



## What you draw is what you search: the analog gesture

Benoit Rouxel, Franck Poirier, Jean-Yves Antoine, Gilles Coppin

### ► To cite this version:

Benoit Rouxel, Franck Poirier, Jean-Yves Antoine, Gilles Coppin. What you draw is what you search: the analog gesture. HCI International 2014, Jun 2014, Heraklion, Crete, Greece. Springer, Lecture Notes in Computer Science - Volume 8511, 8511, pp 139-147, 2014, Human-Computer Interaction. Advanced Interaction Modalities and Techniques. <hal-01141923>

**HAL Id: hal-01141923**

**<https://hal.archives-ouvertes.fr/hal-01141923>**

Submitted on 14 Apr 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# What you draw is what you search: the analog gesture

Benoit Rouxel (1), Franck Poirier (2), Jean-Yves Antoine (3),  
Gilles Coppin (1)

1 – Lab-STICC, Telecom Bretagne CS 83818, 29238 Brest, France  
benoit.rouxel@telecom-bretagne.eu, gilles.coppin@telecom-  
bretagne.eu

2 – Lab-STICC, Université de Bretagne-Sud 56000 Vannes, France  
franck.poirier@univ-ubs.fr

3 – Université François Rabelais de Tours, LI, 3 place Jean Jaurès, 41000 Blois, France  
jean-yves.antoine@univ-tours.fr

**Abstract.** This paper presents a new type of gesture for identifying spatio-temporal patterns: the analog gesture. Analog gestures can be characterized by some features (speed, acceleration, direction, and angle) which describe the dynamic morphology of the gesture. At first, we detail interactive tasks that should benefit for the use of analog gestures. Then we give a state of the art concerning gesture recognition and investigate the specificity and the main properties of the analog gesture. Then, we propose a review of the surveillance maritime system called Hyperion which uses analog gestures. Finally, we give an example of the use of this type of gesture by the operator. It concerns the interactive detection of ship abnormal trajectories in the context of maritime surveillance.

**Keywords:** Gesture recognition, time-space pattern search, tabletop computing.

## 1 Introduction

Gestural interfaces are increasingly present in our daily lives. Nowadays, many different gestures have been investigated to enrich interaction. It is possible to use 3D-gestures to control characters in video games or 2D-gestures to make a call with a smartphone. Gestures may be associated with different commands; for instance, symbol drawing can be used as a shortcut for calling software functions [1].

The recognized gestures refer to an action, a symbol or an idea, which refer themselves to software functions. However, to the best of our knowledge, there is no attempt in the literature [6,7,8,12] to consider gestures which can directly refer to a time-space pattern of reference. We call time-space pattern (TSP) a series of positions that takes into account space and time simultaneously. We consider positions composed of a location and a timestamp. TSPs are useful in many domains, such as the behavioral analysis of pedestrians in a crowd, optical character recognition, and more generally any study that aims at spatio-temporal clustering and classification. In this

paper, we considered the application domain of maritime surveillance, where TSPs correspond to boat trajectories that take into consideration the boats speed. Schematic examples of such trajectories are given in figure 1. The TSP on the left corresponds to a uniform speed while the right corresponds to a deceleration.



**Fig. 1.** TSP with uniform speed (left) and deceleration (right). All positions are sampled with the same frequency.

Our proposal consists in investigating the potential uses a new gesture type called analog gesture (AG) which is based on the explicit specification of a TSP. An analog gesture is a gesture dedicated to the expression of spatial and temporal features of a trajectory (or by extension a shape). The gesture is called “analog”, because we expect lengths, orientations and speeds to be proportional or representative of the real trajectory, which is expressed through a TSP.

The paper is organized as follows. Section 2 presents some works related to our problematic. To the best of our knowledge, gestures have not been used so far to express of all the features describing a TSP. This is the aim of the analog gesture, which is defined in section 3. We then described the Hyperion platform, a maritime surveillance application where analog gesture is useful. Finally, we focus on the integration of the analog gesture in Hyperion before a final conclusion.

