Salient Frame Extraction for Structure from Motion

Whitehead, A.; Roth, Gerhard

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

NRC · CNRC

Frame Extraction for Structure from Motion *

Whitehead, A., Roth, G.
September 2001

* published in the Proceedings of Visualization, Imaging and Image Processing. pp. 658-663, Marbella, Spain. September 2001. NRC 45868.

Canada

Salient Frame Extraction for Structure from Motion

Anthony Whitehead*
School of Computer Science
Carleton University
awhitehe@scs.carleton.ca

Gerhard Roth
National Research Council of Canada
gerhard.roth@nrc.ca

* Partially funded by Nortel Networks

ABSTRACT

We propose a key frame extraction mechanism to aid the Structure from Motion (SfM) problem when dealing with image sequences from video cameras. Due to high frame rates (15 frames per second or more) the baseline between frames can be very small and the number of frames can become unpractical to deal with effectively. The mechanism described in this paper is a preprocessing step designed to make an ideal image sequence from larger sequence of video frame data. Based on a proven tracking mechanism, the algorithm remains quite simple yet effective for identifying and extracting salient frame data for the SfM problem in effect removing degeneracy cases.

KEYWORDS: Image sequence acquisition, Projective Computer Vision.

1. INTRODUCTION

There has been much progress in the field of uncalibrated projective vision in the last decade. Algorithms have become mature and robust to the point where the creation of a practical and complete system has become feasible. One such system, shown in figure 1, would be the recovery of camera positions and scene structure of various objects from a single video sequence. While there has been significant advancement in the process of automatically going from an uncalibrated image sequence to a three-dimensional model [1-16], little work has been devoted to effective creation of the image sequence in the first place. In order to build a complete system one needs to consider the entire problem, from data acquisition to model generation and output. Only when the entire process has been addressed, can effective, practical and reliable systems be built. Due to the wide availability and simplicity of use, video cameras are ideal image acquisition devices. The high frame rate ensures that full coverage of the scene is possible, however this advantage is surprisingly a disadvantage as well. The large volume of frame data is not only impractical to process in a timely manner, but the minute baselines between frames can also cause problems during the bundle adjustment phase of the structure from motion (SfM) algorithms.

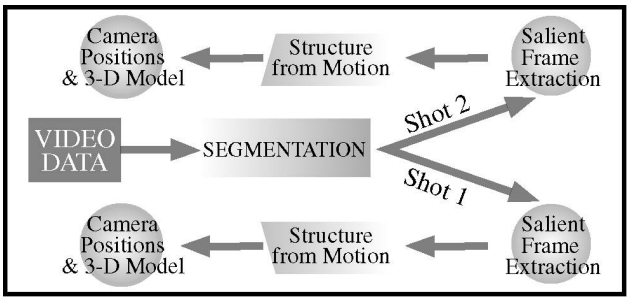

Figure 1: A complete system for going from video data to reconstruction

The obvious approach of regular frame sampling (effectively reducing the frame rate) shows its inadequacies quickly. Due to banding and interlaced video, the regularly selected frames may not be ideal for image processing. Another problem is that frames selected in this manner have not been picked for their suitability to the SfM problem, but rather on a frame rate assumed to be good. The SfM algorithms work best on images with large overlap to allow for good feature matching yet significantly large baseline to ensure parallax large to enough to keep the problem well conditioned. By simply changing the frame rate it is clear that the images produced by this method may cause the SfM algorithms to be ill conditioned. High frame rates increase the chance that parallax will not be sufficient and low frame rates reduce the amount of overlap required to adequately match features. Clearly selecting a fixed frame rate is not an effective approach to salient frame extraction for the SfM problem. In fact, the ideal frame rate turns out to be variable, depending on the two factors that make the SfM problem well conditioned: overlap and parallax.

A good selection of frames from a video sequence can produce a much better set of input images for the SfM algorithms and therefore ensure a more reliable reconstruction. By extracting salient frame data from a large sequence we are in effect reducing the size and time requirements for the system as well as ensuring the reconstruction is likely to be well conditioned. This gives the SfM phase of the system input governed by a preprocess that ensures the data is ideal for such algorithms. This in turn takes the data acquisition frame rate, camera motion, and user error out of the system.

This paper continues by briefly describing the requirements for the preprocessing routines, and by going into each component in greater depth. Finally some results are examined and conclusions are presented.

3 Preprocessor Requirements

In order to effectively deal with the vast amounts of data that video presents us, it is important that the preprocessor maintain some stringent requirements. Most importantly, the preprocessor must perform its work with a single pass through the video data. Multiple passes through large volumes of data lacks scalability and prevents any future for real-time embedded frame extraction into video capture hardware.

