iPanel: A Computer-Vision Based Solution for Interactive Keyboard and Mouse

H. Chathushka Dilhan Hettipathirana[1] and Pragathi Weerakoon[2]

[1] Department of Computing
Informatics Institute of Technology, Sri Lanka.
Collaboration with University of Westminster
h.hettipathirana@my.westminster.ac.lk
[2] Informatics Institute of Technology, Sri Lanka
Pragathi.w@iit.ac.lk

Abstract. This paper represents an implementation of a computer vision based interface; iPanel which employs an arbitrary panel and tip pointers as a spontaneous, wireless and mobility device. Also the proposed system can accurately identify the tip movements of the panel and simulate the relevant events on the target environment. By detecting the key pressing, mouse clicking and dragging actions, the system can fulfill many tasks. Therefore, it enables users to use their fingers naturally to interact with any application as well as with any mobility enabled devices.

Keywords: Computer vision, Human computer interaction, gesture recognition, optical character recognition, wearable computing.

1 Introduction

Human computer interaction is the technique of studding the relations between people and computer or computer mediated information. Thus it involves the design, development and evaluation of models, systems and applications from a human-centered perspective. Since its inception in the 1980s, HCI has been primarily concerned with designing more usable computer systems, attractive conventional computing devices, be it the computer desktop, the Web, or the mobile phone. It evaluates the existing designs and shows how to improve them. And, it attempts to apply its methods to design more user friendly systems from the start. Human-computer interaction comprise many sub domains such as gesture reconstruction, event detection, video tracking, object recognition, learning, indexing, motion estimation, and image restoration. Each sub domain is a unique concept of computer vision and it attempt to address a particular area of HCI, where the computers are pre-programmed to solve tasks or the interactions (e.g. touch screens, tablet PCs).

It has been identified and observed that many researches are adopting gesture reconstruction and ended up with implementing excellent results (e.g. Microsoft is researching on how user can interact with computers or computational devices in more efficient and user friendly manner [7]. Thus gesture recognition, sensor based

M. Kurosu (Ed.): Human-Computer Interaction, Part II, HCII 2014, LNCS 8511, pp. 36–47, 2014.

interactions and augmented realities are becoming more and more popular. This is the main reason that gesture recognition is more important and required to simplify the user interaction with the computers of computational devices. As an example, several people are discussing in a meeting room using a large display. They may need to draw, to suggest their ideas. However, it is unrealistic to facilitate every user a keyboard and a mouse. Even more, in a large lecture room, the lecturer sometimes needs to write down something on a small whiteboard. However, the audience far from him or remote audience may not be able to see clearly what he writes on the board. Therefore, need for a vision based system is necessary to analyze and understand what the lecturer writes and retrieve on a remote display, while avoiding bandwidth constraints. Furthermore, most of the smart mobile provide a QWERTY keyboard with tiny keys. It is really difficult to type with those keys. Yet, providing a large screen would lead to unnecessary problems such as size of the phone. In order to address the above a vision based solution has been suggested.

2 Human Computer Interaction

Human-Computer Interaction (HCI) is a technology researching on people, computer and the communications between them. Designing interactive computer systems to be effective, efficient, easy and enjoyable to use is important, so that people and society may realize the benefits of computation based devices such as mouse or keyboard. According to [2] use of these devices are recognized way of interaction with interfaces based on Window, Icon, Menu and Pointer (WIMP) paradigms which have succeeded for decades. Eventually software interfaces have got improved and interactive, lot of effort and code has been put behind the development of interactive software. Nonetheless, the use of traditional computational devices such as keyboard and mouse do not provide an expected way of interaction.

Most of the innovative interfaces such as Microsoft Surface [11] tend to support multi user interaction and are recognized to be augmented reality based products. Due to that reason there has been lot of concern on development on alternative and natural interaction methods to support interaction with such interfaces, while supporting for the existing conventional computing devices. Thus human-computer interaction design is human centered approach where human is given more priority. Also the previous work done by [2] note that “The human, the user, is, after all, the one whom computer systems are designed to assist. The requirements of the users should therefore be our first priority”.

