Sistema de simulação e design para realidade aumentada baseada em vídeo-mapping

Lucas Meirelles do Prado

Resumo – A realidade aumentada possui grande potencial para se espalhar pelo mundo. Existem hoje diversos meios de misturar a realidade com o mundo virtual. Neste projeto, interessa a utilização do vídeo-mapping como um meio de aumentar a realidade. Esse tipo de tecnologia usa objetos reais como meio básico para ser aumentado, que será colorido e modificado com um projetor. Interessou-se ainda mais pelo projeto e pelo desenvolvimento dessas estruturas projetivas, sobre como pode-se criar, da maneira mais fácil e produtiva, um sistema de projeção que seja capaz de iluminar um objeto de determinado tamanho. Quais deverão ser as dimensões para isso? Estas perguntas motivam o desenvolvimento de um sistema de CAD específico para projeções. Esse software foi desenvolvido e testado nesta pesquisa. Apresentam-se aqui as principais descobertas e os algoritmos utilizados para a construção de tal equipamento.

Palavras-chave – Realidade aumentada, Simulação, CAD, Vídeo-mapping

1 Introdução

Sempre se sonha com a possibilidade de modificar a realidade. Utilizam-se livros para imaginar mundos novos, criam-se mundos em que a magia é abundante e nos quais coisas inimagináveis são possíveis com um estalar de dedos. Desenvolve-se tecnologia para fazer o que não se consegue, inspira-se nos pássaros para conseguir voar, sempre se teve como objetivo controlar cada vez mais a realidade, com a era da informação consegue-se criar mundos virtuais inteiros. Assim, o sonho de integrar cada vez mais tecnologia continua.

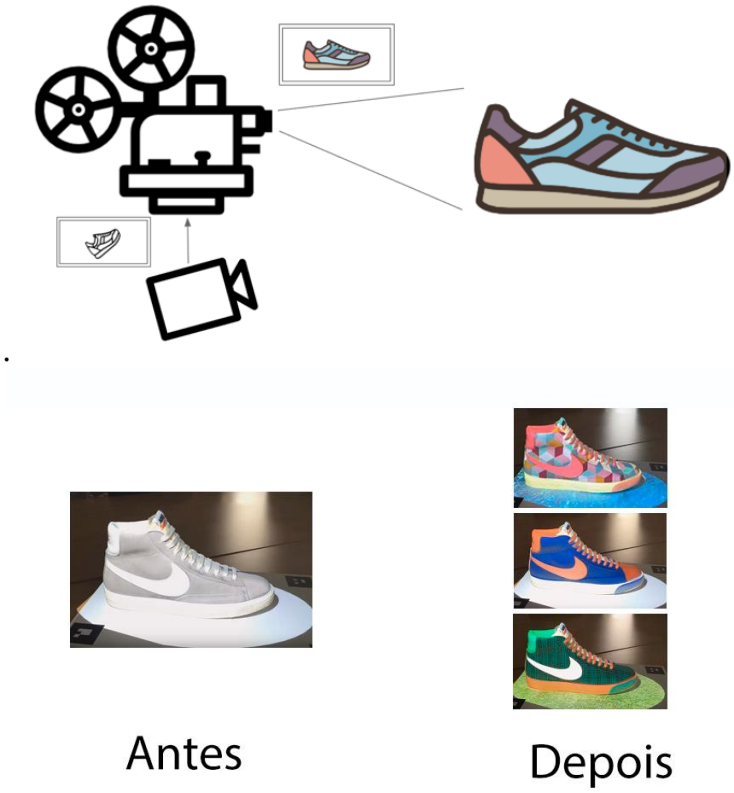
A realidade aumentada é uma dessas tentativas. Colocar uma camada virtual sobre o mundo real traz toda a capacidade da tecnologia para quando mais se precisa. Com ela, tenta-se melhorar ainda mais, aprender melhor, ter a informação quando realmente se precisa. As possibilidades são infinitas. Entretanto, a maioria das tecnologias de realidade aumentada ainda está longe desse ideal, como descrito pelo inventor da realidade aumentada, I. E. Sutherland (1965), nos anos 60, sobre o que seria sua tela perfeita. Ele imagina uma tela que fosse capaz de materializar qualquer objeto e em que as pessoas pudessem interagir diretamente com o ambiente.

Ainda se está longe da tela perfeita sonhada por Sutherland, mas existe uma alternativa ainda pouco explorada para fazer realidade aumentada, que é a utilização de projeções. Quando se pensa em realidade aumentada, sempre se imagina a utilização de óculos ou de algum implante que seja capaz de modificar os sentidos, quando na realidade, pode-se imaginar sistemas mais simples. Quando se fazem exposições de luz (vídeo-mapping) em grandes prédios, se está, de certa forma, modificando a realidade. Pode-se então pensar em fazer isto em menor escala e aproveitar os projetores como um meio de colocar uma imagem sobre objetos reais.

