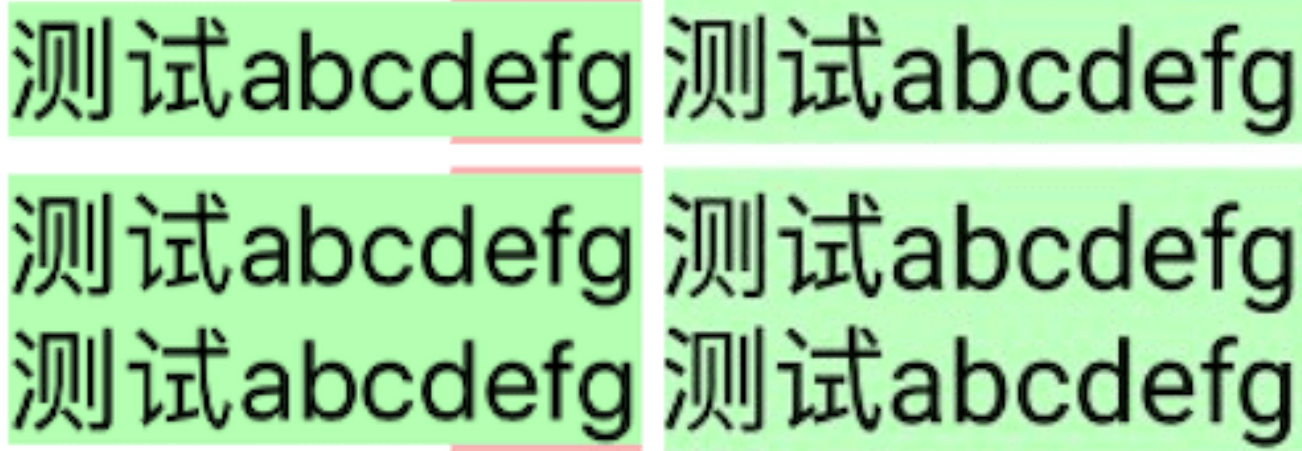
```
NSMutableParagraphStyle *paragraphStyle = [NSMutableParagraphStyle new];
paragraphStyle.lineSpacing = 10 - (label.font.lineHeight - [UIFont systemFontOfSize:14]);
NSMutableDictionary *attributes = [NSMutableDictionary dictionary];
[attributes setObject:paragraphStyle forKey:NSParagraphStyleKey];
label.attributedText = [[NSAttributedString alloc] initWithString:@"MVVM" attributes:attributes];
```

观察一下效果，完美契合：

MVVM是微软于2005年开发出的一种软件架构设计模式，主要是为了在WPF和Silverlight中更简单的对UI实现事件驱动编程。在WPF和Silverlight中，通过MVVM成功的实现了UI布局和数据逻辑的剥离。虽然WPF和Silverlight最后都没有推广开来，但是还是让大家看到了MVVM设计模式的优秀之处。

## 关于行高 `lineHeight`

如果你只关心 iOS 设备上的文本展示效果，那么看到这里就已经够了。但是我需要的是 iOS 和 Android 展现出一模一样的效果，所以光有行间距是不能满足需求的。主要的原因在前言也提到了，Android 设备上的文字上下默认留白（上一节图中蓝色和红色重叠的部分）和 iOS 设备上的是不一致的：



左侧是 iOS 设备，右侧 Android 设备，可以看到同样是显示 20 号的字体，安卓的行高会偏高一些。在不同的 Android 设备上使用的字体不一样，可能还会出现更多的差别。如果不想办法抹平这差别，就不能真正意义上实现双端一致了。

这时候我们可以通过设置 `lineHeight` 来使得每一行文本的高度一致，`lineHeight` 设置为 30pt 的情况下，一行文本高度一定是 30pt，两行文本高度一定是 60pt。虽然文字的渲染上会有细微的差别，但是布局上的差别将被完全的抹除。`lineHeight` 同样可以借助 `NSAttributedString` 来实现，示意代码：

```
NSMutableParagraphStyle *paragraphStyle = [NSMutableParagraphStyle new];
paragraphStyle.maximumLineHeight = lineHeight;
paragraphStyle.minimumLineHeight = lineHeight;
NSMutableDictionary *attributes = [NSMutableDictionary dictionary];
[attributes setObject:paragraphStyle forKey:NSParagraphStyleKey];
label.attributedText = [[NSAttributedString alloc] initWith
```

