

Auto Feature Weight for Interactive Image Segmentation using Particle Competition and Cooperation

Fabricio Breve

Department of Statistics, Applied Mathematics and Computation (DEMAC)

Institute of Geosciences and Exact Sciences (IGCE)

São Paulo State University (UNESP)

Rio Claro, São Paulo, Brazil

Email: fabricio@rc.unesp.br

Abstract—Particle competition and cooperation (PCC) is a graph-based semi-supervised learning algorithm. It employs particles walking in a graph to classify data items corresponding to graph nodes. PCC may be applied to the interactive image segmentation problem, in which a human specialist labels some pixels of the image, and the algorithm propagates them to the remaining pixels, splitting the foreground and the background. In this scenario, pixels are converted into graph nodes, and each node is connected to its k -nearest neighbors, according to a set of features extracted from the image. However, some features may be more important than others to identify the segments, depending on the particularities of the image. In this paper, four methods to automatically weight the features to the neighbor selecting task are presented. Computer simulations are performed on real-world images from the Microsoft GrabCut dataset, and the segmentation results show the effectiveness of the proposed methods.

I. INTRODUÇÃO

Segmentação de imagens é o processo de dividir uma imagem digital em múltiplas partes, identificando objetos ou outras informações relevantes [1]. Tal tarefa é considerada uma das mais difíceis no processamento de imagens [2]. Além disso, a segmentação completamente automática ainda é um grande desafio, pois os métodos existentes frequentemente são dependentes de domínio, não podendo ser aplicados à qualquer imagem. Portanto, a segmentação interativa de imagens, que conta com a supervisão parcial de um especialista, tornou-se uma abordagem de bastante interesse nas últimas décadas [3]–[5]

Muitas abordagens de processamento interativo de imagens são baseadas em aprendizado semi-supervisionado. Este tipo de aprendizado é utilizado em problemas em que dados não rotulados são abundantes, mas o processo de rotulá-los é caro e/ou demorado, requerendo o trabalho intenso de especialistas humanos [6], [7]. As técnicas de aprendizado semi-supervisionado empregam ambos os dados rotulados e não rotulados em seu processo de treinamento, superando limitações das técnicas de aprendizado supervisionado e não-supervisionado, que utilizam apenas dados rotulados ou não rotulados, respectivamente. Na tarefa de segmentação interativa,

os algoritmos de aprendizado semi-supervisionado propagam os rótulos dos pixels rotulados para os pixels não rotulados da imagem com base na similaridade entre eles.

Competição e cooperação entre partículas [8] é um modelo de aprendizado semi-supervisionado baseado em grafos que, apesar de relativamente novo, já foi aplicado com sucesso em várias tarefas importantes de aprendizado de máquina, como detecção de comunidades sobrepostas [9], [10], aprendizado com ruído nos rótulos [11]–[13], aprendizado com mudança de conceito [14], [15], e aprendizado ativo e semi-supervisionado combinados [16]–[18].

O modelo de partículas é diferente da maioria dos outros métodos de aprendizado semi-supervisionado baseados em grafos [19]–[24], que são similares e compartilham o mesmo *framework* de regularização [6]. Enquanto a maioria dos modelos propagam rótulos de forma global, de todos os vértices para todos os vértices de acordo com o peso das arestas, o modelo de partículas utiliza uma abordagem local de propagação de rótulos, através das partículas, que selecionam e alteram apenas um nó de interesse a cada movimento. Consequentemente, seu custo computacional é próximo de linear ($O(N)$) na *loop* principal, onde N é a quantidade de pixels da imagem, enquanto muitos outros métodos do estado da arte tem complexidade cúbica ($O(N^3)$).

Na tarefa de segmentação interativa, o modelo de partículas é aplicado da seguinte forma. Um grafo é construído de forma que cada pixel é representado por um vértice. Arestas são criadas entre vértices que correspondem a pixels similares. Em seguida, partículas, que correspondem aos pixels rotulados, caminham pelo grafo cooperando com outras partículas que representam o mesmo segmento, e competindo com partículas que representam segmentos diferentes. Cada grupo de partículas tenta dominar a maior quantidade possível de vértices não rotulados, espalhando seus rótulos, e evitando que partículas inimigas invadam seu território. No final do processo, é esperado que as fronteiras entre os territórios coincidam com as fronteiras entre os segmentos.

