任意状态(AnyState) & 混合树(Direct B Trees)

AnyState

无论AnyState切换的数量多少，AnyState的性能与非AnyState切换相似。这表明，转换检查的数量对帧时间的影响并不显著，这一点也在其他测试中得到了验证。

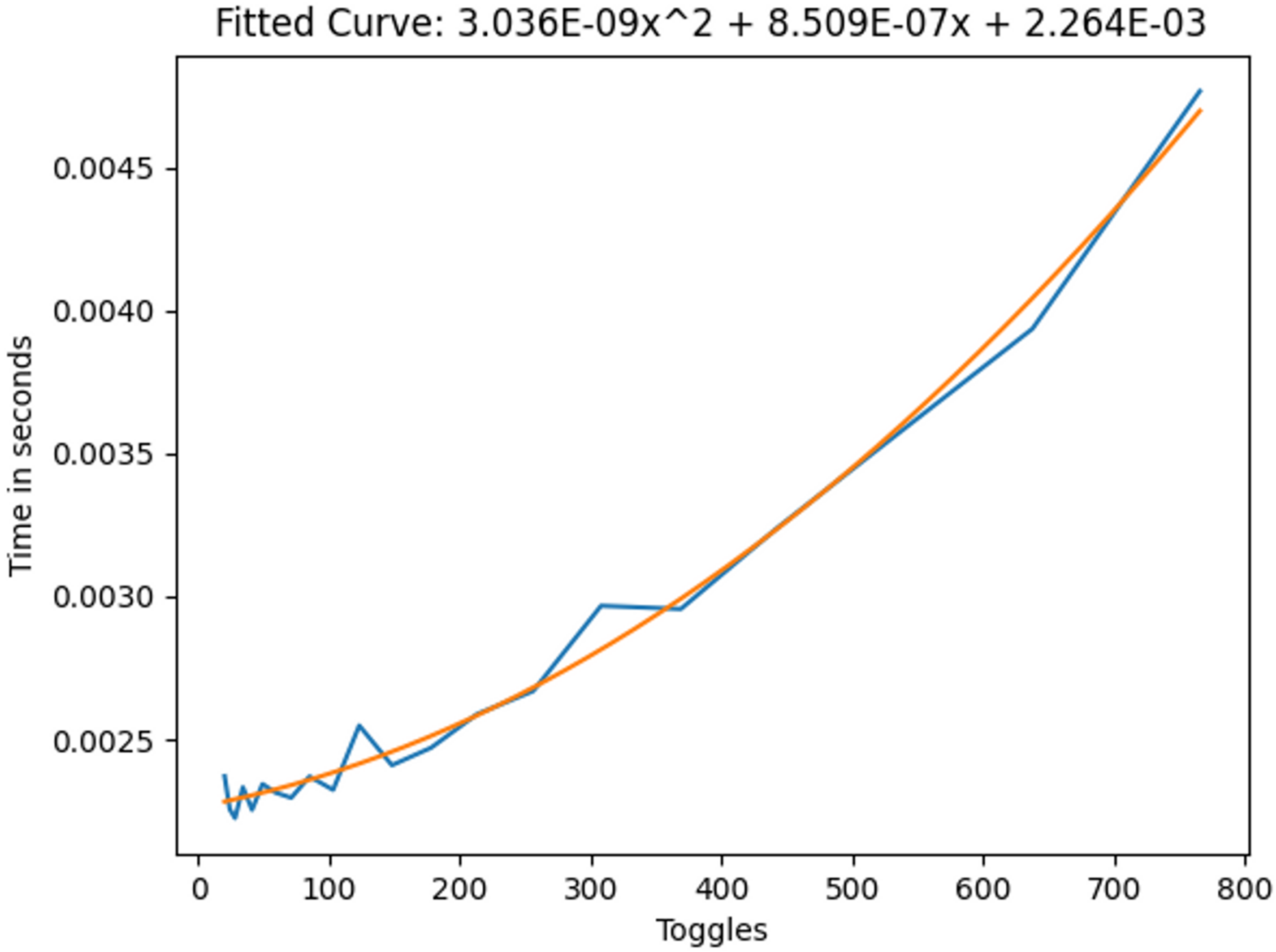
唯一的例外是启用了“可[illegible]20%[illegible]。

Direct Blend Trees

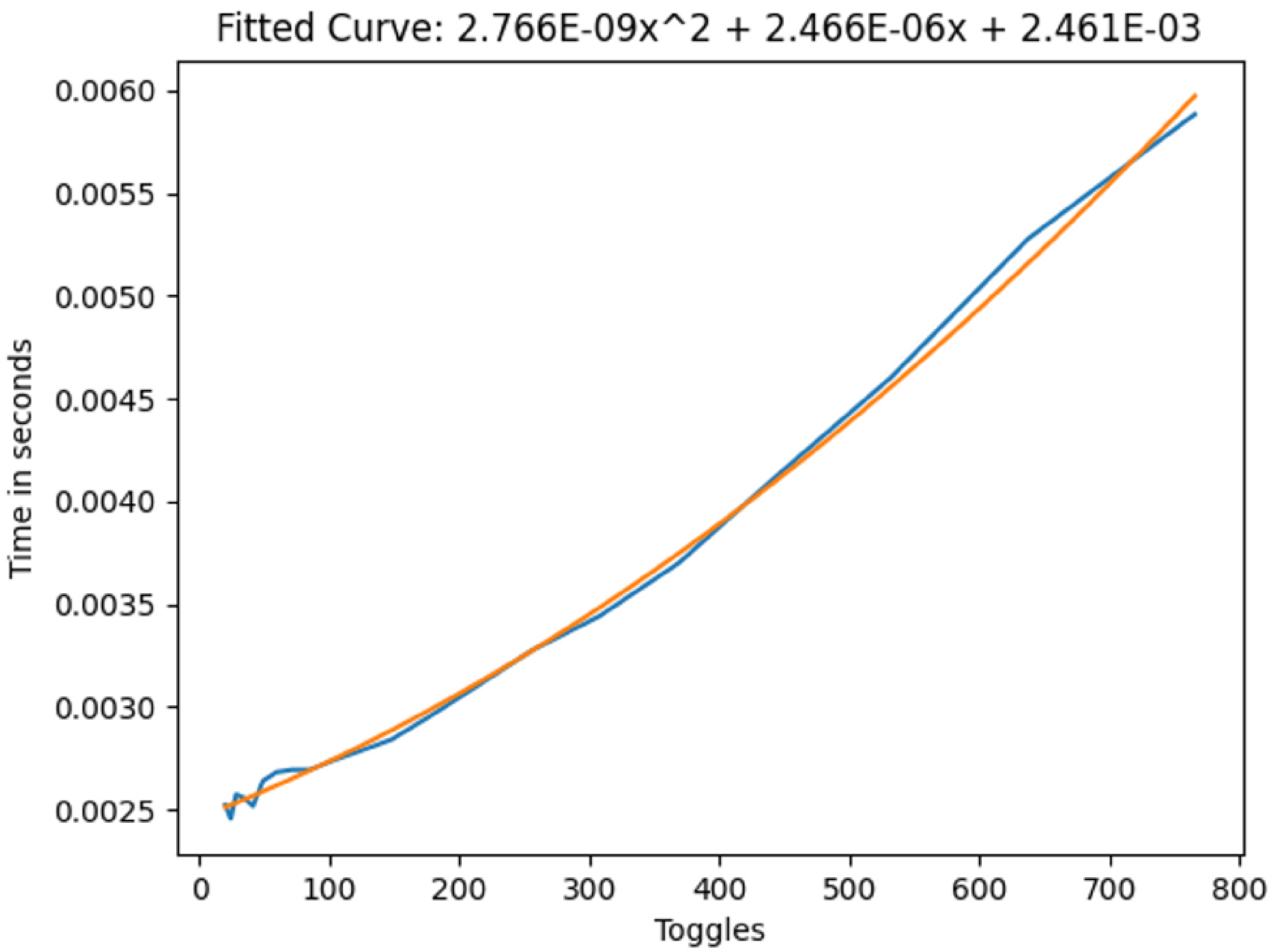
当首次发现动画层的巨大开销时，很多人曾希望直接Blend Trees能够作为一种魔法般的解决方案，大幅度降低帧时间。我的测试结果表明，虽然它们并不能完全消除帧时间，但确实在减少帧时间方面表现出色。

我发现了两种比普通动画层切换更高效的设置：一种较为简单，但略显迟缓；另一种设置较为复杂，但性能稍好。

对于较慢的设置，我使用了一个直接Blend Tree，包含多个1D Blend Trees作为子项。所有子项的权重均为1，而1D Blend Tree的混合值则根据切换进行调整。



