Multitouch Sketch Based Modeling

Leandro M. Cruz*, Luiz P. Velho†, Luiz A. Rivera‡
*Universidade Candido Mendes
Campos dos Goytacazes - RJ - Brasil
lcruz@impa.br
†Instituto de Matematica Pura e Aplicada
Rio de Janeiro - RJ - Brasil
lvelho@impa.br
‡Universidade Estadual do Norte Fluminense
Campos dos Goytacazes - RJ - Brasil
rivera@uenf.br

***Abstract*—This work will show an approach for multitouch sketch-based modeling system development. Our system uses a natural interface as opposed to classical window-based systems. Natural interfaces are another paradigm in man-machine interaction. The use of the multiple touches lets to include functionality in this system, that in windows-based systems, are more complicated to implement and is not intuitive for the users. Some of this new possibilities have been attacked in this work. We will show the possibility of the creation and manipulation of the shape of a curve based on multiple touches. For this purpose, we will show how to transform a sketch, done by the user through a touch, in a b-spline curve. We can manipulate the shape of the curve, based on the change of the position of a set of points with fingers. An important element to do interaction in this systems is gesture. The user's gestures are interpreted to sketch and deform curves or to transform the coordinate systems. Therefore, it is necessary to recognize a touch and associate it with a gesture.**

***Keywords*-Sketch-based Modeling; Natural Interface; Multi-touch System; B-spline curves.**

I. Introduction

This work will approach the aspects of the development of a graphic system to sketch-based modeling using a natural interface to man-machine interaction. We fit a curve based on a sampling done in a sketch drawn by the user. The sketch is done on a touch sensible table.

A typical interaction between man and computer is done by keyboards and mouse, on windows based system. This paradigm is called WIMP (Window, Icon, Menu, Pointer). These systems are based on selecting operations from menus and floating pallettes, entering parameters in dialog boxes. A natural interface is a mode which the user interacts with a system through gestures, natural in its life. Currently there are many types of natural interface, as gesture based systems. We work in a specific platform of gesture based systems: the multitouch systems.

This work shows a method to fit a curve based on a sketch in a multitouch system. The user interacts with the computer by touch, gestures and also by tangible objects called fiducials. Touch based systems are common currently, such as touchscreen monitor, tablet, iphone, etc. In portable devices, touch interfaces are necessary. For example, the iphone mobile has many options of tools that are impossible to implement in another cellphone. Gestures is one of the most important elements to interact with any system. They can be used to generate event, create input and manipulate data. For this, it is necessary to define a set of gestures and associate them with functionalities.

There are many system of sketch-based modeling based on WIMP paradigm. But this is not always intuitive interface for man-machine interaction. Mouse and keyboards are more difficult in activities like drawing, playing music, playing games, handling graphics objects, etc. Natural interface systems are another important interaction paradigm that are better used in cited cases. These systems are more than a hardware able to capture touches, voices, moves, etc. The focus on software development has to be intuitive and natural.

Because of this, developing this software is different. So, it is necessary to pay attention to details and not imitate behaviors commonly done in softwares which use a window system, when it can cause a loss of naturalness in the use of the system. Some aspects, that are different in this type of gesture based system, will be shown in this article.

The system introduced in this article approaches some aspects of curve modeling, as CAD systems. However, our system was developed with a natural interface, enabling an interaction with our system more intuitive than in classical applications. In this article, first it will be shown some architecture aspects of this system, and after it will be shown a method to create and manipulate curves, based in touches.

To model an object with a sketch is a powerful tool for designs. In many situations, it is useful to make a first sketch of an object. With this drawing, it is possible to convey the essence of an object. Depending of the artist skill, it is possible to get a drawing with wealth of details only with sketch. From a sketch, created by the user with fingers, the curve fitting is done through a sampling in this trace. So, with this set of points is fitted by a b-spline curve. After modeling, the shape of this curve can be modified.

Commonly in systems, that uses curves as b-splines, the modeling is done determining a set of points that are approximated to generate the curve. After drawing, is possible to deform it changing the position, or some other feature, of control points. Another approach, more intuitive, is to deform a curve when the user chooses a point and move it. Our approach to modeling is to combine these two techniques to unite the power of to draw a curve, of an intuitive mode, and after to deform properly pieces of the curve. In two process the action is done based on gestures. Because of this, it is possible to draw two curves and to deform several points of a curve simultaneously, as shown in Figure 1. This approach is because we believe that it is a natural form for modeling curves.

