Cimarron
Stabilisation of videos in modern `C++`

Marius Herget

in partnership with

Institute for Computer Science
Ludwig-Maximilians-Universität München

10th October 2018

1 Idea

Video stabilization is used ever since cameras evolved. In the early days physicial stabilization techniques as tripods were used. In the following centuries cameras enhanced step by step. New solid and dynamic methods were invented like steady cams, dollys, shoulder rigs and many more. With the invention of digital photography and videos another possible solutionwas found: digital image stabilization. Different techniques like optical flow analysis or warp stabilization were developed. `Cimarron` implements such a feature tracking method for motion compensation.

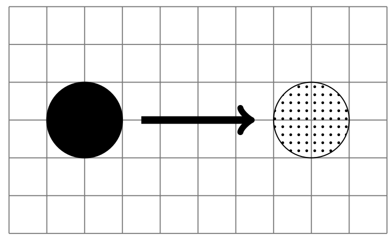
2 Theoratical introduction

New technologies emerge each year. In the last years espacially phones and small cameras were published. Under ideal condition recent smartphone's cameras pictures cannot be distinguished from professional camers anymore. Nevertheless, a smartphone video is often detectable by its *handheld*, shaky look. As already mentioned within the short introduction different methods can be used to compensate this motion.

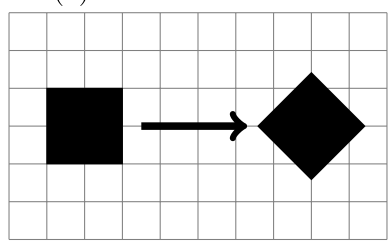
The general idea of video stabilization is to counter, smoothen or to minimize unwanted shakes. In general video motion stabilization can be classified in three categories: mechanical based, optical based and electronical based. Instead of using specific hardware like the first two methods, the electronical approach uses computing power to implement image processing techniques in the postproduction step. [1]

In order to compensate the unwanted movement of the camera motion can be described in various forms. *Translation* is the simplest form of expression. In this concept direct, linear movement of a single point is described as the distance it covered within a certain time. This can enhanced with the combination of *rotational motion*. In comparision to translational movement it specifies the angle a point / body covers in a given timeframe.

