

Autolayout和Autoresizing页面布局效率比较

实验目的：比较Autolayout和Autoresizing布局效率，取优应用在App中，提高页面滑动流畅性。

实验IDE：Xcode、Instrument、Swift

实验设计：使用tableView加载50行Cell，利用Instrument中的Time Profiler 观察滑动列表到底部主要耗时，其中包括cell生成（包含重用）、cell行高计算、总时间；cell总共设计了三种类型：

简单布局的cell：行内元素数量为3且显示数量不变，行高跟随图片高度变化，行内元素都是简单视图，1个UIImageView，2个UILabel。

中等布局的cell：行内元素数量6，显示元素数量根据数据改变，行高自适应，当然只有一个元素会根据数据是否隐藏，行内元素都是简单视图，2个UIImageView，3个UILabel，1个UIView。

复杂布局的cell：行内元素数量9，完全匹配流式布局，大部分元素会根据数据是否隐藏，行内元素有UIImageView、UILabel、UIView、UIScrollView、UITableView。

然后在heightForRow方法中之行[cell layoutIfNeeded]，计算行高。

实验结果：从10次实验数据中看，Autoresizing耗时要低于Autolayout，跳帧方面大体一致。

个人观点：

- 1.Autolayout只是简化了简单视图布局方式，以及行高计算，但并没有提高效率。
- 2.Autolayout不适用于流式布局，反复的添加删除约束，肯能达到最终要求，但逻辑复杂。
- 3.建议使用Autoresizing，只是计算高度过程恶心些，效率还是有保障的。

实验数据：

简单布局Cell：

简单的cell布局

Autolayout			Autoresizing		
cellforrow	heightforrow	all	cellforrow	heightforrow	all
457	908	2004	168	255	1112
874	1811	3893	341	498	2077
1334	2679	5737	510	753	3057
1815	3605	7728	672	1012	4083
2219	4469	9415	844	1273	4986
2639	5350	11159	1011	1522	5892
3137	6280	13157	1183	1768	6976
3568	7163	14923	1343	2028	7941
3993	8063	16730	1571	2423	9363
4004	8625	17455	1732	2682	10213
备注：红色部分由于操作失误，导致时间变高，但并不影响试验结果，结果时间单位是毫秒，从上到下的数据是执行相同操作累加的时间					
表象：上下滑动没有明显的跳帧					
结论：时间消耗上Autoresizing完爆Autolayout					

