

Probabilistic Tracking of Moving Hands in Video Sequences with Applications to Dynamic Gesture Recognition on Mobile Handsets

Addmore Machanja and Vladimir B Bajic**

Department of Computer Science

University of the Western Cape; P. Bag X17; Bellville 7535; South Africa

Email: amachanja@uwc.ac.za

**South African National Bioinformatics Institute (SANBI)

University of the Western Cape, South Africa

Abstract—The ability for computers to visually recognize and track human hand motion is important for a wide range of applications in the field of Human-Computer Interaction (HCI). Though it is effortless for the human eye to locate and track gesturing hands in video sequences, it is far more complex for computers to achieve this tracking. Hand tracking typically fails in the absence of perfect hand segmentation. Many hand movement video-based segmentation algorithms have been proposed in the literature. However, these algorithms remain largely inefficient, imprecise and insufficient. This paper presents a robust segmentation and tracking algorithm. The algorithm first extracts a small list of candidate hand regions from the image data by combining edge, motion and skin color information. The resulting skin-colored maps are usually contaminated by noise, and hence morphological filtering techniques are applied to alleviate the impact of noise. We implement an improved border-tracing algorithm in order to isolate connected regions in the output image. A statistical feature set that uniquely defines each connected region is extracted. In each subsequent image frame, the tracking algorithm aligns each segmented region with the best candidate hand region from the previous set of hand candidates. This is achieved by grouping together all regions with similar data patterns and disposing of all regions with non-persistent data patterns. Simulation results of our algorithm demonstrate improved segmentation and tracking results.

Keywords: hand tracking, dynamic gesture, hand segmentation, boundary- tracing

1 INTRODUCTION

Computer vision researchers often face the need to detect and track objects in images. According to Fei and Reid [14], depending on the complexity of the task at hand, hand tracking is achieved using either kinematic models or appearance-based models. Although kinematic model-based tracking methods provide detailed information about object configurations, the process of fitting in the model during tracking is cumbersome and computationally expensive. More often, kinematic models fail to maintain tracking if the object changes its shape rapidly after short time intervals. On the other hand, appearance-based tracking models are fast but are heavily dependant on the

accuracy of the segmentation results. Despite the numerous research projects dedicated to computer vision, no pervasive generic segmentation algorithm has been reported as most applications require a careful study of the alternatives and even the invention of new techniques [4]. In vision-based dynamic gesture recognition systems, accurate hand segmentation is a requirement, otherwise crucial information about a gesture would be missed [7]. Threshold-based segmentation techniques are classified under appearance-based models. During thresholding, objects are classified according to some predefined similarity criteria that are estimated from some visual features of an image [9]. Motion [6][7], skin color, texture, and/or edges are traditionally used to segment gesturing hands. However, these cues individually fail to unambiguously locate and track the hands as they often output multiple candidate regions.

We propose improved hand segmentation and tracking algorithm that utilizes several features, such as skin-color, motion and edge information. The new feature integration scheme incorporates the Support Vector Machine (SVM) [20] as a tool for hand shape verification. The implementation of the algorithm is evaluated on video sequences of people using sign language.

2 BACKGROUND WORK

Computer vision researchers often recognize objects by analyzing data from their crude outlines [1][2]. Edges typically characterize object boundaries; and edge-based segmentation is one of the earliest segmentation approaches to be used by computer vision researchers. However, edges merely represent regions of sharp lighting discontinuities [3]; hence not all edges represent image boundaries. Some edges are generated by noisy conditions in the image data [4]. On the other hand, if two or more similarly colored regions overlap, sometimes no edges are detected in the region of overlap even if true image boundaries exist in that region. In order to compute acceptable image boundaries, additional constraints, such as object shape, color, and texture are often required. Object boundaries identify connected components in an image. In our algorithm, once connected components are established, objects are isolated and tracked basing on a statistical feature set, $F(v_1, v_2, \dots, v_n)$, derived from each connected region [5][6].

