



# University of HUDDERSFIELD

## University of Huddersfield Repository

O'Grady, Michael and Wang, Jing

Graffiti detection: Automated crime detection in security camera feeds

### Original Citation

O'Grady, Michael and Wang, Jing (2012) Graffiti detection: Automated crime detection in security camera feeds. *Global Forensic Science Today* (12). pp. 4-15.

This version is available at <http://eprints.hud.ac.uk/id/eprint/17765/>

The University Repository is a digital collection of the research output of the University, available on Open Access. Copyright and Moral Rights for the items on this site are retained by the individual author and/or other copyright owners. Users may access full items free of charge; copies of full text items generally can be reproduced, displayed or performed and given to third parties in any format or medium for personal research or study, educational or not-for-profit purposes without prior permission or charge, provided:

- The authors, title and full bibliographic details is credited in any copy;
- A hyperlink and/or URL is included for the original metadata page; and
- The content is not changed in any way.

For more information, including our policy and submission procedure, please contact the Repository Team at: [E.mailbox@hud.ac.uk](mailto:E.mailbox@hud.ac.uk).

<http://eprints.hud.ac.uk/>

# **Graffiti detection: Automated crime detection in security camera feeds**

---

---

**Dr Michael O'Grady & Dr Jing Wang**  
**Department of Informatics,**  
**School of Computing and Engineering,**  
**University of Huddersfield,**  
**Queensgate, Huddersfield,**  
**HD1 3DH, UK.**

[m.ogrady@hud.ac.uk](mailto:m.ogrady@hud.ac.uk)  
[j.wang2@hud.ac.uk](mailto:j.wang2@hud.ac.uk)

## **1. Introduction**

The use of CCTV is prevalent in many towns and cities around the developed world. Used for their deterrent, real-time intelligence and evidence providing abilities, they serve a useful purpose, especially in social and open settings. However, for specific problem areas, or local hotspots of anti-social behaviour, a more focussed solution is desirable: specific camera positions; set angles of view; picture clarity; and recording ability. The author has been working with a company providing specialised security camera installations for regional Passenger Transport Executives (PTEs) in the UK, organisations comprising: Local Authorities; bus companies; and local Police forces. Anti-social behaviour can be a predominant deterrent against using UK public transport, where it is not necessarily the level of crime that is a problem, but the public's perception of crime, that stops them using buses. Damaged bus shelters can provide a negative reinforcement in the form of: graffiti; broken or missing glass; scratched paintwork; as well as damaged advertising and timetable frames.

Camera installations are generally at bus shelters in verified problem locations and include a series of individual protected cameras, a storage device and wireless transmitter. Accompanying advertisements along the bus route inform all passengers and users of the system that video surveillance may be in use. High definition continuous video streams from installations allow evidence from crimes and anti-social behaviour to be processed, but with typically eight cameras on a single shelter, it can take many man-hours to sift through raw footage unless events can be pinpointed in advance.

This article discusses the start of a process to automate the detection of prescribed events in real-time by software embedded on the installation hardware. This will allow alarms to be triggered resulting in:

- Detection of specific events: damage, graffiti, fighting etc.;
- Messages to be directed to appropriate authorities as events are happening;
- Placement of digital markers pointing to events on stored files for easy and rapid human assessment when required;
- More rapid production of video evidence packs for Police and Crown Prosecution Service personnel.

Detection of graffiti was investigated first, partly because it can be detected easily by permanent pixel colour change in successive video frames and partly because it involves the ability to detect involvement of humans with the Omega ( $\Omega$ ) head detection process, outlined below.

## **2. Methodology**

In trying to automate analysis of the video streams, it is necessary to cover the gap between: pixel elements on the images of each frame (typically running at 25 frames per second); and the resulting knowledge that an individual is in the process of creating graffiti. The research covered in this article splits this gap into three steps and these are covered in some detail below:

- data construction;
- feature extraction; and
- pattern recognition.