As a preprocessing mechanism, we must keep in mind that there may be other processing mechanisms further down the processing chain that requires the original unaltered data, thus it is also important that the method is read-only and does not alter the original frame data. The process should also be capable of detecting shot boundaries. In the absence of any other boundary detection mechanisms this mechanism will detect the boundaries. Finally, it is necessary that the preprocessor be frame rate independent. Whether the video input is 15 frames per second (fps), 24 fps, or 30 fps, (all common frame rates), the preprocessor must be able to detect appropriate frames.

4 Boundary Detection

In order to keep the preprocessor robust, it is important that different sequences be segmented. Multiple shot sequences could occur when the operator of a hand-held video camera stops one sequence and starts a second immediately following the first. Many modern cameras can provide this information since the technology for scene detection is so mature that it has been implemented into many consumer level video cameras. However, providing such information to the preprocessor may not be an option depending on the input source.

Much work has been completed in the area of scene detection, shot detection and annotation and as a result, the methods and algorithms are quite mature. In 1965, Seyler [17] developed a frame difference encoding technique for television signals. The technique is based on the fact that only a few elements of any picture change in amplitude in consecutive frames. Since then much research has been devoted to video segmentation techniques based on Seyler. A variety of metrics have been suggested to work on either raw video or compressed data. These metrics are used to quantify the difference between two adjacent frames and can be further sub-classified into 4 major categories:

- Pixel-Level Change Detection [18,19,20]
- Histogram Change Detection [18,19,20]
- DCT Change Detection [21,22,23]
- Subband Feature Change Detection [24]

The basic concept of shot/scene detection is to evaluate the similarity of adjacent frames using one of the aforementioned methods. When the similarity measure crosses a certain threshold, a scene change or shot boundary has been detected. Equations (1) and (2) below describe a pixel level change metric.

$$DI_i(x,y) = \begin{cases} 1 & \text{if } |I_i(x,y) - I_{i+1}(x,y)| > t \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$[\sum_{x,y=1}^{X,Y} DI_i(x,y)] / X*Y > T \quad (2)$$

In (1), we compute the difference between pixel values between image i and i+1 and create a difference image DI, where t is a threshold signifying individual pixel difference. We then compute the overall image difference using (2). If the percentage of image change is greater than our threshold T, we declare a shot boundary.

Any of the aforementioned metrics is sufficient for this step, however they rely on certain types of video, MPEG compressed, for example. The pixel level detection metric displayed in (1) and (2) is the most basic form for raw, uncompressed video which is easily achieved regardless of the input. Suitable values for the thresholds t and T are 8 and 0.75 respectively. Depending on the precision requirements and time requirements, one may find another metric more suitable.

5 Motion Estimation and Feature Tracking

The feature tracker we use is based on the early work of Lucas and Kanade [25] that was developed fully by Tomasi and Kanade [26], Shi and Tomasi provide a complete description [27] that is readily available. Recently, Tomasi proposed a slight modification, which makes the computation symmetric with respect to the two images; the resulting equation is fully derived in the unpublished note by Birchfield [28]. Briefly, features are located by examining the minimum eigenvalue of a 2x2 image gradient matrix that is noticeably very similar to the Harris corner detector [29]. The features are tracked using a Newton-Raphson method [30] of minimizing the difference between the two windows.

We continue by presenting a very brief outline of the work by Tomasi etal [25,26,27,28]. Given a point p in an image I, and its corresponding point q in an image J, the

displacement vector δ between p and q is best described using an affine motion field:

$$\delta = Dp + t \tag{3}$$

where

$$D = \begin{bmatrix} d_{xx} & d_{xy} \\ d_{yx} & d_{yy} \end{bmatrix} \tag{4}$$

is a deformation matrix and t is the translation vector of center point of the tracked feature window. The translation vector t is measured with respect to the feature in question. Tracking feature p to feature q is simply determining the six parameters that comprise the deformation matrix D and the translation vector t.

Clearly in the case of pure translation, D will be the identity and thus

$$\delta = p + t \tag{5}$$

Because of this, the case of pure translation is computationally simpler and thus preferable. Since the motion between adjacent frames of standard video is generally quite small, it turns out that setting the deformation matrix to identity is the safest computation [30], leaving us with the translation vector being exactly the displacement vector.

While tracking features, it is possible, although unlikely that large motion between frames does occur. It has been noticed that in such cases the tracking mechanism begins to fail because the disparity between adjacent frames is too large. The result, features are lost and cannot be tracked any further. This fact indicates that some large shift in the adjacent frames has occurred and can be handled at the cost of substantially higher processing time.

In the preprocessing system described in section 2, our goal is to monitor the parallax and overlap between frames in order to ensure the stability and well conditioning of the SfM algorithms. Monitoring the motion through lost features and feature parallax via feature tracking allows us to decide when there is suitable parallax and overlap between frames for the SfM algorithms. The exact criterion for extracting two frames to be fed into the SfM algorithms is described in the next section.

