郭老师，各位同学，大家好！

今天我毕业设计中期汇报的题目是《高效视频编码器在多核异构平台上的并行优化》。

简单阐述一下题目的含义和背景。所谓“高效视频编码器”是指基于最新的HEVC标准的视频编码器，“多核异构平台”是指有多个计算核心的CPU+GPU混合平台。

由于目前基于HEVC标准的视频编码器已经有很多实现，但是大多是基于CPU和串行执行的。因此我毕设的主要工作内容就是将这些串行算法通过GPU来进行并行加速。

首先介绍一下这一问题目前的研究现状。首先对于视频编码器的优化方式主要有两大思路，分别是优化帧间预测和优化帧内预测。而在优化帧内预测这一方向下，也有很多不同的研究思路。比如有针对候选模式选择方式的优化，有针对变换函数的优化，有针对量化函数的优化等等。

其中比较适合进行“并行优化”是其中的变换函数和量化和函数部分。因为这两类函数中包含了比较多的矩阵操作，可以最大地利用GPU的并行性能，同时这些函数每次调用时数据基本没有相关关系和依赖，所以可以比较便于由串行改写为为并行。

基于这一思路，首先分析帧内编码的代码实现。这里我基于的代码是HEVC test Model，也就是HEVC官方给出的编码器实现。其中帧内编码相关函数的主要调用层次如下。其中Hadamard变换主要是用于计算每种预测模式的SATD，以便粗选出其中的最优的模式放进候选模式中。我目前主要对SATD计算的这一部分完成了优化，首先分析一下这一部分中可以并行优化的部分。

在estIntra这个函数中，对HEVC的35中预测模式进行了遍历，对于每一个模式分别调用xGetHADs这个函数。这里是第一个可以并发执行的地方，也就是可以同时计算35种模式的SAD。接下来xGetHADs这个函数又会再遍历64×64块中的每个8×8的小块，然后对其调用xCalcHADs8x8这一函数。这里显然也可以并行。最后xCalc这一函数会计算一个8阶矩阵的Hadamard变换，并且累加每一项系数和。这个过程同样可以并行。

基于这一分析，将上述可以并行的过程进行CUDA实现，主要思路如下。然后对于优化后的代码使用不同的测试序列进行测试。其中GPU-SAD是指仅优化SAD计算后的结果，GPU-35是指同时计算35种模式的测试结果。实验结果如下。平均加速比分别约为1.89倍和7.16倍。

最后简述一下我后续打算进行的工作。相对整个HM代码而言，目前优化的部分所占的比重其实相对比较小，大约为10%，而DCT变换函数和量化函数等等所占的比重其实较大，约为32%。因而之后我会进一步对这些函数进行优化，从而获得更好的加速效果。另外目前我使用的矩阵乘法是自己实现的，效率比CUBLAS提供的矩阵乘法效率要低，这主要是因为。