No estágio de formação do grafo, a conexão entre os vértices é dada pela semelhança entre os pixels correspondentes, dada pela distância Euclidiana entre os atributos dos pixels,

This work was supported by the São Paulo State Research Foundation (FAPESP) and the Brazilian National Research Council (CNPq).

os quais são extraídos da imagem. Uma grande variedade de atributos pode ser extraída de um pixel, incluindo componentes de cores RGB (vermelho, verde e azul), brilho, saturação, matiz, além de medidas derivadas e medidas que levam em consideração a localização e vizinhança espacial dos pixels na imagem. Dependendo da imagem, cada atributo pode ter maior ou menor capacidade de discriminar as classes/objetos de interesse. Portanto, é interessante ponderar cada atributo de acordo com sua capacidade de discriminação com o objetivo de maximizar a capacidade de discriminação do algoritmo de partículas como um todo. Infelizmente, porém, não há uma maneira única ou infalível de definir tais pesos para os atributos.

Neste artigo, são propostos quatro métodos para definição de pesos para os atributos a serem utilizados no modelo de competição e cooperação entre partículas. Os vértices do grafo, representando cada pixel, são conectados de acordo com a distância Euclidiana entre os atributos ponderados pelos pesos definidos. Então, o algoritmo de partículas é aplicado para obter a imagem segmentada. O desempenho do algoritmo utilizando cada um dos quatro métodos de definição automática de pesos é aferido e comparado com o desempenho obtido quando o método de partículas é aplicado diretamente sobre todos os atributos sem ponderação.

As simulações computacionais são realizadas em algumas imagens do mundo real, extraídas da base de dados Microsoft GrabCut [25], e o desempenho absoluto de classificação é obtido comparando a imagem de saída do algoritmo de partículas com a imagem *ground truth* da base de dados, rotulada manualmente por especialistas humanos. Os resultados mostram e comparam a eficácia dos quatro métodos propostos.

O restante deste artigo é organizado como segue. A Seção II apresenta o modelo de competição e cooperação entre partículas. Na Seção III, são apresentados os métodos de definição automática de pesos propostos. A Seção IV apresenta algumas simulações computacionais. Finalmente, algumas conclusões são apresentadas na Seção V.

II. COMPETIÇÃO E COOPERAÇÃO ENTRE PARTÍCULAS PARA SEGMENTAÇÃO DE IMAGENS

De maneira geral, o método de aprendizado semi-supervisionado de competição e cooperação entre partículas [8] pode ser aplicado em segmentação de imagens da seguinte forma. Cada pixel da imagem é convertido em um vértice de um grafo sem pesos e não direcionado. Arestas entre pares de vértices são criadas entre pixels semelhantes, de acordo com a distância Euclidiana entre os atributos extraídos do pixel. Em seguida, uma partícula é criada para cada vértice rotulado, ou seja, vértices para qual o pixel correspondente já tem uma classe atribuída. Partículas representando vértices com o mesmo rótulo compõem um time, e cooperam entre si para dominar vértices não rotulados. Finalmente, conforme o sistema executa, as partículas caminham pelo grafo seguindo uma regra aleatório-gulosa.

Cada vértice tem um conjunto de níveis de domínio, sendo um nível para cada time. Quando uma partícula visita um vértice, ela aumenta o nível de domínio de seu time nele, ao mesmo tempo em que reduz o nível de domínio dos demais times. Cada partícula tem um nível de força, o qual muda

de acordo com o nível de domínio de seu time no nó em que ela está visitando. Cada partícula também tem uma tabela que armazena a distância entre seu vértice de origem e cada vértice que ela tenha visitado. Estas tabelas são atualizadas dinamicamente conforme a partícula caminha. No final do processo iterativo, cada vértice (e o correspondente pixel) recebe o rótulo do time que nele possui o nível de domínio mais alto.

