Abstract

Gesture-based interfaces provide an intuitive and natural way to convey 2D graphical command symbols for interacting with applications and services.

Users can prefer performing gestures “in the air” or through suitable tools, e.g. tablets, depending on the context of usage and the available tools. A collection of device-independent algorithms and tools for analysing gestures would facilitate such a plasticity in the interaction process.

This paper describes how a set of 2D graphical symbols, traced by moving a led pen in front of a single-camera, can be recognised in a gesture recognition system, using the same algorithms employed for recognising these symbols when used on a tablet.

1 Introduction

The main goal in the design of multimodal interfaces is to increase the naturality of human-machine communication process, exploiting typical communicational behaviours of human beings. Gesture- and speech-based modalities are considered among the most promising ways for human-machine communication, as they are most common in human-human communication. For the case of visual concepts (e.g. graphical symbols), these are better expressed through gesture modality, when it is possible to achieve an immediate abstract visualization of temporal, geometrical and spatial characteristics.

Users can perform gestures finalized to command a device using a specific tool (e.g. movement sensors, tablet PC) or simply by tracing signs with their hands captured by some optical device. Depending on several factors, e.g. application environments (e.g. kind of application/service, kind of device) or technical contexts (e.g. indoor, outdoor, presence of light), different interaction supports may be preferred. The type of support, as well as the choice to focus on the hand movements, heavily influence the approaches used in the recognition process and the involved algorithms.

A possible classification of gesture recognition algorithms is based on the way by which they extract movement information from low level data. In particular, algorithms can be classified as tracker based or vision based [3]. To the first class algorithms belong which base the recognition process on haptic devices (e.g. data gloves, body suits) by which to capture the whole gesture informative content, while in the second class we find algorithms that obtain information by analyzing a sequential set of images representing a human-computer interaction scene.

This paper describes a simple gesture recognition process in to the second class, where gesture movements, performed by the user through the management of a led pen (or some other simple gadget coloured with a uniform tonality), is detected by a color tracking algorithm that analyzes a sequential set of images captured by a single-camera. Once the spatial and temporal information of the tracked color has been obtained, a sketch recognition engine is adopted to interpret the related graphical symbol.

The basic idea of the system is that the user can define, through a sketch-based interface (i.e. based on the stroke identification), a set of recognizable gestures. In fact, the core system is based on a sketch recognition algorithm built to face the main aspects in this field multi-domain definition, real-time recognition and tracing style interpretation [1]. For this reason, our approach identifies the trajectory (i.e. spatial and temporal information) of the tracked color as a set of sequential points through which to define a stroke representing an abstraction of the user’s gesture.

In the following, Section 2 presents the system architecture and shows some preliminary experimental results and Section 3 concludes the paper.
2 The System Architecture

The main aspects of the system architecture are discussed with reference to Figure 1.

![Figure 1. System Architecture](image)

The architecture relies on the definition of a set of sketched graphical symbols to be recognised. An editor is available, which allows skilled user to define unistroke symbols. With each symbol, an agent is associated able to recognise instances of the symbol based on an analysis of mathematical and geometrical features extracted from the strokes drawn with the gestures.

The system is structured into three modules. Given a set of predefined symbols, previously defined with a sketch editor, and characterised in terms of strokes, the modules process a set of video frames captured by the camera to extract the points of the fixed color trajectory and identify a traced stroke, to be matched with the strokes defining the symbols.

The first module (Color Tracking) adopts a color tracking technique [2] to detect, within an image, all the regions that have a fixed target color. The algorithm follows, frame by frame, the spatial movement of these regions and produces, for each of them, a bounding box surrounding it. The set of bounding boxes defines the trajectory of the gestures performed by moving the led pen in front to the single-camera.

The second module (S&N, Skeleton and Normalization) receives the set of bounding boxes and reduces each of them to a single spatial point by calculating their barycenters. It also interpolates these points to deal with the possible lack of spatial contiguity due to the speed of the gesture in comparison with the \( \text{fps} \) (frames per second) capacity of the camera. At the end of the process the module will have produced a representation of the user’s gesture as a stroke.

The third module (FcBD, Feature calculation Bid Decision) is an agent-based sketch recognition engine. Given a sketch/gesture library created with the editor, the FcBD module extracts features from the strokes produced by the S&N module and matches them against the features characterising the symbols in the library. Figure 2 shows a test library.

![Figure 2. Basic Test Library](image)

In particular, each agent checks the conformance between the evaluated features and the description of the symbol for which it is responsible, and decides whether the stroke can be recognized as a specific graphical symbol. When a conflict occurs, i.e. two or more agents claim recognition for a same stroke, a mediator agent solves them through some heuristic strategy.

The FcBD engine has been tested, with good results using traditional sketching tools (mouse and tablet). Experiments are now starting with the camera-based gesture-recognition. The first results are promising, but they have been obtained in a controlled indoor environment and with regular, not too fast gestures.

3 Conclusion

We have presented an agent-based sketch recognition engine for definition of 2D graphical symbols and their recognition through gesture color tracking. We need to improve the current color tracking algorithm to obtain independency from both the application environments and technical contexts. Moreover, we are working on a freehand gesture recognition system.

References

