Abstract

In this thesis we present a custom implementation of the Tree Structure Model Face Detection, Pose Estimation and Landmark localization presented by Zhian Zhu and Deva Ramanan in 2013 [x]. Our implementation has be carefully optimized making the TSM system procedure faster both in a single core processor and in large multiprocessors systems. It is also memory optimized making the TSM system able to run in low memory resources hardware like the embedded systems. We also propose methods that can make the TSM system even faster with the risk of a tiny reduction of the system face detection performance. We compared our implementation with the one released in September 2015 by Hang Su [x] and we discover that ours is at least two time faster and about ten times less memory consuming.

Introduction

The processing time for a face detection or an object one is definitely depended by the input image size. As most of the algorithms for object detection are trained to detect objects in a specific size the image pyramid method is the one used for detecting an object in various scales. For that reason, large images allows the detection of small objects within the image while small ones only allow the detection of large ones. In the TSM system [x] the processing time needed for the top level of the image pyramid is about the 25% of the time needed for all the pyramid levels. Despite the image size the most time consuming procedure in the TSM system is the convolution one used for the landmarks detection. The landmarks detection is a critical stage for every face detection system and this is why most of them use only a small set of landmarks, usually less than ten. The TSM system uses at least 99. Unfortunately no landmark localization task can be applied without the convolution procedure using trained filters.

In our implementation we gain speedup by optimizing the code and using multithreading techniques. We compare our implementation with the one Hang Su created in different hardware architectures to see which one is faster. In addition we propose a method for speeding up the detection process when there are no detectable objects in the image. This method is very affective in empty of faces images or images with large size.

As far as the memory consumption of the TSM system it reaches its maximum one during the detection process of the top level of the image pyramid. Code optimization was not enough for reducing the temporary memory needed so we actually changed the TSM system architecture in a custom one in contrast to the original one in order to make our system less memory consuming. We compare again our implementation with the Hang Su one. The difference is very significant.

Related

As far as we knew, no previous work was introduced jointly addressing the tasks of face detection, landmark localization and pose estimation until the June of 2012 when X. Zhu and D. Ramanan proposed the “Face detection, pose estimation and landmark localization in the wild” [1] work. This work was supposed to be the state-of-the-art that time and was used as a baseline for further research leading to the presentation of more proposals for systems trying to make the face detection process a much faster and efficient. To succeed this, new models was used except of discriminant parts models like neural networks. The neural networks are considered to be the more efficient and fast models that can detect faces and estimate pose. We are not going to mention all of them but only the most recent like [3], [4], [5], [6] and [7]. The most similar work to [1] is the [8], [9] and [10].

Our work does not try to present a new face detection or object detection method but to make the Discriminant Part Models and Tree Structural Model systems faster and less memory consumption ones. For this reason the only related work that can be referred is the [25] that implements the same algorithm. The reason of choosing this algorithm is because except of face detection and pose estimation it also offer landmark localization of the 68 or 39 (depends on the viewing angle) human face landmarks. Another task it also implements is the face detection of faces in the range of over 60 degrees viewing angle. Many algorithms have been deployed since then, like [26], [27], [28], [29], [30], [31] and [32] but most of them do not offer all these tasks the same way. Many of them do not offer landmark localization at all or they detect few of them, the most significant for the face detection (ex. Eyes). The need of the landmarks localization demand the convolution procedure of at least 68 cascade windows of the image features space that is a very heavy procedure. Others does not offer pose estimation at all while the most of them that does, only offer pose estimation in the range of 60 degrees. Only the [25] does offer the complete set of tasks and it’s the one to compare with.

As far as we know, there are also many other freeware algorithms offered in the web but none of the uses the TSM method meaning that all of the have a lack of tasks. They usually offer face detection or/and pose estimation but not the 68 landmark localization or face detection in more centered faces as referred in the previous paragraph. Some of these algorithms are [3], [4], [5], [6], [7], [8] and [10].