Reconhecimento de rascunhos *offline* em cenários para simulação de corpos rígidos

M.F. Bouzon, R. Zulli, A. Andrijauskas, E. Grassl, F.M. Lopes, R.M. Santos, P.S. Rodrigues
Computer Science Department, Centro Universitário FEI
São Bernardo do Campo, SP, Brazil
{unifmbouzon,unifrsantos,psergio}@fei.edu.br, rafael.zulli0@gmail.com,
adriana.jks@gmail.com, ebgrassl@uol.com.br, fernandomoraes.lopes@gmail.com.

***Resumo*—O reconhecimento de rascunhos à mão livre é uma pré-tarefa importante para aplicações de simulações físicas. A interpretação de uma primitiva geométrica pode ser uma tarefa simples ou demasiadamente complexa, dependendo da orientação da figura e do ângulo de perspectiva da câmera. Este trabalho propõe 7 modelos de *Deep learning*, para reconhecimento de rascunhos feitos à mão, que são comparados e analisados. Além disso, é proposta uma base de dados chamada *PhySketch*, contendo 9.008 rascunhos de elementos naturais e, a partir destes, 359.784 rascunhos artificiais. De todos os modelos analisados, o modelo $PHS{-}TA_8$ obteve a melhor capacidade de detecção, com mAP de 79.31% em cenários naturais mostrando-se invariante à escala, distorção, localização e orientação dos elementos em cenários de ruídos variados.**

***Abstract*—Sketch Recognition is an important pre-task for physical simulations applications. Interpretation of a geometric primitive can be a simple or overly complex task depending on the orientation of the figure and the perspective angle of the camera. This work proposes 7 Deep learning models, for recognition of hand-drawn sketches, which are compared and analyzed. In addition, a database called *PhySketch* is proposed, containing 9,008 drafts of natural elements and, from these, 359,784 artificial drafts. Of all the analyzed models, the $PHS - TA_8$ model obtained the best detection capacity, with mAP of 79.31 % in natural scenarios showing invariant scale, distortion, location and orientation of the elements in scenarios of varied noises .**

I. Introdução

Simulações computacionais são ferramentas importantes no ambiente acadêmico por facilitar a abstração de conceitos complexos. Ao longo do tempo, essas ferramentas têm aberto novas oportunidades de aprendizado que estendem as capacidades de tecnologias tradicionais de ensino [1]. Atualmente, currículos de ciências naturais para a educação básica já integram simuladores que abrangem uma grande variedade de conteúdos na apresentação de seus conceitos [2]. O ensino de conteúdos científicos, em especial os relacionados à física, são enriquecidos pelo uso de simuladores por facilitar o entendimento e propiciar maior visualização dos conceitos representados [3]. Alguns dos simuladores desenvolvidos para o ensino de conteúdos científicos contam com a possibilidade de interação utilizando rascunhos produzidos à mão livre. Estes simuladores aproveitam o fato de que a visualização e o ato de desenhar auxiliam na solidificação deste tipo de conceito [4]. Entretanto, o acesso a interfaces deste gênero pode em muitos casos ser limitada pelo uso de tecnologias nem sempre presentes nos variados ambientes de aprendizagem. Sendo assim, a possibilidade da utilização de rascunhos *offline*, rascunhos produzidos sem auxílio de sistemas computadorizados, em simuladores, impactaria a abrangência e o acesso a este tipo de metodologia de ensino. Portanto, o presente trabalho propõe o desenvolvimento de uma técnica para reconhecimento e interpretação de rascunhos *offline* de cenários de simulação de corpos rígidos.

O uso de desenhos produzidos em sistemas computadorizados, como forma de interação homem-máquina, estabeleceu-se inicialmente com a apresentação do sistema *SketchPad* proposto em [5]. Esse método de interação fomentou estudos na área de reconhecimento de gestos e, por serem áreas de natureza similar, adaptações de suas técnicas podem ser observadas com frequência em trabalhos com enfoque no reconhecimento de rascunhos.

Uma dessas técnicas com uso em ambas as áreas é o *Rubine Classifier* que foi descrito em [6] como uma metodologia para o reconhecimento de gestos. Esta técnica, posteriormente adaptada para o reconhecimento de rascunhos, utiliza um classificador linear que distingue uma série de características visuais extraídas de exemplos previamente disponibilizados, sendo capaz de detectar gestos compostos de um segmento.

Pode-se observar a aplicação do *Rubine Classifier* em sistemas de reconhecimento de rascunhos como DENIM, que foi descrito em [7] e utiliza esse classificador para auxiliar *web designers* nas fases iniciais de projeto, disponibilizando um ambiente de simulação de interação com páginas criadas a partir de rascunhos. Após a criação dos rascunhos no ambiente, o reconhecedor detecta componentes comuns de *websites* como *links* e botões, e produz uma simulação interativa do comportamento esperado de uma página *web*.

