

Handling Occlusions in Augmented Reality for Mobile Devices

Tratamento de Oclusão em Realidade Aumentada para Dispositivos Móveis

Silvio R. R. Sanches¹, Vitor V. G. Silva¹, Antonio C. Sementille², Cléber G. Corrêa¹, Claiton Oliveira¹

¹Universidade Tecnológica Federal do Paraná, Cornélio Procopio, Brasil
silviosanches, clebergimenez, claitonoliveira{ @utfpr.edu.br}, vitorsilva.2014@alunos.utfpr.edu.br

²Universidade Estadual Paulista “Julio de Mesquita Filho”, Bauru, Brasil
antonio.semente@unesp.br

Abstract—The coherent display of the spatial relationship between real and virtual elements is a desirable feature for realistic Augmented Reality (AR) systems. The system must be able to automatically estimate the depths of the real and virtual elements that compose the scene and display in the foreground of the frame the element closest to the camera. In other words, real and virtual elements must hide (totally or partially) each other according to their depths in the frame. This paper presents a method for occlusion treatment of scene elements for mobile applications. The solution presented uses pixel proportions, relative to the total pixels of the frame, occupied by the real and virtual elements to estimate their depths. Then, a skin color model based segmentation method is applied to extract the real element in frames in which it is in the foreground of the scene and, as a result, must overlap the virtual object. The results of the evaluation of the proposed method showed its superiority to those found in the literature when a set of important features are compared.

Keywords— Augmented Reality, Mobile Devices, Handling Occlusion

Resumo—A exibição coerente do relacionamento espacial entre elementos reais e virtuais é uma característica desejável nos sistemas realistas de Realidade Aumentada (RA). O sistema deve ser capaz de estimar automaticamente as profundidades dos elementos reais e virtuais que compõem a cena e exibir no primeiro plano do quadro aquele que estiver mais próximo da câmera. Em outras palavras, os elementos reais e virtuais devem ocultar (total ou parcialmente) uns aos outros conforme suas profundidades. Neste artigo é apresentado um método para tratamento da oclusão voltado para aplicações executadas em dispositivos móveis. A solução apresentada utiliza as proporções de pixels, em relação ao total de pixels do quadro, ocupadas pelos elementos real e virtual para estimar suas profundidades. Em seguida, um método de segmentação baseado em modelo de tons de pele é aplicado para extrair o elemento real nos quadros em que esse elemento encontra-se no primeiro plano da cena e, em consequência disso, deve sobrepor o objeto virtual. Os resultados da avaliação do método proposto mostraram sua superioridade em relação aos encontrados na literatura quando um conjunto de características importantes são comparadas.

Palavras-chave— Realidade Aumentada, Dispositivos Móveis, Tratamento de Oclusão

I. INTRODUÇÃO

A utilização de marcadores tem se mostrado uma solução eficiente para resolver o problema do registro 3D em ambientes de Realidade Aumentada (RA) [1]. Tanto as aplicações desenvolvidas para *desktops* quanto as que executam em dispositivos móveis utilizam esse recurso, normalmente auxiliadas por bibliotecas de software [2]–[4] que trazem implementações otimizadas de algoritmos que permitem identificar a distância, a posição e a orientação desses marcadores. Em seguida, os objetos virtuais são inseridos, a cada quadro, sobrepostos a eles.

Uma limitação importante das aplicações que utilizam esse recurso para o registro 3D é a impossibilidade de realizar o tratamento da oclusão entre os elementos reais e virtuais que compõem o ambiente [5]. Nessas aplicações, um elemento real normalmente não sobrepõe um virtual, ainda que o primeiro esteja mais próximo da câmera que o segundo. Essa limitação ocorre porque não existe informação sobre a profundidade dos elementos reais e, como consequência, o sistema pode gerar cenas incoerentes que – em termos de psicologia cognitiva – podem confundir os usuários [6].

Como pode ser observado na Fig. 1, essa situação ocorre com frequência quando um marcador é reconhecido mesmo que exista um elemento real obstruindo parcialmente sua visibilidade. Os métodos recentes de reconhecimento de marcadores são capazes de estimar a posição e orientação do marcador mesmo quando sua visibilidade não é total [7], [8]. No entanto, as bibliotecas de software usualmente renderizam o objeto virtual sempre no primeiro plano da cena (Fig. 1b) Um sistema de RA deve ser capaz de tratar a oclusão entre os elementos reais e virtuais automaticamente [9], exibindo no primeiro plano aquele que se encontra mais próximo da câmera de vídeo.

Neste trabalho é apresentado um método para tratamento de oclusão nas aplicações de RA baseadas em marcadores e executadas em dispositivos móveis. O cenário típico dessas

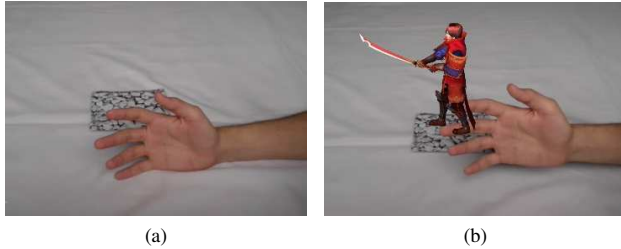


Figura 1. Representação do problema do conflito da oclusão em RA. No quadro de vídeo exibido em (a) o elemento real (mão do usuário) está posicionado mais próximo da câmera que o marcador. Quando o elemento virtual é gerado ocorre um conflito de oclusão, uma vez que o objeto virtual associado ao marcador é exibido em primeiro plano.

aplicações é o usuário manipulando o marcador sobre o qual um objeto virtual é inserido. Nesse cenário, apenas as mãos (ou uma das mãos) e parte do braço do usuário são visualizados pela câmera de vídeo. Este artigo está organizado da seguinte forma. Na seção II são apresentados os trabalhos que representam o estado-da-arte no tratamento de oclusão em sistemas de RA. O processo de desenvolvimento do método de tratamento de oclusão é apresentado na seção III. Os resultados obtidos são discutidos na seção V e, finalmente, na seção VI são apresentadas as conclusões e perspectivas de trabalhos futuros.