No futuro, certamente se conseguirá projetar sobre qualquer coisa ou transformar qualquer coisa em tela. No momento, se está ainda um pouco limitado. Interessou-se especificamente pela projeção em objetos pré-fabricados. O processo é simples e pode ser visto na Figura 1: esses objetos são escaneados; aponta-se um projetor na sua direção; a imagem escaneada é modificada e reprojetada no objeto; finalmente tem-se um objeto aumentado. Ou seja, tem-se um tênis branco que é transformado a partir da projeção em um tênis colorido.

Versão inicial submetida em 05 de Mar. de 2019. Versão final aceita em 17 de Dez. de 2019. Publicado em 18 de Dez. de 2019. Digital Object Identifier 10.11606/issn.2526-8260.mecatrone.2019.155396

Figura 1 - Projeção em um objeto para transformá-lo.



Com essa nova forma de fazer a realidade aumentada, chegam vários desafios: como planejar uma estrutura que garanta que um objeto será iluminado em um ângulo correto; como maximizar a superfície iluminada do projeto. Durante a execução desse projeto, desenvolveu-se um sistema de CAD que permite a construção de estruturas de projeção para a realidade aumentada.

2 Estrutura global e objetivos

Neste sistema de simulação e de design, tem-se objetivos simples: definir um projetor direcionado para um objeto e tentar encontrar a melhor disposição possível para esse conjunto. Para fazê-lo, é preciso colocar um projetor no ambiente, posicioná-lo e ser capaz de testar a qualidade da projeção nesse objeto.

Além disso, existe um elemento ainda não mencionado: a maioria dos sistemas de projeção em lugares fechados precisa ter o mínimo de espaço possível; isto é feito através de um espelho. Espelhos têm também de ser modelados no software para que se esteja em condições ideais.

Portanto, as funcionalidades principais são:

- Ser capaz de criar um projetor realista no sistema de CAD;
- Ser capaz de modificar o projetor com um espelho plano de qualquer formato e direção;
- Ser capaz de verificar a qualidade da imagem e da projeção.

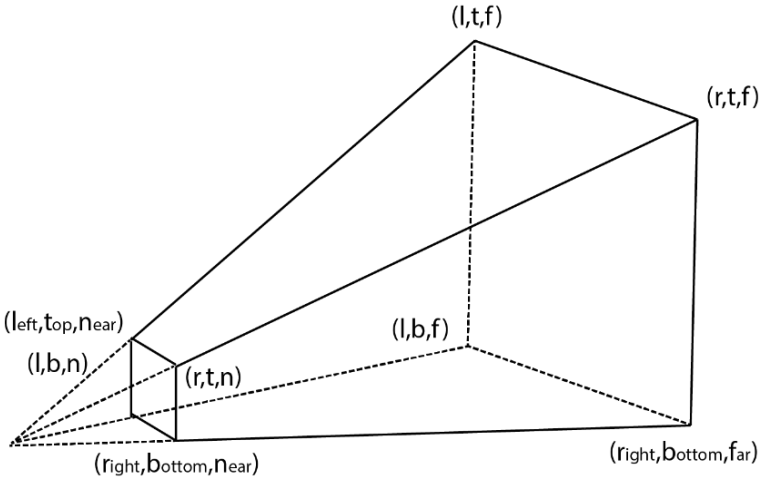
A integração desse novo sistema será feita diretamente com um software comercial, o *Solidworks*. Para isto, cria-se um módulo que se acopla ao *Solidworks*, adicionando a ele funcionalidades novas, baseando-se na sua API.

3 O projetor como um sólido

É possível encontrar uma modelagem simples para o projetor, que nada mais é do que um ampliador de imagens planas. Uma imagem pequena é sujeita a uma fonte luminosa e essa imagem vai crescendo até atingir a superfície. É fácil perceber que se está efetivamente criando um tronco de pirâmide feito de luz. O nome desse cone de projeção é chamado, na literatura, de *Viewing frustum*. Modeliza-se o

projetor como um sólido, com dois planos paralelos (*near*, perto da fonte luminosa e *far*, um plano distante, no qual a imagem é projetada) e quatro planos formando os lados da pirâmide: *left, top, right, bottom* , como pode ser visto na Figura 2.

Figura 2 - *Viewing Frustum.*



O processo para encontrar esse sólido é chamado de calibração. Esse processo é capaz de extrair as características fundamentais de um projetor real, tanto as intrínsecas quanto as extrínsecas (a forma do sólido e também a sua posição no espaço, respectivamente). A calibração é um algoritmo especializado que, com a ajuda de uma câmera e de um objeto físico, consegue determinar essas informações do projetor. Esse processo de calibração dá como resultado duas matrizes: projeção e *ModelView*. A primeira representa as características físicas do projetor e a segunda o posicionamento no mundo. Essas matrizes servem como base para os algoritmos aqui apresentados.

