Programando Comportamento de Pontos de Ação em Aplicações de Realidade Aumentada

Christopher Shneider Cerqueira
Departamento de Matemática e Computação - DMC
Universidade Federal de Itajubá - MG
Itajubá - BR
christophercerqueira@gmail.com

Claudio Kirner
Departamento de Matemática e Computação - DMC
Universidade Federal de Itajubá - MG
Itajubá - BR
ckirner@gmail.com

***Resumo*—A evolução das aplicações de realidade aumentada está ocorrendo no sentido de incorporar-se inteligência nos ambientes. Nesse sentido, é essencial que os elementos virtuais apresentem comportamentos. Este artigo trata do comportamento entre pontos de ação num ambiente de realidade aumentada. Os pontos de ação são inseridos numa máquina de estados, e de acordo com a ação realizada, os estados podem ser alterados, possibilitando que uma lógica comportamental possa ser "programada", manipulando a aplicação.**

Palavras-chaves: Realidade aumentada; comportamento; interaçã; autoria de comportamento

I. INTRODUÇÃO

Tradicionalmente, as aplicações de realidade aumentada têm sido muito simples, resumindo-se na apresentação de objetos e anotações virtuais no ambiente real [1][2]. Com o avanço da área, estão surgindo aplicações interativas que disparam ou controlam objetos virtuais. No entanto, há uma tendência de ter-se anotações e objetos virtuais inteligentes (com comportamento) que reagem a certas situações do ambiente e dos usuários. O grande problema dessas aplicações é a dificuldade de concepção e programação dos ambientes inteligentes.

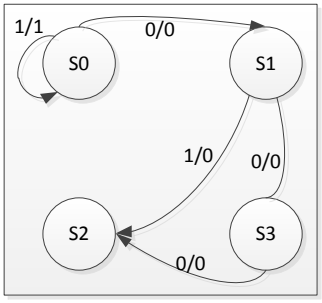


Figura 1 - Exemplo de máquina de estado. Máquina de Mealy.

Uma maneira de facilitar a modelagem e a programação de comportamentos, em aplicações de Realidade Aumentada, é criar uma camada de comportamento e deixar que o usuário criador de aplicações de realidade aumentada possa escolher o que fazer com os itens que compõem a sua aplicação. Este artigo concentra-se numa metodologia que possibilita a descrição do comportamento através da configuração de uma máquina de estados [3], como a máquina de Mealy da Figura 1, por exemplo.

II. SISTEMAS COM COMPORTAMENTO

Dentre as aplicações de RA, existem algumas que apresentam elementos virtuais interativos com comportamento. Por exemplo, alguns destes mais recentes como, os jogos de realidade aumentada do videogame portátil 3DS da *Nintendo*[4] e o aplicativo de *SmartGrid* da *GE* (*General Electric*)[5].

Na aplicação do 3DS, a tela do videogame tem um alvo que interage com componentes colocados na tela, no caso da Figura 2 o alvo é representado com uma varinha de condão.

Figura 2 - Exemplo de jogo de realidade aumentada no Nintendo 3DS.

O aplicativo da GE, como visto na Figura 3, mostra algumas formas de energia renovável, este é sensível a movimentação do marcador (inclinação, distância) e ao som capturado pelo microfone.

Figura 3 - Aplicativo interativo sensível à movimentação do marcador e ao som, desenvolvido pela GE.

Além disso, há algumas ferramentas de autoria como o SACRA [6] e o *Unifeye Design* [7], que apresenta uma camada que permite usar comportamento entre pontos ou objetos virtuais.

III. PONTOS DE AÇÃO

O SACRA introduziu o conceito de pontos de ação sobre uma referência de infraestrutura, no caso, o marcador fiducial. Os pontos de ação são regiões do espaço relativas à origem da infraestrutura que podem realizar alguma atividade. No SACRA, estes podem ser movimentados e ou alternar modelos pré-programados.

Ambas as atividades descritas estão intrínsecas em todos os pontos de ação do SACRA, onde marcadores específicos são utilizados para ativar cada ação.

Na Figura 4 é exibido um exemplo dos pontos de ação do SACRA.

