

# Interacção Multi-toque no contexto do DJing

Pedro A. Lopes J. A. Madeiras Pereira Alfredo Ferreira

Departamento Engenharia Informática

INESC-ID/IST/Universidade Técnica de Lisboa

R. Alves Redol, 9, 1000-029 Lisboa, Portugal

pedro.lopes@ist.utl.pt, jap@inesc-id.pt, alfredo.ferreira@inesc-id.pt

## Resumo

Nos últimos anos têm-se procurado novas formas de interacção para os Disc-Jockeys (DJs), no entanto poucas soluções conseguem capturar a sinergia entre os benefícios da interacção multi-toque e a virtualização do ambiente de manipulação áudio, evitando o input através de dispositivos tradicionais, como o rato ou o teclado. Centrado no envolvimento de um grupo de DJs profissionais, neste artigo propomos um sistema modular, caracterizado pela interacção natural, consistente com os gestos tradicionalmente usados e sem necessidade de objectos tangíveis adicionais. O sistema proposto foca os principais desafios de interacção em aplicações de DJing virtual, usando superfícies multi-toque.

## Palavras-Chave

Superfície Multi-toque, DJ, Desafios Interacção, Interface, Aplicação Modular, Concepção Centrada no Utilizador

## 1. Introdução

A actividade do DJ contemporâneo ultrapassou há muito a simples escolha de discos em ambientes festivos e, actualmente, o DJ é possuidor de conhecimentos técnicos extensos e capaz de executar tarefas complexas na interacção com o hardware que utiliza. Durante três décadas este equipamento (denominado *DJ Setup*) manteve-se quase inalterável, sendo tipicamente composto por fontes sonoras (gira-discos, leitores de CD), uma mesa de mistura (dispositivo capaz de misturar pelo menos dois sinais áudio) e discos (registos no formato físico: vinil ou CD) [7].

Com o virar do milénio, operou-se uma revolução na digitalização do *Setup Tradicional* dos DJs, fruto da massificação dos computadores portáteis. No entanto, as aplicações comuns de DJing virtual possuem um modelo de interacção não-natural, que recorre tipicamente a dispositivos de entrada tradicionais (rato e teclado) ou controladores hardware MIDI. Assim sendo, torna-se difícil a interacção com os utilizadores, consequência da disparidade entre os comandos virtuais e as acções já aprendidas do *Setup Tradicional*. Observa-se nestas aplicações um aumento da curva de aprendizagem limitando-se a actividade, tipicamente bimanual, a um fluxo de controlo único.

Ainda na primeira metade da década transacta a indústria concebeu um novo tipo de sistema, capaz de minimizar os problemas de interacção do *Setup Virtual*; tal sistema é composto pela união dos anteriores, como está patente na Figura 1, e denomina-se *Setup Híbrido*. Trata-se de um sistema digital de controlo de uma aplicação de DJing virtual através dos componentes típicos: gira-discos ou leitores

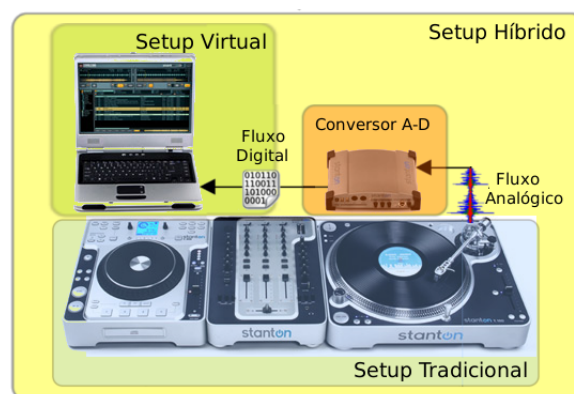


Figura 1. Setup Tradicional, Virtual e Híbrido.

de CD. Neste *Setup* o utilizador pode controlar o *playlist* do ficheiro digital, manipulando directamente o disco de vinil que contém informação codificada de *tracking*, o que permite ao software determinar posição do ficheiro áudio. Embora os problemas inerentes à manipulação não-natural do *Setup Virtual* sejam resolvidos, tais soluções híbridas voltaram a despoletar a necessidade de recorrer aos limitados equipamentos analógicos tradicionais com elevados custos de manutenção, aquisição e transporte associados.

Para minimizar estas desvantagens propõe-se uma alternativa baseada em superfícies de interacção multi-toque. Pretendemos assim virtualizar as tarefas do DJ, e aliar às vantagens da manipulação de áudio digital uma interacção natural que se apresenta consistente com os gestos tradi-

cionais, identificados através do envolvimento de um grupo de DJs profissionais dispostos a dar o seu contributo para a evolução deste projecto.

