QUALCOMM

Qualcomm Technologies, Inc.

Virtual Reality Content Creation Technology

Pawan Baheti

Feb 27, 2017

Qualcomm Technologies, Inc.

Qualcomm Snapdragon is a product of Qualcomm Technologies, Inc..

This technical data may be subject to U.S. and international export, re-export, or transfer (“export”) laws. Diversion contrary to U.S. and international law is strictly prohibited

Qualcomm Technologies, Inc.

5775 Morehouse Drive

San Diego, CA 92121

U.S.A.

Table of Contents

Figures

Abstract

Virtual Reality (VR) content creation is a challenging problem. There are multiple technology dimensions which must be addressed in order to make the content quality supreme. Some of the key aspects include a) synchronization of multiple sensors collaborating to capturing the 360-degree field-of-view (FOV), b) color equalization across the cameras, and c) advanced stitching algorithms which combine the outputs from the multiple sensors into one seamless panoramic video. In addition, the resolution of the stitched canvas must be high since we stream to HMDs where pixelation can be much easily noticed. For live VR experiences, it becomes extremely important to enable real-time stitching, and streaming of the panoramic video. Other applications for 360-degree video include surveillance, drones, action cameras, smartphone imaging etc. In this paper, we present the system overview of QTI's VR content creation technology. We include a discussion on what are the potential improvement areas on the stitching and equalization front, and what are the next leading directions on the VR video content creation side.

Introduction

Virtual Reality (VR) is currently going through a phase of rapid technology development as well as strong customer adoption. The excitement around VR is apparent across the consumer gaming and entertainment industries. There were many consumer VR gaming devices in the market by the end of 2016.

There are two flavors of VR today – "wired" and "wireless". In a typical wired VR solution, the head mounted display (HMD) is generally wired to a gaming platform (a dedicated gaming console or a high-end personal computer) in order to take advantage of the power of the compute solution that the device offers. Additional hardware may be needed to achieve position tracking of the HMD. The "Wireless VR" segment is far more popular. The wireless solution is comprised of inexpensive "slot-in" plastics / cardboard with an "insert slot" for a smartphone. A smartphone which serves as both the computing device for VR processing as well as the display is inserted into the HMD.

The content that VR users will consume includes VR games and other computer-generated virtual environments as well as VR videos. As much as gaming and computer-generated VR content is popular today, it is believed that as VR adoption increases, much of of VR consumption will be VR video. Professional 360-degree/VR content (movies, TV) creation will bring about big changes to the content creation industry including film and TV production. Many filmmakers are already experimenting with this new format and re-writing the rules of film making for VR. VR filmmaking also requires a re-design of the entire professional content production process and toolchain. There is also very strong adoption of user-generated content in VR. The biggest social networks already support 360-degree videos on their platform and invite their subscribers to enjoy the 360-degree video content on VR headsets.

VR video content creation is an extremely challenging problem due to the high resolution and frame-rate needed, the need for elaborate synchronization and alignment across the sensors, as well as the highly complex stitching algorithms that must be employed.

Resolution and frame rate

The 360-degree scene must be captured and stitched together into a high resolution canvas at a high frame rate, preferably in stereoscopic 3D. A high resolution (>4K) and frame rate is critical since the content will be viewed in an HMD, which magnifies the scene and presents it very close to the user's eye. The high-resolution requirement and the 360-degree field-of-view (FOV) drive the need for the use of multiple high-resolution sensors. There are a wide array of products in the market that trade-off the resolution of the canvas against the number of sensors needed.

In the professional camera segment, there are some products in the market that have use several independent cameras (From 6 to 100s). Typically, each camera independently records a video of a portion of the scene and the recorded videos are synchronized and stitched off-line using a powerful desktop. These camera systems are priced for professional content producers (USD 10K-250K across these products).

In the user-generated content segment, there are a few "integrated" consumer 360-cameras targeted at the drones, action cameras and surveillance market. These products usually have between 2 to 4 sensors connected to a single processor system, all enclosed in a single device. Since the number of

sensors is limited, these products can function with a single System-on-Chip (SoC), thus bringing the product within the price-range of a typical consumer product. However, the resolution of the canvas is limited due to the smaller number of sensors.

Multi-camera Synchronization and Alignment

The camera sensors which are each capturing a portion of the entire scene must be closely coordinated such that they're capturing the individual frames in perfect synchronization with each other where timing, color, and sensor orientation are critical.

