

Desenvolvimento dos módulos de edição e de controle para sistema de treinamento interativo para profissionais de segurança

Antonio Valerio Netto¹

Resumo

O objetivo do sistema foi criar um ambiente de treinamento para seguranças privados, guardas municipais e policiais utilizando simulação, na qual os mesmos poderão interagir com os recursos visuais e sonoros criando uma sensação de presença física no ambiente (cenário de confronto). Para a criação dessa simulação, o sistema exibe em um anteparo, um filme 2D. A projeção desse filme no anteparo de metal é na proporção 1:1, exibindo os personagens em tamanho real. O sistema é responsável por alterar a sequência do vídeo da ocorrência de acordo com o momento e a coordenada de um disparo com arma de fogo real com projétil de silicone efetuado pelo usuário que está sendo treinado. Neste artigo é descrita o desenvolvimento da parte do sistema que é responsável por alterar a sequência de filme 2D de acordo com o momento e a coordenada de um disparo efetuado pelo usuário. Essa parte do sistema é constituído por três módulos: "Aquisição", "Edição" e "Controle". Neste artigo é destacado o desenvolvimento de dois desses módulos. O de "Edição" e o de "Controle".

Palavras-chave: sistema de treinamento, segurança privada, simulador, rastreamento de objetos em tempo real.

Abstract

The objective of the system was to create a training environment for private security guards, municipal and police guards using simulation, in which they can interact with the visual and sound resources creating a sense of physical presence in the environment (confrontation scenario). To create this simulation, the system displays a 2D movie on a screen. The projection of this film on the metal screen is in the ratio 1: 1, displaying the characters in real size. The system is responsible for changing the video sequence of the occurrence according to the timing and coordinate of a real firearm shot with silicon projectile performed by the user being trained. In this article we describe the development of the part of the system that is responsible for changing the sequence of 2D film according to the moment and the coordinate of a shot made by the user. This part of the system consists of three modules: "Acquisition", "Editing" and "Control". This article highlights the development of two such modules. The "Edit" and the "Control".

Keywords: Training system, private security, Simulator, object tracking.

¹ DT CNPq, SHIS QI 1 Conjunto B - Bloco A, Brasília - DF, antonio.valerio@pq.cnpq.com

1 Introdução

Os profissionais da área de segurança pública e privada necessitam realizar seus treinamentos de forma continuada e com um custo acessível, além de ter acesso a uma dinâmica mais próxima da realidade do cotidiano de trabalho de campo. O objetivo é que tenham uma formação mais adequada, aprimorando o desempenho de suas funções de forma segura e eficaz. Por estas razões citadas anteriormente são justificadas a criação de sistemas de treinamento que forneçam uma metodologia com um grau maior de assimilação do conhecimento transmitido, permitindo estímulos visuais, sonoros e interativos, gerando um ganho de qualidade no aprendizado do profissional (Valerio Netto, 2006). A utilização de simuladores tem o aspecto, inclusive de envolver pontos de interação, envolvimento e imersão. Não somente para simular o manuseio correto da arma de fogo, como é o caso dos simuladores de tiro (primeira geração), mas também, para treinar a verbalização, postura corporal, técnica de OMD (Observação, Memorizar e Descrever) como é o caso do emprego dos simuladores de abordagem (segunda geração) (Valerio Netto, 2015).

Os simuladores podem suprir a necessidade de treinamento da polícia e da segurança privada no que condiz com a utilização de arma de fogo como última opção diante de uma situação de conflito. Antes de atirar, existe todo um processo que vai desde a aplicação da técnica de observação, passa pela verbalização, postura corporal, e posteriormente, como última opção, o disparo de uma arma de fogo. Este tipo de simulador está alinhado com a aplicação da metodologia do uso progressivo ou seletivo da força (Sandes, 2007). Isto é, a ação deverá se dar de maneira compatível a gravidade da ameaça representada pela ação do infrator, sem se desviar do princípio da

legalidade que norteia o processo de uma intervenção (Xavier, 2009).

Ao longo dos anos foram realizados diversos estudos sobre o uso da tecnologia de simuladores virtuais para treinamento na área de segurança, além de testes de validação de campo que comprovaram que a tecnologia consegue atender as questões relacionadas ao treinamento continuado (Furuie, 2013) (Rosa & Pavanati, 2015) (Pinheiro *et al.*, 2016). Inclusive, Pinto (2009) relatou em sua pesquisa que os agentes de segurança que recorrem ao uso de força letal, na sua grande maioria, não têm a devida preparação para intervir em situações reais, tendo implicações no seu sucesso, que raramente excede 50% do total de casos. Segundo a revisão bibliográfica realizada pelo autor, o mesmo concluiu que as situações de treino de tiro real não são suficientemente representativas da realidade, porque acabam focando somente no uso da arma de fogo em condições estáticas, dando pouco ênfase à complexidade do conjunto de decisões que normalmente são solicitadas e uma ocorrência real.

Além de possibilitar a aplicação do método do uso progressivo ou seletivo da força e de abaixar os custos com treinamentos em stands de tiro reais, os simuladores virtuais de forma geral permitem aumentar a frequência dos treinamentos o que impacta diretamente na memória motora/muscular do profissional. Também possibilita realizar os treinamentos dentro das próprias bases operacionais o que evita deslocamento. Além disso, permite treinar equipes menores de profissionais em sessões de curta duração durante o expediente normal sem precisar realizar horas extras (Gomes, 2016) (Alves, 2014).

O projeto o qual o artigo está relacionado teve como meta desenvolver um protótipo de um sistema de multimídia interativo de treinamento na área de segurança pública e

privada. O propósito foi criar um sistema que pudesse “transportar” os profissionais da área de segurança para uma realidade mais próxima do cotidiano de trabalho, permitindo que os mesmos tenham um treinamento continuado para aprimorar suas memórias motoras/musculares. Muitas vezes, esses profissionais são colocados em situações de treinamento de tiro onde os alvos a serem alvejados são fixos ou imóveis, não transmitindo a sensação de realidade com a qual esse profissional irá realmente se defrontar no momento em que estiver prestando seus serviços no dia-a-dia. O projeto buscou gerar uma evolução nos equipamentos de treinamento levando maior interatividade e aumentando o sentimento tanto de imersão quanto de envolvimento.