## 2. Trabalho Relacionado

Recentemente, foram propostas metáforas alternativas à interacção entre o DJ e o seu *Setup*. Patten et al. desenvolveram o sistema AudioPad [12], uma interface tangível em que um conjunto de objectos físicos permite ao DJ manipular o *playout* dos ficheiros áudio e interligar os diversos componentes. Na mesma linha conceptual surge a proposta de Jordà et. al, a reacTable [9], demonstrando a possibilidade de uma extensa manipulação de trechos sonoros (*loops*) através de objectos tangíveis e multi-toque. Por fim, vale a pena analisar o contributo de Hansen e Alonso na implementação de um conjunto de objectos para *scratch*<sup>1</sup> na reacTable. No entanto, o resultado dos testes efectuados mostrou que o DJ acostumado aos gestos tradicionais teve dificuldade em interagir com os novos objectos [5]. As propostas tangíveis supramencionadas ilustram a possibilidade de novas metáforas perante o paradigma da sequenciação de *loops* áudio ou síntese em tempo real, mas não foram desenhadas para o contexto concreto do DJing, faltando nelas muitos dos requisitos que identificamos no nosso estudo com o grupo de acompanhamento, nomeadamente: controlo por potenciómetros deslizantes e rotativos, acesso directo à mesa de mistura, manipulação directa do áudio e pré-escuta dos canais de áudio. Tais requisitos podem, em última instância, ser traduzidos por mapeamento directo dos conceitos do DJing tradicional nas interfaces virtuais.

Num eixo ortogonal Slayden et al. propõe uma sinergia entre sensores de flexão, óptico e acelerómetro num dispositivo-luva, denominada DjHammer [13]. Também no campo da emulação do *scratch*, Beamish et al. propõe o D'Groove [3], um gira-discos háptico capaz de uma emulação credível, mas que pouco oferece quando comparado a um Híbrido. Em suma, estas novas metáforas apresentam uma maior curva de aprendizagem para o utilizador-DJ, forçando-o a aprender um léxico de gestos não-familiares.

Fora do espectro académico surge um dos sistemas com maior sucesso das últimas décadas - o controlador multi-toque Lemur, Figura 2 - um dispositivo que permite enviar mensagens MIDI/OSC de forma a permitir ao utilizador controlar remotamente o software de DJing. Nas diversas análises que lhe foram dirigidas [4, 1] salienta-se a sua interface modular, composta por *widgets*, que o utilizador escolhe e posiciona no ecrã. Por outro lado, trata-se de um sistema proprietário fechado, não sendo possível ao utilizador/comunidade criar novos objectos. Destinado a uma vasta área de controlo de software de produção musical, não contempla até à data objectos específicos para

<sup>1</sup>Para o leitor não familiarizado, *scratch* denota uma forma de expressão musical realizada num gira-discos [13]. O DJ manipula directamente o áudio, movendo o disco para frente e para trás, complementando com uma acção na mesa de mistura de forma a "abrir" e "fechar" o volume com a outra mão, o que mostra claramente a necessidade de interacção bimanual neste tipo de tarefas.

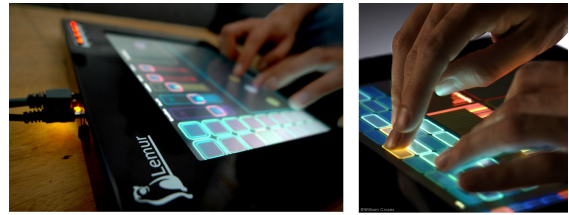


Figura 2. O controlador multi-toque, Lemur. (Extraído de [8])

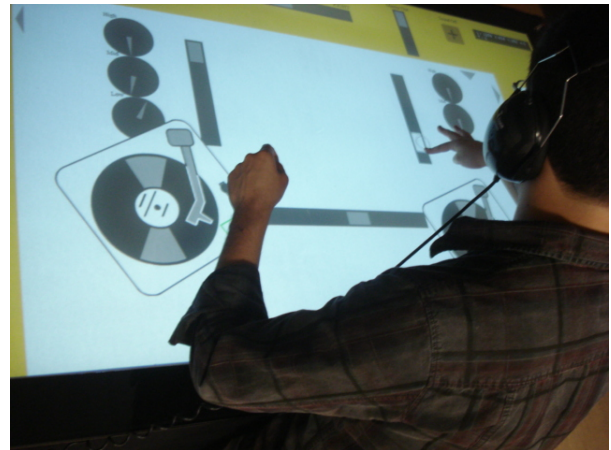


Figura 3. Utilizador testando o MtDjing

as tarefas do DJ (e.g., é notória a falta de um *widget* para simular o comportamento de um gira-discos). Daqui resulta que, apesar de tirar partido da interface multi-toque, o Lemur não oferece aos seus utilizadores DJ uma interacção natural, semelhante aos sistemas tradicionais.