中等布局Cell：

中等的cell布局

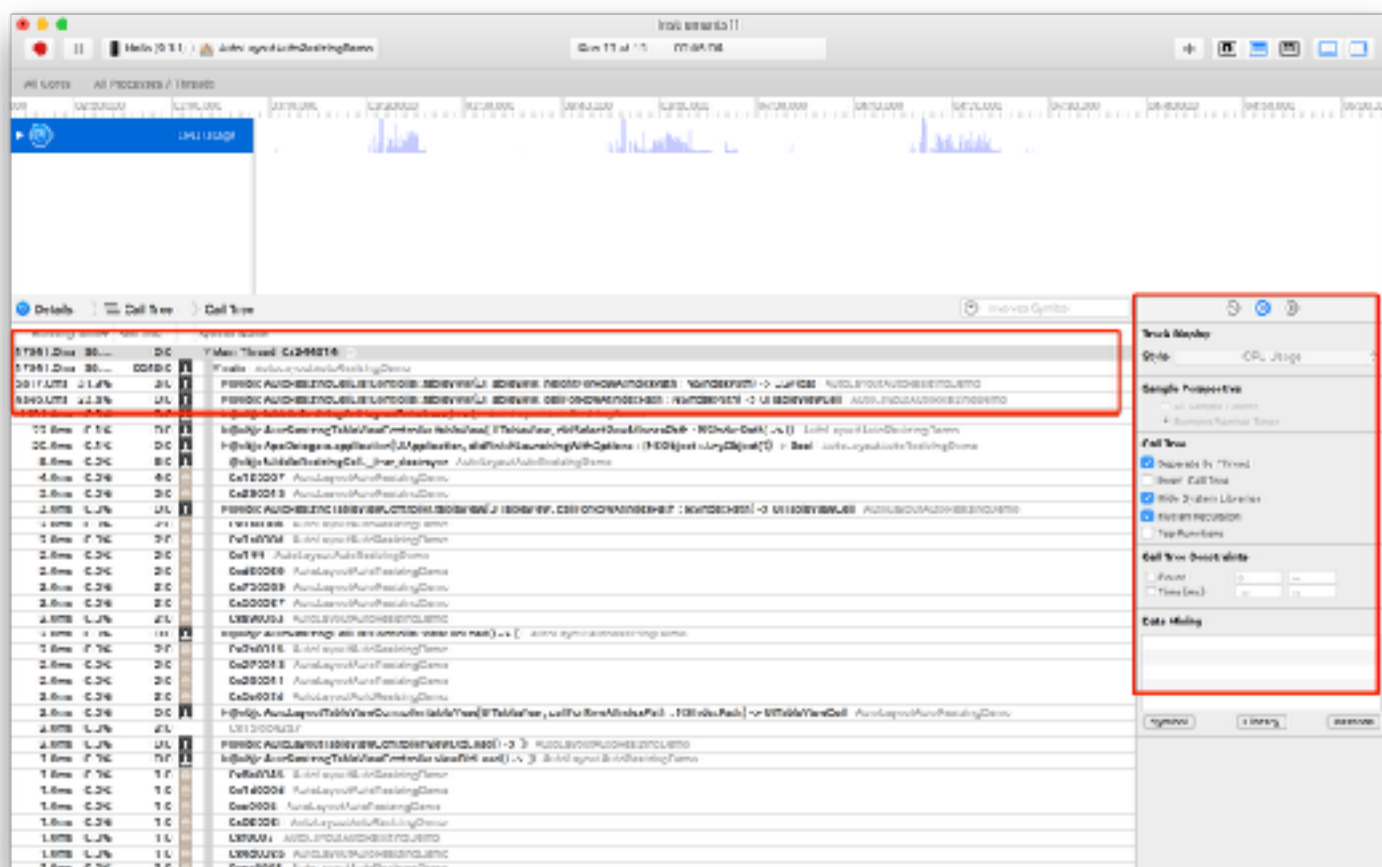
Autolayout			Autoresizing		
cellforrow	heightforrow	all	cellforrow	heightforrow	all
537	931	2272	388	579	1795
1131	1857	4526	707	1120	3284
1642	2795	6737	1081	1686	4854
2071	3658	8615	1533	2269	6622
2552	4549	10699	1964	2857	8378
2993	5443	12734	2386	3452	10137
3389	6337	14647	2907	4044	12075
3814	7213	16588	3264	4594	12650
4448	8139	18987	3799	5195	15773
5091	9067	21426	4346	5817	17941
备注：结果时间单位是毫秒，从上到下的数据是执行相同操作累加的时间					
表象：上下滑动没有明显的跳帧					
结论：时间消耗上Autoresizing完爆Autolayout					

复杂布局Cell：

复杂的cell布局

Autolayout			Autoresizing		
cellforrow	heightforrow	all	cellforrow	heightforrow	all
633	2511	4377	656	963	2926
1265	4904	8511	1178	1829	5334
1879	7293	12600	1562	2650	7321
2521	9671	16700	1847	3432	9103
2957	10966	19221	2287	4289	11203
3421	12290	21718	2717	5097	13277
3883	13582	24374	3261	5981	15704
4412	14859	26861	3683	6809	17844
4972	16188	29630	4255	7716	20374
5452	17507	32149	4628	8513	22288
备注：结果时间单位是毫秒，从上到下的数据是执行相同操作累加的时间					
表象：上下滑动跳帧明显，给人直观感觉差不多					
结论：时间消耗上Autoresizing完爆Autolayout					

实验数据来源:



遇到问题：

1. Autolayout没有使用系统中自动计算方法，原因如下：

自动计算方法适用显示元素数量固定不变，且元素之间的存在约束。

使用Autolayout自动计算方法即

2. 没有手动添加或删除视图约束，因为不确定哪些视图什么情况下隐藏不显示，逻辑复杂。

3. Autolayout在使用layoutSubviews计算高度时,误差很大，页面布局不准确。