Multi-Level Cartooning for Context-Aware Privacy Protection in Visual Sensor Networks

Ádám Erdélyi, Thomas Winkler, and Bernhard Rinner
Institute of Networked and Embedded Systems
Alpen-Adria Universität Klagenfurt and Lakeside Labs
Lakeside Park B02b, 9020 Klagenfurt, Austria
{adam.erdelyi, thomas.winkler, bernhard.rinner}@aau.at

ABSTRACT

Our solution to the MediaEval 2014 Visual Privacy Task [4] is a privacy-preserving video filter that is able to maintain a high intelligibility level in surveillance systems while providing a reasonable privacy protection level to monitored people and a pleasant view to observers. This paper describes our context-aware method that is based on cartooning and pixelation effects. Subjective evaluation results are also presented to demonstrate the performance of our algorithm.

1. INTRODUCTION

Surveillance cameras play various roles in our everyday lives and their increasing number attracted attention to privacy issues. The goal of the MediaEval 2014 Visual Privacy Task [4] is to find a method that protects privacy while the original purpose of surveillance can be maintained. Together with annotations of sensitive regions such as faces, people and carried items the PEViD data-set [7] is provided to evaluate solutions submitted by task participants. Desired privacy levels ([H]igh, [M]edium, or [L]ow) are also included in the annotations for each region so that various filters can be combined and adjusted accordingly.

Traditional CCTV cameras are continually being replaced by more modern smart cameras which are usually part of Visual Sensor Networks (VSNs). Other widespread video-capable devices such as smart phones, tablets or web-cams also pose privacy threats due to their frequent use in public spaces. Processing capabilities of these devices allow the integration of privacy protection methods directly into the camera. Our aim is to create such an integrated filter. In order to simulate the limited computational power of the above mentioned embedded devices, we ran our privacy-preserving algorithm on the Jetson TK1 [1] development board to process the provided videos.

Our method is based on a cartooning effect which is applied both globally and locally. In sensitive regions the filter intensity is adjusted according to the annotation. Faces are further protected with an extra pixelation effect.

2. IMPLEMENTATION

A prototype of our filter is implemented in C++ by using OpenCV [2] for video processing and pugixml [3] to parse the annotation files. Figure 1 depicts the processing pipeline of the proposed algorithm. A detailed description of our privacy protection filter is provided in Sections 2.1 to 2.3. The submitted videos have been generated on the Jetson TK1 platform in the following software environment: Linux for Tegra R19 (Kernel version 3.10.24) and OpenCV 2.4.9 with GPU support via CUDA 6.0.

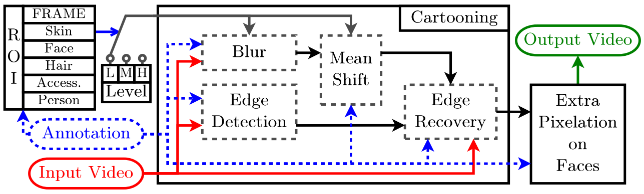

Figure 1: Processing pipeline of our privacy protection filter.

2.1 Global Cartooning

First, a medium-intensity cartooning effect is applied to the whole video frame. This always ensures a default level of privacy thereby preparing the filter for real-world use where privacy loss may occur at sensitive regions due to inaccurate feature extractors. Additionally, implicit privacy channels [8] are also protected. The cartooning effect (represented by the box labelled "Cartooning" in Figure 1) is a result of the following main steps:

1. Preliminary blurring with a $k \times k$ size kernel is applied in order to reduce noise. Edges are detected by the Sobel edge detector for later use.

2. Then the blurred video frame goes through a Mean Shift [5] filter with a spatial window radius of sp and a colour window radius of sr. This makes the frame smoother and replaces fine details with solid colour patches as if it was drawn like a cartoon.

3. Finally, edges are recovered along object contours by performing a bitwise weighted copy from the original input frame. This makes the final output less blurry and more similar to hand-drawn cartoons where object contours are usually emphasized.

The parameters used in the steps above are dependent on desired privacy levels taken from the annotation files. k=17, sp=30, sr=60 are used for high level; k=9, sp=20, sr=40 for medium level; and k=3, sp=10, sr=20 for low level privacy. For global cartooning we used the medium level.

2.2 Local Cartooning

After global cartooning, protection levels of sensitive regions are adjusted locally according to Table 1 in [4]. More

Copyright is held by the author/owner(s).
MediaEval 2014 Workshop, October 16-17, 2014, Barcelona, Spain

sensitive image regions such as faces are further protected with high-intensity cartooning while less sensitive ones are downgraded to a lower privacy level in order to increase intelligibility. The same cartooning effect is used locally that was described in Section 2.1 except the parameters are changed according to the annotations.

2.3 Pixelation

In the final step of our processing pipeline an extra pixelation effect is applied on faces in order to further obscure the identity of people. The region of pixelation is the maximum inscribed ellipse of the face's bounding box and the pixel size is one-fifteenth of its larger dimension.