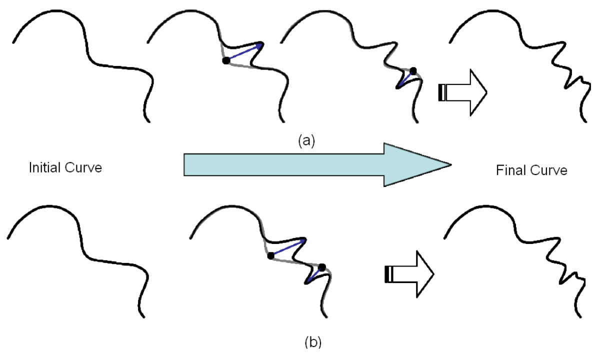

Figure 1. (a) Manipulation of one point; (b) Manipulation of several points.

This work is organized in following sections: Multitouch system, that shows the architecture of the system and of the application developed in this work, approaching the links between gesture recognitions, curves modeling and manipulations; a section showing modeling curve; and another section approach Curve Manipulation.

II. Related Works and Contributions

Currently some works have used resource of sketch to draw a curve, in a interface sensible to touch, as base to generation of a 3D objects. We may cite [1], [2], [3]. Three related works about gesture based systems are [4], [5], [6].

In this work, we were focused on the modeling and manipulation of curves, because we believe that, from a good tool for generation of curves, it is possible improve the obtaining of volumetric objects. Moreover, with the advent of the touch sensitive interfaces, we intend explore the user capability of to do simultaneous traces to generate and manipulate curves. We believe that it is more intuitive, and therefore more easy to use and to achieve better and faster results.

The most direct way to trace a curve is through a draw. This approach has some difficulties. Or the system save the draw done by the user as a image (then, to modify the geometry of the curve is necessary to delete a piece of what was done and redo), or to do a fitting of the sketch for some kind of curve, more convenient (commonly some kind of spline curve, that will be show later). Thus, although a sketch is good for the user to draw a curve, is bad for manipulation of the shape of one. Furthermore, in many curve modeling systems, to draw a curve, is necessary the user defines the position of a set of points (called control points), that the curve obtained will be an approximation of these.

An approach more intuitive is the showed in [7], [8]. In this, the user does a sketch that is sampled and then approximated by a spline. However, the deformation has still been done through to change the position of control points. Another proposal, more intuitive to manipulate the shape of curve, is showed in [9]. In this, the user freely choses a point in a curve and conveniently manipulates the geometry of this one.

In a multitouch sketch-based system, one of the most important elements to data input and manipulation are gestures. In these systems, it is necessary to define a set of gestures that will be used. For each, should be created a method to recognize a trace done in interface sensible to touch, and links it to the gesture. A related work is [10].

III. Multitouch System

This work covers the construction and deformation of curves. In many models of building of three-dimensional objects (which boundary is modeled by surfaces of dimension two) the primitives are based in curves. There are also some deformation of surfaces systems based on curves on the boundary of object.

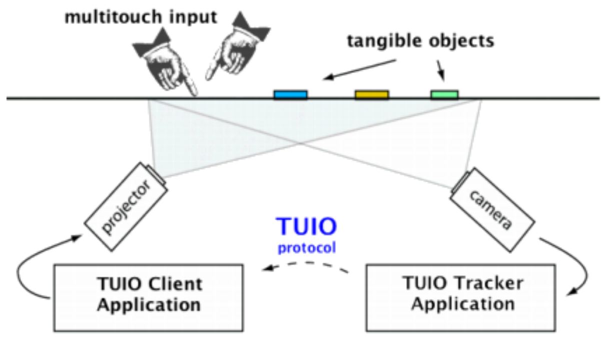

Figure 2. Architecture of System

The general multitouch systems architecture is shown in Figure 2. In the system developed by us the TUIO Tracker is the ReactiVision [11], a computer vision framework, to capture information from the iTable, as Figure 3, a multi-touch table constructed on IMPA (Instituto de Matematica

Pura e Aplicada), based on the ReacTable [12]; and the TUIO Client is the application developed by us using the pyMt (a toolkit to develop a multitouch system with Python Programming Language). The communication between the ReactiVision and de PyMt is done by TUIO Protocol, a TCP protocol. TUIO is an open framework that defines a common protocol and API for tangible multitouch surfaces. The TUIO protocol allows the transmission of an abstract description of interactive surfaces, including touch events and tangible object states. When a event is created in iTable, the ReactVision sends messages that are handled by the application developed by us. The pyMt listens a specific port waiting for messages, sent by ReactiVision, and generates some events. This event generation occurs when:

- The user places a finger on the iTable (the application receives the initial position of the touch).
- The finger is moved (the application receives a set of points that make up a polygon that represents the path of the touch).
- When the user take fingers off the iTable (the application receives the last position of touch)
- When the user inserts a fiducial on the iTable (the application receives the position and the angle,for the guidance of the table, of fiducial)
- When the user move or rotate the fiducial (the application receives the new position and angle of fiducial).
- When the fiducial is removed of table (the application receives the last position of the fiducial).