II. TRABALHOS RELACIONADOS

O tratamento de oclusão é um problema tratado em vários tipos de sistemas de RA, desde os baseados em equipamentos sofisticados até os que utilizam equipamentos convencionais. Nesses sistemas, são várias as soluções para resolver o problema do registro 3D. Segundo Tian *et al.* [9], os métodos que tratam a oclusão podem ser divididos em três categorias: reconstrução 3D, baseados em mapas de profundidade e baseados na identificação do contorno de elementos reais.

Os métodos do primeiro grupo, os que reconstróem o ambiente real, alcançam a oclusão correta comparando um modelo 3D, gerado após um mapeamento do ambiente real, com a profundidade dos objetos virtuais [9] que é conhecida do sistema. Esses métodos são normalmente custosos computacionalmente e, por esse motivo, a maioria deles são utilizados em aplicações *desktop*.

Entre as soluções mais recentes que se baseiam em reconstrução 3D, pode ser destacada, por exemplo, a abordagem apresentada por Lepetit e Berger [10], em que a cena é tridimensionalmente reconstruída a partir de informações obtidas do rastreamento da silhueta dos elementos virtuais utilizando apenas uma câmera. O método requer a intervenção do usuário para ajustes, o que o torna não recomendável para aplicações de tempo real. Gheorghe *et al.* [11] desenvolveram um método capaz de tratar a oclusão em tempo real baseado na representação geométrica do ambiente real que é obtida em uma etapa *offline* e inserida no ambiente virtual durante a execução da aplicação. Tian *et al.* [9] propuseram um método que usa uma câmera RGB-D (*red, green, blue* e *depth*) para

obter as coordenadas 3D da cena em uma etapa *offline*. Essas coordenadas são usadas para reconstruir o ambiente real. Utilizando uma GPU (*Graphics Processing Unit*), a profundidade dos objetos virtuais e os elementos reais reconstruídos são alinhados para obter a oclusão correta entre esses elementos. Uma evolução desse método foi apresentado por Tian *et al.* [12], em que a acurácia do mapa de profundidade obtido da câmera RGB-D é melhorado por meio de técnicas de *ray casting*.

Os métodos baseados em mapas de profundidade, diferentemente dos baseados em reconstrução 3D, não criam um modelo tridimensional do ambiente. Esses métodos obtêm a oclusão correta comparando os valores de profundidade dos pixels dos elementos reais com a profundidade dos objetos virtuais, que são conhecidas da aplicação [9].

No trabalho de Zhu *et al.* [13], dois núcleos gaussianos foram utilizados para analisar pixels vizinhos gerando informações que são combinadas com um mapa de profundidade, obtido por câmeras estereoscópicas, por meio de um *framework* probabilístico. Essa informação é utilizada para extrair elementos da cena e possibilitar o tratamento de oclusão. Leal-Meléndrez *et al.* [14] apresentaram um método simples para tratamento de oclusão baseado em mapas de profundidade. Os autores basicamente alinham as coordenadas tridimensionais do objeto virtual fornecida pelo *frame buffer* com o mapa de profundidade obtido pelo sensor. Os resultados mostram erros de segmentação do elemento real quando esse elemento deve ser exibido no primeiro plano da cena.

Dong *et al.* [15] resolvem o problema da oclusão em ambientes de RA utilizando um mapa de profundidade obtido de sensores TOF (*Time of Flight*). O método manipula o *frame buffer* para estimar valores de profundidade e exibir a oclusão correta. Apesar do sistema não restringir o ponto de vista do usuário, o tratamento de oclusão é desabilitado quando o sensor TOF está em movimento. O sistema depende de *hardware* sofisticado e pode ser executado em plataforma *desktop*. Du *et al.* [16] propuseram um algoritmo em que a solução é um encaixe de arestas formulado como uma otimização discreta, que refina a combinação entre os dados RGB e de profundidade dos objetos virtuais. O sistema não executa em tempo real quando plataformas móveis são utilizadas. Narducci *et al.* [17] apresentaram uma solução voltada para equipamentos *video see-through*, que permite estimar profundidade dos elementos reais e combiná-la com os valores de profundidade dos elementos virtuais presentes no *frame buffer*. O método foi implementado em um sistema que executa em plataforma *desktops*.

O terceiro grupo de métodos para tratamento de oclusão são os baseados na identificação de contornos dos elementos reais. Nesses métodos, a oclusão correta é estimada sem a utilização de sensores que capturam o mapa de profundidade da cena e, em consequência disso, a profundidade desses elementos deve ser estimada de outras formas.

O método apresentado por Tian *et al.* [18] obtém as regiões ocupadas pelos objetos reais e estima seus contornos em uma etapa *offline*. Esse contorno é utilizado durante o processa-

mento *online* em que os elementos reais são redesenhados na cena aumentada nos momentos em que devem ser exibidos em primeiro plano. O método é apresentado em um sistema para *desktop* e pode apresentar problemas no tratamento da oclusão quando elementos reais mais espessos estão presentes na cena. Fukiage *et al.* [19] propuseram uma solução que não estima a posição dos objetos reais e não requer segmentação de primeiro plano precisa. O método controla a transparência dos elementos reais para que em determinados momentos possam ser percebidos pelo usuário sendo obstruído pelo objeto virtual. O algoritmo apenas reduz as informações de oclusão contraditórias em cenas de ambientes AR e, devido ao custo computacional requerido, pode ser executado apenas em plataforma *desktop*.

