Web Based Video Change Detection

Keerthi Reddy B[1], Madhushree C[2], Namitha N[3], Ramya M[4]
Department of CSE
[1, 2, 3, 4] Bangalore Technological Institute

Abstract-*We present an efficient open-source implementation of a novel video alignment algorithm, based on low dimensionality frame matching and the recently introduced ECC image registration algorithm. We write the algorithm in the Java language, which facilitates prototyping, using the OpenCV library, and then in C++. We test our application on realistic video benchmarks and obtain a good alignment quality. Compared to an initial sequential implementation, we speedup the application by: i) algorithmic optimization, ii) improving data locality, iii) executing parts of the application on parallel hardware. We achieve a sequential performance of at most 10.47x real-time performance when processing a query video of resolution 240x320, with 30 fps. When offloading parts of the computation on NVIDIA GPUs, we obtain a factor of at most 12.39x speedup when compared to the sequential version run on CPU, when using High-Definition resolution. These results pave the way for further optimization in order to run the application on energy-efficient embedded CPU and GPU processors*

Keywords-OpenCv, Optimization, High definition Resolution

I. INTRODUCTION

Deep web resides in the inner part of web where ordinary search engine cannot index. These data contain vast amount of valuable information. It's a challenging to locate deep websites because it not indexed by search engine, are usually sparsely distributed and its content keeps on changing dynamically. To address this problem a specific crawler having the capability of form focused and adaptive crawler so as to cover deep web directory have to be developed. Link classifiers in this crawler play a pivotal roles achieving higher crawler efficiency than best first crawler. However these link classifiers are used to predict the distance to the page containing searchable form which is difficult to estimate for delayed benefit link. Crawler must produce large quality of high quality results from the most relevant content sources.

II.EXISTING SYSTEM

A. Lucas-Kanade's method

In this work, the authors address the problem of aligning two video sequences. Such alignment refers to synchronization, i.e., the establishment of temporal correspondence between frames of the first and second video, followed by spatial registration of all the temporally corresponding frames. Video synchronization and alignment have been attempted before, but most often in the relatively simple cases of fixed or rigidly attached cameras and simultaneous acquisition. In addition, restrictive assumptions have been applied, including linear time correspondence or the knowledge of the complete trajectories of corresponding scene points; to some extent, these assumptions limit the practical applicability of any solutions developed. We intend to solve the more general problem of aligning video sequences recorded by independently moving cameras that follow similar trajectories, based only on the fusion of image intensity and GPS information.

The novelty of our approach is to pose the synchronization as a MAP inference problem on a Bayesian network including the observations from these two sensor types, which have been proved complementary. Alignment results are presented in the context of videos recorded from vehicles driving along the same track at different times, for different road types. In addition, we explore two applications of the proposed video alignment method, both based on change detection between aligned videos. One is the detection of vehicles, which could be of use in ADAS. The other is online difference spotting videos of surveillance rounds

B. Drawbacks Of Existing System

The Accuracy of this approach is still questionable. The system is not accurate to produce the best matching frames.

II. PROPOSED SYSTEM

In this paper we bring the following meaningful contributions:

We provide an efficient open-source implementation of the video alignment algorithm, which is able to run on parallel hardware such as General Purpose Graphical Processing Units (GPGPUs, or GPUs for short) and on Chip Multiprocessors (CMPs), using OpenCV, an industry supported open-source library.

We tested our implementation on demanding datasets: a group of videos recorded from a drone flying over a

terrain with many visible landmarks and another set of videos done while driving on a highway. Flying a drone poses a serious challenge to our algorithm because we can experiment rather big differences in trajectories, in the order of meters, when flying over the same area. These variations can be the result of wind direction changes or pilot errors, which could be avoided on drones equipped with GPS. We are able to cope with these rather big differences in field of view (FOV) in the input videos, in certain cases having to trade performance to obtain better alignment quality.

We provide an optimal algorithm to compute the temporal alignment using a standard critical path algorithm applied on the graph defined over the vote space introduced. This algorithm increases the quality of the temporal alignment on the experiments we perform.

The Advantages of ECC algorithm, being an area-based (or direct) image registration method, provides more accurate alignment results than Lucas-Kanade's method

III. SYSTEM DESIGN

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering.

System design is one of the most important phases of software development process. The purpose of the design is to plan the solution of a problem specified by the requirement documentation. In other words the first step in the solution to the problem is the design of the paper.

The design will contain the specification of all these modules, their interaction with other modules and the desired output from each module. The output of the design process is a description of the software architecture.

The design phase is followed by two sub phases

- High Level Design
- Detailed Level Design.

A. High Level Design

In the high level design, the proposed functional and non functional requirements of the software are depicted. Overall solution to the architecture is developed which can handle those needs. It involves the following consideration.