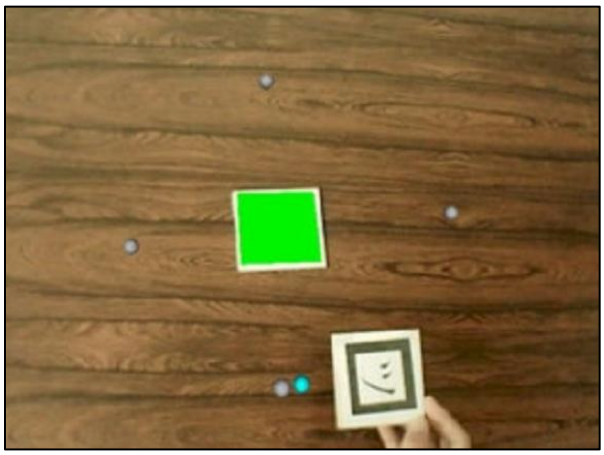


Figura 4 - Pontos de ação do S.A.C.R.A.

A. *Comportamento entre pontos de ação.*

A relação ou a interação entre os pontos de ação definem os comportamentos que podem ser tomados por uma aplicação.

Uma interação pode ser dividida em três etapas: Seleção, Ação e Liberação [8], como na Figura 5. Assim, um ponto deve ser selecionado para que uma ação ocorra, e ao fim desta ação o ponto deve ser liberado de alguma forma ou alguma propriedade configurada.

Figura 5 - Estrutura de uma interação

Expandindo a sequência da Figura 4, podemos criar um subconjunto de comportamentos e modelá-los dentro desta estrutura.

Os comportamentos baseiam-se na seleção de um ponto de ação, seu transporte e como é realizada sua liberação. Na ausência de pontos de ação na região de liberação, o ponto de ação em transporte pode ser deixado em qualquer local, porém numa zona reativa, que possui pontos de ação, a liberação pode ocorrer controlada pelo sistema, onde o ponto de ação é atraído para um ponto ou repelido para origem, ou controlado pelo usuário, neste caso a liberação é testada para ver se é possível liberar na região. Um esquema destes comportamentos está na Figura 6.

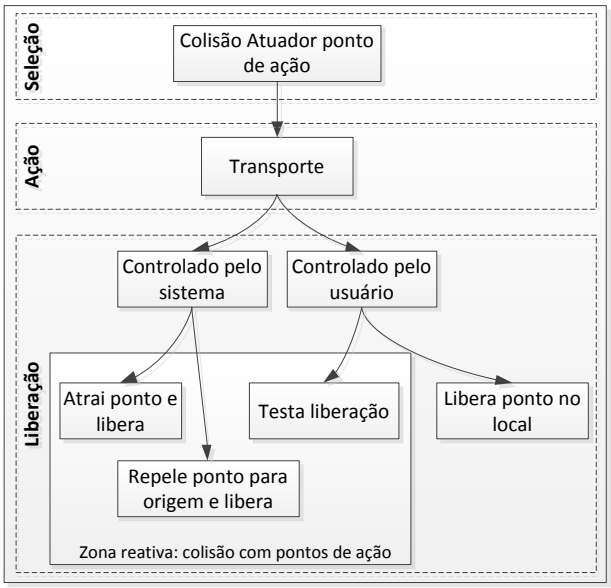


Figura 6 - Expansão da estrutura de interação com comportamentos controlados pelo sistema e pelo usuário.

IV. UTILIZANDO COMANDOS PARA DESCREVER COMPORTAMENTO DO PONTO DE AÇÃO

Uma maneira de externar o comportamento de um ponto de ação é criando estruturas, ou comandos, que descrevem o comportamento do ponto.

Numa aplicação de realidade aumentada, esses comandos podem descrever atributos e ações do ponto de ação; como um modelo deve ser exibido; se este ponto atrai ou repele um ponto específico ou qualquer ponto; se toca som ou se simplesmente fica estático compondo um cenário.

A disponibilização de um conjunto de comandos para interação entre pontos de ação possibilita ao desenvolvedor de Realidade Aumentada uma ferramenta de composição de comportamento que era até então inexistente. Com isso, o desenvolvedor não precisa programar o comportamento e sim apenas utilizar o comando específico para ser implantado no ponto de ação.

Genericamente, os comandos podem assumir a seguinte estrutura básica:

- Identificador do ponto
- Modo de exibição
- Tipo de operação (*OpCode*)
- Argumentos

O identificador indica qual é o ponto de ação que irá ter as características e ação do comando.