These are basic events, base to generate other events. These new events can be associated with the fiducial or the gestures. The fiducials are recognized by ReactiVision. The application links the fiducial to an event associated. But gestures is more complex. Initially it is necessary to define a set of pattern to gestures and, for each, create a method to recognize the respective touch.

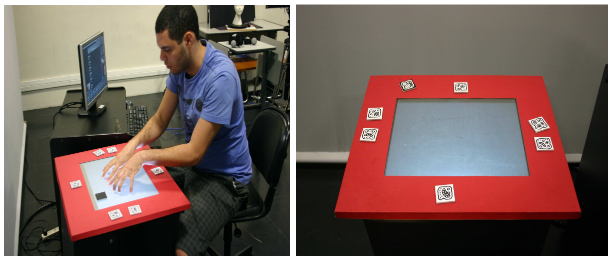

Figure 3. iTable

In this work, we define two approaches for gestures recognition in natural systems. The first one is, when the touch is completed, it is associated to a gesture (we call as Complete Gestures). Other one is: the gesture (associated to a touch) manipulates, interactively, a data in systems. We consider the first approach to generate events such selection, deletion, inclusion, division, union, painting, etc. And the second one, alters the coordinated systems of the drawing (we call as Modifiers Gestures). The difference of context can be doing, for example, based on states, or using fiducials, or in a structure of hypotheses whenever each category of trace occurs.

In this work we will classify Complete Gestures into three subcategories: simple, parallel and linked. Simple Gestures are based in just one touch; Parallel Gestures occurs when two or more touches are done simultaneously; and, Linked Gestures are a sequence of Simple Gestures (one after the other). Simple and Parallel Gestures have been sub-classified as directional, opened shapes and closed shapes.

Directional Gestures are rectilineal touches (almost rectilinear) which can be classified based on some directions of Compass Rose as east, northeast, north, northwest, west, southwest, south and southeast (where north and east are the positive direction of axis x and y of coordinates systems of touches interfaces, respectively). From a sampling of the touch, for each point is calculated the ”derivative” of the curve and it is associated to an angle, defining a touch angle histogram.

Others touch shapes are analyzed similarly. For example, the angles histogram of an Opened Shape in wave shape has a peak around of 90 and 270 degrees. A Closed Shape (when the initial point is around of the last), can be classified by a circle, or rectangles or triangle, etc. In case of circle, the histogram of angle is almost uniformly distributed. In case of rectangles, the accumulation of the angles are close to those angles of 0, 90, 180 and 270 degrees. Similar approaches can be used to define other gestures pattern. A same discuss is showed in [10].

Modifiers Gestures can be used to do operation, in system coordinates of scene, as scale, rotation and translation; or to manipulate the shape of element drawn. The operations of coordinates system transform are based, in our system, in two touches. In each interaction, we have the vector of difference of position of the two fingers. So in the next iteraction, it is possible to verify the translation given by the difference of the current and initial vector between the postion of the fingers $(t1, t2)$; the rotation given by the angle θ between these vectors; and the scale given by ratio s between the size of this vectors.

$$T\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s(xcos\theta - ysen\theta + t_1) \\ s(xsen\theta + ycos\theta + t_2) \end{bmatrix}$$

The gestures recognition is the central module of the application architecture. It defines if the touch will create a curve (so it calls the modeling module), it will manipulate a created curve (so it calls the deform module) or it will transform the coordinate system (so it calls the transform module). These elements compose the architecture of the application shown in figure (4).

As stated, in our application, to draw a curve, the user do

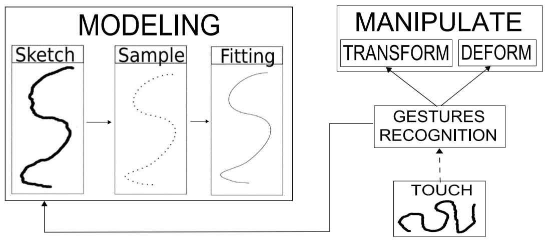

Figure 4. Architecture of Application

a sketch; this is sampled; and these points are fitting by a b-spline curve. This is done in the modeling module. The procedure to fit a set of control points will be shown in the Curve Modeling section.

There are several sketch based modeling systems, including systems of interaction by touch. But in my best knowledge, there are not a system that uses manipulation form of the curve with several restriction in several points. This procedure is good in systems to sketch based modeling in a multitouch system, because takes the possibility of use several fingers to draw and manipulate the curve. With a mouse, it is more difficult to manipulate several points at once. The theory for it will be shown in Curve Manipulation section, and the routine for it, is done, in our application, on deform module.