- Design consideration
- Dataflow diagram

There are several design consideration issues that need to be addressed or resolved before getting down designing a complete solution for the system.

1) Assumptions and dependencies: The main assumptions and dependencies identified are

a) JDK has to be installed in the machine where all the three subcomponent will be executing.

b) The application servers like either the JBOSS or the Apache Tomcat will have to be supported by the host machines.

c) There shall not be any firewall or other engines that prevents the remote requests from the portal.

d) There shouldn't be any permission related issues on any cluster. The host operating system should take of permitting all the requests to the cluster from the interface layer.

B. Mode of operation of a System

The system is capable of operating in two different modes.

Video Alignment Algorithm: The video alignment algorithm takes two videos, named reference and query, recorded at different times on similar trajectories, possibly at different speeds, and finds the best matching pairs of frames for the two videos. Once we perform these matches, we can easily detect changes between the two different recordings, which can point to important, even critical, events such as missing objects.

Motion Detection: This component detects motion if the objects. We use a video source to capture the motion of the objects. The video sources can be a web camera or any video stored in the computer. This component detects motion in the source file provided to it. The path of the source file is provided as input to the component. This path is provided by the user after logging into the system. The user can log into the system only if he is a registered user. Hence the user has to register before logging in. This component is put in a jar file which is run in the user interface.

C. User operations

Account Operations: User must be creating a new account in our portal to get access to the rest of the modules. The user can also perform various other account operations like retrieving the forgotten password, login, logout, delete profile, change password, and edit profile

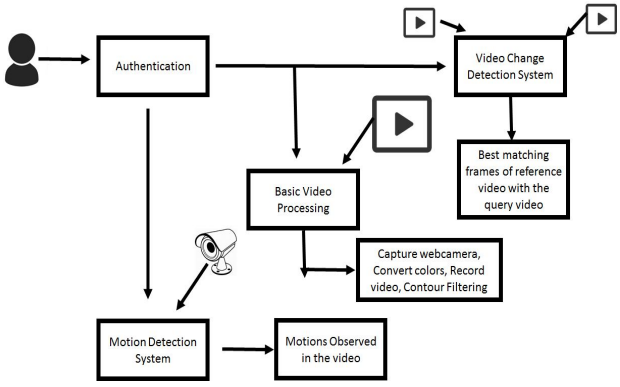
Video Alignment: Here the user will have to provide two videos (reference and query video) along with the

accuracy factor as inputs and will be getting the best matching frames of reference video with the query video as an output.

Motion Detection: Here the user will have to provide the video source as an input and will be getting the changes seen within that video as an output Contour filtering: Here the user will be able to filter the objects within the image based on their color properties.

2) Dataflow diagram: A data flow diagram is the graphical representation of the flow of data through an information system. DFD is very useful in understanding a system and can be efficiently used during analysis. With a data flow diagram, users are able to visualize how the system will operate that the system will accomplish and how the system will be implemented, old system data flow diagrams can be drawn up and compared with a new systems data flow diagram to draw comparisons to implement a more efficient systems.

Data flow diagrams can be used to provide the end user with a physical idea of where the data they input, ultimately as an effect upon the structure of the whole system.



Overall Data flow diagramt.

During the detailed phase, the view of the application developed during the high level design is broken down into modules and programs. Logic design is done for every program and then documented as program specifications. For every program, a unit test plan is created.The entry criteria for this will be the HLD document. And the exit criteria will the program specification and unit test plan (LLD).

IV. CONCLUSION

We provided an efficient open-source implementation of the video alignment algorithm, which can run on parallel hardware such as General Purpose Graphical Processing Units (GPGPUs, or GPUs for short) and on Chip Multi Processors (CMPs), using OpenCV, an industry supported open-source library. As future work, we consider continuing optimizing the application to run it on energy-efficient embedded CPU and GPU processors, such as ARM Cortex and NVIDIA Tegra K1.

REFERENCES

[1] Georgios Evangelidis and Christian Bauckhage. Efficient Subframe Video Alignment Using Short Descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(10):2371–2386, October 2013.

[2] Marius Muja and David G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 36, 2014.

[3] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An Efficient Alternative to SIFT or SURF. In Proceedings of the 2011 International Conference on Computer Vision, ICCV '11, pages 2564–2571,Washington, DC, USA, 2011. IEEE Computer Society.

[4] J. Howse. OpenCV Computer Vision with Python. Packt Publishing, 2013.

[5] J. Solem. Programming Computer Vision with Python: Tools and Algorithms for Analyzing Images. O'Reilly and Associate Series. O'Reilly Media, Incorporated, 2012.

[6] Erik Smistad, Thomas L. Falch, Mohammadmehdi Bozorgi, Anne C. Elster, and Frank Lindseth. Medical image segmentation on GPUs - A comprehensive review. Medical Image Analysis, 20(1):1 – 18, 2015