

Jogos Web: O Reconhecimento de Imagens como meio de Interação

Thatiane Benz Pisco Leandro Ramos da Silva*

Faculdade de Tecnologia de São Caetano do Sul, Tecnologia em Jogos Digitais, Brasil

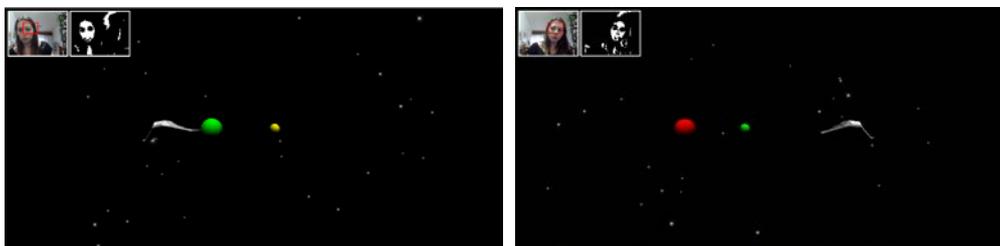


Figura 1: Protótipo de interação em ambiente web utilizando o reconhecimento de objetos

Resumo

Com a evolução tecnológica dos navegadores web, torna-se possível a exploração de novos tipos de interação para o meio. Baseado no conceito de flow de Csikszentmihalyi, importante para a criação de atividades imersivas, surgem a cada momento novos modos de comunicação entre o mundo virtual e o real visando sempre facilitar o processo cognitivo do jogador, aumentando, desta forma, seu estado de flow. Este trabalho, portanto, estudou a evolução das interfaces nos jogos digitais e propôs uma alternativa de interação para jogos web, utilizando o reconhecimento de objetos com o uso dos modelos de cores RGB, HSV e YCbCr e da linguagem JavaScript, que ganha força com o surgimento do HTML5 e da API webRTC, proporcionando uma experiência de usuário gratificante em um meio de fácil acesso.

Palavras-chave: flow, HTML5, jogos digitais, modelo de cores, reconhecimento de objetos, webRTC.

Contatos:

{Thatiane}thatianepisco@gmail.com.br
*lrsilva@fateccscs.edu.br (orientador)

Abstract

With the technological evolution of web browsers, new types of interfaces are possible to explore. Based on the concept of flow developed by Csikszentmihalyi, which is important to the creation of immersive activities, new kinds of communication interfaces, between the virtual and real world, are developed aiming to facilitate the cognitive process of the player thus amplifying their state of flow.

This work, therefore, studied the evolution of interfaces in digital games and proposed an alternative interaction for web games, using object recognition with the use of color models such as RGB, HSV and YCbCr and the JavaScript language, which is strengthened by the arise of HTML5 WebRTC API, providing a satisfying user experience in a accessible way.

Keywords: color models, digital games, flow, HTML5, object recognition, webRTC.

1. Introdução

O *designer* de interação é responsável por desenvolver sistemas interativos que possibilitem experiências positivas para seus usuários. As evoluções tecnológicas, como os mecanismos de reconhecimento de fala e de gestos, permitiram o desenvolvimento de outros tipos de interações ampliando as possibilidades de experiência do usuário.

Conceitos como o de realidade aumentada estão cada vez mais próximos da vida cotidiana. Consoles que dispõem de reconhecimento gestual ganham mercado e trazem “não jogadores” ao universo dos *games* por apresentarem processos cognitivos intuitivos.

Muitos jogos apresentam a necessidade de um gasto excessivo de tempo para aprender a utilizar-se de seu modo de interação. Para jogadores casuais e que não possuem os jogos como atividade marcante em suas vidas, este fato vira um motivo desencorajador para participar deste tipo de atividade.

Com o avanço da tecnologia, o *designer* de interação pode se utilizar de recursos como o reconhecimento da fala e de gestos para o desenvolvimento de novos processos interativos que permitem amplificar o processo de experiência do usuário.

Este estudo, portanto, se fundamentou nas teorias “*Primitive Actions*” (ações primitivas) usadas por Gregerson e Grodal, termo usado para definir a ação real que corresponde a uma ação dentro do mundo simulado, e no conceito de *flow* de Csikszentmihalyi, para estudo de quais fatores ajudam uma atividade a gerar experiências gratificantes, pesquisando sobre novas tecnologias web vistas no HTML 5 e sobre o uso de modelos de cores no reconhecimento de objetos para utilizar estes conceitos no desenvolvimento de uma nova possibilidade de interação no ambiente web por meio da webcam.

2. Trabalhos relacionados

Nos últimos anos surgiram trabalhos propondo novas formas de interações em jogos digitais. Dentre eles pode-se citar o jogo de luta “Camera Kombat” que teve como objetivo o uso de uma interação livre entre máquina e jogador sem a utilização de dispositivos como intermediários. [Paula et al. 2006]

Com o auxílio do reconhecimento de objetos e de uma webcam, Camera Kombat utiliza-se do próprio corpo do indivíduo para interagir com o mundo virtual em uma interação sem dependências na qual a atividade física é proporcionada gerando benefícios ao jogador.

Outro trabalho que merece destaque é “Visão Computacional e Reconhecimento de Comandos de Voz Aplicados na Interação com Jogos e Ambientes 3D”, que apresenta um novo sistema de interação entre o mundo virtual tridimensional e o real com o uso de visão computacional e comandos de voz. [Lima et al. 2010]

Neste estudo foi desenvolvido um protótipo de um jogo de RPG que utiliza o movimento da face do jogador para controlar as ações do personagem em cena. Além disso, é utilizado comandos de voz para interagir com outros personagens, como por exemplo, o comando “oi” para iniciar uma conversa.

