https://doi.org/10.2352/ISSN.2470-1173.2020.2.SDA-053

Morpholo: A Hologram Generator Algorithm

Enrique Canessa and Livio Tenze; Science Dissemination Unit (SDU), SciFabLab, ICTP - International Centre for Theoretical Physics; Trieste, Italy

Abstract

We introduce the Morpholo library which is able to convert a stereoscopic snapshot into a native multi-view image through morphing and takes into account display calibration data for specific slanted lenticular monitors. Holograms are generated fast by creating Lookup Tables to replace runtime computation. The implementation of Morpholo for glasses-free streaming of live 3D video is considered.

Introduction

A 2D image is worth a thousand words. Notwithstanding a 3D reconstruction of a scene is worth a million [1]. The new class of standalone HDMI Looking Glass HoloPlayer Monitors (LKHM) allow to display a hologram of simultaneous *nxm* different views at 60 *fps* formed through a collage of images (or Quilt) without the need of wearing unnatural headsets [2]. This device combines light-field and volumetric technologies. Indeed, LKHM allow to visualize readily, and perceive better, the complexity of real world objects and 3D scenarios for around 40^o to 50^o fields of view.

The multi-views display technology is still under development even if the 3D TV technology was emerging together with the raising of the XXI century. It still needs to be optimized so as to reduce the high production costs and some of the health concerns induced by such 3D presentations. Nevertheless, LKGM may have great potentialities for diverse innovative scientific, engineering and 3D video game applications [4]. For future fast, large-scale, server-side live applications of 3D streaming further research is still needed. For instance, applications of LKFM under Linux O.S.-based systems and different low-cost computer models, such as the new Raspberry Pi 4-range, are not available yet. As of today, native images can mainly be displayed smoothly on a LKHM under MS Windows and Mac O.S., having specific computer display calibration values each for a correct rendering.

In this paper we introduce an extended Morpholo library which is able to convert a stereoscopic snapshot into a native multi-view image by morphing. The resulting native images are generated fast via Lookup Tables (LUT). Within Morpholo, we compare results obtained from three different morphing methods: Disparity map (see ,e.g., [5]), DeepFlow [6] and View morphing [7]. We discuss the limits of each approach to generate the intermediate multi-view images needed to create native holographic frames from Quilts. This study can be useful in future applications for real time 3D streaming in a specific LKGM.

Morpholo

The Morpholo library provides the following features: firstly, communication with the LKGM to get the calibration data; secondly, the transformation of a stereoscopic image (that could form a 3D video frame) to a collection of views by morphing; and thirdly, the mapping of these multi-views to the destination holographic space.

In other words, Morpholo consists of several different C++ programs: one to (*i)* read display calibration values via a specific LGKM's eeprom interface, (*ii)* generate an associated LUT from these values and also according to some given resolution for the screen, *etc.,* (*iii)* retrieve synchronous stereoscopic images taken from a stereo webcam, (*iv)* generate *n*-intermediate views by morphing, (*v*) place these views into a single standard Quilt image, and (*vi)* treat this collection of views into native images consisting of light-fields that can be correctly displayed on the specific LKGM.

Within Morpholo, the Quilt representation is form by implementing two distinct morphing processes for the stereo images: (*i*) a Disparity map approach [5], which allows us to convert the stereo image to depth maps. By combining the left-*to*-right and right-*to*-left depth maps we create the intermediate views starting from the left (L) image to the right (R) one; (*ii)* a DeepFlow approach [6] to estimate a dense optical flow field. In this case each point of the L image is compensated to reach the related point in the R image. It is very important to properly treat occlusion and opening regions to get a homogeneous motion field. This field is then used to create all Quilt' s intermediate views.

Up scaling multiple views from the stereo (L,R) images to the native image can require substantial computer resources depending on the Quilt resolution. In order to feed the holographic display, each point of these Quilts has to be mapped to the destination image –usually a native holographic image of 2560x1600*p.* The mapping procedure needs to be very accurate according to the Quilt dimensions and has to take into account the specific display calibration data for each LGKM set in the phase of manufacturing.

The relation between pixels of a slanted lenticular 3D LCD monitor and multi-views in the native image was derived in [8]. Each sub-pixel on the 3D display is mapped to a certain view number and color value (in the light-field domain). With i and j denoting the panel coordinates for each sub-pixel, then

$$N_{i,j} = N_{tot}\ (i - i_{off} - 3j\ \tan(\alpha))\ mod(P_x\ )/P\ , \qquad (1)$$

where N is the view number at a certain angle, α is the slanted angle between the lenticular lens and the LCD panel and P_x the lenticular pitch. This equation also depends on the input parameters of the physical structure of each unique LKGM.

