

## 基于视觉的互动游戏手势界面工具箱\*

武汇岳<sup>1,2+</sup>, 张凤军<sup>1</sup>, 刘玉进<sup>1</sup>, 胡银焕<sup>1</sup>, 戴国忠<sup>1</sup>

<sup>1</sup>(中国科学院 软件研究所 人机交互技术与智能信息处理实验室, 北京 100190)

<sup>2</sup>(中国科学院 研究生院, 北京 100049)

### Vision-Based Gesture Interfaces Toolkit for Interactive Games

WU Hui-Yue<sup>1,2+</sup>, ZHANG Feng-Jun<sup>1</sup>, LIU Yu-Jin<sup>1</sup>, HU Yin-Huan<sup>1</sup>, DAI Guo-Zhong<sup>1</sup>

<sup>1</sup>(Intelligence Engineering Laboratory, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

<sup>2</sup>(Graduate University, The Chinese Academy of Sciences, Beijing 100049, China)

+ Corresponding author: E-mail: wuhuiyue@iel.iscas.ac.cn

**Wu HY, Zhang FJ, Liu YJ, Hu YH, Dai GZ. Vision-Based gesture interfaces toolkit for interactive games.**

*Journal of Software*, 2011, 22(5): 1067-1081. <http://www.jos.org.cn/1000-9825/3733.htm>

**Abstract:** In this paper, a toolkit is created for designing vision-based gesture interactions. First, an abstract model for non-contact devices is proposed. Then, based on a data flow diagram method and an interactive learning approach, the IEToolkit is presented. It is designed based on the attributes of vision interaction and shields the underlying details of the computer vision algorithms. It has the following characteristics: a scalable interface to facilitate developers to add new classifiers, a unified management mechanism that provides dynamic configuration for all of the classifiers, and a visual user interface that supports the definition of a high-level semantic gesture. Finally, several prototypes are given. Experimental results show that the IEToolkit can provide a unified platform and a general solution for vision-based hand gesture games.

**Key words:** HCI (human-computer interaction); hand gesture interface; interaction technology; toolkit

**摘要:** 分析了视觉手势的交互特征,提出了非接触型设备交互模型,基于数据流图方法建立了支持连续信息输入的数据流模型来描述视觉手势信息的处理流程,基于“paint-view-correct”隐喻构建了手势界面开发工具 IEToolkit. 系统的主要特点包括:使用插件设计思想提供了可扩展的接口,方便开发人员对系统进行扩展;实现了对平台内部多个分类器的统一管理,方便用户对这些分类器进行动态配置;提供了可视化的用户界面,用户能够根据不同的应用灵活地定义高层的手势交互语义;屏蔽了图像处理、机器学习等底层的技术细节,降低了界面开发难度.最后介绍了基于 IEToolkit 的软件系统开发方法与应用实例.实例表明,该界面工具能够为基于视觉手势的互动游戏的设计开发提供统一的平台和有效的解决方案.

**关键词:** 人机交互;手势界面;交互技术;工具箱

**中图法分类号:** TP391      **文献标识码:** A

\* 基金项目: 国家自然科学基金(U0735004, 60673188); 国家重点基础研究发展计划(973)(2009CB320804); 国家高技术研究发展计划(863)(2009AA01Z337, 2008AA01Z303)

收稿时间: 2009-01-15; 修改时间: 2009-07-08; 定稿时间: 2009-09-26

基于视觉的用户界面(vision based interfaces,简称 VBI)是 post-WIMP 时代的一种重要的界面形式,许多计算机系统都将摄像头作为一种输入媒介,目的是使用户能够在真实环境中以更加自然和直观的方式与计算机交互<sup>[1]</sup>.与传统的 WIMP 交互方式相比,VBI 能够充分利用用户的头、双手、双脚或者身体的其他部位<sup>[2-11]</sup>参与交互,从而提供给用户更大的交互空间、更多的交互自由度和更逼真的交互体验,因此迅速成为国内外研究的热点,并被广泛应用于虚拟/增强现实、普适计算、智能空间,尤其是基于计算机的互动游戏等多个领域.

但是,基于 VBI 的系统设计是一项十分困难的工作<sup>[12-16]</sup>,因为此类系统的构造往往涉及复杂的图像处理和机器学习算法等方面的知识,只有 VBI 研究领域内的专家用户才能掌握,并且常常需要对各种算法进行繁琐、重复的开发,这极大地阻碍了 VBI 的应用.对于非计算机视觉专业的普通游戏开发人员来说,需要一种通用的 VBI 开发工具,能够方便、快速地实现特定领域的 VBI 应用,从而提高设计和开发的效率.在计算机视觉领域,尽管研究者们根据不同的应用需求开发了许多手势识别工具箱或者专门的手势库,但大多数已有的视觉工具箱或者类库难以直接使用或者与已有的游戏开发工具快速、有效地集成.其主要原因表现为以下几方面:

- (1) 基于 VBI 的游戏开发要求开发人员不仅要具备丰富的游戏设计经验,还要掌握视觉处理以及交互技术方面的经验知识.大多数游戏开发人员虽然熟悉传统的可视化游戏设计工具如 Flash,Photoshop 或者 Maya, 3DMax 等,却并不具备开发 VBI 所必备的图像处理、机器学习以及模式识别等专门的领域知识.
- (2) 它们主要致力于底层的图像处理算法和手势识别技术,对高层的手势事件的处理和管理等方面提供的支持非常少,对复杂的交互设计支持不足,从而把大量的交互设计工作留给界面开发者.
- (3) 这些工具箱在开发技术、手势处理和应用领域等方面的个体差异非常明显,因此就其本身来说就很难使用,更难以与传统的游戏设计工具有效集成,并达到在不同的开发平台之间互相通用的目的.

基于视觉手势的互动游戏目的是给用户提供一种新的交互体验,通常并不包含过于复杂的游戏情节.因此,开发一个基于视觉手势的界面工具的主要目的就是使开发人员不必局限于复杂的图像处理、机器学习等底层的技术细节,从而有效地加速游戏开发过程.本文面向普适娱乐领域给出了一个支持视觉交互的手势界面工具 IEToolkit(interactive entertainment toolkit,简称 IEToolkit)的设计和实现,它从底层到高层完全基于视觉手势交互的特征进行设计,能够支持灵活的视觉手势交互信息的处理,支持高层的语义事件的处理和管理,简化基于视觉的手势界面开发过程,从而支持快速的游戏原型系统开发.

## 1 相关研究

目前已有的视觉手势库如 HandVu<sup>[17]</sup>和 GT<sup>2</sup>K<sup>[18]</sup>等都是由计算机视觉的专业人员设计开发的,而且其面向的用户也基本上是计算机视觉专业人员,普通的游戏开发人员必须深入到底层才能了解其工作机制,因此它们大都不能有效地支持快速开发.Lego Mindstorms Kit<sup>[19]</sup>是使计算机视觉技术逐渐走向普通开发者的先驱工作之一,其可视化的编程界面能够辅助用户迅速地实现他们的设计思想.但是,该工具仅仅利用了颜色或亮度等简单的视觉信息来辅助完成特定领域的任务例如机器人装配,不能识别较为复杂的视觉特征,因而其应用范围比较有限.Cambience<sup>[20]</sup>提供了一个简单的界面,利用用户的手势动作为来控制不同的音效,例如将用户的动作强度映射为音量的大小.与 Mindstorms Kit 类似的缺陷是,Cambience 仅仅利用手势运动信息来控制音效,因而其通用性不强.Fails 等人提出了一个基于视觉交互的设计工具 Crayons<sup>[12]</sup>.它使用了一种“绘画隐喻(painting metaphor)”方式和“描绘-观察-修正(paint-view-correct)”的逐步求精的视觉处理过程,将传统的机器学习和图像处理算法进行修改和封装,以满足便捷开发的需求.Crayons 的缺点是不支持运动信息的捕获与分析.总的来说,这些工具箱所提供的功能普遍来说都较为简单,缺乏灵活性与可扩展性.因此,它们虽然能够在一定程度上降低开发门槛,但同时也限制了其应用领域,无法支持更为复杂的应用.