Threshold-based segmentation methods are suitable for applications where high processing speeds and automatic processing are required. In the case of dynamic gesture recognition systems, it has been noted that skin color enables fast and a fairly robust way of segmenting gesturing hands under partial occlusion, changing camera resolutions and/or fluctuating lighting conditions [11][12][13]. However, skin color alone is not a sufficient cue as it fails to distinguish hands from other skin-colored objects that are often found in the image background. In some vision-based gesture recognition researches, coarse motion data is used for segmenting moving hands [6]. Crude motion maps are obtained by thresholding the difference in light intensities between corresponding pixels that are contained in any two consecutive image frames as depicted by the following equation.

$$D(x, y, t) = \begin{cases} 0: |I_1(x, y, t) - I_2(x, y, t')| < T \\ 1: |I_1(x, y, t) - I_2(x, y, t')| \geq T \end{cases}$$

In short, the above equation says that the motion $D(x, y, t)$ at pixel $P(x, y)$ at time t is 0 if the modulus of the difference in intensities, $|I_1(x, y, t) - I_2(x, y, t')|$, between pixel $P(x, y)$ and the corresponding pixel $P'(x, y)$ from the preceding image frame, I_1 and I_2 , is less than a given threshold T . Otherwise $D(x, y, t)$ is 1. The resultant motion map is sometimes called the difference image or the Motion History Images (MHI) [6]. It is a scalar quantity that provides coarse motion information. On the other hand, optical flow methods compute motion as a vector quantity and represent motion at each pixel by its magnitude and its direction. Optical flow methods generally have a high computational complexity. Although it is easy to compute pixel motion using the MHI approach, the resultant motion maps also represent fluctuating lighting conditions, moving background objects, and/or small jagged movements of a camera as part of hand motion. This compromises the accuracy of the hand segmentation and tracking processes.

Human hands are highly flexible objects. It is very difficult to simultaneously estimate hand poses and track hand motion in video images. Segmentation results obtained exclusively from motion maps or skin color maps are not adequate for comprehensive gesture recognition. In many gesture recognition researches, significantly better segmentation results are attained after fusing together skin color, motion, edges, and/or other visual cues [10][11][17]. Suat [10] further demonstrated that segmentation is faster when motion and skin color information are processed simultaneously than when they are processed sequentially [15]. However, Suat's work requires an offline histogram based skin-color detection process. He isolates object boundaries by implementing the watershed algorithm. The watershed algorithm often produces over-segmented images which may require further processing and this increases the computational complexity of Suat's method.

It has been found that skin colors of different races of people are narrowly clustered in a 2-D chromatic color space [7][8][12]. The perceived differences in human skin appearances are premised on skin color intensities. Various chromatic color spaces have been identified for image processing. The YCrCb, HSV, RGB and the normalized RGB are some of the most commonly used color spaces [12][13].

The HSV color space is more distinctive and less sensitive to color changes than both the YCrCb and the normalized RGB color spaces [13][16]. However, most videos are available in YCrCb format, while the process of transforming video data from one color space into another is time-consuming [7][12][14]. Chai and Ngan [12] also found that the Cr and Cb values of most skin colored objects are contained in the ranges, $Cr = [133, 173]$ and $Cb = [77, 127]$ of the YCrCb color space. This discovery makes the YCrCb format an attractive alternative for real-time applications where no prior skin color sampling is required.

3 SEGMENTING AND TRACKING GESTURING HANDS

Perfect or close to perfect hand segmentation is difficult to achieve as it involves delineating the moving hand regions from a complex background that often includes other moving objects [16]. When the hand moves fast, the image is blurred and the hand contours cannot be detected accurately [17]. Accurate segmentation is a crucial step towards achieving high level hand tracking in video streams. Since the human hand is highly flexible, its segmentation cannot solely depend on hand shape information. In this research, we fuse skin color, motion and edge information in order to segment moving skin colored regions. The resultant bitmaps of skin colored regions are passed through several morphological filters that remove some noise and regularize the bitmap regions into meaningful sections. Before a statistical feature set that describes each region is determined, the boundaries of each moving skin colored region are identified. Also, before applying a tracking algorithm, all skin colored blobs whose surface area are less than a given threshold, A , are discarded.