- Timing – The sensors must each capture the individual frames that will be stitched together at precisely the same instant of time. Since most sensors use rolling shutters, this implies sensor lines must be individually synchronized.
- Color - The raw sensor processing happens in the Image Signal Processor (ISP) in a manner that is content adaptive. 360-degree cameras must synchronize the camera parameters across multiple sensors such that the colors appear smooth in the overlap regions across the multiple sensors.
- Orientation – The camera sensors must be rigidly placed with respect to each other and must not move, once they've been installed. The relative positions of the camera sensors must be accurately calibrated at setup and used during the stitching process to create a seamless panorama.

Stitching Algorithms

The synchronized high-resolution images from the multiple cameras must be stitched together into a canvas. This is a particularly tough problem since the sensors are typically separated from each other spatially and the captured images are, therefore, subject to parallax. More precisely, in the overlap region, the objects in the common view of two cameras (which are spatially apart from each other), appear slightly different in each camera and must therefore be stitched together using special techniques which ensure that the seam is not visible. Often, these techniques require significant processing including depth-estimation and object tracking, which are not friendly to real-time processing at high frame-rates and resolutions.

Many cameras systems are therefore content with capturing the individual video sequences from each of the sensors and deferring the stitching to an off-line process on a powerful desktop computer or a cloud server. While this works for professional cameras, this step reduces the user-friendliness of the consumer 360-degree camera and precludes interesting applications such low-latency live 360-degree camera streaming.

We contend later in this whitepaper that modern smartphone SoCs have increased sufficiently in performance and power efficiency in the last 10 years, enabling them to efficiently interface with multiple sensors and handle the requirements for high quality live 360-degree stitching.

Formats and Resolution

VR content is best captured in stereoscopic 3D. A good capture solution must create a seamless high-resolution 3D "canvas" that represents the surround view. 3D canvases, however, are very

difficult to create and reconstruct without special optical equipment. The generated views are also inaccurate outside a narrow FOV thereby requiring 10s of narrow FOV camera sensors for high accuracy.

Another push towards using more sensors is the canvas resolution. The images from the camera sensors are stitched together into an output panoramic "canvas" much like a three-dimensional globe is mapped onto a rectangular map. The format of the map is usually the "equi-rectangular" format, where (in the map analogy), the latitudes and longitudes are equally spaced. The rectangle therefore has a 2:1 aspect ratio (360-degrees of longitudes and 180-degrees of latitudes). Other formats such as equal-area and cube-map are also possible and offer additional efficiencies for compression, at the cost of intuitive representation. In this discussion, we constrain the discussion to the equi-rectangular format but the arguments can be trivially extended to the other formats as well. A typical equi-rectangular canvas size is 4K (4096x2048) which offers equivalently, an effective angular granularity of approximately 11.4 pixels/degree of rotation (4096/360-degree or 2048/180-degree). While the 4K resolution is generally considered sufficient for regular (narrow FOV) videos, it barely serves as a baseline quality for 360-degree videos. The human eye is capable of discerning angles as sharp as 1/60th of a degree in the foveal section (where the eyes are focused) of the view. Thus, for zero perceived pixelation, the 360-degree canvas must be approximately 20Kx10K pixels per frame, which is likely possible by using 10s of sensors working in synchronization.

Consumer VR cameras, on the other hand, typically restrict themselves to a 360-degree/2D format due to the cost constraint. They may have 2-4 sensors fitted with wide-angle/fish-eye lenses. With fewer sensors, it becomes practical to do all the image processing in a single System-on-Chip (SoC) solution. This processing includes the capture, input image processing in the ISP, de-warping of the inputs to correct for the geometric distortion caused by the optics and stitching together of the inputs into a single canvas and encoding the canvas using real-time video encoding hardware. In terms of configurations, the most popular configuration for the consumer 360-degree cameras is 360-degree/mono using a dual fisheye setup, with 2 sensors fitted with fish-eye lenses covering 185-200° FOV in opposite directions (5-20 degrees of overlap across the 2 cameras). Another configuration adopted by some of the products for 360-degree/mono is the tetrahedron configuration with 4 sensors. A practical canvas size for these cameras is 4K at 30 frames per second.

QTI's SoCs can support simultaneous real-time processing of 2 Bayer sensors, and have powerful DSPs and GPUs to de-warp and stitch the content into a canvas. They can encode the video at 4K30. Some processors can encode at 4K60 also. Hence, QTI chips are well-suited to consumer VR content creation.

VR Video Content Creation Pipeline

VR video content creation flow is quite involved and includes a variety of technology components. These components are synchronized working of multiple image sensors (w.r.t. time, exposure), color equalization across these multiple inputs, stitching algorithms (static and dynamic) and video compression. Figure 1 shows the high-level flow for a typical VR content creation pipeline.

