

DIGIEZ — Repair a corrupted video

CENTRALESUPLEC — Max Spahn — P2018

1 Idea

Given the task of reconstructing the video, my first step was to frame the video in images so that i can treat them individually. The main idea of my program is to compare individual pixel of all picture created that way. It is up to the user how many pixels he wants to be compared. An increase in pixel-number increases the accuracy of the program but causes a longer execution time. Once the images are put in the right order, you need to rebuild the video.

2 Technical details and problems

The programming language I am the most familiar with is C++, but I couldn't find a good library for my purpose. That's was the reason to choose FFMPEG as an external programe that needed to be combined with the cpp code in a bash file. On the other hand, the bash-file gave me much opportunities to automate the program. For example, it is kind difficult to compute the number of images created by ffmpeg, but a simple line in the bash-file counts the number of images. The number is than used for the iteration in the C++ program. The images are stored in the folder "images/" and they are named *outxxx.bmp* where xxx is a place holder for the number of a particular image. I decided to use the bitmap-format because it is one of the easiest to work with. The drawback is that there is now compression so the storage might become problematic if the video becomes too long. Then the C++ program is executed with two command-line arguments, the first is the number of pixel in each picture direction, the second is the number of images previously counted.

An array of changings is created to keep track of the order of the images. When every picture is put into the right position, the image-files are re-named with regard to the changings-array. They are now called *newxxx.bmp*. FFMPEG is used a second time to reform the video, now in order.

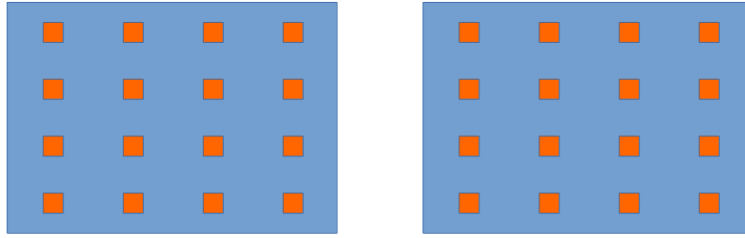


Figure 1: Explanation of how two images are compared, for divison-factor $\text{div}=4$

3 Problems

The major problem might be the compability with other operating systems than linux which i used for the program. On top of that you need to preinstall FFMPEG and the C++-library for treating bitmaps. I put all the files needed to use EASYBMP in my github, <https://github.com/maxspahn/digiez> with the other file, and FFMPEG can be downloaded on the link: <https://ffmpeg.org/>. In case of problems I would be happy to talk to about my solution.