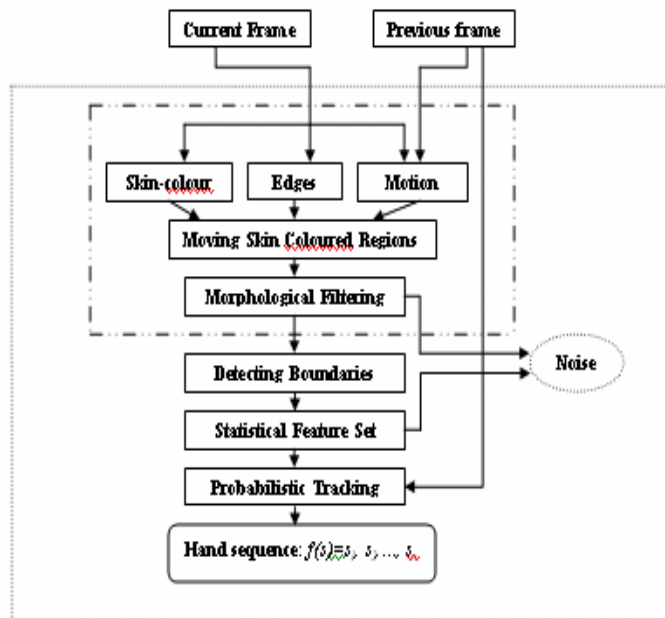


Figure 1: System Design

Discarding the small isolated blobs help to reduce the number of candidate hand regions available in each image frame. The tracking process is achieved through a probabilistic alignment of each candidate region from previous frames, with

a best matching candidate region from the current frame. A gesturing hand is deemed successfully tracked if a sequential count of the aligned skin colored regions is greater than half the number of image frames processed from each gesture. The following four subsections explain the details of the hand segmentation and the process of tracking gesturing hands.

3.1 Segmenting Gesturing Hands

The YCrCb color space is best suited for unsupervised hand tracking applications for which histogram based skin color thresholding is not a convenient approach. According to Chai [12], Cr values ranging from 128 to 173 and Cb values ranging from 77 to 132 are most suitable for isolating skin colors of persons of different ethnic origins. However, if the background objects include polished wooden materials and other brown colored objects, then, many false positives will be created. Some of the extracted skin-colored regions are not true skin regions. In order to increase the probability of extracting true skin regions, $F(x,y,t)$, a joint distribution of motion, skin color and edge information is extracted as shown in the equation below:

$$F(x, y, t) = \begin{cases} 0 : S_c(x, y, t) = 0 \\ 1 : S_c(x, y, t) > 0 \cap (|D(x, y, t)| > T_1 \cap Ed(x, y, t) > 0) \cup (Ed(x, y, t) = 0 \cap |D(x, y, t)| > T_2) \end{cases} \quad T_1 < T_2$$

where $S_c(x,y,t)$ and $Ed(x,y,t)$ are binary values that represent skin color and edge information respectively. $|D(x,y,t)|$ is the magnitude of motion at pixel $P(x,y)$ of the input image X at time t . The magnitude of motion exhibited at each pixel is obtained by computing the intensity difference between successive image frames. T_1 and T_2 are threshold values for determining the motion exhibited by edge-based and non-edge-based skin-colored pixels respectively. A very small motion threshold is ideal for capturing small movements, especially fingertip motion which is important for the gesture recognition process. However, a small threshold amplifies the negative effects of noise on the segmentation results. On the other hand, if only motion and color cues are used, high threshold values lead to loss of some hand shape information. In this research we use a very small motion threshold, T_1 , for edge based pixels so as to preserve possible hand boundaries in the resultant skin-colored bitmap. At the same time we use slightly higher values of T_2 since such values reduce the impact of background noises on the overall segmentation result. In the resulting binary image, the moving skin-colored pixels are represented by '1's, and the rest of the image pixels are represented by '0's. However, as in most cases, segmentation results always contain some forms of noise. The side effects of noise on the output data can be mitigated by applying some morphological filters.

3.2 Filtering the Skin-Colored Regions

The bitmap region is first subdivided into 3 x 3 pixel sized sub-regions. The algorithm eliminates all skin-colored pixels in each sub-region if that sub-region has less than x moving skin colored pixels and if the sub-region is also totally surrounded by sub-regions with less than $x-1$ skin-colored pixels. In our case, we used x equal to 5. We eliminate some undesirable thin elongated regions by increasing the size of the sub-regions. If 6 x 6 sub-regions are used, all partitions with less than 7 skin-colored pixels that are totally surrounded by

sub-regions with less than 7 skin-colored pixels are eroded. Erosion of the isolated skin colored regions reduces the effect of noise on the output bitmap. In a 24 x 24 sub-region category, all sub-regions with less than 150 moving skin colored pixels that are completely surrounded with sub-regions with less than 40 moving skin colored pixels are eroded. We implement a parameterized module that automatically adjusts the sizes of the image sub-regions. If the dimensions of the next sub-region are double of those of the preceding sub-region, then the process of determining the number of moving skin colored pixels contained in a bigger sub-region is reduced to merely summing the totals of the four smaller sub-regions encompassed in one big sub-region. This approach reduces the computational complexity of our algorithm. In this algorithm, dilation is sparingly applied since it often distorts object boundaries especially where finger-like projections are present.

