

Desenvolvimento de um Sistema de Visão para Inspeção Automática de Tecidos

Maurício Edgar Stivanello, Saulo Vargas
Departamento de Metal-Mecânica
IFSC - Instituto Federal de Santa Catarina
Florianópolis, Brasil
Email: {mauricio.stivanello, saulo.vargas}@ifsc.edu.br

Juliano Emir Nunes Masson
Departamento de Automação e Sistemas
UFSC - Universidade Federal de Santa Catarina
Blumenau, Brasil
Email: juliano.masson@grad.ufsc.br

Abstract—A number of defects can arise in the processes of spinning and weaving of textile industry. The identification of defective tissues in the early stages of the process reduces waste and discards. The objective of this work is to develop a system for automated inspection of fabrics. Statistical techniques based on intensity discontinuities on images of tissue are used to perform the detection of defective regions. The achieved results have shown the effectiveness of the employed approach for detecting different types of defects.

Keywords—automated inspection; computer vision; segmentation; textile

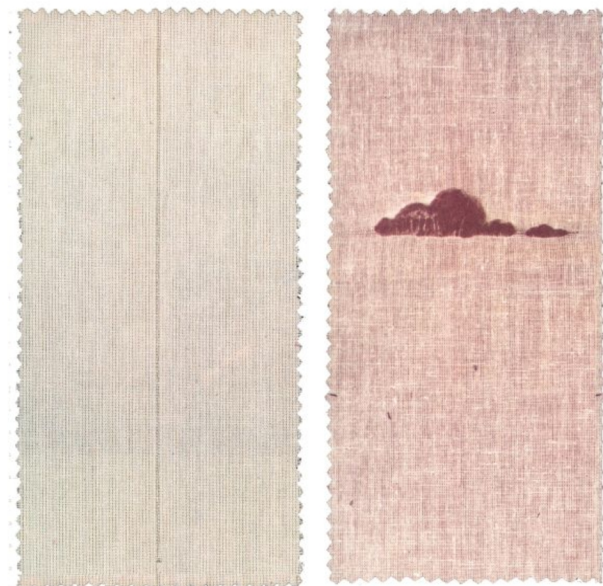
Resumo—Uma série de defeitos podem surgir nos processos de fiação e tecelagem da indústria têxtil. A identificação de tecidos defeituosos nas etapas iniciais do processo reduz desperdícios e devoluções. O objetivo do presente trabalho é desenvolver um sistema de inspeção automatizada de tecidos. Técnicas estatísticas de processamento baseadas na descontinuidade de intensidade das imagens dos tecidos são utilizadas para realizar a detecção de regiões defeituosas. Os resultados alcançados têm demonstrado a efetividade da abordagem proposta na identificação de diferentes tipos de defeito.

Keywords—inspeção automatizada; segmentação; tecido; visão computacional

I. INTRODUÇÃO

O cenário atual do mercado exige que a indústria têxtil aumente a eficiência, com linhas de produção mais ágeis, redução ao mínimo da criação de refugos e melhor qualidade de formação. Quando se considera o aumento do preço dos insumos, pequenos erros podem ser a fonte de grandes perdas na produção. Apesar de a qualidade ser essencial em qualquer tipo de economia, ela nunca foi tão importante quanto agora [1].

Na fabricação de tecidos, é essencial detectar os defeitos nos estágios iniciais do processo. Uma série de defeitos podem surgir nos processos de fiação e tecelagem, como os apresentados na Figura 1. Por exemplo, a detecção prévia de artefatos como *passamento errado de malha*, permite a classificação correta do tecido quanto a sua qualidade sem que um material defeituoso seja repassado aos processos de maior valor agregado, reduzindo assim os refugos e aumentando o rendimento. De modo geral, este tipo de inspeção reduz o volume de refugos, aumenta o rendimento e diminui devoluções.



a) Passamento errado no pente

b) Apelotado

Figura 1. Exemplos de defeitos na produção de tecidos

A automação de etapas da produção tem se mostrado como uma excelente ferramenta para aumentar a agilidade da produção, sem redução de qualidade. Para o caso de identificação de defeitos em tecidos, por exemplo, podemos fazer uso desta ferramenta para minimizar a necessidade de inspeções manuais.

No presente trabalho é apresentado o desenvolvimento de um sistema de inspeção automatizada de tecidos empregando visão computacional. Na seção II são apresentadas estratégias normalmente utilizadas em inspeção automatizada. Na seção III é apresentada a abordagem proposta, utilizada no sistema implementado. Os resultados obtidos são descritos na seção IV. Finalmente, as conclusões são apresentadas na seção V.

II. INSPEÇÃO AUTOMATIZADA DE TECIDOS

O procedimento tradicional de inspeção de tecidos ocorre retirando-se os rolos fabricados da máquina de tecelagem e

desenrolando-os sobre uma mesa de inspeção a uma velocidade relativamente alta. Quando o inspetor humano percebe um defeito no tecido, ele para o motor que move o rolo, registra o defeito e sua localização, e inicia novamente o motor. Esta tarefa se mostra extremamente repetitiva e propensa a erros.

A visão computacional tem sido empregada com sucesso na análise ou inspeção automática em diferentes cenários. Os métodos utilizados para a inspeção de tecidos variam de acordo com as características do tecido e são classificadas em três categorias, conforme a abordagem adotada para o reconhecimento do defeito: estatística, espectral e baseada em modelo [2].

A abordagem estatística pressupõe que as regiões do tecido livres de defeitos são de alguma forma estacionárias, o que torna possível localizar na imagem inspecionada as regiões com comportamento estatístico diferente. As abordagens dessa categoria se baseiam nos níveis de intensidades das imagens dos tecidos e empregam técnicas como estimativa de dimensão fractal [3], estatísticas de nível de cinza, correlação cruzada, detecção de borda, operações morfológicas [4], matriz de co-ocorrência e transformação linear local [5].