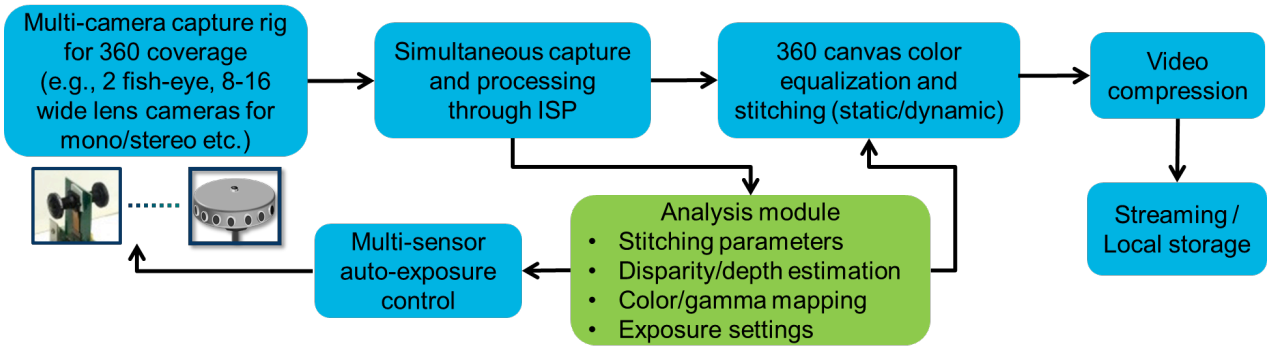

Figure 1: VR content creation high-level pipeline flow

There can be variety of capture setups involving multiple sensors (dual fisheye back-to-back, multiple wide-angle camera rig etc.) which enable the capture of complete 360-degree FOV. Note that it is important to have good amount of overlap (at least 10-20 degrees) in the FOV captured in adjacent cameras in such rigs. The overlap regions are used for global color correction, calibration, seam placing and blending. Figure 2 shows the visualization of the stitched canvas and overlaps with an imaginary 4 camera set up. The other dimension in capture setups is stereo (3D/depth) information which generally doubles the number of camera modules required in order to maintain similar resolution as mono 360-degree setups.

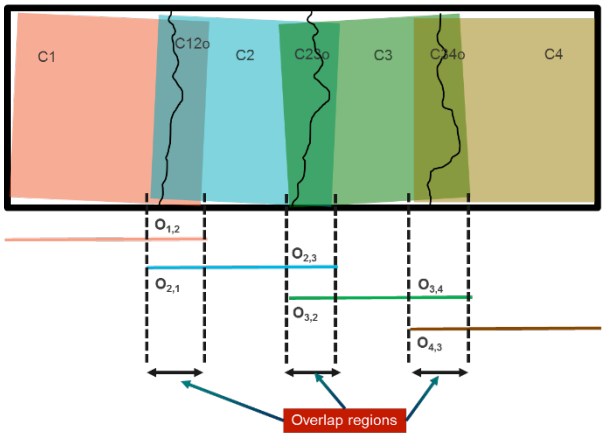

Figure 2: Visualization of the stitched canvas

The output from the sensors in 360-degree capture rig are processed in the image signal processor (ISP), where basic image processing operations including de-mosaic, denoising, color correction and sharpening are performed. The processed outputs from each of the sensor are then fed to stitching algorithm for generating the 360-degree canvas. The stitching algorithm relies on the camera undistortion and calibration parameters and seam location within the overlap region captured from adjacent sensors. This information is provided by the dynamic analysis module shown in green color in Figure 1. In later sections, we cover the details of calibration and stitching process. The analysis module is also responsible for disparity/depth estimation in case of stereo 360-degree rig, and these estimates will be used for reconstructing the left and right eye views (with appropriate baseline). The stitching algorithm will be required to work on left eye views (covering entire 360-degree span) and right eye views (covering entire 360-degree span) in a synchronous manner (w.r.t. time, seam and color).

Besides the stitching of inputs from all sensors, the other key component is to make sure brightness and color are consistent across the seam. The brightness equalization to some extent is taken care of by enabling joint exposure control approach. The color equalization can be achieved by generating gamma/tone maps based on statistics captured from each sensor input and dominant illuminant estimation. In the next chapter, we will cover details on brightness and color synchronization schemes. The stitched frames are fed to video encoder for compression and the output bitstream can be streamed to a server/HMD directly and/or can be stored locally in the capture device itself. In case of stereo rigs, the left and right view stitched frames can be video compressed into two separate bitstreams. Alternately, multi-view video encoding could be used to compress the left and right views in a more efficient manner as well.