Em 2006, Jungles e Battaiola, em seu trabalho “Análise de Tecnologias para a Implementação de Jogos Web”, já explicavam sobre a relevância de produzir entretenimento com o uso de tecnologias web possibilitando aos utilizadores destas aplicações a vantagem de acesso em qualquer lugar do mundo. Com o crescente acesso à internet e com a redução de preços dos hardwares para gráficos 3D, este desenvolvimento foi impulsionado. [Jungles e Battaiola 2006]

Desta forma, surgem trabalhos que utilizam da web e do reconhecimento de objetos, como por exemplo, como o desenvolvido no artigo “Development of an interactive game using a webcam”, que se utiliza da web, webcam e reconhecimento de cores para criação de novos tipos de interação, trazendo para os jogadores uma forma intuitiva e de fácil acesso para se comunicar com o meio virtual. [Souza Jr. et al. 2009]

A proposta deste trabalho, portanto, é continuar a busca por novas alternativas de interações que possibilitem experiências gratificantes, baseadas no flow de Csikszentmihalyi [Csikszentmihalyi 1990], e em ambientes web com a utilização de novas tecnologias, como o HTML5 e o webRTC, simplificando o desenvolvimento e ampliando a experiência do indivíduo com o uso de interfaces intuitivas, que necessitam de uma nova abordagem da relação entre cognição, corpo e gesto, como abordado no trabalho “Dedões - Corpo – Gestos: A inteligência corporal no design dos jogos digitais” de Edson P. Pfitzenreuter e Roger Tavares. [Pfitzenreuter e Tavares 2010]

3. A evolução cognitiva

3.1. O conceito flow

Nos jogos digitais é de extrema importância que haja a garantia de que os jogadores se divirtam. Para isto, podemos nos apoiar em estudos que nos mostram o uso de regras de usabilidade e de experiência do usuário para garantir jogos e aplicações mais eficazes.

Csikszentmihalyi, estudioso de psicologia, fez uma série de pesquisas e entrevistas para entender e buscar o que conduzia uma experiência a ser agradável e divertida. Ele iniciou o processo com pessoas que gastavam muito tempo em atividades que não lhe traziam recompensas externas. Ao fim, ele descobriu que o modo com que as pessoas descreviam o quão divertido eram suas atividades eram parecidas independente de classe social, idade e gênero.

Csikszentmihalyi diz que estas experiências “são tão gratificantes que as pessoas estão dispostas a fazê-las pelo seu próprio bem, com pouca preocupação com o que eles vão obter com isso, mesmo se é difícil ou perigoso” [Csikszentmihalyi 1990] e definiu que elas baseiam-se em oito elementos que juntos causam uma sensação de diversão profunda. São estes: completar uma tarefa; concentração; objetivos claros; *feedback* imediato; sensação de controle sobre as ações; profundidade de envolvimento que remove as preocupações com problemas da vida cotidiana; sentido de duração do tempo alterada; e a diminuição da preocupação exterior ao longo da atividade com uma imersão mais forte depois.

Estas atividades são autotélicas, ou seja, possuem o fim nelas mesmo sem gerar recompensas externas e dão a sensação de descobertas e de novas realidades. Csikszentmihalyi denomina estas gratificantes experiências como *flow*.

Os jogos, portanto, estão extremamente ligados a este processo, já que possuem o fim neles mesmos e há neles o transporte para novas realidades e conquistas.

Esta relação entre os elementos do *flow* com os jogos pode ser vista nos estudos sobre o *GameFlow* de Penelope Sweetser e Peta Wyeth em seu artigo “GameFlow: A Model for Evaluating Player Enjoyment in Games”. [Sweetser e Wyeth 2005]

Neste estudo, o *GameFlow* é dividido em oito elementos: concentração, missões, *skills*, controle, metas claras, *feedback*, imersão e interação social. Os jogos necessitam que o jogador permaneça concentrado na sua tarefa e estas devem ser desafiadoras e agradáveis. Para isto, as atividades devem ser claras e apresentarem um *feedback* imediato ao completá-las, dando ao jogador um senso de controle sobre elas. Este sentimento de domínio provoca a sensação de imersão que, conseqüentemente, gera a perda do senso de vida cotidiana, retira a preocupação consigo mesmo e altera a percepção do tempo.

Portanto, para que um jogo seja eficaz, deve-se manter estes oito elementos em conjunto, ou seja, o jogo deve exigir a concentração do jogador ao mesmo tempo em que facilita esta atividade e cativa a sua atenção, deve possuir missões de acordo com a experiência do jogador para que não sejam nem muito

fáceis para serem tediosas e nem muito difíceis que façam o jogador desistir da experiência, deve ter um controle fácil, para que o jogador seja capaz de aprender a jogar sem muito esforço, evoluindo sua interação com os controles conforme acontece seu progresso no jogo, não necessitando assim de manuais externos para explicações extensas. Outro aspecto importante no desenvolvimento do *flow* consiste em deixar o jogador sentir-se confiante no controle do jogo sendo que este deve apresentar objetivos claros para que haja um direcionamento correto fazendo com que o jogador receba o *feedback* de suas ações nas horas apropriadas. Por fim, o jogo deve proporcionar a imersão fazendo com que o jogador esqueça que está em um mundo paralelo através de outros meios.

Segundo Sweetser e Wyeth, a interação social, apesar de não ser englobada na teoria do *flow* de Csikszentmihalyi, é inserida no conceito de *GameFlow* por ser uma característica importante observada nos jogos. [Sweetser e Wyeth 2005] A atividade de jogar propicia elementos para interações sociais, seja para discutir o conteúdo do jogo, para elogiá-lo ou criticá-lo, ou propriamente através dos jogos, como nos casos de jogos cooperativos ou competitivos.

3.2. A evolução das interfaces

Ao longo da história dos jogos digitais, vemos diversos tipos de *interfaces* que permitem a comunicação do mundo real com o virtual através do corpo, tornando-o fundamental para o uso do *videogame*. Estes dispositivos devem apresentar um mapeamento de botões de fácil aprendizagem e, inclusive, metáforas com o mundo real para que não haja a perda de concentração e imersão do jogador, fazendo com que o dispositivo seja praticamente imperceptível na experiência.

Seguindo a cronologia, temos os *arcades*, *videogames* colocados em móveis para uso público. Estes apresentavam *interfaces* de controle físico específicos para cada jogo que durou até o advento dos *videogames* domésticos que possuíam uma *interface* única que serviria apenas para um número limitado de jogos.

Para o estudo destas interfaces podemos nos apoiar no conceito de “*PrimitiveActions*” (ações primitivas) usado por Gregerson e Grodal. O termo *p-action* é usado para definir a ação real que corresponde a uma ação dentro do mundo simulado. [Audi 2011] Por exemplo, para um personagem realizar um chute dentro de um jogo, o jogador necessita apenas acionar um botão (*p-action*), enquanto o personagem no mundo simulado realiza o chute, levantamento de sua perna.