3.3 Boundary Tracing and Component Labeling

Boundaries of the resulting binary image are used to isolate connected image components. Starting from the top most pixel of each region, we search for image boundaries by analyzing the 8-connectivity of the skin colored pixels (see figure 2). A skin-colored pixel $P(x,y)$ is directly connected to one or more of its neighborhood pixels $P(i,j)$ if $P(i,j)$ is also skin-colored and is preceded by or proceeds a non-skin colored pixel; where $P(i,j) \in \{P(x+1,y+1), P(x,y+1), P(x-1,y+1), P(x-1,y), P(x-1,y-1), P(x,y-1), P(x+1,y-1), P(x+1,y)\}$.

$P_{(x-1,y-1)}$	$P_{(x,y-1)}$	$P_{(x+1,y-1)}$
$P_{(x-1,y)}$	$P_{(x,y)}$	$P_{(x+1,y)}$
$P_{(x-1,y+1)}$	$P_{(x,y+1)}$	$P_{(x+1,y+1)}$

a) 8-Neighboring Pixels of $P(x,y)$

3	2	1
4	$P_{(x,y)}$	0
5	6	7

b) Tracing Directions

Figure 2: Neighbor Pixels and Tracing Direction

The problem of finding connected components has been extensively researched [4][18][19]. Although the accuracy of the binarization thresholds directly impacts on the resulting bitmap region, it is very difficult to find threshold values that effectively isolate the regions of interest (ROIs) [4]. As a result, object boundaries are often corrupted by noise and are difficult to trace due to their jagged nature. Traditional boundary tracing algorithms loop through all boundary pixels until the first and the second pixels visited at the start of the tracing process are revisited in the same order as they were first encountered. However, this approach may either result in premature exits or endless loops if some sections of the object boundaries are corrupted by noise. We propose a boundary tracing algorithm that scan and mark each boundary pixel once, and stops scanning when all connected boundary pixels are visited once. In order to ensure that all boundary pixels are visited during the boundary tracing process, our algorithm scans an image in both clockwise and anticlockwise directions. Twelve possible external boundary views for each image are used to determine the direction and the position from which boundary tracing should resume, in the event that a dead end is met before all boundary pixels are traced. This boundary tracing algorithms locates all fingerlike projections that are associated with each region of interest. Some of the

fundamental steps of the improved boundary tracing algorithm are outlined below.

Improved Boundary Tracing Algorithm:

1. Set $dir \leftarrow 4$.
2. Scan image pixels from top to bottom, left to right until a skin-colored pixel $P(x,y)$ is found
3. Set $P(x,y)$ to white and assign a negative value to it; check whether $P(x+1,y)$ is also skin-colored
 - a. If $P(x+1,y)$ is skin-colored, set $M \leftarrow -5$ else $M \leftarrow 0$;
 - b. Push point $P(x,y)$ and $dir=1$ into clockwise stack;
4. REPEAT UNTIL all anticlockwise boundary pixels are visited
 - a. $dir = [(dir+1) \text{ MOD } 8]$
 - b. if current pixel, $P(i,j)$, is skin-colored
 - i. set $P(i,j)$ to white and assign it a negative value;
 - ii. Analyze image pattern and determine $dir2$
 - If pattern $\in \{12 \text{ pattern set}\}$ then
 - Push $P(i,j)$ and $dir2$ into either clockwise or anticlockwise stack
 - iii. If dir is odd then $dir = [(dir+5) \text{ MOD } 8]$ else $dir = [(dir+6) \text{ MOD } 8]$
 - c. If $P(i,j)$ is negative, POP UP anticlockwise stack; **else if** anticlockwise stack is empty CALL clockwise tracing module **else if** clockwise stack is empty, then exit trace
5. Clockwise Module
 - REPEAT UNTIL clockwise stack is empty
 - a. $dir = [(dir-1) \text{ MOD } 8]$
 - If $dir == -1$ then $dir == 7$
 - b. Repeat STEP 4. b. i) and 4. b. ii) above
 - a. if dir is odd then $dir = [(dir+3) \text{ MOD } 8]$ else $dir = [(dir+2) \text{ MOD } 8]$
 - c. If $P(i,j)$ is negative, Pop up the clockwise stack; **else if** anticlockwise stack is empty Return to anticlockwise tracing module