O reconhecimento de rascunhos *offline*, desenhados em um quadro branco, foi feito nos trabalhos [8] e [9] com sistemas que permitem interação humana. No trabalho de [10] foi feito algo semelhante, porém os rascunhos reconhecidos foram textos manuscritos. O autor [11] propôs um método para reconhecimento de fluxogramas manuscritos, extraindo características visuais do rascunho. Posteriormente, [12] apresentou uma técnica para produzir um fluxograma computadorizado, a partir da interpretação de uma foto contendo formas e conectores.

A área de Realidade Aumentada também pode se beneficiar de técnicas de reconhecimento de rascunhos *offline*. [13]

propôs um sistema de estimativa de posição de rascunhos feitos em superfícies planas para aplicações de Realidade Aumentada, onde é utilizada a câmera de um *smartphone* para capturar imagens que são projetadas em formato 3D sobre os respectivos rascunhos.

Construir simulações a partir de rascunhos feitos à mão é uma demanda na área de reconhecimento de rascunho *offline*. Nessa linha, simulação de máquina de Turing escritas manualmente foi feita em [14], onde uma foto é capturada por um dispositivo móvel sendo então enviada para um sistema de reconhecimento.

Assim, neste trabalho, foram propostos sete modelos baseados em Redes Neurais Convolutivas para reconhecimento de rascunhos feitos à mão, com aplicações em simulações físicas. O presente trabalho está organizado da seguinte maneira: A Seção II é descrito o método *You Only Look Once V2*. Na Seção III é apresentada a metodologia proposta. A Seção IV descreve a base de dados desenvolvida nesse trabalho. Na Seção V são mostrados os resultados obtidos pelos experimentos e a Seção VI apresenta a conclusão do trabalho.

II. You Only Look Once V2

You Only Look Once (YOLO) é uma técnica de detecção de objetos em tempo real baseada em Redes Neurais Convolutivas (RNC) proposta em [15] que apresenta resultados ao nível do estado da arte para detecção em bases de cenas naturais como VOC2007.

Em [16], uma nova versão do detector de objetos YOLO, a *You Only Look Once V2* (YOLOv2), foi apresentada. A arquitetura da YOLOv2 é composta por 19 camadas convolutivas e 5 camadas do tipo *Maxpool*, que realizam uma redução da dimensionalidade dos *feature maps* ao aplicar um filtro que extrai valores máximos de sub-regiões disjuntas com objetivo de reduzir o custo computacional do detector e auxiliar na redução de modelos que apresentem comportamentos *overfit*.

A metodologia de predição na arquitetura da YOLOv2 foi alterada para gerar uma maior estabilidade e precisão na localização de dimensionamento de *bounding boxes*, sendo duas mudanças as mais relevantes: a utilização de *anchor boxes* e a utilização de funções logísticas para determinação de posição e dimensão relativa a célula preditiva.

Uma *anchor box* é definida como um tensor $A = (p_w, p_h)$ que descreve dimensões de uma região retangular genérica que, ao ser redimensionada para determinação do tamanho de uma *bounding box*, evita a necessidade de predição de dimensões absolutas. A segunda mudança altera a metodologia de cálculo de posicionamento e dimensionamento de uma *bounding box* com intuito de tornar o treinamento e predição de processos mais estáveis. Para uma predição de uma *bounding box* qualquer $B = (t_x, t_y, t_w, t_h, t_o)$ em uma célula de posição (c_x, c_y), a determinação da posição (b_x, b_y) e da dimensão (b_w, b_h) final da *bounding box* detectada pela YOLOv2 é descrita pela Equação (1).

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \\ b_w &= p_w \epsilon^{t_w} \\ b_h &= p_h \epsilon^{t_h} \end{aligned} \tag{1}$$

O uso de *anchor boxes* associados à predição de fatores relativos de localização e dimensão torna mais simples a tarefa de aprendizado da rede neural e gera aproximadamente 5% de aumento na métrica média de *Average Precision* (mAP), proposta em [17], nos modelos testados por [16].

III. Metodologia

A partir de rascunhos *offlines* de cenário de simulação, as técnicas utilizadas nesta metodologia produzem uma detecção que descreve a localização, com uso de *bounding boxes*, e a classificação de cada elemento presente no cenário. A extração de características e geração de simulações interativas não são contempladas por este trabalho. Um fluxo descrevendo como a metodologia apresentada realiza o reconhecimento é mostrado na Figura 2.