与本文提出的 IEToolkit 目标较为接近的工作是 Maynes-Aminzade 等人<sup>[13]</sup>设计开发的一个支持快速的视觉交互开发的工具箱 Eyepatch.他们采用一种基于样本的学习方法(example-based approach)对分类器进行训练,并从视频流中快速提取出有用的数据信息.传统的界面设计者即便不具有任何的计算机视觉专业知识也能很容易地将训练结果与其他原型开发工具如 Flash,d.tools 等有效集成,从而支持不同的视觉交互原型系统的快速

开发.但是从交互的角度来看,Eyepatch 缺乏较好的交互机制,并没有考虑一些基本的视觉交互特征,缺少高层的语义事件处理策略,暴露出对于复杂交互的设计支持不足的问题.与此对比的是,我们提出的 IEToolkit 在支持颜色、形状、特征、运动等多种视觉交互信息的灵活处理的同时,围绕着视觉交互的主要特征提供了对于高层的语义交互事件的处理和管理机制,以及提供了各种常用的视觉交互技术统一支持机制,并且其可扩展的体系结构能够支持无处不在的计算环境下,面向互动娱乐的各类应用的便捷式开发.

## 2 IEToolkit 的设计目标

基于前面的分析,本文给出了 IEToolkit 的设计目标,主要包括以下几方面:

- (1) 一个有效的视觉界面工具应该支持开发人员迅速地从视频流中提取出 VBI 通常所关注的有用信息.传统的视觉工具箱已经在这方面进行了一些探索,如文献[17-18,21]等,但这类工具箱主要面向的是计算机视觉领域的专家级用户,对普通开发人员来说门槛过高.
- (2) 支持快速而有效的原型系统开发.视觉工具箱应该提供可视化的设计界面,从而降低对于普通游戏开发者在视觉方面的技能要求.只要他们熟悉类似于 Visual Basic 这样的可视化工具,并不需要深入了解太多底层的图像处理、机器学习等算法,就能够迅速掌握并熟练使用这个软件系统,从而将开发注意力集中在游戏设计任务本身.文献[12,19,20]等虽然能够在一定程度上降低开发门槛,但同时却也限制了其应用领域的范围.
- (3) 提供灵活的交互机制,支持高层的视觉交互事件处理和交互技术管理机制.主要表现在两方面:一方面,VBI 是通过视觉事件驱动的<sup>[1]</sup>,目前许多的工具箱系统仅仅停留在底层的视觉信息处理层次上,普遍缺乏高层的语义事件抽象;另一方面,传统的 GUI 是接触型和离散交互为主的方式,而 VBI 是非接触型的连续交互方式,二者的交互机制有很大的不同.Eyepatch 能够支持开发不同类型的应用<sup>[13]</sup>,但它没有从本质上分析视觉交互的特征,缺乏对各种通用的视觉交互技术的统一管理.我们提出的 IEToolkit 将力求提供一个面向视觉交互特征的高级事件处理模型,并使用一种可重用的模式来管理各种通用的视觉交互技术.
- (4) 软件架构的可扩展性.VBI 的研究已受到了越来越多人的关注,大量的新技术和交互设备层出不穷,IEToolkit 必须提供一种方便的机制,使得新的交互技术、视觉处理算法或者交互设备可以方便、容易地加入.

## 3 视觉手势交互机制

### 3.1 手势事件模型

VBI 主要是通过视觉事件驱动的,一个设计良好的手势界面工具箱首先应该具备好的手势事件模型.手势事件是形成交互的最终数据形态,事件的语义信息由界面设计人员定义,通常与任务直接相关.下面给出一个有效的视觉手势事件模型.

**定义.** 手势事件是由用户通过摄像头所引发的交互活动的信息承载体.一个静态手势事件(static gesture event,简称 SGE)的结构模型可以形式化表示为如下的元组形式:

$$SGE=(ID,Type,“posture”,t,r,x,y,s,a,TimeStamp),$$

其中,*ID* 是事件的唯一标识符;*Type* 指定该事件的类别;“*posture*”是字符串,标识事件的语义信息,如果为空则为不带语义信息的手势事件,例如位置的移动、速度的改变等,如果不为空则为带语义信息的手势事件.例如,“挥手”表示关闭一个文档;*t* 为标志位,如果目标正在被跟踪则为 1,否则为 0;*r* 为标志位,如果手势识别成功则为 1,否则为 0;*x,y* 通常为图像坐标系中跟踪对象的质心位置,其中,图像的原点在左上方;*s,a* 分别表示缩放系数(scale)和旋转角度(angle);*TimeStamp* 表示从检测到目标对象的第 1 帧起总的持续时间,单位为秒.

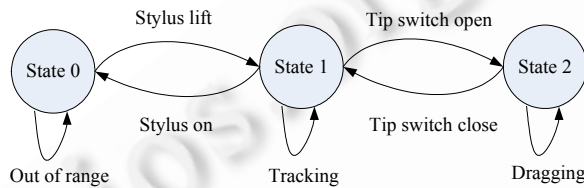
由于 XML 具有平台无关性、自描述性、易于标准化等特点,因此本文选择 XML 语言来表达视觉交互应用中的各种事件.用户引发的各种手势事件被封装成 XML 节,每一个 XML 节表示一个单一的事件,游戏平台端

通过解析 XML 字符流还原事件,然后根据会话标志符和事件参数将事件发送给上层的逻辑处理模块.

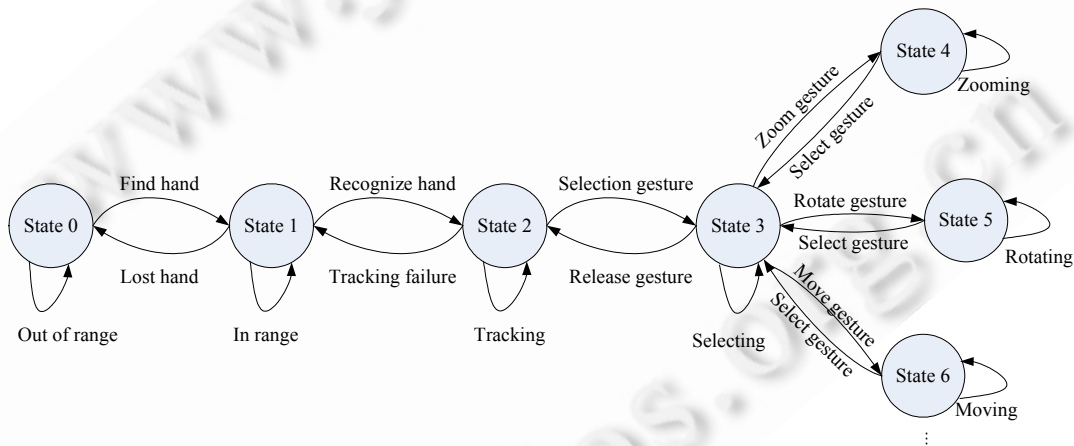
### 3.2 交互模型

#### 3.2.1 状态转移模型

在 HCI 中,几乎所有的交互技术都是依赖于特定的交互设备而设计的.因此,设备抽象是很多用户界面的标准需求.一个被广泛应用于 GUI 的成功例子是 Buxton 基于人类运动/感知系统提出的设备状态转移模型<sup>[22]</sup>,如图 1(a)所示.其中,State 0 状态表示输入设备不在其物理跟踪范围之内;State 1 状态表示跟踪设备进入了跟踪范围之内,因此在移动设备的同时会引起屏幕跟踪符号(光标)相应的移动;State 2 状态表示正在利用输入设备对界面对象进行操作.其中,State 1 与 State 2 之间的转换是利用额外的设备按键切换来完成的;当用户释放额外的按键时,系统自动完成从 State 2 到 State 1 之间的切换.VBI 的输入设备为摄像头,但它只是用来捕获人体动作及感知周围环境的变化,并没有实质性地参与交互.因此在交互的过程中,用户无法像操作鼠标或者手写笔等设备那样利用额外的按键切换来完成 State 1 与 State 2 之间的状态转移.因此,Buxton 所提出的基于接触型设备的状态转移模型并不适合于描述具有非接触型特征的 VBI 的交互状态.而迄今为止,从公开发表的文献来看,尚没有一种适合描述非接触型交互设备的状态模型.