Em linhas gerais, o funcionamento desse simulador se baseia na exibição, em um anteparo de metal, de um filme 2D com situações reais do cotidiano do agente de segurança, tais como assaltos, invasões entre outros, utilizando personagens em tamanho real (1:1). Isto produz por meio das imagens e dos sons emitidos, uma sensação de presença no momento em que a cena está ocorrendo. Para realizar a interação com a cena é utilizada uma arma real (revolver 38) carregada com projéteis de silicone ou óleo de mamona. O simulador possui um sistema de captura baseado em uma malha laser que identifica a posição que este disparo do tiro atingiu a imagem projetada por um projetor de 3500 lux em um anteparo de metal (Valerio Netto, 2017b). O profissional de segurança terá que tomar decisões, tais como o momento exato de sacar a arma e atirar; e posteriormente, a esta ação, ocorrerá uma reação do filme 2D de acordo com o resultado da posição (x, y) do disparo. Isto é, após efetuar o disparo, o sistema será responsável por definir qual a nova sequência do filme 2D será exibida na tela.

No artigo é descrito o desenvolvimento de parte do sistema computacional que é

responsável por alterar a sequência do filme 2D de acordo com o momento e a coordenada de um disparo efetuado pelo usuário. Essa parte do sistema é constituído por três módulos: “Aquisição”, “Edição” e “Controle”. Neste artigo é destacado o desenvolvimento de dois desses módulos. O de “Edição” e o de “Controle”.

Em linhas gerais, o módulo de “Aquisição” basicamente é responsável por identificar as coordenadas (x, y) onde o tiro simulado atingiu um anteparo de metal, e enviar estas informações para o módulo de “Controle” por uma porta serial RS 232C. É neste anteparo de metal que é exibido o filme 2D por meio de um projetor de 3500 lux conectado a um computador onde está sendo executado o *software* do simulador.

O módulo de “Edição” é responsável por automatizar a identificação de quais regiões (máscaras) no filme 2D, serão consideradas fatais ou não. Fatais, por exemplo, são cabeça e peito dos personagens que aparecem na cena. Esta atividade é realizada antes do treinamento operacional com os profissionais de segurança. Isto é, esse módulo é disponibilizado apenas à pessoa autorizada por manipular as imagens, para que a mesma possa construir as máscaras que identificam as regiões chaves nas cenas de cada filme 2D. Essas máscaras criadas e suas informações são armazenadas em um arquivo para serem usadas pelo módulo de “Controle”. Em linhas gerais o módulo de “Edição” é utilizado para configurar as máscaras no filme 2D que identificam ao longo de todo o filme 2D, as regiões fatais e não fatais.

O módulo de “Controle” é responsável por receber as coordenadas (x,y) do módulo de “Aquisição” e analisar essas coordenadas no *frame* do momento que o disparo do tiro de silicone atingiu o anteparo para determinar qual a sequência da ramificação do filme 2D deverá ser exibida. Os personagens desse

filme 2D já deverão possuir as máscaras que identificam as regiões fatais e não fatais que serão analisadas para determinar qual sequência de vídeo será exibida. Lembrando que essas máscaras são criadas no módulo de “Edição” e armazenadas em um arquivo que será lido pelo módulo de “Controle”.

A seção 2 descreve a lógica de funcionamento do filme 2D principal e suas ramificações. As seções 3 e 4 apresentam o desenvolvimento dos *softwares* dos módulos de “Edição” e de “Controle”, respectivamente. Por fim, a seção 5 descreve as considerações finais relacionadas à experiência do desenvolvimento do protótipo do sistema em questão.

2 Noções sobre o filme 2D principal e suas ramificações

Com relação à geração do filme 2D, o objetivo foi à confecção (roteiro e gravação) de um “banco de filmes”, isto é, o filme principal (*mastershot*) juntamente com os seus filmes secundários (ramificações) que são utilizados para permitir a continuação no vídeo dependendo de onde o disparo do projétil atingiu a cena. Para isto foi criado um roteiro principal com três continuações para efeito de testes do protótipo. O roteiro principal retrata um assalto a um posto bancário sendo realizado por um casal (situação fictícia com o objetivo de testar os módulos do sistema). Foram realizadas e editadas seis filmagens, sendo que a *mastershot* teve a duração em torno de dois minutos cada e as ramificações, em torno de 10 segundos. Essas filmagens tiveram o seu *viewpoint* (posição da câmera) colocado no “olhos” do único segurança armado do posto bancário. Esse posto bancário fictício foi montada de forma simplificada, isto é, apenas um caixa e sem porta giratória, em uma alocação externa com a participação de cinco atores (dois assaltantes, duas pessoas na fila e um funcionário no caixa). Esses vídeos foram gravados em formato MPEG

1, uma vez que esse formato apresenta a maior taxa de compressão com a menor perda de qualidade da imagem.

A seguir é descrito o roteiro (*Mastershot*). A cena se inicia com a visão panorâmica da rotina diária de um posto bancário sendo observado pelos “olhos” de um vigilante. Aos seus “olhos”, a cena começa com um homem adentrando em uma agência bancária e se posicionando no final de uma fila de espera do caixa. Passados alguns segundos, uma mulher entra na agência e vai em direção ao mesmo caixa, passando direto pela fila e ao chegar ao caixa, ela saca uma arma e grita: “É um assalto! Todos para o chão”. As pessoas que estão na fila se assustam e começam a se deitar no chão. A mulher assaltante então pula por cima do balcão, jogando o funcionário do banco que estava no caixa para o chão. Em uma bolsa grande, ela começa a colocar o dinheiro que está no caixa.

Neste mesmo momento, o homem que estava no final da fila, também saca uma arma e pega a mulher que estava na frente dele na fila como refém. Esse assaltante encara o segurança do banco e aponta a arma para a refém e grita para o segurança largar a arma no chão senão vai atirar na refém.