Sanches *et al.* [5] apresentaram um método que utiliza marcadores fiduciais fixados nos elementos reais para obter suas posições (inclusive suas profundidades). Para realizar o controle de oclusão, esses valores são comparados com os valores das posições dos objetos virtuais, que também são obtidas de marcadores. O método recupera do *frame buffer* a região do quadro ocupada pelo objeto virtual. Nos quadros em que o elemento real deve ser apresentado em primeiro plano, um método de segmentação é aplicado apenas na região do objeto virtual. O método foi aplicado em sistemas que executam em *desktops* [20], [21]. Frikha *et al.* [22] propuseram um método de tratamento de oclusão que rastreia instrumentos cirúrgicos e os posicionam no ambiente de RA. O método estima a distância entre um objeto real (instrumento cirúrgico) e a câmera usando um modelo de projeção em perspectiva. O método foi implementado em um sistema *desktop*.

III. DESENVOLVIMENTO DO MÉTODO

A solução proposta neste trabalho tem como objetivo possibilitar o tratamento da oclusão nas aplicações de RA baseadas em marcadores e executadas em dispositivos móveis. O cenário típico dessas aplicações é o usuário manipulando o marcador sobre o qual um objeto virtual é inserido. Nesse cenário, apenas as mãos (ou uma única mão) e parte do braço do usuário são visualizados pela câmera de vídeo. A Fig. 2 mostra uma visão geral da abordagem proposta.

Como pode ser observado, uma etapa ocorre *offline*, na qual parâmetros utilizados no tratamento de oclusão são estimados. O tratamento da oclusão é realizado em tempo real (*online*) e possibilita a geração de cenas de RA coerentes em relação a profundidade dos elementos da cena. A etapa de calibração é discutida nesta seção e o tratamento da oclusão, realizado *online*, é detalhado seção IV.

A etapa executada *offline* consiste na calibração necessária para encontrar a proporção da região ocupada no quadro pela mão do usuário e a proporção ocupada no mesmo quadro pelo objeto virtual com o qual esse usuário interage. Quando o elemento real (a mão) é identificado, o sistema considera apenas parte desse elemento, delimitado por pontos chaves, para o cálculo da proporção.

Para a identificação dos pontos chaves que delimita a área da mão do usuário que deve ser considerada no tratamento da

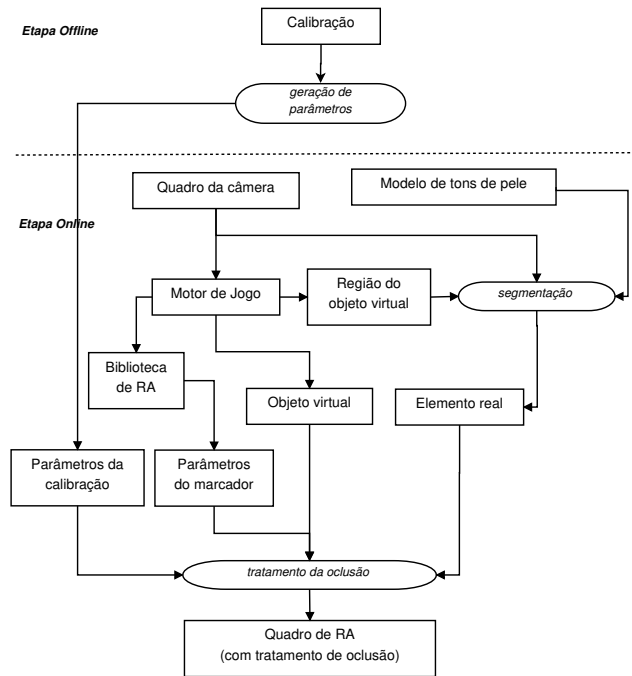


Figura 2. Visão geral do método de tratamento de oclusão. Uma etapa de calibração é executada *offline* para estimar a proporção entre o elemento real sob o qual a oclusão é tratada e o objeto virtual. Na etapa *online*, os parâmetros obtidos na etapa de calibração são utilizados para identificar qual dos elementos deve ser exibido no primeiro plano da cena.

occlusão é realizado inicialmente o reconhecimento das bordas do elemento real. Em seguida, identifica-se o casco convexo da região de pele para detectar pontos que representam as pontas dos dedos e membranas interdigitais (Fig 3a), que são os espaços entre os dedos [23]. O passo seguinte consiste em delimitar a área que abrange todos os pontos chaves identificados.

A região formada pelos pontos chaves interconectados – os que foram reconhecidos no quadro que está sendo processado – auxiliam a definição da área que serve como referência para identificar se o elemento real está mais próximo ou mais distante da câmera que o elemento virtual. A área de referência é representada pelo círculo verde exibido na Fig. 3a. Uma vez que os elementos real e virtual são posicionados alinhados (lado a lado) no quadro de vídeo, o usuário ativa o comando que armazena a proporção de cada elemento e estima a distância d entre eles. Nesse momento, o círculo verde torna-se vermelho (Fig. 3b e Fig. 3c).