### **2.1 Data Construction**

Introduced in 1985 by Aldelson and Bergen [1], the Spatio-Temporal Volume (STV) data structure was used in this research to indicate embedded temporal-related features in video streams. Figure 1 shows the concept of a stack of individual video frames in consecutive time order (left) and a solid 3D volume created from stacked image frames (right). STV allows mathematical differentiation of dynamic events which is useful when considering human movement. The human torso is conveniently reduced to a stick-man frame (Figure 2) with the joints and extremities matched to the voxel (the STV equivalent of a 2D image pixel) [2]. When points of interest are plotted in STV space, results such as arm waving are

produced (Figure 3).

Using STV as a starting point for event detection, it was discovered that matching extracted 3D volume curve patterns with the many predetermined templates was leading to unacceptable errors: extracted curves from different events may have similar features and so an additional step involving curvature feature characterisation was progressed. This additional approach involves analysis of 2D curve geometries and a String-based 3D curve matching process involving torsional features (the latter is not covered in this article).

Figure 1. STV structure

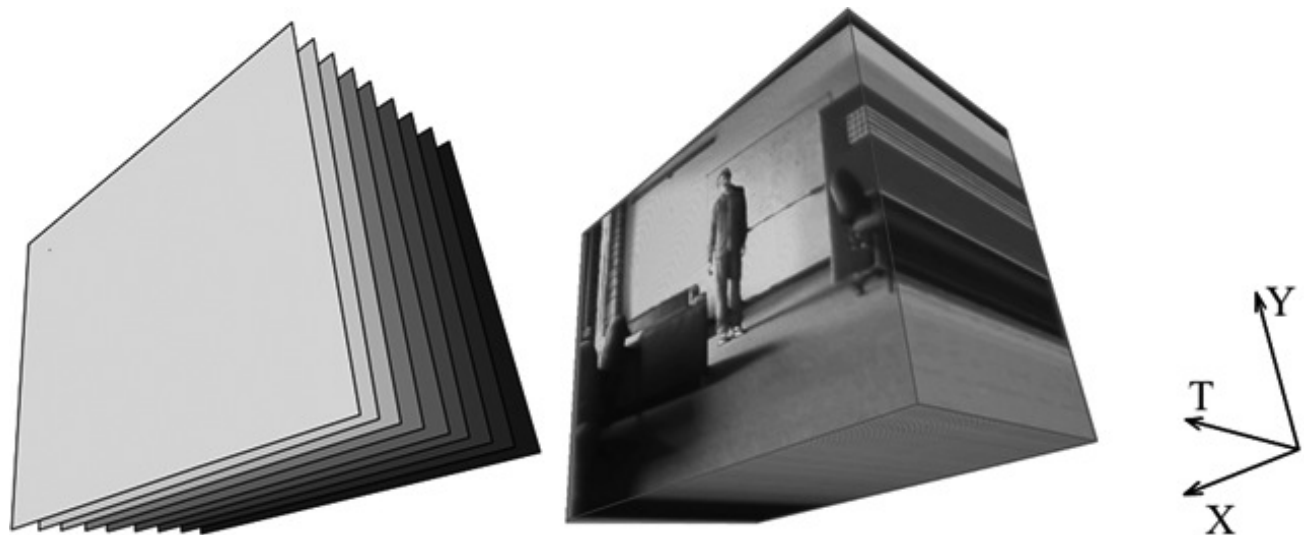
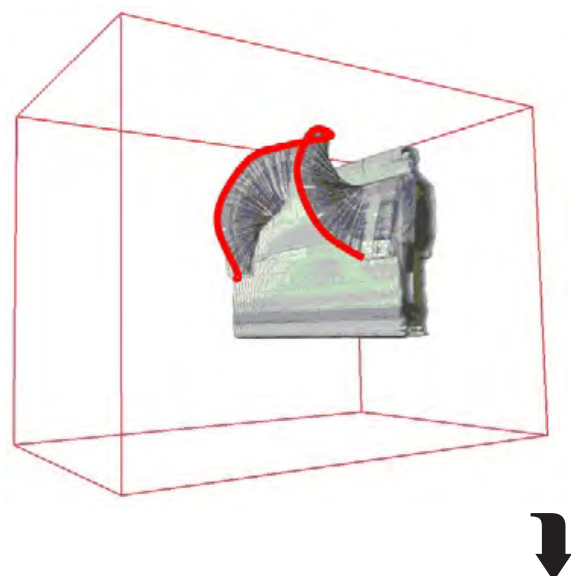


Figure 2. Human stick-model



Figure 3. Waving event curve in STV feature space



## 2.2 Feature Extraction – Head Detection and Curve Matching

Human body detection has become a research hotspot during the last decade. Accurate human modelling is based on effective human shape segmentation from 2D images and can be applied to video content analysis applications. This research develops two flexible human detection approaches based on popular pattern recognition techniques:

- a method based on the Haar-like features and the AdaBoost classification algorithm [3];
- an algorithm based on a head detection approach through curve mapping.