Diante disso, o segurança precisa tomar uma decisão do que fazer. Isto é, ele tem uma situação crítica e tem poucos segundos para decidir como agir diante do fato. O sistema foi modelado para atender a seis possíveis ações do segurança. São elas:

- O segurança não reage ao assalto;
- O segurança reage e o seu disparo atinge o assaltante homem;
- O segurança reage e o seu disparo atinge o funcionário atrás do balcão;
- O segurança reage e o seu disparo atinge a assaltante mulher;

- O segurança reage e o seu disparo atinge a refém;
- O segurança reage e o seu disparo não atinge ninguém.

As possíveis interações que o segurança poderia ter e a reação causada em decorrências (desfecho) dessas interações são apresentadas nas Tabelas 1a e 1b.

Tabela 1a: Descrição da situação onde o segurança não reage. Esta situação é chamada de *Martershot*.

Ação	Reação/Desfecho passivo (<i>Martershot</i>)
O segurança não reage ao assalto	Nenhuma interação. Este é o roteiro do vídeo <i>Martershot</i> . 1.) A mulher assaltante se abaixa para pegar o dinheiro atrás do balcão. Neste mesmo momento a refém consegue se afastar um pouco do assaltante. Mas ele consegue segurar a refém e se posicionar atrás dela em relação a posição do segurança. A mulher assaltante pula novamente o balcão. Usando a refém como escudo, o casal escapa do posto bancário pela porta.

Tabela 1b: Descrição das cinco possíveis reações (desfechos) caso o segurança reaja e dispare um tiro. Essas possibilidades ainda se ramificam em mais situações caso o tiro atinja uma região fatal ou não.

Ações	Reações / desfechos (ramificações do filme 2D)
Segurança reage e o seu disparo atinge o assaltante homem	(tiro em região fatal) Assaltante homem é ferido gravemente e cai no chão e a mulher assaltante se rende ao ver a cena. Figura 1(a)
	(tiro em região não fatal) Assaltante homem leva um tiro no ombro ou perna e solta a sua arma que cai no chão, a refém acaba correndo para a saída e a mulher assaltante se rende. Figura 1(b).
Segurança reage e o seu disparo atinge o bancário atrás do balcão	(tiro em região fatal ou não) Bancário é atingido gravemente; assaltante homem revida, atira contra o segurança e foge, mulher assaltante se esconde atrás do balcão.
Segurança reage e o	(tiro em região fatal ou não)

Ações	Reações / desfechos (ramificações do filme 2D)
seu disparo atinge a assaltante mulher	Assaltante mulher é ferida enquanto está atrás do balcão e o assaltante homem dispara contra o segurança e joga a refém no chão.
Segurança reage e o seu disparo atinge a refém Figura 1(c)	(tiro em região fatal ou não) Refém é atingida na barriga e assaltante homem revida e atira contra o segurança e foge, mulher assaltante se esconde atrás do balcão.
Segurança reage e o seu disparo não atinge ninguém Figura 1(d)	Assaltante homem revida e atira contra o segurança.

Na Figura 1, são mostradas possíveis sequências tomadas para a exibição do filme 2D, caso seja efetuado um disparo. Se não houver disparo (segurança não reage), o filme principal (*martershot*) será exibido até o final. Do contrário, poderá tomar a sequência da Figura 1(a), quando o assaltante é atingido na região fatal, ou da Figura 1(b), onde o assaltante é atingido na perna. Ainda, na Figura 1(c), a refém é atingida. E por fim, na Figura 1(d), o disparo não atinge nenhuma das máscaras, e o assaltante reage atirando contra o segurança.

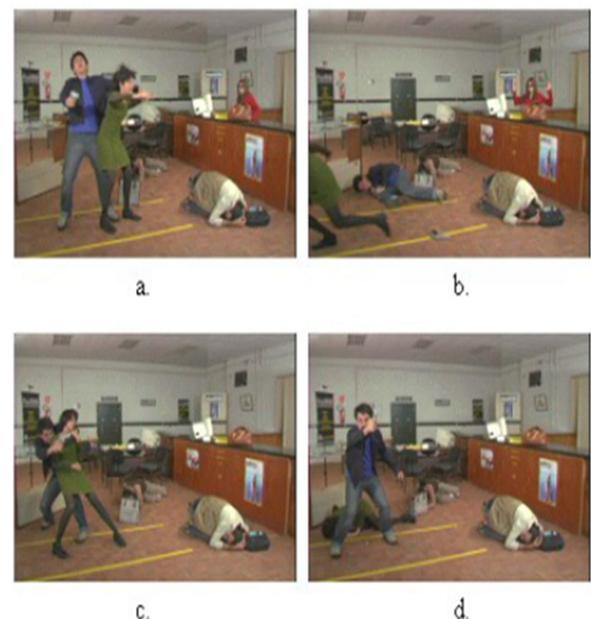


Figura 1: Possíveis sequências do filme 2D depois que o segurança dispara: (a) Assaltante é atingido fatalmente; (b) Assaltante é ferido na perna; (c)

Refém é atingida; (d) Ninguém é atingido e o assaltante reage atirando contra o segurança.

Os testes foram realizados utilizando um projetor de 3500 lux (permitir que as cores sobressaltassem mesmo em uma sala com maior iluminação, isto é, evitar cenas esbranquiçadas) que é acionado pelo módulo de “Controle” a uma distância de 3,4 m do anteparo de metal para manter um tamanho real (1:1) dos personagens e do cenário. O sistema de som conectado ao computador possuía surround *dolby* 5.1.

3 Módulo de “Edição”

Esse módulo tem como objetivo a montagem de uma cena que permita a interação do usuário com o filme 2D. Basicamente, a partir de um vídeo principal (*mastershot*), são inseridas máscaras para rastreamento dos objetos de interesse para que no momento da interação com estes objetos, a cena seja conduzida para outra situação (cena). Cada máscara é um contorno fechado que contém informação sobre qual vídeo secundário (ramificação) deverá ser exibido caso haja interação em qualquer ponto dentro dessa máscara. Com este intuito, foi criado um programa em linguagem C++ para a montagem das cenas.