## 2 Related Work

Previous works in gesture recognition mainly focus on path recognition. *\$I Gesture Recognizer* [14] is a 4-step algorithm that recognize a predefined gesture extracted from a finite alphabet of unistroke gestures. The algorithm was later extended (*\$-family*) to enable multistroke gesture recognition [13]. Other algorithms as like in *Octopocus* [2] or the turning angle algorithm [5] use template alphabets to recognize gestures. Though originally applied on images, the turning angle algorithm can be applied on gesture recognition.

Contrary to the aforementioned algorithms, *PaleoSketch* [10] does not use any alphabet. This application improves free hand draws by replacing parts of the sketch with ideal shapes. For example, it replaces a round sketch with a circle and a line sketch with a straight line. When a new shape is drawn, a corner finding algorithm produces a polyline interpretation that closely fits the original shape. After that, each subpart of the computed polyline is analyzed and replaced with the closest simple shape. The simple shape library is composed of several shapes, like line, arc, circle and ellipse. This approach does not use templates to recognize a big shape; the shape is seen as an addition of simple shapes. A very large number of shapes can thus be represented with gestures as long as those shapes can be decomposed in simple

shapes. Nevertheless, all those works only focus on the shape. They don't take into account the speed of the gesture which was used to draw the shape.

Holz and al. [4] takes into account the speed of the gesture in a selection technique they proposed for querying time-series graphs. To select a part of the graph, users sketch over part of the graph, establishing the level of similarity through the speed at which they sketch. Whereas this technique uses a temporal parameter (speed of sketch), it does not allow the expression of free shapes. To search a specific shape, a similar one has to exist.

Rubine's algorithm [11] classifies gestures according to 13 criteria like the length and the angle of the bounding box diagonal, the maximum speed and the duration of the gesture. This algorithm requires an initial training by drawing sample gestures. Even though time is taken into accounts in gesture classification with maximum speed and duration criteria, those two time parameters are too few to express acceleration in a TSP for example. In addition, the template alphabet required by this technique does not allow the expression of all spatial characteristics of a TSP.

While some works focus on the spatial dimension, others use template alphabets which reduce the number of recognized path to the size of the alphabet.

### 3 The analog gesture

The analog gesture is a gesture dedicated to the expression of spatial and temporal characteristics of a trajectory. This gesture taken as a whole is devoted to be decomposed into a series of segments. Spatial characteristics will mostly correspond to the lengths of the segments as well as their orientations; while temporal the features are expressed via the speed or acceleration within the segment.

**Table 1.** Parameters used to characterize a trajectory.

Dimension	Parameter	Recognized feature	
Space	Number of dimension	2D, 3D	
	Shape of the drawing	Angle	Each 15°-interval
		Direction	Continuous
		Spatial inking	Previously defined area
		Path	Sequence of remarkable object in the environment crossed by the drawing
		Orientation	A direction or the opposite direction
Time	Speed	Zero, slow, medium, fast	
	Acceleration	Strong deceleration, deceleration, acceleration, strong acceleration	
Force	Variation of pressure	Same pressure, increasing pressure, decreasing pressure	

The speed and variation of pressure are important when we produce the gesture. Table 1 presents the three dimensions of the gesture realization were used to charac-

terize the gesture and therefore the intended TSP. These three dimensions are space, time and force. For each dimension, one or more parameters are recognized.

For each parameter, the assigned value is either continuous or discrete according to human abilities. Since human can draw a direction with a fairly good accuracy, therefore this is a continuous parameter. On the contrary, people are unable to draw a TSP with an accurate speed, so this parameter can take only few values (zero, slow, normal and fast), and therefore is a discrete parameter.

Analog gestures are multi-touch gestures whenever the objective is to express relative evolutions of multiples trajectories. For instance, in the maritime surveillance domain, if we want to indicate that two vessels are sailing close together on near parallel courses (this situation occurs in boarding situation), we have to use two fingers.

Analog gestures allow in one hit, to express simultaneously various parameters of a segment (length, orientation, speed) as well as complex objects composed of chained segments.