In order to speed-up the mapping procedure in Eq.(1) we implemented a LUT which provides an extremely fast mapping related to the chosen Quilt dimensions and to the monitor calibration data (including lenticular pitches, slope, screen height and width and number of lens per inch). The LUT is evaluated and generated only once, at startup. After the initialization of the map, the mapping procedure consists in a memory access loop to put each Quilt pixel in the output destination image.

Results and Discussion

The Morpholo library has been written in C++ and implemented using the openCV library with the goal of minimizing the image processing time to approach future real time applications in 3D streaming without the need for the viewer to wear 3D glasses.

We take (L,R) stereoscopic pictures using a compact ELP-960P2CAM USB stereo webcam with no distortion dual lens aligned parallel toward the horizon [9]. According to its technical specifications, the small size camera –having standard rolling shutter and 1/3” CMOS OV9750 sensor for high quality image– can reach high frame rates in MJpg. It supports Linux O.S. USB video class UVC with adjustable brightness, contrast, saturation, hue, sharpness, color balance and exposure. The two camera video frames are synchronous with 1280x960*p* max. resolution. This feature enables us to mimic the manner in which human eyes observe “simultaneously” one scene from two different viewpoints. By one single shot we retrieve a single image in HD resolution, containing (L,R) views, without the need for any extra images calibration. The PC used were standard Intel dual Core, 1366x768*p* resolution, with O.S. Linux Ubuntu 19.04. Also we have tested Morpholo in Raspberry Pi 3 and 4 computer devices Model B.

Since animating stereograms by morphing between (L,R) images can give an acceptable illusion of depth and parallax in the horizontal direction as discussed in [10,11], we place the intermediate views generated using Morpholo in a Quilt collage sequentially. Before this morphing process, we resize each view to a smaller resolution to avoid running “*out of RAM memory*” in the calculations. The fading out from L to R images, aims to render a whole panoramic scene into just a given number of intermediate snapshots.

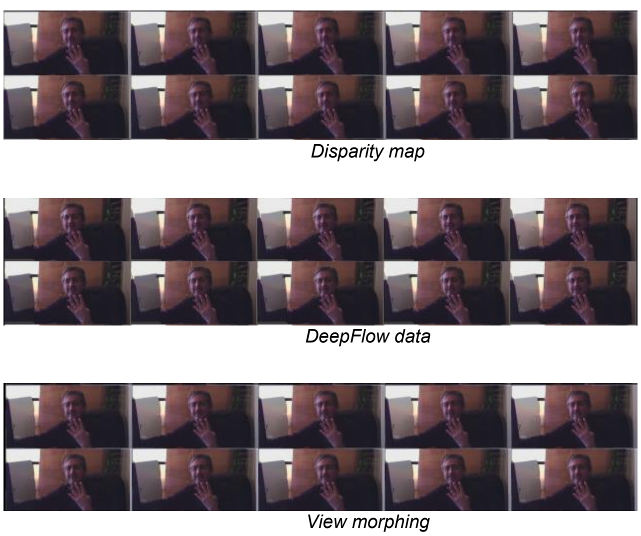

*Figure 1: Example of intermediate views (*e.g., *positions from number 11 to number 20 in the Quilt) by morphing a stereoscopic image from L-to-R using our Morpholo library: (i) Upper set of images are generated via Disparity map, (ii) Middle images via DeepFlow and (iii) Lower images through View morphing.*

The gradual cross-dissolving between pair of stereo images is illustrated in Fig.1 for Disparity map and DeepFlow approaches. For comparison we also show in the figure morphed images generated via View morphing [7]. View morphing requires to specify (manually) a set of at least 4 line correspondences before the image can be properly morphed to get realistic results. It is by far more time consuming due to the pre- and post- warp process of images specified by these correspondence parameters.

We have found that to generate the set of morphed images of Fig. 1, disparity map and optical flow techniques can lead to a reasonable accuracy in the reconstruction of 3D reality from just two stereoscopic images. Fading from misaligned L to R images can cause the other intermediate parts to blur and get distorted. In our case, though, this is reduced considerably by the use of the *ad-hoc* ELP stereo webcam. We position objects in the scenes, at least 2 meter distant apart from the stereo webcam to avoid well-known image distortions preserved through the morphing process. At this distance, the middle morphed image reasonably fades out and becomes clear without excessive ghosting.

The morphed images in Fig.1, forming the Quilt, are then converted into a native light-field image for the LKGM via Eq.(1) as shown in Fig.2. As mentioned above, the resulting holograms are generated fast because of the automatic pre-configuration of a LUT by Morpholo based on the per-device HoloPlayer calibration “json” values. The beautiful light-field arrangements visualized in LKGM tricks our visual perception into seeing volumetric display by parallax (*i.e,* moving the head around a middle Zero-Parallax Plane), and by 3D stereo vision (i.e., presenting different views to each eye*)*. Different depths show different optical properties.