(a) Interactive model presented by Buxton<sup>[22]</sup>  
(a) Buxton 提出的交互模型<sup>[22]</sup>



(b) Interactive model presented by this paper  
(b) 本文提出的交互模型

Fig.1 State transaction model

图 1 状态转移模型

本文将手作为一种抽象的输入设备,通过人手的移动模仿鼠标/笔的移动控制屏幕光标的移动,通过用户做出的各种手势事件模仿鼠标/笔的各种按键行为触发不同的系统命令,以此来模仿鼠标/笔对于界面对象的控制及操作过程.因此,在 Buxton 模型的基础上提出了一种可扩展的视觉手势交互状态转移模型,如图 1(b)所示,用来指导交互手势的设计.图中,State 0~State 6 均表示人手这一抽象的“输入设备”在某一系统时刻下的状态.其中,State 0 状态表示用户的手处于摄像头视野范围之外时的 OOR 状态.此时,手无论如何移动都不会对交互产生

影响,界面也没有任何反馈.State 1 为当用户的手移入到摄像头的视野范围但尚未被系统识别的状态.此时,手的移动对交互也没有什么影响,但是界面上应当有所反馈,例如实时显示摄像头所捕获的手的图像.这一反馈能够充分利用用户的前庭感知和运动感知,从而有效地促进交互.State 2 为用户的手被检测出来以后的状态.此时,界面跟踪符号将随着手的移动而移动;当某一特定的手势类型(拾取手势)被识别成功或者根据上下文环境判断满足了一定的拾取条件之后,系统自动转入 State 3 状态;一旦获取了对象的控制权,就可以从 State 3 衍生出许多扩展状态.例如,当检测到 Zoom, Rotate 或 Move 等手势事件后,系统跳转到对界面对象的缩放、旋转、平移状态等.与此相反的过程不再赘述.

一个可扩展的状态转移模型具有以下结构特征:

- (1) 一个有限的离散状态集合  $S_0, S_1, S_2, \dots, S_n$ , 其中,  $S_0$  为一个初始状态.
- (2) 对应于每一个状态  $S_i$ , 定义一套属性和一个函数  $F_i$  用于处理所接收的数据流.
- (3) 对于每一个状态  $S_i$ , 定义一套状态转移规则  $G_{ij}$ , 其中  $j=0, \dots, m \leq n$ , 用于产生不同状态之间的切换.

本文提出的 IEToolkit 内部实现了以 XML 为主的应用语义管理机制,我们将应用程序的内部特征如应用语义处理的相关内容在图 1(b)的状态转移模型中加以实现.设计者可以在各个状态节点上设置语义处理函数,或者通过动态配置状态转移模型来体现应用相关的特征.对于应用系统内部语义的管理,则通过对各个状态节点的属性动态配置而实现.

基于图 1(b)的可扩展的状态转移模型,我们设计并实现了几类通用的视觉交互技术,这些交互技术大都基于一些基本的交互隐喻实现.例如,基于“stretch and squeeze”隐喻<sup>[23]</sup>实现了对对象的缩放技术<sup>[24]</sup>、基于驾驶隐喻的场景漫游技术<sup>[25]</sup>和基于斜率的速度控制技术等(如图 2 所示).每一种隐喻形成一种交互技术的基础心理模型,即,使用这种技术用户可以做什么、不能做什么的直观表现,特定的交互技术可以看作是这些基本隐喻的不同实现.为了便于交互,系统将这些常用的视觉交互技术封装为一系列的界面组件库,主要包括:

- (1) 对象操作器(ObjManipulator):提供了操纵场景对象的通用的接口,封装了一系列的子操作器分别实现不同的功能和操作.目前, IEToolkit 提供了 ObjSelectControl(对象拾取器)、ObjRotateControl(对象旋转器)、ObjZoomControl(对象缩放器)、ObjMoveControl(对象平移器)等基本组件.
- (2) 漫游控制器(NavCtlManipulator):提供了操作场景视点的通用接口,它提供了 ViewTransferControl(视点平移器)、ViewTurnControl(视点旋转器)、ViewPushPullControl(视点拉近/拉远)等子操作器.
- (3) 角色动画控制器(RolCtlManipulator):主要用来控制场景角色的动作,将用户的肢体动作映射为一系列场景角色的相关动作.它充分利用了游戏引擎的事件机制,手势事件被用来触发播放头在不同的帧之间跳转,从而促使游戏角色完成一定的动画情节.

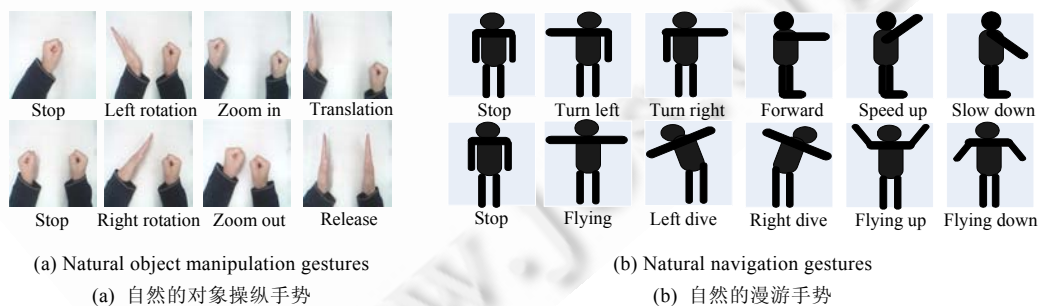


Fig.2 Natural interactive gestures

图 2 自然的交互手势

### 3.2.2 交互意图推理机制

指点和选取是用户界面中最通用的交互技术,同时也是其他交互技术的基础.对于传统的接触型设备来说,可以通过额外的按键消息或者压力感知等交互技术实现选取,而视觉交互却无法将按键消息或压力感知等技

术集成到系统中来。目前主要有两种解决办法:一是通过设计显性的“抓取”手势模仿鼠标/笔的按键消息。该方法的缺点是,由于受各种干扰,手势识别率无法保证 100%准确,因此会出现无法有效完成拾取任务的情况;另一类是隐性的方法,通过判断手势所控制的光标在某对象上的停留时间与系统事先设定的时间阈值  $\tau$  之间的大小关系来确定对象是否被选中。该方法的缺点是,如果  $\tau$  设定过小,则会使系统自动选取用户本无意选择的对象;如果  $\tau$  设置过大,则过长的时间潜伏期容易使用户对操作失去耐心。本文结合上下文感知技术对上述两种方法进行融合,提出一种基于规则的时间推理机制来推断用户的交互意图,从而实现图 1(b)中 State 2-State 3-State 2 的转换。下面给出规则推理机制的具体描述:

给定规则条件集合  $C: \{c_i | c_1 = \text{“不识别状态,即手势识别不成功无返回结果”}; c_2 = \text{“误识别状态,即系统返回的手势识别结果并非是当前所预期的手势”}; c_3 = \text{“正确识别状态,即手势识别成功并且是当前所预期的手势”}; c_4 = \text{“识别时间 } t < \tau \text{”}; c_5 = \text{“识别时间 } t > \tau \text{”}; c_6 = \text{“光标指向界面对象”}; c_7 = \text{“光标不指向界面对象”}\}$  和规则结论集合  $R: \{r_i | r_1 = \text{“选中对象”}; r_2 = \text{“继续识别”}; r_3 = \text{“继续其他操作”}; r_4 = \text{“无操作”}; r_5 = \text{“x”}\}$ ; 其中,“x”表示不会出现的状态;“无操作”表示系统无法猜测用户的交互意图,因此界面不提供任何反馈;“继续其他操作”表示系统将根据上下文环境,响应用户当前的手势,比如漫游等}, 则有规则集合  $G: \{g_{ij} | g_{11} = c_1 \wedge c_5 \wedge c_6 \Rightarrow r_1; g_{12} = c_2 \wedge c_5 \wedge c_6 \Rightarrow r_1; g_{13} = c_3 \wedge c_4 \wedge c_6 \Rightarrow r_1; g_{21} = c_1 \wedge c_4 \wedge c_6 \Rightarrow r_2; g_{22} = c_2 \wedge c_4 \wedge c_6 \Rightarrow r_2; g_{31} = c_2 \wedge c_4 \wedge c_7 \Rightarrow r_3; g_{32} = c_2 \wedge c_5 \wedge c_7 \Rightarrow r_3; g_{41} = c_1 \wedge c_4 \wedge c_7 \Rightarrow r_4; g_{42} = c_1 \wedge c_5 \wedge c_7 \Rightarrow r_4; g_{43} = c_3 \wedge c_4 \wedge c_7 \Rightarrow r_4; g_{44} = c_3 \wedge c_5 \wedge c_7 \Rightarrow r_4; g_{51} = c_3 \wedge c_5 \wedge c_6 \Rightarrow r_5\}$ 。