Utilizou-se um software de calibração baseado na calibração de TSAI (1986 e 1987) e também nos trabalhos de Audet e Okutomi (2009), que foi gentilmente fornecido pela empresa de vídeo-mapping *Smartpixels.* Partiu-se diretamente dos resultados obtidos por um software de calibração.

3.1 Construindo o *frustum*

A primeira etapa do projeto é construir um projetor virtual, a partir dos resultados da calibração. Para tal, utilizaram-se os conceitos apresentados acima: partindo de um projetor, que nada mais é do que um cone de projeção posicionado corretamente, utilizam-se os softwares de calibração, que fornecem a matriz de projeção já construída, que deverá ser revertida para permitir que as coordenadas dos pontos que formam o sólido sejam encontradas.

Essa inversão é feita desconstruindo a matriz de projeção em dois formatos equivalentes, a fim de voltar aos pontos principais. Para simplificar, na expressão (1), tem-se o sistema linear necessário para encontrar esses pontos, representado em função das características intrínsecas.

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ -\frac{2c_x}{res_x} & \frac{2c_x}{res_x} - 2 & 0 & 0 \\ 0 & 0 & \frac{2c_y}{res_y} - 2 & -\frac{2c_y}{res_y} \end{bmatrix} \begin{bmatrix} r \\ l \\ t \\ b \end{bmatrix} = \begin{bmatrix} \frac{n.res_x}{a_x} \\ \frac{n\,.res_y}{a_y} \\ 0 \\ 0 \end{bmatrix} \quad (1)$$

Sendo c_x e c_y as coordenadas dos pontos centrais da imagem; res_x e res_y as resoluções do projetor em pixels nas direções x e y da imagem, respectivamente; a_x e a_y os fatores de escala que

convertem o tamanho dos pixels em metros; e n a posição do plano *near*, arbitrária. Todas são características intrínsecas do projetor.

Resolvendo o sistema linear (1), tem-se a posição das oito coordenadas que definem os dois retângulos principais e, portanto, o *frustum*. Resta a questão de como construí-lo. É preciso posicionar todos esses 8 pontos e construir o sólido a partir deles. A questão não é trivial, uma vez que se tem de conectar os pontos na ordem certa.

O algoritmo desenvolvido permite a criação de sólidos projetados a partir de pontos de qualquer figura geométrica convexa. Nele, parte-se dos pontos principais, em que cada ponto é a combinação de todas as coordenadas:

$$(x,y,z) \rightarrow (r,t,n), (l,t,n), (l,b,n), (r,b,n), (r,t,f), (l,t,f), (l,b,f), (r,b,f).$$

A partir desses pontos, extraiu-se o envelope côncavo do *far* e do *near* que, por definição de um conjunto de pontos convexo, será único. Pega-se, em primeiro lugar, um ponto arbitrário no near, que possui dois vizinhos. Definindo (a,b) como uma das arestas escolhidas e (a', b') como os seus equivalentes no plano *far* (i.e r,t,n é equivalente de r,t,f), basta construir o segmento de plano que conecta (a → a' → b' → b), nessa ordem. Repetindo esse procedimento para cada uma das arestas formadas, tem-se um sólido formado para qualquer forma geométrica plana, desde que não haja cruzamentos entre as linhas. O algoritmo foi resumido em Algoritmo 3.1

Algoritmo 3.1: *Criação de um frustum*

Entrada: Coordenadas r,t,b,l
Saída: Frustum construído

1) Construir os pontos (r,t,n), (l,t,n), (l,b,n), (r,b,n), (r,t,f), (l,t,f), (l,b,f), (r,b,f);
2) Construir as superfícies near e far usando o envelope côncavo dos pontos;
3) Partir de um ponto arbitrário do near (a):
   3.1 Pegar um dos dois pontos conectados a ele (b);
   3.2 Construir a superfície (a → a' → b' → b);
   3.3 Ir para o ponto (b) e repetir o procedimento até que o ponto (a) seja selecionado novamente.

Essa generalização será muito importante no caso do espelho, pois permite que qualquer formato de *frustum* seja criado.

Na implementação desse algoritmo, utilizou-se um conceito chamado no *Solidworks* de *MacroFeature* (DASSAULT SYSTÈMES, 2018), que permite a criação de funcionalidades próprias por um usuário que conheça a sua API. A grande vantagem de usar esse tipo de estrutura é poder reconstruir o sólido a cada mudança de posição. Isto dá a possiblidade de ser reativo e mudar a cada movimento feito dentro do software. Mais uma vez, utilizou-se isso principalmente na presença do espelho, que age como um modificador do *frustum* original e faz com que se desvie e aumente de tamanho.