Camera Synchronization

A typical 360-degree VR content capture system would typically have multiple cameras capturing independent streams, which would be stitched to create immersive VR content. As the camera sensors in the multi-camera system are pointing towards different directions, each sensor may potentially observe a different set of lighting and illumination conditions. In a default configuration, the ISP processing pipeline would therefore independently choose parameters (exposure, white balance) to suit its specific observations across its sensor's FOV. As a result, where two adjacent cameras overlap, the overlap region would look visually different in the two sensors, in terms of general brightness and tone (see Figure 3 for the stitching artifacts, when no equalization is performed). To create an immersive or panoramic view, one needs to match the images by equalizing the brightness and the colors across images before they are merged together so that the stitching boundaries are not visible. Two problems must be addressed here:

1. Brightness mismatch due to difference in exposure settings
2. Color mismatch across cameras because of different illuminants (hence, different white balance) across different cameras

A possible solution requires multiple sensors, with the sensor timing and capture parameters controlled jointly across the sensors for best results. The camera ISP parameter optimization approach (e.g. auto-exposure, auto-white balance) must be re-designed for panoramic video. Current QTI SOCs can support concurrent processing and computation of statistics from two Bayer sensors, which is then subsequently used for joint auto-exposure and auto-white balance control. One example camera system which can be easily supported in this manner, would comprise of two fish-eye modules (with FOV greater than 180-degrees) setup in back-to-back manner.

Figure 3: Visible seams in the content captured and stitched (8-camera system)

We achieve brightness and color equalization by a combination of techniques. Firstly, we propose to perform joint 2A (auto exposure and auto white balance) synchronization across the cameras, so that the images captured by multiple cameras do not differ much in terms of brightness and color. This synchronization can be achieved by looking at the image statistics from all the images to decide the exposure and white balance settings for the multiple cameras. The proposed algorithm can be easily implemented in our 3A engine (which anyway looks at the statistics collected from individual cameras to decide the control settings for that camera). Although this 2A synchronization would equalize the different camera captured images to a great extent, in order to equalize the images completely, we would require a post processing step to be performed. For the post-processing, our proposed method identifies “luminance correspondences” between overlapping areas of two adjacent images, and then determines the color correction function (see details of non-linear parameterization of color space in Reference 1), based on statistics captured in the registered overlap regions. Additional smoothening step is employed at the end to make the image look continuous across the overlap region. Figure 4 shows the output with this proposed color equalization strategy. The color consistency across the seam region is much better compared to when no equalization is performed in Figure 3.

Figure 4: Stitched output (from 8-camera setup) post color equalization approach

Camera Calibration

Camera calibration involves estimation of intrinsic parameters of each camera and extrinsic parameters or the relative poses between all cameras in the capture system. Intrinsic parameters are the ones that model the projection of a 3D point in the camera's FOV onto the image plane. These include the focal length, principal points and lens parameters including those representing the non-linear distortions introduced by the lens and the sensor. Extrinsic calibration parameters (rotation-translation) map the camera's position in a world coordinate system. This is useful in determining the relative positions of multiple cameras in the capture system so that the images captured by different cameras can be registered (aligned) correctly.

Accurate intrinsic calibration ensures that distortion artifacts due to imperfect lens design are minimized, where as accurate calibration of both intrinsic and extrinsic parameters ensures that stitching errors in the output panorama created from multiple camera images are minimized. Typically, the calibration needs to be performed only once for a given camera system, during the manufacturing process. In case the camera system is non-rigid, the extrinsic calibration step may need to be performed more than once – such as once every few frames or at the start of a recording session. The imaging measurement/projection model for ideal rectilinear and non-rectilinear cameras including fish-eye camera is discussed in References 2 and 3.

Calibration Flow Overview: Intrinsic and Extrinsic

Challenges involved in intrinsic calibration process is a function of the camera configuration. For VR camera capture system, it is useful to configure a system that uses wide angle cameras. These cameras typically have much wider FOV than the cameras found in typical smartphone. While the wide-angle cameras give us increased overlapping FOV between adjacent cameras, typically these cameras have larger distortions to correct for and hence a larger set of intrinsic parameters to calibrate.