本文在实践过程中对该推理机制进行了验证,结果表明,它能够保证交互任务的顺利完成并达到很好的效果。该策略具有动态可配置性,即拾取策略和条件可以在系统运行时确定,并可以在任何时候改变策略。例如,可以通过改变时间阈值的大小来调整时间分配机制,适应不同用户的交互习惯。VBI 的隐式交互特征决定了交互任务的生成过程往往伴随着大量的识别技术和上下文感知等技术,为了保证任务完成万无一失,系统将交互意图解析器设计为一个多输入端口节点,方便各种用户干预/修正技术的加入。也就是说,当识别技术和上下文感知技术都失效时,用户还能使用键盘、鼠标或者语音通道对当前产生的交互结果进行修正。

#### 4 工具箱设计

Marr 理论将视觉信息系统看作一组相对独立的功能模块<sup>[26]</sup>,心理学研究也表明,人类使用多种线索或从它们的组合来获得各种视觉信息<sup>[27]</sup>。这启示计算机视觉系统也应该包括许多模块,每个模块获取某一特定的视觉线索进行一定的加工,从而可以根据环境上下文用不同的权系数组合不同的模块来最终完成视觉任务。在 Marr 理论上并结合视觉手势交互机制,本文设计并实现了一个视觉手势界面工具 IEToolkit,如图 3 所示。

传统的 GUI 中,系统接收和处理的是键盘/鼠标的离散输入信息,而 VBI 下跟踪系统输入的是连续的数据流。因此,本文采用数据流图的方法<sup>[28]</sup>建立了支持连续信息输入的数据流模型来表达视觉信息的处理流程,目的是将整体的数据加工过程划分为许多单独的操作步骤并建立起一个有向无环图,用来描述不同操作步骤之间的相互联系。从图 3 可以看出,整个视觉信息的处理过程划分为 4 个不同的处理阶段,每个处理阶段都由一类特定的数据流节点负责完成,它们分别为源节点、过滤器节点、分类器节点和用户界面节点。每一类数据节点又分为许多子节点,每个子节点负责对某一特定的视觉信息进行加工。在数据传输过程中,每一次的变换都可以通过数据流图中的一个节点进行描述。每个节点都有一个输出端口但有一个或者多个输入端口,这种设计方法能够有效解决多元数据处理问题。每个端口都有一条有向边与之相连,节点之间的有向边表示数据的流向,其起始节点称为子节点,终止节点称为父节点。节点一旦通过其某一个输入端口接收到新的数据后立即进行自我更新,并将更新结果通过输出端口发送至父节点。下面对工具箱的各个组成部分进行描述。

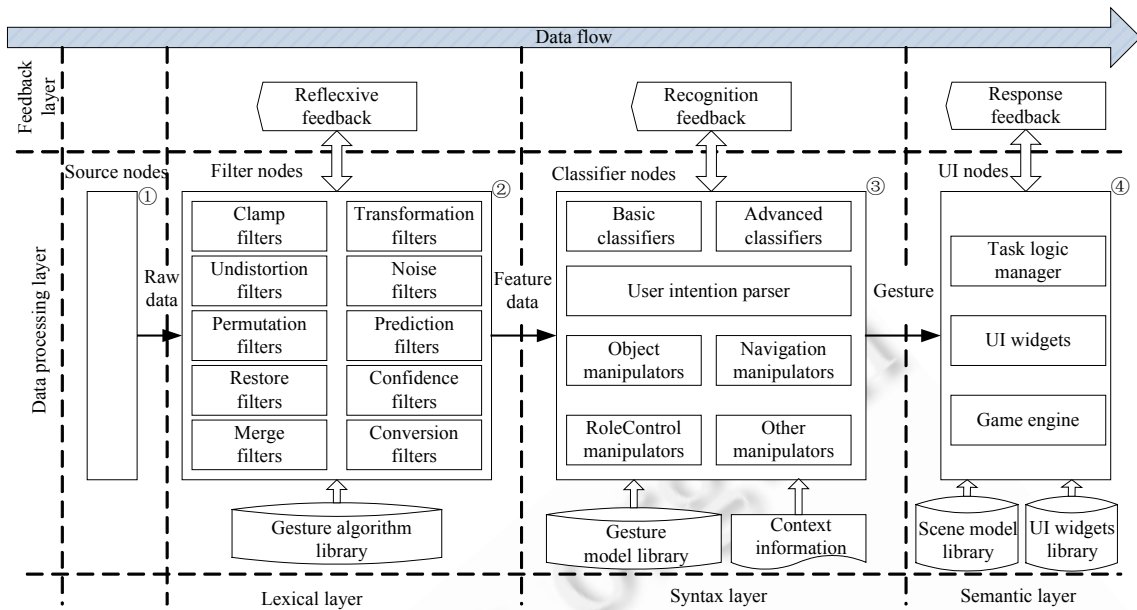


Fig. 3 IEToolkit architecture

图 3 IEToolkit 的体系结构

#### 4.1 数据源节点

源节点是数据流图中的叶子节点,主要负责从外部设备接收数据,可以分为 3 类:第 1 类负责对常见的视频捕获设备驱动进行封装;第 2 类为一些类似于 ARToolkit 的自包含系统提供一个桥连接,以使多种多样的新的识别算法或图像处理技术能够容易地加入;第 3 类预留了鼠标键盘接口,能够接收鼠标/键盘等传统设备的消息,从而使得工具箱能够与传统的 GUI 有效集成.源节点允许用户使用属性页的方法来设置或指定摄像头的驱动、视频捕获图像的尺寸、帧率、相机校正文件的位置等系统配置参数.鉴于视频处理算法的复杂性,为了提高系统的性能,源节点通常以多线程的方式来实现.

#### 4.2 过滤器节点

过滤器节点负责对从数据源节点中接收的数据进行过滤,并实时更新自身的状态,然后将计算结果发送给分类器节点.系统提供了可扩展的接口,用户可以方便地添加各种过滤器,并对各个过滤器进行统一的管理和调度.这些过滤器可以支持不同的图像处理集合,整个图像处理的过程就是这些过滤器的并集,而它们在系统流程中的顺序则说明了它们在数据流处理过程中的优先权.通过调整优先权、禁止或者开启某一过滤器,可以做到数据表示与处理的动态绑定.目前,系统提供的过滤器节点包括夹子过滤器、几何变换过滤器、校正过滤器、噪声过滤器、置换过滤器、预测过滤器、修复过滤器、置信过滤器、融合过滤器、转换过滤器等.

#### 4.3 分类器节点

分类器节点负责对特征数据进行识别和高层的处理与管理,并将处理结果发送给用户界面节点完成交互任务.系统采用了一种可视化的基于样本的方法<sup>[13]</sup>,供非计算机视觉用户学习不同种类的分类器.整个分类器的设计思想是基于一种“paint-view-correct”<sup>[12]</sup>的交互式学习方法.

用户首先在系统提供的具有回放功能的视频编辑窗口中对一段视频帧序列进行操作,提取出感兴趣的学习样本,然后从分类器列表中指定本次学习所使用的分类器.对于一些简单的对象识别,系统提供了颜色分类器、形状分类器、亮度分类器等基本分类器;对于一些较为复杂的特征识别,例如人脸识别,可以使用 Adaboost, SIFT, Motion 等高级分类器.鉴于目前大多数的互动游戏都从用户的双手中提取重要的交互输入信息,单纯的颜

色等基本分类器不能得到满意的效果.因此,系统专门提供了一类手势分类器.手势分类器中封装了常见的静态和动态手势识别算法<sup>[29,30]</sup>,能够满足大多数的应用需求.在选取了分类器和合适的样本之后,用户就可以对当前的分类器进行训练,并通过拖动视频编辑窗口中的滑动条来实时查看训练结果.如果不满意,则用户可以返回编辑状态提取更多的样本用来学习,或者选择其他分类器或者使用不同的分类器的组合来获取更好的效果.通过这种快速的迭代方式,用户能够在短时间内训练出有效的分类器,并手工标定需要输出的信息,例如被检测对象的数目、对象的最小包围盒、对象的质心位置、对象的倾斜角度等,以及数据的保存形式例如通过 TCP 向外部输出 XML 文件并传递给具体的应用.