O processo de calibração tem como objetivo obter uma constante de proporção P que possibilite estimar qual elemento (real ou virtual) encontra-se mais próximo da câmera, sem a necessidade de conhecer a profundidade do elemento real (a mão do usuário) em relação a câmera. A distância para o objeto virtual é uma informação que pode ser obtida do motor de jogo Unity3D [24]. A constante P pode ser obtida

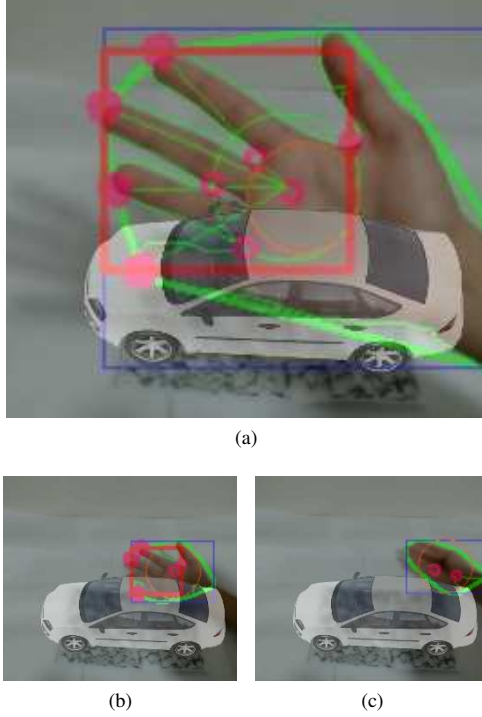


Figura 3. Identificação dos pontos chave para identificação das áreas ocupadas pelos elementos real e virtual no processo de calibração. (a) Definição da área de referência, especificando o casco convexo da região da pele com pontos representativos das pontas dos dedos e membranas interdigitais. (b) Área de referência refinada para a mão do usuário. (c) Área de referência no recolhimento dos dedos do usuário.

de acordo com a equação

$$P = \frac{A}{W*H} \quad (1)$$

onde A é a área ocupada pelo elemento real; W e H a largura e a altura da imagem, respectivamente; e d a distância entre o elemento real e o virtual.

A utilização de uma região de referência (e não de toda a região do elemento real) no processo de calibração é necessária pela própria natureza das ações realizadas pelo usuário durante a interação com a aplicação. Quando o usuário recolhe seus dedos, por exemplo, a área de referência não se altera significativamente (Fig. 3c). Dessa forma, o sistema não confunde essa ocorrência com o afastamento do elemento real.

IV. TRATAMENTO DA OCLUSÃO

A etapa executada em tempo real utiliza os parâmetros gerados na etapa de calibração para possibilitar o tratamento de oclusão. Nesta etapa, o quadro de vídeo obtido pela câmera é processado por um motor de jogo com o auxílio de uma biblioteca de RA. O motor de jogo Unity3D [24] é responsável pela inserção do objeto virtual no ambiente. Por meio dos recursos da ferramenta é possível utilizar objetos virtuais criados por diversas ferramentas de modelagem 3D. A biblioteca de RA Vuforia [2] é utilizada em conjunto com o Unity3D e adiciona ao motor de jogo funcionalidades que permitem transformar

a cena em um ambiente de RA. Utilizando as funções do Vuforia, o marcador real é rastreado quadro a quadro de forma que o objeto virtual pode ser sobreposto a esse marcador. O rastreamento do marcador em tempo real e a geração do objeto virtual sobreposto a esse marcador são desafios cujas soluções estão implementadas no Vuforia.

A. Delimitação da área de segmentação

Uma vez que o objeto virtual é inserido na cena, um quadro de vídeo, sem tratamento de oclusão, é gerado pelo sistema. Ainda que esse quadro possa apresentar conflito de oclusão, algumas de suas informações podem ser obtidas com a finalidade de auxiliar o processo de segmentação do elemento real. A segmentação do elemento real é necessária para que esse elemento, isolado do fundo original, possa ser inserido no primeiro plano da cena quando ele se encontra mais próximo da câmera que o objeto virtual.

Utilizando as funcionalidade do motor de jogo, as coordenadas dos pixels do quadro que pertencem ao objeto virtual foram obtidas. Um quadro de vídeo que mostra uma cena de RA (sem tratamento de oclusão) e um quadro correspondente que delimita a região formada pelos pixels que pertencem ao objeto virtual podem ser visualizados nas Fig. 4a e Fig. 4b, respectivamente.

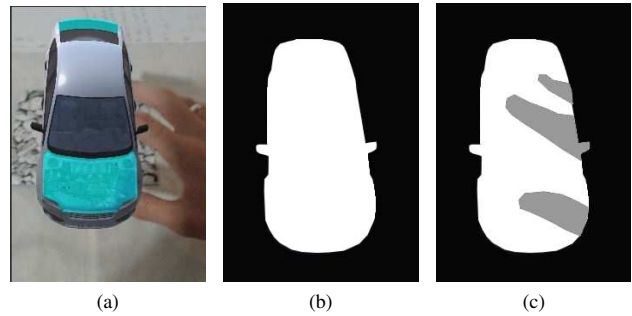


Figura 4. (a) Quadro de vídeo que mostra uma cena de RA. (b) Quadro correspondente que delimita a região ocupada pelo objeto virtual (pixels brancos). (c) Região em que o elemento real deve ser redesenhado quando esse elemento encontra-se em primeiro plano.

A região ocupada pelo objeto virtual mostrada na Fig. 4b foi delimitada com o objetivo de reduzir a área de atuação do algoritmo de segmentação utilizado para extrair o elemento real de interesse para a aplicação, no caso, a mão do usuário. Importa ressaltar que apenas nessa região (pixels brancos da Fig. 4b) é que o algoritmo de segmentação deve atuar, uma vez que na região preta deve ser preservado o fundo da cena, que são os pixels da imagem original obtida pela câmera de vídeo. A região cinza da Fig 4c representa os pixels que devem ser substituídos pelos pixels do elemento real quando este elemento estiver em primeiro plano. Na região branca da Fig. 4c devem ser exibidos os pixels pertencentes ao objeto virtual e na região preta devem ser preservados os pixels da imagem capturada pela câmera.