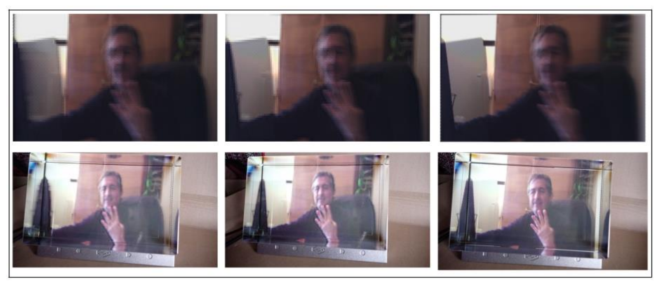

Figure 2: Example of multi-view image outputs by Morpholo library (upper row) and their actual visualization through the Looking Glass HoloPlayer (lower row) generated from the intermediate images in Figure 1, ordered in a Quilt and morphed via Disparity map (left image), DeepFlow (central image), and View morphing (right image) approaches.

The CPU time for creating all these sets of images, and especially the holographic multi-view outputs for the LKGM, varies considerably between the Disparity map and DeepFlow approaches as can be seen in Fig.3. Key processing steps in the flow field include the arbitrary, and time-consuming, matching by polynomial interpolation to approximate neighborhood pixel intensities, warping and optimization [6]. Optical flow techniques are sensitive to the presence of occlusions, illumination variations and out-of-plane movements, whereas disparity map leads to obtain smoother and faster translation motion between consecutive images.

The use of a per-device calibration procedure, the class of morphing algorithm selected, the stereoscopic images resolution adopted, and the application of the geometric transformations in Eq. (1), can require large processing times.

As quantified in the curves of Fig.3, reasonable total processing time steps below 0.1 *sec,* and around 10-*to*-35 different views, can be obtained via Morpholo. This implies that in one second we could actually generate 10+ different 3D video frames. PC-1 in the plot corresponds to a old PC Intel Core i3 with 14G RAM and PC-2 to a newer PC Intel Core i5 with 8G RAM. Small-resolution stereo pairs in the plot (having, e.g., 64x128*p* –the size of an OLED Graphic Display Module), lead to a 9x5 Quilt of size 576x640*p.* The s2- and s4- levels in the curves indicate the generation of intermediate images starting from the actual (*n,m*) stereo pairs to then continue the morphing by alternate set of "equivalent" (*n+s,m-s*) stereo pairs. This shift nicely leads to further reduce the total processing time and maintain the principal scene characteristics in the native images.

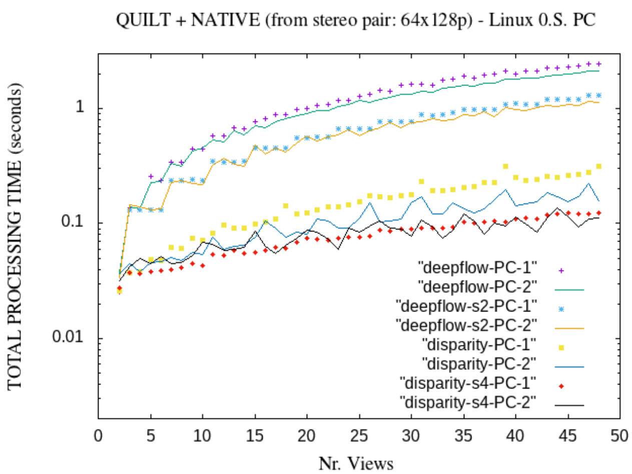

Figure 3: *Morpholo performance test of the total processing time employed for the generation of multi-view holograms starting from a set of stereoscopic images.*

To output a native image at a resolution of 2560x1600*p* with RGB color channels –such that, once the pixel to be mapped is fixed– the mapped value for each color channel implies separated calculations.

Streaming in real time a hologram feed is complex and computationally demanding, because of the larger amount of information contained in the many light-fields to be streamed, as compared to sending a single-view 2D video stream. With the use of several cameras, it takes a lot of time-processing and hardware capacity to convert and encode live multi-views, having, e.g., 720p resolution with a low latency of 20 *fps* or less. One main problem is to control two (or more) cameras and to process synchronously the video inputs of each one with some codec and format. For the software program, the difficulty is to encode 3D video data that result into "realistic" multi-view video frames to be visualized on lenticular monitors. Existing 3D TV video technologies are mainly based on the use of stereoscopic cameras.