As interfaces dos jogos digitais permaneceram simples até que em 2007 surgiu o DualShock3, interface controladora do PS3. Com ele temos um progresso tecnológico que exigiu habilidades cognitivas e motoras mais complexas por parte de seus utilizadores. Esta *interface* possui sensores de balanço, acelerômetro linear, gatilhos, botões com sensibilidade

ao tipo de toque, além de controles analógicos com sensibilidade à pressão.

A facilidade ou dificuldade na utilização destas *interfaces* deve-se à habilidade do homem de adaptar seu processo perceptivo conforme a necessidade. À medida em que as *p-actions* são usadas, estas ficam mais naturais e invisíveis.

Desta forma, o surgimento de *interfaces* complexas restringiu o uso dos videogames a pessoas que possuíam tempo para treinar e desenvolver suas habilidades cognitivas.

Esta primeira etapa de *interfaces* foi, portanto, dominada pelo uso de *gamepads* no qual o jogador praticamente movimentava os dedos. Edson P. Pfitzenreuter e Roger Tavares, em seu artigo “Dedões - Corpo – Gestos: A inteligência corporal no design dos jogos digitais”, chegaram a nomear os *videogames* como a “arte dos dedões”. [Pfitzenreuter e Tavares 2010]. Portanto, conclui-se, que neste período dos *games* as *p-actions* se dariam por movimentos reais muito distantes dos virtuais.

Com o advento de novas tecnologias de realidade aumentada que permitem o uso das mãos e de comandos de voz para manipular um mundo simulado e a de realidade misturada que complementa o mundo real com objetos virtuais em tempo real, fez com que as *p-actions* se aproximassem de rotinas de ação e percepção já adquiridas socialmente permitindo o desenvolvimento de interfaces mais intuitivas. Temos, então, o surgimento de um novo tipo de interface, a interface baseada em gestos no ar (*air-based gestures*).

O surgimento de técnicas de captura de imagens e de sensoriamento tornou possível a criação de umas das primeiras aplicações comerciais usando gestos, o EyeToy da Sony. Esta ferramenta era uma câmera ligada ao Playstation que permitia a filmagem do jogador que estava em frente ao videogame. Havia a projeção de sua imagem na tela do jogo, possibilitando que o jogador virasse o personagem central.

A Sony também apresentou um controle em forma de “varinha” usado para detecção de movimentos chamado Move. Este dispositivo usa a câmera do Playstation Eye para identificar os movimentos dos jogadores através do reconhecimento de luz.

A Nintendo também apresentou novas tecnologias baseadas nos gestos. Seu console Wii, além de apresentar o padrão de apertar botões, possui também como entrada de dados o movimento real do jogador ao movimentar o controle. Seu controlador Wiimote usa-se do acelerômetro para reconhecimento de gestos e também apresenta feedback de som e força, o que permite a melhoria da experiência do jogador. Outro fato é a possibilidade de adquirir um controlador chamado Nunchuk para ser utilizado em conjunto com o Wiimote, auxiliando-o como um modo de interação adicional.

Em 2010, a Microsoft lançou para seu console Xbox 360 o dispositivo Kinect, que permite o reconhecimento corporal do jogador, para que este o use como controle de seu jogo. Para este dispositivo não é necessário o uso de controladores. O Kinect usa a

tecnologia de câmera junto com sensores de profundidade para reconhecer o corpo do indivíduo.

Desta forma, o ambiente ganha importância, já que todas estas tecnologias necessitam dele para funcionar. O processo de imersão também evolui, ele não depende mais somente de estímulos visuais e auditivos, o tato e o espaço físico o complementam em seu processo.

Estas tecnologias e conceitos estão cada vez mais presentes em outros meios com a finalidade de ampliar os processos de imersão e facilitar a comunicação do real com o mundo simulado, inclusive em aplicações fora do circuito de jogos digitais.

4. A evolução da Web

No desenvolvimento web temos o HTML como principal suporte à publicação de conteúdo. Segundo o W3C (World Wide Web Consortium) o HTML5 é definido como “uma linguagem de *Hypertexto*, para fácil navegação entre as fontes de informação” [W3C 2014], usado para publicação de conteúdo como textos, imagens, vídeos, áudio, entre outros. Sendo o hipertexto definido como o conjunto de elementos ligados por conexões.

Esta tecnologia passou por algumas evoluções até chegar no seu estado atual, contudo apresentou uma mudança mais expressiva na sua passagem do HTML 4.01 para o HTML 5.

Com o aumento de vendas de *smartphones* que cada vez mais possuem um melhor processamento e resolução de imagem, juntamente com a ampliação do uso da banda larga móvel, o HTML5 torna-se uma tecnologia poderosa e dinâmica capaz de produzir aplicações web com diversos recursos, permitindo inclusive o desenvolvimento de aplicações compatíveis com dispositivos móveis.

Com este aumento de recursos disponíveis para desenvolvedores web pode-se pensar em explorar novas formas de interação do usuário com a web, trazendo interfaces mais intuitivas e melhorando o processo de imersão.

Esta evolução do HTML trouxe uma série de vantagens como a redução da necessidade de *plug-ins*, relacionar-se melhor com os erros, possibilitar a manipulação dos elementos do HTML mantendo a aplicação leve e funcional, uma melhor organização da estrutura do código com o surgimento de novas *tags*, além de tornar-se independente do dispositivo.

Novos elementos de áudio e vídeo permitem uma rápida implementação destes recursos assim como já acontecia com imagens estáticas. Há também o canvas, uma área destinada ao desenho de gráficos interativos e dinâmicos. Podem-se criar atualmente aplicações web com os mais variados recursos, como interações *drag-and-drop*, armazenamento local de dados, cache do navegador e acesso a arquivos locais.

O HTML5 veio, portanto, para revolucionar o desenvolvimento web, trazendo algo dinâmico e poderoso, ao mesmo tempo em que se torna fácil de codificar e programar em diversas plataformas.

Atualmente, também têm-se as APIs (Application programming interface) que facilitam o desenvolvimento web. Os navegadores competem pela implementação de APIs, e por este motivo pode-se encontrar algumas diferenças de recursos entre um navegador e outro. MathML e SVG são exemplos de APIs usadas para facilitar o uso de formas matemáticas e a criação de gráficos 2D, respectivamente.