To prove the utility of this type of gesture and how it works in real situation, we will show how we integrated AG in an effective application of maritime surveillance called Hyperion. In the two next parts, we will present the Hyperion platform and after that, we will expose how we use the gesture in this platform.

## **4 Hyperion platform**

VTS are control centers from which the maritime traffic is monitored. They aim at improving the safety and the organization of the traffic and at protecting the environment. The VTS controllers deal with many different types of information at the same time (AIS, radar, weather ...). This considerable amount of raw information involves a heavy cognitive load which reduces the efficiency of the operator.

We propose to develop a domain-specific maritime surveillance system (Hyperion) to reduce cognitive load through a process of computer-aided decision-making. Hyperion is an application dedicated to help vessel traffic service (VTS) controllers. It is developed in Java on Diamond Touch DT107 a touch table.

The main aim of Hyperion platform is to highlight abnormal behaviors of moving vessels. The abnormal behaviors are defined by rules. These rules can contain static properties, a behavior (trajectory) and an anchorage (restriction of a rule to a specific area). Since there rules are strongly related to TSP patterns of sailing behaviour, such behaviours are defined by the controller (expert) using AG.

In order to detect abnormal behaviors of moving vessels in a maritime area, we propose to combine a bottom-up and a top-down approach. The analysis of how surveillance operators work [9] has shown that they were more looking for abnormal trajectories than checking normal ones. This is why we propose in a rule based expert system devoted to the maritime traffic analysis, to focus on the detection of abnormal behaviors.

The top-down approach allows the operator to define a rule characterizing what he/she considers an abnormal situation in a given area. These rules work like a filtering function. Any vessel matching the rules is highlighted (fig. 2).

The bottom-up approach restricts the identification of abnormal behaviors to predefined but well established rules. For example, if a vessel breaks a rule of navigation, it must be reported to the operator. This type of detection is robust in trivial situations. Without these predefined rules, the operator would have to define a larger number of rules for simple situations. Subsequently, he would not be focusing on the detection of abnormal behaviors.

Hyperion works on 2 main modes: an operational mode and a rule edition mode. When the application starts, the operational mode is on. In this mode, a ship breaking a rule is highlighted, and the user has to check the alert report and possibly report a false positive recognition. In rule edition mode, the user can create, search and delete rules, and apply them on map elements (vessels, harbors and areas).

#### 4.1 Operational mode

Figure 2 shows the interface in operational mode: a map represents the current situation. Vessels, harbors and areas appear on the map while alert reports are displayed on right of the map. Rules can be applied on every map elements. Moreover, the areas can be created, modified and deleted by the operator.

When an alert occurs, an alert box which give an overview of the alert goes down from top to bottom right of the screen while, an animation catches the operator attention on the boat triggering the alert. Until alert is treated, the alert box stays in the stack of alerts. Alert box presents the icon of the rule, the time since the alert, the boat name, the rule name, and a number and a color that corresponds to the rule priority. If a vessel breaks a rule, it takes the color of the rule priority or a color corresponding to the sum of broken rules priorities, if more than one rule is broken.



Fig. 2. Hyperion GUI in operational mode.

## 4.2 Rule edition mode

To add a new rule on map elements, the user has to tap and hold on an element. A circular menu appears and the operator can choose the create rule item. If the tap and hold gesture is performed directly on the map (not on a map element) the created rule will apply on the entire area monitored by the VTS. The user can also apply an existing rule on a map element. In this case, the user has to select an existing rule and apply it on a map element.

The set of applied rules can be modified in rule edition mode. In Hyperion platform, a rule is composed of 5 properties: a rule name, a priority, an anchorage, a dynamic behavior (trajectory) and a set of static properties like the boat name or its size.

Figure 3 shows Hyperion GUI when the user is creating a new rule. In this situation, the map and its elements are displayed in the background and two new boxes appear in the bottom left corner. The first one is a detailed view of the currently rule edited and the second one allows the testing of the rule.

