

Research Log

JeffGWood@mavs.uta.edu

May 23, 2016

March 30, 2016	Established research log after 3 hours of learning new \LaTeX
----------------	--

April 2, 2016	Added some additional comments to the Process
---------------	--

April 3, 2016	<p>Have been reading [ImageBasedRendering] [1].</p> <p>Question for Kamangar: regarding [ImageBasedRendering] [1] about difference between:</p> <ul style="list-style-type: none">• Camera Plane : Coordinates u, v• Focal Plane : Coordinates s, t
---------------	---

April 11, 2016	<p>Reviewing blog articles located at:</p> <ul style="list-style-type: none">• https://erget.wordpress.com/2014/02/01/calibrating-a-stereo-camera-with-opencv/• https://erget.wordpress.com/2014/02/28/calibrating-a-stereo-pair-with-python/• https://erget.wordpress.com/2014/03/13/building-an-interactive-gui-with-opencv/• https://erget.wordpress.com/2014/04/27/producing-3d-point-clouds-with-a-stereo-camera-in-opencv/ <p>for process to get webcam up and running. Previous issues related to fine-tuning <i>block matching</i> parameters. Need to review sources at list at bottom of http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html to understand.</p>
----------------	---

April 19, 2016	<p>Made adjustments to python for image acquisition scripts (from blogs mentioned on April 11, 2016.)</p> <p>NOTE: Consider creating rig with glue to keep stereo camera placement / direction constant.</p>
----------------	---

April 19, 2016	<p>UPDATE: Error with <code>calibrate_cameras</code> python code causing linux machine to crash. If can't be resolved switch over to MacBook.</p> <p>NOTE: Package should be setup by calling <code>\$ python setup.py install</code></p>
----------------	---

April 19, 2016	<p>UPDATE: Crash due to recursive shell call and was fixed. OpenCV not detecting all chessboard corners. Will try a new board.</p>
----------------	---

April 20, 2016	<p>Did small amount of work on Change of Reference section in the paper. Added a section to the intro containing a map of commonly used symbols and notation</p>
----------------	---

April 29, 2016	<p>Read following sections of [Chen93] [2]:</p> <ul style="list-style-type: none"> • Abstract • Introduction • Visibility Morphing <p>SUMMARY: Explicit Geometry is ignored (i.e. surface mesh and 3d-points). Geometry is kept in 2-d. Whereas Image Morphing interpolates between <i>pixel intensity values in fixed locations</i> the method in this article interpolates between <i>pixel locations with (relatively) fixed intensity values</i>. Question: Sections read mention that pixel positions are stored in 3d (3-tuple) data structure. I'm not sure I understand this correctly, since</p> <ol style="list-style-type: none"> 1. This would effectively make this structure a point cloud (but no mention of it in the paper). 2. There is no mention of special "depth-based" hardware or cameras (Far as I know this is upposed to be a regular image).
April 30, 2016	<p>Checked understanding of <i>epipolar constraint</i> through reading of [Hartley2004] [3] and its derivation of</p> $\begin{aligned} \mathbf{x}'^T \cdot \mathbf{E} \cdot \mathbf{x} &= \mathbf{x}'^T \cdot [\mathbf{t}]_{\times} \cdot \mathbf{R} \cdot \mathbf{x} \\ &= \mathbf{x}'^T \cdot \mathbf{l} \end{aligned}$ <p>and creation of MatLab code verifying this.</p> <p>I may have been mistaken about relation of Fundamental Matrix and Essential Matrix.</p> <p>My current understanding is the <i>Fundamental Matrix</i> describes point/epipolar line correspondance for images under scale invariant conditions (i.e. point correspondance and Fundamental matrix does not change when one image (or both images) are scaled (uniformly or omni-directionally)).</p> <p><i>Essential Matrix</i> describes point/epipolar line correspondance for images under normalized conditions (i.e. unit-length is set equal to focal-length, and projection center is set at (0,0,1)).</p>
May 2, 2016	<p>Additional wording to Stereo-vision section. I am unsure of best order to present ideas related to <i>multi-view</i> geometry.</p>
May 18, 2016	<p>Reviewed [Chen93] [2] Section 2. Consider reviewing follow relevant articles:</p> <ul style="list-style-type: none"> • Disparity [Gosh89] • Optical Flow [Nage86] • Look-up tables [Wolb89] • 3d scenes [Pogg91] <p>Working on MatLab code to pick correspondig points in stereo-images, and calculate pixel offset vectors.</p>
May 19, 2016	<p>Read Section 2.3 of [Chen93] [2]. View interpolation is limited by:</p> <ul style="list-style-type: none"> • Penumbra: pixels visible in one source image <i>but not both</i> • Umbra, pixels visible in neither source image, and <i>invisible</i> in destination image. • Holes, pixels visible in neither source image, but <i>visible</i> in destination image. <p>Calclatred formula for <i>pre-displaced</i> quad-pixel calculation using a bi-linear interpolation as:</p> $\mathbf{P}(u, v) = \mathbf{P}(0, 0) \cdot (1-u) \cdot (1-v) + \mathbf{P}(1, 0) \cdot u \cdot (1-v) + \mathbf{P}(0, 1) \cdot (1-u) \cdot v + \mathbf{P}(1, 1) \cdot u \cdot v$

