

为什么 zAI 的流式视频方案都是独立图像而不是视频流

视频流需要关键帧来保持对画面动态块补码连续性,如果画面是连续的,视频流就可以用很小的 size 在网络传输,但是如果画面是非连续性的,在断帧间,就会需要关键帧,或则出现画面丢失(马赛克).

H264+H265 的在监控中的图像补码能力非常强悍,关键帧记录静态背景,只对有动作的物体进行块补码,一个 10 秒的街道行人监控视频,能轻松压缩到百 K 内,但是这在 GPU 上实时处理,并不实用,因为 GPU 都会将视频解码成独立图像,处理完成一帧,然后才去处理下一帧.如果帧处理带有业务逻辑的性质,延迟就会更大.

假如我们拍摄了 2 秒的标准手机视频会有 60 帧,这意味着,使用流式视频机制,会反复让 GPU 计算 60 张图片.如果使用 1080p 标清的多路视频输入,某些检测和分类的算法在 GPU 上肯定会跑不动全帧率,需要降低帧处理频率,以及码率.而这样干,就没有流式视频的意义了,使用独立图像,会节省硬件和带宽的开销.

处于以上考虑,所以 zAI 在设计 CS 流式视频方案时,采用了独立图像机制.

我们向服务器发送的数据都是独立图像,并不是实时视频,待服务器处理完成反馈以后,我们才发送下一帧图像.帧间间隔,并不是实时视频,而是由网速+GPU 共同协调好的整体系统.

开发完整流式视频方案,我们需要自己动手在视觉呈现部分,让画面看起来像是流畅的,而不是暴力的让服务器按全帧率去工作.尽量避免极限的程序优化工作!

我们通过 zAI 提供的 CS 流式视频 Demo 可以从中体验到这一机制.

2019-7

By.qq600585