3.2 Modificando o frustum

Algumas funcionalidades interessantes foram também implementadas para ajudar o projetista a testar novas configurações. É o caso do deslocamento de lente (*lens shift*). Projetores mais caros e profissionais provavelmente possuem essa característica. Pode-se, a qualquer momento, aplicar um *lens shift*. No projetor real, isso é feito deslocando a lente e, portanto, alterando as características intrínsecas. No caso dessa pesquisa, pode-se usar uma alavanca virtual. Isso desloca a posição dos pontos de base do projetor e, afinal, a estrutura do projetor.

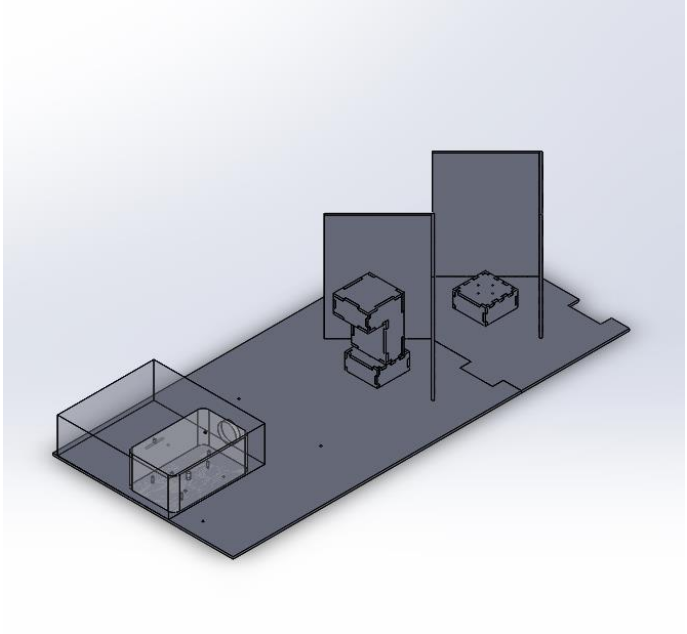
Uma outra possibilidade, apesar de não ter sido implementada, é simular também o zoom por um procedimento parecido. Ao modificar a posição dos pontos, modificam-se efetivamente as características intrínsecas do sistema.

3.3 Posicionando o *frustum* em relação ao projetor

A posição do *frustum* no espaço é um desafio muito maior do que sua criação, isto porque as características extrínsecas dependem fortemente do equipamento de medição. Ou seja, uma vez feita a calibração, as características intrínsecas ficarão constantes para qualquer posição, enquanto as características extrínsecas estarão acopladas à posição da câmera ou de algum objeto do ambiente. Isto é dado porque um projetor não consegue determinar sozinho a própria posição, ele precisa de uma câmera, que é capaz de encontrar a própria posição e, com alguns truques, encontrar a posição do projetor em relação à câmera.

Para que se consiga posicionar o *frustum* de luz na posição correta, é preciso ter um projetor, uma câmera e um objeto conhecido. E todos têm que ficar totalmente estáticos uns em relação aos outros. Para isso, cria-se um equipamento específico de calibração, apelidado *rig* de calibração. Seu projeto pode ser visto na Figura 3: ele possui uma peça cortada a laser de base, com posições para posicionar tanto o projetor quanto o objeto de calibração. Outra peça extra foi criada para permitir projetores maiores ou focos mais distantes. Executa-se a calibração nesse ambiente controlado, que resultará em uma matriz de ModelView que, por sua vez, será importada no ambiente virtual equivalente, para o posicionamento do *frustum* em relação ao projetor. Será preciso fazer algumas transformações nessa matriz, para que se tenha que adequar o sistema de coordenadas com o sistema usado no *Solidworks*. Isso é feito com uma transformação de coordenadas, através da matriz que relaciona a posição do mundo (centrado no objeto conhecido de calibração) com a tríade do software.

Figura 3 - Projeto de *rig* de calibração criado para estimar a posição do *frustum*.