IV. Curve Modeling

In this article, a curve is created based on a sketch done by the user. We sample the sketch and interpreted the result as control points of a b-spline curve. Based on these points we use b-spline fitting to aproximate the sketch.

Splines are piecewise polynomial curves generated by interpolating a set of points. In 1983, Pavlidis [13] used mechanical elasticity theory to show that cubic splines are piecewise equivalent to cubic polynomial curve. Hence, cubic splines are C^2 continuous. Two spline curves commonly used for modeling curves are Bèzier and B-Spline. These primitives were widely studied. Curve fitting was studied by Reinsch [14] and Grossman [15]. Yamagushi [16], Wu et all [17] and Chong [18] described b-spline curves; Boehm et al [19] and Davis [20] described how to use bèzier curves to modeling curves. Some works that talk about curve fitting using Bèzier curves is Schneider [7] and Frisken [8].

Casteljau and Bèzier developed a method to describe graphic objects such as cars, engines, airplanes, ship, etc. The method created by them was used in CAD systems in Renault (in 1959) and in Citroën (in 1962). In this article, the curve fitting method used was the b-spline. The term spline comes from drafting jargon. It means drawing a smooth curve through a set of points.

The B-Spline theory was suggested by Schoenberg, in 1946. A recursive definition, used by numerical methods, was independently shown by M. Cox, in 1971, and by C. Boor, in 1972; W. Gordon and W. Riesefeld, in 1973, applied this base for generation of curve. A better discuss of history, and of the theory about b-spline is shown in [21], [22].

A b-spline curve is a piecewise polynomial, of degree $k-1$ generated by $c(t) = \sum_{0\leq i\leq n} P_i N_i^k(t)$, where $P = [P_0, P_1, ..., P_n]$ is the vector of control points and $N_i^k(t)$ is the b-spline function, of order k and degree $k-1$, defined by Cox-Boor Algorithm:

$$N_i^1(t) = \begin{cases} 1, x_i \leq t \leq x_{i+1} \\ 0, otherwise \end{cases}$$

$$N_i^k(t) = \frac{(t-x_i)\, N_i^{k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k}-t)\, N_{i+1}^{k-1}(t)}{x_{i+k} - x_{i+1}}$$

The vector $X = [x_0, x_1, ..., x_{n+k}]$ is called Vector of Nodes, sucessive nodes satisfying $x_i \leq x_{i+1}$ and the parameter t varies from t_{min} to t_{max}.

It is common create b-spline curves using a periodical Vector of Nodes, ie, the variation of the nodes is constant. In this case, it is called as periodic b-spline curve. In the software of this project, we use cubic and periodic b-spline curves.

The Cox-Boor Algorithm, for periodic b-spline curves, can be implemented through multiplication of a matrix, in each segment. For example, the $i-th$ segment of cubic and periodic b-spline curve is generated by:

$$c(t) = \begin{bmatrix} N_0^4(t) & N_1^4(t) & N_2^4(t) & N_3^4(t) \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \quad (1)$$

for $t \in [x_i, x_{i+1}]$. In this case, the matrix is:

$$N(t) = \begin{bmatrix} N_0^4(t) & N_1^4(t) & N_2^4(t) & N_3^4(t) \end{bmatrix} \quad (2)$$

can be decomposed as:

$$N(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \quad (3)$$

To model a curve, just define the position of a set of control points and interpolate them with a b-spline. Although, the curve generated usually has not a desired shape. In this case, it is necessary to manipulate the geometry of this object. This theme will be approached in next sections.

V. Manipulation of Curves

In a uniform b-splines curve (the degree do not change in all curve), in order to modify its shape, it is sufficient just modify the control points position. However, this is not intuitive. Because of this, a better technique is to choose a point in the curve and set a new position for it. Then

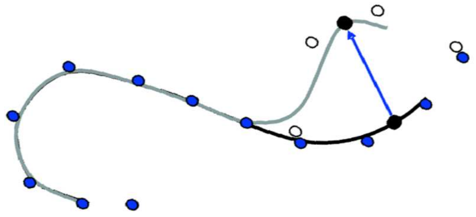

Figure 5. Free form manipulation

just calculate the new positions of the control points of the segment that contains the chosen point. This is an inverse problem, based in equation (1). This discuss will be better in subsections of this section.