De cada pixel x_i , é extraído um conjunto de 23 atributos, que são: a linha e a coluna da imagem em que o pixel está localizado, cada um dos componentes RGB e HSV da imagem, as médias de cada um dos seis componentes RGB e HSV no pixel e na sua adjacência, os desvios padrão de cada um dos seis componentes RGB e HSV no pixel e sua adjacência, e os componentes ExR, ExG e ExB, dados por:

$$\begin{aligned} \text{ExR} &= (2R - (G + B)), \\ \text{ExG} &= (2G - (R + B)), \\ \text{ExB} &= (2B - (G + R)), \end{aligned} \quad (1)$$

como usado no Image Segmentation Data Set [26]. Deve ser observado que para as medidas considerando pixels adjacentes, são considerados os 8 vizinhos conectados na imagem, exceto nas bordas onde nenhuma “dobra” é aplicada. Todos os componentes são normalizados para terem média 0 e desvio padrão 1. Os componentes HSV foram obtidos a partir dos componentes RGB, utilizando o método descrito em [27].

O grafo não direcionado e sem pesos é gerado tal que $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, em que $\mathbf{V} = \{v_1, v_2, \dots, v_N\}$ é o conjunto de vértices, e \mathbf{E} é o conjunto de arestas (v_i, v_j) . Cada vértice v_i corresponde ao pixel x_i . Dois vértices v_i e v_j estão conectados se v_j está entre os k vizinhos mais próximos de v_i , ou vice-versa, usando a distância Euclidiana entre os atributos de x_i e x_j . Caso contrário, v_i e v_j estão desconectados.

Com o grafo gerado, o algoritmo de competição e cooperação entre partículas é aplicado da maneira tradicional, conforme descrito em [8]. Porém, para reduzir a complexidade de armazenamento e permitir a aplicação em imagens maiores, em vez de utilizar uma tabela de distância para cada partícula, são utilizadas apenas uma tabela de distâncias para cada time, e todas as partículas do mesmo time compartilham da mesma tabela. Com tal modificação, a complexidade de armazenamento que originalmente era $O(NP)$, onde N é a quantidade de vértices e P é a quantidade de partículas (pixels rotulados), passa a ser apenas $O(NC)$, onde C é a quantidade de classes do problema.

As novas tabelas de distância são definidas como $\mathbf{d}_c(\mathbf{t}) = \{d_c^1(t), \dots, d_c^N(t)\}$. Cada elemento $d_c^i(t) \in [0, N-1]$ armazena a distância em saltos entre o vértice v_i e o vértice mais próximo da classe c . Partículas iniciam sabendo apenas que a distância para vértices rotulados de sua classe é zero ($d_c^i = 0$ if $y(x_i) = c$). Outras distâncias são inicialmente ajustadas para o maior valor possível ($d_c^i = n - 1$ if $y(x_i) \neq c$) e atualizadas dinamicamente conforme as partículas caminham.

A média dos níveis de domínio mais altos em cada vértice ($\langle v_i^{\omega_m} \rangle$, $m = \arg \max_c v_i^{\omega_c}$) é usada para identificar quando o algoritmo alcançou um nível satisfatório de estabilidade. Neste momento, para cada vértice v_i , se $v_i^{\omega_c} > 0,9$, a classe c é atribuída ao pixel correspondente ($y(x_i) = c$). Os vértices remanescentes (se houver) serão rotulados em uma segunda

fase, pois quando $v_i^{\omega_c} \leq 0,9$, considera-se que o algoritmo de partículas apresenta baixa confiança na classe atribuída. Em alguns casos é até possível que o vértice tenha ficado desconectado, por conta da escolha de um valor baixo para k ou por ser o pixel correspondente um *outlier*, e assim nenhuma partícula conseguiu alcançá-lo.

A segunda fase consiste em um segundo processo iterativo, onde cada pixel não rotulado x_i ajusta o seu correspondente \mathbf{v}_i^ω da seguinte forma:

$$\mathbf{v}_i^\omega(t+1) = \frac{1}{a} \sum_{j \in \eta} \mathbf{v}_j^\omega(t) \text{dist}(x_i, x_j), \quad (2)$$

onde η é o subconjunto dos a pixels adjacentes de x_i . $a = 8$, exceto nas bordas da imagem, que tem somente 3 ou 5 pixels adjacentes. $\text{dist}(x_i, x_j)$ é uma função que retorna a distância Euclidiana entre os atributos x_i e x_j . Portanto, cada pixel não rotulado receberá contribuições de seus pixels adjacentes, proporcionais a similaridade entre eles. A segunda fase termina quando $\langle v_i^{\omega_m} \rangle$ estabiliza. Agora os pixels ainda não rotulados finalmente recebem seu rótulo $y(x_i) = \arg \max_c v_i^{\omega_c}(t)$.