O método proposto trabalha em um domínio definido por um conjunto de rascunhos manuscritos de figuras bidimensionais que representam cenários de simulação de corpos rígidos. Um cenário de simulação é definido pela composição de N elementos dispostos de forma coerente seguindo as restrições do domínio. Estes elementos são divididos em duas classes: primitivas geométricas e comandos. A Figura 1 mostra um exemplo de um cenário de simulação com seus elementos dispostos.

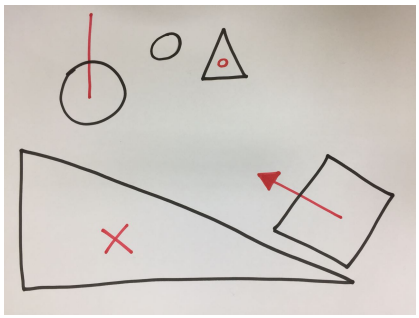

Figura 1. Exemplo de disposições de primitivas geométricas e comandos em um cenário de simulação. Fonte: Autor, 2018

Primitivas geométricas descrevem corpos rígidos, sendo adotados os seguintes tipos de primitivas: círculo, quadrado, triângulo equilátero, triângulo retângulo e triângulo obtusângulo. Os comandos estão sempre associados a uma primitiva e são adotados os seguintes comandos: Letra 'x', vetor, corda e círculo.

Na metodologia proposta foram comparados sete modelos desenvolvidos utilizando como base a arquitetura da Rede Neural Convolutiva para detecção de objetos YOLOv2 apresentada na Seção II. A metodologia de construção destes modelos segue a mesma premissa apresentada em [16] utilizada

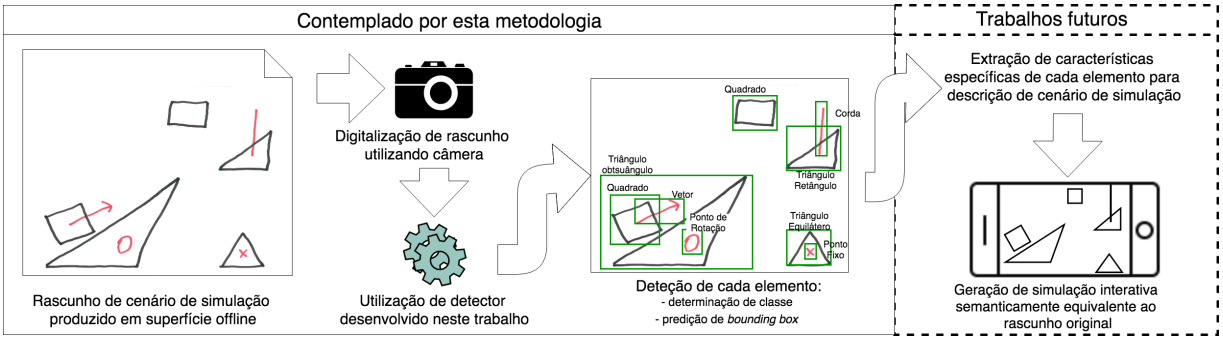

Figura 2. Esquema geral da metodologia proposta. Fonte: Autor, 2018

na comparação das arquiteturas que culminaram no desenvolvimento da YOLOv2. Os sete modelos foram concebidos com o intuito de avaliar o impacto de variadas estratégias na melhora de performance de detecção. Cada estratégia foi incorporada incrementalmente sobre o modelo original YOLOv2, neste trabalho denominado PHS, seguindo uma hierarquia de modelos, baseada nas modificações dessa arquitetura. O treinamento de todos os modelos seguiu uma metodologia idêntica. Todos foram treinados durante 85 épocas nas imagens da base de treinamento B_t.

Uma descrição de cada modelo é dada na lista a seguir:

- PHS é a reprodução sem modificações da arquitetura YOLOv2.
- $PHS-T$ possui arquitetura equivalente a PHS. Entretanto, aplica-se a técnica de transferência de aprendizado, proposta em [18].
- $PHS-TA_5$ utiliza o conjunto de *anchor boxes* A_5 e possui a mesma quantidade de *anchor boxes* da arquitetura original YOLOv2.
- $PHS-TA_2$ utiliza o conjunto de *anchor boxes* A_2 e possui quantidade menor de *anchor boxes* do que a arquitetura original.
- $PHS-TA_8$ utiliza o conjunto de *anchor boxes* A_8 e possui quantidade maior de *anchor boxes* do que utilizado a arquitetura original.
- $PHS-TA_5G$ é equivalente ao modelo $PHS-TA_5$. Sua diferença consiste na modificação do estilo de treinamento empregado, que generaliza as classes do tipo Triângulo presentes no domínio.
- $PHS-TA_5G+$ é a reprodução do modelo $PHS-TA_5G$ e também emprega a generalização da classe Triângulo. Sua diferença está na modificação das dimensões da camada de entrada da rede de 416×416 para 640×640.