We will consider, in this section, a curve of degree k. Although the effect of changing the position of one control point is local, this change modifies $2k$ segments. Then, when to modify $k+1$ control points, to change the position of one point in a segment of a b-spline curve of degree k, will be modified $2k+1$ successive segments (k segments to left, the segment that contain the point modified and more k segments to right). A possible solution to control the variation of the curve is restrict features of several points. This will be discussed in subsection Manipulation of several points. In this article, we quickly show the result defined by [9] about Manipulation of One Point, but we will extend the results to the case of manipulation of the position of several points.

A. Manipulation of One Point

In B-spline cubic and periodic curves this manipulation can be done recalculating the position of the control points of the segment that contains the chosen point (p). For this, it is necessary to choose the parameter $\tilde{t}$, where $p = c(\tilde{t})$, of the point. The variation of the position of each control points of the segment containing p is obtained based on the variation of this point:

$$\begin{bmatrix}\Delta P_0\\ \Delta P_1\\ \Delta P_2\\ \Delta P_3\end{bmatrix} = \frac{\Delta c(\tilde{t})}{\sum_{i=0}^{4} N_i^4(\tilde{t})^2}\begin{bmatrix}N_0^4(\tilde{t})\\ N_1^4(\tilde{t})\\ N_2^4(\tilde{t})\\ N_3^4(\tilde{t})\end{bmatrix} \quad (4)$$

The manipulation done in last paragraph is known as *Free Direct Manipulation*. The Figure 5. The equation (4) derives of the equation (1) by method of Minimum-Length Method (MLM), done in [9]. This equation can be represented in matrix form as:

$$[\Delta P] = N(\tilde{t})^T(N(\tilde{t})N(\tilde{t})^T)^{-1}[\Delta c(\tilde{t})] \quad (5)$$

Other types of manipulations can be obtained with restriction in the first and second derivatives, besides the position in curve. We will refer to the variation of position as $\Delta c(\tilde{t})^{(0)}$, the variation of first derivative as $\Delta c(\tilde{t})^{(1)}$ and the variation

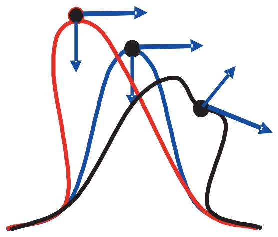

Figure 6. Manipulation of constraints: The center curve is the original curve; the left curve is the manipulation of position keeping the first derivative; the right curve is the manipulation of position and second derivative

of second derivative as $\Delta c(\tilde{t})^{(2)}$. We can vary an attribute keeping invariant others, as shown in Figure 6.

To manipulate features relationship with first and the second derivative, of a b-spline curve, it is necessary to take the derivatives of $N(t)$ in equation (1). So the result, in matrix form, similar equation (5), is:

$$\Delta P = [N(\tilde{t})^{(j)}]^T\left(N(\tilde{t})^{(j)}(N(\tilde{t})^{(j)})^T\right)^{-1}[\Delta c(\tilde{t})^{(j)}] \quad (6)$$

For $j = 0$ equation (6) is equal to equation (1).

Recalculating the position of control points by equation (6), for any value of j, we do not guarantee nothing about features relationship to other values of j. For example, changing the second derivative vector ($j = 2$) of one point, it can modify its tangent ($j = 1$) and / or its position ($j = 0$). In this way, it is possible to modify a curve with more than one restriction. For example, the next equation, similar as equation (1) shows how to modify the vector of second derivatives keeping its position and tangent.

$$\begin{bmatrix}0\\ 0\\ c(\tilde{t})^{(2)}\end{bmatrix} = \begin{bmatrix}N(\tilde{t})^{(0)})\\ N(\tilde{t})^{(1)})\\ N(\tilde{t})^{(2)})\end{bmatrix}[P]$$

So, varying $c(\tilde{t})^{(2)}$, and calling $B = N(\tilde{t})N(\tilde{t})^T$ this equation is:

$$\Delta P = det(B)^{-1}N(\tilde{t})^T adj(B)\begin{bmatrix}0\\ 0\\ c(\tilde{t})^{(2)}\end{bmatrix} \quad (7)$$

Other combination of variations are analogous. In any case of the constraint (refers position, first and second derivatives) onto a point the matrix N is LI (Linearly Independent). So the MLM ensures the solution to equation (5).