B. Segmentação e Tratamento da Oclusão

Uma vez que a aplicação em potencial considera que o elemento real será a mão do usuário que manipula o objeto virtual por meio do marcador, o método de segmentação utilizado se baseia na identificação de tons de pele humana [23] e atua apenas na região branca da Fig. 4b.

Inicialmente, aplica-se um pré-processamento no quadro de entrada (Fig 5a) por meio de operações de erosão e dilatação sobre a área de pele extraída pelo algoritmo de segmentação (Fig 5b). Dessa forma, minimizam-se imperfeições que podem ocorrer na imagem segmentada do elemento real.

O controle de oclusão utiliza os parâmetros obtidos na etapa de calibração para estimar a profundidade D da área de referência de acordo com a equação

$$D = \frac{P}{\frac{A}{W*H}} \quad (2)$$

onde P é a constante de proporção obtida na etapa de calibração (Equação 1), A é a área de referência ocupada pelo elemento real no quadro atual (círculo vermelho da Fig. 5c, da Fig. 5d e da Fig. 5e) e; W e H a largura e a altura do quadro, respectivamente.

Quando a distância D é menor que a distância do elemento virtual (conhecida do sistema), a área de referência está mais próxima da câmera que o elemento virtual. Nesse caso, o sistema realiza a segmentação na região branca da Fig. 4b e redesenha os pixels pertencentes ao elemento real de interesse de forma que esse elemento seja exibido em primeiro plano.

V. RESULTADOS

Para avaliar o resultado da aplicação do método em sistemas móveis de RA que utilizam marcadores fiduciais para registro 3D dos objetos virtuais, são apresentados quadros de vídeo em que o tratamento da oclusão pode ser visualizado. Além disso, algumas características consideradas importantes, definidas em Tian *et al.* [9], serviram como base para comparação do método com outros que representam o estado-da-arte.

O método proposto foi implementado utilizando o Unity3D [24] e executado em um dispositivo móvel Moto Z play, com 32 GB de memória RAM, sistema operacional Android 8.0, equipado com um processador de 2 GHz (Octa Core). A aplicação alcança uma taxa média de 6,7 quadros

por segundo. A Fig. 6 mostra quadros de vídeo em que o tratamento da oclusão permite visualizar o elemento real em primeiro plano quando esse elemento encontra-se mais próximo da câmera que o objeto virtual.

Na Fig. 6a e na Fig. 6e podem ser visualizados dois modelos de objetos virtuais – modelos 3D de um automóvel e de um avatar, respectivamente – renderizados sobre marcadores reais. Nesses quadros, os elementos reais sobre os quais o tratamento da oclusão é realizado ainda não estão próximos o suficiente do marcador para que o tratamento da oclusão entre em funcionamento. Na Fig. 6b e na Fig. 6f, existem elementos reais visíveis na cena que são sobrepostos pelos objetos virtuais. Nesses quadros, no entanto, as distâncias dos elementos reais (a mão do usuário) em relação a câmera são maiores que as dos elementos virtuais. Nessas situações, os objetos virtuais devem ser exibidos em primeiro plano e, por esse motivo, a cena apresentada não apresenta qualquer conflito de oclusão.

No momento em que o elemento real cuja oclusão é tratada pelo sistema (a mão do usuário) se aproxima da câmera, como ocorre na Fig. 6c e na Fig. 6g, o sistema identifica esse movimento, aciona o método de segmentação, que recorta o elemento real e, em seguida, sobrepõe a imagem recortada da mão do usuário sobre a imagem do modelo virtual. Na Fig. 6d e na Fig. 6h os elementos reais continuam mais próximos da câmera que os objetos virtuais e, por esse motivo, continuam sendo exibidos no primeiro plano da cena. Em todos os quadros exibidos na Fig. 6 não existe conflito de oclusão.

Para verificar a superioridade do método proposto quando comparado com outros que representam o estado-da-arte foram considerados seis características [9], descritas na Tab. I. A categoria conforme a qual o método pertence também é apresentada na Tab. I, conforme Tian *et al.* [9]. Primeiramente, foi analisado o tipo de cena de RA que a solução pode ser utilizada. Essa característica é importante porque alguns métodos tem aplicação restrita a ambientes estáticos. A segunda característica trata da necessidade de haver ou não alguma restrição quanto ao ponto de vista da câmera de vídeo. A terceira característica se relaciona com a exigência de alguma condição adicional, normalmente relacionada com a utilização

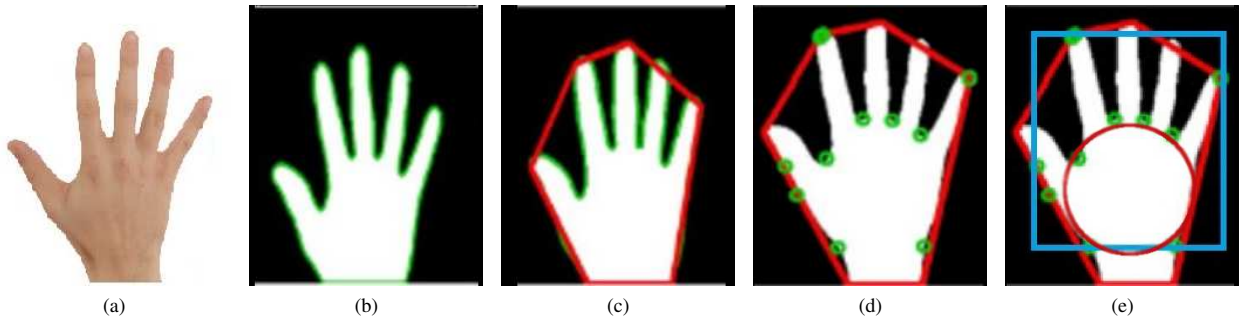


Figura 5. (a) Imagem da mão do usuário segmentada. (b-e) Etapas do cálculo da profundidade do elemento real.