3 System diagram



(a) Translation motion



(b) Rotational motion

Figure 1: Differnt motion models

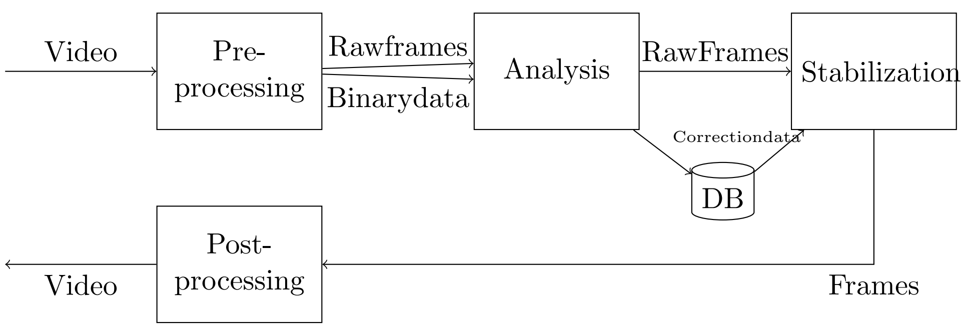


Figure 2: High-level system diagram

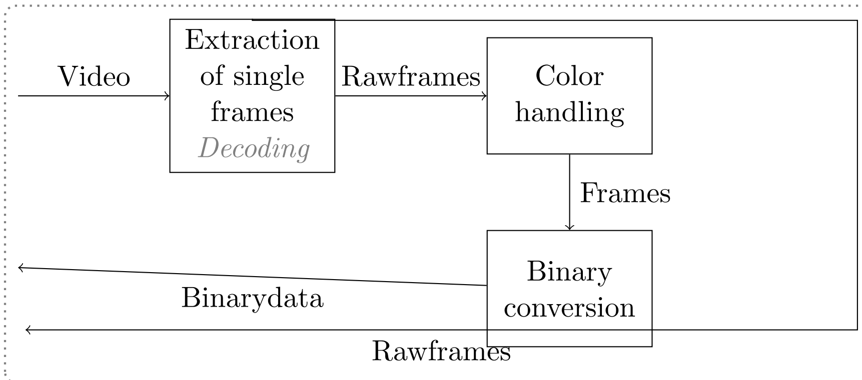


Figure 3: Detailed system diagram of the *Preprocessing*

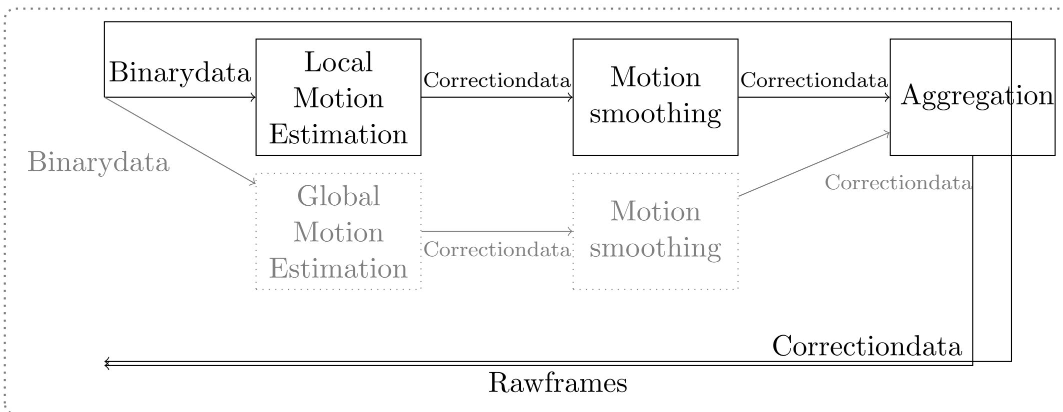


Figure 4: Detailed system diagram of the *Analysis*

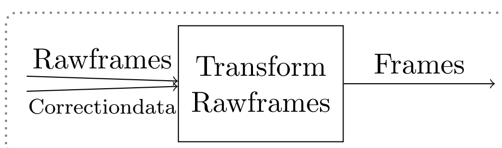


Figure 5: Detailed system diagram of the *Stabilization*

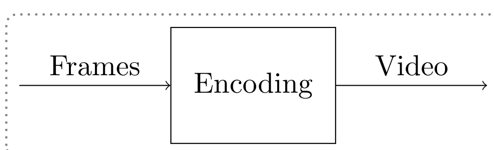


Figure 6: Detailed system diagram of the *Postprocessing*

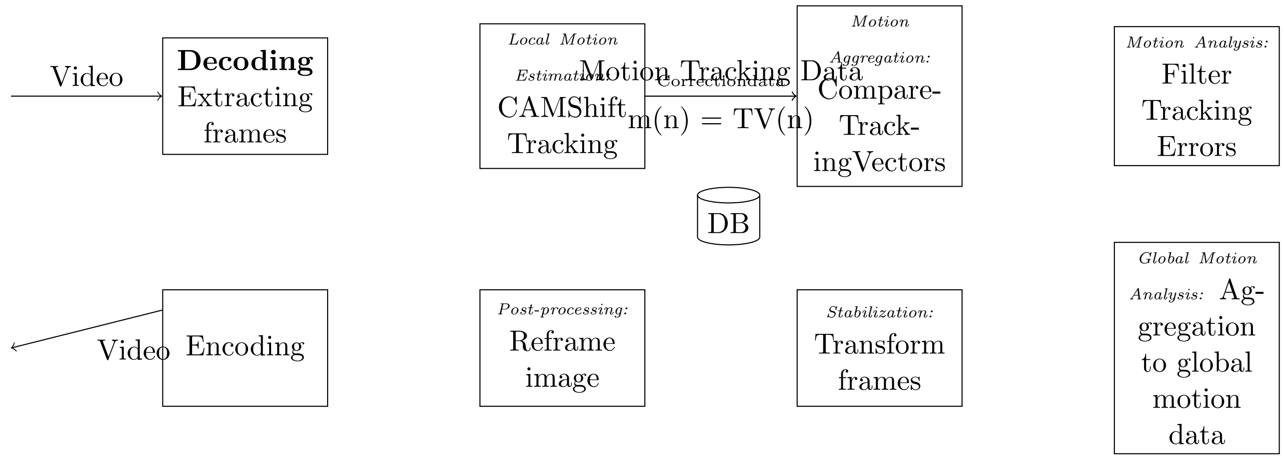


Figure 7: High-level system diagram

4 Coding Concepts

Expression	Return	Equivalent expression	Notes

preprocessing

References

[1] Chongwu Tang et al. *A fast video stabilization algorithm based on block matching and edge completion*. Oct. 2011.