3 Computer Vision and Gesture Recognition

In its most general meaning, a gesture is any physical configuration of the body, whether the person is aware of it or not, whether performed with the entire body or just the facial muscles, whether static in nature or involving a movement. In the computer vision literature, gesture usually refers to a continuous, dynamic motion, whereas a posture is a static configuration.

Computer vision based gesture recognition is a sub domain of HCI and it comprises of a wide range of shapes, motions and texture based variations. And also it includes different gesture recognition methods such as applying Fourier transform ([6]), wavelet transform ([4]) or Principal Component Analysis (PCA) ([16]) on images, Edge orientation histogram, temporal templates ([17]) and oriented rectangular patches ([8]). Thus, it is very important to study on these gesture recognition method differences and select good features to define simple and natural gestures which will be easily adoptable to be used for human computer interaction. Recognition of gestures includes object detection, motion analysis, extraction of features, and machine learning. Besides real time recognition has been a stimulating task in all the time. Efficient recognition of positions can be adopted for an effectively simulation of keyboard events. For example, posture classification refers to the estimation of finger configurations, that is, the ability to distinguish a fist from a flat palm and so on. As described by [9] describes different kinds of gestures from what has become known as Kendon's Gesture Continuum. However practical limitations due to varying luminous conditions and complex backgrounds can exist. Thus, finger tracking and use of non-geometric features such as color and outline are also important for a reliable and strong recognition. There is an extensive body of related computer vision research which could fill many books. Here, author has summarized the major works that could fit the bill for real-time user interface operation through hand gesture recognition in a fairly unconstrained environment. To get an independent overview, the reader is referred to a paper by [5] a survey on “computer vision for interactive computer graphics” and an evaluation of the state of the art by [15]. Three common tasks for computer vision processing are; (1) The detection of the presence of an object in an image. (2) The spatial tracking of a once-acquired object over time. (3) Recognition of one of many object types

3.1 Preprocessing

Preprocessing is the progression of color space conversion, edge detection, morphological operations, noise removal and thresholding. Therefore, it is implemented in almost every vision based algorithms as an entry point to be suitable for the image processing. According to [14] color space conversion, noise removal, edge detection and outlines extraction has to be carried out during the preprocessing stage.

3.2 Detection

As [3] showed in early neuron scientific experiments the human visual system has the amazing ability to detect hands in almost any configuration and situation, and possibly a single “hand neuron” is responsible for recording and signaling such an event. The computer vision researches have not quite yet achieved this goal. However, it is vital that a hand is supposed to function as an input mechanism to the computer is strongly and consistently perceived in front of arbitrary background, for the reason that all further stages and functionalities depends on it. Object detection of artificial objects, such as colored sticks as in [18] can achieve very high detection rates regardless of low false positive rates. According to [20] face detection has attracted a great amount of interest and many methods relying on shape, texture, and/or temporal information have been thoroughly investigated over the years. Author has carried out some researches on finding hands in grey-level images based on their appearance and texture. As assert by [19] "Combining with skin color segmentation, view independent posture recognition can be used to detect hands. Since skin color segmentation has already limited the searching range, hand detection can be very efficient". [12] demonstrated that, lately improved classifiers have succeeded compelling results for view and posture independent hand detection. However, most of the hand detection methods resort to less object-specific approaches and as an alternative employ color information (see, for example [21]), sometimes in combination with location priors (for example [10]), motion flow or background differencing (for example [13]).

3.3 Tracking

Background subtraction is very important for motion analysis and object tracking because of it's a basic function that enables to build statistical model of background. And used for segmenting moving objects for the background. If the detection method is flexible and fast enough to operate at image acquisition frame rate, it can be used for tracking as well. However, tracking hands is extremely difficult since they can move very fast and their appearance can change enormously within a few frames. As [1] asserts that some of the most effective head trackers, for example, use a fixed oval shape model which is fast and appropriate for the inelastic head structure. Similarly, more or flexible hand models work well for a few select hand configurations and relatively static lighting conditions. Since tracking with an inflexible appearance model is not possible for hands in general, most approaches alternative to shape-free color information or background differencing as in the mentioned works by [10], and [13]. Other methods incorporate for example, texture and color information and can then recognize and track a small number of fixed shapes regardless of arbitrary backgrounds (for example, [22]). As per the research work, a particle filtering method is optimized for speed mean shift, and dynamic weights determine the blend of color with motion data. That explains, the faster the object moves, the more weight is given to the motion data, and slower object movements result in the color cue being weighted higher. Some of their performance is surely due to simple, however usually effective dynamical model (of the object velocity), which could add to the suggested solution as well. Object breakdown based on visual flow (for example, normalized graph cuts as proposed by [23]) can produce good results for tracking objects that display a limited amount of twists during global motions and, thus have a fairly unchanging flow ([24])