The former requires a time-consuming pattern

learning phase whilst the latter does not and will directly search for the contours of head shapes defined by geometrical curves. The Haar-like feature-based method carries out classification operations through multiple thresholding and learning processes. A more direct method is to track only the head of a human body and then combine the filtered results of other likley body parts such a limbs and torso, based on spacial distances. Compared with the complex task of modelling human bodies and postures, a human head is much simpler to define since it is a relatively rigid part of the human body. More importantly, all people seem to show identical head contours, an Omega ( $\Omega$ ) shape, even when facing different directions as shown in Figure 4.



Figure 4. Heads with Matching  $\Omega$  Contours

The  $\Omega$  shape human head detection was introduced by Zhao in 2002 [4, 5], where the locations of human heads were used for human ellipsoid model initialisation, but this requires calibration and proportion assumption. Therefore, the rotation and scale changes of the head in a scene are hard to detect using only a single  $\Omega$  shape template. Additional work was undertaken

to improve this approach, based on 2D open curve matching algorithms. The modified method developed can detect a human head by matching an  $\Omega$  shape without a pre-requisite knowledge of human head proportion, or rotation and scale registration. The algorithm is illustrated as a flowchart in Figure 5.

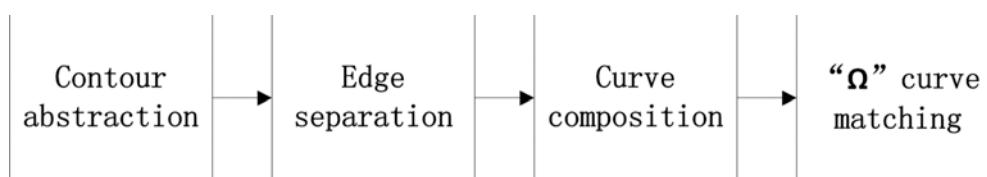


Fig.5 Head Curve Matching Flowchart

## 2.2.1 Contour abstraction

Head contours are commonly recognised as a collection of edges extracted from intensity images. Various popular filters can be adopted for this purpose, for example, Zhao [4] has chosen the Canny algorithm [6] to highlight the foreground of the human silhouette. In this project, this operation has been improved in terms of flexibility by using edges generated from the Mean-shift (MS) segmentation algorithm [7] and to deploy the Roberts' edge detection techniques. Different from the Canny filter, the MS-based method can display different colour intensity if the segmentation result is marked by different values. Edge pixels sharing the same value relate to pixels from a same curve. Multiple sets of those edge pixels are defined as edge groups which can be readily used for further curve reconstruction as shown in figure 6.



Figure 6. MS based Edge Groups

## 2.2.2 Edge separation

This step separates different edges into different groups by assessing their intensity values (set as labels), which are realised through manipulating the threshold values for each group. As shown Figure 7, a threshold value has been assessed to highlight edge groups with the same intensity value. Different groups holding same labels might be identified by connectivity. Figure 8 shows all the identified edge groups from Figure 7.