An extreme example of this is fisheye cameras with > 180-degree FOV. Many existing camera calibration techniques (described in Reference 2) designed to work for conventional narrow FOV cameras. Reference 3 explains an intrinsic calibration procedure designed specifically for wide angle fisheye cameras. A planar calibration object is used for calibration purpose. A good choice is a two-dimensional array of squares (chess board) or circles for intrinsic calibration. The planar object is shown to the camera from multiple viewpoints and several pictures are taken. The intrinsic calibration flow then uses the generic projection model to project the 3D control points on calibration plane onto the image plane (using initial estimates of the extrinsic and intrinsic parameters). Position error between projected control points and actual images of control points in the image plane are measured and used as the cost metric by non-linear optimization engine. The process yields optimized set of intrinsic parameters. It is important to perform the intrinsic calibration accurately to ensure that the pose estimation in the extrinsic calibration works accurately.

Extrinsic calibration involves finding pose of two or more cameras with respect to each other. Typically, this can be done by using a calibration object that is visible to both the cameras. One can then find the pose of each camera with respect to the calibration object (discussed in Reference 4),

and thus find the relative pose between the two cameras. Another approach to pose estimation called “hand-eye calibration” (5) is widely used in robotics applications. This approach does not require the cameras to have a common FOV. Moreover, different cameras can be shown different (but static) calibration objects. For multiple fisheye VR capture systems, however, the requirement is quite different than the traditional calibration techniques. The primary goal of extrinsic calibration is to aid in achieving good quality stitching of images captured from multiple cameras. Due to highly non-linear nature of the fisheye projection model, good quality stitching requires that calibration process takes into account nonlinearities at the regions where the multiple fisheye images overlap. Control points positioned in overlapping portion (see Figure **5**) of the field of views of the two cameras are used for calibrating the cameras relative to each other using a custom model. This ensures that the objects at large distance from cameras align well and objects that are closer to the cameras have alignment errors within the expected level of parallax (which is a function of camera baseline).

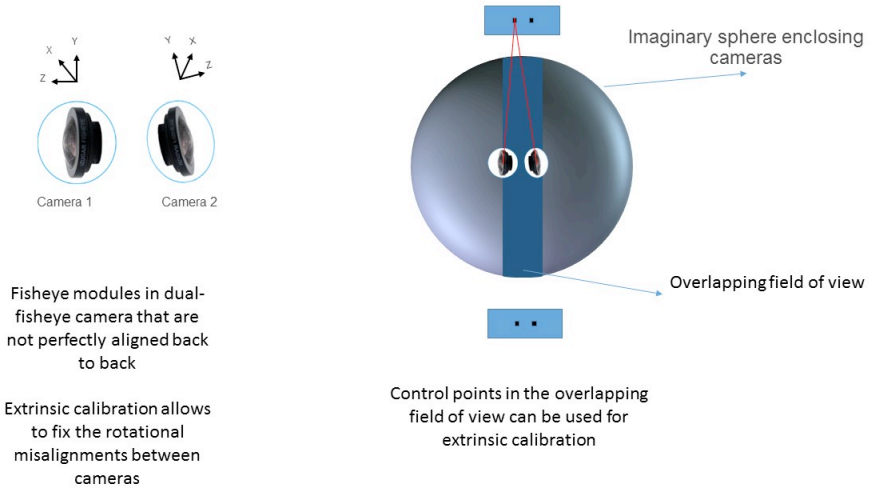

Figure 5: Extrinsic calibration for rotational alignment

Stitching

The intrinsic and extrinsic camera parameters are used to obtain calibrated images from the two fisheye cameras. The overlapping regions in the calibrated images will be used to stitch the two pictures and compute the final 360-degree output. Although stitching algorithms have been well studied in the context of creating panoramas (discussed in Reference 4(4)), 360-degree VR content creation systems have unique requirements that require careful engineering to provide a good user experience. We list some of these challenges below:

1. Parallax causes geometric misalignments in the stitched region. When viewed through a VR headset, these misalignments appear as severe artifacts in the overlap image region.

2. When viewing stitched 360-degree video content on a VR headset, immense care is required to ensure that no temporal artifacts are introduced. Image flickering in the overlapping region is very distracting to the user.

3. First person VR applications impose stringent real time requirements on the stitching algorithms.

Parallax

A key issue when combining images from multiple sensors is parallax, which is the difference in the view that a 3D object generates at two viewpoints that are spatially separated from each other. Notwithstanding the parallax, the images must be stitched together to create an output canvas with minimal perceivable artifacts. In addition, stitching requires accurate calibration, which is the precise determination of the relative location and orientation of the multiple sensors used for capture. No stitching algorithm can be perfect, for example, parallax implies that the stitched canvas will have artifacts such as edge discontinuities and ghosting in the overlap region. This is especially true when the objects in view, are close to the camera. Figure 6 illustrates these issues for a dual fisheye camera system. Here, the equi-rectangular images of the two cameras are blended uniformly in the whole overlapping image region. Since the user in the image is very close to the cameras, the parallax of the face image is large. Hence, severe ghosting artifacts are visible over these regions in the output canvas.