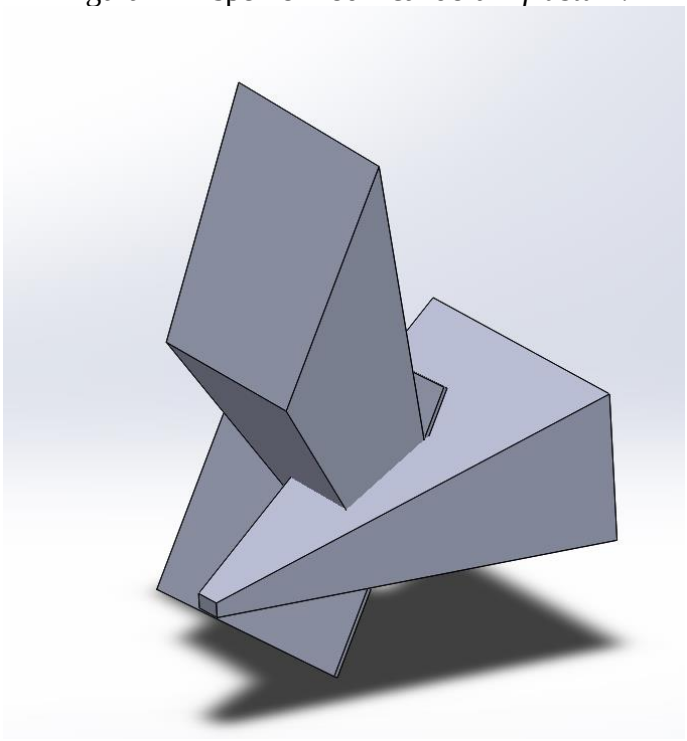
3.4 Espelhos

Espelhos são, efetivamente, objetos capazes de modificar o sólido de luz formado pelo projetor, mantendo iguais quase todas as suas características (o espelho pode ser visto como um projetor *virtual* que possui o mesmo *frustum* que o original, mudando sua forma e sua posição).

Para implementação desse projeto, os espelhos serão somente faces marcadas com a característica *espelho*. Toda a lógica de construção do sólido fica unificada na criação do *frustum*. Ao detectar sua presença, o próprio *frustum* criará um novo *braço*. O algoritmo de construção é relativamente simples. Precisa-se, entretanto, aplicar a lei da reflexão em três dimensões. Cada

extremidade do espelho é ligada ao centro óptico do projetor e refletido, seguindo a lei da reflexão. Para manter o plano de foco, estabelece-se que a distância projetada é igual à distância que o raio atingiria no plano *far*. O algoritmo funciona até mesmo para espelhos parciais, como pode ser visto na Figura 4.

Figura 4 - Espelho modificando um *frustum*.



3.5 Simulando um objeto

Com o que se tem em mãos, já se consegue ter uma ideia de como um projetor ilumina um objeto e qual o tamanho máximo desse objeto. Grande parte do trabalho manual de testes de um projetor vem de saber a concentração de luz no objeto e a presença possível de sombras. Para isso, foi desenvolvido um algoritmo que simula a projeção sobre um objeto. Isto é feito projetando raios do centro óptico por uma base quadriculada no plano *near* e calculando a intersecção desses vetores com o objeto. Na Figura 5, vê-se o resultado. Quanto mais densos os pontos azuis, mais iluminado está o objeto, também se vê claramente a zona de sombra criada. No caso especial do espelho, projetaram-se os pontos primeiro no espelho e então reprojetaram-se eles no objeto, da mesma forma.

Algoritmo 3.2: *Simulação por raios*

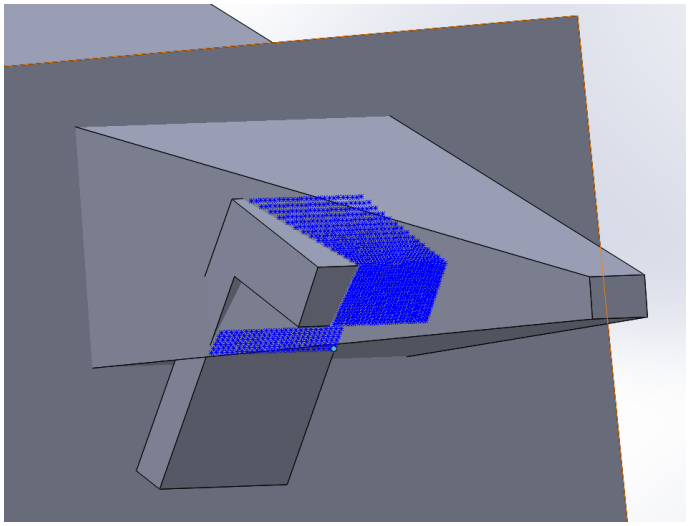
Entrada: Um *frustum* e um objeto a ser testado.
Saída: Iluminação planejada

1) Toma-se como base o segmento de plano *near*;
2) Nesse ponto, pegar duas arestas consecutivas $\vec{H}$ e $\vec{V}$;
3) Pegar um ponto qualquer seguindo a função (x e y frações)

$$P \rightarrow x\vec{H} + y\vec{V} \; \forall \; (x, y) \; decimal$$

4) Para cada ponto P, calcular o vetor $\overrightarrow{OP}$ e a intersecção entre o vetor $\overrightarrow{OP}$ e o objeto.

Figura 5 - Simulação por raios completada com sucesso.