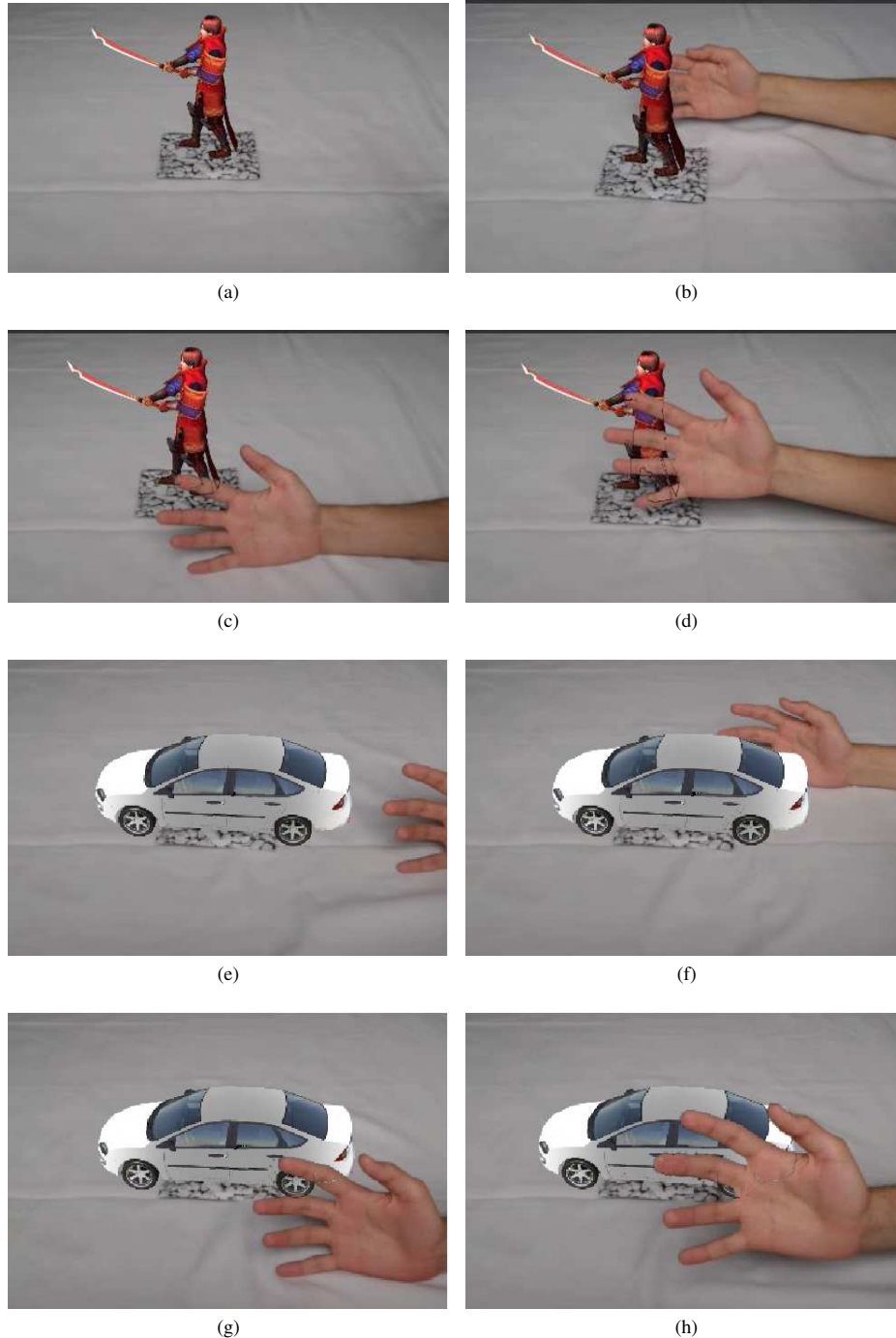


Figura 6. Tratamento de oclusão. Objeto Virtual sobreposto ao marcador real (a,e). Elemento real no fundo da cena, mais distante da câmera que o objeto virtual (b,f). Elemento real exibido em primeiro plano quando está mais próximo da câmera que o objeto virtual (c,d,g,h).

de sensores de profundidade utilizados para auxiliar a captura. A quarta característica avalia se a solução efetivamente trata a oclusão ou apenas possibilita exibir o elemento real em primeiro plano.

Finalmente, as duas últimas características utilizadas para

comparar o método proposto com métodos recentes considera, respectivamente, a capacidade do método de executar em tempo real e a plataforma para a qual o método foi implementado.

Como pode ser observado, o método proposto possui res-

Tabela I

COMPARAÇÃO DO MÉTODO PROPOSTO COM OUTROS QUE REPRESENTAM O ESTADO-DA-ARTE NO TRATAMENTO DE OCLUSÃO EM SISTEMAS DE RA.