O funcionamento básico deste *software* consiste em dado um filme *mastershot* em formato WMV (Windows Media Vídeo), o usuário deve inserir as máscaras sobre objetos de interesse (rosto, perna, braço, etc.) para que, no momento da interação com estes objetos, a cena ramifique-se para outra situação. As informações sobre essas máscaras e as ramificações correspondentes caso o tiro atinja as mesmas são armazenadas em um arquivo de extensão “.vmp”.

O conjunto dos vídeos (*mastershot* e os trechos gravados das continuações) e o arquivo contendo as informações sobre as máscaras e ramificações (relações entre

máscaras e vídeos) são chamados de “Cena”.

Uma primeira versão do *software* desenvolvido criava imagens para cada *frame* do vídeo principal. A inclusão de máscaras era realizada, portanto, em cada uma das imagens estáticas geradas. Isto é, as máscaras eram inseridas manualmente em cada *frame* (lembrando que são 30 *frames* por segundo), sendo criado um arquivo binário com as informações, que, em seguida, eram lidas pelo módulo de “Controle”. Esta atividade era dispendiosa e onerava o tempo de trabalho com os filmes 2D. Praticamente tornava inviável o projeto do simulador.

Foi fundamental a introdução da utilização de técnicas avançadas de processamento e análise de imagens e vídeo que viabilizou o processo automático de edição, tornando a solução mais rápida e dando características inovadoras ao sistema. Em Valerio Netto (2017) é relatado detalhadamente a justificativa do emprego de determinadas técnicas de processamento de imagens e os passos realizados para o desenvolvimento dos algoritmos utilizados no módulo de “Edição” do sistema para a criação e para a automatização das máscaras.

3.1 Software para gerenciamento de edição

O *software* para gerenciamento de edição foi desenvolvido em linguagem C++ e suas funcionalidades estão relacionadas à criação de uma nova “Cena”, abrir e editar as máscaras e suas ramificações. Na Figura 2 é apresentado um *print screen* da tela inicial do aplicativo. Esse sistema foi desenvolvido para permitir visualizar os vídeos pertencentes àquela cena (Figura 3a), utilizando componentes da biblioteca VideoLab (Mitov, 2016) para captura de vídeo. E, para cada vídeo, permite visualizar as máscaras e as informações sobre as

mesmas (Figura 3b). O aplicativo ainda permite realizar a inserção das máscaras utilizando os arquivos de vídeos diretamente, sem a necessidade de particioná-los. Essas informações sobre os vídeos e as máscaras correspondentes são armazenadas em um arquivo com extensão “.vmp”. Esse arquivo juntamente com os vídeos são transportados para o módulo de “Controle”, que é responsável por permitir a interação do usuário com a “Cena”.



Figura 2: Software do módulo de edição.

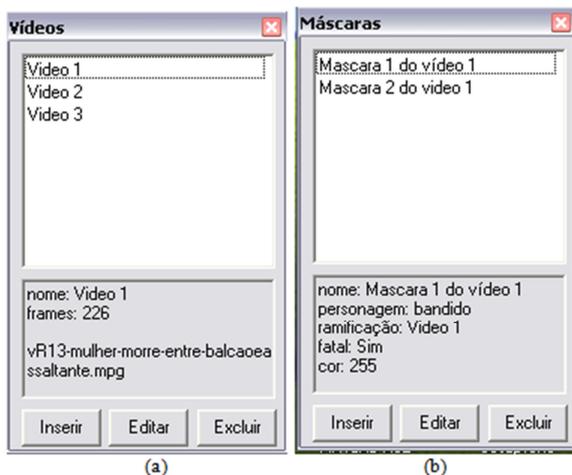


Figura 3: Janelas para controle de vídeos (a) e máscaras (b).

3.2 Construção e automatização das máscaras

As máscaras são criadas por meio da janela “Máscaras” do software de gerenciamento de edição. Elas armazenam um conjunto de

informações sobre uma determinada região do *frame* do *mastershot* e qual o vídeo (ramificação) deve ser exibido ao existir a interação do tiro disparado pelo usuário. Essas informações referente à região são: nome de identificação da máscara, personagem do vídeo onde é inserida a máscara, ramificação ou vídeo de continuação, se é região fatal ou não, cor do contorno da máscara e os pontos que formam o contorno dessa máscara.

Como citado anteriormente, em uma versão anterior do aplicativo a máscara era criada de forma manual, por meio da união de retas consecutivas, e por isto não eram definidas corretamente, armazenando pontos que não estavam dentro da região de interesse (Figura 4). Isso gerava problemas para o módulo de “Controle” durante a interação do usuário, pois pontos que deveriam estar fora da máscara eram atingidos e uma ramificação era executada não trazendo uma realidade adequada para o sistema.



Figura 4: Contorno criado manualmente pela união de retas consecutivas onde é possível verificar que pontos que não deveria estar dentro da máscaras acabavam sendo incluídos.

Para se obter um contorno mais preciso, várias técnicas de segmentação de imagens foram estudadas e testadas inicialmente em MatLab. Entre elas, o método de Contorno Ativo também conhecido como *snakes* (Kass, Witkin e Terzopoulos, 1988), usando Fluxo de Vetor Gradiente (GVF – *Gradient Flow Vector*) (Xu & Prince, 1998), foi o que

forneceu melhores resultados, como apresentado na Figura 5, após ter sido implementado em C++. O método do contorno ativo ou *snakes* utilizando GVF usa um campo de vetores atrativos para direcionar o contorno. Este método pode segmentar formas que apresentam problemas associados com inicialização e convergência pobre para concavidades de bordas. Portanto, foi considerado satisfatório para este trabalho. Bons resultados também foram obtidos na segmentação de outras partes do corpo das pessoas no vídeo. A partir do contorno obtido inicialmente para a máscara, o próximo passo é rastrear-lo durante a execução do filme (Valerio Netto, 2017).



Figura 5: Contorno obtido com Contorno Ativo usando GVF.

