Title

Visual Attention Modelling in 360 degree videos

Mentor

Manish Narwaria

Author

1) Devansh Bansal (B20CS094)

2) Diksha Jena (B20CS013)

Abstract

With advancing technology, 360oimages/videos have gained much popularity. Increased viewing range in such images/videos have led to severe issues in image processing of such images/videos due to limited storage and methods to transmit only 2 dimensional planar images/videos. Below we discuss the 360oimages/videos, and the problems faced during their processing, along with the solutions of some of the problems. One of the major problems is the distortion of image due to transformation into various forms, and it’s solution is discussed clearly below.

Introduction

A 360oimage/video is a multimedia type, storing views in all possible directions. It is quite similar to the panorama view with the field of view being spherical, hence occupying the entire vision of the viewer, and the viewer having control over the direction of view. It has a curvilinear perspective such that a lower amount of focus is needed in the middle and more focus is to be given to utilise the information at the bottom and top, as they will be distorted more due to dimension reduction during transmission.. Multiple devices like Nokia Oze, Ricoh Theta S, Samsung Gear 360 provide this perspective and output a stereoscopic display. Two images, one from each eye is stored for better perception. Storing of these images requires various factors to be considered like image quality, depth quality, visual comfort, resolution, information etc so that the overall image quality is the least affected and can be later rendered without any distortion.

Design Problem Formulation

Storage and Rendering of a 360oimage/video requires a more sophisticated model as compared to a 2 dimensional, planar 180oimage/video that can be easily stored as a rectangular projection. It has more information to store and the coordinates have 3 variables to follow. High Resolution and High Frame Rate are required along with High Speed Internet Connection to transfer data efficiently. Using the usual approach of converting Spherical to Planar images induces a lot of distortion and information loss, which are crucial for the image retrival. The Detection and Recognition of objects requires both Head and Eye movements to be monitored which increases complexity and differences in identification. Objectiveness is sensitive to the perspective image being sampled.

Our aim is to identify a better method to compress, store and transfer 360oimage/video. Currently the major method is to project the 360oimage/video onto a plane, and then do the usual image processing of 2 dimensional planar image/video. It has the drawbacks of data loss, and distortion. Multiple approaches of doing the same thing mentioned above are there, one of them projects the whole sphere onto a plane at once incurring lots of data loss and distortion; whereas another one of them takes subparts of the 360oimage/video, and then project them onto multiple planes, and then superimpose them to obtain the final image/video, incurring less data loss but it requires huge computation cost and faces problems in line detection and backprojection.

Possible Methods to solve the problem

● CNN models are used for training images from perspective cameras.

Existing Methods:

(1) Distorted Planar Image Projection: 360oinput is projected onto a single plane and then CNN is applied on distorted planar image to get the resultant processed image. Fast but less accurate.

(2) Multiple Tangent Planar Projection: 360oinput is sampled into multiple tangent planar projections to obtain multiple perspective images, on which applying CNN locally gives resultant processed image. Slow but more accurate.

● Visual Attention: focuses on the attractive content and use silency models.

(1) Heuristic Approach

Requires hand crafted features and is easily interpretable

(2) Data Driven Approach

Large data set is required and uses saliency models

(a) Top down approach

(b) Bottom up approach

● Filtering of data uses features like intensity, chromatic aberration, and orientation. A gaussian or gabor pyramid is used and up-scaling and normalization is done via reinforcement learning. ● The distortions can occur from both Compression and Pixel-Wise Distortion due to averaging of values to compensate for the data loss. Gaussian noise or blur box methodology can be adopted for the purpose.

● View port based coding / streaming can be used for saving high and low resolutions, along with mechanism modification and padding for motion estimation.

● Reprojection is done via rotation or decompression.

● It is also possible to store polar coordinates for mapping instead of using 3D to 2D relation for indexing and featuring and the data can itself be stored as a spherical instead of planar projection. ● Implementation of Machine Learning can also be done. There are broadly two approaches to apply

Machine Learning. One of them directly computes saliency of whole equirectangular image, and then we try to fit our model based on input (saliency mapping computed of whole image) and output (ground truth saliency mapping); whereas the other one involves some kind of projection of equirectangular image into another form or forms, and computing their saliency mapping which can further be used as input to the Machine Learning method, and output being the same as previous (ground truth saliency mapping)

Methodology adopted for the project

For the available methods, we had 2 : (i) GBVS and (ii) Itti Koch ; to compute the saliency mapping. Both the methods take in 3-D array as an input and return a 2-D array i.e. Saliency Mapping in grayscale format. To minimise the error in the mapping returned by the formats, it was studied that error is mostly occurring at the top and bottom faces due to point distortion and back face (left and right borders) due to center bias effect. The center bias effect is reduced using the saliency mapping of the flipped image of the original one, while the image is first projected to its cubemap projection, to neglect point effect.