Ainda assim, o Lemur mostra a possibilidade duma utilização proveitosa do multi-toque no contexto do DJing, mas deixa antever alguns desafios de interacção que exploraremos de seguida.

## 3. DJing em Superfícies Multitoque

A actividade de um DJ é muito particular; a análise, realizada durante uma sessão de DJing, permitiu-nos validar um conjunto de requisitos inerentes às suas acções, sendo que as actividades identificadas encontram-se na linha de pensamento de Hansen [6]. A nossa contribuição, MtDjing, posiciona-se nesta linha. Trata-se de uma interface horizontal multi-toque, organizada pelos conceitos fundamentais que encontramos no modelo mental dos DJs: fontes sonoras (gira-discos, sistema digital, leitor de CD), registos áudio (vinil, CD, ficheiro digital), manipuladores de sinal (potenciómetros deslizantes, interruptores, etc...) e as ligações possíveis entre eles. Estes conceitos, claros para o utilizador DJ, são mapeados nos objectos manipuláveis da nossa interface, como ilustrado na Figura 3.

Com o conhecimento adquirido pela análise dos vários *Setups* existentes, compreendemos como o interface para os

DJs pode ter dois papéis vitais: primeiro, reforçando a correlação estreita entre os conceitos virtuais (da aplicação em questão) e os gestos tipificados do DJing; e, em segundo plano, permite-nos desenhar um *workflow* com vantagens face aos *Setups* tradicionais. A escolha do multi-toque, permite-nos beneficiar das seguintes vantagens:

- **Interação Horizontal**, consistente com os *Setups* analisados, sem recorrer a artefactos/objectos adicionais para interagir com o sistema.
- **Interação Natural**, a mesma que encontramos nos sistemas tradicionais, despoletada pela acção das mãos/dedos sob objectos do *Setup*, com retorno visual na mesma superfície.
- **Operação Multi-tarefa** caracterizada pela actuação bimanual que oferece vantagens face ao *input* por dispositivos tradicionais, melhorando o desempenho das tarefas do DJ face aos sistemas virtuais [10, 2].

Assim sendo, para virtualizar as actividades dos DJs foi primeiro necessário desenhar interações compatíveis com os gestos tradicionais, tendo-se recorrido ao grupo de acompanhamento para identificar os gestos típicos para as seguintes acções: play, stop, mistura de dois sinais, interagir com equalizador, fazer pré-escuta do sinal de um canal, alternar entre dois sinais (*fast crossfading*) e alterar volume/velocidade/posição da faixa. Identificou-se que a maioria dos gestos é realizada por acção directa nos componentes hardware (i.e.: potenciómetros, interruptores, disco, agulha), podendo ser mapeada directamente nos componentes "virtuais" da interface. No entanto, existem tarefas mencionadas pelo grupo de acompanhamento que lidam directamente com a preparação/organização do *Setup*, nomeadamente o posicionamento do hardware e interligação dos dispositivos. Estas acções, suscitam alguns conflitos no contexto do multi-toque, pois assumem que é possível efectuar um conjunto de operações no mesmo objecto, usando sempre o mesmo *input* (toque); são elas: mover, rodar, escalar, interligar e "usar o objecto". Para solucionar este problema, optou-se pela aplicação de *handles* nos objectos como proposto por Nacenta [11], permitindo que a interacção dê sempre prioridade às acções específicas do DJing em detrimento do posicionamento dos objectos na cena virtual.

Outro desafio reside no facto da maioria das aplicações de áudio separar o processo de configuração (criar novas pistas, novos objectos DSP, etc...) da interacção com esses objectos, sendo que é uma das desvantagens reconhecidas ao controlador Lemur, que não permite a adição de objectos em tempo real. Para o sistema multi-toque pretendido, o utilizador pode adicionar os seus objectos de DJing e simultaneamente realizar as suas tarefas musicais, como vemos na Figura 3.

Por último, ainda mencionando os desafios da área do DJing, a emulação do comportamento do gira-discos afigura-se como a maior desvantagem identificada para os sistemas baseados em toque, face aos sistemas

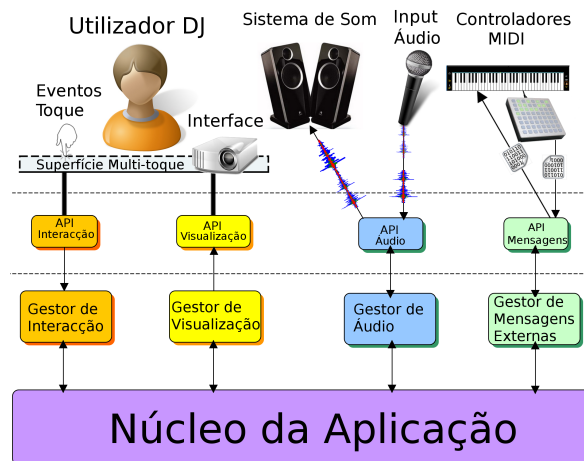


Figura 4. Diagrama da arquitectura proposta.

hápticos [3]. Testar-se-ão diferentes possibilidades com o grupo de acompanhamento de forma a obter a melhor interacção possível com um objecto virtual com a física emulada do gira-discos.