Imagens de tecidos com textura uniforme, compostos por características primitivas que possuem um padrão de repetição, são mais facilmente inspecionadas por abordagem espectral (que avalia o domínio da frequência), transformadas de Fourier [6], filtro de Gabor [7], distribuição de Wigner [8] e transformada Wavelet [9], são algumas das técnicas usadas para avaliar alterações espectrais desse tipo de imagem.

As abordagens baseadas em modelos são mais adequadas para imagens de tecidos texturizados aleatoriamente do que as abordagens estatísticas e espectrais. Nessa abordagem o tecido sem defeitos é modelado a partir de um processo estocástico, e o problema de detecção de defeitos passa a ser tratado como um problema de teste de hipóteses estatísticas do modelo. Modelo de Poisson [10] e campo aleatório de Markov Gauss [11] são exemplos de modelos empregados nessa abordagem.

III. METODOLOGIA PROPOSTA

No desenvolvimento do sistema apresentado, considerou-se como requisito principal a *identificação de defeitos de forma automática* na imagem do tecido inspecionado. Para atender a este requisito foi selecionado neste momento inicial de pesquisa a utilização de uma abordagem estatística. O fluxograma das etapas de processamento utilizadas é apresentado na Figura 2.

Na etapa de *aquisição* é realizada a captura de imagens digitais dos tecidos de interesse. Ainda nesta etapa, podem ser adicionados pré-processamentos, como a aplicação do filtro da mediana, por exemplo, para eliminar possíveis ruídos existentes.

Na etapa de *segmentação* é realizada a separação dos elementos de interesse da cena. Neste momento é necessário separar nas imagens a fase correspondente à textura original do tecido da fase correspondente aos defeitos presentes na amostra. Para isso, inicialmente é aplicada uma filtragem passa

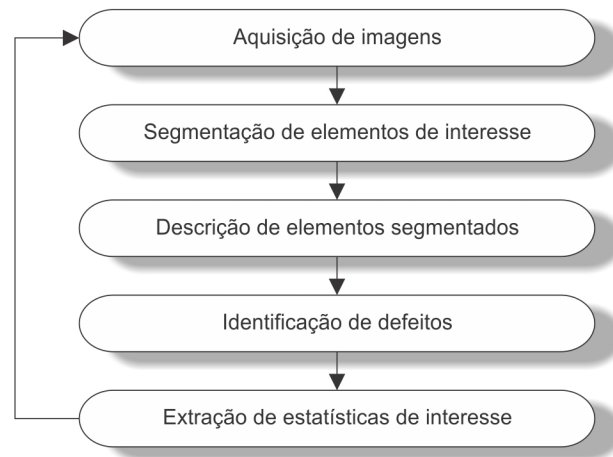


Figura 2. Fluxo de processamento proposto para identificação automática de defeitos

banda com o objetivo de remover a textura repetitiva obtida em função do entrelaçamento dos fios do tecido. Em seguida é empregada a técnica de limiarização, combinada com o método de Otsu [12] para a determinação automática do valor de limiar para cada amostra inspecionada. Como saída da etapa de segmentação é obtida uma imagem binária onde os defeitos encontram-se destacados do fundo. Entretanto, em função de ruído ou outros artefatos presentes nas imagens adquiridas, pode ser obtido um grande número de pequenos componentes que não representam defeitos e que devem ser eliminados. Para isso, é empregada uma sequência de operações morfológicas de erosão e dilatação, de modo que apenas componentes com um tamanho pré-definido sejam mantidos.

Na etapa de *descrição* cada um dos elementos restantes na imagem são detectados através de um algoritmo de rotulação de componentes conexos. Neste momento é gerada uma lista que contém a posição de cada elemento, além da área calculada com base no número de pontos pertencentes ao mesmo.

Por sua vez, na etapa de *identificação* os defeitos são encontrados de forma direta, avaliando-se a área de cada elemento e desconsiderando aqueles com valores fora de uma faixa aceitável. Finalmente, a partir das informações dos defeitos devidamente identificados e descritos é possível levantar estatísticas ou mesmo enviar comandos a sistemas atuadores integrados à linha de produção.

IV. RESULTADOS EXPERIMENTAIS

O sistema descrito foi implementado em linguagem C++, utilizando a biblioteca de processamento de imagens OpenCV. Para validar o protótipo desenvolvido foi realizada uma bateria de testes utilizando a base de imagens de tecidos apresentada na Figura 3, onde as amostras apresentam diferentes tipos de defeitos [13].

Na Figura 4 é ilustrado o processo de inspeção automatizada, partindo da imagem original até o resultado final com o

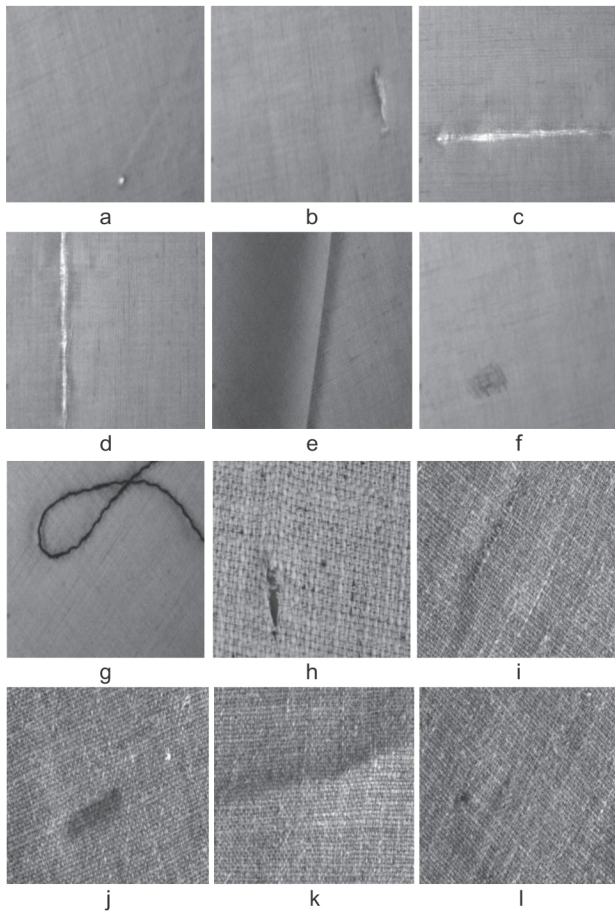


Figura 3. Defeitos avaliados

defeito identificado, para as amostras *b* e *d*. Os resultados obtidos com o algoritmo de identificação automática de defeitos que implementa o método descrito na seção III sobre a base de imagens de teste são apresentados na Tabela I.