● First, it was important to resize the image and saliency ground truth to the same size so that they can be compared easily later and to take care of any error that might occur due to pixel distortion.

● Then 3 projections were taken: cubemap, original equirectangular projection, and the flipped (all as 3-D matrix) and the saliency mapping was obtained. It was noted that the output was a 2-D matrix and range was not strictly between 0 to 225. The data was scaled (normalized) whenever the maxima or minima lay out of bounds, and an extra axis was added to convert the 2-D matrix to 3-D, as the projection functions available to us had input format of 3-D matrix.

● Apart from this, weights were added to each face of the cube mapping as it was observed that each face has different probability to be looked at, depending on its place, and thus have different levels of contribution to the ground truth.

● From the 3-mappings obtained, the pixel wise maxima was taken for the final output. ● Four measures of comparisons were adopted

(i) KLD - Average difference in the amount of bits required for encoding samples of output using code optimized for ground truth.

(ii) CC - Measure of Linear Correlation between output and ground truth.

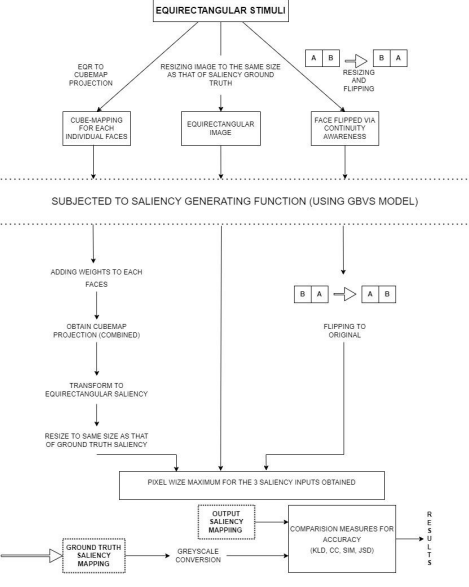
(iii) SIM - Real Valued Measure that depicts the similarity between output and ground truth.

(iv) JSD - Similar to KLD, but with additional features like Symmetricity and finiteness.

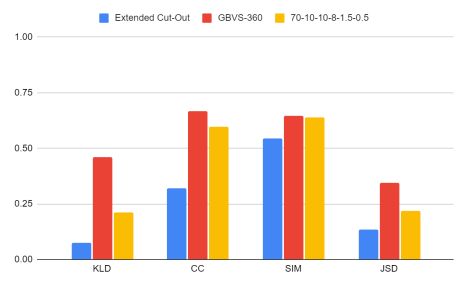
Focus was mainly on improving the KLD and CC towards ideality and the other two measures were taken as a safety measure to verify the the results were going in the right direction.

We also tried combining the extended cube cut out instead of cubemap projection, to reduce continuity errors. But due to extra smoothening of output, it increased the error in pixel-wise maximum function used to compute final output saliency, as compared to the error encountered with six cube-faces. We even tried using linear regression to compute the best possible contribution weights ratio but the results were far poorer than expected, reason being the whole idea of saliency is based on content present in the image, and Machine Learning will give results based on earlier training data-set, which may have very different content distribution in the image. We also tried to establish a content based relation between content in cube-face and it’s contribution to final output (for 4 images). We found some similarities and some differences also, as discussed in the Results and Analysis section.

The final method chosen to compute saliency was GBVS, using all the extra changes made to the image for the best output as the output seemed better as compared to the case when saliency was computed using Itti-Koch,which was not giving much accurate results for CC and SIM.



Justification of the design choices

GBVS360 and extended cut-out using both gbvs and itti koch were used to make comparisons that our model was better. As we can see in the graph above, our model (having face ratio as 70%-10%-10%-8%-1.5%-0.5% for faces in order Front-Left-Right-Top-Bottom-Back) is better than GBVS-360 (already existing model to compute saliency of equirectangular images) for KLD and JSD metrics, and for CC and SIM also there is only a subtle difference.Whereas when we compare our model with Extended Cut-Out model which involves duplicating the cube faces around themselves along with rotation of some of them, CC and SIM seem better for our model, whereas KLD and JSD are better for Extended Cut-Out Model.

GBVS360 methods is an extension of the regular GBCS approach, with the modification of working on 3-D image instead of 2-D by integrating the specific properties of equirectangular images to model visual attention in omnidirectional images watched using Head Mounted Displays. The extended cutout model does reduce the point and center bias distortion but also, does not take into account the face contributions and content effect that is likely to have an effect on the output. Thus proving our method to be better to the other two.Above results can be attributed to metric calculation method i.e. KLD and JSD work on pixel-wise data, whereas CC and SIM work on the complete image as a single entity.