May 20, 2016

Derived formula for uv calculation using *geometry matrix*, *blending matrix* and *basis vectors* of $\mathbf{u} = [u \ 1]^T$ and $\mathbf{v} = [v \ 1]^T$

$$\begin{aligned} x_{uv} &= [v \ 1] \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_{00} & x_{10} \\ x_{01} & x_{11} \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} \\ y_{uv} &= [v \ 1] \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y_{00} & y_{10} \\ y_{01} & y_{11} \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} \end{aligned}$$

Question for Kamangar: Is there a way given x and y to solve for u and v ?

May 22, 2016

Added more to thesis document.

Worked on singular-value of previous blending equation. where:

$$\begin{bmatrix} x_{uv} & 0 \\ 0 & y_{uv} \end{bmatrix} = \begin{bmatrix} \mathbf{u} & \mathbf{0} \\ \mathbf{0} & \mathbf{u} \end{bmatrix}^T \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{G} \end{bmatrix}^T \begin{bmatrix} \mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y} \end{bmatrix} \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{v} & \mathbf{0} \\ \mathbf{0} & \mathbf{v} \end{bmatrix}$$

where

$$\mathbf{u} = \begin{bmatrix} u \\ 1 \end{bmatrix}, \mathbf{v} = \begin{bmatrix} v \\ 1 \end{bmatrix}, \mathbf{X} = \begin{bmatrix} x_{00} & x_{01} \\ x_{10} & x_{11} \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} y_{00} & y_{01} \\ y_{10} & y_{11} \end{bmatrix}, \text{ and } \mathbf{G} = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}$$

May 23, 2016

Read [Chen93] [2] section 2.4 on *Block Compression*.

SUMMARY: Blocks are established by *threshold* where each block contains pixels that are *offset by no more than the threshold*, allowing all pixels to be offset at once.

Question for Kamangar: Doesn't this assume that all pixels in the block have a uniform offset?

Working on MatLab program to perform pixel offsets of corresponding points (i.e. assign corresponding points to pixels in MatLab by non automatic methods)

References

- [1] Sing Bing Kang Heung-Yeung Shum, Shing-Chow Chan. *Image Based Rendering*. Springer Publishing, 1 edition, 2007. Available online at: <http://link.springer.com/content/pdf/10.1007%2F978-0-387-32668-9.pdf> Pages cited are **Book Page** Numbers. Formula for **PDF Page** Number is (**PDF Page Number** = **Book Page Number** + 17).
- [2] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '93, pages 279–288, New York, NY, USA, 1993. ACM.
- [3] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.