Tabela I
RESULTADOS OBTIDOS NA INSPEÇÃO AUTOMATIZADA

Tipos de defeitos avaliados	Tipos de Defeitos identificados	Amostras com defeitos identificados	Amostras com defeitos não identificados
12	10	a,b,c,d,e,f,g,h,j,l	i,k

Os resultados obtidos são considerados promissores no que se refere à detecção automática, sendo que apenas dois dos defeitos avaliados não foram corretamente identificados. O algoritmo implementado é simples de ser utilizado. O valor de limiar utilizado na segmentação é obtido satisfatoriamente pelo método de OTSU, sendo que apenas a definição dos parâmetros do filtro passa banda, que é dependente da textura do tecido, se faz necessária.

Na Tabela II são apresentados os tempos médios necessários ao processamento das principais etapas envolvidas. Para o processamento de todo o ciclo de inspeção foi obtido um tempo

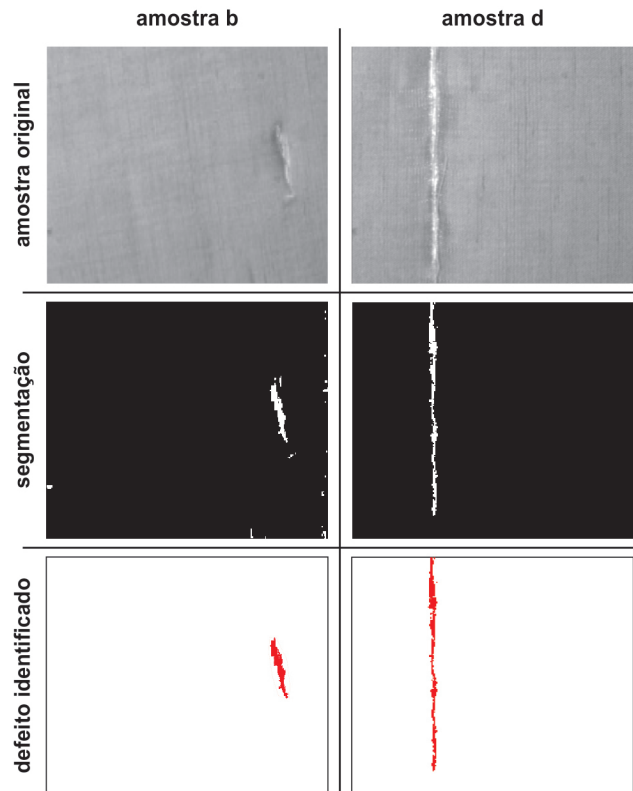


Figura 4. Ilustração do Processo de Identificação

médio de $10,46ms$ em um computador com processador INTEL Core i7-4500U de 1.80 GHz e 8GB de memória RAM, sobre amostras de 690 X 682 pixels. É importante observar que os algoritmos podem ainda ser otimizados empregando recursos específicos de hardware.

Tabela II
TEMPOS MÉDIOS DE PROCESSAMENTO

Etapa	Tempo (ms)
Aquisição e pré-processamento	8,24
Segmentação	0,10
Descrição, identificação e extração de dados	2,09
Fluxo geral	10,43

É importante avaliar os tempos de processamento obtidos frente ao domínio do problema. O sistema proposto será integrado a uma máquina revisora já em uso na indústria, conforme ilustrado na Figura 5, de modo a automatizar a etapa de inspeção de defeitos.

Este tipo de revisora opera a uma velocidade aproximada 8 metros por segundo. Se considerarmos a utilização de câmeras em uma configuração cujo campo de visão seja de 40 cm no sentido longitudinal do rolo de tecido sendo inspecionado, temos uma restrição que determina que o ciclo de inspeção deve operar a, pelo menos, 20 Hz. O tempo médio obtido atende a este requisito ao permitir operação de até 95 Hz. Os tempos obtidos permitem ainda a utilização de mais de uma

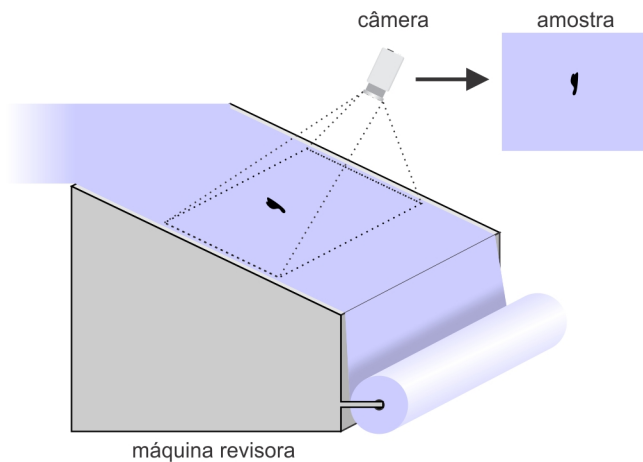


Figura 5. Ilustração da implantação em máquina revisora

câmera no sentido transversal do tecido inspecionado, de modo a cobrir rolos de tecido com larguras que excedam o campo de visão da câmera.

V. CONCLUSÕES E PERSPECTIVAS

No presente trabalho foi apresentado o desenvolvimento de um sistema de identificação automática de defeitos em tecidos por meio de visão computacional. O sistema corrente, que ainda se encontra em um estágio inicial de desenvolvimento, apresentou resultados satisfatórios na identificação de diferentes tipos de defeitos. Desta forma, sua utilização de forma integrada a linhas de produção para controle de qualidade mostra-se promissora.

