Free view synthesis

Varsha Pendyala

Multiple views with stereo cameras

- Except the focal length, all other intrinsic parameters can be found using the standard calibration procedures where we use an external calibration pattern like chessboard.
- Also, any distortion parameters can be found once, and they will remain same forever.
- However, focal length is the one which may not remain constant over the entire imaging process in any application.
- Hence, we use the auto-calibration framework to obtain the focal lengths of the cameras.

- We have seen how auto-calibration is done for a 3 camera network where two of them form a stereo pair with known baseline separation.
- We can make use of that framework for this application of view synthesis.
- In case of view synthesis applications using multiple stereo views, we would also have rotation and translation parameters of the cameras.
- If we use the two stereo views to create an intermediate view:
- Unknown parameters: f1, f2, f3, f4
- Known parameters: external parameters of the stereo pairs (since we know their baseline separation and their placement)
- By considering stereo images from one stereo pair and images of one of the cameras of other stereo pair, we arrived at the same arrangement that we know already.
- Result of this calibration: {f1, f2, f3}
- Similarly, can we find f4?

- Image formation by mono-camera:

$$s_{3j}\bar{x}_{3j} = \begin{bmatrix} f_3 & 0 & 0 \\ 0 & f_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} r_1 & r_2 & r_3 & t_X \\ r_4 & r_5 & r_6 & t_Y \\ r_7 & r_8 & r_9 & t_Z \end{array} \right] \bar{X}_j$$

- Similarly, for the fourth camera(Assuming each camera of stereo pair has same rotation matrix):

$$s_{4j}x_{4j} = f_4\left(r_1X_j + r_2Y_j + r_3Z_j + (t_X - a)\right) \quad \text{----------} \ (1)$$

$$s_{4j}y_{4j} = f_4\left(r_4X_j + r_5Y_j + r_6Z_j + (t_Y - b)\right) \quad \text{-----------} \ (2)$$

$$s_{4j} = r_7X_j + r_8Y_j + r_9Z_j + (t_Z - c) \quad \text{----------} \ (3)$$

where, a, b, c depends upon the baseline separation

- Dividing (1) by (3):

$$x_{4j} = \frac{f_4(r_1X_j + r_2Y_j + r_3Z_j + (t_X - a))}{r_7X_j + r_8Y_j + r_9Z_j + (t_Z - c)}$$

- Hence, f_4 is found using the above equation.

Using the results of calibration to perform view synthesis:

- Even, if there were many non-overlapping regions in the two stereo pairs of images, we will have enough correspondences to find out the unknown parameters.
- Once, we have these unknown parameters we can deal with each individual stereo pair separately.
- Because each stereo pair of images will have almost complete overlap.
- This will help us in creating a dense disparity/ depth map.

DIBR

- Since we have the depth maps for each stereo pair, we can use DIBR to create an intermediate view.
- At this point, we can treat a stereo pair of images as image plus depth information and thus we can use the DIBR formulation as follows:
- An original view $'k'$ is given by

$$I_k(u_k, v_k),\ \ d_k(u_k, v_k)\ \tilde{P}_k$$

Where $\tilde{P}_k = K_k[R_k|t_k]$. Consider $\widetilde{P}_k = [P_k|p_k]$

- Each (u_k, v_k) is projected into 3D space as follows:

$$s\_k \begin{pmatrix} u_k \\ v_k \\ 1 \end{pmatrix} = [P_k|p_k] \begin{pmatrix} \bar{X} \\ 1 \end{pmatrix}$$

$$\bar{X} = \mathrm{P_k^{-1}} \left(s_k \begin{pmatrix} u_k \\ v_k \\ 1 \end{pmatrix} - \mathrm{p_k} \right)$$

- In the next step, the 3D point is forward projected into the intermediate view.

$\begin{pmatrix} u_\lambda \\ v_\lambda \\ 1 \end{pmatrix} = \mathrm{P}_\lambda \begin{pmatrix} \bar{X} \\ 1 \end{pmatrix}$ where P_λ corresponds to the parameters of the new view point and $\bar{X}$ is the 3D representation of the scene point.

- For obtaining the colour at (u_λ, v_λ), all colour values $I_k(u_k, v_k)$ from view $'k'$ that map onto position (u_λ, v_λ) are collected.
- Next, the front most pixel with minimum projected depth $z_{min,k,\lambda}$ is selected.
- For the colour projection, the associated position $\left(u_{k,min}, v_{k,min}\right)$ in the original view is found as follows:

$$\left(u_{k,min}, v_{k,min}\right) = \underset{\forall u_k, v_k}{argmin} \left\{ z_{k,\lambda,u_k,v_k}(u_\lambda, v_\lambda) \middle| \begin{pmatrix} u_\lambda \\ v_\lambda \\ 1 \end{pmatrix} = \mathrm{P}_\lambda \mathrm{P}_\mathrm{k}^{-1} \left(s_k \begin{pmatrix} u_k \\ v_k \\ 1 \end{pmatrix} - \mathrm{p}_\mathrm{k} \right) \right\}$$

- The colour contribution $I_{\lambda,k}(u_\lambda, v_\lambda)$ from view $'k'$ in the intermediate view: $I_{\lambda,k}(u_\lambda, v_\lambda) = I_k\left(u_{k,min}, v_{k,min}\right)$.
- The above process is repeated for view $'n'$ to obtain the colour contribution $I_{\lambda,n}(u_\lambda, v_\lambda) = I_n\left(u_{n,min}, v_{n,min}\right)$
- The general intermediate view interpolation between views k and n can be formulated as follows:

$$\begin{aligned} I_\lambda(u_\lambda, v_\lambda) = (1-\lambda).\, I_k\left(u_{k,min}, v_{k,min}\right) \\ + \lambda.\, I_n\left(u_{n,min}, v_{n,min}\right) \end{aligned}$$

References

- https://drive.google.com/drive/u/0/folders/0BwwIhuCdkbWYNTZXQ0NPVl9Sbk0
- http://jivp.eurasipjournals.springeropen.com/articles/10.1155/2008/438148
- https://www.disneyresearch.com/publication/3d-video-and-free-viewpoint-video-from-capture-to-display/
- https://ecommons.cornell.edu/bitstream/handle/1813/7259/96-1604.pdf?sequence=1