B. Manipulation of Several Points

Similarly to have several restriction into a point simultaneously, it is possible to impose one restriction for several points simultaneously. The manipulated points do not need to be in unique segment. Taken two points, these can be classified as local, adjacent and nonadjacent, in respect to their segments, using the concept of *distance*. We will say that the distance of two points is the quantity of segments

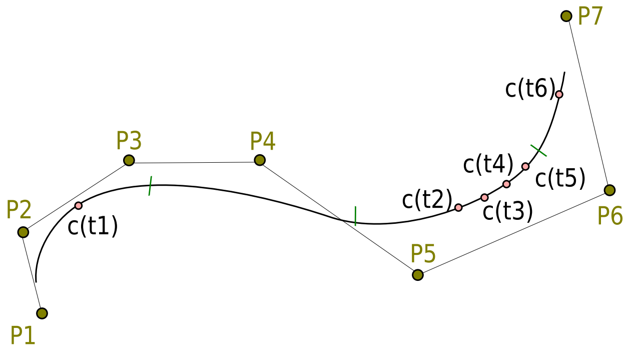

Figure 7. Manipulation of several points: $\{Pi)\}_{i=1,\ldots,7}$ are control points and $\{c(tj)\}_{j=1,\ldots,6}$ are points chosen in a curve. The green trace are delimiters of segments

between they and we will denoted by $distance(P_1, P_2)$. Another concept that we will introduce is: *order*. We will say that the set of points $c(t_1), \ldots, c(t_n)$ is ordered if: taken a injector parameterization $\varphi : I \longrightarrow \mathbb{R}^n$ of the curve such that $\varphi(\bar{t}_1) = c(t_1), \ldots, \varphi(\bar{t}_n) = c(t_n)$ then $\bar{t}_1 \leq \cdots \leq \bar{t}_n$.

In figure (1) we show a case of manipulation several points in a curve. Six points are chosen in four adjacent segments. In this section, we will discuss about as define the position of a set of control point, when we move several points in a curve. First we will present an example of this situation and show how to manipulate it. After it, this example will be widespread.

Each $\{c(t_j)\}_{j=1,\ldots,6}$ satisfies the equation 1. Then, is possible join these six matrix in one matrix equation $[c] = [N][P]$, where: $[c] = \begin{bmatrix} c(t_1) & c(t_2) & c(t_3) & c(t_4) & c(t_5) & c(t_6) \end{bmatrix}^T$, $[P] = \begin{bmatrix} P1 & P2 & P3 & P4 & P5 & P6 & P7 \end{bmatrix}^T$ and $[N]$ is the matrix of base b-spline of each point. Note that by Cox-Boor algorithm the base $N_j(t_i)$ is equal to zero when t varies out off the support of the function, ie, outside of the range determined by the respective nodes of control points of the segment that contains this point. Because $N_j[t_i] = 0$ if $j < distance(P_1, P_i)$ or $j > distance(P_1, P_i) + 4$. So, in this example $[N]$:

$$\begin{bmatrix} N_1[t_1] & N_2[t_1] & N_3[t_1] & N_4[t_1] & 0 & 0 & 0 \\ 0 & 0 & N_3[t_2] & N_4[t_2] & N_5[t_2] & N_6[t_2] & 0 \\ 0 & 0 & N_3[t_3] & N_4[t_3] & N_5[t_3] & N_6[t_3] & 0 \\ 0 & 0 & N_3[t_4] & N_4[t_4] & N_5[t_4] & N_6[t_4] & 0 \\ 0 & 0 & N_3[t_5] & N_4[t_5] & N_5[t_5] & N_6[t_5] & 0 \\ 0 & 0 & 0 & N_4[t_6] & N_5[t_6] & N_6[t_6] & N_3[t_6] \end{bmatrix}$$

We can generalize this example by the following theorem:

Theorem 1: Taken $P_1, \ldots, P_n$ different ordered points in a curve of degree k, if any subset $P_i, \ldots, P_{i+j}$ satisfies $j <= distance(P_i, P_{i+j}) + 3$, the positions of these can be manipulated by MLM.

The base matrix is a block matrix. Then, we can say that N has m block nonzero $B1, \ldots, Bm$. Each block can be decomposed as in the equation (3). The matrix of the potential parameters is LI. Indeed, as, by hypothesis, among the n chosen points has not more than $k+1$ points in the same segment. We can say that was chosen $l < k+1$ points in a segment, these l points determines l rows in some B. We can complete B with $k+1-l$ similar rows, but with diferent parameters; in this case the matrix obtained is a Vandermonde Matrix, so this is LI, and the l rows vectors is a subset of a LI set, then LI. As each block is LI, N is LI. So is possible use the MLM method to solve the equation

$$[N][\Delta P] = [\Delta c] \tag{8}$$

It is similar to equation (5), but in this the matrix N has n rows and $distance(P1, Pn) + 4$ columns. The ij-th elements $N_j[t_i]$ is zero if $j < distance(P_1, P_i)$ or $j > distance(P_1, P_i) + 4$, as the example. The solution for equation (8) is sufficient to manipulate the curve. Proving the theorem. □

When satisfied the assumptions of Theorem 1, we can guarantee the solution to the problem of manipulation of the position of points by MLM. This method can only be used when the rows of the base matrix are LI. If the set of manipulated points not satisfy this theorem, we can insert control points in segments, in such a way that does not change the geometry of the curve, as shown in [23]. Thus, the new matrix of the base attends the assumptions of the theorem.