Posteriormente, com a melhora do contorno das máscaras, era necessário que as mesmas fossem replicadas em todos os *frames* do filme 2D, de forma automática sem a necessidade do usuário ir colocando *frame a frame*. Para isto, foram estudadas técnicas para rastreamento de objetos em vídeo. Geralmente os métodos encontrados tratavam de rastreamento de objetos específicos ou com apenas uma característica ou forma, como por exemplo: uma pessoa, uma arma, círculos, quadrados, uma cor específica, etc.

No caso dessa aplicação foi preciso uma solução mais sofisticada, pois o contorno das máscaras deveria acompanhar objetos de

diferentes formas e com intensidades diferentes, mantendo-se os pontos iniciais definidos pelo usuário. No *software* anterior foi desenvolvido uma técnica de rastreamento de objetos em vídeo para que o usuário pudesse inserir máscaras nos personagens da “Cena” no *frame* em que desejasse iniciar a interação e, por meio de um comando, as máscaras se propagassem pelo vídeo, acompanhando o objeto inicial. A técnica desenvolvida misturava segmentação, rastreamento de objetos e detecção de movimento para tratar deste problema. Porém, os resultados não foram satisfatórios e por isto novas técnicas de rastreamento foram estudadas e avaliadas para substituir a anterior (Valerio Netto, 2017). Foi trabalhada a abordagem utilizando Fluxo Óptico. Nessa abordagem foram aplicados os métodos de obtenção do fluxo óptico, por Lucas-Kanade (Lucas e Kanade, 1981) juntamente com a abordagem de *snakes* utilizada para a construção das máscaras. Dada uma máscara no *frame* referência, calcula-se o fluxo óptico em relação ao próximo *frame*.

3.3 Criação de classes para controle da “Cena”

A estrutura se baseia na programação C++ orientada a objeto (POO), utilizando-se de classes criadas especificamente para este projeto para o armazenamento de dados. São elas: “TVideo”, “TMascara” e “TImageContour”.

A classe “TVideo” é responsável por armazenar os atributos dos vídeos utilizados pelo usuário, tais como ‘nome’, ‘diretório’, ‘número de *frames*’ e um vetor para armazenar as máscaras que serão definidas na sequência. Os atributos são:

1. nomeVideo: Identifica o vídeo. O software bloqueia a existência de dois vídeos com o mesmo nome;
2. nomeArquivo: Identifica o nome do arquivo de vídeo;

3. `diretorio`: Armazena o diretório onde o vídeo está gravado. Ex: `c:\videos \video1.mpg`
4. `frames`: Número de frames (quadros) que o vídeo contém;
5. `vetMaskara`: Guarda as máscaras que são criadas.

A classe “`TMascara`” possui os atributos das máscaras definidos pelo usuário, tais como ‘nome’, ‘personagem’, ‘fatal’ ou ‘não-fatal’, ‘cor’, nome do vídeo de ramificação e um vetor para armazenar as coordenadas da máscara. Os atributos são:

1. `nome`: Identifica a máscara. O software não permite a criação de duas máscaras com o mesmo nome;
2. `personagem`: Para organização do usuário. Pode ser preenchido, por exemplo, com “`criminoso`”, “`vítima`” ou “`coadjuvante`”;
3. `fatal` ou `não-fatal`: Distingue se uma região do corpo do personagem é fatal (mata) ou não-fatal (fere);
4. `cor`: Para facilitar a visualização das máscaras;
5. `ramificação`: Guarda o nome do vídeo que sucede o vídeo principal;
6. `vetContourGroup`: Armazena um conjunto de pontos para cada frame do vídeo que contém a máscara.

A classe “`TImageContour`” é responsável pelo processamento da imagem e pelo cálculo dos pontos do contorno da máscara. Nela estão implementadas as funções do cálculo do contorno ativo com GVF. Existe apenas um atributo:

1. `matFx`, `matFy`: Matrizes utilizadas para guardar valores referentes as coordenadas `x` e `y` calculados para criação do contorno.

3.4 Criação de arquivo decorrente da edição do vídeo

Mediante a necessidade do usuário de salvar as evoluções durante a edição, foi criado um arquivo “.h” que possui basicamente as funções ‘`salvarArquivo`’ e ‘`lerArquivo`’. Essas funções são responsáveis por gravar em um arquivo de extensão “.vmp” toda a estrutura de classes utilizada pelo sistema e depois carregá-la. Para isso são utilizadas TAGs que distinguem os blocos de salvamento (vídeo, máscara e contorno). Abaixo segue uma lista do TAGs disponíveis:

```
>>VIDEO inicio de bloco de vídeo
<<VIDEO fim de bloco de vídeo

>>MASCARA inicio de bloco de máscara
<<MASCARA fim de bloco de máscara

>>CONTORNO inicio de bloco de pontos
<<CONTORNO fim de bloco de pontos

~~ inicio de bloco de dado
.. fim de bloco de dado

!! fim do arquivo
```

A seguir é apresentado um exemplo prático:

```
>> VIDEO
  ~~ 'nome do vídeo' ..
  ~~ 'nome do arquivo de vídeo' ..
  ~~ 'diretório' ..
  ~~ 'numero de frames' ..
>> MASCARA
  ~~ 'nome' ..
  ~~ 'personagem' ..
  ~~ 'fatal ou não-fatal' ..
  ~~ 'cor' ..
  ~~ 'nome do vídeo a ser chamado' ..
>> CONTORNO
  ~~ 'frame ao qual pertence' ..
  ~~ 'X1' ..      ~~ 'Y1' ..
  ~~ '...' ..    ~~ '...' ..
  ~~ 'Xn' ..     ~~ 'Yn' ..
  << CONTORNO..
  << MASCARA..
<< VIDEO..
!!
```

Este arquivo com extensão “.vmp” é utilizado como entrada para o módulo de

“Controle” e permite a exibição do filme 2D com suas ramificações.