III. AJUSTE AUTOMÁTICO DE PESOS

No estágio de formação do grafo, cada vértice é conectado com seus k vizinhos mais próximos, de acordo com a distância Euclidiana entre os atributos extraídos dos pixels da imagem. Uma grande variedade de atributos pode ser extraída da imagem. Neste artigo são utilizados 23 deles, mas centenas de outros atributos poderiam ser extraídos e utilizados, dependendo do tipo de imagem e da aplicação. Porém, os métodos de aprendizado de máquina não costumam ter bom desempenho ao trabalhar com muitos atributos. Tal problema é conhecido como “maldição da dimensionalidade”. Dependendo da imagem, cada atributo pode ter maior ou menor capacidade de discriminar as classes/objetos de interesse. Portanto, é interessante escolher apenas um subconjunto de características que melhor discrimine as classes de interesse ou mesmo reduzir a dimensionalidade com técnicas como a *principal component analysis* [28].

No caso da competição e cooperação entre partículas, como a distância entre vértices é definida pela distância Euclidiana entre os atributos, também é possível ponderar tais atributos de acordo com sua capacidade de discriminação, de modo que eles tenham maior ou menor influência na definição das distâncias e, conseqüentemente, da vizinhança entre os vértices. Dessa forma, com uma boa escolha de pesos é possível maximizar a capacidade de discriminação do algoritmo de partículas. Infelizmente, porém, não há uma maneira única ou infalível de definir tais pesos. O atributo que pode ser fundamental para discriminar classes em uma determinada imagem, pode ter pouca ou nenhuma capacidade de discriminar classes em outra imagem.

Para melhor compreender e tentar solucionar o problema de ponderar atributos, são propostos quatro métodos que utilizam apenas informação disponível a priori (subconjunto de pixels rotulados pelo usuário) para calcular um peso para cada atributo, com base em algumas hipóteses. Cada atributo é então multiplicado pelo peso definido antes do cálculo da distância Euclidiana. Estes métodos serão descritos a seguir.

A. Diferença entre as Médias Intra-classe

A Diferença entre as Médias Intra-classe (DMI) consiste em obter a média de valores de um atributo f entre os pixels rotulados de cada classe separadamente e então verificar a diferença entre essas médias. A hipótese é de que atributos que discriminem bem entre duas classes terão valores médios bastante diferentes. Dessa forma, o peso w_f será dado por:

$$w_f = \sum_{i|y_i=1} x_{if} - \sum_{i|y_i=2} x_{if} \quad (3)$$

onde x_{if} é o valor do atributo f no pixel x_i , e y_i é o rótulo de x_i .

B. Diferença entre as Médias Intra-classe dividida pela Soma do Desvio Padrão Intra-classe

A Diferença entre as Médias Intra-classe dividida pela Soma do Desvio Padrão Intra-classe (DMI/SDPI) é semelhante à DMI, porém a diferença entre as médias é dividida pela soma dos desvios padrões do atributo em cada classe. A hipótese é de que atributos que discriminem bem entre duas classes terão baixa variância intra-classe. Assim, o peso w_f será dado por:

$$w_f = \frac{\sum_{i|y_i=1} x_{if} - \sum_{i|y_i=2} x_{if}}{\sigma_{f1} + \sigma_{f2}} \quad (4)$$

onde σ_{f1} e σ_{f2} correspondem ao desvio padrão do atributo f nas classes 1 e 2 respectivamente.

C. Diferença entre Histogramas

A Diferença entre Histogramas (DH) é calculada da seguinte forma. Dado um atributo f , para cada subconjunto de pixels de uma mesma classe, um histograma de 10 barras é gerado para mostrar a distribuição dos valores de f no subconjunto. Os histogramas tem seus valores normalizados dividindo cada barra pelo total de pixels rotulados da classe, de modo que a soma das 10 barras seja sempre 1. Em seguida, o peso é definido pela soma das diferenças absolutas entre os valores de cada par de barras (uma de cada histograma). A hipótese é de que atributos que discriminem bem entre as classes terão histogramas bem diferentes, e assim o valor da soma das diferenças será maior. Esta medida também visa destacar diferenças que não fiquem evidentes usando a DMI. Nesse caso, o peso w_f é dado por:

$$w_f = \sum_{b=1}^{10} |h_{b1} - h_{b2}| \quad (5)$$

onde h_{b1} e h_{b2} correspondem às barras b dos histogramas normalizados do atributo f nas classe 1 e 2, respectivamente.