3.4 Skin Color

Skin color detection is widely used to detect hand configurations and thus it is very important in gesture recognition. Skin color classification is preferred for fast processing and due to its effective response to non-rigid objects such as hands.

Previously absorbed results shows it can be achieved only by skin color properties, for example, by [25] who used it in combination with a neural network to estimate gaze direction. [26] Demonstrate interface quality hand gesture recognition only with color segmentation means. Their method uses an HSV like color space, which is possibly beneficial to skin color identification.

The appearance of skin color differs mostly in intensity while the chrominance remains fairly consistent. Thus, and according to [27], color spaces that separate intensity from chrominance are suitable to skin segmentation when simple threshold-based segmentation is used. However, their results are based on a few images only, while a paper from [28] examined a huge number of images and found an excellent classification performance with a histogram-based method in RGB color space. It appears that very simple threshold methods or other linear filters accomplish better results in HSV space, while more complex methods, particularly learning-based, nonlinear models excel in any color space. [28] Also state that Gaussian mixture models are lower to histogram based approaches. This is true as long as a large enough training set is available. Otherwise, Gaussians can fill in for inadequate training data and achieve better classification results. [29] Showed that object tracking based on color information is possible with a method called CamShift which is based on the mean shift algorithm. These methods dynamically slide a “color window” along the color probability distribution to dynamically parameterize thresholding segmentation. A certain amount of lighting changes can be allocated with. Patches or drops of uniform color have also been used, especially in fairly controlled scenes. According to [30] achieve excellent segmentation with dynamic adaptation of the skin color model based on the observed image.

3.5 Contours Extraction

Contour processing is performed on images typically after performing edge detection or thresholding. Contour extraction is used after canny edge detection algorithm, to detect an inserted object using color cluster feature. In theory, the contour or outline of an object reveals a lot about its shape and orientation. If perfect segmentation is possible, comparison based on curve matching is a feasible approach to object classification. For example as [31] assert that, based on polar coordinates above can be done. One can benefit even more from curve descriptors that are invariant to scale differences and rigid transformations such as those by [32] and Shape Context descriptors. For less-than-perfect conditions however, more powerful 2D methods must be used. Those usually set on finding enough local clues in the image to place a shape model close to where the most likely placed of this shape can be found in the image data. Iterative methods frequently try to minimize an energy defined as images which are not aligned properly (far from an edge). For a hand in top view, these modes could theoretically be the movements of each finger. Statistical models of an object's 3D shape, often called “point clouds,” can also be built (as did, for example, [33]), but they shall not be further measured since their speed performance might drop. According to [34] took a popular approach and had their recognition method learn from extracted hand images instead of from actual photographs. During testing, edge data between the observation and the learned database are compared and 3D hand configurations can be estimated from 2D grey-level edges. According to their paper, matching takes less than a second for an approximate result, but too long for interactive frame rates. [35] assert that; detect hands uniquely in postures regardless of messy backgrounds. The distance between two curves or contours is the mean of the distances between each point on one curve and its closest point on the other curve.

4 Implementation

4.1 Hand Gesture Recognition

Hand gestures can be recognized with various means and varying fidelity. They are not in one particular identification technique, but with various sensing mechanisms. Hand detection for user interfaces must favor reliability over expressiveness: false positives are less tolerable than false negatives. Since detecting hands in arbitrary configurations is a largely unsolved problem in computer vision, the detector for iPanel allows reliable and fast detection of the hand in one particular posture from a particular view direction. Starting the interaction from this initiation pose is particularly important for a hand gesture interface that serves as the sole input modality as it functions as a switch to turn on the interface: without this and instead with an always-on interface, any gesture might inadvertently be interpreted as a command. The output of the detection stage amounts to the extent of the detected hand area in image coordinates. This software system is capable of detecting the human hand in monocular video, tracking its location over time, and recognizing a set of finger configurations (postures). It operates in real time on commodity hardware and its output can thus function as a user interface.