O modo de exibição indica a maneira como o ponto de ação é exibido. Por exemplo: pode exibir somente o modelo

(OBJECT), exibir o ponto de ação (BALL), ambos (BOTH) ou nenhum (HIDE).

O tipo de operação indica que naquele ponto há uma ação especificada e que ela é do tipo especificado. Por exemplo: um ponto estático (STAT), um ponto que pode se mover (DRGF) ou um objeto que atrai um ponto específico (ATTO).

Os argumentos variam de acordo com o tipo de operação.

Um comando simples seria o de definir que um ponto exibe um modelo e não tem ação nenhuma.

`1 OBJECT STAT`
[PointID] [ViewMode] [Opcode]

Um comando mais complexo seria o de definir que um ponto de ação, exibindo apenas o modelo de ação atrai um segundo ponto especificado.

`1 BALL ATTO 2`
[PointID] [ViewMode] [Opcode] [PointIDWaited]

Um conjunto de comandos pode ser agrupado formando uma configuração de um cenário específico. Este agrupamento pode ser chamado de estado e será definido na seção 5.

V. Estados e Máquinas de Estados

Estado, num aplicativo de realidade aumentada baseado em pontos de ação, pode ser definido como um conjunto de pontos que possuem suas ações e características definidas.

Para definir estas ações e características, são utilizados os comandos descritos na seção 4.

Um estado com quatro pontos pode ter quatro comandos, como o exemplo da Figura 7(a), e exemplificado na Figura 7(b).

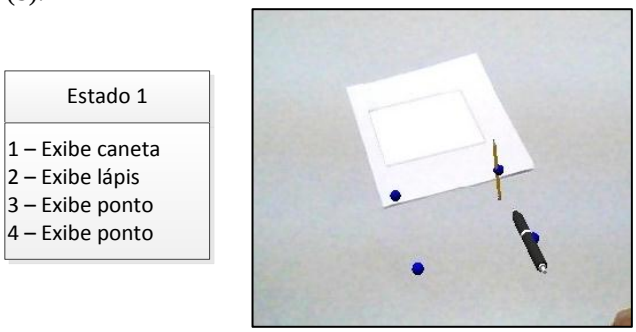


Figura 7(a) - Estado com 4 pontos. Figura 7(b) - Exemplo do estado com quatro pontos.

Uma aplicação pode ter apenas um estado ou possuir vários estados. A utilização de vários estados permite a possibilidade de uma aplicação dinâmica.

A interligação destes estados é semelhante conceitualmente às máquinas de estado propostas por Moore ou Mealy, onde a escolha do estado seguinte é feita, a partir de uma condição a ser aceita. Contudo, as máquinas de Moore ou Mealy tratam objetos binários, numa aplicação de realidade aumentada podemos tratar sucessos de ações comportamentais.

Por exemplo, definimos um ponto como um ponto de atração, que atrai um específico segundo ponto. Se este segundo ponto é levado ao ponto de atração a ação é realizada com sucesso, e assim direcionar a execução do programa para outro estado que reconfigura as ações e características dos pontos de ação.

A Figura 8 exibe uma "lógica" possível de organização dos estados, exemplificando que, partindo de um estado inicial, é possível fazer diversas ações que podem se juntar em estados intermediários. No exemplo, o estado 1 pode ir para os estados 2, 3 ou 4. Dependendo de uma situação, os estados 4 e 5 podem ficar num ciclo. Se algum evento ocorrer, este ciclo pode ser quebrado e ir para o estado 3. Os estados 2 e 3 por sua vez vão para o estado 6 que retorna para o estado 1.

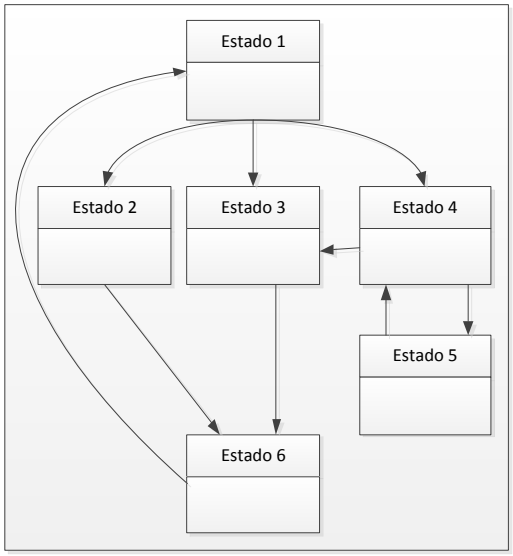


Figura 8 - Lógica de estados

Esta composição de comportamento, em estados (conjuntos de comandos) e máquinas de estados (conjunto de estados), permite ao criador de aplicações de realidade aumentada inserir componentes comportamentais sem precisar entrar no código.