#### 4. Arquitectura do software MtDjing

Uma aplicação tem, neste contexto, de responder a requisitos técnicos ditados tanto pela indústria como pelas expectativas dos utilizadores-alvo; salientam-se assim as medidas de desempenho relativamente à latência percebida pelos utilizadores, capacidade de controlar ou receber mensagens de dispositivos externos<sup>2</sup> e ser independente das escolhas de hardware. As aplicações de áudio profissional operam tipicamente a latências de  $\approx 10\text{ms}$  [3]; na arquitectura MtDjing, ilustrada na Figura 4, o componente de áudio é independente do funcionamento dos restantes, podendo por isso ser executado num ambiente de alto desempenho. A separação dos módulos por funcionalidades e a sua inter-comunicação através do protocolo OSC [14] assegura simultaneamente a interoperabilidade e a possibilidade de um ambiente de execução distribuído, *multithreaded* assente sob *standards* e aberto à mudança.

O sistema encontra-se estruturado em quatro camadas: Núcleo da Aplicação, Gestão, API e Dispositivos. Desta forma os componentes externos interagem com a aplicação através da sua API, e todas as comunicações passam pelo núcleo da aplicação. Este é o módulo principal, responsável por assegurar a comunicação entre os restantes e garantir a persistência dos dados relevantes a cada instante. Enquanto isso, os restantes módulos ocupam-se da gestão de tarefas específicas. O Gestor da Interação interpreta o *input* do utilizador, que mais tarde será encaminhado para o Gestor de Visualização, para que as alterações na interface gráfica sejam consolidadas e o uti-

<sup>2</sup>No DJing, tal como nas restantes áreas da indústria musical, o standard "de facto" é o MIDI [14], embora comecem agora a surgir equipamentos hardware capazes de comunicar no protocolo OSC.

lizador receba *feedback* visual da sua acção. Simultaneamente, as operações no sistema afectam o retorno auditivo, responsabilidade atribuída ao Gestor de Áudio. Este módulo encontra-se implementado em Pure Data, um ambiente de síntese e manipulação de áudio em tempo real usado em projectos como reacTable [9], executando código C para as tarefas de DSP. Por último, a comunicação com dispositivos MIDI/OSC é realizada pelo Gestor de Mensagens Externas, capaz de traduzir as mensagens recebidas, de forma a estender o controlo do sistema, a múltiplos interfaces. Excluindo a camada de visualização, implementada em ActionScript, os restantes módulos foram desenvolvidos em C++, numa abordagem *cross-platform*, restando apenas desenvolver o módulo de comunicação com dispositivos MIDI externos.

## 5. Conclusões e Trabalho Futuro

Apresentámos uma proposta para um sistema baseado em interacção com superfícies multi-toque, capaz de oferecer aos DJs as funcionalidades típicas dos *Setups* tradicionais e as vantagens da virtualização dessas mesmas tarefas. Propomos um modelo conceptual simples, baseado no modelo mental dos utilizadores, obtido pelo envolvimento de um grupo formado por DJs profissionais de diversos géneros musicais. A diversidade dos utilizadores-DJ que acompanharam o desenvolver do sistema trouxe diferentes experiências e necessidades para o projecto, permitindo-nos desenhar um sistema para o qual os conceitos da interface sejam a chave do sucesso. No eixo arquitectural optámos por uma aplicação modular, que facilitará a introdução de novas funcionalidades, optimização do desempenho e manutenção do protótipo.

O trabalho futuro passará pela análise dos testes com novos DJs, de forma a avaliar a proposta e comparar a curva de aprendizagem face às soluções híbridas e virtuais. Por fim, continuar-se-á a adicionar funcionalidade ao protótipo para que este possa ser facilmente equiparado às soluções encontradas no mercado DJ.