Along with this, we can say that extracting each cube-face and then applying a particular ratio to each of them is valid and justifiable because of the way human viewing technique has evolved over the years. We mainly focus on our front viewing zone as compared to our backside, and other sides also have a similarly corresponding ratio. Consider yourself being sent into a completely new room for a time period of 10-12s (orientation in which you enter remains fixed after entering, you can view in all directions using head and eye movements only, no movement allowed for body lower than neck), then on an average you will focus more than 5-6s on front side of yourself only, and just a second or like that viewing the back-side of yourself.

If you are sent into the same room for a longer duration like 40-50s, then your viewing would be more content based rather than just head-eye-movement easiness. Though in that scenario also, our model works great up to a certain extent, since then also, a quite large amount of time will be spent viewing the front side only (though it can be less than 50% for a long period of time).

Results and Analysis

|  | KLD | CC SIM JSD |
| --- | --- | --- |
| Extended Cut-Out | 0.07605975965 | 0.3215949218 0.5445205242 0.1354766676 |
| GBVS-360 | 0.461961269 | 0.6683108717 0.6444546303 0.3460351296 |
| 70-10-10-8-1.5-0.5 | 0.2127789315 | 0.5965382767 0.6383505787 0.2175098121 |

The table gives the relation measures of outputs obtained when compared to the grayscale version of the ground truth saliency mapping. As noticeable the GBVS method along with face wights introduced is the best in an overall comparison, even though the KLD and JSD for extended cut-out is very close to ideal.

| Image | Ratios (F-B-U-L-R-D) | KLD CC SIM | JSD |
| --- | --- | --- | --- |
| 8.jpg | 0.226-0.205-0.065-0.191-0.2 42-0.071 | 0.163117838 0.7740433832 0.6651314201 | 0.1982372927 |
| 52.jpg | 0.274-0.167-0.1-0.221-0.164- 0.074 | 0.2119957097 0.5916757376 0.6369492245 | 2.220471044 |
| 63.jpg | 0.14-0.226-0.169-0.107-0.25 3-0.105 | 0.116606128 0.6278830792 0.7492873369 | 0.1693880769 |
| 97.jpg | 0.213-0.201-0.077-0.158-0.2 54-0.097 | 0.2010173699 0.5684733479 0.6451101267 | 0.2206078661 |

The face ratios chosen for the images were initially based on the face position, but no measure was taken into account for content basis. For example the image 97 is a photo of a supermarket counter, and the person is more likely to look front and down as compared to a usual basis. Similar situations can be encountered where the face ratio is more dependent based on the situation. Thus 4 images were selected to verify the same.

The above table gives the relation measures obtained when face-ratio was applied based on content, brightness, color contrast and many other features present in a particular face on 4 different images. These four images were chosen specifically for this purpose as they were not having major content along the front-face, and hence deviated slightly from our early model since we were viewing these 4 images for a longer duration. After applying the new ratio based on content and which was different for each of the four images, we can see that content-based ratio seems much better when we want a longer duration, and generalized ratio seems much better when we view for comparatively shorter duration of time.

Conclusions

While handling 3D images being converted to 2D images, the continuity was broken, so that needed to be corrected while calculating the final saliency output. Moreover as our human eyes and the viewing mechanism of humans have evolved, face ratio was a good idea to be thought of and used to weigh each face for final saliency output. Also averaging various results may drop the value and essence of a correct result, so pixel-wise maximum was also a good point for final saliency output.

Further new approaches can be implemented, some of them maybe non-linear regression (though it’s still based on data-fitting; it should be better than linear regression), or one may even also work with object identification in an image, and then weigh that face (content-based ratio) for more accurate results.

References

1) https://www.researchgate.net/publication/335500379\_Advancing\_User\_Quality\_of\_Experience\_i n\_360-degree\_Video\_Streaming

2) https://support.apple.com/en-in/guide/motion/motnfccf9e10/mac 3) https://arxiv.org/pdf/1905.00161

4) https://www.researchgate.net/publication/344837879\_360-Degree\_Video\_Streaming\_A\_Survey\_ of\_the\_State\_of\_the\_Art

5) https://repository.kulib.kyoto-u.ac.jp/dspace/bitstream/2433/171749/1/IEICE\_E96-D\_3\_562.pdf 6) http://www.cvlibs.net/publications/Coors2018ECCV.pdf

7) https://michaeldorr.de/salient360/360\_aware\_preprint.pdf

8) https://github.com/marcellacornia/sam

9) http://www.vision.caltech.edu/harel/share/gbvs/

10) https://github.com/sunset1995/py360convert

11) https://drive.google.com/file/d/1QEYkq9YCwe8egE6RgQtXCE43XbluFPAB/view?usp=sharing