The software system that realizes the vision-based hand gesture recognition and allows for its utilization as a user interface consists of a number of software components that will be described in the following. iPanel main component pronounced “skindetector" is a library and the core gesture recognition module that implements all of the computer vision methods for detection, tracking, and recognition of hand gestures. This module receives the direct video feed from a camera and generated the analyzed gesture results to the main application. This application called WinTalk class library, which handles pipeline initialization and implements convenience functions. In addition to these runtime components, there is also an offline module that implements ANN (Artificial Neural Networking) training for the detection and recognition components.

The core module is a combination of recently developed methods with novel algorithms to achieve real-time performance and robustness. A careful orchestration and automatic parameterization is largely responsible for the high speed performance while multi-modal image cue integration guarantees robustness. Yet, initially an hard-coded values has been used to identify the hand gestures of the author in order to make sure that, development phase has not been misguided with different values. There are three stages: the first stage detects the presence of the hand in series of posture (It is required to have the vision interface always active since hand gestures which are used as mouse movements may be used as commands). Yet, identifying a series of postures could cause different errors in the application due to the misleading of generated events. However, the issue has been addressed by a different mechanism, and to be discussed in the following paragraphs. After this gesture based activation, the second stage serves as an initialization to the third stage, the main tracking- and posture recognition stage. This multi-stage approach makes it possible to take advantage of less general situations at each stage. Exploiting spatial and other constraints that limit the dimensionality and/or extent of the search space achieves better quality and faster processing speed. Author uses this at a number of places: the generic skin color model is adapted to the specifics of the observed user for posture recognition is positioned with fast model free tracking. However, staged systems are more prone to error propagation and failures at each stage. To avoid these pitfalls, every stage makes conservative estimations and uses multiple image cues (grey-level texture and local color information) to increase confidence in the results. "SkinDetector" assists as a library for gesture recognition that can be built into any windows application that demands a hand gesture user interface. However, it does not handle any user display-specific operations such as image acquisition or display. Thus, it requires some programming before it can be used. The final output of the vision system indicates for every frame the 2D location of the hand with the number of fingers if is tracked, or that it has not been detected yet. If the dominant hand's posture is recognized, it is described with a string identifier as a classification into a set of predefined, recognizable hand configurations.

4.2 Hand Detection

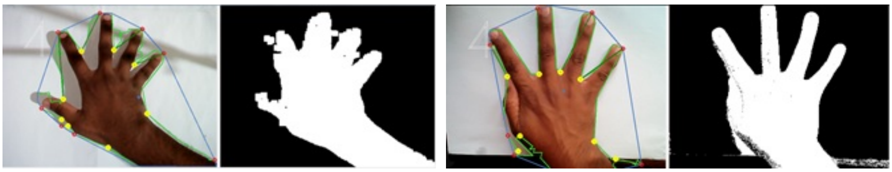

Fig. 1. (1) Actual output of the custom YCrCb based skin detection algorithm. (2) Output - Hsv based skin detection algorithm.

YCrCb / Custom YCrCb Based Skin Detection. Y′ is the luminance component and Cr and Cb are the red-difference and blue-difference chroma components. Y′ (with prime) is distinguished from Y which is luminance, meaning that light intensity is nonlinearly encoded based on gamma corrected RGB primaries. Yet, identifying the skin using YCrCb algorithm is very challenging because of the difficulty of identifying the correct Ycc color range.

Hsv Based Skin Detection. In the HSV color system, the colors of maximum saturation are not necessarily pure. The HSV, an alternate representation of a given RGB color space, and the saturated colors in HSV are in fact the colors bordering the corresponding RGB triangle in the chromaticity diagram. For this reason, the HSV color system has be identified as device dependent, meaning that it is not an absolute colorimetric space, but relative to the gamut of the RGB color space it describes. The third coordinate in HSV has the value or brightness; black has zero brightness. Starting from the hues disk one can imagine the HSV space as a collection of hues circles with varying color value, one on top of the other and of the same size or of sizes diminishing with value. The Fig.1 (2) shows, the identified hand postures using the Hsv skin detection algorithm.