## Referências

- [1] Products of interest: Lemur review. *Computer Music Journal*, 29(4):99–112, 2005.
- [2] R. Balakrishnan and K. Hinckley. Symmetric bimanual interaction. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 33–40, New York, NY, USA, 2000. ACM.
- [3] T. Beamish, K. Van Den Doel, K. Maclean, and S. Fels. Manipulating music: multimodal interaction for djs. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 327–334, New York, NY, USA, 2004. ACM.
- [4] Philip L. Davidson and Jefferson Y. Han. Synthesis and control on large scale multi-touch sensing displays. In *NIME '06: Proceedings of the 2006 conference on New interfaces for musical expression*, pages 216–219, Paris, France, France, 2006. IRCAM — Centre Pompidou.
- [5] K. Hansen and M. Alonso. More dj techniques on the reactable. In *Proc. of the Conference on New Interfaces for Musical Expression*, 2008.
- [6] K. F. Hansen. *The acoustics and performance of DJ scratching. Analysis and modeling*. PhD thesis, KTH, 2010.
- [7] K. F. Hansen and R. Bresin. *Analysis of a genuine scratch performance*, pages 519–528. LNCS 2915. Springer Verlag, Genova, Italy, selected revised papers, berlin heidelberg edition, apr 2004.
- [8] JazzMutant. <http://www.jazzmutant.com>. consultado em 27/05/2010.
- [9] Sergi Jordà, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. In *TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 139–146, New York, NY, USA, 2007. ACM.
- [10] Kenrick Kin, Maneesh Agrawala, and Tony DeRose. Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In *GI '09: Proceedings of Graphics Interface 2009*, pages 119–124, Toronto, Ont., Canada, Canada, 2009. Canadian Information Processing Society.
- [11] Miguel Nacenta, Patrick Baudisch, Hrvoje Benko, and Andy Wilson. Separability of spatial manipulations in multi-touch interfaces. In *Graphics Interface*, pages 175–182, Kelowna, B.C., Canada, 2009.
- [12] James Patten, Ben Recht, and Hiroshi Ishii. Interaction techniques for musical performance with tabletop tangible interfaces. In *ACE '06: Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*, page 27, New York, NY, USA, 2006. ACM.
- [13] April Slayden, Mirjana Spasojevic, Mat Hans, and Mark Smith. The djammer: "air-scratching" and freeing the dj to join the party. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1789–1792, New York, NY, USA, 2005. ACM.
- [14] Matthew Wright, Roger Dannenberg, Stephen Pope, Xavier Rodet, Xavier Serra, and David Wessel. Standards from the computer music community. In *International Computer Music Conference*, pages 711–714, Miami, FL, 2004. International Computer Music Association.

# Combining Different Types of Interaction on Multi touch Surfaces

Tarquínio Mota

CITI, Departamento de Informática, Faculdade de Ciências e Tecnologia,  
Universidade Nova de Lisboa, 2829-516 Caparica, Portugal  
tarquinio@gmail.com

## Abstract

This paper presents a survey of different methods of interaction with single-touch and multi-touch surfaces. While many methods are currently under research, they tend to be difficult to combine. This paper proposes a system that allows the use of different types of interaction on the same surface.

## Keywords

Interface, usability, pervasive, evaluation.

## 1. INTRODUCTION

Touchable interfaces are getting more common every day. Due to the popularization of devices such as PDAs, Tablet PCs, mobile phones and portable music players, touch screens are a technology that most electronic products consumers are getting used to. Initially the applications for these devices were more oriented to use a pen-like device (stylus). However with the appearance of devices such as the iPod Touch, in 2007, touch based interaction became more popular, allowing also the usage of multi-touch gestures. Another feature of the iPod, which is present in many other similar devices, is the presence of an accelerometer. This allows another type of interaction by moving the devices in 3D space.

On small devices, the type of technology used on touchscreens is either capacitive or resistive. When working on a larger scale, these solutions are not viable. The technologies used on smaller displays would be much harder to implement for something as big as a table or a wall, not to mention much more expensive. So there are different approaches that allow to interact in similar way with larger surfaces, using different types of sensors, as will be shown on the next chapter [2-8].

When talking about larger surfaces, the interaction is not only limited to touching the screen. It is also possible to interact by manipulating objects on the surface of the screen. These may be objects with specific visual markers, active objects that can also provide some type of feedback, or simply everyday objects.

In the next section existing prototypes will be presented showing different approaches to multi-touch interaction, focusing specially on surfaces with larger dimensions.

Section 3 presents a new approach to combine different types of interaction and the paper ends with preliminary conclusions and directions for future work.

## 2. RELATED WORK

Most of the approaches to create large scale interactive surfaces involve combining the image from a projector with some way to track the user's hands/fingers. This process is often based on using infrared illumination and capturing a reflection of that light with an infrared camera. Then by processing the captured images it is possible to extract the coordinates.