D. Diferença entre Histogramas Acumulados

A Diferença entre Histogramas Acumulados (DHA) é bastante similar à DH. Porém, se utiliza o histograma acumulado, dado por:

$$a_{bc} = \sum_1^b h_{bc} \quad (6)$$

onde a_{bc} corresponde à barra b do histograma acumulado da classe c , e h_{bc} corresponde à barra b do histograma original da classe c . Em seguida, o peso w_f é dado por:

$$w_f = \sum_{b=1}^{10} |a_{b1} - a_{b2}| \quad (7)$$

onde a_{b1} e a_{b2} correspondem às barras b dos histogramas acumulados do atributo f nas classes 1 e 2, respectivamente.

IV. SIMULAÇÕES COMPUTACIONAIS

Nesta seção, simulações computacionais usando imagens do mundo real extraídas da Microsoft GrabCut dataset¹ [25] são apresentadas para verificar a eficiência dos métodos de definição automática de pesos propostos. As três imagens selecionadas da base, os *trimaps* oferecendo as regiões de semente, e as imagens originais de *ground truth* são mostradas na Figura 1.

Os seguintes parâmetros do modelo de competição e cooperação entre partículas são fixados: $p_{grad} = 0,5$ e $\Delta_v = 0,1$, conforme sugerido em outros trabalhos [8], [11]–[13]. O parâmetro k é otimizado para cada imagem e cada método de definição de pesos utilizando o algoritmo genético do *Global Optimization Toolbox* do MATLAB, no intervalo $1 \leq k \leq 1000$.

A Figura 2 mostra os resultados de segmentação do objeto principal obtidos utilizando os métodos propostos, bem como as taxas de erro de classificação. As taxas de erro são computadas como a fração entre a quantidade de pixels classificados incorretamente e o total de pixels não rotulados (cinza claro nas imagens *trimap*). Note que as imagens *ground truth* tem um fino contorno de pixels cinzas, que correspondem à incerteza, ou seja, pixels que receberam rótulos diferentes pelas diferentes pessoas que fizeram a classificação manual. Estes pixels não são computados na taxa de erro de classificação.

Nos resultados da Figura 2, é possível observar que o método DHA foi o melhor nas duas primeiras imagens e o método DMI/SDPI foi o melhor na terceira imagem. Na primeira imagem, apenas o método DHA se mostrou eficaz. Na segunda imagem, ambos DHA e DMI/SDPI se mostraram eficazes. Finalmente, na terceira imagem, DMI e DMI/SDPI se mostraram eficazes. Tais resultados comprovam a eficácia de três dos métodos propostos em pelo menos uma das três imagens propostas, porém não é possível apontar um único método como mais indicado, pois todos também falharam em pelo menos uma das imagens.

V. CONCLUSÕES

Este artigo propõe quatro métodos para definição automática de pesos para atributos, os quais são utilizados para definir um grafo onde é aplicado o modelo conhecido como competição e cooperação entre partículas. Neste método, partículas caminham em um grafo gerado a partir da imagem a ser segmentada. As simulações computacionais com imagens reais mostram que o uso dos métodos propostos aumenta a taxa de classificação correta dos pixels em alguns casos, tornando a abordagem promissora.

Dando continuidade a este trabalho, pretende-se realizar mais simulações, com uma quantidade maior de imagens e atributos, visando obter resultados mais conclusivos. Além disso, pretende-se refinar os métodos de definição automática de pesos que tiveram melhor desempenho, com o objetivo de aumentar a eficácia dos mesmos.