4 Módulo de “Controle”

O módulo de “Controle” tem como objetivo exibir uma “Cena” e permitir treinar profissionais por meio de interações com esta “Cena”. Isto é possível utilizando filmes 2D que foram editados pelo módulo de “Edição”. Ao iniciar o treinamento, o usuário realiza disparos com projeteis de silicone no anteparo de metal que exibe o filme 2D. O disparo aciona o módulo de “Aquisição” que envia sinais por uma porta serial para o módulo de “Controle”, por meio de um protocolo de comunicação. Esses sinais correspondem à posição (x, y) do tiro disparado. Com esta posição, o módulo de “Controle” verifica naquele instante da “Cena” se este ponto sensibiliza algum ponto de interação (máscara). Caso seja positiva a sensibilização, o módulo interrompe o vídeo *mastershot* e aciona a ramificação para outro vídeo. A leitura da porta serial com a informação da posição do disparo é realizada constantemente após o usuário clicar no botão 'Iniciar Aquisição' na tela do aplicativo do módulo de “Controle”. Esse aplicativo do módulo ainda realiza duas atividades importantes para o funcionamento do simulador. Trata-se da calibração do referencial absoluto da coordenada (x,y) e o algoritmo de verificação da posição do disparo.

4.1 Calibração do referencial absoluto da coordenada (x, y)

Quando o usuário que está em treinamento efetua um disparo que atinge o anteparo de metal durante uma “Cena”, o módulo de “Aquisição” que possui uma malha laser deve gerar uma interrupção nas placas receptoras da parte de *hardware* do simulador. Depois, o *software* embarcado embutido nesse módulo de “Aquisição” deve calcular uma posição (x,y) e enviar

para o módulo de “Controle” que utilizará essa posição para verificar qual é o ponto de interação no *frame* do filme 2D que está sendo projetado no anteparo (Valerio Netto, 2017b).

Para isto, esse sistema de captura da posição necessita de um tipo de calibração que faça uma correspondência entre a posição física capturada pela malha laser e sua posição relativa dentro do sistema computacional do simulador. Isto se deve, pois a estrutura física dos lasers da malha, o anteparo de metal e o projetor de 3500 lux não possuem posições fixas. Os três elementos podem mudar de posição seja por um deslocamento intencional ou não. Diante disso, este ponto físico detectado pelo disparo pode sofrer com a mudança física de um desses três elementos. Por isto, faz-se necessário realizar uma calibração com o sistema para gerar uma correlação entre o ponto físico vindo do disparo do laser e o seu referencial digital que será utilizado pelos algoritmos do sistema.

Foi desenvolvido um método de calibração que se encontra dentro do módulo de “Controle” como uma de suas funcionalidades. Ela é invocada pelo botão 'Calibrar' presente na tela principal do sistema. Basicamente, é exibido no anteparo de metal uma tela com dois pontos numerados (Figura 6). O sistema pede ao usuário que utilize um objeto de espessura fina para interromper a malha laser exatamente nestes dois pontos. Tendo a correspondência entre os dois pontos físicos e os referentes ao vídeo, é possível calcular um fator para x e para y (Equação 1), de forma que a correspondência para qualquer ponto capturado pela malha laser seja realizada apenas multiplicando-se a posição recebida pelo fator. Esse fator é dado por:

$$Fator(i) = \frac{Pv_{i1} - Pv_{i2}}{Pm_{i1} - Pm_{i2}} \quad (\text{Equação 1})$$

onde $i = \{x, y\}$, Pv_{i1} e Pv_{i2} são os pontos exibidos no vídeo e Pm_{i1} e Pm_{i2} são os pontos correspondentes na malha laser.

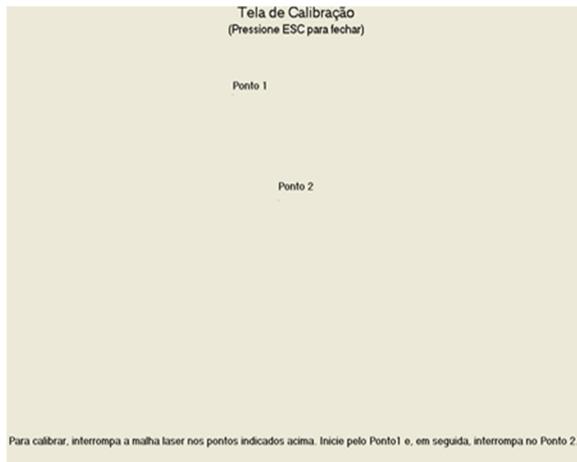


Figura 6: Tela para calibração responsável por fazer a correlação entre os pontos físicos gerados pela projeção no anteparo de metal e seus correspondentes digitais.

4.2 Algoritmo de verificação da posição do disparo e troca dos filmes 2D

Como comentado anteriormente, o módulo de “Controle” deve verificar, se dado um disparo com projétil de silicone no anteparo de metal que está sendo projetado o filme 2D, o ponto de interação atingiu algum objeto de interesse (cabeça, perna, braço da pessoa). Esta informação está em forma de pontos (x,y) que formam um polígono, que cobrem as regiões de interesse e estão definidos em um arquivo gerado pelo módulo de “Edição”, que foi descrito na seção anterior. Portanto, o problema se resume a verificar se um ponto está dentro de um polígono fechado, definido por um número arbitrário de pontos.

Foram realizados estudos em computação geométrica que permitiram conhecer o funcionamento de alguns algoritmos para prescrever se um determinado ponto obtido pertence a um polígono. Dentre os pesquisados, o mais interessante foi o chamado *Ray Crossings* (O'Rourke, 1998). A ideia fundamental do algoritmo é traçar, a partir do ponto obtido, uma semirreta em

uma direção qualquer. Quando esta semirreta intercepta o polígono um número ímpar de vezes, o ponto está dentro do polígono; caso seja um número par, o ponto está fora. Por conveniência, adota-se em geral uma semirreta paralela ao eixo das abscissas passando pelo ponto obtido e seguindo para a direita. A contagem do número de intersecções é realizada verificando quantos segmentos da borda do polígono têm um ponto extremo acima e outro abaixo da ordenada do ponto. Destes, é preciso verificar em quantos a intersecção com a semirreta ocorre em um ponto de abscissa maior que a abscissa do ponto. A maior dificuldade foi no tratamento de casos especiais, como:

- Semirreta passa por uma aresta do polígono (Figura 9a);
- Semirreta passa por um vértice do polígono (Figura 9b);
- Ponto está sobre a borda do polígono (Figura 9c);
- Ponto coincide com um vértice do polígono (Figura 9d).