Como perspectivas futuras, pretende-se aumentar a robustez do sistema e ampliar a gama de defeitos detectados através da combinação das técnicas estatística, espectral e baseada em modelo, conforme sugerido por Kumar [2]. A validação do sistema nas versões futuras será realizada sobre uma base de imagens maior do que a utilizada neste trabalho, empregando um conjunto de imagens que está sendo criado a partir de amostras de tecidos obtidos junto à indústria têxtil. Quando o sistema apresentar robustez e eficiência adequadas, pretende-se integrar o mesmo uma máquina revisora de tecidos, de modo a avaliar a sua efetividade em condições reais de uso.

AGRADECIMENTOS

Os autores gostariam de agradecer ao CNPq pelo suporte financeiro provido para a realização do projeto na qual este estudo foi realizado.

REFERÊNCIAS

- [1] L. C. Rodrigues, C. E. N. Bizzotto, and D. Souza, "Estratégias competitivas no setor têxtil na região de Blumenau," in *Anais do XIX Simpósio de Gest. da Inovação Tecnológica*, 1996.
- [2] A. Kumar, "Computer-vision-based fabric defect detection: A survey," *IEEE Transactions on Industrial Electronics*, vol. 55, pp. 348–363, 2008.
- [3] A. Conci and C. B. Proença, "A fractal image analysis systems for fabric inspection based on a box-counting method," *Comput. Netw. ISDN Syst.*, 1998.
- [4] Y. F. Zhang and R. R. Bresee, "Fabric defect detection and classification using image analysis," *Textile Research Journal*, vol. 65, pp. 1–9, January 1995.

- [5] C. Neubauer, "Segmentation of defects in textile fabric," in *Pattern Recognition, 1992. Vol. I. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on*, 1992.
- [6] R. N. Bracewell, *The Fourier transform and its applications*, 3, Ed. McGraw-Hill, 2000.
- [7] A. Bodnarova, M. Bennamoun, and S. J. Latham, "Textile flaw detection using optimal gabor filters," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, 2000.
- [8] K. Song, M. Petrou, and J. Kittler, "Texture crack detection," *Machine Vision and Applications*, 1995.
- [9] H. Sari-Sarraf and J. S. Goddard, "Vision system for on-loom fabric inspection," *Industry Applications, IEEE Transactions on*, 1999.
- [10] D. P. Brzakovic, P. R. Bakic, and A. Liakopoulos, "An approach to quality control of texture web materials," in *SPIE 2597*, 1995.
- [11] X. Z. Yang, G. K. H. Pang, and N. H. C. Yung, "Discriminative fabric defect detection using adaptive wavelets," *Optical Engineering*, 2002.
- [12] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, pp. 62–66, 1979.
- [13] P. Li, Y. Jiao, J. Jing, and J. Li, "The high-speed fabric defect detection algorithm based on the image layered model," *Journal of Fiber Bioengineering & Informatics*, vol. 6, pp. 161–173, 2013.

Gesture Recognition using shape characteristics

Igor L. O. Bastos
Math Institute
Federal University of Bahia
Salvador, Brazil
igorcrexito@gmail.com

Michele F. Angelo, Angelo C. Loula
Dept. of Technology, Dept. of Exact Sciences
State University of Feira de Santana
Feira de Santana, Brazil
mfangelo@uefs.ecomp.br, angelocl@gmail.com

Abstract —The present work brings up a system for recognizing gestures using two (2) shape descriptors, the histogram of oriented gradients (HoG) and Zernike invariant moments. These are used together to feed a two stage classifier composed of Artificial Neural Networks that perform the gesture classification. To train and test the classifier, 2208 images were used representing 12 signs from the Brazilian sign language (Libras). High accuracy rates were obtained, reaching in average 94% of accuracy for all 12 signs.

Index Terms — Histogram of oriented gradients, Zernike moments, Artificial Neural Networks.

I. INTRODUCTION

Gesture recognition pertains to recognizing meaningful expressions of motion by a human, involving hands, arms, face, head, and/or body [1]. Many techniques and strategies have been used on last years [2] ranging from works where digital image processing is exclusively used to systems that use computer vision and classification based on feature extraction.

Among the many fields where gesture recognition is applied, the field of signal languages is an important one. One of these languages is the Brazilian Sign Language (Libras), which is considered, since 2002, as an official Brazilian language [3].

In this context fits the present work. It describes the development of an approach for recognition of 12 signs from Libras. This approach takes as input images containing the hand signals and presents, at the end of the recognition process, the recognized sign as a response. Note that the signs from Libras present parameters, which include body language, movement and orientation [3]. However, for this work, only the hand configuration is used and signs are recognized statically.

II. RELATED WORK

Works applying computer vision techniques for gesture recognition have greatly increased in the last years [4]. Some of them are closely related to the present work.

The project proposed by Carneiro, Cortez and Costa [5] presents an approach for classifying signs from Libras using neural networks. However, the features used by Carneiro for classifying signs were Hu Invariant Moments. This project recognized 26 signals from Libras and showed accuracy rates ranging from 78% to 97%.

The project proposed by Rodriguez, Chavez and Menotti [6] used the classifier Support Vector Machine (SVM) to perform the classification. In this study, 24 Libras signs were recognized using the descriptors Hu invariant moments and Zernike invariant moments. These descriptors were used separately and the results obtained with them were compared. The Zernike descriptor showed better results, reaching, for some signs, 96% of accuracy rate.

Besides the existence of works related to Libras, others were found using similar approaches for recognizing signs from other languages as the one proposed by Bowden [7], which operates by recognizing signs of British Sign Language (BSL). It also uses a two stage classification process where shape features are extracted from hand images. This work also operates with non-static signs and to recognize them, motion features are also extracted and Hidden Markov Models (HMM) are used to perform the classification of these signs.