The Theorem 1 include cases as the manipulation of two chosen points are independent. We will discuss when this occurs, in following theorem:

Theorem 2: Taken a uniform b-spline curve of degree k, and a set of n ordered points $P = \{p_1, \ldots, p_n\}$ if all pair of adjacent points p_i and p_{i+1} are such that $distance(p_i, p_{i+1}) > k$ then the manipulation problem is reduced to n isolated problems.

Indeed, if $distance(p_i, p_j) > k$ then the set of control points of the segment that contains p_i does not interfere in the segment that contains p_j, because the set of control points of a segment affects only $2k+1$ segments, as was previously stated. Particularly, if all the manipulated points do not interfere in its adjacents then they will not interfere in any other point, among of others in P. Hence, the manipulation of all points of P is independent, proving the theorem. □

Based on Theorem 2 we can divide a manipulation problem in several minor problems, when two adjacents points satisfy $distance(P_i, P_{i+1}) > k$. This procedure is good to reduce the quantity of the calculations on MLM. Furthermore, the matrix N, in this case, would have a high degree of sparse and may cause numerical problems.

In section of Manipulation of One point we introduced the possibility of manipulation of features relationship first and second derivatives. But the Theorem 1 talk about only position. This occurs, because to use the MLM is necessary that the rows of bases matrix are LI. If we manipulate, for example, the first derivative of four points in a same segment, the decomposition done in (bspline:base) will have the last column equals zero. So the rows are LD (Linear Dependent), then we can not use the MLM.

There are some cases, than those already presented in the theorem, that is possible to use the MLM. We can cite if the user manipulate in two points, the position and the first derivative. In this case, the base matrix will be LI, so is possible use the MLM. A proposal to solve some pathological cases is to use the Least-Square Method (LSM). This result is an aproximated solutions. It is important pay attention in fact that it can show an unexpected result by user. Some of cases are:

- When the quantity of chosen points in a same segment exceeds the curve order. In this case, say that the curve has degree k (so, order $k+1$) and taken n points in same segment with $n>k+1$. In this case, $N(t) \in M(n \times (k+1))$. So, $N(t)$ has more rows than column, ie, the set of rows of $N(t)$ is LD. Thus, is not possible use the Minimum-Length Method (MLM) to find the new position of the control points of a curve to pass for these points, because this method assume as hypothesis that the matrix has rows LI. In this case, is possible use the Least-Square Method (LSM) to obtain an approximated solution (but this may generate an unexpected result by user). Other solution is insert control points in these segments (increase the resolution of the curve) making the system by Soluble MLM.
- When two equals points in curve are taken in different points, $c(t_1) = c(t_2)$ with $t_1 = t_2$, but $\Delta c(t_1) \neq \Delta c(t_2)$. In this case, $N(t)$ has two equals rows, so LD (Linearly Dependent). Then, is not possible to use the method MLM, as in [9]. But if we take the matrix $\bar{N}(t)$, which the columns are a subset LI, of maximum cardinality, of columns of $N(t)$. Then we can find a projection of $[\Delta c]$ on solution space of $N^T N[\Delta P] = N^T[\Delta c]$ (we considers this projection as $[\Delta \bar{c}]$) and so by LSM $[\Delta P] = \bar{N}(\bar{N}^T \bar{N})^{-1} \bar{N}^T [\Delta \bar{c}]$. With this solution of $[\Delta P]$ it is possible apply the MLM and obtain a solution that minimize the error. This solution is an approximation, and again, the solution can be an unexpected result to the user.

A generic solution for this problem is not possible using MLM. In fact, this is an Hermite Birkoff Interpolation Problem. An indepth discussion can be found in [24].

VI. Future Work

A natural evolution of this system is the generation of volumetric objects, based on curves drawn on a plane, and deform it, through curves drawn in boundary of this one.

About curves, there are three interesting functionalities to add in the system. First we could change the resolution of the curve allowing manipulation of the curve in several resolution. Second, we could implement the concepts shown in this article with other types of splines. Finally, we could define generic methods to solve the Hermite Birkoff Interpolation Problem.

VII. Conclusion

In this work, we show some aspects of the construct of curve modeling system, based in a multitouch interface. This is a new paradigm compared to WIMP systems. So, it is necessary to focus on the differences not to lose the advantages of a system with a natural interface.