Para este estudo, as APIs de grande destaque foram as encontradas no pacote webRTC que trata-se de um conjunto de normas, protocolos e APIs JavaScript que possibilitam peer-to-peer de áudio, vídeo e compartilhamento de dados entre os navegadores. Em vez de depender de *plug-ins* de terceiros o webRTC transforma comunicação em tempo real em um recurso padrão que qualquer aplicativo web pode alavancar através de uma API JavaScript simples. [Grigorik 2013]

Ou seja, este conjunto de APIs permite que se elaborem novos tipos de interações na Web utilizando apenas JavaScript. Com ela têm-se o acesso a hardwares de áudio e vídeo do usuário sem a necessidade de nenhum *plug-in*, tornando a aplicação mais leve.

O trabalho do navegador para realizar tais tarefas é simplificado com o uso de duas principais APIs: a MediaStream, que é responsável por captar os recursos de áudio e vídeo; e a PeerConnection, que é usada para realizar o processo de comunicação dos dados captados.

Conclui-se que o webRTC juntamente com o HTML5 oferecem muitas vantagens para a criação de aplicações web. Estas tecnologias ainda permitem o desenvolvimento de uma aplicação que não depende de um único dispositivo, já que os recursos do HTML5 e do webRTC estão disponíveis não somente para computadores, mas também para dispositivos móveis.

5. Desenvolvimento

A evolução tecnológica tem ajudado a expandir possibilidades de desenvolvimento de novas interações entre máquina e indivíduo. Estes novos meios de comunicação entre mundo real e virtual têm ganhado grande destaque pela sua facilidade de uso, já que são construídos com a intenção de aproximar o movimento concreto do movimento simulado.

Dentro dos estudos sobre jogos digitais estas interações baseadas em gestos e áudio têm atraído um público que anteriormente não era atingido pela indústria de jogos digitais.

Um deles são os indivíduos que não apresentavam interesse nesse tipo de entretenimento, já que não gostariam de deprender grande energia para o aprendizado de interfaces complexas, ou seja, interfaces que necessitassem de grande habilidade cognitiva. Outro grupo são os de pessoas que apresentam algum tipo de deficiência física que a impeça de manusear complexos controles com botões para interagir.

Neste sentido, a fim de explorar este tipo de interação, decidiu-se elaborar uma aplicação simples para obter dados sobre o assunto e analisar sobre as possibilidades que isto geraria. Foi escolhido, portanto, a elaboração de uma aplicação no qual o cenário seria o universo e o jogador controlaria uma nave espacial utilizando o seu corpo como controle.

Para que esta experiência fosse realizada foi necessário que a tecnologia dos navegadores permitisse a comunicação com o hardware do computador do jogador possibilitando a entrada de dados através de uma *webcam*. Esta evolução foi apresentada através do pacote de APIs webRTC que surgiu após a apresentação da nova versão do HTML, o HTML 5.

As imagens captadas pela *webcam* são frequentemente modificadas por sombras causadas pelo próprio movimento do jogador, ou ainda, modificadas conforme a qualidade do equipamento utilizado. Câmeras diferentes podem apresentar informações diferentes de um mesmo ambiente. Portanto, tem-se o desafio de rastrear a movimentação do jogador utilizando a captação de dados que podem sofrer alterações, uma vez que a iluminação pode ser instável.

Sendo assim, pode-se dividir a elaboração de uma alternativa de interação web para jogos digitais em três etapas:

- A etapa que trata da requisição de captura de imagem com o uso da *webcam*, neste caso representada pela API webRTC, utilizando, desta forma, as novidades apresentadas pela tecnologia do HTML 5;
- A etapa na qual realiza-se o processamento da imagem captada com o uso da API Canvas, a qual foi baseada nos estudos sobre reconhecimento de objetos através de modelos de cores;
- A etapa da elaboração da aplicação utilizada pelo jogador, neste caso, construída através da WebGL utilizada para desenvolver os gráficos 3D.

5.1. Captura de imagens

Esta etapa requer que o navegador seja capaz de acessar o hardware do sistema para capturar vídeos sem o uso de *plug-ins* ou *drivers*. As especificações de MediaCapture e Streams da W3C apresentam uma série de APIs JavaScript que permitem este processo de captura e o pós-processamento da imagem capturada.

Os dados adquiridos, no entanto, são limitados pela capacidade da fonte de entrada, neste caso, a *webcam*. Esta pode apresentar fluxo de vídeos de alta resolução, enquanto outras não. Sendo assim, o método `getUserMedia()` do pacote webRTC permite especificar restrições obrigatórias e opcionais para atender as necessidades da aplicação. O código abaixo, retirado do livro “High Performance Browser Networking”, demonstra estes aspectos [Grigorik 2013]:

```
<video autoplay></video> (1)
<script>
  var constraints = {
```

```
    audio: true, (2)
    video: { (3)
      mandatory: { (4)
        width: { min: 320 },
        height: { min: 180 }
      },
      optional: [ (5)
        {width: { max: 1280 }},
        {frameRate: 30 },
        { facingMode: "user" }
      ]
    }
  }
navigator.getUserMedia(constraints,
  gotStream, logError); (6)
function gotStream(stream) { (7)
  var video =
  document.querySelector('video');
  video.src =
  window.URL.createObjectURL(stream);
}
function logError(error) { ... }
</script>
```

Seguem as descrições das principais etapas do código apresentado:

- I. A linha (1) mostra a saída da captura no HTML.
- II. A linha (2), requisição obrigatória de áudio.
- III. A linha (3) mostra a requisição obrigatória de vídeo.
- IV. A linha (4) mostra a lista de restrições obrigatórias no vídeo.
- V. A linha (5) mostra a lista de restrições opcionais no vídeo.
- VI. A linha (6) mostra a requisição de áudio e vídeo a partir do navegador.
- VII. A linha (7) mostra a função de Callback para processar os dados adquiridos pela API.

Portanto, vê-se que a função `getUserMedia()` é responsável por solicitar o acesso à *webcam* e fazer com que os dados de saída correspondam às restrições especificadas durante o processo de solicitação.

Desta forma, a saída de dados adquiridos é usada pela API Canvas para o pós-processamento de quadros individuais. Esta API pode ser definida como uma tela retangular que permite, com o uso do JavaScript, o desenho de diversas formas e imagens.

Assim, as imagens capturadas pela API webRTC são, portanto, mostradas na tag de vídeo do HTML5 permitindo a obtenção destes dados para desenhá-las no Canvas utilizando a função `drawImage()`.

Para trabalhar o processamento de imagem de cada frame adquirido, usa-se a função `getImageData()` que retorna os valores em RGB de cada pixel apresentado. Logo, têm-se as informações necessárias para iniciar a próxima etapa referente ao reconhecimento de objetos.