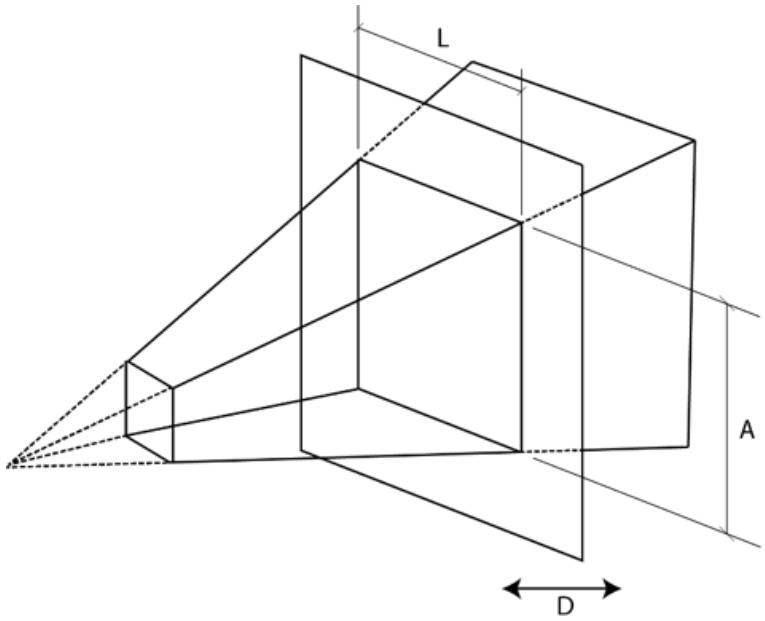
4 Validação do sistema de simulação

4.1 Método de experimento

Como todo equipamento de simulação, é importante medir o erro existente. Isso é feito comparando os resultados encontrados com outro método de medida. No caso dessa pesquisa, medir um cone de luz é extremamente complicado de fazer diretamente; criou-se um algoritmo alternativo de medida para comparar com a simulação. É importante dizer que a calibração é efetivamente um método de medir o *frustum*. No caso dessa pesquisa, cria-se efetivamente uma visualização para a calibração. Validar o sistema com esses resultados só indicaria se se criou corretamente o sólido, mas não o quanto ele está próximo da realidade.

O procedimento é o seguinte: corta-se o *frustum* com um plano perpendicular; a cada corte, medem-se a largura e a altura da imagem, como pode ser visto na Figura 6. Isso dirá o tamanho exato da imagem, que será comparado com o tamanho encontrado na simulação para uma mesma distância de projeção. Com isso, cria-se um novo método de calibração, que se baseia na medição física do cone de luz projetado. Esse procedimento foi automatizado (Anexo A) para obter medidas mais confiáveis.

Figura 6 - Procedimento de medição do *frustum*.



O aparato de medição utiliza um plano autoportante, que é usado para cortar a luz. Nesse plano, tem-se um quadriculado de tamanho conhecido. Ao longe, tem-se uma câmera calibrada (Logitech C525) apontada para o plano; utilizam-se técnicas de visão computacional para medir o tamanho da imagem. Além disso, foi desenvolvido um pequeno sensor, à base de um LIDAR comercial (GY-530 VL53L0X), para medição da distância entre o projetor e o plano.

O principal problema dessa abordagem vem do fato de que a imagem é nítida somente na distância determinada pelo foco. Por isso, limitou-se a janela de medida entre 600mm (foco feito) e 400mm. A partir daí, começa-se a ter uma incerteza muito grande na borda real do *frustum*.

Tiraram-se fotos a cerca de cada 10mm nesse intervalo. O algoritmo extraiu, então, as extremidades do retângulo e fez as medidas. No final, comparou-se a distância encontrada com o corte na mesma distância na simulação.

Para o espelho e para a simulação de raios, o procedimento é parecido, mas feito manualmente, mede-se o tamanho da imagem refletida e compara-se com o da simulada. Para a simulação de raios, mede-se o tamanho da sombra criada e compara-se com o da sombra simulada.

4.2 Resultados

Os resultados encontrados pelo sistema foram resumidos na Tabela 1.

Tabela 1 – Erro encontrado para diferentes distâncias

Distância (mm)	*Erro Altura*	*Erro Largura*
422,6	-4,3%	-3,5%
442,3	-3,7%	-3,2%
449,8	-3,4%	-2,1%
456,8	-1,1%	-1,4%
469,1	-2,9%	-2,0%
476,00	-2,8%	-1,9%
484,4	-2,6%	-1,5%
494,3	-2,3%	-1,4%
501,9	-2,2%	-1,3%
512,8	-1,7%	-1,1%
521,6	-1,5%	-1,2%
532,9	-1,3%	-1,4%
541,8	-0,7%	-1,4%
552,8	-1,2%	-1,8%
553,5	-0,9%	0,2%
568,3	-0,9%	-0,6%
575,1	-0,5%	-0,3%
585,2	-0,6%	-0,4%
594,2	-0,2%	-0,1%
618	0,1%	-0,2%
632,5	0,3%	0,7%