Both algorithms custom YCrCb and Hsv based skin detection can used to identify the user skin of hand gestures. However, most of the researches shows, that the source frame is converted to both Ycc and Hsv color spaces and observed that Hsv color space provides better segmentation in practice over Ycc. Furthermore, it has been tested with different skin colors during various times of the day. The reason was Hsv provides clear separation of luminance and chrominance. Yet, it is more vital to train the algorithms through an ANN approach to recognize any type of skin in order to ensure that the every skin component has been identified in the image stream.

4.3 Recognition

In order to place flock features, initially context hulls are recognized through the identified counters, a centered point on top of the detected skin and a clock wise rotation. The Flock of Features follows small grey-level image artifacts. A weak global constraint on the features' locations is enforced, keeping the features tightly together. Features that are not likely to still be on an area of the hand appearance are relocated to close proximity of the remaining features and on an area with high skin color probability. This technique integrates grey-level texture and dimensionless color cues, resulting in more robustness towards tracking disturbances cause by background artifacts. From the feature locations a small area is determined that is scanned for the key postures that recognition is attempted for. Once prefect detection has been performed events are bind with the fingers.

4.4 Execution

In order to perform key pressing events author has been developed on his own algorithm though out series of researches. The identified method was to capture three different postures of figure movements and analyze them to meet the requirements of performing key press actions. In the algorithm each of the fingertip positions are stored on a collection along with a finger number. When a key press is performed through the gestures it is noticeable that particular fingertip's positions are change through Y axis of the screen while X axis on constant (But X axis could be slightly changed based on the movement). For example, if the initial X and Y position of a fingertip is X = 150, Y = 200; in the event of performing key press tip positions are changed to X = 155, Y = 265 and then again it changes to a position X = 152, Y = 225. [37] Discussed a similar algorithm in there research related to wearable multi-touch interaction. According to their solution three dimensional (X, Y and Z) fingertip detection has been used. However, author insists to use his own algorithm in order to minimize the complexity and to improve effective mechanism to identify the event of key press.

4.5 Finding Rectangles and Identify Characters

Finding rectangles involves finding axis aligned rectangles in binary image. These rectangles will help in separating the area of the image that contains text from the rest of the image. Even though, this is a huge process to carry out Tesseract API, provide all the required processing and identifying algorithms. Therefore, no custom algorithms have been developed in order to analyze images or to identify. Furthermore, to identify language dependent characters Tesseract API required initializing with relevant tesseract data and language dictionaries. Following Fig.4 (1) illustrates characters identified by the module.

Fig. 2. (1) Characters identified by the OCR module. (2) User designed key arrangement and module generated virtual keyboard.

5 Evaluation

5.1 Expert Evaluation

An expert evaluation was conducted on the research to measure the validity and appropriateness of the approaches, methodologies, and models used by the author. The expert evaluation process has been started at the requirement gathering stage in order to understand and evaluate the user requirements. Then a thorough analysis and evaluation has been conducted in the design phase to avoid expensive mistakes, since the design can be altered prior to any major recourses commitment. Therefore, sample design and prototypes have been provided though it is difficult to get an accurate assessment of the experience of interaction. [4] assert that four different approaches that could adopt to expert analysis: (1) cognitive walkthrough, (2) heuristic evaluation, (3) the use of models and (4) use of previous work. The author uses three approaches in order evaluate system properly.

Cognitive walkthrough approach has been adopted and it evaluation is the code walkthrough to check certain characteristics (for example, that coding style is adhered to proper coding standards). The general idea behind the heuristic evaluation is that several evaluators independently critique the system to identify potential usability problems and to understand the severity of the problems. The final approach used to evaluate the iPanel system is “use of previous work”. Expert knowledge on previous experience also has been involved in order to provide the feedback for the system.

Furthermore, their comment regarding research holds a significant impact for the future enhancements on the application. A questioner has been used to gather expert reviews on the iPanel. With the intension of understanding the accuracy of the output, and the suitability of implemented algorithms, along with identifying the independency on user hand gestures; 71% of the participation has been awarded “Excellent” for the accuracy of the hand gesture recognition algorithm, while 28% stated “Good”. Also it is notable to discuss all the experts have been agreed that, Hsv based skin color detection along with structural analysis can provide an effective person independent hand and finger detection.