5.2. Reconhecimento de objetos

Com o processamento de imagens, que consiste em processar dados no qual a entrada e saída são imagens

tais como fotografias ou quadros de vídeo, consegue-se analisar dados para o reconhecimento automático de objetos em imagens naturais, ou seja, aquelas que não são produzidas através da computação gráfica.

O uso deste processo de reconhecimento tem crescido bastante não só em aplicações para entretenimento, mas também para uso comercial, como em *engines* de busca de imagens na Web, em sistemas de câmeras fotográficas, localizando rostos para o foco automático, ou ainda em tecnologias mais complexas como a *MobileEye*, usada por motoristas para detectar a que distância outros carros estão a frente do seu veículo.

Para que o reconhecimento de objetos seja possível, usa-se a nossa experiência de definição padrão sobre o que é um determinado elemento para poder estabelecer uma regra de reconhecimento. Por exemplo, para identificar uma bola de basquete em uma imagem 2D, pode-se usar o princípio da busca por um objeto circular de tom alaranjado. A Figura 2 mostra uma imagem na qual seria prático utilizar a regra citada para o reconhecimento da bola de basquete.



Figura 2: Exemplo de uma imagem que mostra efetivamente o objeto "bola de basquete"

Porém, estes objetos podem se apresentar de diversas formas em uma imagem, o que nos traz o desafio desta área. Sendo alguns destes problemas: a variação do ponto de vista, a iluminação, a oclusão, a escala, a deformação/articulação, a variação física de uma classe de objetos e o fundo da imagem com muita informação.

O problema do ponto de vista obriga o algoritmo de busca a se preocupar com todas as possibilidades de posicionamento do objeto, ou seja, o algoritmo deve estar preparado para reconhecer o objeto em seus diferentes ângulos. Por exemplo, o algoritmo de reconhecimento de uma xícara deve atentar-se para os diversos ângulos que ela pode ocorrer assim como mostram as imagens da Figura 3.



Figura 3: Exemplos de imagens de uma xícara de café em diferentes ângulos

Estas diferenças de ponto de vista podem ocasionar em oclusões de parte destes objetos, o que dificulta o processo de análise da imagem. Podemos ver isto nas imagens da Figura 2, na qual a imagem I da xícara apresenta a alça quase imperceptível, podendo gerar dúvidas no processo de reconhecimento. A bola de basquete pode apresentar o mesmo problema, fazendo

com que a regra de busca que consistia em encontrar um elemento circular de tom alaranjado não possa mais ser utilizada, como vê-se na Figura 4.



Figura 4: Exemplo de imagem com uma bola de basquete em oclusão

A iluminação também traz algumas dificuldades para o processo, podendo gerar oclusões ou alterar a tonalidade do objeto que deve-se buscar. Na Figura 4, a sombra na bola de basquete gera uma oclusão parcial do objeto e uma coloração bem variada de um extremo ao outro deste.

Quanto à escala, esta pode enganar o algoritmo nas diferenciações, por exemplo, entre miniaturas e objetos reais. No caso da Figura 5 tem-se um carro de brinquedo que pode ser confundido com um carro real.



Figura 5: Diferença de escala entre objetos parecidos

Além disso, uma classe de objetos pode apresentar variações físicas. Nem todos os vasos, por exemplo, apresentam o mesmo aspecto, como exemplificado na Figura 6.



Figura 6: Exemplo de vasos com formatos diferentes

Objetos que permitem uma deformação também geram dúvidas, como a articulação do corpo do gato demonstrada na Figura 7.



Figura 7: Diferentes articulações de um gato

Por fim, ao trabalhar com análises de imagens, quanto maior for o número de informações localizadas no material de estudo, maior será a dificuldade em encontrar um elemento específico. Muitas vezes elementos localizados no fundo da imagem atrapalham a localização do elemento principal.

Desta forma, como o objetivo desta etapa é reconhecer objetos em tempo real através de câmeras, opta-se por estudar o processo de reconhecimento de objetos através de sistemas de cores por ser um método mais rápido do que se fosse se basear por outras características dos objetos.

Entretanto, os valores que a câmera obtém podem ser influenciados por alguns fatores. Como já discutido, ambientes com variações de luz podem atrapalhar o processo, assim como o próprio movimento do alvo a ser detectado que pode gerar mudanças nas sombras que afetam a sua superfície. Além disso, o próprio processo de captura da imagem apresenta alterações. Diferentes câmeras podem produzir diferentes cores de um mesmo elemento em um mesmo local com a mesma iluminação. Em ambientes com uma luz ideal não existiria variações de cores o que facilitaria o processo de reconhecimento e de *tracking* de um elemento.

Focando no objetivo de desenvolvimento de uma aplicação que rastreie a posição do jogador perante a tela do dispositivo utilizado, pode-se dividir esta etapa em três processos:

- A busca por pixels relevantes que representem cores de pele usando modelos de cores;
- O processo de redução de ruídos usando as técnicas de dilatação e erosão;
- O cálculo da área de maior densidade de pixels frame a frame para desta forma estimar a localização e, consequentemente, a movimentação do jogador.

Embora pareça que as cores da pele humana tenham uma enorme diferença, elas estão muito próximas em relação à tonalidade, sendo a saturação e o brilho o fator que as difere. Desta forma, deve-se estudar cada modelo de cor e estabelecer os valores ideais para separação dos pixels relacionados à pele dos pixels não relevantes.

Portanto, para garantir uma maior efetividade neste estudo, foi utilizado, simultaneamente, três modelos de cores no processo de reconhecimento da cor de pele: o RGB, o HSV e o YCbCr.

O modelo RGB é composto por três cores primárias, vermelho, verde e azul que se combinam para produzir uma cor resultante, sendo que seus três componentes representam não somente as cores, mas também a luminosidade. Isto dificulta o processo de análise da imagem já que a cor e a luminosidade encontram-se relacionadas em um mesmo valor.

Para o reconhecimento da cor de pele no modelo de cor RGB escolheu-se o algoritmo apresentado no artigo “Human skin colour clustering for face detection” de Jure Kovac, Peter Peer, e Franc Solina, definido para ambientes com iluminação uniforme, composto das seguintes regras [Kovac 2003]:

- I. $R > 95$ e $G > 40$ e $B > 20$ e $\max\{R, G, B\} - \min\{R, G, B\} > 15$, para que os componentes não apresentem valores próximos e, portanto, eliminar os tons de cinza;
- II. e $|R - G| > 15$, para que os valores de R e G não sejam próximos;
- III. e $R > G$ e $R > B$, para garantir que o componente R seja o de mais alto valor entre todos;

Com este algoritmo consegue-se transformar a imagem captada da webcam em uma imagem binária como apresentado na Figura 8.