Próximo da calibração, tem-se um erro de cerca de 0,5%. Isso equivale a uma precisão de 0,69mm para esse plano. Como esperado, o erro cresce conforme se sai da zona de foco e as medidas são mais imprecisas. Apesar disso, ainda se está em um valor aceitável. O erro médio é de cerca de 1,6% para a altura e de 1,2% para a largura. Ignorando-se os pontos mais distantes do foco, tem-se uma média de 1,3% para a altura e de 1,0% para a largura. Não é possível saber, com esse experimento, qual é a origem desse erro. Ele pode estar incluído no sistema de medição de distâncias, pode estar na calibração ou também na precisão da transformação para o modelo da calibração, ou mesmo em possíveis distorções causadas pela lente. Não é o objetivo desse projeto detectar de onde vem o erro; deixa-se isso para uma etapa de refinamento do software, mas o importante é saber o

quanto um projetista pode esperar de erro ao usar o sistema.

Para o espelho, tem-se um erro um pouco maior, de 1,49% na altura e de 6% na largura no primeiro experimento e de 3,15% na altura e de 5,4% na largura. O software, porém, espera um espelho especial de reflexão direta, que não foi possível reproduzir. Nesse caso, o erro pode ser sistemático, da ordem de 5mm (o que causa um erro relativo maior para distâncias menores, como a altura da projeção).

A distorção pode ter sido causada pela refração no vidro ou outros fenômenos parecidos.

Para a simulação por raios, tem-se um erro relativo de 0,96% ou 0,3mm, valores considerados satisfatórios.

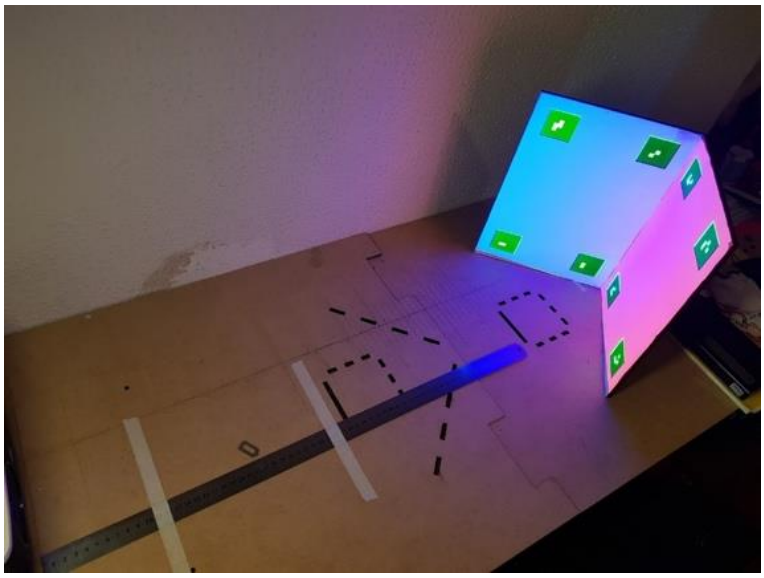
5 Conclusões

Os resultados encontrados indicam o bom funcionamento do software desenvolvido, assim como os algoritmos e os procedimentos criados para o auxílio ao desenvolvimento de equipamentos em realidade aumentada. Na Figura 7, tem-se um exemplo de realidade aumentada em uma das incontáveis calibrações feitas durante o projeto.

Muitas funcionalidades ainda podem ser implementadas e melhoradas para serem a caixa de ferramentas absoluta no desenvolvimento de sistemas projetivos. Considerou-se o resultado satisfatório. Infelizmente, depende-se de efetuar a calibração para conseguir determinar as características do projetor. Seria muito interessante se se conseguisse ter esses dados ainda antes de ter o projetor em mãos. Além disso, o processo de calibração é relativamente laborioso e necessita de um equipamento feito sob medida para tamanhos de projetores diferentes. Algumas dessas limitações poderão ser tratadas no futuro para obter algo ainda mais eficiente.

Apesar disso, acredita-se no potencial da realidade aumentada na vida das pessoas e na possibilidade de utilizar esse projeto para visualizar e para testar sistemas projetivos, transformando o processo de design em algo repetível e confiável, aspectos essenciais para o trabalho de engenharia.

Figura 7 - Sistema de calibração aumentado para verificação da qualidade da calibração.



AGRADECIMENTOS

Agradeço a todos que me ajudaram e que me forçaram a dar mais um passo neste projeto. Principalmente ao Prof. Dr. Celso M. Furukawa, que me guiou nessa jornada.

Agradeço também à empresa SmartPixels, principalmente a Oksana Tovstolytkina, que me deu a possibilidade de conhecer a realidade aumentada e me deu ajuda e motivação no caminho.

REFERÊNCIAS

AUDET, Samuel; OKUTOMI, Masatoshi. A user-friendly method to geometrically calibrate projector-camera systems. 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. Miami, FL, USA, 2009.