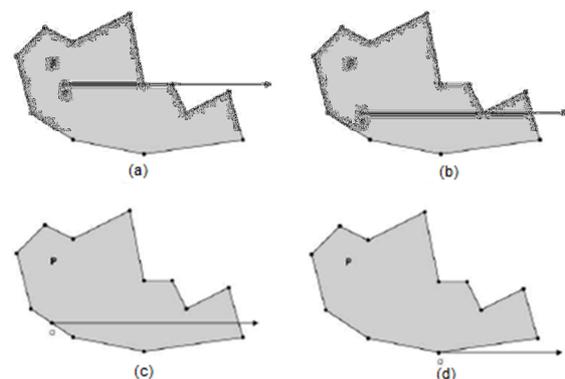


Figura 9: Tratamento de casos especiais.

Para esses casos, a solução adotada é a contagem de intersecções de modo que:

- Se a reta passa por um vértice, a intersecção deve ser considerada apenas se for o vértice com maior ordenada do segmento;

- Se a reta passa por um segmento do contorno do polígono, nenhuma intersecção deve ser considerada;
- Se o ponto pertence a um segmento do contorno considerar como intersecção.

A complexidade desse algoritmo é da ordem $O(n)$. Ele foi utilizado por adequar-se tanto para polígonos convexos como não convexos, pois, em certos quadros do filme 2D, houve a construção de polígonos não-convexos. O algoritmo foi adaptado à estrutura de dados do módulo de “Controle” e obteve bons resultados.

Quando o módulo de “Controle” recebe a coordenada do disparo (x,y) , o sistema analisa o *frame* correspondente ao disparo. Caso o tiro de silicone atinja uma máscara no filme 2D que está sendo exibido, é necessário realizar a troca do vídeo principal (*mastershot*) por uma das sequências (ramificações ou desfechos). Para que isto ocorra cada máscara possui um índice para uma lista de nomes de vídeos de ramificações na estrutura de dados do arquivo “.vmp”. Quando o sistema obtém esse índice, uma função retorna o nome dessa ramificação. Esse nome de arquivo de vídeo é passado como argumento para outra função do sistema que realiza a troca. Isto é, utiliza o nome obtido para executar o vídeo da ramificação que será exibido no lugar do *mastershot*.

5 Resultados

Foi realizada a integração entre os módulos (Figura 10). Em linhas gerais, o fluxo de funcionalidade inicia-se com o recebimento dos filmes 2D (*mastershot* e das ramificações) vindos da produtora de vídeo. Depois é empregado o módulo de “Edição”, com o qual é gerada a “Cena” e gravada na forma de um arquivo '.vmp'. Este arquivo juntamente com os filmes 2D são lidos pelo módulo de “Controle”. Que por sua vez é responsável por projetar o *mastershot* no

anteparo de metal, aguardando uma interação gerada por um disparo de tiro de silicone realizado pelo usuário. Por meio do módulo de “Aquisição” que possui uma malha laser (Valerio Netto, 2017b) é possível identificar as coordenadas (x, y) desse disparo. Por meio desta posição, o módulo de “Controle” calcula onde foi realizada a interação e projeta o vídeo com a ramificação correspondente, finalizando o treinamento simulado.

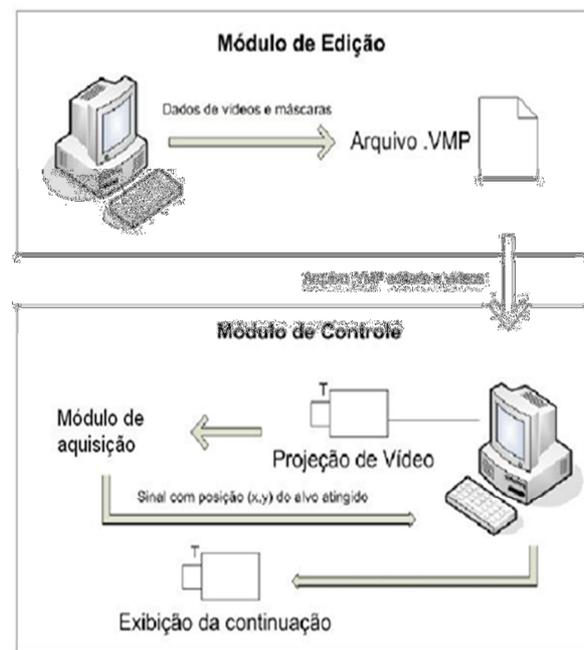


Figura 10: Integração dos módulos do sistema.

Os testes foram realizados em formato de baterias com 20 tiros cada e utilizando uma arma de fogo real (revolver 38) com os projeteis de silicone. No total foram realizados cinco baterias que totalizam 100 disparos e que permitiram testar todas as ramificações.

6 Considerações finais

Os módulos de “Edição” e de “Controle” apresentaram melhoras significativas em relação à versão anterior demonstrando que o caminho da adoção de novos algoritmos está adequado. Diversos erros de funcionalidades foram corrigidos para esta nova versão. E a troca do filme *mastershot*

com as ramificações foi realizada com rapidez sem atrasos. Contudo, faz-se necessário estudar formas de melhorar esta passagem, pois em algumas situações foram geradas uma sensação para o usuário de ruptura. Dependendo do tempo da “Cena”, a passagem entre um filme e outro pareceu artificial. Uma sugestão que foi testada para mudar este sentimento do usuário foi realizar um corte de cena. Isto é, foram introduzidos *frames* pretos bem na hora que o disparo atinge o anteparo de metal. A solução foi satisfatória.