Figura 8: Teste de reconhecimento de cor de pele com o sistema RGB

Outro modelo utilizado foi o HSV que é composto pelo *Hue* (tonalidade), *Saturation* (saturação) e *Value* (valor) e foi concebido pensando na técnica que os artistas usam para misturar cores primárias para obter outros tons. Portanto, se relacionar estes conceitos com a arte da pintura, podemos dizer que a tonalidade está para a tinta assim como a saturação para o sombreado e o valor para o brilho. Portanto, o *Hue* é a cor pura, expressa em graus, *Saturation* é a pureza da cor, que descreve o quanto uma cor tende para o branco, sendo a saturação igual a zero a cor branca. O valor representa o brilho, descrevendo o quão escura uma cor é, sendo que o valor igual a zero representa a cor preta.

Diferentemente do RGB, este modelo facilita a busca pela cor desejada já que separa a cor pura no elemento *Hue* (tonalidade).

Para o reconhecimento da cor de pele no modelo de cor HSV foram escolhidas as regras apresentadas por Shervin Emami no seu artigo “Color-based Blob Detection” que consiste em [Emami 2010]:

- I. $20 < V < 90$, usa-se esta regra para eliminar cores muito claras e escuras (brancos e pretos), já que o V representa o brilho, ou seja, valores baixos aproximam-se do preto, valores altos aproximam-se do branco;
- II. e $S > 10$, usa-se esta regra para eliminar cores intensamente claras, já que o S representa a saturação, ou seja, quanto mais baixo o seu valor, mais a cor tende ao branco;
- III. e $0 < H < 18$, usa-se esta regra para eliminar tons que se distanciam da cor vermelha.

Como os dados dos pixels captados através da webcam e do JavaScript são em formato RGB, faz-se necessária a conversão deste modelo de cor para o sistema de cor HSV conforme o algoritmo apresentado por Hélio Pedrini [Pedrini 2013] na Figura 9.

$$H = \begin{cases} 60 \frac{(G - B)}{(M - m)}, & \text{se } M = R \\ 60 \frac{(B - R)}{(M - m)} + 120, & \text{se } M = G \\ 60 \frac{(R - G)}{(M - m)} + 240, & \text{se } M = B \end{cases}$$

$$S = \begin{cases} \frac{(M - m)}{M}, & \text{se } M \neq 0 \\ 0, & \text{caso contrário} \end{cases}$$

$$V = M$$

em que $m = \min(R, G, B)$ e $M = \max(R, G, B)$.

Figura 9: Conversão do sistema de cor RGB para o sistema HSV

Desta forma, com a conversão e a regra proposta por Shervin Emami transforma-se a imagem captada em uma imagem binária resultante, como mostra a Figura 10.

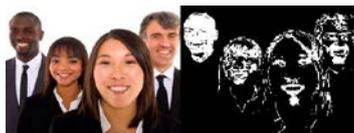


Figura 10: Teste de reconhecimento de cor de pele com o sistema HSV

Por último, utilizou-se o modelo YCbCr que é composto pela Y (luminância), que representa o brilho, ou seja, quanto mais baixo o seu valor, mais escura a cor será, e pelas informações de cores de Cr (diferença entre o componente vermelho e a luminância) e de Cb (diferença entre o componente azul e a luminância).

Para o reconhecimento da cor de pele no modelo de cor YCbCr escolheu-se as regras apresentadas por Francesca Gasparini e Raimondo Schettini no seu artigo “Skin segmentation using multiple thresholding” que consiste em [Gasparini e Schettini 2014]:

I. $77 \leq Cb \leq 127$, ou

II. $133 \leq Cr \leq 173$

Tem-se, portanto, um algoritmo que busca as cores encontradas no espectro de cor de pele humana sem considerar a variação de intensidade dada pelo Y.

Para uso destas instruções faz-se necessária a conversão do sistema de cor RGB para o sistema de cor YCbCr que é dado pelo algoritmo apresentado no artigo “Face Detection Using Color Thresholding, and Eigenimage Template Matching” de Diedrick Marius, Sumita Pennathur e Klint Rose [Marius et al. 2003], demonstrado na Figura 11.

$$Y = 0.257 * R + 0.504 * G + 0.098 * B + 16$$

$$Cb = 0.148 * R - 0.291 * G + 0.439 * B + 128$$

$$Cr = 0.439 * R - 0.368 * G - 0.071 * B + 128$$

Figura 11: Conversão do sistema de cor RGB para o sistema de cor YCbCr

Desta maneira, com a conversão para o sistema de cor YCbCr e a utilização do algoritmo para busca de pixels que representem a cor de pele tem-se uma imagem binária resultante como visto na Figura 12.



Figura 12: Teste de reconhecimento de cor de pele com o sistema YCbCr

Define-se, portanto, para este estudo a utilização em conjunto dos algoritmos de reconhecimento de cor de pele baseado nos modelos RGB, HSV e YCbCr. Têm-se, portanto, na Figura 13 um exemplo do resultado destas regras utilizadas simultaneamente.



Figura 13: Teste de reconhecimento de pele com os sistemas RGB, HSV e YCbCr

Desta forma, para o desenvolvimento da aplicação obtendo as imagens com o uso da webcam, utiliza-se os mesmos modelos de cores RGB, YCbCr e HSV para localizar os pixels que possivelmente correspondem à cor de pele transformando a imagem RGB captada em uma imagem binária na qual os pixels brancos representam os pixels relevantes, e os pixels pretos representam pixels não relevantes como mostra a Figura 14.



Figura 14: Imagem binária captada através da webcam

Para reduzir os ruídos, utiliza-se o processo de dilatação e erosão, obtendo assim formas mais demarcadas na imagem binária. A dilatação faz com que *pixels* não relevantes que estavam próximos a *pixels* relevantes, tornem-se importantes. Esta técnica é responsável por “engrossar” ou fechar as formas.