VI. Inserindo nos comandos informações de estado.

Para que haja mudança de estados, é necessário inserir nos comandos informações sobre quais são os próximos estados. Desta forma, adicionando mais um argumento na composição do comando que é o próximo estado, tem-se a nova estrutura dada por:

- Identificador do ponto
- Modo de exibição
- Tipo de operação (*OpCode*)
- Argumentos

- Próximo estado

Os exemplos de comandos da seção 4 poderiam agora ser representados desta forma:

1 OBJECT STAT 4
[PointID] [ViewMode] [Opcode] <NextState>

1 BALL ATTO 2 8
[PointID] [ViewMode] [Opcode] [PointIDWaited] <NextState>

O primeiro comando indica que o ponto de ação direciona para o estado 4 e o segundo comando indica que o ponto de ação direciona para o estado 8.

VII. EXEMPLO

Um exemplo simples é a composição de um cenário sobre uma infraestrutura com seis itens, sendo três de movimentação e três de "encaixe". Ilustrativamente os três itens serão uma caneta, um lápis e uma borracha, e os três pontos de encaixe serão representados por bolas.

Cada objeto pode ser encaixado em apenas um lugar. A organização dos pontos fica desta forma:

Ponto 1	Exibe o modelo da caneta e o local do ponto de ação. Este pode ser movimentado.
Ponto 2	Exibe o modelo do lápis e o local do ponto de ação. Este pode ser movimentado.
Ponto 3	Exibe o modelo da borracha e o local do ponto de ação. Este pode ser movimentado
Ponto 4	Atrai o ponto 1 se houver colisão.
Ponto 5	Atrai o ponto 2 se houver colisão.
Ponto 6	Atrai o ponto 3 se houver colisão.

O procedimento de montagem, que é utilizado neste exemplo, inicia-se pela montagem da caneta ou do lápis, não importando a ordem do primeiro objeto a ser encaixado. Quando um dos dois objetos é encaixado o próximo estado permite apenas o encaixe do lápis ou da caneta, dependendo da primeira ação realizada. Após ambos (caneta e lápis) montados é permitida a montagem da borracha, e assim, finalizar o procedimento. A visualização dos estados desta operação está na Figura 9, com um diagrama dos estados da montagem e as ações que levam para cada estado.

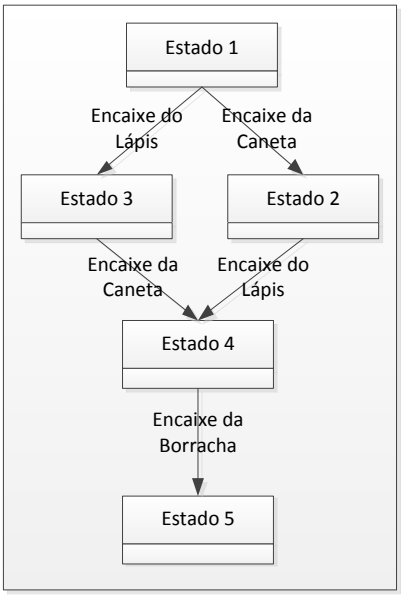
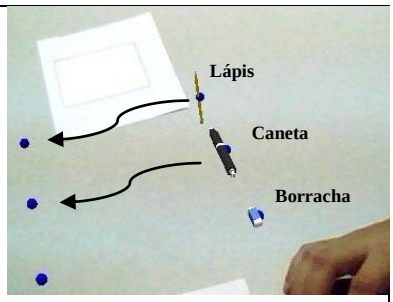


Figura 9 - Lógica dos estados do exemplo

A descrição da máquina de estados para este cenário de pontos, na forma de comandos, é exemplificada na sequência de passos das Figuras 10(a) até 10(e). Na sequência, estão os estados com os comandos e uma imagem de como ficam os pontos e modelos.