6 Salient Frame Extraction

Once the input video sequence has been segmented into its individual shots, each shot can then independently processed to extract salient frames and then further processed using the SfM phase of the reconstruction system. Since the salient frame extraction and SfM parts of the system are independent, this processing is distributable and easily made parallel.

Briefly, the extraction is done by selecting a set of features smaller than the set used by SfM algorithms and tracking them across adjacent frames. A salient frame is signaled when enough features have surpassed a certain user specified parallax and/or enough features have disappeared and can no longer be tracked. This criterion is exactly what is required to ensure the success of the SfM algorithms.

In algorithmic form, salient frame extraction

```
1. Select good features for frame 1
   place in feature list FL
2. For each frame x in the video
   a. Track features from FL in
      current frame x
   b. Count number of lost features
   c. Count number of features that
      have passed the parallax
      threshold
   d. If detecting boundaries
        i. If (features lost > boundary
             threshold (95%))
           1. Signal boundary and
              extract boundary frame
           2. Refresh the feature
              list FL using boundary
              frame
       ii. Endif (i)
   e. Endif (d)
   f. If (features lost + features
      over parallax threshold) >
      sensitivity threshold (75%)
        i. Signal & extract frame
       ii. Refresh feature list FL
           using current frame x
   g. Endif (f)
3. Endfor (2)
```

The number of features to track the parallax threshold will vary depending on video dimensions, however a good rule of thumb is the following: 2.5 features for every 1000 pixels, the parallax threshold should be 1/8 of the smallest dimension, and a sensitivity threshold of 75 percent. For example, a video that is 320x240 would have 192 features to track, a parallax threshold of 30 pixels (240/8), and a sensitivity threshold of 75 percent. This will supply images with significant overlap and sufficient parallax. When detecting images, a boundary threshold of 95% or greater is sufficient.

7 Results

A series of small video clips was created to test the accuracy and capabilities of the algorithm. These master

clips consist of three smaller subsequences that are cut together. These sequences were created using a standard analog video camera commonly found in many stores and digitized using a video capture card and converted to an MPEG sequence.

The cut detection capabilities proved flawless and correctly identified all sequence start and end points. Surprisingly, this was more accurate than a pixel level cut detection mechanism used to initially verify the sequence start and end points. The pixel level cut detection missed one of the sequence starting points.

Master Sequence	Sub-Sequence	Frame count	Salient frames	Reduction rounded to nearest %
1	Medical Centre	350	13	96
1	Warehouse	336	13	96
1	Body Shop	153	6	96
2	KFC	207	6	97
2	Caisse Populaire	252	10	96
2	Doctors Office	319	5	98
3	House 1	333	16	95
3	House 2	542	15	97
3	House 3	369	12	97
4	Play structure	653	25	96
4	Little House	662	20	97
4	Slide	375	12	97
5	Barn **	374	26	93
5	Temple 1	481	7	99
5	Temple 2 **	429	36	92

Table 1: Frame reduction for image sequences

** denotes that the sequences were taken from a moving vehicle.

One final note about sequences 4 and 5: during the digitization process, syncing errors between subsequences occurred and created some very small sequences of frames that caused cut detection. This was expected and is technically correct however these small "digitization errors" were omitted from the table above.

Figure 2: Every 2nd frame from example sequence (Master Sequence 3, House 1)

As one can see from figures 2 and 3, the spacing of the extracted frames is very consistent and regular. These baselines also make the SfM algorithms well conditioned and hence the images taken from the sequences are well suited.

Figure 3: All frames from example sequence (Master Sequence 2, KFC)

8 Conclusions

A method to preprocess video data that computes both shot boundaries and automatic frame selection for the structure from motion problem has been proposed. The results show that use of this preprocessing mechanism reduces the size of the input data dramatically and helps to keep the structure from motion algorithms very well constrained. This preprocessing step helps to keep the reconstruction problem manageable without altering the SfM algorithms to consider frame rates and degeneracy cases due to insufficient parallax or insufficient motion. The results further show that any application that uses video data will most likely require some amount of salient frame extraction due to the large redundancies of video frame data. In summary, any problem that deals with video will require a preprocessing step to reduce the volume of data in order to keep the problem tractable. Our automated frame extraction method is an important step that cannot be omitted for problems dealing with scene structure and projective vision methods that use video sequences as input.

Binary versions of the salient frame extractor will be made available in the next release of the Projective Vision Toolkit [16]. The toolkit documentation and downloads can be found at

http://www.scs.carleton.ca/~awhitehe/PVT/

9 Acknowledgements

The authors are grateful to Stan Birchfield of Stanford University for making his implementation of the KLT feature tracker publicly available. http://vision.stanford.edu/~birch/klt/. .