Já o processo de erosão faz com que *pixels* relevantes tornem-se não relevantes quando estão próximos a *pixels* não importantes. Portanto, esta técnica é responsável por eliminar pixels indesejados, reduzindo as bordas das formas nas imagens binárias. Na Figura 15 têm-se o resultado da Figura 14 após o processo de dilatação e erosão.



Figura 15: Imagem binária captada através da webcam com o processo de dilatação e erosão

Ao final destes processos consegue-se uma imagem binária resultante, na qual pode-se observar a área com maior densidade de pixels relevantes e, conseqüentemente, definir em qual posição o jogador está localizado à frente do dispositivo utilizado.

Com estes dados tem-se como seguir a movimentação do jogador e pode-se assim iniciar o processo de criação da aplicação para interagir com estes dados. Logo, chega-se na terceira etapa do processo, na qual utilizamos o WebGL para a criação de gráficos 3D nos navegadores através do JavaScript.

5.3. Explorando a interação

Cria-se, portanto, o cenário do universo e o seu objeto principal, o qual representará o jogador, a nave espacial. Esta realizará seus movimentos conforme a movimentação do corpo do jogador, construindo, desta forma, um mundo virtual controlado por movimentos reais. Um vídeo da aplicação pode ser visto em: <http://tinyurl.com/mtcpsq>.

A partir deste desenvolvimento pode-se concluir e constatar algumas dificuldades apresentadas no processo. Para elucidar estes aspectos testa-se a aplicação em diversos ambientes com iluminações

variadas, são estes: ambiente com luz artificial fluorescente, ambiente pouco iluminado com luz natural ao fundo, ambiente com luz natural frontal, ambiente com luz natural suave, ambiente com luz natural intensa.

Neste contexto, foi constatado que em ambientes onde a luz distorce a cor original do elemento a ser identificado o funcionamento da aplicação é prejudicado, pois não será possível identificar o jogador e analisar os seus movimentos.

Outro ponto que pode agravar a eficiência se dá quando o algoritmo detecta muitos elementos que não deveria ao fundo. Em alguns casos, o algoritmo detectou como possível cor de pele o fundo da sala no qual o jogador estava. Este resultado atrapalhou o cálculo de maior densidade de *pixels* relevantes no eixo “x”, impossibilitando que a movimentação do jogador controlasse a direção da espaçonave. Outro ponto a destacar é a detecção de algumas cores de peças de roupa como cor de pele. Este acontecimento pode gerar alguma dificuldade no processo de interação, já que os ombros de um indivíduo trariam uma maior densidade de *pixels* do que o rosto propriamente.

6. Considerações finais

O conceito de *flow*, que define atividades com alto grau de imersão e satisfação, auxilia neste trabalho na busca de uma alternativa de interação que permita com que o jogador tenha a sua concentração e habilidades focadas no conteúdo do jogo, ou seja, no seu mundo virtual. Busca-se, então, uma interação de fácil aprendizagem que promova uma comunicação quase imperceptível entre máquina e homem.

Com a modernização dos navegadores, e consequentemente com o surgimento da *Web Real Time Communication (WebRTC)*, um conjunto de APIs que permitem com que os navegadores possam capturar a webcam e microfones diretamente sem utilização de *plug-ins*, pode-se então pensar em interações focadas no reconhecimento de objetos.

Logo, escolheu-se o desenvolvimento de uma aplicação que detectasse o movimento do corpo humano para usá-lo como entrada de dados em um jogo digital, promovendo o desenvolvimento de uma interface intuitiva.

O maior desafio neste processo seria o uso de técnicas de processamento de imagens que não sobrecarregassem o *hardware* do usuário a ponto de dificultar a captura e processamento em tempo real, já que o processo se faz todo do lado do cliente.

Desta maneira, optou-se como forma de rastreamento do movimento do jogador a técnica de reconhecimento de cor de pele através dos modelos de cores RGB, HSV e YCbCr, simultaneamente, para que o processo tivesse maior eficácia.

Este método garantiu a transformação da imagem captada da webcam em uma imagem binária, recurso que facilitaria o processo de refinamento dos pontos relevantes na imagem. Nesta etapa usa-se, portanto, as

técnicas de dilatação e erosão, métodos utilizados com a finalidade de reduzir ruídos na imagem binária.

É com o resultado deste processo de refinamento que posteriormente calcula-se a área de maior densidade com o objetivo de encontrar a posição do jogador perante a tela e, consequentemente, realizar o rastreamento de seu movimento.

Para aumentar este valor de efetividade deve-se continuar os estudos e a evolução dos algoritmos para o tratamento de algumas dificuldades encontradas, sendo elas: tratamento da retirada de *pixels* relevantes que indiquem cores de roupas e tratamento de subtração de fundo.

Com o desenvolvimento deste exemplo de interação, na qual o jogador controla a espaçonave com o movimento do seu corpo, pode-se constatar que existe um processo que possibilita uma grande imersão do jogador, já que este não precisa passar por um processo de aprendizagem de controle da espaçonave. O jogador se coloca como sendo o próprio objeto, sendo que a ação da espaçonave de ir para a direita ou para esquerda, assemelha-se à sua movimentação real de realizar estes movimentos.

Este fato coloca a aplicação em harmonia com as características de concentração, facilidade de controle e *feedback* imediato vistas no conceito de *flow*, já que esta possibilita um controle intuitivo, no qual não há gasto de energia e desvio de concentração para aprender o controle da espaçonave. Tem-se, portanto, uma ação primitiva igual a ação realizada virtualmente, deixando o jogador livre do esforço de aprendizagem cognitiva, permitindo que seu foco de concentração fique totalmente no conteúdo da aplicação ampliando sua imersão, já que suas ações reais estão diretamente envolvidas nas virtuais.