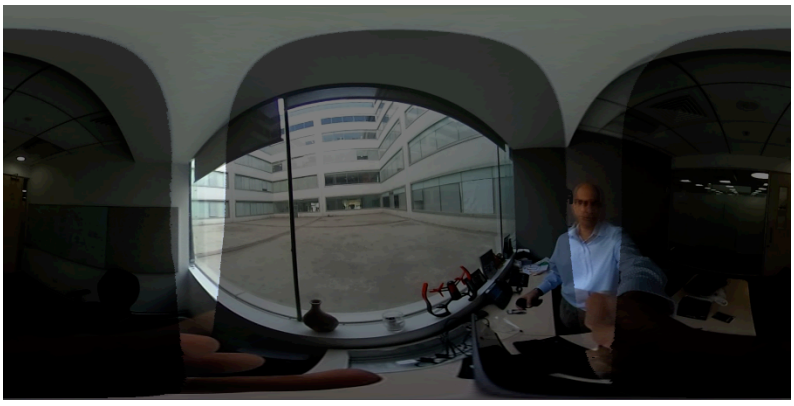

Figure 6: Illustration of stitching challenges due to parallax

We discuss three type of stitching solutions: (I) Static seam-based (II) Dynamic seam-based, and (III) Dynamic warp-based. The static seam based stitching technique has very low complexity. It performs an alpha blending of the two output equi-rectangular images around a fixed seam. When the parallax in the input images is high, the static seam technique introduces severe ghosting artifacts. To reduce these artifacts, the seam has to be dynamically chosen to avoid passing through foreground image regions. However, as we show later, dynamic seam based stitching introduces temporal artifacts. Hence, we develop dynamic warp algorithms to solve these issues. Clearly, the computation complexity required to improve the stitching performance increases when more sophisticated techniques such as dynamic warp are adopted. Figure 7 provides a visualization of these tradeoffs.

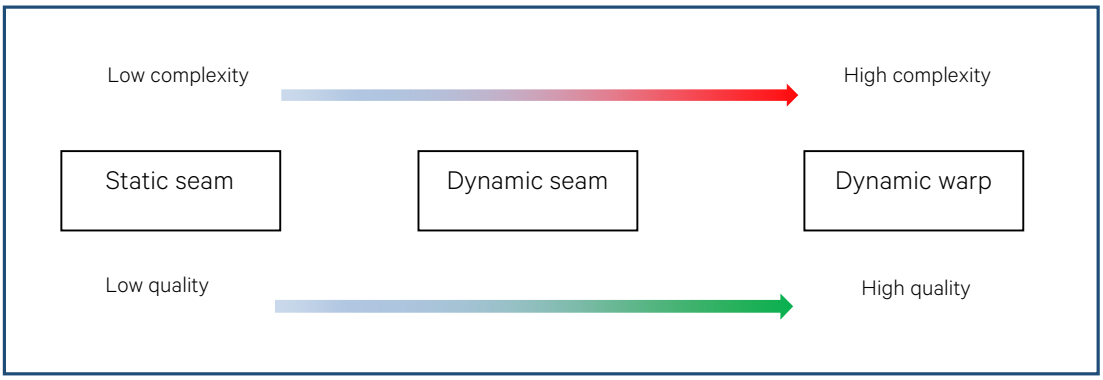

Figure 7: Stitching solutions with varying complexity and quality

Static and Dynamic Seam based stitching

Objects that are close to the camera will have parallax and therefore large stitching error. To alleviate this problem, a seam is computed in the overlap region and the two images are blended around the seam pixels as shown below in Figure 8.

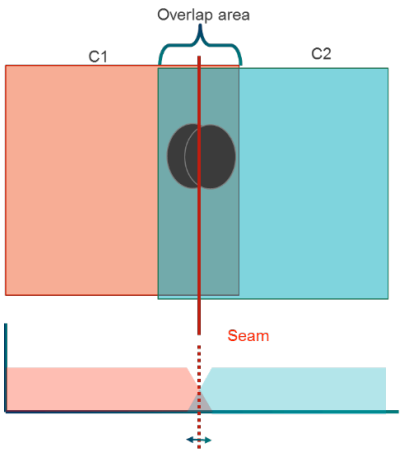

Figure 8: Seam computation and blending

The seam position is computed by minimizing a per-pixel cost function in the overlap region, where this function comprises of various penalty components including:

1. Impact of intensity difference between corresponding pixels in the overlap region from both camera images,
2. Impact of using the seam location from previous frame [temporal hysteresis], and
3. Impact of motion due to moving objects.

