Evaluation of Optimizations for Object Tracking – Feedback-Based Head-Tracking

Anjo Vahldiek, Ansgar Schneider, Stefan Schubert
Baden-Wuerttemberg State University Stuttgart
Computer Science Department
Rotebuehlplatz 41
70178 Stuttgart, Germany

Abstract. Today's head tracking algorithms use the haar classifier from P. Viola and M. Jones. The method is trained by a specific object example database. In case of a complex object as a head it is difficult to find a good general detection. The used head tracker was trained by a typical example database and detects heads with a gradient of about 15°. Images containing a high gradient also have misplaced regions of interest (nose, mouth, and forehead). The aim of this paper is to discuss solutions to increase the gradient as well as reducing the inaccuracies of misplaced regions of interest.
The head tracker is used in a real time emotion recognition system for videos which takes information from the regions of interest. Placing the regions of interest mostly relies on the head detection. The actual environment is not robust and often leads to wrong interpretations of the expressed emotions.
As main idea the image will be rotated by the gradient of the head calculated using the previous image in front of the head tracker process. This brings the head always into an upper-right position which the head tracker was trained before. Our solution feedback-based head-tracking takes information from last detected object of the image sequence to optimize the actual image. In this case the gradient of the head is tracked also by accumulating the eye angle of the sequence.
In conclusion the proposed algorithm tracks twice as many images of an example video with better placed regions of interest. It can be used as a module in all object tracking systems to increase the robustness. In the future features like lazy return or dynamicity increase the accuracy of the algorithm.

Keywords: Head-Tracking, Emotion Recognition, Image Processing, Optimization, OpenCV, Haar-Classifier

1 Introduction

Actual Head-Tracker (HT) are not robust, they often miss heads or their locations. Optimizing the input images before the HT detects the head, gives the ability to have a better tracking algorithm. Our approach tracks objects in sequences by using the general Haar Classifier of OpenCV. The optimization of each input image is done by reporting a feedback from previously detected objects and their characteristics like gradient or head position. Each of these values is used to process image operations in order to simplify the input image for the HT. Simplification operations are rotation, zoom and others.

Fig. 1 depicts the current implementation of an real time emotion recognition environment for web-cameras developed at the Baden-Wuerttemberg State University. Each image of a sequence is processed itself. It starts by detecting the face. According to this location regions of interest like the eyes, nose, mouth or forehead will be found. Each region specifies a feature of a face which is able to show several emotions. During the last phase the features are used to classify the emotion. If a failure occurs in one of the modules, the following modules will also fail. This dependency leads us to our solution, because we improve the first step by making it more robust.

The reminder of this paper is organized as follows. At first we present our solution feedback-based HT. Afterwards we introduce evaluation mechanisms which show the advantages of the new methodology. Future Work and a Conclusion will be presented at the end.

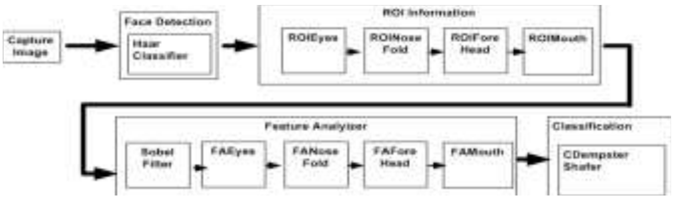

Fig. 1: Overview of the emotion recognition process

2 Feedback-Based Head-Tracking

Feedback-Based Head-Tracking inserts two modules in the architecture of the emotion recognition process one before the Face Detection and one after the Region of Interest (ROI) Information. Fig. 2 depicts the overview of the additional modules.

The module after the ROI Information starts at 3a. It sends a feedback in order to remember a certain set of information about the objects within the image. In our case we track the gradient of the eyes which enables us to calculate the angle of the head. This information is used by the other module to optimize the next image of the sequence. Since the sequence of images is a continuous motion of a head, we predict the head to be in the next image in some kind of similar position. As always such predictions can mislead. This is avoided by using advanced techniques (lazy return & dynamicity) as described in the section future work.

The optimization module takes the recorded information to rotate the next input image by the angle of the head around the center of the head of the previous image.

As depicted by Fig. 2 the algorithm starts with an unknown head angle $Kw_{n-1} = 0$ of image B_{n-1}. When a head was detected it tires to define regions of interest which are used to calculate the current head angle Kw_n. The most significant impact is made by calculating the gradient of the line connected by the eyes. In this case the gradient will be added to the current head angle $Kw_n = Kw_{n-1} + Aw_{n-1}$. Now the head angle

$Kw_n \neq 0$ and the next algorithm starts at 4. In these cases the optimization module will be processed before the Face Detection starts.