É fato que o treinamento continuado dos profissionais de segurança com relação à abordagem, pode evitar diversos problemas. Um exemplo é o disparo indevido de arma de fogo. Este tipo de situação é um fator de relevância social e, porque não dizer, econômica. O treinamento simulado, além do aumento da frequência com que é aplicado pode evitar o afastamento do profissional de segurança da sua rotina de trabalho por acidente de trabalho. Além disso, pode repercutir negativamente para a imagem da instituição, e até mesmo, sofrer um processo judicial de ressarcimento e danos morais advindo da parte prejudicada.

Quando ocorre um treinamento de tiro real em stands, o mesmo é realizado em pouca quantidade e espaçado no tempo devido aos altos custos envolvidos, além da dificuldade da logística com transporte e mobilização dos grupos de profissionais que precisam ir até o stand de tiro para realizar as atividades. Esses stands não permitem uma interatividade maior, todos utilizam alvos de papel que são inanimados e fixos, não transmitindo a sensação de realidade com a qual o usuário irá realmente se defrontar no momento em que estiver prestando seu serviço. Muitos inclusive acabam não tendo bom desempenho no dia do stand devido à escassez desse tipo de atividade contínua. Fazendo uma analogia com o aprendizado de tocar violão, a ausência de prática o

instrumento continuamente nunca dará a pessoa, à habilidade motora correta e a dinâmica para realmente aprender.

Por fim, é importante destacar que um treinamento continuado é fundamental para permitir que o profissional exerça adequadamente suas atividades, pois gera uma segurança mental e a real compreensão de sua capacidade de reação diante de uma situação de risco. Com o uso do simulador, não se treina somente a destreza motora, mas sim, a capacidade mental de reação instintiva.

7 Agradecimentos

O autor agradece o apoio financeiro do CNPq por meio do programa RHAPE Pesquisador na empresa e PIPE FAPESP. Além do programa de subvenção da FINEP.

Referências

ALVES, D. (2014). O emprego da realidade virtual no treinamento policial para o enfrentamento de criminosos com ênfase nos chamados encontros mortais: uma abordagem baseada na teoria geral de sistemas, Doctoral dissertation, Universidade Federal de Santa Catarina. Disponível em <http://btd.egc.ufsc.br/wp-content/uploads/2014/03/Aur%C3%A9lio-Jos%C3%A9-Pelozato-da-Rosa.pdf> Acesso em Novembro de 2017.

FURUIE, L. M. A. Melhoria na segurança pública com treinamento continuado do policial militar em estande de tiro modelo. 2013. Recuperado em <http://www.acervodigital.ufpr.br/bitstream/handle/1884/30433/R%20-%20D%20-%20LESLY%20MIKI%20ABE%20FURUIE.pdf> Acesso em Novembro de 2017.

GOMES, P. A Simulação nos Desafios do Futuro para a Operacionalidade das Forças de Segurança: o seu Emprego na Guarda Nacional Republicana. 2016. Disponível em

- https://comum.rcaap.pt/bitstream/10400.26/14632/1/TII_Maj%20GNR%20Gomes_A%20Simula%C3%A7%C3%A3o%20na%20GNR_Final.pdf Acesso em Novembro de 2017.
- LUCAS, B. D.; KANADE, T. An iterative image registration technique with an application to stereo vision. In *IJCAI81*, pages 674–679, 1981.
- KASS, M., TERZOPOULOS. D. e WITKIN, A. Snakes: Active contour models. *IJCV*, 1(4):321–331, Jan. 1988.
- MITOV *Software*. VideoLab 2.2. Disponível em: <http://www.mitov.com/html/videolab.html>. Acesso em Maio, 2016.
- O'ROURKE, J. *Computational Geometry in C*. Cambridge University Press Segunda Edição, 1998.
- PINHEIRO, E. B., GOMES, G. A., COUTINHO, E. F., VIDAL, C. A., NETO, J. B. C. Requirements for Development of a Low Cost Portable Simulator for Shooting Skill Training. In *Virtual and Augmented Reality (SVR), 2016 XVIII Symposium on* (pp. 234-238). IEEE, 2016.
- PINTO, P. R. P. (2009). Treino de tiro: análise de situações de treino em contexto policial, Doctoral dissertation, Universidade da Beira Interior. Disponível em <https://ubibliorum.ubi.pt/handle/10400.6/1805>. Acesso em
- ROSA, A. J. P., PAVANATI, I. A utilização da realidade virtual e aumentada na formação dos policiais militares em Santa Catarina. *Revista Ordem Pública*, 7(2), 37-51, 2015.
- SANDES, W. F. Uso não-letal da Força na Ação Policial: Inteligência, Pesquisa, Tecnologia e Intervenção Sócio-Educativa. *Fórum Brasileiro de Segurança Pública*, 24, 2007.
- VALERIO NETTO, A. Development of a security training system based on interactive multimedia. *Revista IEEE América Latina, São Paulo*, v. 4, n.5, p. 379-384, 2006.
- VALERIO NETTO, A. Application of Interactive Technology for Training in the Security Area. In *Virtual and Augmented Reality (SVR), 2015 XVII Symposium on* (pp. 127-132). IEEE, 2015.
- VALERIO NETTO, A. Desenvolvimento de rastreamento de máscaras em vídeo para simulador de treinamento. *REVISTA DE SISTEMAS E COMPUTAÇÃO - RSC*, v. 7, p. 128-146, 2017.
- VALERIO NETTO, A. Desenvolvimento e implementação do protótipo de malha laser para simulador de treinamento na área de segurança. *REVISTA CIÊNCIA E TECNOLOGIA*, v. 20, p. 73-82, 2017b.
- XAVIER, F. M. A importância da formação na mudança de paradigmas no uso da força. *Cadernos Temáticos da Conseg*. 2009. Disponível em http://www.justica.gov.br/sua-seguranca/seguranca-publica/analise-e-pesquisa/download/outras_publicacoes/pagina-2/5cadernotematico_uso-progressivo-da-forca.pdf Acesso em Novembro, 2016.
- XU, C. & PRINCE, J. Snakes, shapes, and gradient vector flow. *IEEE Trans. Image Proc.*, 7(3):359–369, Mar. 1998.