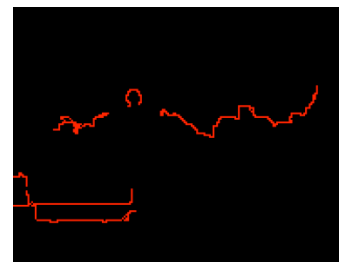


Figure 7. Example of Edge Groups with same Label

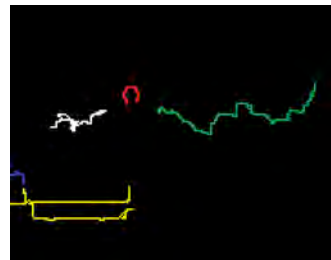


Figure 8. Labelled Edge Groups

## 2.2.3 Curve composition

A labelled edge group is just a container for a set of pixels. The  $\Omega$  contour matching operation requires a unified coordinate system which enables the linking of pixels into curves by exercising curve composition algorithms. A composition algorithm is implemented through evaluating neighbourhood edge pixels based on Hart's method [8]. Firstly, it applies the so-called hit-miss transformation [9] to locate the start and end locations in one edge group by applying eight different templates as shown in Figure 9. These templates cover all curve start and end point combinations. The process then projects the curve from the start position towards the end following directions determined by the distribution of the group of edges. The coordinate series of this curve can then be assembled by the tracking order.

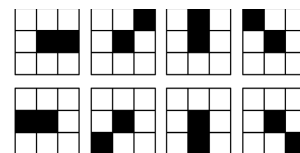


Figure 9. Hit-Miss Transformation Templates



### 2.2.4 $\Omega$ curve matching

The research has adopted a correlation curve matching algorithm introduced by Cui in 2009 [10], which is an efficient matching approach for 2D open curves, especially when the template curve is a subset of another. The basic theory is to match two curves by correlating and curvature mapping through employing a curvature integral. The curvature description and correlation operation significantly reduce the problem caused by scaling and rotational transformations. Compared with Zhao's work [5], this approach needs only a single  $\Omega$  curve template to describe a head shape without the time-consuming tasks of calibration.

Typical development and testing results of the algorithm are shown in Figure 10.A and 10.B, where the system starts from calculating the curvature defined by points outlining the distribution of an  $\Omega$  curve, after noise removal using the B-spline approximation [11], the projectile is then generated by integrating of curvature as shown in the Figure 10.C. The final operation of the system is to correlate the template and pattern curves by using the integrals of the curvature mapping distributions as shows in Figure 10.D.

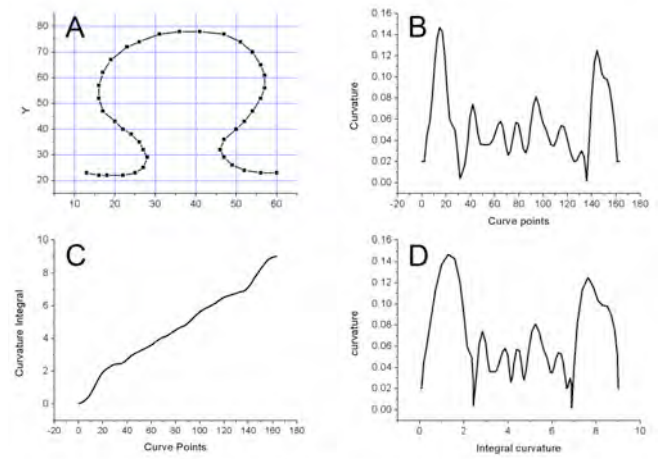


Figure 10.  $\Omega$  Curve Curvature Features

Due to the over-segmentation problem often occurring in MS applications, a number of curves appeared as short and separated pieces in this project especially with complex backgrounds. For example, as shown in Figure 6, the curve outlining the head is actually formed from two sections. This increases the challenge curve matching practice

when sections of a curve are missing. Therefore, related curves need to be identified before matching the  $\Omega$  template. Figure 11 shows a match-making connection of some related curves, which illustrates a curve clearly and can be used for curve matching directly to locate the human head coordinates as shown in Figure 12.

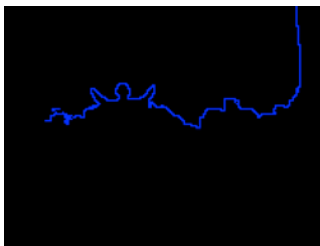


Figure 11.  $\Omega$  Curve Rebuild Result

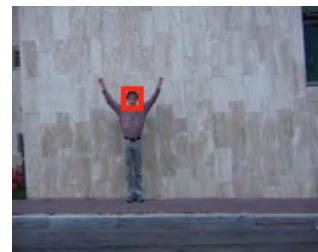


Figure 12. Head Detection Result



### 2.3 Pattern Recognition - 2D open curve matching

After defining the start point of a 2D curve, the curvature parameter sequence can be realized as a string which denotes the specific 2D curve in the

feature space. As shown in the Figure 13, the left hand graph is the geometrical distribution of a simple curve starting from the round dot and the right hand plot shows the absolute values of the curvature.