3.4 Tracking Probable Hand Regions

The head region is an elliptical shaped region covering a significantly large surface area when compared to the rest of the skin colored blobs. It is usually situated at the central-top part of an image. On the other hand, the hand region is an elongated region that often has a steadily increasing or decreasing cross-sectional distance. Hand regions are often characterized by finger-like projections at one end. We also differentiate the hand from the head by comparing the sizes of the major axis t_0 and the minor axis t_1 of their bounding ellipses.

The segmentation result for each frame is a set of probable hand and head regions. A hand is deemed successfully tracked if consecutive candidate hand regions from more than half the total number of input frames are successfully matched. The matching process is based on three factors; namely total surface area, position, and the x and y standard deviations of the skin colored blob. At first, the surface area and the two standard deviations for each candidate hand region are matched with those from the previous frame. If the difference between the areas and the standard deviations of any two skin colored blobs, which belong to different image frames, are

below some prescribed thresholds and if no other blobs have comparatively similar differences, then a match is found. On the other hand, if more than one candidate from either the current frame or the previous frame produces almost similar differences, then the best match will be decided basing on the probable motion vectors. All factors being equal, a blob which suffered little motion is considered the best match. Each candidate region is only aligned to one or none of the candidates regions from the other frames.

4 EXPERIMENTAL RESULTS

Our hand segmentation and tracking algorithm was tested on video sequences of people using sign language. Firstly, consecutive image frames were extracted from the videos and fed into our program. Samples of the segmentation results produced by this algorithm are shown Figure 3. We used SVM with a linear kernel to classify the hand configurations that associated with each gesture. In the SVM module, a high specificity (SP) value indicates high recognition rates. Seven dynamic gestures from the SASL were repeatedly executed by five different people. On average each dynamic gesture exhibited 35 person-dependant variations of each hand shape, which in turn amounts to more than 175 person-independent hand shape variations for each gesture. Data obtained from different gesture classes and those obtained from noisy hand images are used as negative train samples; while data collected from a sequence of aligned hand blobs are used as positive train samples for each gesture. We also extracted separate lists of positive and negative test samples. Hand shape recognition was tested for both person independent and person-dependant situations. It is assumed that the hand configuration does not change much during the execution of a gesture, and hence a set of features that describe a hand configuration can be traced from the start frame to the end frame of each gesture. The adequacy of the hand shape estimation process for a cherokee-based dynamic gesture recognition system is evidenced by the high SVM recognition rates.



Figure 3: Samples of Segmentation Results.

Despite the fact that hand segmentation is very sensitive to noise, our algorithm achieved high quality segmentation and tracking results (see Figure 3 and Table 1). From Table 1, a decrease in the overall hand shape recognition rates is attributed to the large class variations that arise from the different ways in which different people execute the same sign. On the other hand, the 96.38% overall tracking rate does not reflect the various conditions to which each signing person is exposed since tracking is always person-dependant. These results indicate that our algorithm can effectively track the gesturing hands under complex background settings, whereas most available gesture recognition systems ignore the need to accurately segment and track the hand.

	Sent Frames	Tracked Frames	Tracked frames (%)	SVM (SP) recognition rates (%)
Person 1	40	38	95	83.33
Person 2	24	24	100	91.67
Person 3	44	40	90.91	100
Person 4	50	48	96	100
Person 5	21	21	100	86.67
Overall	179	171	96.38	85.67

Table 1: Analysis of person-dependent and person-independent segmentation and tracking results for gesture 1

5 CONCLUSION AND FUTURE WORK

Although our algorithm achieved commendable tracking results, currently it has not been tested for real-time video processing. We used still sequences of consecutive image frames in all our experiments, and it should be observed that it is computationally expensive to process large numbers of high dimensional still-images. Higher processing speeds can be achieved if prediction algorithms are used to determine consecutive hand position before segmenting the hand region in the preceding frame as this reduces the amount of information processed at the segmentation phase. This work also demonstrates that, irrespective of the complex background colors, gesticulating hands can be effectively isolated and tracked from the background objects.

