



Core Animation:

Simplified Animation Techniques for

Mac and iPhone Development

第一部分

核心动画开篇

第二章

我们可以和应该做哪些动画？

版本 1.0

翻译时间：2012-09-24

DevDiv 翻译：animeng

DevDiv 校对：symbian_love BeyondVincent (破船)

DevDiv 编辑：BeyondVincent (破船)

写在前面

目前，移动开发被广大的开发者们看好，并大量的加入移动领域的开发。

鉴于以下原因：

- 国内的相关中文资料缺乏
- 许多开发者对 E 文很是感冒
- 电子版的文档利于技术传播和交流

[DevDiv.com](http://www.devdiv.com) [移动开发论坛](#) 特此成立了翻译组，翻译组成员具有丰富的移动开发经验和英语翻译水平。组员们利用业余时间，把一些好的相关英文资料翻译成中文，为广大移动开发者尽一点绵薄之力，希望能对读者有些许作用，在此也感谢组员们的辛勤付出。

关于 DevDiv

DevDiv 已成长为国内最具人气的综合性移动开发社区

更多相关信息请访问 [DevDiv 移动开发论坛](#)。

技术支持

首先 DevDiv 翻译组对您能够阅读本文以及关注 DevDiv 表示由衷的感谢。

在您学习和开发过程中，或多或少会遇到一些问题。DevDiv 论坛集结了一流的移动专家，我们很乐意与您一起探讨移动开发。如果您有什么问题和技術需要支持的话，请访问网站 www.devdiv.com 或者发送邮件到 BeyondVincent@DevDiv.com，我们将尽力所能及的帮助您。

关于本文的翻译

感谢 animeng 对本文的翻译，同时非常感谢 symbian_love 和 BeyondVincent(破船)在百忙中抽出时间对翻译初稿的认真校验。才使本文与读者尽快见面。由于书稿内容多，我们的知识有限，尽管我们进行了细心的检查，但是还是会存在错误，这里恳请广大读者批评指正，并发送邮件至 BeyondVincent@devdiv.com，在此我们表示衷心的感谢。

推荐资源

iOS

[iOS 5 Programming Cookbook 中文翻译各章节汇总](#)

[iOS6 新特征：参考资料和示例汇总](#)

Android

[DEVDIV 原创 ANDROID 学习系列教程实例](#)

Windows Phone

[Windows Phone 8 新特征讲义与示例汇总](#)

Windows 8

[Building Windows 8 apps with XAML and C#中文翻译全部汇总](#)

[Building Windows 8 apps with HTML5 and JavaScript 中文翻译汇总](#)

[Windows 8 Metro 开发书籍汇总](#)

[Windows 8 Metro App 开发 Step by Step](#)

其它

[DevDiv 出版作品汇总](#)

目录

写在前面	2
关于 DevDiv	2
技术支持	2
关于本文的翻译	2
推荐资源	3
目录	4
本书翻译贴各章汇总	5
Core Animation 中文翻译各章节汇总	5
第一部分 核心动画开篇	5
第 2 章 我们可以和应该做哪些动画？	6
2.1. 那些属性可以让我们做动画呢？	6
2.1.1. 颜色	6
2.1.2. 动作	7
2.1.3. 特征	7
2.1.4. 可视度	7
2.1.5. 滤镜	8
2.1.6. 内容	8
2.1.7. 遮罩	9
2.2. 我们应该使哪些属性激活呢？	10
2.2.1. 设计法则	10
2.2.2. 什么时候我们应该使用核心动画呢？	11
2.2.3. 哪些我们不应该做动画	11
2.3. 总结	11

本书翻译贴各章汇总

Core Animation 中文翻译各章节汇总

第一部分 核心动画开篇

第一章 什么是核心动画

DevDiv 翻译

第 2 章 我们可以和应该做哪些动画？

核心动画提供了大量的属性，你可以用这些属性在核心动画的层、cocoa 的窗口和视图上面做动画。在你的工具包中，核心动画不应该在你的应用程序的每处都使用。这章就是讲你可以使用的属性以及讨论哪些属性你应该使用。

核心动画是一个伟大的技术，但是每个伟大的技术都不能被滥用。这里有一些建议，可以避免你进入过度使用的误区。你会感激你采取了这个建议，用户也会感谢的。

2.1. 那些属性可以让我们做动画呢？

动画是要应用到可视的东西上：窗口，视图，和层。除非有些可视的东西要被改变，否则动画没有任何意义。这就是为什么虽然你使用了一个层，这个层也包含了这些动画属性的键值，你也不能激活这些动画属性。这些动画属性只有根据核心动画指定改变时，才能提供视觉的反馈。