IV. *PhySketch Dataset*

Até onde sabemos, não há bases de dados de rascunhos *offline* que atendam às necessidades do domínio descrito na Seção III. Portanto, para este trabalho, foi desenvolvida a base de dados *PhySketch Dataset*[1] para validar a metodologia proposta em cenários para simulação de corpos rígidos.

[1]Disponível em: https://github.com/PhySketch

A referida base de dados é definida aqui pelo conjunto B de imagens de rascunhos *offline* que, por sua vez, é composto por dois subconjuntos: rascunhos de cenários de simulação, B_c, e rascunhos de elementos, B_e.

A coleta de dados foi feita por voluntários, que desenharam imagens de rascunhos *offline* para compor a base de dados. A coleta de cenários de simulação e elementos (primitivas e comandos), ocorreu com a aplicação de três atividades padronizadas. Foram aplicadas 73 atividades que geraram ao todo, aproximadamente, 376 rascunhos de cenários de simulação e 5.529 rascunhos de elementos individuais.

Além disso, foi feita uma extensão artificial da base de dados coletada, uma vez que foi gerada uma quantidade insuficiente de rascunhos de cenários de simulação. A expansão foi feita com a mesma técnica utilizada em [19] e foi gerado ao todo 25.800 cenários sintéticos, sendo 70% destes utilizado para a base de treinamento e 30% para a base de validação.

V. Resultados obtidos

Foi feito um experimento para a validação da metodologia proposta com o objetivo de comparar e avaliar a performance de cada modelo desenvolvido tanto com imagens naturais de cenário de simulação e quanto com sintéticas. Essa avaliação processou as detecções realizadas por cada modelo sobre a base de cenários naturais B_c e sobre a base de cenários sintéticos para validação B_v. Esse processo ocorreu com a comparação das *bounding boxes* geradas pela predição do modelo de detecção com as *bounding boxes* anotadas.

A avaliação dos modelos ocorreu com a geração de valores de *average precision* (AP) [17]. Além disso, para facilitar a comparação entre modelos e seguindo as técnicas de avaliação apresentadas em trabalhos de natureza similar, o *mean Average Precision* foi computado seguindo a metodologia de avaliação específica (mAP_e) e avaliação generalizada (mAP_g), onde a mAP_e considera todas as 9 classes definidas pelo domínio de rascunho apresentado na Seção III e a mAP_g une todas as três classes específicas que representam os triângulos em uma única classe genérica "Triângulo". Seguindo esta metodologia de avaliação, computou-se valores de mAP_e e mAP_g para os modelos PHS, $PHS-T$, $PHS-TA_2$, $PHS-TA_5$, $PHS-TA_8$ uma vez que estes são capazes de gerar classificações específicas para a classe Triângulo.

Por outro lado, considerando que os modelos $PHS - TA_5G$ e $PHS - TA_5G+$ são incapazes de detectar essas classes específicas, foram gerados apenas valores de mAP_g oriundos da avaliação generalizada. A Tabela I apresenta os resultados deste experimento.

Tabela I
TABELA QUE MOSTRA OS RESULTADOS DA AVALIAÇÃO GENERALIZADA mAP_g E DA AVALIAÇÃO ESPECÍFICA mAP_e DE CADA MODELO AO REALIZAREM PREDIÇÕES SOBRE AS BASES B_c E B_v. EM NEGRITO, A MELHOR PERFORMANCE DE DETECÇÃO APRESENTADA EM CADA BASE PARA A LINHA.

Classe	Avaliação Generalizada		Avaliação Especifíca	
	mAP_g		mAP_e	
	B_c	B_v	B_c	B_v
PHS	64.71	73.46	57.47	67.44
$PHS - T$	67.51	75.92	55.66	68.87
$PHS - TA_5$	68.80	71.48	60.64	64.15
$PHS - TA_2$	33.39	38.42	26.36	30.78
$PHS - TA_8$	**79.31**	**82.41**	**70.73**	**76.58**
$PHS - TA_5G$	65.83	70.36	-	-
$PHS - TA_5G+$	69.69	76.49	-	-

VI. CONCLUSÃO

O presente trabalho discorreu sobre a detecção automática de rascunhos *offline* de cenários de simulação, com intuito de desenvolver uma técnica capaz de extrair informações de rascunhos a mão livre, visando aumentar as possibilidades de formas de interação com simuladores de corpos rígidos já utilizados em sala de aula para ensino de conceitos de física mecânica.