REFERÊNCIAS

- [1] L. Shapiro and G. Stockman, *Computer Vision*. Prentice Hall, 2001.
- [2] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2008.
- [3] Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images,” in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1, 2001, pp. 105–112 vol.1.
- [4] L. Grady, “Random walks for image segmentation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 11, pp. 1768–1783, Nov 2006.
- [5] A. Protiere and G. Sapiro, “Interactive image segmentation via adaptive weighted distances,” *Image Processing, IEEE Transactions on*, vol. 16, no. 4, pp. 1046–1057, April 2007.
- [6] X. Zhu, “Semi-supervised learning literature survey,” Computer Sciences, University of Wisconsin-Madison, Tech. Rep. 1530, 2005.
- [7] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*, ser. Adaptive Computation and Machine Learning. Cambridge, MA: The MIT Press, 2006.
- [8] F. Breve, L. Zhao, M. Quiles, W. Pedrycz, and J. Liu, “Particle competition and cooperation in networks for semi-supervised learning,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no. 9, pp. 1686–1698, sept. 2012.
- [9] F. Breve and L. Zhao, “Fuzzy community structure detection by particle competition and cooperation,” *Soft Computing*, vol. 17, no. 4, pp. 659–673, 2013. [Online]. Available: <http://dx.doi.org/10.1007/s00500-012-0924-3>
- [10] F. Breve, L. Zhao, M. Quiles, W. Pedrycz, and J. Liu, “Particle competition and cooperation for uncovering network overlap community structure,” in *Advances in Neural Networks – ISNN 2011*, ser. Lecture Notes in Computer Science, D. Liu, H. Zhang, M. Polycarpou, C. Alippi, and H. He, Eds. Springer Berlin Heidelberg, 2011, vol. 6677, pp. 426–433. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-21111-9_48
- [11] F. A. Breve, L. Zhao, and M. G. Quiles, “Semi-supervised learning from imperfect data through particle cooperation and competition,” in *Neural Networks (IJCNN), The 2010 International Joint Conference on*, July 2010, pp. 1–8.
- [12] —, “Particle competition and cooperation for semi-supervised learning with label noise,” *Neurocomputing*, 2015, article in Press.
- [13] F. Breve and L. Zhao, “Particle competition and cooperation to prevent error propagation from mislabeled data in semi-supervised learning,” in *Neural Networks (SBRN), 2012 Brazilian Symposium on*, Oct 2012, pp. 79–84.
- [14] —, “Particle competition and cooperation in networks for semi-supervised learning with concept drift,” in *Neural Networks (IJCNN), The 2012 International Joint Conference on*, June 2012, pp. 1–6.
- [15] F. A. Breve and L. Zhao, “Semi-supervised learning with concept drift using particle dynamics applied to network intrusion detection data,” in *Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI CBIC), 2013 BRICS Congress on*, Sept 2013, pp. 335–340.
- [16] F. Breve, “Active semi-supervised learning using particle competition and cooperation in networks,” in *Neural Networks (IJCNN), The 2013 International Joint Conference on*, Aug 2013, pp. 1–6.
- [17] F. A. Breve, “Combined active and semi-supervised learning using particle walking temporal dynamics,” in *Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI CBIC), 2013 BRICS Congress on*, Sept 2013, pp. 15–20.

¹Disponível em <http://research.microsoft.com/en-us/um/cambridge/projects/visionimagevideoediting/segmentation/grabcut.htm>