We have constructed a Lookup Table (LUT) to replace runtime computation and save calculation power. In the absence of a LUT procedure, it would become computationally expensive and difficult to apply morphing techniques for real-time video in 3D. Our LUT is created only once at the beginning of the needed mapping starting from the creation of the (L,R) stereo images up to the displayed light-fields. We allocate 3 matrices for the 3 color channels RGB of size 2560x1600*p* times 2. Each matrix provides the X coordinate of the Quilt from which we take the corresponding value and the Y coordinate. The multiplication by 2, allows to avoid unnecessary waste of resources and consume the least possible amount of RAM memory. All positions of the 2560x1600 pixels are browsed and we calculate the mapping value for each pixel in the Quilt image. This value is stored in the three different allocated matrices and each element of the matrices is made of type uint16_t. The matrices are saved in binary format and these are reload (without recalculating) when applying the mapping to all the intermediate images.

As seen from Fig.3, the implementation of LUT allows to reduce and optimize computing time. This procedure also allows to speed up significantly the needed mapping procedure –the rendering operation of the final native image is essentially achieved by accessing the elements of the 3 matrices to map the Quilt pixels. These possibilities open the path for producing multi-view 3D streaming in real-time with a simpler and faster algorithm than direct calculations. Morpholo at present is essentially a purely near-neighbours-pixel method. We have mainly extrapolated the parallax encoded in the images by combining information from nearby pixels only.

Conclusions

We have proposed a fast conversion Morpholo library to transform stereoscopic images into reasonable holograms via the automatic implementation of a one-time configuration LUT and unsupervised morphing deformations between (L,R) images.

We compared results obtained from three different morphing methods: DeepFlow, Disparity map and View morphing, and discussed the limits of each method to generate intermediate multi-view images ready for display on slanted lenticular monitors with horizontal parallax only, such as the HoloPlayer. Although morphing cannot provide complete depth information on distant regions, the generation of multi-view images starting from a pair of stereoscopic images through Morpholo could still offer a potential alternative method.

Future work can include the application of the Morpholo hologram generator algorithm in the field of live 3D streaming using low-cost hardware and without the use of any headset. Binaries of our code will be made available at www.morpholo.it

References

[1] J. Geng, "Three-dimensional display technologies", Adv. Opt. and Photon. Vol.5, pp. 456-535, 2013. DOI 10.1364/AOP.5.000456

[2] The *Looking Glass* Holoplay combines together light-field and volumetric display technologies within a single 3D display system: www.lookingglassfactory.com/

[3] docs.lookingglassfactory.com/Appendix/how-it-works/ "How the Looking Glass monitor works", 2019.

[4] E. Canessa, E. and L. Tenze, "Morphing a stereogram into hologram", submitted 2019. Available arxiv.org/abs/1905.01727

[5] H. Huang, S. Nain, Y. Hung and T. Cheng, "Disparity-based view morphing a new technique for image-based rendering", in Proc. ACM symp. Virtual Reality Soft. and Tech., pp. 9-16, 1998. DOI 10.1145/293701.293702

[6] P. Weinzaepfel, J. Revaud, Z. Harchaoui, C. Schmid, "DeepFlow: Large displacement optical flow with deep matching", in ICCV - IEEE Int. Conf. Comp. Vision, Sydney, Australia. pp. 1385-1392, 2013. DOI 10.1109/ICCV.2013.175. Hal-00873592

[7] S.M. Seitz and C.M. Dyer, "View morphing", *in* SIGGRAPH Conf. Proc., pp. 21-30, 1996. DOI 10.1117/12.349368

[8] C. van Berkel, "Image preparation for 3D-LCD", Proc. SPIE 3639, Stereoscopic Displays and Virtual Reality Syst. VI, 1999. DOI 10.1145/237170.237196

[9] Ailipu Technology Co., manufacturer surveillance systems: www.webcamerausb.com

[10] R. Baumann, "Animating stereograms with optical flow morphing", 2016, Blog ryanfb.github.io

[11] J.A. Davis and D.F. McAllister, *"Morphing in stereo animation",* Proc. SPIE 3639, Stereoscopic Displays and Virtual Reality Systems VI, 1999, DOI10.1117/12.349397

Author Biography

Enrique Canessa received his PhD in Physics from East Anglia University, Norwich, UK (1990). Since then, his interests have also been on open software development and applications in the field of STEM education and science dissemination at ICTP, Trieste.

Livio Tenze received his PhD in Image Processing from the University of Trieste (2002). He has worked in the Engine Dynamics Department, Ferrari Racing Team. His work focus on software development and optimization, lecturing, cloud computing and new IT technologies.