III. THEORETICAL FOUNDATION

In order to recognize the signs from the input images, shape-based features are used to extract information that allows the system to recognize the sign present from the input image. These feature descriptors are the Histogram of Oriented Gradients (HoG) and Zernike Invariant Moments.

A. Histogram of Oriented Gradients

Histograms of Oriented Gradients (HoG) are gradient based feature descriptors that are commonly used for object detection and recognition [8].

HoG's functionality is based on the principle that the shape of an object can be described by the distribution of intensity gradients and edge directions. To achieve a more precise representation of the shape of an object, the input (image) is divided in small regions called cells. For each cell the descriptor is computed and the final result consists on the output for all cells.

In addition, some parameters of HoG have to be adjusted. The size of cells and blocks (group of cells to perform gradient strength normalization), number of bins and the use or not of the orientation of gradients are some of these parameters.

B. Zernike Invariant Moments

Zernike moments are the mapping of an image onto a set of complex Zernike polynomials [9]. Since Zernike polynomials are orthogonal to each other, Zernike moments can represent the properties of an image with no redundancy or overlap of

information between the moments [9]. Because of these characteristics, Zernike moments have been used in pattern recognition applications [6, 10], like the present project.

One of the most important characteristic of the Zernike moments is the independence of their magnitude to the rotation. Hence, they can be used to describe shape characteristic of objects on images [9]. For the present project, where the input images can present variation in terms of rotation of the hands (Libras signs), these moments proved to be appropriated.

The Zernike moments are defined by two parameters: order and repetition. They differ on a rotated image from those of the original unrotated image in phase shifts, but no in magnitudes [11]. So, these magnitudes can be used as a rotation invariant feature.

IV. METHODOLOGY

The methodology used in this study started with the building of the training image dataset, followed by feature selection and data classification in two stages.

A. Composing the training dataset

The present work intends to classify gestures from the Brazilian Sign Language (Libras). As this work represents an intermediate stage in the development of a bigger project, which aims at recognizing about 50 distinct signs from Libras, it was decided that it would recognize 12 signs that represent letters of the Libras alphabet [3]. The selected gestures are the signs A, B, C, D, E, G, I, L, O, U, W and Y. They were selected because the HoG descriptor and the Zernike moments are shape-based features and these gestures have a very different shape one from another signs.

A database containing 1104 grayscale images was built for these 12 different Libras signs. These images have resolution of 50 by 50 pixels and contain the hand positions that represent the selected signs from Libras. All images were taken using only one person as model and for a better image acquisition, the light conditions were controlled. On Figure 1, some of these images can be seen.

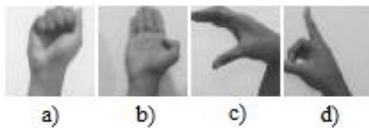


Fig. 1. Images representing Libras signs. a) letter A. b) letter B. c) letter C. d) letter D.

On the other hand, an auxiliary dataset was composed with another 1104 images. They are binary masks obtained with a skin detection approach [12] applied on the original images (RGB images before converting them to grayscale) and represent the skin zones of the 1104 images obtained before (Figure 2).



Fig. 2. Binary masks representing the signs. a) letter A. b) letter B. c) letter C. d) letter D.

To grant to the classifier the ability to recognize gestures that have variations on the positions, the dataset was built with images presenting variations in rotation, and variations resulting from the acquisition process and skin recognition (Figure 3).



Fig. 3. Images of dataset showing rotational variance and acquisition variance

B. Extracting features from the images

To extract relevant information from the images, the HoG descriptor and the Zernike moments were applied in order to generate a feature vector. The HoG descriptor, to be associated to the edges (gradients) of the image, was applied directly on the grayscale images. On the other hand, to apply the Zernike moments, that is associated to the shape of objects on images, it was necessary to use the binary masks over the grayscale images, applying the Zernike descriptor on the resulting images.

C. Selecting features

As it was mentioned before, the histogram of oriented gradients (HoG) and the Zernike invariant moments were used to compose a feature vector used to feed the classifier because of their shape-based character. To select the parameters of HoG, as number of bins and size of blocks and cells; and the order and repetition of Zernike moments that would be used, tests were performed and the accuracy of the classifier was measured, considering the variation of these parameters. The selected parameters of both descriptors were those for which the classifier had the highest accuracy rates.

Table 1 shows the selected parameters for both descriptors.

TABLE I. SELECTED FEATURES

Zernike moments (order, repetition)	HoG parameters (bins, cell, block, oriented)
(10,3), (9,2), (8,3), (7,2), (6,3), (5,2), (4,3), (3, 2), (2, 2)	9, 4, 8, not oriented

D. Assembling the groups

As mentioned before, 12 signs from the Libras alphabet were chosen to be recognized. Those gestures were grouped into 3 small groups (each with 4 gestures).

Signs were split in groups to minimize difficulties in classification of a wide range of patterns, dividing the classification task into smaller classifications. In addition, to promote the classifier the ability to distinguish so many

different patterns, it would probably be necessary to have a large number of neurons in its hidden layers. This could cause problems such as an increase in computational cost / time for training and classification, as well as other possible problems, such as excessive adaptation to the trained patterns (*overfitting*).

So, it was decided that the classification would be performed in 2 stages: 1) Recognition of the group which the input image belongs. 2) Recognition of the sign of the input image.

Deciding the signs that would compose each group was the first task to be done. As the HoG and Zernike moments are shape descriptors, we chose to put similar signs (those that present conformable shape) in the same group. The visual inspection was the only criteria to do this grouping. Thus, we have reached: Group 1: “A”, “B”, “E” and “G”; Group 2: “C”, “D”, “O” and “W”; Group 3: “I”, “L”, “U” and “Y”.

Images from each group can be seen on Figure 4.

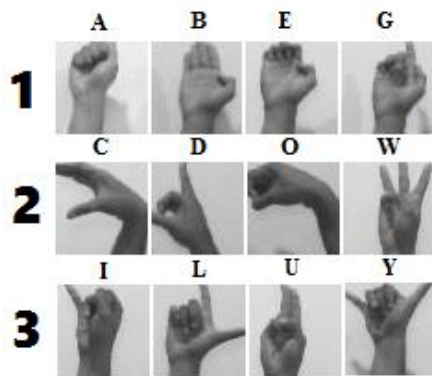


Fig. 4. Signs of each group.