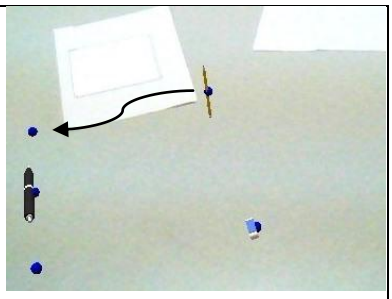
```
BEGIN_STATE 1
1 BOTH DRGF
2 BOTH DRGF
3 BOTH DRGF
4 BALL ATTO 1 2
5 BALL ATTO 2 3
6 BALL STAT
END_STATE
```



No estado inicial, são vistos os seis pontos de ação, três deles são móveis (DRGF), exibindo os modelos de caneta, lápis e borracha. Movendo a "caneta" (Ponto 1) para o ponto 4, que esta configurado para atrair o ponto 1, passa para o estado 2. Movendo o "lápis" (Ponto 2) para o ponto 5, que esta configurado para atrair o ponto 1, passa para o estado 3. A "borracha" (Ponto 3) não é esperada por nenhum ponto de ação.

Figura 10(a) – Estado Inicial

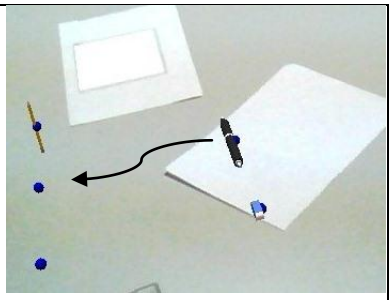
```
BEGIN_STATE 2
1 OBJECT STAT
2 BOTH DRGF
3 BOTH DRGF
4 HIDE STAT
5 BALL ATTO 2 4
6 BALL STAT
END_STATE
```



O segundo estado ocorre com a bem sucedida atração do ponto 1 pelo ponto 4, no estado 1. Neste estado, é desativada a ação e a exibição do ponto 4 e a movimentação do ponto 1.

Figura 10(b) – Segundo estado

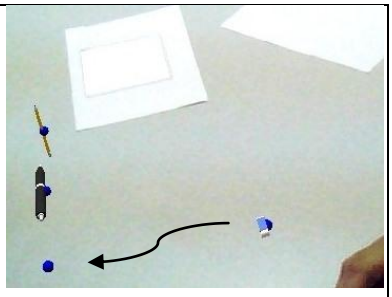
```
BEGIN_STATE 3
1 BOTH DRGF
2 OBJECT STAT
3 BOTH DRGF
4 BALL ATTO 1 4
5 HIDE STAT
6 BALL STAT
END_STATE
```



O terceiro estado ocorre com o sucesso da atração do ponto 2 pelo ponto 5, no estado 1. Neste estado, é desativada a ação e a exibição do ponto 5 e a movimentação do ponto 2.

Figura 10(c) – Terceiro estado

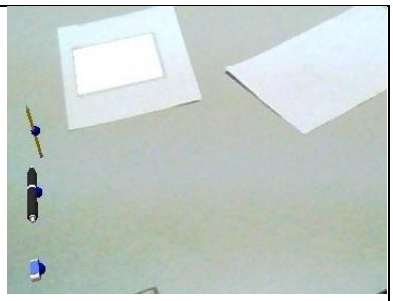
```
BEGIN_STATE 4
1 OBJECT STAT
2 OBJECT STAT
3 BOTH DRGF
4 HIDE STAT
5 HIDE STAT
6 BALL ATTO 3 5
END_STATE
```



O quarto estado ocorre com o sucesso de atração do ponto 2 pelo ponto 5, no estado 3 ou pelo sucesso de atração do ponto 1 pelo ponto 4, no estado 2. Neste estado, são novamente desativadas as ações e exibições dos pontos de atração e a movimentação do ponto atraído. Neste estado habilita-se a atração do ponto 3 pelo ponto 6.

Figura 10(d) – Quarto estado

```
BEGIN_STATE 5
1 OBJECT STAT
2 OBJECT STAT
3 OBJECT STAT
4 HIDE STAT
5 HIDE STAT
6 HIDE STAT
END_STATE
```



O quinto estado ocorre com o sucesso da atração do ponto 3 pelo ponto 6, no estado 4, finalizando a máquina de estado.