这个章节的段落，罗列了一些条目，这些条目可以帮助你更好的理解，哪些动画属性对你来说是可用的。

2.1.1. 颜色

核心动画使用了一个内部的算法来计算颜色的改变。当你在 `CABasicAnimation` 中指定一个 `fromValue` 和一个 `toValue`，或者你提供给 `CAKeyframeAnimation` 一个颜色数组，数组中的值为 `CGColorRef`，那么核心动画就在动画期间，在视图上把所有的颜色都填充。

列表 2-1 演示了一个基本的动画，如何在 5 秒内改变背景颜色从红色到绿色。

```
- (CABasicAnimation*)backgroundColorAnimation;
{
    CABasicAnimation *anim =
    [CABasicAnimation
    animationWithKeyPath:@"backgroundColor"];
    [anim setDuration:5.0];
    CGColorRef red =
    CGColorCreateGenericRGB(1.0, 0.0, 0.0, 1.0);
    CGColorRef green =
    CGColorCreateGenericRGB(0.0, 1.0, 0.0, 1.0);
    [anim setFromValue:(id)red];
    [anim setToValue:(id)green];
    CFRelease(red);
    CFRelease(green);
    return anim;
}
```

表 2-1

这个示例代码创建了一个基础的动画对象 `CABasicAnimation`，使用了关键路径 `backgroundColor`，并且通过 `setFromValue` 和 `setToValue` 为它设定了参数。通过 `setDuration: 5.0` 设置了执行时间为 5 秒。当这个动画增加到层中时，层的背景颜色就立刻开始了动画。

在 iPhone 下颜色的改变是通过 `UIColor` 类来操控的。列表 2-2 演示了在 iPhone 下如何创建同样的动画。

```
- (CABasicAnimation*)backgroundColorAnimation;
{
    CABasicAnimation *anim =
    [CABasicAnimation
    animationWithKeyPath:@"backgroundColor"];
    [anim setDuration:5.0];
    [anim setFromValue:(id)[[UIColor redColor] CGColor]];
}
```

```
[anim setToValue:(id)[UIColor greenColor] CGColor];  
return anim;  
}
```

列表 2-2

你可以做动画的颜色属性有

backgroundColor, 层的背景颜色

borderColor, 层的边框颜色

shadowColor, 层的阴影颜色, 目前, shadowColor 这个属性在 iphone 上不可用。

2.1.2. 动作

当你在屏幕上移动时, 核心动画为每个可以做动画的层上, 提供了矩阵和点。你可以移动的属性有 anchorPoint

默认的锚点是 0.5,0.5, 表示在层的中间。如果你改变锚点为 0.0,0.0, 那么层的锚点就移动到了左上角。

Bounds

边框属性只能使边框矩阵的形状做动画, 但是不会影响到层的位置。

Frame

框架的属性在层中不能够做动画。你可以设定这个参数和层内部改变的值, 然而你却不能看到真实的动画。如果你想使层的边框做动画, 你需要使用 bounds 属性改变边框的矩阵。如果你想改变位置, 使用 position 这个属性。在视图和窗口上, 设定 frame 会引起边框的动画。

Position

使用 position 属性来移动一个层的位置。Position 属性依赖于 anchorPoint 属性。如果你想在父视图中居中一个层, 那么你需要把层的 anchorPoint 设定为默认的 0.5,0.5, 然后计算父视图的中心位置。表 2-3 演示了怎么去居中一个层。

```
CGRect parentViewRect = [[window contentView] frame];  
[layer setPosition:  
CGPointMake(parentView.size.width/2.0,  
parentView.size.height/2)];
```

表 2-3

zPosition

zPosition 控制层的 z 轴。这决定了层从前到后的位置。它能做这样的动画, 在一个层的顶部转换出一个层。

2.1.3. 特征

一个层每一个可视的特性都能做动画。这些是被称为 attributes, 因为他们是可以影响层可视属性的字段。你能做动画的属性字段有

borderWidth

borderWidth 属性可以设定层周围的边框宽度。

cornerRadius

cornerRadius 属性可以设定一个层的圆角。

2.1.4. 可视度

层是一个可视的组件, 但是你可以决定怎么可视。有时, 你需要全部或者部分的看到, 这就需要你来控制这些可视字段。你可以做动画的可视字段如下:

Opacity

Opacity 属性决定层的透明度。1.0 代表层不透明, 0.0 代表全透明。使用这两个数值之间的值, 来设定你要显示的层的透明情况。

Hidden

理论上, hidden 属性是具有动画性质的。然而, 它仅仅是一个开关。如果你想要做透明的动画变化, 最

好使用 `opacity` 属性代替 `hidden` 属性。

2.1.5. 滤镜

当它和核心图像一起使用时，核心动画的层可以有复杂的滤镜效果，通过设定一个或者更多的字段来应用。核心图像滤镜在写本书之前，在 `iphone` 下是不可用的。你可以做动画的滤镜字段如下：

Filters

`Filters` 属性是一个滤镜数组，你可以直接应用到层的内容上。它影响整个层，包括边框和背景。

compositingFilter

不同于 `filters` 属性，`compositingFilter` 属性使用了一个单一的滤镜作为参数。

backgroundFilters

`backgroundFilters` 属性提供了一个唯一的方法，应用滤镜到层的背景上。

2.1.6. 内容

一个层的内容字段就是设定使用的图像，就是 `CGImageRef`。如果你使用 `CAKeyframeAnimation` 来动画这个字段，你可以使用很少的代码创建一个简单的幻灯片应用。当这个动画是被增加到层中时，每张图片会按照你在数组中指定的图片位置进行放映。你也能够改变放映的类型。我们在第六章“Layer Filters”会详细讲解转变。

你可以做动画的内容字段包括：

Contents

`Contents` 这个属性就是你要展示的图片对象 `CGImageRef`

contentsRect

想像下 `contentsRect` 作为层内容的可视区域。内容的矩形区域 (`x`, `y`, `width`, `height`) 也是绑定到层的框架上的。`contentRect` 是一个比例值，而不是屏幕上真实的像素点。默认的是 `0.0,0.0,1.0,1.0`。当你改变 `x` 时，例如，在 `0` 和 `1` 之间你设定为 `0.25`，那么内容的观察窗口就展示在 `400` 像素中的 `100` 像素的位置 (`400 * 0.25`)，也就是一个层原始宽度的 `25%`。如果你设定了 `width` 属性为 `0.25`，那么观察窗口的宽度就是 `100`，也就是一个层原始宽度的 `25%`。图 2-1 和 2-2 演示了 `contentsRect` 如何展示原始的图像层的区域。