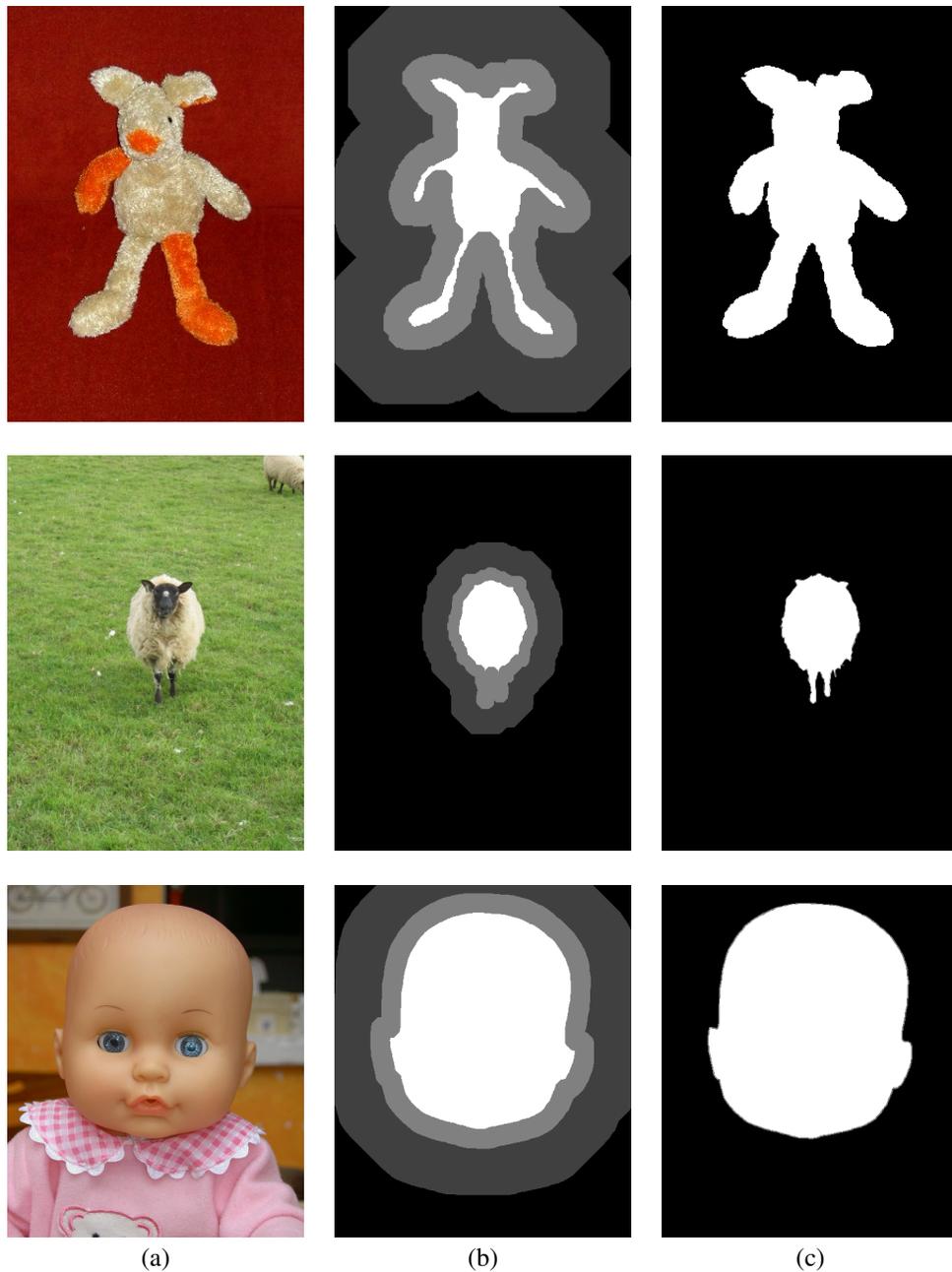


Figura 1. (a) Imagens originais do GrabCut dataset, (b) os *trimaps* fornecendo as regiões semente, e (c) as imagens *ground truth* originais.

[18] —, “Query rules study on active semi-supervised learning using particle competition and cooperation,” in *Anais do Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)*, São Carlos, 2014, pp. 134–140.

[19] A. Blum and S. Chawla, “Learning from labeled and unlabeled data using graph mincuts,” in *Proceedings of the Eighteenth International Conference on Machine Learning*. San Francisco: Morgan Kaufmann, 2001, pp. 19–26.

[20] X. Zhu, Z. Ghahramani, and J. Lafferty, “Semi-supervised learning using gaussian fields and harmonic functions,” in *Proceedings of the Twentieth International Conference on Machine Learning*, 2003, pp. 912–919.

[21] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, “Learning with local and global consistency,” in *Advances in Neural Information Processing Systems*, vol. 16. MIT Press, 2004, pp. 321–328. [Online]. Available: <http://www.kyb.tuebingen.mpg.de/bs/people/weston/localglobal.pdf>

[22] M. Belkin, I. Matveeva, and P. Niyogi, “Regularization and semisupervised learning on large graphs,” in *Conference on Learning Theory*. Springer, 2004, pp. 624–638.

[23] M. Belkin, N. P., and V. Sindhwani, “On manifold regularization,” in *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTAT 2005)*. New Jersey: Society for Artificial Intelligence and Statistics, 2005, pp. 17–24.

[24] T. Joachims, “Transductive learning via spectral graph partitioning,” in *Proceedings of International Conference on Machine Learning*. AAAI Press, 2003, pp. 290–297.