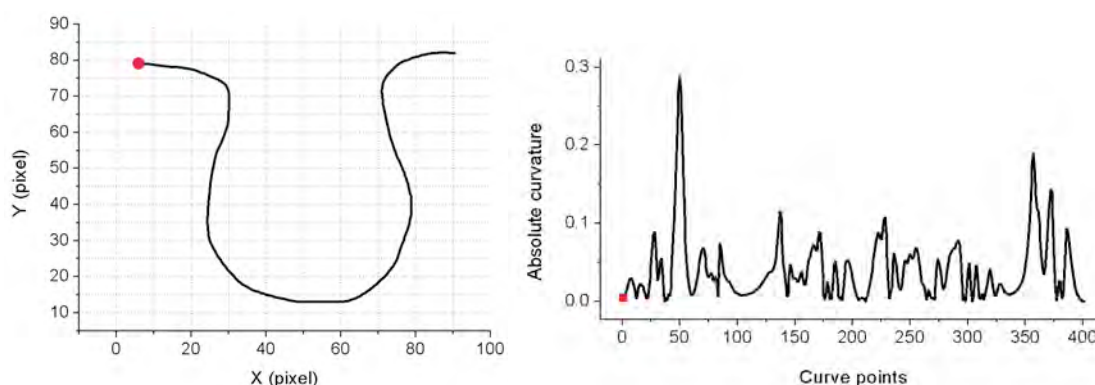


Figure 13. 2D Curve and Absolute Curvature

The 2D open curve matching can be represented by following algorithm:

Given two curves  $C_1 = \{(x_{11}, y_{11}), (x_{12}, y_{12}), \dots, (x_{1n}, y_{1n})\}$  and  $C_2 = \{(x_{21}, y_{21}), (x_{22}, y_{22}), \dots, (x_{2m}, y_{2m})\}$  in an Euclidean space, each one can be denoted by its curvature  $K$ , in the form of  $C_1 = \{K_{11}, K_{12}, \dots, K_{1n}\}$  and  $C_2 = \{K_{21}, K_{22}, \dots, K_{2m}\}$ . Initialise a  $(n + 1) \times (m + 1)$  matrix named as distance matrix  $D(i, j)$ , this string edit operation computes each element of  $D(i, j)$  by the edit algorithm. Shown as Equation 1.

$$D(i, j) = d(C_1, C_2) \quad C_1 = \{K_{11}, K_{12}, \dots, K_{1i}\}, C_2 = \{K_{21}, K_{22}, \dots, K_{2j}\}$$

The lower right corner value of the matrix records the difference of the two curves. Each element of this matrix is calculated based on a previous iteration operation. As shown in the Figure 14, there are three probable predecessors, each one denoting one specific string operation:  $D(i-1, j-1)$  -substitute,  $D(i-1, j)$  -delete, and  $D(i, j-1)$  -insert. In addition, it can also be used to trace the changes from one curve into another one based on the three string edit operations. The complexity of this algorithm is of  $O(nm)$ .

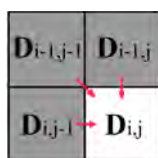


Figure 14. String Edit Operation



Pseudocode for implementing the algorithm in the experiment is shown below:

```
Pseudocode stringEdit (C1,C2)  
START  
//Initialization  
int n = length of C1;  
int m = length of C2;  
array D[n+1][m+1] = 0;  
//Initialization first row and column of D  
For i = 1 to n  
    D[i][0] = D[i-1][0] + c(deletei0);  
For j = 1 to m  
    D [0][j] = D[0][j-1] + c(insert0j);  
//Distance iterative calculation  
For i = 1 to n  
    For j = 1 to m {  
        m1 = D[i-1][j-1] + c(substituteij);  
        m2 = D[i-1][j] + c(deleteij);  
        m3 = D [i][j-1] + c(insertij);  
        D [i][j] = min (m1, m2 , m3);  
    }  
//Output result  
float difference = D [n][m]  
END Pseudocode
```

