Tampering detection for low-power smart camera

Adriano Gaibotti[1], Claudio Marchisio[1], Alexandro Sentinelli[1], and Giacomo Boracchi[2]

[1] STMicroelectronics, Advanced System Technology, Via Camillo Olivetti 2, 20864, Agrate Brianza (MB), Italy
{adriano.gaibotti, claudio.marchisio, alexandro.sentinelli}@st.com
[2] Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Via Ponzio 34/5, 20133, Milano (MI), Italy
giacomo.boracchi@polimi.it

Abstract. One of the most critical issues in smart cameras, used as nodes in wireless multimedia sensor networks (WMSN), is the automatic detection of events that could compromise the correct acquisition of the scene, such as the presence of water on the camera lens, or the displacement of the device, or tampering events. In general, techniques used to identify these type of events are called tampering detection algorithms. We introduce a solution for the tampering detection that can be used for low-power and embedded smart cameras, in which the acquisition is done with low framerate, such as one frame every minute.

1 Introduction

[**Giacomo**: Adriano: metti un paio di foto per spiegare il problema che consideriamo.]

Random Toughs:

- Smart cameras, low-power monitoring of scene. Description of the application scenario. Low frame rate.
- Cameras organized in a multimedia network, continuous acquisition and streaming is not feasible
- The problem of false alarms, radio module activation
- Other tampering attacks like obfuscation (??) which might be due to environmental phenomena such as rain, fog and mist over the camera lenses have to be detected by image analysis methods
- Displacement can be perceived by MEMS as well but these device alone are prone to false alarms. Visual inspection is necessary to reduce false alarms
- constrained environment: algorithms have to operate with a low computational complexity and memory requirement

2 Related Works

[**Giacomo**: Adriano: metti tutte le reference, ciascuna con un commento di una frase per dire che fa ed una frase (o mezza) per dire i problemi che ha (con particolare riferimento all'ambito low-power) Poi le aggiustiamo in maniera organica]

3 Problem Formulation

[**Giacomo**: Adriano: metti le formule di quello della tesi circa displacemente e out of focus. Poi condensiamo il tutto]

4 Proposed Solution

4.1 Scene Segmentation

4.2 Indicators

[**Giacomo**: Adriano: mettere formule degli indicatori e anche del frame difference qua]

4.3 Outlier Detection

4.4 Algorithm Summary

[**Giacomo**: Adriano: inserisci qui l'algoritmo e traducilo in inglese. Se riusciamo lo spostiamo prima di tutte le sottosezioni]

5 Experiments

5.1 Dataset Description

[**Giacomo**: Adriano: Prova a mettere qua le info]

5.2 Alternative Approaches

- Full
- Adaptive Region
- Voronoi Regions

5.3 Performance Assessment

[**Giacomo**: Adriano: Dire come vengono calcolate le ROC curves TPR e FPR, le cifre di merito insomma, spiegando bene che parametro varia]

[**Giacomo**: Adriano: metti entrambe le ROC curves, affiancate e per bene ed alcuni esempi di sequenze]

5.4 Discussion

[**Giacomo**: Adiano: Aggiungi qua la complessitá computazionale]

6 Conclusion

[**Giacomo**: Adriano: butta in inglese gli ongoing works (come ultima cosa)] [1].

Algorithm 1: Blur detection algorithm

Configuration:

Extract regions $\{R_k\}, k = 1, \ldots, K$
for $t = 1, \ldots, T_o$ **do**
    Acquire frame z_t
    for $k = 1, \ldots, K$ **do**
        Compute $g^k(t)$, $\frac{\partial g^k}{\partial t}(t)$ for the region R_k
    end
    Compute $g(t)$, $\frac{\partial g}{\partial t}(t)$
end
for $k = 1, \ldots, K$ **do**
    Define thresholds Γ^k_{min} and Γ^k_{max}
end
Define CDT parameters on $g(t)$ variance

Operational phase:

for $t = T_o, \ldots, \infty$ **do**
    Acquire frame z_t
    Compute $g(t)$, $\frac{\partial g}{\partial t}(t)$
    $n = 0$
    for $k = 1, \ldots, K$ **do**
        Compute $g^k(t)$, $\frac{\partial g^k}{\partial t}(t)$ for the region R_k
        if $\frac{\partial g^k}{\partial t}(t) < \Gamma^k_{min} \vee \frac{\partial g^k}{\partial t}(t) > \Gamma^k_{max}$ **then**
            $n = n + 1$
        end
    end
    if $n \geq K - 1$ **then**
        z_t is a defocused frame
    end
    if *CDT detect a change on* $g(t)$ *variance* **then**
        z_t is a defocused frame
    end
end

Algorithm 2: Displacement detection algorithm

Configuration:
1. Extract regions $\{R_k\}, k = 1, \ldots, K$
2. **for** $t = 1, \ldots, T_o$ **do**
3.   Acquire frame z_t
4.   **for** $k = 1, \ldots, K$ **do**
5.   Compute $l^k(t)$, $\frac{\partial l^k}{\partial t}(t)$ for the region R_k
    end
end
6. **for** $k = 1, \ldots, K$ **do**
7.   Define thresholds Γ^k_{min} and Γ^k_{max}
end
Operational phase:
8. **for** $t = T_o, \ldots, \infty$ **do**
9.   Acquire frame z_t
10. $n = 0$
11.   **for** $k = 1, \ldots, K$ **do**
12.   Compute $l^k(t)$, $\frac{\partial l^k}{\partial t}(t)$ for the region R_k
13.   **if** $\frac{\partial l^k}{\partial t}(t) < \Gamma^k_{min} \vee \frac{\partial l^k}{\partial t}(t) > \Gamma^k_{max}$ **then**
14. $n = n + 1$
        end
    end
15.   **if** $n \geq K - 1$ **then**
16. z_t is a displaced frame
    end
end

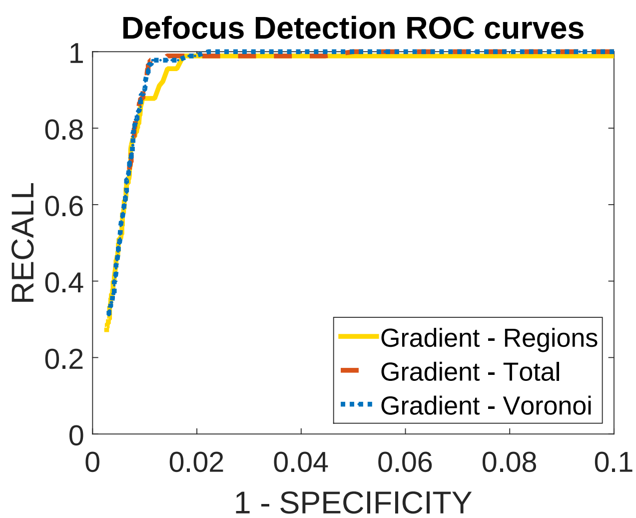

Fig. 1. Defocus

Acknowledgments

Authors would like to thank ST for supporting Adriano Gaibotti.

References

1. Anil Aksay, Alptekin Temizel, and A. Enis Cetin. Camera tamper detection using wavelet analysis for video surveillance. In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, pages 558–562. IEEE, 2007.