与传统的工具箱 HandVu<sup>[17]</sup>,GT<sup>2k</sup><sup>[18]</sup>等相比,系统屏蔽了具体的图像处理、机器学习算法等技术细节,用户所关心的仅仅是如何通过实时的效果验证和迭代式的方法来选取能够解决当前问题的最优分类器或者分类器的组合.在实现上,系统借鉴了 MediaPlayer 的插件思想(plug-in architecture),提供了可扩展的接口,方便开发人员加入新的分类器.系统本身实现了对不同分类器的统一管理,用户可以对这些分类器进行动态的配置,例如,禁止或者开启某一分类器,用于完成不同的识别任务.

#### 4.4 用户界面节点

基于视觉的互动游戏最近几年有了飞速的发展,并出现了不少成功的商业系统,例如 Sony Eyetoy<sup>[31]</sup>等.纵观大多数的视觉游戏界面节点,其不外乎分为两大类.一类是类似于传统 GUI 的一组预定义的界面节点,我们称其为热点或者热区(hotspot).用户可以通过身体的运动触发这些热区从而得到一系列的反馈,常见的如拳击游戏、跳舞游戏、体育运动游戏、棋盘类游戏等.另一类是场景节点,我们称其为化身(avatar).系统可以将用户身体动作直接映射为化身的一系列动作,例如,通过身体移动驱动化身走路、跳跃、飞行等.在基于 GUI 的游戏界面中,用户主要通过鼠标、键盘等接触型输入设备来显性地点击这些界面节点获得反馈.在 VBI 中,用户无法通过显性的点击操作来完成交互任务.在 IEToolkit 中,充分运用了本文第 3 节中所讨论的基于 VBI 的视觉交互机制来支持各种不同的交互任务的有效完成.

任务逻辑管理器是分类器节点与界面节点之间的桥梁,它根据具体的任务模型<sup>[24]</sup>为分类器节点所输出的手势事件与交互任务之间建立起了多种逻辑映射关系,包括一对一映射、一对多映射、多对一映射、多对多映射 4 种情况.通过任务映射之后,avatar 在自身逻辑里处理各种事件.每个 avatar 自身保存一定数量的任务体和一定数量的任务变量.每个 avatar 可以开启哪些任务以及开启任务的条件,需要事先编辑到场景数据文件中.当特定的交互事件发生时,首先查询场景数据文件,看是否开启一个新的任务,然后通知现有任务的脚本,由脚本来执行各自独立的任务逻辑.目前,大多数的游戏引擎都支持通过配置文件来处理问题,这种数据驱动方式不但使得编辑逻辑更加简单,并能使得游戏引擎与工具箱之间通过定义好的配置文件进行有效集成.

#### 4.5 IEToolkit的反馈机制

界面反馈是 HCI 的重要环节<sup>[32]</sup>.它是指传递给用户、帮助用户理解系统状态的有效信息.它使得人机交互的输入和输出之间形成一个不断循环的交互闭环系统.VBI 是一种非接触型的连续交互界面,传统 GUI 的反馈机制已不再适合于 VBI.因此,设计良好的适合 VBI 特征的界面反馈机制显得尤为重要,它将直接影响到交互的成功与否,并有助于不断提高交互的效率.本文针对视觉数据流的 3 个不同的处理阶段设计了 3 种不同的界面反馈形式:

- (1) 反身反馈(reflexive feedback):VBI 的初级界面反馈形式,系统将交互过程中的被跟踪对象绘制出来,为用户提供一种对象自跟踪的视觉呈现机制.它能够提供给用户其自身运动信息的可视化表现,有助于用户获取摄像头交互区域的空间感知.反身反馈通常用来提供一些预识别的信息,根据不同的交互环境(上下文)有不同的实现方法,例如,向用户实时展示背景差分后的手部分割结果.
- (2) 识别反馈(recognition feedback):主要发生于图像处理的后阶段,用于给用户提供一种机制来表明究竟哪个手势被识别成功的信息,类似于传统 GUI 下弹出式窗口向用户确认一个命令的执行情况.
- (3) 系统反馈(response feedback):一种高级的反馈形式,是指交互操作的最终响应情况.VBI 是一种非接触



型交互,用户虽然获得了更大的交互空间,但同时也失去了重要的实物感知线索.因此,在 IEToolkit 中除了视觉反馈以外,本文还增加了声音反馈机制,例如,将跟踪对象的方向和速度映射为一定的音量、音阶、音色等,以丰富用户的前庭感知和运动感知.

## 5 开发过程及应用实例

### 5.1 基于 IEToolkit 的开发过程

IEToolkit 的目标是提供一个便利的视觉工具箱,以便能够很容易地集成到互动娱乐游戏设计应用中,方便游戏开发人员迅速地将他们的设计转换为游戏作品.图 4 给出了一个基于 IEToolkit 的游戏开发实现过程,图中纵向上分为两层,下层为 IEToolkit 的用户训练过程(training process),上层为应用 IEToolkit 时的系统实际运行过程(run-time process).

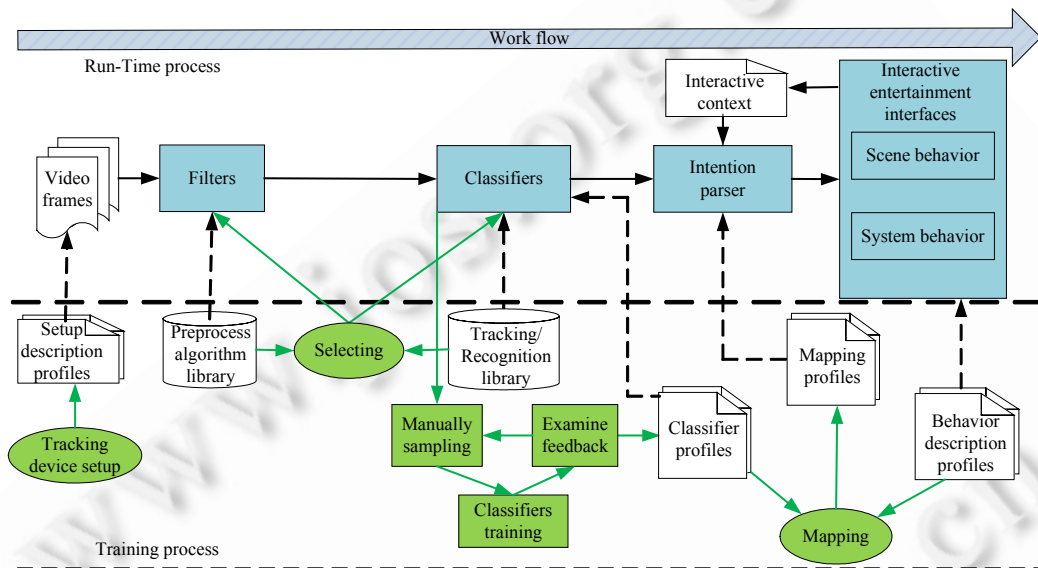


Fig.4 IEToolkit-Based process of interactive entertainment system development

图 4 基于 IEToolkit 的互动娱乐系统开发过程

下面给出一个通用的软件系统开发步骤:

步骤 1. 跟踪设备的配置.根据具体应用的功能以及界面需求,确定摄像头的驱动、视频捕获图像的尺寸、帧率、相机校正文件的位置等系统配置参数,生成基于 XML 的跟踪配置描述文件.

步骤 2. 原始数据预处理.根据步骤 1 生成的跟踪设备配置描述以及具体的应用需求,选取适当的过滤器或过滤器集合,对从设备驱动中所读取的原始数据进行预处理.

步骤 3. 分类器学习.打开一段已存的视频帧序列,或者通过摄像头现场录制一段视频,从中手工标定出待识别对象比如人脸或人手等并生成学习样本.使用不同的分类器将需要生成不同种类的样本,比如,仅仅使用一般的颜色分类器或者亮度分类器则只需生成正样本,而如果使用特征分类器,例如 Adaboost,则同时需要生成一定数量的正样本和负样本.用户可以通过实时检查训练学习结果来决定是否结束分类器学习过程,如果对学习结果不满意可以重新选取新的样本加入到样本集合,或者尝试其他分类器甚至使用多个分类器一起工作.以上学习过程是一个迭代渐进的过程,通过对当前反馈结果的实时修正,可以使用户在短时间内迅速掌握选取与应用领域最为相关的样本以及最为合适的分类器组合策略等技巧.通过这种学习策略,有助于用户迅速了解哪些分类器最适合于哪些应用领域,从而帮助他们建立起一种本能的直觉,从而在以后的开发过程中大幅度缩减训

练时间,提高开发效率<sup>[13,16]</sup>.这一步骤的训练结束后,系统将自动生成学习结果库文件,该库文件将在系统实际运行过程与分类器一起作为视频跟踪或识别的重要依据.