(a) Original. (b) Filtered. (c) Original. (d) Filtered.

Figure 2: Comparison of original and protected frames.

3. EVALUATION RESULTS

Two pairs of video frames (original and filtered) in Figure 2 demonstrate the visual effect of our privacy filter.

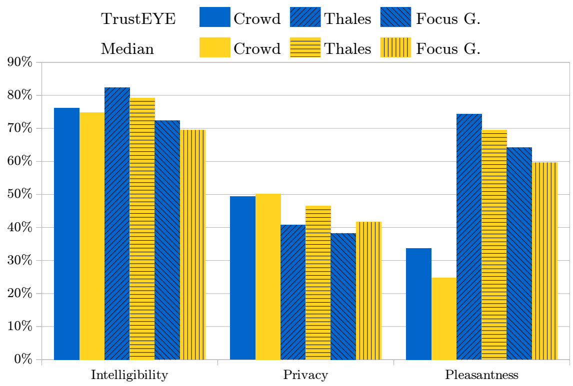

Figure 3: Subjective evaluation results for intelligibility, privacy and pleasantness criteria derived from a survey that has been conducted in three different groups.

In terms of processing speed the Jetson TK1 board is capable of ∼5 fps for 320×180, ∼2 fps for 640×360, ∼1 fps for 800×450, ∼0.8 fps for 1024×576, and ∼0.2 fps for the provided full HD resolution videos. It proves that privacy protection can really be done directly inside the camera. Despite running the Mean Shift filter on the GPU instead of the CPU it remains the bottleneck of our algorithm. Thus, the current version of our prototype cannot filter full HD videos in real time, although acceptable frame-rates can be achieved at lower resolutions. More detailed discussion about achievable frame-rates on embedded devices can be found in [6] where we show a scenario-adaptive version of cartooning filter. An alternative implementation of cartooning is presented in [9] proving that acceptable frame-rates are possible on even more resource-constrained devices.

Figure 3 shows the subjective evaluation results provided by the Visual Privacy Task organizers. These numbers are calculated from the outcome of a 12-question survey that has been conducted in three different groups. The first group consists of 230 regular people and the questionnaire was filled out in frame of a crowd-sourcing campaign. The second group is constituted of 65 participants from Thales, France. And the third is a focus group with 59 participants from all over the world. Questions of the survey are assigned around the following three criteria: intelligibility, privacy, and pleasantness.

After analysing Figure 3, it is clear that the performance of our method is always better than the median performance among the 8 participants in terms of intelligibility and pleasantness. We also achieved competitive results as for privacy, although we slightly underperform the median.

4. CONCLUSION AND FUTURE WORK

By introducing a global component to our privacy protection filter we cover implicit privacy channels and ensure a default level of privacy even if inaccurate real-world feature detectors are being used. Our method provides a pleasant view and high intelligibility while reasonably protecting privacy. It works reasonably well on the Jetson TK1 board for lower resolution videos, although further improvements are necessary to reach acceptable frame-rates for full HD videos.

5. ACKNOWLEDGEMENT

This work was partly funded by the European Regional Development Fund and the Carinthian Economic Promotion Fund (KWF) under grant KWF-3520/23312/35521.

6. REFERENCES

[1] NVIDIA – Jetson TK1 Development Kit. `https://developer.nvidia.com/jetson-tk1` (last visited: Sept. 2014).

[2] OpenCV – Open Source Computer Vision. `http://opencv.org` (last visited: Sept. 2014).

[3] pugixml – Light-weight, simple and fast XML parser for C++ with XPath support. `http://pugixml.org` (last visited: Sept. 2014).

[4] A. Badii, T. Ebrahimi, C. Fedorczak, P. Korshunov, T. Piatrik, V. Eiselein, and A. Al-Obaidi. Overview of the MediaEval 2014 Visual Privacy Task. In *Proceedings of the MediaEval Workshop*, Barcelona, Spain, 2014.

[5] Yizong Cheng. Mean shift, Mode Seeking, and Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

[6] Adám Erdélyi, Tibor Barát, Patrick Valet, Thomas Winkler, and Bernhard Rinner. Adaptive Cartooning for Privacy Protection in Camera Networks. In *Proceedings of the Int. Conf. on Advanced Video and Signal Based Surveillance*, volume 6, page 6, 2014.

[7] P. Korshunov and T. Ebrahimi. PEViD: Privacy Evaluation Video Dataset. In *Proceedings of SPIE Applications of Digital Image Processing XXXVI*, volume 8856, 2013.

[8] M. Saini, P. K. Atrey, S. Mehrotra, and M. S. Kankanhalli. Considering Implicit Channels in Privacy Analysis of Video Data. *IEEE Communications Society E-Letters*, 6(11):27–30, 2011.

[9] Thomas Winkler, Adám Erdélyi, and Bernhard Rinner. TrustEYE.M4: Protecting the Sensor–not the Camera. In *Proceedings of the Int. Conf. on Advanced Video and Signal Based Surveillance*, page 6, 2014.