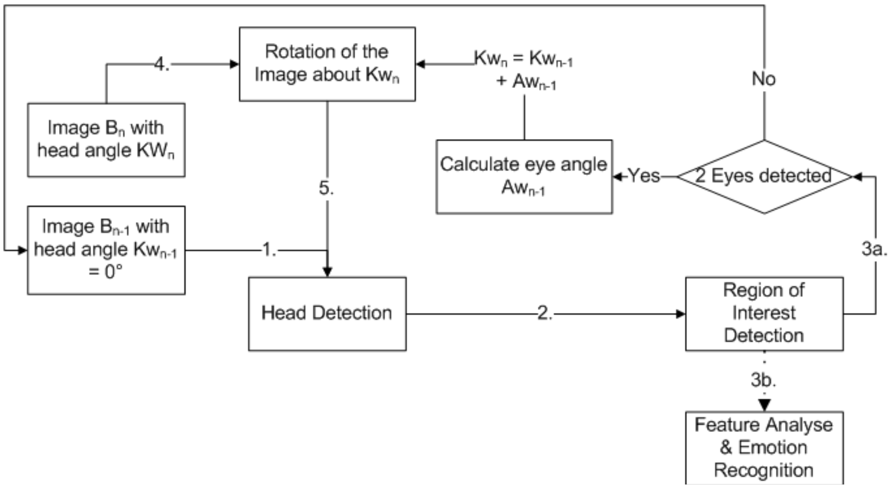

Fig. 2: Overview of Feedback-Based Head-Tracking

3 Evaluation

The evaluation was split into two phases. During the first phase an example video[1] was taken to ensure the advantages of the algorithm. As depicted by Fig. 3 and 4 various advantages occur. The most obvious is the accuracy of the regions of interest. As described before the Feature Analysis algorithms tend to be problematic when the regions of interest do not hit the real region. Fig. 3 compares the old algorithm which hits e.g. the region of the mouth only by half whereas the new implementation finds the whole mouth.

These advantages are also shown by several other indicators of the video which are compared in Tab. 1. We double both the number of detected heads and the range of the head angle. In addition the accuracy is shown by the tremendous increase of small eye angle ranges. Fig. 4 shows that head angles below 5° have a good accuracy on positioning the regions of interest. In conclusion this example video has shown that the new algorithm increases the accuracy and the quantity of the head tracking algorithm.

During the second phase we tried to prove the accuracy of the head angle detection, because the algorithm mostly relies on this angle. Therefore we build an apparatus shown in Fig. 5 which is able to show the current head angle on the back of the image which is rotated in front. As a result we resolve a standard deviation of $< 1°$ and different apparatus related failures. The apparatus related failures evolve, because the image was not mounted totally horizontal on the apparatus, the person holds the

1 All videos are published on YouTube: Playlist HeadDetection (http://www.youtube.com/view_play_list?p=41AD297EAEA53968)

head about a small angle left or right or the person shown on the image has a not symmetric face. We tried to decrease the failure by using prepared symmetric images of heads and simplified the apparatus, but still have an apparatus failure of up to 2°.

Fig. 3: Comparision of Regions of Interest at 21° head angle (right figure with Feedback-Based Head-Tracking)

Fig. 4: Beginning inaccuracies at 5° elegance (right figure with Feedback-Based Head-Tracking)

	New Algorithms	Old Algorithms
Max Rotation	55.25	24.15
Min Rotation	-62.70	-22.17
Angle Range	117.95	46.31
Detected Frames	890.00	392.00
Avg Actual Eye Angle	2.40	7.54
Angles between -0,5 – 0,5	259.00	23.00
Small Angles of All detected Frames	29.10%	5.87%
Angles between +-5	868.00	175.00
Normal Angles of All detected Frames	97.53%	44.64%
Max Actual Eye Angle	13.24	24.15
Min Actual Eye Angle	-14.42	-22.17
Actual Eye Angle Range	27.66	46.31
Avg Time of Recognition per detected Frame	32.74	29.38
Frames per Second	30.54	34.04

Comparison		
Improvement Angle Range	254.69%	wider range
Improvement Frames Detected	227.04%	more detected
Improvement Small Range Angles	495.98%	more head angles in Rage (+- 0,5)
Improvement Avg Eye Angle	40.27%	lower Avg Eye Angle (lower --> better ROIs)
Avg Additional Time in ms	2.72	additional time
Avg Part in Head Detection Algo	8.44%	of total Recognition Process

Tab. 1: Overview of evaluation statistics

Fig. 5: Apparatus to evaluate the accuracy of the head angle

4 Future Work

The analysis of the example video shows that the speed by which the head is moved can also be predicted by previous images. Therefore we analyzed sub sequences of 100 images with different prediction values. The prediction value is used as indicator for the next image rotation. It consists of the head gradient and a second value which is at the moment defined at compile time. The second value specifies at what range the head angle is supposed to be within the next image. It is called normalized head speed and is a floating point number greater 0.

As a result the most optimal normalized head speed of the example video was 1.125 which increases the current head gradient by 12.5%. Optimizing this value has to be dynamic, because for each sub sequence a different normalized head speeds won. Several local minima show the adaptability to the head movement. This impression is enhanced when the normalized head speed is compared to the current angle graph. In a sub sequence with very fast head movement the most optimal normalized head speed increased to 1.3.

In future implementations this value should adjust to the current speed of the head. Such value could be generated by locking at the last 10 head gradients in order to predict the direction of movement.

Another improvement is made by locking at the time when the HT loses the head. Because the algorithm uses the progress of the head gradient the actual algorithm falls back to a head gradient of 0. In such cases the base HT algorithm is used which has only the ability to detect aligned heads. The first improvement was made by introducing lazy return. Using this mechanism the head gradient is not set to 0 after a loss. It is reduced by a certain percentage and it is possible to catch the head again after a few frames.

Unfortunately lazy return also returns to a head gradient of 0 which is a non-robust solution. At this point symmetry detection helps to find the head gradient. A combination of this algorithm and a symmetry detection algorithm solves the problem of a loss and the problem of starting in an aligned position. Whenever no head was detected in the previous image the algorithm uses a symmetry detector to have a head gradient for the next image. Using the symmetry detector all the time is very expensive, because current implementations take about 50 ms per image and may detect several symmetry aches.

5 Conclusion

The presented solution increases the robustness of current head-tracking algorithms. An evaluation has shown that the precision of this system works by using typical web-cameras. As described the number of positive detected frames doubled. Even more important the range of head leaning increased by 250%. The required time for computation relies on the optimization operations which is only rotation at the moment. For a rotation of a 320x240 image about 3 ms are required. This extends the run time of the emotion recognition environment by 8%. In terms of a real time analysis the advantage is much higher than the costs.

References

1. Viola, P.; Jones, M.: Rapid object detection using a boosted cascade of simple features