步骤 4. 交互任务关联.在步骤 3 的基础上,用户根据第 4.4 节提出的交互任务映射机制,并根据交互任务模型<sup>[24]</sup>适当选取前面第 3.2.1 节中提出的不同的交互操作器,在其中为手势事件与场景对象、场景行为或者系统行为之间建立起有效的映射关系.系统为这一步的映射结果生成基于 XML 的关联文件,在系统实际运行时,用户意图解析器将根据该关联文件并结合上下文反馈信息完成用户交互意图的解析或自动感知,从而完成一系列交互任务.

## 5.2 应用实例与评估

我们的目的是将 IEToolkit 应用于互动娱乐领域,使开发人员能够快速构建基于视觉界面的互动娱乐原型系统.互动娱乐环境具有虚实叠加交互、多显示设备并存等特点,工具箱必须能够为不同的系统配置提供统一的支持.图 5(a)是基于 IEToolkit 开发的一个基于视觉手势的虚拟家居展示系统,与键盘鼠标等其他交互方式相比,视频手势交互技术使得用户的操作自由度更大,其自然、直接、高效的特点很适合于家居展示.

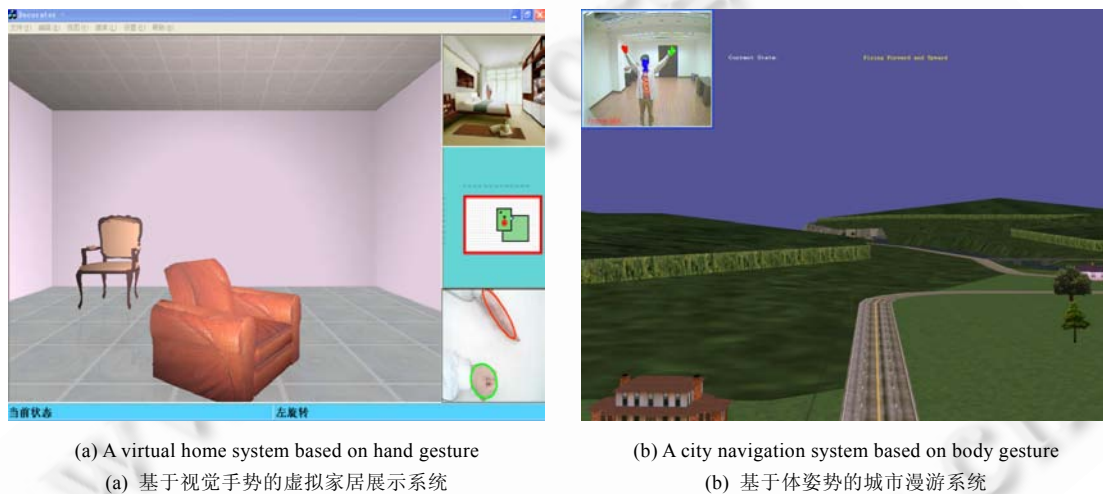


Fig.5 Prototype systems based on IEToolkit

图 5 基于 IEToolkit 开发的原型系统

为了给用户提供一种新的交互体验,我们使用普通网络摄像头来实时捕获用户的手部动作.然后,系统通过一定的交互机制映射为不同的交互行为与场景互动,从而完成场景漫游和家具操作等不同的交互任务.该实例的开发过程完全是基于第 4.1 节中所描述的软件开发流程完成的.图 6 为 IEToolkit 的运行界面,其中,界面最上方提供了菜单栏,用户可以在弹出的跟踪配置属性页对话框中完成数据源的配置以及各种不同预处理过滤器的选择与配置工作.界面的主操作区主要分为 3 部分:

- 左侧面板为样本编辑及效果预览区,其中,左上角为样本编辑窗口,用户可以利用鼠标或手写笔等设备在视频流的帧间选取不同的手势样本;左下角为应用了分类器后的识别效果预览.
- 界面中间面板为用户学习区,其中,上半部分显示用户收集的样本缩略图,下半部分为系统提供的分类器列表.用户可以根据不同的需求选择不同的分类器,或者利用不同的分类器组合更好地完成训练任务.
- 界面右侧面板为应用配置区,开发人员可以根据具体的应用任务需求对分类器的输出结果进行配置,并且根据应用上下文添加各种约束条件以及为手势添加高层的交互语义,从而为手势与场景/系统行为之间建立起一定的映射关系.

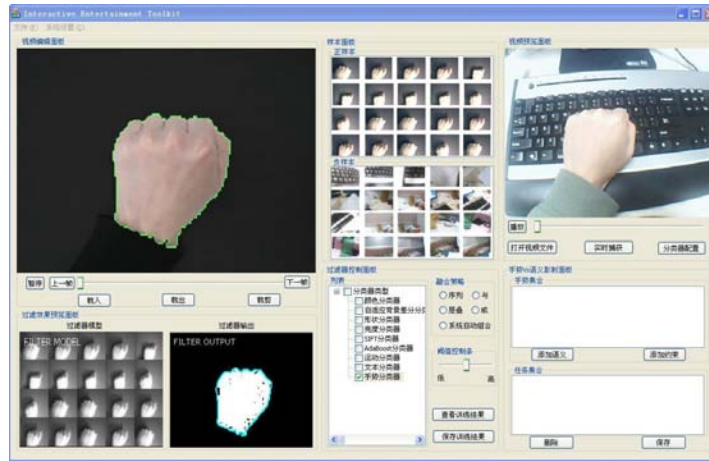
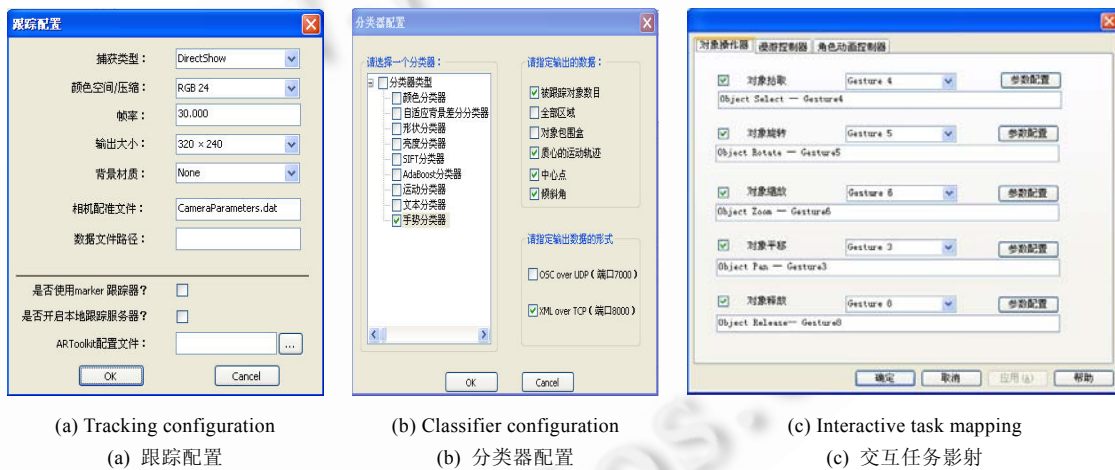


Fig.6 IEToolkit run-time interface  
图 6 IEToolkit 运行界面

下面简单介绍该应用实例的开发流程:

步骤 1. 首先是对跟踪设备的配置,设计开发人员利用 IEToolkit 提供的属性页来对跟踪设备进行设置,例如定制实时视频捕获相机的参数(驱动、大小、帧率、相机校正的数据文件)、应用是否需要使用 marker 跟踪等.图 7(a)给出了一个工具箱属性页配置界面.