Método	Categoria	Cena estática/dinâmica	Ponto de vista restrito/arbitrário	Condição adicional	Oclusão automática	Tempo real	Plataforma
Lepetit e Berger [10]	Reconstrução	Estática	Restrito	Não	Não	Sim	Desktop
Sanches <i>et al.</i> [5]	Contorno	Dinâmica (com restrições)	Arbitrário	Um marcador fixado no elemento real	Sim	Sim	Desktop
Dong <i>et al.</i> [15]	Profundidade	Dinâmica	Restrito	Câmera TOF	Sim	Sim	Desktop
Tian <i>et al.</i> [9]	Reconstrução	Estática	Arbitrário	Câmera RGB-D	Sim	Sim	Desktop
Tian <i>et al.</i> [12]	Reconstrução	Dinâmica	Arbitrário	Câmera RGB-D	Sim	Sim	Desktop
Método proposto	Contorno	Dinâmica (com restrições)	Arbitrário	Não	Sim	Sim	Móvel

trições quanto ao tipo de cena em que pode ser utilizado. Isso ocorre porque a segmentação baseada em modelo de tons de pele permite extrair somente partes do elemento de interesse onde a pele humana é visível ao dispositivo que captura o vídeo. Essa restrição é aceitável em grande número de sistemas, uma vez que um cenário típico de aplicações móveis de RA consiste em um usuário interagindo com um elemento virtual sobreposto a um marcador colocado sobre alguma superfície. Nessa aplicações, apenas a mão (ou as mãos) desse usuário tem relevância quanto a necessidade do tratamento da oclusão.

Por outro lado, ao contrário dos demais encontrados na literatura, o método proposto pode ser utilizado em aplicações de RA executadas em dispositivos móveis. Isso ocorre porque o custo computacional exigido é compatível com esse tipo de plataforma e, além disso, o tratamento da oclusão se realiza ainda que não exista sensores para obtenção de mapas de profundidade. A maioria dos dispositivos móveis, sobretudo os *smartphones*, não estão equipados com sensores desse tipo.

É importante ressaltar que o método proposto apresenta algumas limitações. Uma vez calibrado, o sistema se baseia nas medidas da mão do usuário que realizou a calibração. Por exemplo, em situações em que um adulto realiza a calibração e uma criança utiliza o sistema sem repetir esse processo, o tratamento da oclusão pode perder sua precisão. Nesse caso, a calibração deve ser refeita pelo novo usuário para que as medidas da mão da criança sejam consideradas.

Além disso, como qualquer aplicação que realiza segmentação em ambiente não controlado [25], variações no ambiente – por exemplo, na iluminação – podem provocar falhas na segmentação do elemento real e, como consequência, prejudicar a qualidade da cena exibida. Movimentos rápidos do usuário também podem provocar problemas tanto na identificação dos pontos da mão quanto na qualidade da segmentação. A segmentação baseada em tom de pele também falha em situações em que o usuário está usando luvas.

Ainda que os marcadores oferecidos pelo Vuforia, que são baseados em texturas, sejam tolerantes a obstrução, a aplicação não os detecta quando são quase totalmente obstruídos por elementos da cena real, como a própria mão do usuário. Nesse caso, o elemento virtual não é exibido.

VI. CONCLUSÃO

Neste artigo foi apresentado um novo método, que não utiliza mapas de profundidade, para tratamento de oclusão em

sistemas de RA. A solução proposta se baseia inicialmente em um processo de calibração, que ocorre *offline*, e permite encontrar as proporções, em relação a quantidade de pixels que ocupam no quadro de vídeo, dos elementos real e virtual. Essa informação é utilizada durante a execução da aplicação para estimar qual desses elementos encontra-se mais próximo da câmera e, em consequência disso, deve ser exibido no primeiro plano da cena. Uma vez identificada a profundidade dos elementos, o tratamento da oclusão é realizado utilizando um método de segmentação baseado na identificação da cor da pele para extrair o elemento real de interesse quando esse elemento encontra-se no primeiro plano da cena e, por esse motivo, deve sobrepor o objeto virtual. Desse modo, um sistema de RA que utiliza o método proposto pode exibir cenas de RA livres de conflitos de oclusão.

Comparado com abordagens que representam o estado-da-arte, o método se mostrou o único desenvolvido para aplicações móveis de RA, ainda que sua utilização seja restrita a cenários específicos em que um usuário utiliza suas mãos para interagir com o objeto virtual sobreposto ao marcador. Esse cenário é típico para um número significativo de aplicações de RA, principalmente as baseadas em marcadores. Como trabalhos futuros, pretende-se encontrar formas de extrair o elemento real que sejam aplicáveis a objetos e partes do usuário em que a pele não esteja visível a câmera de vídeo. Métricas para avaliação dos resultados da aplicação do método proposto também devem ser adotadas, considerando precisão e acurácia de pixels e quadros em diferentes cenários (com variação da iluminação, do objeto virtual, do gesto manual para a interação humano-computador). Essas métricas tornariam possível a comparação com métodos similares. Por meio da solução aqui proposta, os autores acreditam ter contribuído para o desenvolvimento de aplicações de RA baseadas em dispositivos móveis.