6 Conclusion

Author has been developed the iPanel, a computer vision system for recognition of hand gestures in real-time and perform key strokes in order to allow real time interaction with a virtual keyboard. Novel and improved vision methods had to be devised to meet the strict demands of user interfaces. Tailoring system and applications for hand motions within a comfort zone that we have established improves user satisfaction and helps optimizing the vision methods. Multiple applications demonstrated such as windows, web, and mobile showed iPanel in action and indicated that it adds to the options of interaction with non-traditional computing environments.

References

1. Birchfield, S.: Elliptical head tracking using intensity gradients and color histograms. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 23-25, pp. 232–237. IEEE, Stanford University (1998)
2. Dionisio, C.R.P., Cesar Jr., R.M.: A project for hand gesture recognition. In: Symposium on Computer Graphics and Image Processing, p. 345. IEEE, Sao Paulo University (2000)
3. Desimone, R., Albright, T.D., Gross, C.G., Bruce, C.: Stimulus-Selective Properties in Inferior Temporal Neurons in the Macaque. Journal of Neuroscience 4(8), 2051–2062 (1984)
4. Dix, A., Finlay, J.E., Abowd, G.D., Beale, R.: Human-Computer Interaction, 3rd edn. Peaeson, New Delhi (2003)
5. Freeman, W.T., Anderson, D.B., Beardsley, P.A., Dodge, C.N., Roth, M., Weissman, C.D., Yerazunis, W.S.: Computer Vision for Interactive Computer Graphics. In: Computer Graphics and Applications. IEEE Computer Graphics and Applications, pp. 42–53 (May-June 1998)
6. Harding, P.R.G., Ellis, T.: Recognition Hand Gesture Using Fourier Descriptors. In: IEEE Pattern Recognition, August 23-26, vol. 3, pp. 286–289. Buckinghamshire Chilterns Univ. Coll., UK (2004)
7. Harper, R., Rodden, T., Rogers, Y., Sellen, A.: Being Human - human-computer interaction in the year 2020. Microsoft Research, Cambridge (2008)
8. Ikizler, N., Duygulu, P.: Human Action Recognition Using Distribution of Oriented Rectangular Patches. In: Elgammal, A., Rosenhahn, B., Klette, R. (eds.) Human Motion 2007. LNCS, vol. 4814, pp. 271–284. Springer, Heidelberg (2007)
9. Kendon, A.: Cross-cultural perspectives in nonverbal communication. In: How Gestures can Become Like Words, pp. 131–141 (1988)