In the rule box, each part of the rule is visible. First, the text area in the top left corner allows the showing and editing of the rule name. First, a default name “rule n°--” is given to the rule. Below, the slider corresponds to the rule priority that allows the user to order alert treatment by setting the colors and the numbers to the Vigipirate code (a French alert state). The anchorage area shows the “anchorage” property of the rule. The box in the middle (“behavior”) shows the abnormal trajectory expressed by the rule. These two properties are expressed using analog gestures performed directly on the map. Finally, the last box shows the static rule properties. For example, if we want to express a rule forbidding a military ship to enter an area, we have to add the military property. The static property set can be accessed by performing an up swipe on the property box.



Fig. 3. Create a new rule in Hyperion platform.

The second box concerns rule testing. At the top, the button “test” is used to test rule on the last 24 hours or less according to the recorded history. When a rule is tested a text area under test button displays the number of alerts that would have been raised over the recorded period, as well as the number of false alarms (boats triggering an alarm but considered as normal by the operator). Finally, the circle allows user to replay the recorded history to understand better the alerts triggered by his rule. To play history, it is possible to tap on the play symbol or to move the slider around the symbol to go backwards or forwards.

Figure 4 shows the research of rules, where the user is looking for rules expressing a given behaviour: a list appears above the rule view. It contains every rule in the system, which corresponds to the research. To apply an existing rule on a map element, user just has to drag and drop rule from the list, to the element. In this case, the rule is cloned and the “anchorage” property of the cloned rule is replaced with the new map element.

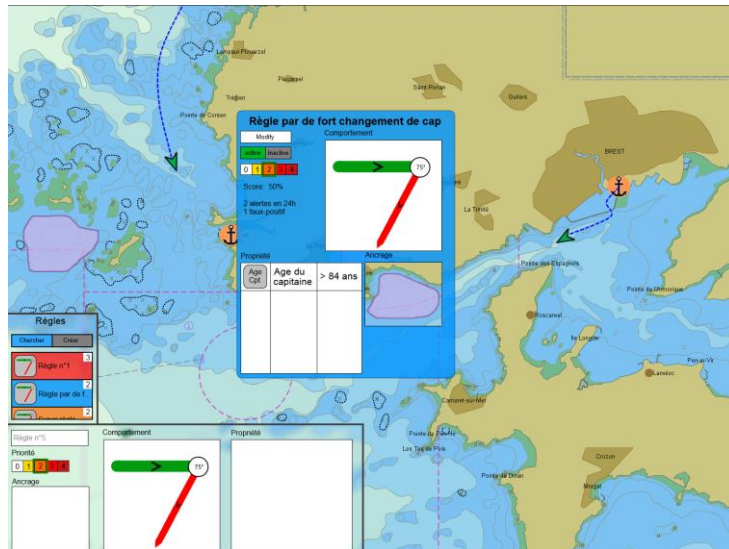


Fig. 4. Rule selection in Hyperion GUI.

This section aimed at the description of the Hyperion system, and how it uses rules: next section focusses more on the use of AG in the platform.

## 5 Use of analog gesture for vessel trajectory appointment

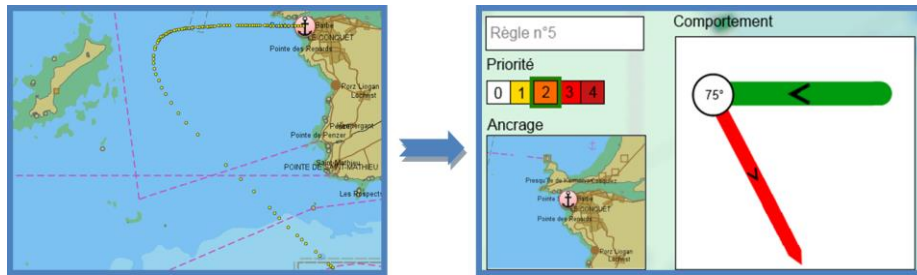
Let us consider a simplified example to describe how AGs are used in Hyperion: suppose the controller wants to describe a vessel trajectory starting from a harbor, suddenly turning to the southeast while accelerating.