There are some systems to model curves based on the WIMP paradigm. But these systems are not intuitive. A better proposal to model curves is to use systems based in a natural interface. In these systems, it is possible to create and manipulate a curve with several points. It is more difficult to do it with a mouse.

The manipulation of a curve with several points requires another theorical approach as compared to CAD systems. This theory was presented in this article is an extention of the theory showed by [9]. The manipulation of curves with several fingers, possible in multitouch systems, enables a new and intuitive mode of generating curves in sketch-based modeling systems.

Acknowledgment

We would like to thank the CNPq, for funding the first author of this article, and the VISGRAF - IMPA, by providing an environment for fruitful research.

References

[1] L. Olsen, F. F. Samavati, M. C. Costa, and J. A. Jorge, "Sketch-based modeling: A survey," vol. 33, no. 1. Computers Graphics, 2009, pp. 85–103.

[2] A. Nealen, O. Sorkine, M. Alexa, and D. Cohen-Or, "A sketch-based interface for detail-preserving mesh editing," vol. 24, no. 3. ACM Transactions on Graphics, July 2005, pp. 1142–1147.

[3] T. Igarashi, S. Matsuoka, and H. Tanaka, "Teddy: A sketching interface for 3d freeform design." ACM SIGGRAPH 2007.

[4] P. Keir, J. Elgoyhen, M. Naef, J. Payne, and P. A. M. Horner, "Gesture-recognition with non-referenced tracking." The Computer Journal, 2006.

[5] D. Bowman, E. Kruijff, J. LaViola, I. Poupyrev, and M. Mine, "The art and science of 3d interaction." in *Tutorial Notes from the IEEE Virtual Reality*. Conf. IEEE Computer Society, 2000.

[6] J. LaViola, "A survey of hand posture and gesture recognition techniques and technology," Tech. Rep., 1999, department of Computer Science, Brown University, Providence, Rhode Island.

[7] P. J. Schneider, “An algorithm for automatically fitting digitized curves,” in *Graphics gems*. Academic Press Professional, 1990, pp. 612–626.

[8] S. F. Frisken, “Efficient curve fitting,” in *Journal of Graphics Tools*, vol. 13, no. 2. A K Peters, 2008, pp. 37–54.

[9] B. Fowler and R. Bartels, “Constraint-based curve manipulation,” in *Computer Graphics and Applications, IEEE*, vol. 13, no. 5, 1993, pp. 43–49.

[10] L. Olsen, F. F. Samavati, and M. C. Costa, “Fast stroke matching by angle quantization.” vol. 33, no. 1. first international conference on immersive telecommunications (ImmersCom 2007), pp. 85–103.

[11] M. Kaltenbrunner and R. Bencina, “Reactivision,” Music Tecnology Group - Pompeu Fabra University - Barcelona, June 13, 2009. [Online]. Available: http://reactivision.sourceforge.net/

[12] “Reactable,” Music Tecnology Group - Pompeu Fabra University - Barcelona, June 13, 2009. [Online]. Available: http://mtg.upf.edu/reactable/

[13] T. Pavlidis, “Curve fitting with conic splines,” vol. 2. ACM Transaction on Graphics, 1983, pp. 1–31.

[14] C. H. Reinsch, “Smoothing by spline function,” vol. 10. Numerische Mathematik, 1967, pp. 177–183.

[15] M. Grossman, “Parametric curve fitting,” vol. 14, no. 2. The Computer Journal, 1970, pp. 169–172.

[16] F. Yamagushi, “A new curve ffitting method using a crt computer display.” Graphics and Processing, 1978, pp. 425–437.

[17] M. Grossman, “Parametric curve fitting,” vol. 14, no. 2. The Computer Journal, 1970, pp. 169–172.

[18] W. L. Chong, “Automatic curve fitting using an adaptative local algorithm,” in *ACM Transaction on Mathematical Software*, vol. 6, no. 1, 1980, pp. 45–57.

[19] W. Boehm, G. Farin, and J. Kahman, “A survey of curve and surface methods in cagd,” in *Computer Aided Geometric Design*, vol. 1, 1984, pp. 1–60.

[20] P. J. Davis, *Interpolation and Aproximation*. Dover Publication, 1980.

[21] C. D. Boor, *A practical guide to splines*. Springer, 2001.

[22] D. Rogers and A. Adams, *Mathematical Elements for Computer Graphics*. McGraw Hill Book, 1998.

[23] L. Piegl and W. Tiller, *The NURBS book*. Springer Editions, 1995.

[24] L. J. Schoenberg, “On hermit-birkhoff interpolation,” vol. 16, no. 3. Journal of Mathematical Analysis and Application, 1966, pp. 538–543.