(a) Tracking configuration  
(a) 跟踪配置

(b) Classifier configuration  
(b) 分类器配置

(c) Interactive task mapping  
(c) 交互任务映射

Fig.7 System configuration interfaces  
图 7 系统配置界面

步骤 2. 其次是原始数据预处理过程.颜色跟踪是一种简单而有效的跟踪方式.它能够根据一定的色度信息(肤色信息)来过滤图像,并且快速、精确地提取图像中与该色度相匹配的部分(手部区域).但是,颜色过滤受光照条件的影响比较大,一旦环境光照剧烈变化,其过滤性能则将大为降低.形状过滤能够有效地提取出独立于颜色信息的轮廓、大小、范围等信息,并且形状过滤不易受环境光照的影响.但是,形状过滤通常夹杂着很多噪声,因此导致过滤结果精度不高.为了得到更稳定的结果,我们使用置信过滤器对颜色与形状两类数据信息的有效性进行启发式的估计,然后使用融合过滤器依据上述估计结果运用先验概率将形状过滤和颜色过滤这两类数据结果进行融合.图 8 给出了颜色与轮廓进行数据流融合的示意图.

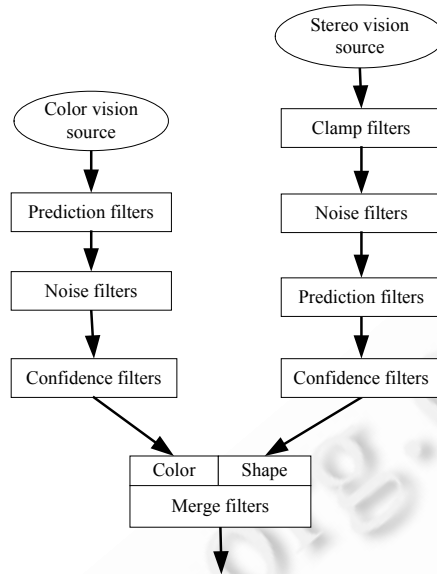


Fig.8 Data-Flow merging

图 8 数据流融合

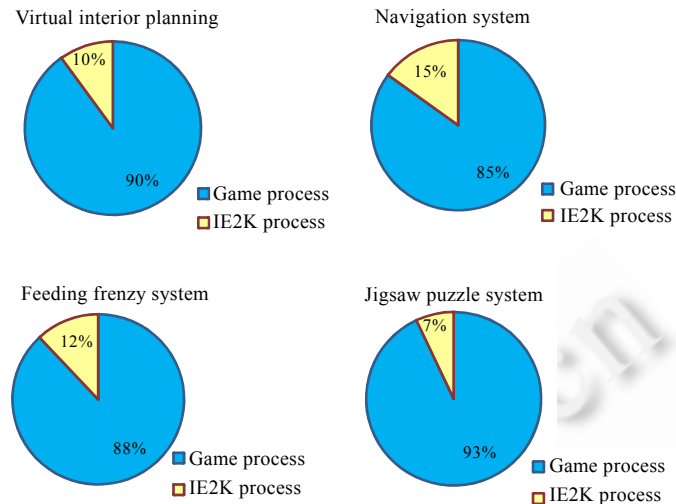
步骤 3. 然后是分类器训练学习过程,在系统中,我们定义了两大类共 10 种手势,包括 5 种漫游类手势和 5 种家具操作类手势.用户需要对 10 种手势使用分类器学习训练,然后结合训练结果对这 10 种手势进行在线识别.使用静态手势分类器对 10 种手势的平均识别率达到了 94%以上.

步骤 4. 最后是交互任务映射过程,我们利用步骤 3 的训练结果对分类器输出结果进行配置(如图 7(b)所示),将带有质心位置、放缩系数、旋转角度、时间戳等参数的手势与两类家居展示任务建立关联,图 7(c)给出了完成交互任务映射的操作器界面.

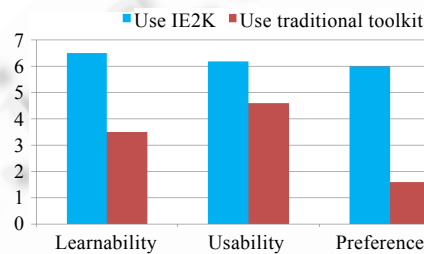
利用 IEToolkit,我们还开发构建了各种其他基于视觉的互动娱乐应用系统.例如,在 OSG 平台上构造了基于视觉的体姿态城市漫游系统(如图 5(b)所示),基于 Flash 平台构造了基于视觉手势的吞食鱼游戏、七巧板游戏等视觉游戏套件.

我们就不同的开发方案进行了初步的用户评估,如图 9 所示.可以看出,评估者对不同开发方案的易学性、易用性及交互方式的偏好等可用性评测结果有很大的差异,达到了显著性水平.使用 IEToolkit,用户开发的大部分工作都集中在系统界面以及游戏情节设计上,视觉开发工作量所占总的开发工作量的比重不足 20%,极大地提高了基于视觉界面的互动娱乐原型系统开发的效率,加快了原型开发和评估过程的迭代速度.

系统开发实例与评估结果表明,IEToolkit 实现了对视频数据的统一处理,如数据提取、数据过滤、特征分析与识别,并且提供了良好的交互机制.它使用了基于 XML 的软件工程方法,带有可视化图形用户界面的 XML 编辑器,使得游戏开发人员不必掌握复杂的语法结构和内部实现,就能方便、灵活地使用通用的 XML 工具进行软件开发、文档撰写、系统集成和功能配置等工作.上述特点使得 IEToolkit 有效地降低了基于视觉界面的交互系统开发门槛,普通开发者在利用 IEToolkit 工具箱开发构建互动娱乐领域的各种应用时,无须关注各种图像处理 and 机器学习等技术的实现细节,无须具备计算机视觉领域的专家经验,因而可以为非专家用户开发领域应用提供一种统一的解决方案.



(a) Distribution of workload  
(a) 工作量分布



(b) Subjective ratings  
(b) 主观评分

Fig.9 Experimental results

图9 实验结果

## 6 结论和展望

VBI 提供给用户更加自然的交互方式,但是基于 VBI 的系统的构造是一项非常困难的工作.本文总结了目前存在的各种开发工具箱的优缺点,针对目前存在的大多数视觉工具箱复杂、难用的普遍问题,针对视觉交互的特点,面向互动娱乐领域设计开发了一个简单、易用、可扩展的工具箱系统 IEToolkit.它包含了构造一个基于 VBI 的交互系统所需要的方方面面,尝试了各种新的技术,力求为界面开发者提供有力的支持.基于这个工具箱,系统开发者可以构建不同的视觉娱乐游戏应用.本文从事件模型、交互模型、数据流模型等几个方面对工具箱的组成结构进行了描述,并给出了一个通用的基于 IEToolkit 的原型系统开发流程.开发人员可以将更多的精力集中在具体的高层逻辑语义处理上,而无须过多地考虑底层的技术细节与支撑结构.本文最后的应用实例及评估结果表明,IEToolkit 能够较好地支持基于视觉交互的应用系统的快速构造.下一步的工作是通过更加严格、深入的用户评估实验和迭代式的原型开发过程,在实践中进一步检验和完善 IEToolkit 的功能.

### References:

- [1] Turk M. Moving from GUIs to PUIs. Technical Report, MSR-TR-98-69, Redmond: Microsoft Corporation, 1998.1-7.
- [2] Maes P, Blumberg B, Darrell T, Pentland A. The alive system: Full-Body interaction with animated autonomous agents. ACM Multimedia Systems, 1997,5(2):105-112. [doi: 10.1007/s005300050046]

