American Association for Science and Technology
AASCIT Communications
Volume 1, Issue 2
August 20, 2014 online

Offline Work Compression of Planar Signal in 1-D Stable Features

Tohid Aribi

Keywords: Image, Signal, Features, Random Space

In data frames, so that the edge of compression could be extracted as stable features. The front of the data should be a nearly planar signal. Although we assume the user would show a near front signal. We do allow certain range of pose variation of scaling in our system. Offline Work During the offline work of initialization, the user is asked to create a template of the data. This could be done through extracting a stable edge image of the data within a number of data frames showing a frontal view of the data.

Introduction

In offline work, the user needs to manually fit a generic wireframe signal model onto one signal image classifying the data. The system will perform the matching of data and also remember the wireframe's relative position and scale with respect to the matched data template. This will be used to perform the signal model fitting later [1]. All manual work needs to be done just once. In the application stage, the system will do the initialization automatically. After receiving the first data frame, the edge detection block uses a Canny detector to extract the edge image. Then, a template matching block searches for an data pattern using Dynamic Algorithm. The model fitting block fits the signal model accordingly [2-3]. We allowed deformation during the trellis searching. We thus allow approximately 20% of deformation for a single template, which means that the system in theory could find a good matching as long as the input edge pattern is within the range of 20% deformation of the template. Since the data template is represented as a 1D chain code. The problem becomes a 1D sequence-matching problem. We could add more constraints within the trellis searching process to ensure the 2D structure [4]. checking the distance between starting site and ending site. More implementation. After template matching, the model fitting block tries to recover the pose of the data. A planar object, which is viewed under scaled orthography can be modeled by an affine transform. The projection of the template could be described by a six-parameter linear transform of coordinates. It "making signal" template, the input edge image and the result of fitting. The major difference compared to example one, is that the user is not asked to prepare data with clear edge, but has to remembers what signal he or she should make for initialization. we explore the possibility of do the initialization based on feature points. We make use of the stable personal facial feature points that are easier to be detected. A pseudo-automatic system has been set up. The user is involved in the first time initialization when making template. The feature points are organized as a Minimal Spanning Tree graph. Later on, the initialization could be done by first match the particular user's feature points by using Dynamic Programming. Then fit the signal model onto signal region in the first data frame based on the matching.

Methods and Materials

We can define the parameter vector as: The new points coordinate and it's correspondent template point coordinates could be modeled as where we could write equation into a linear function, A least square solution to this problem is given by The solution gives the estimated affine transform parameters. The template of data is also drawn in the low part of each matched image. The matched results and template are enlarged just for illustration purpose. It could be clearly seen that 20% of deformation in template encode quite large rotation and scale variation. The signal wireframe model is fitted onto the image using previous knowledge of its relative position with respect to the data. We suppose that within 2D plane motion they share the same affine transform. In fact, there is no awareness of any "signal" concept in this implementation, so it avoids the complex signal detection task completely. Yet it is quite stable, since both the shape of the data and its relative position to the signal are quite stable. The implementation has low complexity. Promising results have been shown in our experiments. On an average, our system needs 18 seconds to find the result at a resolution of pixels frame. The data template consists of 411 pixels in our case. Different light conditions and complex background doesn't have much effect on the result. Since the fitting of the model is dependent only on data matching, the model could be fitted well even if there is occlusion of signals. Figure 1 shows Compression rate in per data frames. Although the data are required for initialization, they are not necessarily a must for the successive tracking process. If the user needs data anyway, it could further helps the successive tracking job in same, Making a Signal for Initialization Data are non-intrusive tools that could be used to facilitate the initialization, since they are rigid object and have stable shape. We also try to use the user's own facial features as the tool for initialization. What we need are any stable and easy detectable features. In our approach, the keys to a successful initialization are clear and stable features and a global search algorithm. But, there are people whose signals are quite "clean" and don't have the mentioned features. To be able to use personal edge features for the initialization purpose, we suggest producing such features by making signal. The implementation is the same in the process. Before the system is running, the user needs to produce a signal template for later matching and needs to manually fit the model onto the signal once. In order to set up an automatic system for the initialization task, one has to choose "common" feature points that would appear in most cases. The performance of the initialization process is largely dependent on the selection of facial feature points. As stated above, facial feature points have to be localized automatically during the application stage. These points should be easy to detect and good for localization. In addition, the selected feature points should be "stable" overtime; they have to appear in both the template image and the first frame of application data. These issues are discussed here from two aspects, the numerical aspect and the semantic aspect. Numerical Aspect From a pure numerical point of view, we need a criterion to check if a feature point is good for localization. Recall that there is a correspondence problem in tracking and there is a lot of discussion on the selection of feature points. We have decided to "borrow" the criterion used for tracking. Let us look at how a criterion for "good" feature points is developed in the tracking problem. Tracking of features stands for finding the similar intensity in two images. It denote a feature point in the first image, denote the same point in the second image. Figure 2 shows data scattering in 2D space. Here it denotes the position vector , p denotes the displacement vector where x, y denote the image coordinates. The residue image intensity is: The summed square error could then be expressed as: The optimal displacement vector could be derived the solution could be written in the form: the intensity gradients direction. In order to solve the equation reliably, the matrix has to be invertible and well-conditioned, or in other word, the minimal value of must be large enough. The value, which corresponds to a feature point is used to examine if it should be used for tracking. We adapt the same criteria in our case. Semantic Aspects using a threshold of the minimal value of matrix. we could find a lot of feature points that are suitable for tracking.

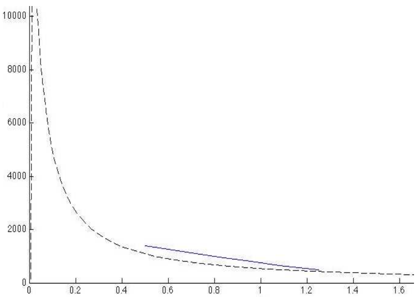

Figure 1. *Compression rate in per data frames*

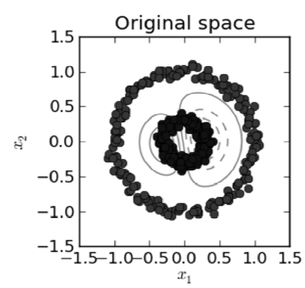

Figure 2. *data scattering in 2D space*

Conclusion

The system will then use a contour finding algorithm to trace the edges of the compression and save the result in a chain code format as the template. The template of data could be prepared in offline work. The matching of the compression to the data frame is performed on the edge image. It can be seen that we work with a rather complex background. To fulfill the model fitting, the system needs to know how to map the model of the data to the model of the signal.■

Tohid Aribi
Department of Electrical Engineering, Miandoab Branch, Islamic Azad University, Miandoab, Iran
tohidaribi@gmail.com

References

[1] W.J. Welsh, "Model-based coding of moving images at very low bit rates," Proc. Int. Picture Coding Symp., Stockholm, Sweden, 2007.

[2] Yao Wang, Jörn Ostermann and Ya-Qin Zhang, Video Processing and Communications, Prentice Hall, 2012.

[3] Sedghi T., "A Fast and Effective Model for cyclic Analysis and its application in classification" Arabian Journal for Science and Engineering Vol. 38 October 2012.

[4] M. Fakheri, T. Sedghi, M. Gh. Shayesteh, M. C. Amirani, "A Framework for Image Retrieval Using Machine Learning and Statistical Similarity Matching Techniques" Image Processing, IET, Vol. 7, Issue:1, 1–11, 2013.