E. Architecture of the classifier

The architecture of the neural network classifier used is shown on Figure 5.

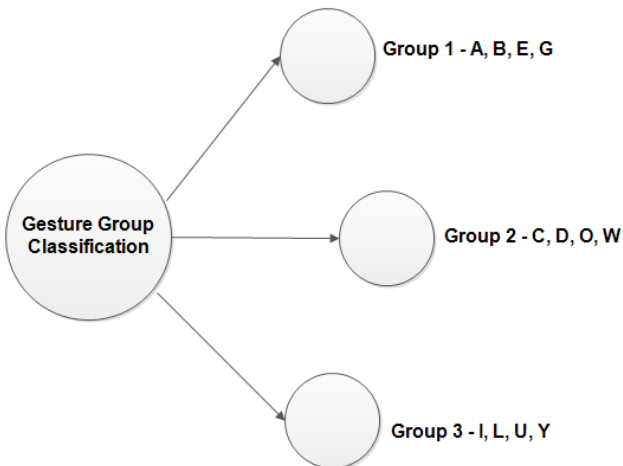


Fig. 5. Architecture of the used classifier.

The recognition of the input gesture is performed in two stages. The first one corresponds to a group classification, which is conducted by a neural network trained with images of

all 12 gestures. The goal of this network is to recognize the group that the input belongs and lead the classification to a next neural network. The output of the general classification determines which subsequent network should be activated. As shown in Figure 5, these subsequent networks are trained to classify individual gestures of a single group and the output of them is the sign to which the input corresponds.

Tests were also performed in order to decide the number of hidden neurons. For the first network (general classification), 45 hidden neurons were used, once it provided a better performance. For the ones responsible for groups 1 and 3, 25 hidden neurons were used. The one responsible for group 2, presented better results using 18 hidden neurons. The results showed on next section were done using these configurations.

V. RESULTS AND DISCUSSION

In order to validate the proposed system, a testing dataset was generated. It was modeled in the same way as the training dataset, which had grayscale images and binary masks. The test images, as well as training, have a resolution of 50 by 50 pixels and this dataset was composed by 2400 images.

The first part to be tested on the present system was the first network (group classification). This network was tested with images from all 12 trained Libras signs and the response was checked to see if it matches the correct group. The results obtained with this classification are shown on Figure 6, where the average accuracy rate for each group can be seen.

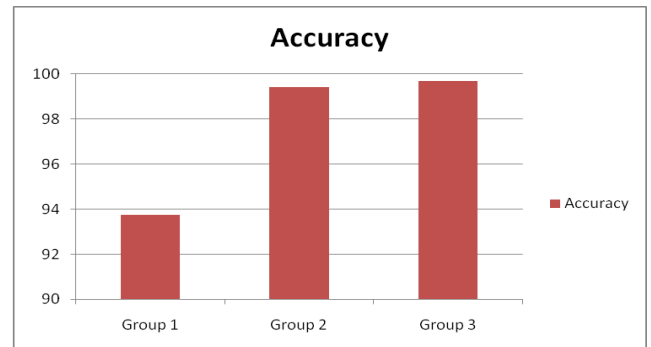


Fig. 6. Average group recognition accuracy rate for each group.

The accuracy rate for all groups was above 93%. On Figure 7, the performance of the group classifier can be seen for each sign.

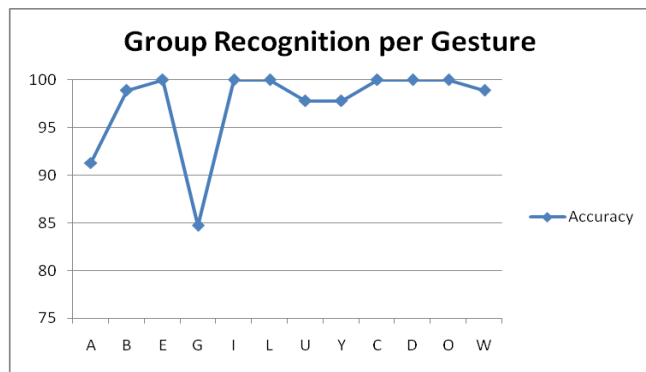


Fig. 7. Average group recognition rates

As shown on Figure 7, the recognition rates for all signs were above 85% of accuracy. It is also noticeable that, for signs "A" and "G", the rates were lower than for the others. That promoted the performance drop for this first group compared to the others.

Lastly, the accuracy of the two stage classification of each gesture was determined and is shown on Figure 8.

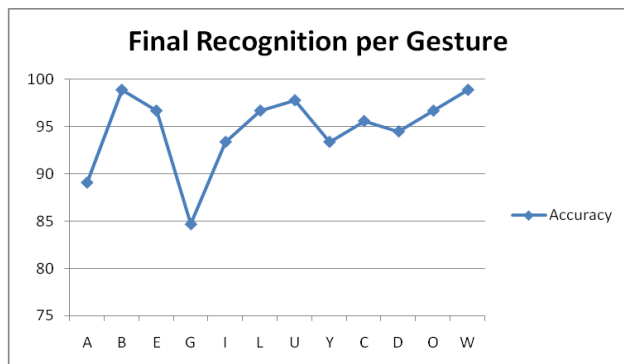


Fig. 8. Final recognition for each gesture

Figures 6, 7 and 8 evidenced that the system developed obtained high accuracy rates. For all gestures, the recognition rate was above 84%. Final (two stage) gesture recognition rates for each sign were very similar to group recognition ones. This evidences that the individual gesture recognition performed well for gestures corrected classified at the first stage.

Gestures "A" and "G" showed the lowest recognition rates. Both cases can be associated to the similarity of these gestures with gestures of other groups, what was pointed by the group classifier performance shown on Figure 7. That evidences that, if only one network was used, the system could have problems to distinguish all the 12 gestures.