Referências

- AUDI, G.M., 2011. *Cognição e Videogame: o jogo narrativo e o uso do corpo como interface*. Em: *SBGames 2011*, Salvador, Brasil.
- BASILIO, J., TORRES, G.A., PÉREZ, G.S., MEDINA, L.K.T., MEANA, H.M.P., 2011. *Explicit Image Detection using YCbCr Space Color Model as Skin Detection*. Em: *AMERICAN-MATH'11/CEA'11 Proceedings of the 2011 American conference on applied mathematics and the 5th WSEAS international conference on Computer engineering and applications*, 123-128, January 29-31, 2011, Puerto Morelos, Mexico.
- BOUMAN, K., 2012. *Image Processing Technology and Applications: Object Recognition and Detection in Natural Images*. Disponível de: http://people.csail.mit.edu/klbouman/pw/papers_and_presentations/ObjectRecognitionDetection-11-25-12.pdf [Acessado 09 abril 2014].
- BRADFORD, A., HAINE, P., 2011. *HTML5 Mastery: Semantics, Standards, and Styling*. New York: Springer.
- CAI, J., GOSHTASBY, A., 1999. *Detecting human faces in color images*. Em: *Image and Vision Computing* 18. 63–75

- CSIKSZENTMIHALYI, M., 1990. *Flow: The Psychology of Optimal Experience*. New York: Harper Perennial.
- EMAMI, S., 2010. Color-based Blob Detection. Disponível de: <http://www.shervinemami.info/blobs.html> [Acessado 10 abril 2014].
- ERICSSON. 2012. Ericsson Review: The communications technology journal since 1924. Edita Västra Aros: Västerås.
- FREEDIGITALPHOTOS.NET. 2014. Multi Ethnic Business Team Stock Photo. Disponível de: http://www.freedigitalphotos.net/images/Business_People_g201-Multi_Ethnic_Business_Team_p66544.html [Acessado 09 abril 2014].
- FREEIMAGES. Disponível de: <http://www.freeimages.com> [Acessado 09 abril 2014].
- GASPARINI, F., SCHETTINI, R., 2006. Skin segmentation using multiple thresholding. Em: *Internet Imaging VII, IS and T/SPIE, 60610F-1-60610F-8*. SPIE.
- GIBSON, J.J., 1986. *The Ecological Approach to Visual Perception*. London: Lawrence Erlbaum.
- GRIGORIK, I., 2013. *High Performance Browser Networking*. Sebastopol: O'Reilly Media.
- JACK, K., 2008. *Digital vídeo and dsp: Instant Access*. Burlington: Newnes.
- JUNGLES, R., BATTAIOLA, A.L., 2006. Análise de Tecnologias para a Implementação de Jogos Web. Em: *SBGames 2006, Recife, Brasil*.
- KIMER, C., SISCOUTO, R., TORI, R., 2006. Fundamentos e tecnologia de realidade virtual e aumentada. Em: *Pré-Simpósio VIII Symposium on Virtual Reality, Belém, Brasil*.
- KOVAC, J., PEER, P., SOLINA, F., 2003. Human skin colour clustering for face detection. *Proceeding of EUROCON 2003. Computer as a Tool. The IEEE Region 8*.
- LIMA, E.E.S., RUBY, C., POZZER, C.T., SILVA C.N., 2010. Visão Computacional e Reconhecimento de Comandos de Voz Aplicados na Interação com Jogos e Ambientes 3D. Em: *SBGames 2010, Florianópolis, Brasil*.
- MARIUS, D., PENNATHUR, S., ROSE, K., 2003. Face Detection Using Color Thresholding, and Eigenimage Template Matching. Disponível de: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.1609&rep=rep1&type=pdf> [Acessado 11 abril 2014].
- NOVAK, J., 2010. *Desenvolvimento De Games*. São Paulo: Cengage Learning.
- OSMAN, G., HITAM, M., ISMAIL, M., 2012. Enhanced skin colour classifier using RGB ratio model. Em: *International Journal on Soft Computing (IJSC) Vol.3, No.4*.
- PARISI, T., 2012. *webGL up and running*. Sebastopol: O'Reilly Media.
- PAULA, L.P.R., BONINI, N.R., MIRANDA, R.F., 2006. *Camera Kombat - Interação Livre para Jogos*. Em: *SBGames 2006, Recife, Brasil*.
- PEDRINI, H., 2013. Análise de imagens. Disponível de: http://www.ic.unicamp.br/~helio/disciplinas/MO445/aula_cores.pdf [Acessado 21 abril 2014].
- PETRAKIS, E.G. 2014. Euripides G.M Petrakis. Disponível de: <http://www.intelligence.tuc.gr/~petrakis/> [Acessado 21 abril 2014].
- PILGRIN, M., 2010. *HTML5 up and running*. Sebastopol: O'Reilly Media.
- PFUTZENREUTER, E.P., TAVARES, R., 2010. *Dedões - Corpo – Gestos: A inteligência corporal no design dos jogos digitais*. Em: *SBGames 2010, Belo Horizonte, Brasil*.
- PHOTL.COM., 2014. *Photl.com: stock photography*. Disponível de: <http://www.photl.com> [Acessado 09 abril 2014].
- RIBEIRO, B., LUCCHESI, F., CASTAÑEDA, Z. 2009. *Interfaces de Jogos Digitais*. Disponível em: <http://www.dca.fee.unicamp.br/~martino/disciplinas/ia369/trabalhos/t3g3.pdf> [Acessado 25 setembro 2013].
- ROGERS, Y. SHARP, Y., PREECE, J., 2013. *Design de interação: Além da interação humano-computador*. Porto Alegre: Bookman.
- SINGH, S., CHAUHAN, D.S., VATSA, M. SINGH, R., 2003. A Robust Skin Color Based Face Detection Algorithm. Em: *Tamkang Journal of Science and Engineering, Vol. 6, No. 4, 227-234*.
- SOUZA JR., E., DELGADO, G.A., DAZZI, R.L.S., ELLERY, N. 2009. Development of an interactive game using a webcam. Em: *SBGames 2009, Rio de Janeiro, Brasil*.
- SWEETSER, P., WYETH, P., 2005. *GameFlow: a model for evaluating player enjoyment in games*. *Computers in Entertainment (CIE)*, New York.
- VARELA, F.J., THOMPSON, E. ROSCH, E., 2003. *A mente incorporada*. Porto Alegre: Artmed.
- WANG, B., CHANG, X., LIU, C., 2011. Skin Detection and Segmentation of Human Face in Color Images," *International Journal of Intelligent Engineering & Systems*, 10-17.
- WANG, P.S., 2013. *Dynamic Web Programming and HTML5*. Boca Raton: CRC Press.
- W3C. *Curso de HTML 5*. Disponível de: <http://www.w3c.br/cursos/html5/conteudo> [Acessado 09 abril 2014].
- ZHANG, Q., ZHANG, J., 2009. RGB Color Analysis for Face Detection. Em: *Advances in Computer Science and IT, D M Akbar Hussain (Ed.), InTech*, Disponível em: <http://www.intechopen.com/books/advances-in-computer-science-and-it/rgb-color-analysis-for-face-detection> [Acessado 06 julho 2014].