[25] C. Rother, V. Kolmogorov, and A. Blake, ““grabcut”: Interactive foreground extraction using iterated graph cuts,” *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, Aug. 2004. [Online]. Available:

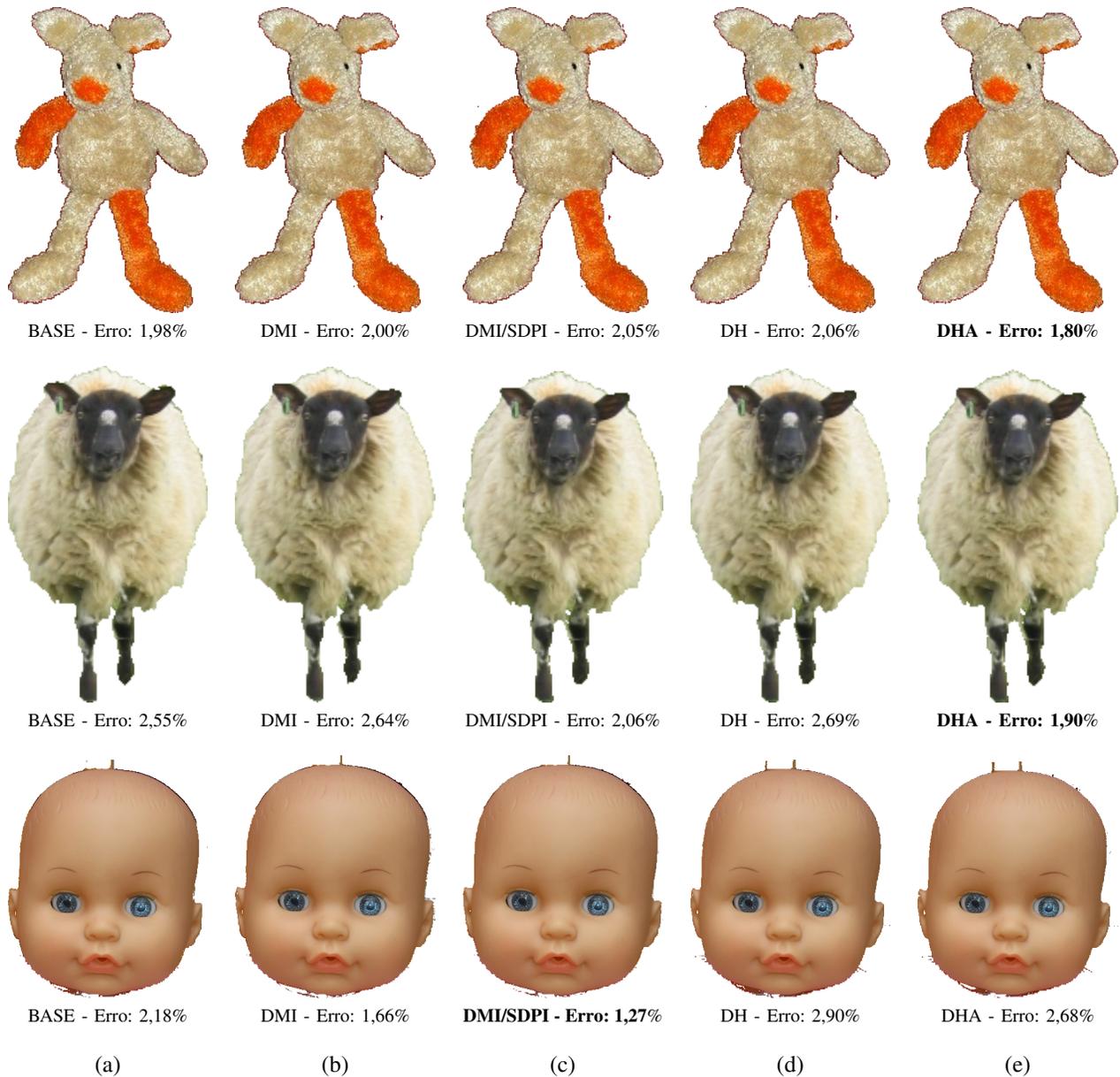


Figura 2. Resultados obtidos com o PCC usando (a) atributos com pesos idênticos (BASE), e atributos com pesos determinados pelos métodos (b) DMI (c) DMI/SDPI (d) DH (e) DHA

<http://doi.acm.org/10.1145/1015706.1015720>

[26] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: <http://archive.ics.uci.edu/ml>

[27] A. R. Smith, "Color gamut transform pairs," in *ACM Siggraph Computer Graphics*, vol. 12, no. 3. ACM, 1978, pp. 12–19.

[28] I. Jolliffe, *Principal component analysis*, 2nd ed. Wiley Online Library, 2002.