REFERÊNCIAS

- [1] V. Mondéjar-Guerra, S. Garrido-Jurado, R. M. noz Salinas, M. J. Marín-Jiménez, and R. Medina-Carnicer, "Robust identification of fiducial markers in challenging conditions," *Expert Systems with Applications*, vol. 93, pp. 336 – 345, 2018.
- [2] PTC Inc., "Vuforia engine," 2018, acessado em junho. 2018. [Online]. Disponível: <https://www.vuforia.com/>
- [3] OpenSceneGraph., "The openscenegraph project website," 2018, acessado em julho. 2018. [Online]. Disponível: <http://www.openscenegraph.org/>
- [4] D. Khan, S. Ullah, D. M. Yan, I. Rabbi, P. Richard, T. Hoang, M. Billingham, and X. Zhang, "Robust tracking through the design of

- high quality fiducial markers: An optimization tool for artoolkit," IEEE Access, vol. 6, pp. 22 421–22 433, 2018.
- [5] S. R. R. Sanches, D. M. Tokunaga, V. F. Silva, A. C. Sementille, and R. Tori, "Mutual occlusion between real and virtual elements in augmented reality based on fiducial markers," in 2012 IEEE Workshop on the Applications of Computer Vision (WACV), Jan 2012, pp. 49–54.
 - [6] A. B. Sekuler and S. E. Palmer, "Perception of partly occluded objects: a microgenetic analysis," Journal of Experimental Psychology: General, vol. 121, no. 1, pp. 95–111, 1992.
 - [7] F. Bergamasco, A. Albarelli, E. Rodola, and A. Torsello, "Rune-tag: A high accuracy fiducial marker with strong occlusion resilience," in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, june 2011, pp. 113–120.
 - [8] A. Sagitov, K. Shabalina, R. Lavrenov, and E. Magid, "Comparing fiducial marker systems in the presence of occlusion," in 2017 International Conference on Mechanical, System and Control Engineering (ICMSC), May 2017, pp. 377–382.
 - [9] Y. Tian, Y. Long, D. Xia, H. Yao, and J. Zhang, "Handling occlusions in augmented reality based on 3D reconstruction method," Neurocomputing, vol. 156, pp. 96–104, 2015. [Online]. Disponível: <http://dx.doi.org/10.1016/j.neucom.2014.12.081>
 - [10] V. Lepetit and M. O. Berger, "A semi-automatic method for resolving occlusion in augmented reality," in Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662), vol. 2, 2000, pp. 225–230.
 - [11] C. Gheorghe, D. Rizzotti, F. Tièche, F. Carrino, O. Abou Khaled, and E. Mugellini, "Occlusion management in augmented reality systems for machine-tools," in Virtual, Augmented and Mixed Reality, R. Shumaker and S. Lackey, Eds. Cham: Springer International Publishing, 2015, pp. 438–446.
 - [12] Y. Tian, X. Wang, H. Yao, J. Chen, Z. Wang, and L. Yi, "Occlusion handling using moving volume and ray casting techniques for augmented reality systems," Multimedia Tools and Applications, Sep 2017. [Online]. Disponível: <https://doi.org/10.1007/s11042-017-5228-2>
 - [13] J. Zhu, Z. Pan, C. Sun, and W. Chen, "Handling occlusions in video-based augmented reality using depth information," Comput. Animat. Virtual Worlds, vol. 21, no. 5, pp. 509–521, Sep. 2010.
 - [14] J. A. Leal-Meléndrez, L. Altamirano-Robles, and J. A. Gonzalez, "Occlusion handling in video-based augmented reality using the kinect sensor for indoor registration," in Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, J. Ruiz-Shulcloper and G. Sanniti di Baja, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 447–454.
 - [15] S. Dong, C. Feng, and V. R. Kamat, "Real-time occlusion handling for dynamic augmented reality using geometric sensing and graphical shading," Journal of Computing in Civil Engineering, vol. 27, no. 6, pp. 607–621, 2013.
 - [16] C. Du, Y. L. Chen, M. Ye, and L. Ren, "Edge snapping-based depth enhancement for dynamic occlusion handling in augmented reality," in 2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Sept 2016, pp. 54–62.
 - [17] F. Narducci, S. Ricciardi, and R. Vertucci, "Enabling consistent hand-based interaction in mixed reality by occlusions handling," Multimedia Tools and Applications, vol. 75, no. 16, pp. 9549–9562, Aug 2016. [Online]. Disponível: <https://doi.org/10.1007/s11042-016-3276-7>
 - [18] Y. Tian, T. Guan, and C. Wang, "An automatic occlusion handling method in augmented reality," Sensor Review, vol. 30, no. 3, pp. 210–218, 2010.
 - [19] T. Fukiage, T. Oishi, and K. Ikeuchi, "Reduction of contradictory partial occlusion in mixed reality by using characteristics of transparency perception," in 2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Nov 2012, pp. 129–139.
 - [20] T. De Gaspari, A. C. Sementille, D. Z. Vielmas, I. A. Aguilar, and J. a. F. Marar, "ARSTUDIO: A Virtual Studio System with Augmented Reality Features," in Proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry, ser. VRCAI '14. New York, NY, USA: ACM, 2014, pp. 17–25. [Online]. Disponível: <http://doi.acm.org/10.1145/2670473.2670491>
 - [21] I. A. Aguilar, A. C. Sementille, and S. R. R. Sanches, "Arstudio," Multimedia Tools and Applications, Aug 2019.
 - [22] R. Frikha, R. Ejballi, and M. Zaied, "Handling occlusion in augmented reality surgical training based instrument tracking," in 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), Nov 2016, pp. 1–5.
 - [23] J. Le, "Hand recognition & gesture use camera and openvc," 2016, acessado em agosto. 2018. [Online]. Disponível: <http://www.jackyle.com/2016/12/hand-recognition-gesture-use-camera-and.html>
 - [24] Unity Technologies, "Unity®," 2018, acessado em junho. 2018. [Online]. Disponível: <https://unity3d.com>
 - [25] S. R. R. Sanches, C. Oliveira, A. C. Sementille, and V. Freire, "Challenging situations for background subtraction algorithms," Applied Intelligence, vol. 49, no. 5, pp. 1771–1784, May 2019.