DASSAULT SYSTÈMES. Solidworks: 2017 Solidworks API Help. Disponível em: <http://help.solidworks.com/2017/english/api/sldworksapiprogguide/.> Acesso em: 24 de julho de 2018.

SMARTPIXELS. Smartpixels: The reality is our canvas. Disponível em: <https://www.smartpixels.fr>. Acesso em: 24 de julho de 2018.

SUTHERLAND, Ivan E. The Ultimate Display. Proceedings of the IFIP Congress 65, 1965, pp. 506– 508.

TSAI, Roger Y. An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, FL, 1986, pp. 364-374.

TSAI, Roger Y. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. IEEE Journal on Robotics and Automation, volume 3, issue 4, 1987, pp. 323-344.

Title – Development of simulation software for projection-based augmented reality

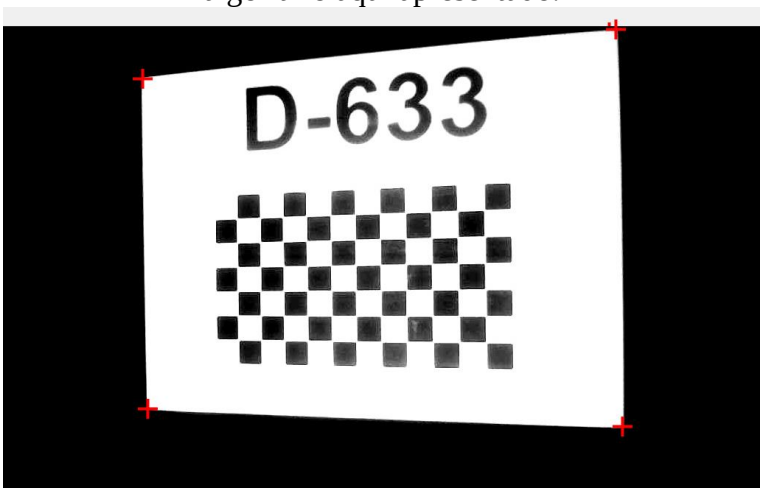
Abstract – Augmented reality is a buzzword. It has an incredible potential to be used every day and everywhere and to bring down all barriers between realities. There are numerous ways of mixing our reality with the virtual world of our computers. No one of them can yet do so in a perfect and reproductible way. There is interest in the use of video-mapping as a medium for augmenting reality. This type of technology uses real objects to be augmented, it doesn't need direct interaction with an user. There is interest in the development of video-mapping structures for augmented reality. How can be created, in the best way possible, a projection system capable of illuminating an object that has a given size? Which projector to choose and what will be the dimensions of this projective system? These questions motivated the research to build a CAD system made for projection. This software was developed and tested in this project. It is presented here some of the discoveries, methods and algorithms used in the development.

Keywords – Augmented reality, Simulation, CAD, *Solidworks*, Video-mapping

Lucas Meirelles do Prado, aluno de Graduação em Engenharia pela Escola Politécnica da Universidade de São Paulo. Artigo elaborado sob orientação do Prof. Dr. Celso M. Furukawa, simplificado e adaptado para o formato da revista Mecatrone.

ANEXO A – ALGORITMO DE EXTRAÇÃO E MEDIÇÃO DE UM FRUSTUM COM UM PLANO

Figura 8 - Exemplo de imagem usada para medição, com os pontos extremos detectados usando o algoritmo aqui apresentado.



Aqui descreve-se o algoritmo utilizado para medir as imagens e para detectar automaticamente os pontos extremos. No exemplo, a imagem foi tratada para eliminar sombras potenciais que degradavam o resultado. Neste algoritmo, são feitas várias transformações para separar qual parte é o retângulo iluminado e qual parte está impressa. O algoritmo completo pode ser visto a seguir.

Algoritmo A.1: *Detecção dos pontos extremos da projeção*

Entrada: Foto da medição de um *frustum.*
Saída: Posição em pixels dos quatro cantos da parte iluminada.

1) Melhora-se a imagem:
   a) Processo de binarização – filtra-se tudo que não está iluminado pelo projetor na foto;
   b) Correção de falhas – preenchimento de buracos.
2) Extração dos cantos:
   a) Detectam-se todos os cantos da imagem (sairá cheio de falsos cantos), mas contêm sempre os cantos da imagem;
   b) Detecta-se o contorno convexo da imagem e excluem-se os pontos internos;
   c) Encontram-se os pontos mais distantes da imagem. Esses são provavelmente ou muito próximos de dois cantos. Assume-se uma diagonal;
   d) Excluem-se todos os pontos a uma distância menor do que a altura e a largura da imagem, utilizando seu *aspect ratio* como base;
   e) Calculam-se dois a dois a distância entre os pontos e os pontos já definidos, até encontrar os pontos mais longe;
   f) Extrai-se o contorno convexo mais uma vez para ordenar os pontos.
3) **Saída**: temos os pontos.