VI. CONCLUSION

Despite the not so high rates for the recognition of the "A" and "G", the system presented good results, with an average recognition rate of about 94%. Some changes on the group organization (change the gestures of each group) could result in better results.

As future work, we hope to extend the system to recognize more signs from Libras (about 50), besides seeking a better strategy to distribute the gestures on groups. It would propitiate a gain in terms of accuracy for signs that the results were not as high, as "A" and "G".

REFERENCES

- [1] S. Mitra and T. Acharya, "Gesture Recognition: A Survey," *IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews*, vol. 37, n. 3, May 2007.
- [2] V. Pavlovic, R. Sharma and T. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, n. 7, July 1997.
- [3] T. Felipe, *A Libras em Contexto*, 8th ed., Walprint Gráfica e Editora, 2007.
- [4] R. Khan and N. Ibraheem, "Survey on Gesture Recognition for Hand Image Postures," *Computer and Information Science Journal*, vol. 5, n. 3, May 2012.
- [5] A. Carneiro, P. Cortez and R. Costa, "Reconhecimento de gestos da Libras com classificadores neurais a partir dos momentos invariantes de Hu," *Interaction South America*, 2010.
- [6] K. Rodríguez, G. Chávez and D. Menotti, "Hu and Zernike Moments for Sign Language Recognition," *International Conference on Image Processing, Computer Vision and Image Recognition*, 2012.
- [7] R. Bowden, D. Windridge, T. Kadir, A. Zisserman and M. Brady, "A Linguistic Feature Vector for the Visual Interpretation of Sign Language," *European Conference of Computer Vision*, 2004.
- [8] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886-893, 2005.
- [9] S. Hwang and W. Kim, "A novel approach to the fast computation of Zernike moments," *Journal of Pattern Recognition*, vol. 39, pp. 2065-2076, March 2006.
- [10] A. Khotanzad and Y. Hong, "Invariant Image Recognition by Zernike Moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, n. 5, May 1990.
- [11] H. Hse and A. Newton, "Sketched Symbol Recognition using Zernike Moments," *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 1, 2004.
- [12] C. Prema and D. Manimegalai, "Survey on Skin Tone Detection using Color Spaces," *International Journal of Applied Information Systems*, vol. 2, n. 2, May 2012.

Segmentação e Classificação de Embarcações em Eclusas

Fagner Pimentel*, Michele Fúlvio Angelo[†], Diego Gervasio Frías[‡]

* Universidade Federal da Bahia - UFBA

Email: fagnerpimentel@gmail.com

[†] Universidade Estadual de Feira de Santana - UEFS

Email: mfangelo@ecomp.uefs.br

[‡] Universidade do Estado da Bahia - UNEB

Email: diegofriass@gmail.com

Abstract—This paper aims to identify and classify ships in river environments using and comparing techniques for segmentation like Background Subtraction and classifiers like Support Vector Machine. This study is expected to define a set of techniques that best fit the segmentation and classification of ships in river environments in order to optimize the flow of ships at locks located on the Tietê river as future application.

Keywords—Segmentation; Classification; Ship; Fluvial Environment;

Resumo—Este trabalho tem como objetivo identificar e classificar embarcações em ambientes fluviais usando e comparando técnicas de segmentação como subtração de background e classificadores como Support Vector Machine. Este estudo deverá definir um conjunto de técnicas que melhor se adequam a segmentação e classificação de embarcações em ambientes fluviais, a fim de otimizar o fluxo de embarcações em eclusas localizadas no rio Tietê como aplicação futura.

Palavras-chave—Segmentação; Classificação; Embarcação; Ambiente fluvial;

I. INTRODUÇÃO

Uma eclusa é uma obra de engenharia hidráulica que permite uma embarcação vencer o desnível de uma barragem, quedas de água ou corredeiras no leito do curso d'água [1], [2]. O processo de eclusagem é iniciada quando uma embarcação chega ao ponto de parada obrigatória (PPO) da jusante (ponto mais baixo do rio) ou montante (ponto mais alto do rio).

A hidrovia Tietê-Paraná no estado de São Paulo, conta com seis eclusas: uma em Barra Bonita, a primeira usina na cascata do rio Tietê, com eclusagem voltada principalmente para o turismo; uma em Bariri com eclusagem voltada para o transporte de cana de açúcar; uma em Ibitinga voltada para o turismo e para carga; uma em Promissão; e por fim, duas em Nova Avanhandava por possuir um canal de eclusagem muito longo. Nesta última existe apenas um operador para as duas eclusas com controle centralizado. As usinas, eclusas e barragens de Barra Bonita, Bariri, Ibitinga, Promissão e Nova Avanhandava são mantidas e operadas pela concessionária AES/Tietê [3].

As passagens nas eclusas são realizadas 24 horas por dia, ininterruptamente, salvo em casos de emergência, relacionados à segurança ou por solicitação da operadora das eclusas. O tempo para realizar a eclusagem nas eclusas do rio Tietê

varia entre 20 e 45 minutos, dependendo fundamentalmente das vazões de enchimento e drenagem pelas comportas e do nível do reservatório. O direito de passagem na eclusa para as embarcações que estejam aguardando, é concedido pelo operador da eclusa considerando a seguinte ordem de prioridade: (1) Embarcações da Marinha do Brasil, Órgãos de fiscalização federal, estadual e municipal em embarcações oficiais; (2) Embarcação comercial de passageiros (turismo); (3) Embarcação destinada à execução de trabalhos de manutenção na hidrovia; (4) Embarcação transportando mercadorias perecíveis ou susceptíveis de perdas na qualidade final do produto; (5) Embarcação de lazer (esporte e recreio), sendo as embarcações de turismo e mercadoria as que trafegam com mais frequência.

Na ausência de embarcações que preencham os requisitos acima, é considerada como prioridade a ordem de chegada da embarcação nos PPO's (jusante e montante) da eclusa [1]. Após finalizar uma eclusagem o operador envia os dados (tipo de embarcação e horário previsto de chegada) no sentido montante ou jusante, para que o operador da próxima eclusa esteja preparado.