Figura 10(e) – Quinto estado

Neste exemplo, foram usados os seguintes comandos:

- **OBJECT STAT**: Define que o ponto de ação está estático e exibe apenas o objeto.
- **BOTH DRFG**: Define que o ponto de ação pode ser movido e tanto o ponto de ação, quanto o objeto, estão visíveis.
- **HIDE STAT**: Define que o ponto de ação está estático e invisível.
- **BALL ATTO**: Define que o ponto de ação atrai um ponto e está exibindo apenas a representação de onde está o ponto de ação.
- **BALL STAT**: Define que o ponto de ação está estático e está exibindo apenas a representação de onde está o ponto de ação.

VIII. CONCLUSÕES

Este artigo demonstra a inclusão de uma camada de comportamento para controlar pontos de ação, mostrando que, para incluir esta camada sem utilizar programação, é possível criar um subconjunto de possíveis comportamentos e descrevê-los na forma de comandos. A união dos comandos possibilita a criação de estados, que são descritores de características e ações num certo momento da execução da aplicação. A variação de um estado para outro permite controlar sequências de eventos e como cada ponto de ação se comporta em cada momento.

A inclusão de comportamento de pontos de ação adiciona para o desenvolvedor de realidade aumentada uma nova possibilidade para gerar diversas aplicações, pois geralmente os programas de autoria concentram-se nos tipos de modelos a serem exibidos ou em comportamentos do marcador, que é item de infraestrutura e não o conteúdo ou estrutura de uma aplicação de realidade aumentada.

Um segundo passo para este trabalho é a criação de um programa com interface gráfica, onde ao invés de descrever os comandos manualmente o desenvolvedor arrasta “caixas” e define os parâmetros dos comandos mais intuitivamente, deixando para o programa a tarefa de montar a máquina de estados e os comandos. Esta camada de modelagem de comportamento é um passo para o desenvolvimento de um

aplicativo de autoria para hiper-realidade [9], introduzindo "inteligência" à aplicação para estimular e responder a situações em sua execução.

OBSERVAÇÕES

Este trabalho foi realizado no âmbito do Projeto "Ambiente Temático Interativo com Realidade Aumentada", financiado pelo CNPq (Proc. 58842/2009-7) e FAPEMIG (Proc. APQ-03643-10).

REFERÊNCIAS

[1] C. Kirner; R. Siscouto. "Fundamentos de Realidade Virtual e Aumentada". In: Kirner, C.; Siscouto, R.. (Org.). "Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações". 1 ed. Porto Alegre – RS: Sociedade Brasileira de Computação - SBC, 2007, v. 1, p. 2-21.

[2] L. C. Botega; P. E. Cruvinel. Realidade Virtual: Histórico, Conceitos e Dispositivos. Aplicações de Realidade Virtual e Aumentada. XI Simpósio de Realidade Virtual e Aumentada. Porto Alegre: SBC - Sociedade Brasileira de Computação, 2009. Pp. 8-30.

[3] R. L. Moreno. "Máquinas de Estado". Disponível em: <www.microeletronica.unifei.edu.br/EAM120/Maq%20Estados1.pdf > Acesso em: 20 de julho de 2011.

[4] NINTENDO 3DS. Disponível em: <http://www.nintendo.com/3ds>. Acesso em 30 de agosto de 2011

[5] GE. SmartGrid Augmented Reality Digital Hologram. Disponível em: <http://ge.ecomagination.com/smartgrid/#/augmented_reality>. Acesso em 30 de agosto de 2011

[6] R. Santin, "Sistema de Autoria Colaborativa com Realidade Aumentada", Dissertação de Mestrado em Ciência da Computação, Universidade Metodista de Piracicaba, 2008.

[7] METAIO. metaio | Augmented Reality 3D – Unifeye Design. Disponível em: < http://www.metaio.com/> Acesso em 20 de julho de 2011.

[8] C. Kirner; R. Santin. "Interaction, Collaboration and Authoring in Augmented Reality Environments". In: XI Symposium on Virtual and Augmented Reality - SVR'2009, 2009, Porto Alegre - RS. "Proceedings of the IX Symposium on Virtual and Augmented Reality". Porto Alegre - RS : Sociedade Brasileira de Computação - SBC, 2009.

[9] C. Kirner; R. Tori. "Introdução à Realidade Virtual, Realidade Misturada e Hiper-realidade". In: Claudio Kirner; Romero Tori. (Ed.). Realidade Virtual: Conceitos, Tecnologia e Tendências. 1ed. São Paulo, 2004, v. 1, p. 3-20.