运行一下观察效果：



MVVM是微软于2005年开发出的一种软件架构设计模式，主要是为了在WPF和Silverlight中更简单的对UI实现事件驱动编程。在WPF和Silverlight中，通过MVVM成功的实现了UI布局和数据逻辑的剥离。虽然WPF和Silverlight最后都没有推广开来，但是还是让大家看到了MVVM设计模式的优秀之处。

在 debug 模式下确认了下文本的高度的确正确的，但是为什么文字都显示在了行底呢？

## 修正行高增加后文字的位置

修正文字在行中展示的位置，我们可以用 [baselineOffset](#) 属性来搞定。这个属性十分有用，在实现上标下标之类的需求时也经常用到它。经过调试，发现最合适的值是  $(lineHeight - label.font.lineHeight) / 4$ （尚未搞清楚为什么是除以 4 而不是除以 2，希望知道的老司机指点一二）。最终的代码示例如下：

```
NSMutableParagraphStyle *paragraphStyle = [NSMutableParagraphStyle new];
paragraphStyle.maximumLineHeight = lineHeight;
paragraphStyle.minimumLineHeight = lineHeight;
NSMutableDictionary *attributes = [NSMutableDictionary dictionary];
[attributes setObject:paragraphStyle forKey:NSParagraphStyle];
CGFloat baselineOffset = (lineHeight - label.font.lineHeight) / 4;
[attributes setObject:@(baselineOffset) forKey:NSBaselineOffset];
label.attributedText = [[NSAttributedString alloc] initWith
```

贴一下在不同字号和行高下的展示效果：

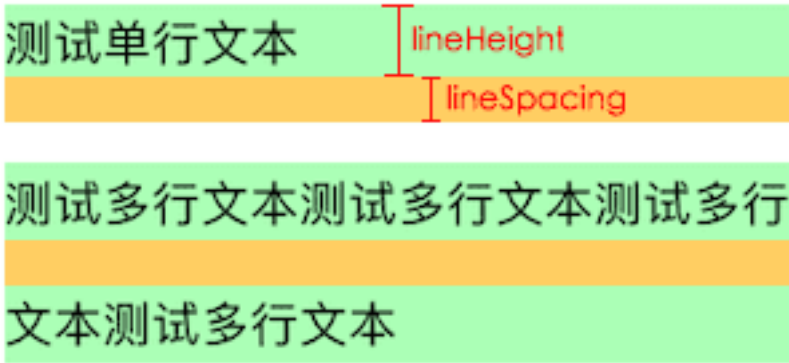
MVVM是微软于2005年开发出的一种软件架构设计模式，主要是为了在WPF和Silverlight中更简单的对UI实现事件驱动编程。

MVVM是微软于2005年开发出的一种软件架构设计模式，主要是为了

MVVM是微软于2005年开发出的一种软件架构设计模式，主要是为了

## 行高和行间距同时使用时的一个问题

不得不说行高和行间距我们都已经可以完美的实现了，但是我在尝试同时使用它们时，发现了 iOS 的一个 bug（当然也可能是一个 feature，毕竟不 crash 都不一定是 bug）：



着色的区域都是文本的绘制区域，其中看上去是橙色的区域是 lineSpacing，绿色的区域是 lineHeight。但是为什么单行的文本系统也要展示一个 lineSpacing 啊！？坑爹呢这是！？

好在我们通常是行高和行间距针对不同的需求分别独立使

用的，它们在分开使用时不会触发这个问题。所以在 [VirtualView-iOS](#) 库中，我暂且将高度计算的逻辑保持和系统一致了。

## 总结

至此，成功的为 [VirtualView-iOS](#) 添加了对 lineHeight 属性的支持，更多的实现细节大家可以到开源库中直接看源代码。希望我们的 Tangram 方案可以更加完善，帮助更多的人一次开发两端同时使用，用一块七巧板拼出大千世界。