In the pseudocode, a function of operation cost is introduced. It defines the cost of the three edit operations. The rules can be changed based on different applications. For this 2D curve matching problem, the cost functions are defined as:

$$c(\text{delete}_{ij}) = c(\text{insert}_{ij}) = 0.1, \quad c(\text{substitute}_{ij}) = |K_i - K_j| \quad (2)$$

### 3. Test Results

The experiments were designed and carried out on: on MS-based curve reconstruction;  $\Omega$  shape recognition for head detection; as well as 3D-based work not discussed here (volume-based curve geometry analysis and 3D feature curve matching). Results of the first two are presented below.

#### 3.1 Mean-Shift Curve Reconstruction

The MS algorithm is implemented based on Comaniciu and Meer's research [7]. A number of edge detection techniques have been also tested in the experiment. The results are shown in the

Figure 15. Based on the detection outputs, the most suitable technique for MS curve reconstruction is Roberts's method [12] (figure 15, F). During the experiments, it is observed that the Laplacian method [13] (Figure 15, B) is over-sensitive at corners of some curves. It can over-divide curves into many small horizontal and vertical short edges, which brings too many start and end points for curve recognition. Similar problems occurs when adopting Prewitt [6] (Figure 15, D), Sobel and Feldman [14] (Figure 15, E) and Diff. Method[12]. In comparison, Roberts's approach can better resolve those



problems due to its extra consideration on different edge directions. This distinctive feature is particularly suitable for forming edges after MS segmentation.

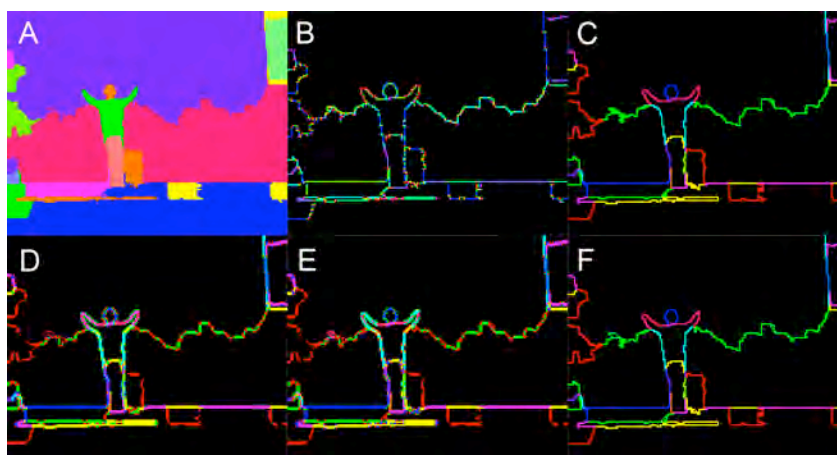


Figure 15. MS Contours Experiment Result

The generated edge groups are then composed into separated curve sections by using the neighbourhood edge pixels connectivity algorithm based on Hart's work [9]. Although the over-segmentation problem persists, this experiment has proven the validity of using the MS segmentation

technique for curve reconstruction. The experiment design has adopted videos from the Weizmann video library. Table 1 shows testing results of MS-based curve reconstruction from sample frames of different clips.

Data name	Frame Index	Sum of Edges Pixels	Sum of Curve Pixels	Curve Numbers	Sum of End Points	Connected Curve Numbers
daria_wave2	11	1402	1323	37	74	12
denis_wave2	31	2033	1979	59	118	20
eli_wave1	5	1762	1211	43	86	13
ido_wave1	24	1919	1408	30	60	22
ira_walk	1	2116	2014	54	108	14
lena_walk1	70	2018	1895	46	92	20
moshe_bend	48	1209	1045	28	56	7
shahar_bend	30	1698	1216	31	62	13
lyova_jump	20	2708	2366	44	88	19
daria_jump	15	1912	1675	23	46	13

Table 1. Curve Reorganisation Experiment Result for Selected Events from Weizmann Database