- [3] Wren C, Azarbayejani A, Darrel T, Pentland A. Pfunder: Real-Time tracking of the human body. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1997,19(7):780–785. [doi: 10.1109/34.598236]
- [4] Wren C, Sparacino F, Azarbayejani A, Darrell T, Starner T. Perceptive spaces for performance and entertainment: Untethered interaction using computer vision and audition. *Applied Artificial Intelligence*, 1997,11(4):267–284. [doi: 10.1080/088395197118154]
- [5] Freeman WT, Anderson DB, Beardsley PA, Dodge CN, Roth M, Weissman CD, Yerazunis WS, Kage H, Kyuma K, Miyake Y, Tanaka K. Computer vision for interactive computer graphics. *IEEE Computer Graphics and Applications*, 1998,18(3):42–53. [doi: 10.1109/38.674971]
- [6] Oliver N, Pentland A, Berard F. LAFTER: A real-time face and lips tracker with facial expression recognition. *Pattern Recognition*, 2000,33(8):1369–1382. [doi: 10.1016/S0031-3203(99)00113-2]
- [7] Betke M, Gips J, Fleming P. The camera mouse: Visual tracking of body features to provide computer access for people with severe disabilities. *IEEE Trans. on Neural and Rehabilitation Engineering*, 2002,10(1):1–10. [doi: 10.1109/TNSRE.2002.1021581]
- [8] Chao H, Meng MQ, Liu PX, Xiang W. Visual gesture recognition for human-machine interface of robot teleoperation. In: Lee CSG, ed. *Proc. of the 2003 IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IEEE RSJ ICIRS 2003)*. Las Vegas, Institute of Electrical and Electronics Engineer, 2003. 1560–1565. [doi: 10.1109/IROS.2003.1248866]
- [9] Larssen AT, Loke L, Robertson T, Edwards J. Understanding movement as input for interaction—A study of two EyeToy™ games. In: Hyland P, Vrazalic L, eds. *Proc. of the Computer-Human Interaction Special Interest Group of Australia (OZCHI 2004)*. New York: ACM Press, 2004.
- [10] Höysniemi J, Hämäläinen P, Turkki L, Rouvi T. Children's intuitive gestures in vision based action games. *Communications of the ACM*, 2005,48(1):45–52. [doi: 10.1145/1096000.1096003]
- [11] Deng Y, Li ZB, Li H. Design and realization of vision-based 3D human body motion tracking system. *Journal of Computer-Aided Design & Computer Graphics*, 2007,19(6):769–780 (in Chinese with English abstract).
- [12] Fails J, Olsen D. A design tool for camera-based interaction. In: Cockton G, ed. *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI 2003)*. New York: ACM Press, 2003. 449–456. [doi: 10.1145/642611.642690]
- [13] Maynes-Aminzade D, Winograd T, Igarashi T. Eyepatch: Prototyping camera-based interaction through examples. In: Shen C, ed. *Proc. of the 20th Annual ACM Symp. on User Interface Software and Technology (UIST 2007)*. New York: ACM Press, 2007. 33–42. [doi: 10.1145/1294211.1294219]
- [14] Fogarty J, Hudson SE. Toolkit support for developing and deploying sensor-based statistical models of human situations. In: Rosson MB, ed. *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI 2007)*. New York: ACM Press, 2007. 135–144. [doi: 10.1145/1240624.1240645]
- [15] Hartmann B, Abdulla L, Mittal M, Klemmer SR. Authoring sensor-based interactions by demonstration with direct manipulation and pattern recognition. In: Rosson MB, ed. *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI 2007)*. New York: ACM Press, 2007. 145–154. [doi: 10.1145/1240624.1240646]
- [16] Patel K, Fogarty J, Landay JA, Harrison B. Investigating statistical machine learning as a tool for software development. In: Czerwinski M, ed. *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI 2008)*. New York: ACM Press, 2007. 667–676. [doi: 10.1145/1357054.1357160]
- [17] Kolsch M, Turk M, Hollerer T. Vision-Based interfaces for mobility. In: Ley M, ed. *Proc. of the 1st Annual Int'l Conf. on Mobile and Ubiquitous Systems: Networking and Services (MOBIQUITOUS 2004)*. Washington: IEEE Computer Society Press, 2004. 86–94. [doi: 10.1109/MOBIQ.2004.1331713]
- [18] Westeyn T, Brashear H, Atrash A, Starner T. Georgia Tech gesture toolkit: Supporting experiments in gesture recognition. In: Oviatt S, ed. *Proc. of the 5th Int'l Conf. on Multimodal Interfaces (ICMI 2003)*. New York: ACM Press, 2003. 85–92. [doi: 10.1145/958432.958452]
- [19] Bagnall B. *Core Lego Mindstorms Programming*. Upper Saddle River: Prentice Hall, Inc., 2002. 1–560.
- [20] Diaz-Marino R, Greenberg S. CAMBIENCE: A video-driven sonic ecology for media spaces. In: Hinds P, ed. *Video Proc. of ACM CSCW 2006 Conf. on Computer Supported Cooperative Work (CSCW 2006)*. New York: ACM Press, 2006.
- [21] Kato H, Billinghamurst M. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: IWAR, ed. *Proc. of the 2nd IEEE and ACM Int'l Workshop on Augmented Reality*. Washington: IEEE Computer Society, 1999. 85–94. [doi: 10.1109/IWAR.1999.803809]
- [22] Buxton W. A three-state model of graphical input. In: Diaper D, ed. *Proc. of the Human-Computer Interaction (INTERACT'90)*.

- Amsterdam: Elsevier Science Publishers B.V., 1990. 449–456.
- [23] Hinckley K, Czerwinski M, Sinclair M. Interaction and modeling techniques for desktop two-handed input. In: Mynatt E, ed. Proc. of the 11th Annual ACM Symp. on User Interface Software and Technology (UIST'98). New York: ACM Press, 1998. 49–58. [doi: 10.1145/288392.288572]
- [24] Wu HY, Zhang FJ, Dai GZ. UIDT: A vision based user interface model. Journal of Computer-Aided Design & Computer Graphics, 2008,20(6):781–786 (in Chinese with English abstract).
- [25] Bowman DA, Kruijff E, Laviola JJ, Poupyrev I. 3D User Interfaces: Theory and Practice. Boston: Addison-Wesley Professional, 2004. 111–134.
- [26] Marr D. Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. San Francisco: W.H. Freeman, 1982. 1–397.
- [27] Jeremy MW, Keith RK, Dennis ML, Linda MB, Rachel SH, Roberta LK, Susan JL, Daniel MM. Sensation and Perception. 2nd ed., Sunderland: Sinauer Associates, 2008. 1–415.
- [28] Reitmayr G, Schmalstieg D. An open software architecture for virtual reality interaction. In: Shaw C, ed. Proc. of the ACM Symp. on Virtual reality software and Technology (VRST 2001). New York: ACM Press, 2001. 47–54. [doi: 10.1145/505008.505018]
- [29] Zhu JY, Wang XY, Wang WX, Dai GZ. Hand gesture recognition based on structure analysis. Chinese Journal of Computers, 2006, 29(12):2130–2137 (in Chinese with English abstract).
- [30] Wang XY, Dai GZ, Zhang XW, Zhang FJ. Recognition of complex dynamic gesture based on HMM-FNN model. Journal of Software, 2008,18(9):2302–2312 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/18/2302.htm> [doi: 10.3724/SP.J.1001.2008.02302]
- [31] Sony eye-toy. <http://www.eyetoy.com>
- [32] Dong SH, Wang J, Dai GZ. Human-Computer Interaction and Multi Model User Interface. Beijing: Science Press, 1999. 145–159 (in Chinese).

#### 附中文参考文献:

- [11] 邓宇,李振波,李华.基于视频的三维人体运动跟踪系统的设计与实现.计算机辅助设计与图形学学报,2007,19(6):769–780.
- [24] 武汇岳,张凤军,戴国忠.UIDT:一种基于摄像头的用户界面模型.计算机辅助设计与图形学学报,2008,20(6):781–786.
- [29] 朱继玉,王西颖,王威信,戴国忠.基于结构分析的手势识别.计算机学报,2006,29(12):2130–2137.
- [30] 王西颖,戴国忠,张习文,张凤军.基于 HMM-FNN 的复杂动态手势识别.软件学报,2008,18(9):2302–2312. <http://www.jos.org.cn/1000-9825/18/2302.htm> [doi: 10.3724/SP.J.1001.2008.02302]
- [32] 董士海,王坚,戴国忠.人机交互和多通道用户界面.北京:科学出版社,1999.145–159.



武汇岳(1979—),男,山东烟台人,博士,讲师,CCF 会员,主要研究领域为人机交互,用户界面.



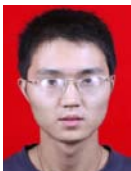
胡银焕(1984—),女,硕士,主要研究领域为计算机视觉.



张凤军(1971—),男,博士,副研究员,主要研究领域为人机交互,虚拟现实,CAD.



戴国忠(1944—),男,研究员,博士生导师,CCF 高级会员,主要研究领域为人机交互,计算机图形学.



刘玉进(1984—),男,硕士,主要研究领域为计算机视觉.