The cost function is minimized over the overlap window for each scanline, using dynamic programming. Seams can be either static or dynamic, for example, they can be straight lines or can curve around foreground objects. When parallax is high, static seam-based stitching will result in blending around discontinuous image structures. This results in ghosting and geometric deformations. In contrast, the dynamic seam algorithm (based on the cost function) will blend around foreground images. Static and dynamic seam stitching techniques are contrasted below in Figure 9.

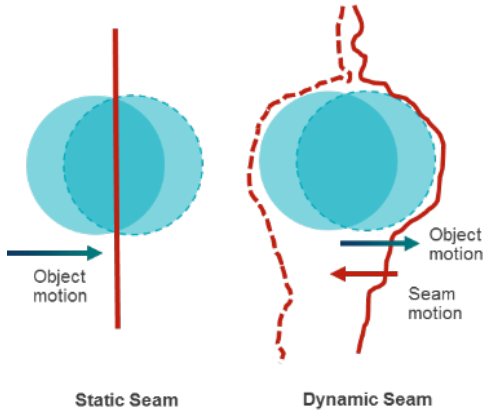

Figure 9: Static vs dynamic stitching

Figure 10 shows the dynamic seam that goes around the foreground image regions. Figure 11 shows the complete canvas obtained by performing blending around the dynamic seam pixel positions.

Figure 10: Dynamic seam curves around foreground image regions

Figure 11: Equi-rectangular image output obtained by blending around the dynamic seam

Although the dynamic seam based stitching technique alleviates some of the issues due to parallax, it is not found to be suitable for video content. It introduces temporal artifacts when applied to videos. We now describe these artifacts and discuss the dynamic warp based technique that we have developed to minimize them.

Temporal artifacts

When stitching videos, along with geometric distortions introduced in the output canvas, temporal artifacts become very significant in determining the final user experience. Unnatural changes of the structural features in the image over multiple frames cause severe distraction. As an example, when the dynamic seam algorithm is applied to a video in which foreground objects are in motion, the position of seam abruptly shifts. This can be seen more clearly in Figure 12. Here, the seam pixels marked in red are placed over the output canvas. As the pedestrian in the image walks towards the left, the seam shifts towards the right which causes a flickering artifact in the output video.

Figure 12: Dynamic seam position abruptly shifts as pedestrian walks across the scene

Dynamic warp-based stitching

To alleviate the temporal artifacts, we have developed a warp-based algorithm. Here, the image content in the overlap image is analyzed using sophisticated image processing algorithms to obtain optimal warping parameters. These parameters are used to warp and blend the two images. Figure 13 shows that the dynamic warp technique accurately stitches the images without introducing artifacts.

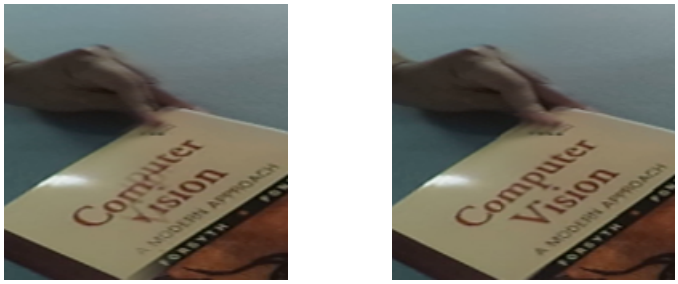

Figure 13: Left: Stitched output without dynamic warp, Right: Stitched output with dynamic warp

Results

In this chapter, we present sample results for the stitched 4K frames using the static stitching approach. The input data is captured via a two fisheye (back-to-back) camera device. The FOV captured with each lens is close to 195 degrees. Figure 14 and 16 shows example input images (indoor and outdoor respectively) captured from the two fisheye views. Figures 15 and 17 shows the stitched 360-degree outputs corresponding to the inputs shown in Figures 14 and 16 respectively.