10. Kurata, T., Okuma, T., Kourogi, M., Sakaue, K.: The Hand Mouse: GMM Hand-color Classification and Mean Shift Tracking. In: Second Intl. Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems (July 2001)
11. Microsoft Cooperation: Microsoft Surface. Microsoft (2012), `http://www.microsoft.com/surface/en/us/default.aspx` (accessed October 08, 2012)
12. Ong, E.J., Bowden, R.: A Boosted Classifier Tree for Hand Shape Detection. In: Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition, pp. 889–894. IEEE (2004)
13. Segen, J., Kumar, S.: GestureVR: Vision-Based 3D Hand Interface for Spatial Interaction. In: The Sixth ACM Intl. Multimedia Conference. ACM (September 1998)
14. Shen, W., Wu, L.: A method of billiard objects detection based on Snooker game video. In: Future Computer and Communication (ICFCC), Beijing, China, May 21-24, vol. 2, pp. 251–255. IEEE (2010)
15. Turk, M.: Computer Vision in the Interface. ACM Communications 47(1), 60–67 (2004)
16. Vafadar, M., Behrad, A.: Human Hand Gesture Recognition using image processing for Human-Computer Interaction. In: IEEE Information and Knowledge Technology (2007)
17. Vafadar, M., Behrad, A.: Human Hand Gesture Recognition Using Motion Orientation Histogram for Interaction of Handicapped Persons with Computer. In: IEEE Image and Signal Processing, pp. 378–385 (2008)
18. Wilson, A., Shafer, S.: UI for Intelligent Spaces. In: XWand. ACM (2003)
19. Wu, Y., Huang, T.S.: View-independent recognition of hand postures. In: IEEE Computer Vision and Pattern Recognition. Beckman Inst. for Adv. Sci. & Technol., vol. 2, pp. 88–94. Illinois Univ., Urbana (2000)
20. Yang, M.H., Kriegman, D.J., Ahuja, N.: Detecting Faces in Images: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(1), 34–58 (2002)
21. Zhu, X., Yang, J., Waibel, A.: Segmenting Hands of Arbitrary Color. In: Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition. Interactive Syst. Labs., pp. 446–453. Carnegie Mellon Univ., Pittsburgh (2000)
22. Bretzner, L., Laptev, I., Lindeberg, T.: Hand Gesture Recognition using Multi-Scale Colour Features, Hierarchical Models and Particle Filtering. In: Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition, pp. 423–428. IEEE, Washington, D.C (2002)
23. Shi, J., Malik, J.: Motion segmentation and tracking using normalized cuts. In: Sixth International Conference on Proc. Computer Vision, January 4-7, pp. 1154–1160. IEEE, Berkeley (1998)
24. Quek, F.K.H.: Unencumbered Gestural Interaction. IEEE Multimedia 4(3), 36–47 (1996)
25. Stiefelhagen, R., Yang, J.: Gaze tracking for multimodal human-computer interaction. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1997, Karlsruhe University, April 21-24, vol. 4, pp. 2617–2620. IEEE (1997)
26. Kjeldsen, R., Kender, J.: Finding Skin in Color Images. In: Proceedings of the International Conference on Automatic Face and Gesture Recognition, pp. 312–317 (October 1996)
27. Zarit, B.D., Super, B.J., Quek, F.K.H.: Comparison of Five Color Models in Skin Pixel Classification. In: Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, Illinois University, Chicago, IL, pp. 58–63. IEEE (September 1999)
28. Jones, M.J., Rehg, J.M.: Statistical Color Models with Application to Skin Detection. Int. Journal of Computer Vision 46(1), 81–96 (2002)

29. Comaniciu, D., Ramesh, V., Meer, P.: Real-Time Tracking of Non-Rigid Objects Using Mean Shift. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 142–149. IEEE (2000)
30. Zhu, Q., Cheng, K.T., Wu, C.T., Wu, Y.L.: Adaptive Learning of an Accurate Skin-Color Model. In: Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition, Santa Barbara, CA, USA, May 17-19, pp. 37–42. IEEE (2004)
31. Hamada, Y., Shimada, N., Shirai, Y.: Hand Shape Estimation Using Sequence of Multi-Ocular Images Based on Transition Network. In: VI 2002 (2002)
32. Gdalyahu, Y., Weinshall, D.: Flexible Syntactic Matching of Curves and its Application to Automatic Hierarchical Classification of Silhouettes. IEEE Transactions on Pattern Analysis and Machine Intelligence 21(12), 1312–1328 (1999)
33. Heap, T., Hogg, D.: Towards 3D Hand Tracking Using a Deformable Model. In: Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition, Leeds University, October 14-16, pp. 140–145. IEEE (1996)
34. Athitsos, V., Sclaroff, S.: Estimating 3D Hand Pose from a Cluttered Image. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Boston University, MA, USA, June 18-20, vol. 2, pp. 432–439. IEEE (2003)
35. Thayananthan, A., Stenger, B., Torr, P.H.S., Cipolla, R.: Shape Context and Chamfer Matching in Cluttered Scenes. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Madison, USA, June 18-20, vol. 1, pp. 127–133. IEEE (2003)
36. Xie, J.: Optical Character Recognition Based on Least Square Support Vector Machine. In: Intelligent Information Technology Application, IITA 2009, School of Electronics, Jiangxi University of Finance and Economics, Nanchang, China, November 21-22, vol. 1, pp. 626–629. IEEE (2009)
37. Harrison, C., Benko, H., Wilson, A.D.: OmniTouch: Wearable Multi-touch Interaction Everyware, pp. 16–19. ACM, Santa Barbara (2011)