6 REFERENCES

1. *Ballard D H, Brown C M; Computer Vision*; Prentice-Hall, 1982
2. *Russ J C; The Image Processing Hand Book*; CRC Press; 1995
3. *Sonka M, Hlavac V, Boyle R; Image Processing, Analysis and Machine Vision*; Brooks and Cole Publishing; 1998
4. *Yi L, Tie-Qi C, Jie C, Anthony T, Jiixin C; An Advanced Machine Vision System for VFD Inspection*; 6th International Conference on Manufacturing; 6-8 Sept 2000
5. *Qian R J, Sezan M I, Matthews K E; A Robust Real-Time Face Tracking Algorithm*; Vol 1, International Conference on Image Processing, Chicago, pp 131-134, IL, USA, 1998
6. *Rigoll G, Kosmala A, Eickenler S; High Performance Real-Time Gesture Recognition Using Hidden Markov Models*; Proceedings of the Gesture Workshop '97; pp 69; Germany; 1997
7. *Ribeiro H L, Gonzaga A; Hand Image Segmentation in Video Sequence by GMM: a comparative analysis*; XIX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAP'06), pp. 357-364, 2006.
8. *Alon J, Athitosos V, Yuan Q, Sclaroff S; Simultaneous Localization and Recognition of Dynamic Hand Gestures*; IEEE Workshop on Motion and Video Computing, Volume 2, pp 254-260, 2005
9. *Gonzalez R. C. and Woods R. E.; Digital Image Processing*; Prentice-Hall, International Edition, USA, 2002

10. *Suat A, Pablo A; Finding Relevant Image Content for mobile Sign Language Recognition*; Journal of Deaf Studies and Deaf Education, Vol. 11, No. 1, pg 94-101; 2005
11. *Jennings C.; Robust Finger Tracking with Multiple Cameras*; International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, pp 152-160, Corfu, Greece, 1999
12. *Chai D, Ngan K N; Locating Facial Region of a Head-and-Shoulder Color Image*; Conference of face and gesture recognition, pp 124-129, Nara, Japan; 1998
13. *Argyros A A., Lourakis M I A.; Real-Time Tracking of Multiple Skin-Colored Objects with a Possibly Moving Camera*; The 8th European Conference on Computer Vision - ECCV, Springer-Verlag, vol. 3, pp 368-379, Prague, Czech Republic, 2004
14. *Fei H, Reid I; Probabilistic Tracking and Recognition of Non-Rigid Hand Motion*; Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures; pg 60; 2003
15. *Hienz H, Grobel K, Offner G; Real-Time Hand-Arm Motion Analysis using a single Video Camera*; 2nd International Conference on Automatic Face and Gesture Recognition (FG '96); pp. 323; 1996
16. *Zhu Y, Ren H, Xu G, Lin X; Toward Real-time Human-computer Interaction with Continuous Dynamic Hand Gestures*; pg 544 – 549; 2000
17. *Awad G, Han Junwei, Sutherland A; A Unified System for Segmentation and Tracking of Face and Hands in Sign*; 18th International Conference on Pattern Recognition (ICPR'06) pp. 239-242, 2006
18. *Danielsson P E; An Improved Segmentation and Coding Algorithm for Binary and Nonbinary Images*; Image processing and Pattern Recognition, IBM Journal of Research and Development, Volume 26, No. 6, pg 698, 1982
19. *Sugiyana T, Kwan P W H, Toraichi K, Katagishi K; A Contour Tracing Algorithm that Avoids Duplicate Tracing Common Boundaries between Regions*; The Journal of the Institute of Image Electronics Engineers of Japan (Academic Journal), Vol. 33, no. 4-B, pg 586-596, 2004
20. *Thorsten Joachims; Transductive Inference for Text Classification using Support Vector Machines*; International Conference on Machine Learning (ICML), 1999

7 BIOGRAPHY

Addmore Machanja is a PhD student in the Computer Science Department at the University of the Western Cape (UWC). Prior to joining UWC, Addmore taught Discrete Mathematics, Data Structures and Algorithms at the University of Zimbabwe. Addmore is currently working on an automatic gesture recognition system but his previous research activities focused on optimizing queries used in Relational Database Management Systems.