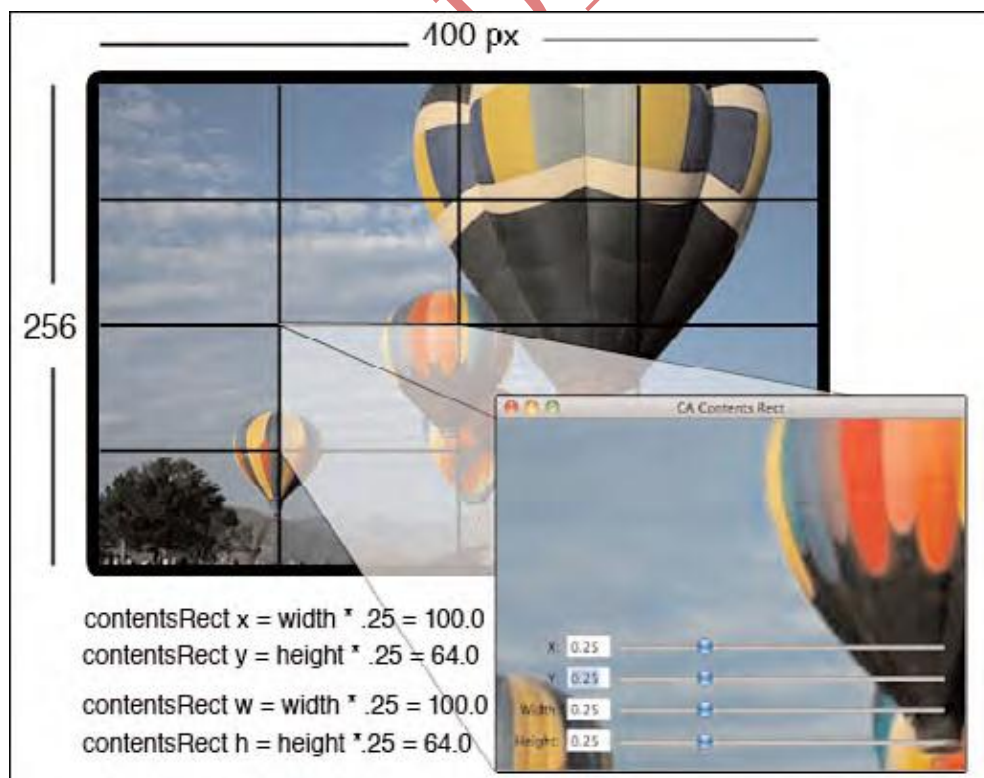


图 2-1

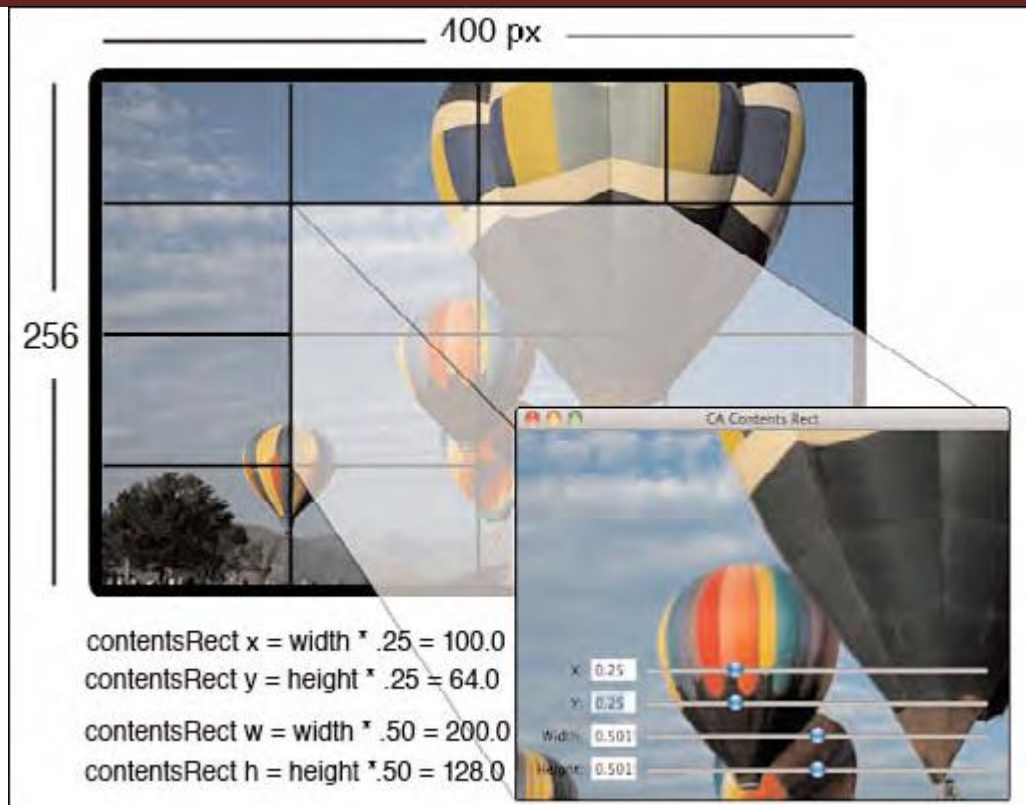


图 2-2

2.1.7. 遮罩

层提供了一个方法，用这个方法去遮罩内容图像。你可以控制的遮罩字段包括：

Mask

Mask 属性就是层本身。遮罩层的透明度决定了渲染的效果。当你指定一个遮罩时，记住遮罩的宽度决定了那些需要被展示，而非来自于视图。换句话说，如果你指定了遮罩层的宽和高为 300x300 像素，指定了遮罩的中心属性为窗口的中心。你会看到一个 300x300 的部分展示在窗口中心。300x300 区域周围的部分就被遮去了。如图 2-3



图 2-3

masksToBounds

`masksToBounds` 这个属性确保了内容是被裁剪到矩形中。当你设定圆角时，这个属性尤其有用。它确保了内容区域不会溢出到内容矩阵的外面。

2.2. 我们应该使哪些属性激活呢？

核心动画给了你一个强大和方便的能力，让你提升应用程序的用户体验。然而，在你使用核心动画的时候，你应该坚持一些法则。换句话说，你可以使某些东西做动画，并不意味着这是你应该让它做这个动画。在下面的段落里，我们谈论一些重要的规律，可以用来指导你的设计，哪些东西可以做动画，哪些不应该。

2.2.1. 设计法则

我们需要面对的是，开发者不是设计者。对于下面的陈述会有一些例外，但是大多数都会是一个规则。假如你是一个设计者，你可能会介意下面的声明，我会感谢你的大度原谅的。对于剩下，这都是一些有用的向导。