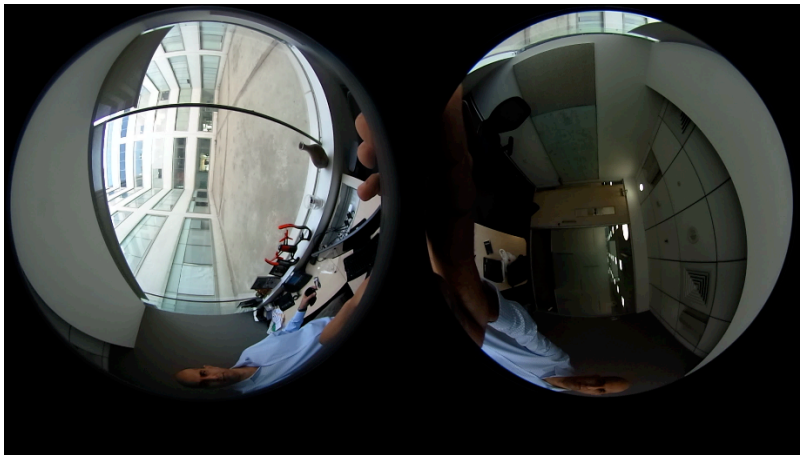

Figure 14: Input fisheye images

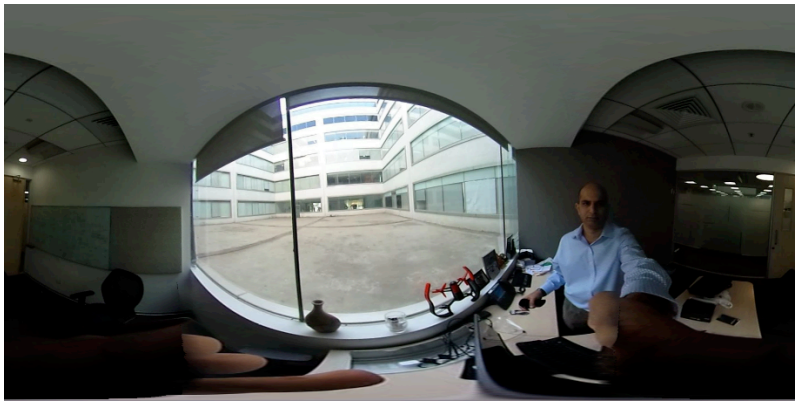

Figure 15: Stitched image corresponding to inputs shown in Figure 14

In the indoor capture, there is significant difference in the brightness condition seen across both fish-eye views, as there is significant portion covered by window with bright light and rest portion is indoors inside an office. Note that in the output stitched image in Figure 15, the brightness and color equalization appears to work well. There is some left-over color mismatch especially in the hand portion of the subject seen in this view.

Figure 16: Input fisheye images

Figure 17: Stitched image corresponding to inputs shown in Figure 16

Figure 17 demonstrates the stitched image corresponding to the outdoor captures in Figure 15, and the overall stitch quality appears very good both in terms of parallax and color/brightness consistency. Note that in order to quantify and benchmark the stitching quality, it is essential to define metrics which capture the stitching artifacts. Metrics which can capture structural similarity across the overlap region and color/tone consistency are key to this evaluation. We are in process of defining and implementing these metrics.

Conclusions

In this white paper we described some of the key technical challenges associated with VR content creation. We also presented a typical VR content creation pipeline flow and key technology components including camera calibration, static/dynamic stitching, brightness and color equalization. Sample results based on the static stitching flow are included in this paper as well, and it validates the visual quality impact with and without color equalization scheme.

Most of the focus in this paper was around monocular (2D) 360-degree stitching flow. We have real-time setup with 4K30 stitching performance on our Qualcomm® Snapdragon™ mid/high-tier platforms. Going forward there is strong focus on extending the platform capabilities to stereoscopic/3D 360-degree video creation. This should make the VR viewing experience on HMD even more immersive. The longer-term goal on video content creation side is to even develop technology components for enabling basic 6-DOF capture, i.e. enable limited parallax for small head movements while viewing in HMD.

References

(1). Y. Xiong and K. Pulli, "Color matching of image sequences with combined gamma and linear corrections," in International Conference on ACM Multimedia, Florence, Italy, 2010.

(2). "Camera calibration with OpenCV", htt*p://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.htmlExample2*

(3). Juho Kannala and Sami S. Brandt, "A generic camera model and calibration method for conventional, wide-angle and fish-eye lenses", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 8, August 2006.

(4). "Matlab camera calibrator", *http://in.mathworks.com/help/vision/ref/cameracalibrator-app.html*

(5). Radu Horaud , Fadi Dornaika , "Hand-eye Calibration"

(6). Brown, M. & Lowe, D.G. "Automatic Panoramic Image Stitching using Invariant Features" *Int J Comput Vision (2007) 74: 59*