REFERENCES

[1] A. Shashua, Algebraic functions for recognition *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8), 1995, 779-789

[2] R. I. Hartley, Euclidean reconstruction from uncalibrated views, *Applications of Invariance in Computer Vision, LNCS*(825)237-256, Springer-Verlag, 1994

[3] R. Koch, M. Pollefeys, and L. VanGool, Multi viewpoint stereo from uncalibrated video sequences, *Computer Vision-ECCV 98*, 1998, 55-71.

[4] M. Pollefeys, R. Koch, M. Vergauwen, and L. VanGool, Automatic generation of 3d models from Photographs, in Proceedings *Virtual Systems and MultiMedia*, 1998.

[5] A. Fitzgibbon and A. Zisserman, Automatic camera recovery for closed or open image sequences, *ECCV'98*, 311-326, Springer Verlag, 1998.

[6] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry, *Artificial Intelligence Journal,* vol. 78, 87-119, October 1995.

[7] P. Torr, and D. Murray, Outlier detection and motion segmentation, in *Sensor Fusion VI*, vol. 2059, 432-443, 1993.

[8] R. C. Bolles and M. A. Fischler, A ransac-based approach to model fitting and its application to finding cylinders in range data, in *Seventh International Joint Conference on Artificial Intelligence*, . 637-643, 1981.

[9] G. Xu and Z. Zhang, Epipolar geometry in stereo, motion and object recognition. *Kluwer Academic*, 1996.

[10] H. Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections *Nature*, vol. 293, 133-135, 1981.

[11] R. Hartley, In defence of the 8 point algorithm, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, 1997.

[12] M. Spetsakis and J. Aloimonos, Structure from motion using line correspondences*, International Journal of Computer Vision*, vol. 4, pp. 171-183, 1990.

[13] R. Hartley, A linear method for reconstruction from lines and points, in *Proceedings of the International Conference on Computer Vision,* 882-87,June 1995.

[14] P. Torr and A. Zisserman, Robust parameterization and computation of the trifocal tensor*, Image and Vision Computing*, vol. 15, 591-605, 1997.

[15] G. Roth and A. Whitehead, Using Projective Vision to Find Camera Positions in an Image Sequence, *VI 2000*.

[16] A. Whitehead and G. Roth, The Projective Vision Toolkit*", in proceedings Modelling and Simulation* (M.H. Hamza Ed), pp204-209

[17] A. Seyler, Probability distribution of television frame difference, *Proc. Institute of Radio Electronic Engineers of Australia* **26**(11), pp 355-366, 1965

[18] R. Kasturi and R. Jain, Dynamic vision, *Computer Vision: Principles* (R. Kasturi and R. Jain Eds), 469-480, IEEE Computer Society Press, Washington, DC, 1991

[19] A. Nagasaka and Y. Tanaka, Automatic video indexing and full-video search for object appearances, in *Visual Database Systems II* (E. Knuth and L.M. Wegner, Eds.), pp. 113-127, Elsevier, 1992

[20] H. Zhang, A. Kankanhalli, S. Smoliar, Automatic partitioning of full-motion video, *ACM/Springer Multimedia Syst.* **1**(1), pp 10-28,1993

[21] F. Arman, A. Hsu, and M. Chiu, Image processing on compressed data for large video databases, in *Proceedings 1st AACM International Conference on Multimedia, Anaheim CA, Aug 1993* pp 267-272

[22] H. Zhang, A. Kankanhalli, and S. Smoliar, Video parsing using compressed data, in *Proceedings IS&T/SPIE, Image and Video Processing II, Feb 1994,* pp 142-149

[23] B. Shahraray, Scene change detection and content-based sampling of video sequences, *Proceedings IS&T/SPIE* **2419,** Feb 1995

[24] J. Lee and B. Dickinson, Multiresolution video indexing for subband coded video databases, in *Proceedings IS&T/SPIE, Conference on Storage and Retrieval for Image and Video Databases, San Jose, CA,* Feb 1994.

[25] Bruce D. Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision**.** *International Joint Conference on Artificial Intelligence*, pages 674-679, 1981.

[26] Carlo Tomasi and Takeo Kanade. Detection and Tracking of Point Features**.** *Carnegie Mellon University Technical Report CMU-CS-91-132*, 1991.

[27] Jianbo Shi and Carlo Tomasi. Good Features to Track**.** *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593-600, 1994.

[28] Stan Birchfield. Derivation of Kanade-Lucas-Tomasi Tracking Equation**.** *Unpublished*, May 1996.

[29] C. Harris and M. Stephens. A combined corner and edge detector. *Proceedings of the 4th Alvey Vision Conference,* 147--151, 1988.

[30] http://uranus.ee.auth.gr/lessons/1/5.html