保持简单

这个建议你听了上千遍了，但是你仍然忽略它。不得不提醒开发者这个基本的规则是，并不因为你可以给如此小的 `iphone` 视图上，增加所有的控件来执行一些任务，就意味着你应该这样做。当你思考简单时，不是让你思考如何更简单的实现，而是要思考用户如何能更简单的使用。对用户保持简单而不是你。

不要设计，借鉴

You are likely in the nondesigner engineer camp as that label applies to the majority. 因此，你从来就不应该白手起家的设计一个用户体验。相反，你应该看一看你比较佩服的应用程序，然后借鉴你喜欢的创意。

聘用一个艺术家

如果你可以支付起一个艺术家的钱，聘用他。我曾经看到过一些令人惊讶的设计，他会让整个项目如此的完美，以至于没有一点可以挑剔的空间。得到一个恰当的设计是值得花这些钱的。你打算销售的应用程序，需要建立这些项目的预算。

通过书籍

当你开始设计的时候，你应该遵循苹果的用户设计向导（HIG）。这点对于 iPhone 开发者尤其重要，这里要是用其他的方法设计，不仅可能使应用看起来不好，而且也会给最终用户带来易用性的问题。例如，如果你尝试在视图的顶端，放置一个视图，而作为一个子视图。而不是通过导航控制器加载视图的。这样你将困难维护代码，并且应用程序的流程也会迷惑最终用户。相反，当你创建一个新视图，你应该像 iPhone 的用户设计向导建议的那样，使用视图控制器，利用导航堆栈把视图进行拉入和推出。

2.2.2. 什么时候我们应该使用核心动画呢？

在某些在线论坛中，人们经常这样问问题：“我听到了核心动画是我应该使用的，但是我不确定什么时候使用它，而什么时候应该使用标准的视图”。我这里这样总结这个答案。使用视图来处理用户交互的，而使用核心动画来处理设计元素的。有些情况可能要混合使用。

用户交互

如果你需要捕获用户的输入，使用标准的控件。这些控件在 Mac 上是基于 `NSView` 的，而在 iPhone 上是基于 `UIView`。如果你设计了某些特效，核心动画的层可以给你一个简单的路径来改变你的控件的外观，来满足你的设计，但是剩下的，就留给视图了。你的用户就会认识他们，自然也就知道怎么用了。

设计元素

如果你的应用程序要求一些动画去装饰视图，使用核心动画层。例如，如果你要在你的嘉年华的游戏里面旋转一个金光璀璨的背景，使用核心动画层。或者你想要一个图片在视图中做动画，装载图片并且设置层的内容字段来展示图片。然后显示的或者隐式的移动它的位置。

混合

有时候你想用层装饰你的视图，因为这样在视觉上提升了视图的效果。例如，当一些数据从网络上下载时，你想展示一个进度条，你可以在界面设计中（interface builder），设计这样的视图，然后放置一个进度条到视图中。然后你可以通过设定 `cornerRadius` 和 `borderwidth` 两个属性，使视图有一个边框和一个圆角的矩阵。

2.2.3. 哪些我们不应该做动画

可以说没有固定的规则，因此如果你已经看过了这些章节，但是仍然不确定那些应该做动画，你应该重温一下到目前为止讨论的规则。这些都可以引导你。

在我使用核心动画工作数月后，了解到了其中的各种细节，然而，确实没有一个绝对的规则，说那些动画不应该做。你可以给任何东西加入支持的动画。但你不应该因为加入了一些动画，使用户界面变的难以使用。事实上，不要在用户准备点击的控件或者超链接上做动画。当你让整个视图在窗口的外面或者里面动画，来引起注意，尤其在 iPhone 上是非常好的，但是你不要在屏幕上移动按钮和文本框。这是讨厌的，像是给用户一个恶作剧一样。但是，最后你需要决定在你的应用程序里，哪些动画不应该有。如果你不确定是否一个动画有意义，那么你就让一些人使用它，看他们的反应。这些将给你比本书更多的设计建议。

2.3. 总结

哪些可以做动画是受限于核心动画提供的东西。幸运的是，它提供了你需要的和你想做的任何事情。好吧，让我们面对哪些动画应该做吧。我希望你坚持我们讨论过的简单原则，但是我们相信你做的无论什么，因为你可以做出来既漂亮又实用的软件。

DevDiv 翻译



点击这里访问: DevDiv.com 移动开发论坛