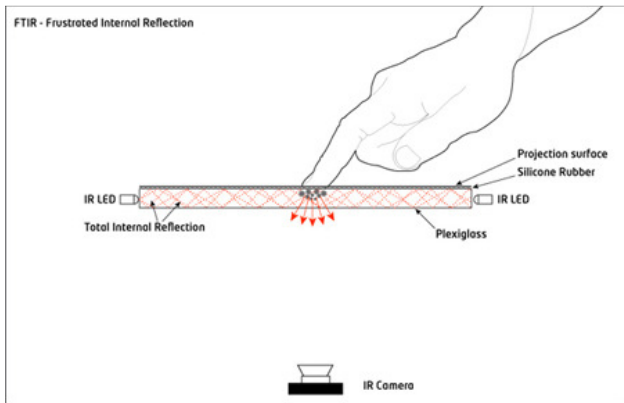
Even though multi-touch technology has been in development for several years [1], the solutions that existed were expensive and the performance was not that good, when compared with the results that are obtained with common devices today.

In 2006 Jeff Han presented a 36 inches wide interactive table with multi-touch sensing capabilities [2]. The main features of the table were high resolution and low cost, while being very scalable in terms of size.

His approach and others that will be presented on this paper are based on the principle of Frustrated Total Internal Reflection (FTIR).

This is a property that certain materials have that allows light to travel within them and always be reflected on the inside. The material commonly used on interactive surfaces is plexiglass.

When an object such as a finger touches the glass the contact area reflects some of the light to the outside. That light is then captured by an infrared camera and that data can be processed to obtain the coordinates.



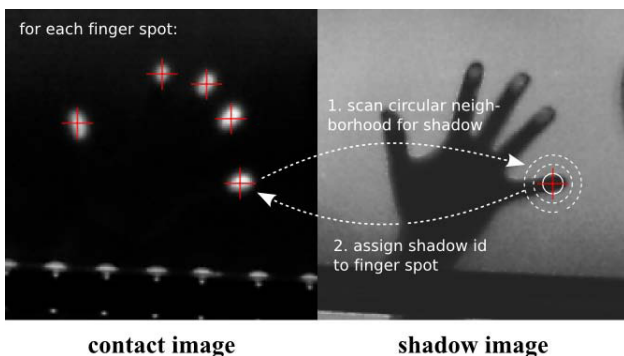
**Figure 1: FTIR touch sensing system**

**1.1 Interaction using only the hands**

Most multi-touch systems are made to detect the fingers only when they are touching the screen. Doing an analogy with a mouse, this means that it is possible to click a specific point by making the pointer jump to a specific spot, it is possible to have a dragging motion (moving the mouse while clicking one button). It is not possible to do the equivalent of just moving the mouse without pressing a button.

Florian et al propose a different system [3] where this is possible. They have two separate sources of light and use them alternately (changing several times per second). This allows the same infrared camera to capture two separate sets of images, one that tracks the objects hovering the screen and the other that tracks the points actually touching the screen (Figure 2).

In this way it is possible to move virtual cursors by hovering the hand over the screen, and detect when each of the fingers are touching the display, simulating a button click.



**Figure 2: Shadow processing [3]**

This system provides some extra information when compared to the previous ones, but some performance is sacrificed in the process. The frame rate of the capture is reduced, since two different sets of images are being

captured, plus changing between them takes time which means some more frames are lost in the process.

Usually when constructing large interactive surfaces the source of the image is a projector, which means the whole system ends up being very large. Wide-angle projectors or a combination of mirrors can be used to minimize the size, but in the end the whole system will still be very large.

Other types of technologies can also be used for touch sensing, like the DiamondTouch [4], that uses electrical signals instead of infrared light in order to identify the users. However this paper will focus mostly on systems using cameras, since they usually require less hardware to develop and are more affordable.

**1.2 INTERACTION BY MANIPULATING OBJECTS**

Another popular way to use interactive surfaces is by manipulating objects on top of them instead of using just the hands.

These objects can be passive and in this case they usually contain a certain type of tag to identify them. They can also be active objects with processing capabilities, sensors or feedback devices. Such objects can measure things like acceleration, distance, position, and offer different feedback types like haptic, visual and audio.

A successful example of this approach is the reacTable [5]. It is an electronic musical instrument, consisting of an interactive table and some objects containing fiduciary markers on them. These markers allow the computer to track and identify objects in the real world, and also identify rotation movements.