Direct Blend Tree在未进行频繁切换时的帧时间



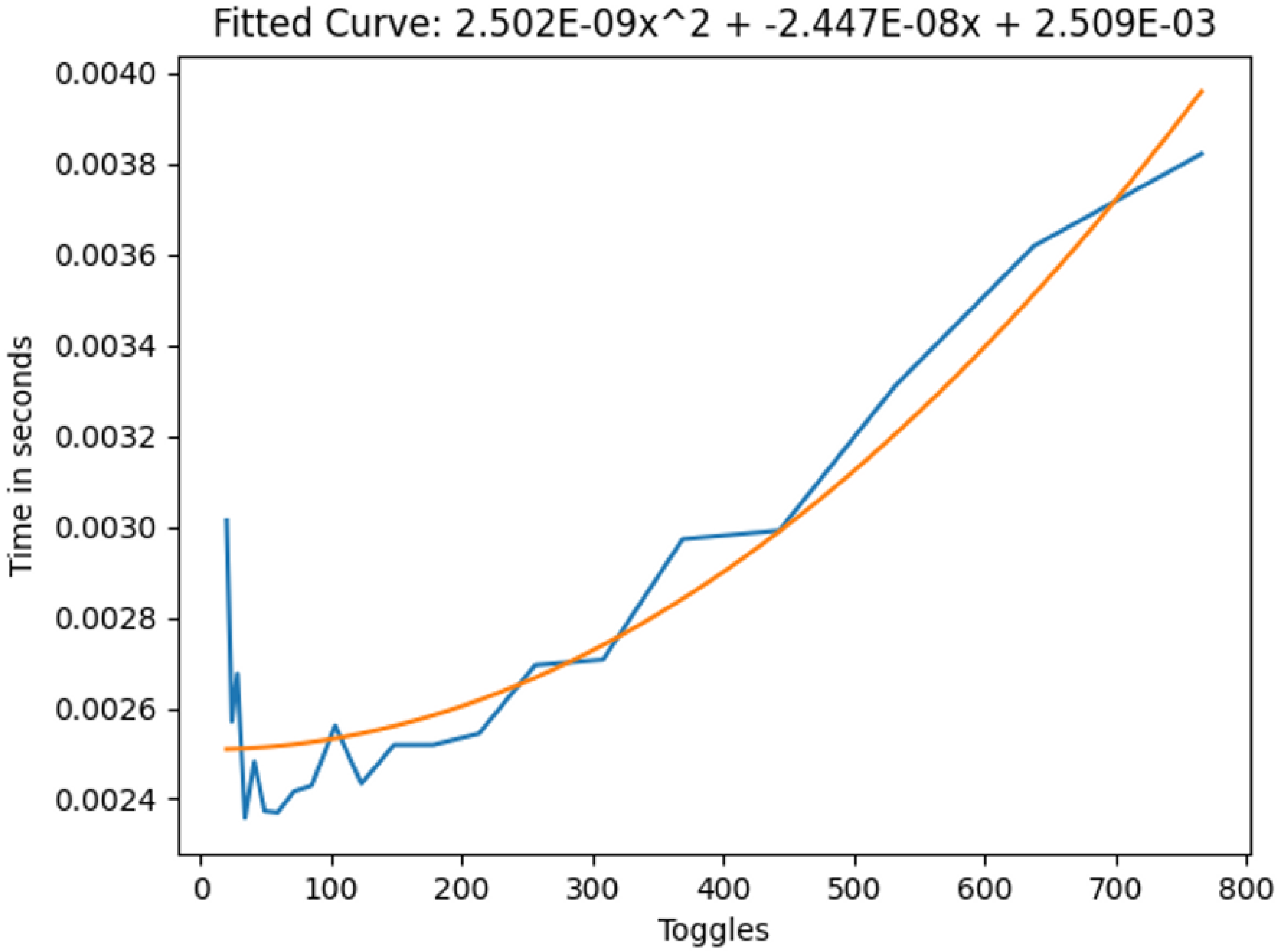
Direct Blend Tree在频繁切换时的帧时间

从这些结果来看，这种设置能将我们的帧时间减少约四分之三，这在性能上有显著的提升。尤其在大量切换的情况下，这种方法能够显著提升性能。

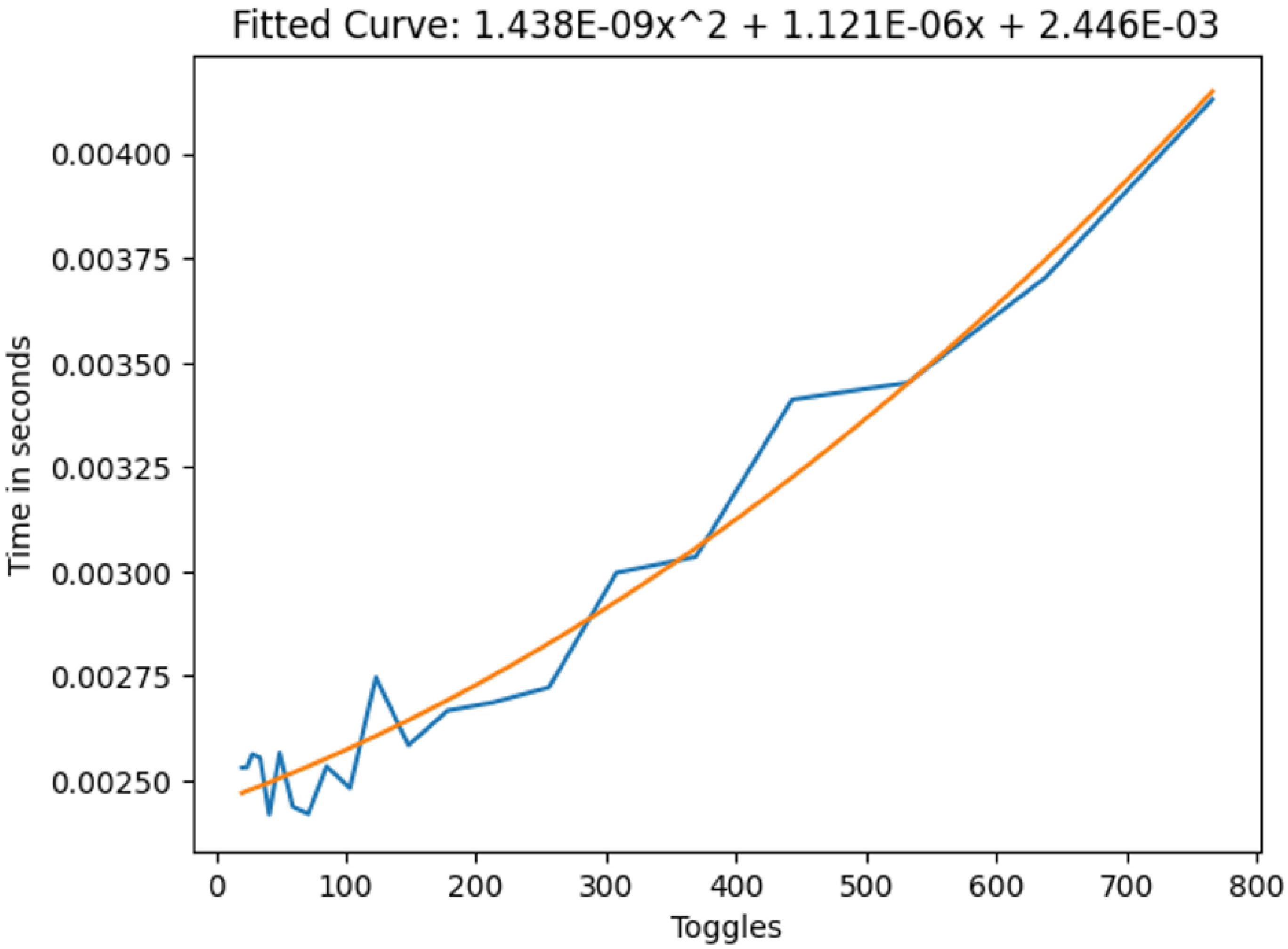
要了解如何创建这样的Blend Trees考Direct Blend Trees组合动画层

另一个稍微更快的办法是，需要两个层。第一个层是一个动画，控制所有切换的默认值；第二个层是一个直接Blend Tree，使用切换参数来控制每个切换的状态，每个切换配有一个动画。

例如，如果我有一双鞋子需要切换，并且默认状态下它们是开启的，那么在第一个层的大动画中，我会将IsActive值设置为true，而在第二个层的大Direct Blend Tree中，我会设置一个动画，



带有默认动画层的Direct Blend Tree在未进行持续切换时的帧时间



带有默认动画层的Direct Blend Tree在持续切换时的帧时间

从这些结果来看，相比简单的双状态切换，这种方法将帧时间减少了约五分之四，相比1D Blend Tree设置在主动情况下减少了50%。这是一项显著的改进，但其工作原理可能不够直观，设置也可能更加复

如果你需要频繁切换（例如，在面部追踪或经常使用语音参数时），这种方法可能会有所帮助。但如果情况允许，我个人建议继续使用双状态设置。

Revision #1
Created 8 August 2024 14:39:18 by 你亲爱的酒保K
Updated 8 August 2024 15:21:07 by 你亲爱的酒保K