Without AG, operator should to select the harbor to indicate the anchorage property of the rule. To express the abnormal trajectory, the operator has to learn the rule syntax and write it in a text editor like in [3].

With AG, the surveillance controller has just to perform a gesture from the point on the map indicating the harbor, going away from the port at normal speed and turning abruptly and rapidly to the bottom right corner of the touch table screen (figure 5a). In a same gesture and without learning process from the operator, “anchorage” and “behavior” properties are added to the edited rule.

When the gesture is completed, a feedback appears on rule view (figure 5b). This feedback allows the user to see what is understood by the system. If the operator does not agree with the recognized TSP, he/she can cancel his/her action by performing another gesture.



**Fig. 5.** a- Example of real gesture (left) and b- its corresponding feedback, with the same sampling frequency.

Analog gesture is limited to maritime surveillance domain. In aerial monitoring domain, it would be possible to use this type of gesture to watch the air traffic. It would be possible to add AG to easily plan the itinerary of an unmanned vehicle like UAV or unmanned car.

## 6 Conclusion

We saw that in some situation, we need to refer to TSP global features. Therefore, we propose the concept of analogical gesture which allows people to directly match to the TSP of reference in the system. Finally, we present the Hyperion platform, our maritime surveillance application which uses the AG.

The analog gesture recognizer is developed. Now, we have to make some experiments to know if it is a real benefit for an operator to express the main characteristics of a trajectory via a gesture, in terms of precision and time to realize this task.

## 7 References

1. Appert, C. and Zhai, S. Using strokes as command shortcuts: cognitive benefits and toolkit support. *CHI 2009*, (2009), 2289–2298.

2. Bau, O. and Mackay, W.E. OctoPocus: a dynamic guide for learning gesture-based command sets. *Proceedings of the 21st annual ACM symposium on User interface software and technology*, ACM (2008), 37–46.
3. Cetin, F.T., Yilmaz, B., Kabak, Y., et al. Increasing Maritime Situational Awareness with Interoperating Distributed Information Sources. *18th Interantional Command and Control Research and Technology Symposium*, (2013), 9–22.
4. Holz, C. and Feiner, S. Relaxed selection techniques for querying time-series graphs. *Proceedings of the 22nd annual ACM symposium on User interface software and technology - UIST '09*, ACM Press (2009), 213.
5. Iannizzotto, G. and Vita, L. A multiscale turning angle representation of object shapes for image retrieval. *Visual Information and Information Systems*, (1999), 609–616.
6. Karam, M. A taxonomy of gestures in human computer interactions. (2005), 1–45.
7. McNeill, D. *Gesture & Thought*. University of Chicago Press, 2005.
8. McNeill, D. Gesture: A Psycholinguistic Approach. *The Encyclopedia of Language and Linguistics*, (2006), 1–15.
9. Nilsson, M., Laere, J. Van, and Ziemke, T. Extracting rules from expert operators to support situation awareness in maritime surveillance. *Fusion, 2008 11th*, (2008), 908–915.
10. Paulson, B. and Hammond, T. PaleoSketch: accurate primitive sketch recognition and beautification. ... *of the 13th international conference on ...*, (2008), 1–10.
11. Rubine, D. Specifying gestures by example. *ACM SIGGRAPH Computer Graphics* 25, 4 (1991), 329–337.
12. Scoditti, A. Gestural interaction techniques for handheld devices combining accelerometers and multipoint touch screens. *Sciences-New York*, 2012.
13. Vatavu, R., Anthony, L., and Wobbrock, J. Gestures as point clouds: a \$ P recognizer for user interface prototypes. *Proceedings of the 14th ACM ...*, (2012), 273–280.
14. Wobbrock, J.O., Wilson, A.D., and Li, Y. Gestures without libraries, toolkits or training: a \$1 recognizer for user interface prototypes. *Proceedings of the 20th annual ACM symposium on User interface software and technology*, ACM (2007), 159–168.