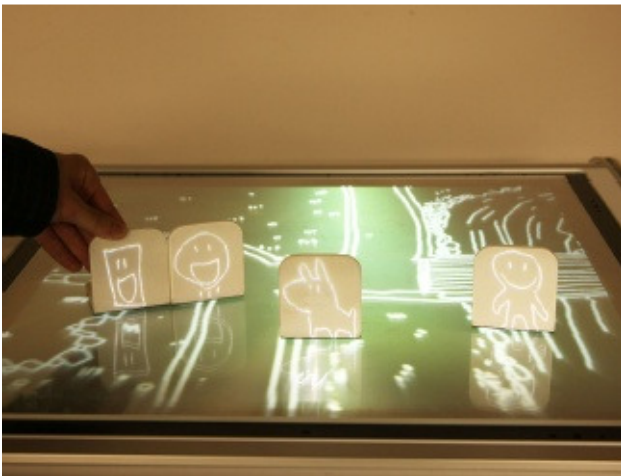
There are different types of objects on the table, and each one has a specific function. By placing objects close to each other links are created between them, changing the output sound wave. Some objects have only one marker, while others have different markers on each face. This means that the same objects can have different functions depending on which face is facing down.



**Figure 3: The reacTable [7]**

Usually these markers need to be touching the table in order to be recognized. Lifting one marker a few

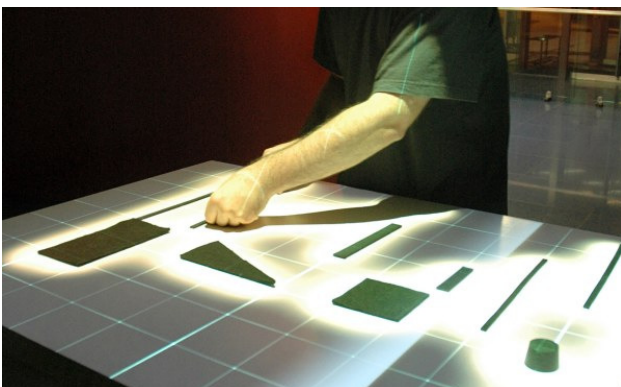
millimeters above the surface will result in a blurred image, unable to be recognized. The reacTable uses the objects only as an input. Other possibility is using the objects both as input and output, as suggested in Tablescape [6]. This work makes the virtual world break out of the screen, by projecting animations on the objects that are placed on the table. In Figure 4 this is used to create an interactive story. The user can move the characters and make them interact by placing them close to each other. According to how the story unfolds, different animations are used for each character.



**Figure 4: Tablescape [6]**

In this case, the objects placed on the screen serve both as input and output.

In 2006 Golan Levin presented Scrapple [7], an interactive table that works as a music sequencer. One interesting thing of this project is that it does not require any special objects or markers; the sequences could be composed by placing random shaped objects on the table.



**Figure 5: Scrapple installation [7]**

By moving, adding or removing objects on the table the users could change the sequence of sounds being played on a loop. The size and shape of the objects also affects how the sound is played.

There is already some work done on combining different types of interaction of the same surface. Leitner et al describe FLUX [8], a multi-touch table/board that can be used with the hands or with a Anoto tracking pen. For the finger tracking they use a normal FTIR approach similar to other described in section 2.1. The only difference is that it is necessary to use a special layer on top on the plexiglass for the Anoto pen to work.

These pens have a small camera aimed at the paper allowing it to identify its position by recognizing small patterns imprinted on the paper. This means that the top layer on this surface must be translucent to have a good projection while being opaque enough for the marks to be visible.

### 3. PROPOSED APPROACH

The goal is to create an interactive surface that combines different types of interaction presented in Section 2. Namely we want to have interaction by using free hands, with a stylus or with objects spread around the table. The detection needs to be as immune to noise as possible since it is meant to be used in public spaces (meaning that there are potentially interferences from other light sources that have to be dealt with).

Combining systems that use FTIR surfaces and fiduciary markers is not possible because the markers have specific patterns that need to be identified, but an FTIR camera will only capture blobs of light. On the other hand, systems that can capture fiduciary markers suffer more interference from normal light, making it harder to track fingers.

The reasons behind choosing a FTIR based system are that it offers a very good price/quality performance, plus it is very scalable and easy to assemble.

Accordingly to these principles, the proposed approach is to use a modified version of an FTIR table that uses active objects with infrared tags. Each object has its own infrared light and is connected wirelessly to the computer. This way the system is totally dependent on infrared light. Each interaction point will be perceived by the system as a blob of infrared light, no matter if it is a finger, a stylus or an object standing on the surface. Instead of using a source of light that is always on, the system will use different sources of light that will be on in different moments in time, blinking several times per second. Since all the lights have a connection to the computer, the software can order the different light sources to blink on request, and then perceive which blobs of light matched the blink order, in order to identify the respective points.