Duas grandes contribuições foram desenvolvidas ao decorrer do projeto: uma base de rascunhos *offline* de cenários de simulação de corpos rígidos denominada *PhySketch* e um modelo capaz de detectar os cenários naturais, presentes nesta base, apresentando precisão de detecção mAP de 79.31%.

A base de rascunhos *PhySketch* é composta por 9.008 elementos naturais anotados manualmente contendo informações semânticas que descrevem cada elemento, seguindo uma padronização determinada. Essas informações podem ser utilizadas em trabalhos futuros para desenvolvimento de técnicas de classificação, técnicas de detecção e técnicas de reconhecimento e descrição, com o uso de informações precisas sobre a representação visual de cada elemento.

Foram apresentados 7 modelos treinados e avaliados por 3 experimentos neste trabalho. Todos estes modelos são alterações da RNC YOLOv2, contendo modificações em sua arquitetura com intuito de avaliar o impacto na precisão de detecção de rascunhos e selecionar a combinação mais bem sucedida. Após avaliação, concluiu-se que o modelo que melhor realiza a tarefa de detecção de rascunhos *offline* de cenários de simulação, nas condições apresentadas pelo domínio, é o $PHS - TA8$ que mostrou precisão superior em todas as comparações.

Espera-se que com as técnicas apresentadas neste trabalho, seja possível o desenvolvimento de novas metodologias de ensino que aproveitem as capacidades apresentadas pelos detectores apresentados. Ao associar estes modelos à aplicações que demonstrem simulações de corpos rígidos, será possível prover uma experiência de aprendizado de conceitos de física mecânica que utilizem uma forma de interação inovadora.

VII. AGRADECIMENTOS

Agradecimentos à FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo - Processo nº 2018/02612-7), CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) e ao Centro Universitário da FEI.

REFERÊNCIAS

[1] V. Heckler, M. d. F. O. Saraiva, and K. d. S. Oliveira Filho, "Uso de simuladores, imagens e animações como ferramentas auxiliares no ensino/aprendizagem de óptica," 2007.

[2] F. Esquembre, "Computers in physics education," *Computer physics communications*, vol. 147, no. 1-2, pp. 13–18, 2002.

[3] A. Serrano and V. Engel, "Uso de simuladores no ensino de física: um estudo da produção gestual de estudantes universitários," *RENOTE*, vol. 10, no. 1, 2012.

[4] M. Brooks, "Drawing, visualisation and young children's exploration of "big ideas"," *International Journal of Science Education*, vol. 31, no. 3, pp. 319–341, 2009.

[5] I. E. Sutherland, "Sketchpad a man-machine graphical communication system," *Transactions of the Society for Computer Simulation*, vol. 2, no. 5, pp. R–3, 1964.

[6] D. Rubine, "The automatic recognition of gestures," Ph.D. dissertation, Carnegie Mellon University, 1991.

[7] J. Lin, M. W. Newman, J. I. Hong, and J. A. Landay, "Denim: finding a tighter fit between tools and practice for web site design," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 2000, pp. 510–517.

[8] E. Saund, "Bringing the marks on a whiteboard to electronic life," in *International Workshop on Cooperative Buildings*. Springer, 1999, pp. 69–78.

[9] Q. Stafford-Fraser and P. Robinson, "Brightboard: A video-augmented environment," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1996, pp. 134–141.

[10] M. Wienecke, G. A. Fink, and G. Sagerer, "Toward automatic video-based whiteboard reading," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 7, no. 2-3, pp. 188–200, 2005.

[11] M. Notowidigdo and R. C. Miller, "Off-line sketch interpretation," in *AAAI fall symposium*, 2004, pp. 120–126.

[12] J. Wu, C. Wang, L. Zhang, and Y. Rui, "Offline sketch parsing via shapeness estimation." in *IJCAI*, vol. 15, 2015, pp. 1200–1206.

[13] N. Hagbi, O. Bergig, J. El-Sana, and M. Billinghurst, "Shape recognition and pose estimation for mobile augmented reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 10, pp. 1369–1379, Oct 2011.

[14] J. Browne and T. Sherwood, "Mobile vision-based sketch recognition with spark," in *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*. Eurographics Association, 2012, pp. 87–96.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 779–788.

[16] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 6517–6525.

[17] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun 2010.

[18] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3320–3328.

[19] H. Su, X. Zhu, and S. Gong, "Deep learning logo detection with data expansion by synthesising context," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2017, pp. 530–539.