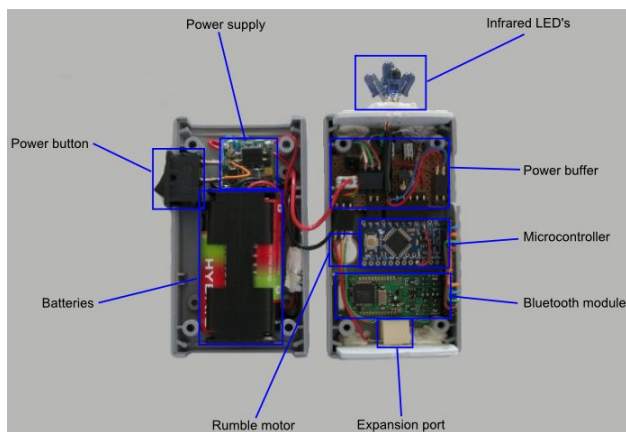
Having active sources of light also helps dealing with the potential interferences to the system. Typically when using infrared systems some type of calibration is used in order to remove interferences. The most common involve capturing the screen with no objects touching it, and then using a background subtraction or similar algorithm to remove the interferences. However this only works well for static

interferences that are already present during the calibration. Changes in room lighting (a light turned on, a flash being fired, or even the movement of shadow/light due to earth rotation) can cause interference to infrared based systems. Some interferences (like the ones caused by the earth moving) can sometimes be dealt with using adaptive background removal. By using active lights it is easier to deal with these situations: the only blobs relevant to the system will blink for sure when requested.

In order to illustrate these mechanisms an example will be presented where a system is designed to track fingers, plus three objects. At a specific moment in time, the software is tracking seven blobs of light. It then orders the lights that illuminate the fingers to blink, and blobs 1, 3 and 5 blink. Then it orders object 1 to blink, and blob 2 blinks. The next step is to order object 2 to blink, and in this case nothing changes. Finally, it orders object 3 to blink, and blob 6 blinks.

After these actions, the system can easily conclude that: blobs 1, 3 and 5 are fingers, blob 2 is object 1, blob 6 is object 3, object 2 is not in range, and blobs 4 and 7 are interferences.

The technology behind the objects to be used will be based on previous work [9]. Some hand controllers were developed that allowed to track and identify the position of different hands in 3D space. The goal here is very similar; we also want to track and identify specific points, so it makes sense to use the same type of hardware, which has proven to work and presents a good performance.



**Figure 6: Hand controller with feedback [9]**

Each object is connected to the computer, enabling the software to request its light to blink on request in order to be identified. Having to wait for the object to blink before it can be recognized causes a slight delay on the detection, but it is a necessary tradeoff in order to get a more trustworthy system.

These objects can be placed on top of the surface or the user can move them hovering over it, and do not require direct contact with the table.

#### 4. CONCLUSIONS AND FUTURE WORK

This paper proposes an alternative to combine different interaction types on the same surface, something that would not be possible with most of the existing approaches for each interaction type.

Having active objects means the system will be more complex but the advantage is that they also offer more possibilities of interaction, like capturing acceleration data or providing different types of feedback.

The system will be implemented and the performance will be compared to other systems with some common features. User tests will be made to evaluate the responsiveness of the users to the combination of these types of interfaces.

#### 5. REFERENCES

1. Lee, S., W. Buxton, and K.C. Smith, A multi-touch three dimensional touch-sensitive tablet, in Proceedings of the SIGCHI conference on Human factors in computing systems. 1985, ACM: San Francisco, California, United States. p. 21-25.
2. Han, J.Y., Multi-touch interaction wall, in ACM SIGGRAPH 2006 Emerging technologies. 2006, ACM: Boston, Massachusetts. p. 25.
3. Echtler, F., M. Huber, and G. Klinker, Shadow tracking on multi-touch tables, in Proceedings of the working conference on Advanced visual interfaces. 2008, ACM: Napoli, Italy. p. 388-391.
4. Dietz, P. and D. Leigh, DiamondTouch: a multi-user touch technology, in Proceedings of the 14th annual ACM symposium on User interface software and technology. 2001, ACM: Orlando, Florida. p. 219-226.
5. Jord, S., et al., The reacTable: a tangible tabletop musical instrument and collaborative workbench, in ACM SIGGRAPH 2006 Sketches. 2006, ACM: Boston, Massachusetts. p. 91.
6. Kakehi, Y., et al., Tablescape animation: a support system for making animations using tabletop physical objects, in SIGGRAPH 2009: Talks. 2009, ACM: New Orleans, Louisiana. p. 1-1.
7. Levin, G., The Table is The Score: An Augmented-Reality Interface for Real-Time, Tangible, Spectrographic Performance. 2006.
8. Leitner, J., et al., Flux: a tilting multi-touch and pen based surface, in Proceedings of the 27th international conference extended abstracts on Human factors in computing systems. 2009, ACM: Boston, MA, USA. p. 3211-3216.
9. Mota, T.M.E., Interfaces based in Gestures and Motion, in Proceedings of Whole Body Interaction Workshop in Conference on Human Factors in Computing Systems 2010, ACM: Atlanta, Georgia, USA.