Conforme descrito, a identificação das embarcações é feita pelo operador da eclusa, e o controle de prioridade para entrar na eclusa se dá de forma manual, sem qualquer tipo de otimização, possibilitando maior ocorrência de erros no controle e gastos desnecessários, como por exemplo, no tempo de espera das embarcações e vertimento da água utilizada. Com base neste contexto, a proposta deste trabalho é desenvolver um sistema de segmentação e classificação de embarcações em ambientes fluviais a partir de imagens extraídas do sistema de vigilância da concessionária AES/Tietê que utilizam câmeras do tipo Pan-Tilt-Zoom, capazes de realizar movimentos na vertical (Pan), horizontal (Tilt) e de aproximação (Zoom).

Os processos de segmentação e classificação das embarcações tornarão possível, futuramente, o desenvolvimento de uma aplicação de monitoramento e controle automático das eclusas durante as operações de eclusagem, possibilitando uma menor ocorrência de erros no controle e menos gastos desnecessários, como por exemplo, no tempo de espera das embarcações e vertimento da água utilizada.

II. TRABALHOS RELACIONADOS

Em [4], no contexto de detecção de embarcações, o classificador SVM (do inglês, Support Vector Machine) é utilizado para reconhecer primeiramente regiões que representam água, para que em seguida seja feita a busca por embarcações nesta região. A abordagem é baseada nas seguintes observações no cenário de vigilância: as embarcações só podem viajar dentro da região de água, e o movimento das embarcações é mais relevante do que o movimento das regiões de água. Partindo do pressuposto que as embarcações apenas trafegam dentro da região de água, a detecção de embarcações pode ser facilitada se a região de água for extraída e fornecida como informação contextual.

Em [5] é aplicada a subtração de background juntamente com um duplo threshold (hysteresis) para a segmentação de embarcações. Como acontece normalmente em imagens com background dinâmico, não somente o movimento das embarcações como também da água e das árvores são captados pela subtração de background. A ideia do threshold com hysteresis é explorar a coerência espacial do objeto na imagem resultante da subtração. Nesta abordagem, primeiramente, é passado um threshold alto, determinando um subconjunto de pixels do objeto, em seguida um threshold baixo contendo um subconjunto maior de pixels. O resultado final é dado pelos pixels de baixo threshold que se interligam recursivamente aos pixels de alto threshold.

Em [6] é utilizada a subtração de background para o rastreamento de embarcações em ambientes marítimos, onde a maior parte do background é formado pelo oceano. A principal dificuldade neste trabalho está no fato do oceano ser dinâmico e os objetos terem que ser extraídos em um ambiente altamente imprevisível. Neste trabalho foi desenvolvido um algoritmo próprio de subtração de background além de utilizar a título de comparação, algoritmos do tipo Mistura de Gaussiana [7] e Sigma-Delta[8].

Em [9] foi utilizada uma RNA (Rede Neural Artificial) para a classificação de embarcações com 12 neurônios de entrada, 20 neurônios na camada escondida e 5 na camada de saída. Foram utilizadas 240 imagens para o treinamento, variando os ângulos de visão das embarcações. Ao final do treinamento foi obtida uma taxa de acerto de 90.1%. Em seguida foram realizados testes com 41.400 imagens, obtendo uma taxa de acerto de 87.3%.

Em [10] foi utilizado um classificador SVM para classificar embarcações a partir de câmeras Pan-Tilt-Zoom e HOG (do inglês Histogram of Oriented Gradients) como extrator de características. Para o treinamento do classificador foram utilizadas 280 imagens positivas e exemplos do background extraído como imagens negativas, obtendo uma taxa de acerto de 85%.

III. METODOLOGIA

O projeto tem como objetivo segmentar e classificar embarcações em rio através de imagens de câmeras pan-tilt-zoom (Figura 1), e o seu desenvolvimento será dividido nas seguintes etapas:

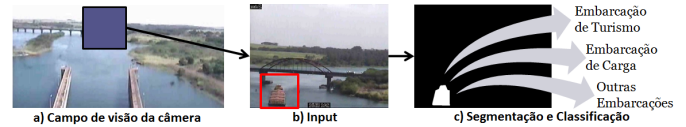


Figura 1. Visão Geral do projeto: A partir do (a) campo de visão da câmera a embarcação é colocada em foco usando pan-tilt-zoom, gerando as imagens que servem de (b) input para o sistema, em seguida é realizada a (c) Segmentação da embarcação e classificação da segmentação em embarcação de turismo ou carga.

A. Criação do Dataset

Serão criados datasets com marcações em binary mask para a fase de avaliação dos algoritmos de subtração de background e para o treinamento e testes dos classificadores. Estes datasets serão criados com imagens positivas (exemplos de embarcações) e negativas (background) das esclusas ao longo do rio Tietê.

Para a criação destes datasets serão selecionados vídeos fornecidos pela concessionária AES/Tietê que possuam exemplos das embarcações que trafegam pelas esclusas em diferentes momentos e com diferentes luminosidades, ângulos e distâncias.

Os datasets de binarymask serão criados com a ajuda de uma ferramenta de anotação semi-automática desenvolvida para este propósito. Com esta ferramenta será possível criar o dataset a partir de um vídeo selecionado.

B. Segmentação

Nesta etapa será realizado um pré-processamento das imagens visando identificar as regiões de água, servindo de informação de contexto para facilitar a segmentação da embarcação.

Em seguida será realizado um estudo de performance de 27 algoritmos de subtração de background implementados na biblioteca bgslibrary [11] com o objetivo de escolher o algoritmo que apresente bons resultados para as imagens que serão utilizadas neste projeto.

O algoritmo de background que apresentar o melhor resultado será adaptado para utilizar a técnica de threshold com hysteresis observado em [5].

C. Extração de Características

Segmentada a embarcação, o próximo passo é extrair as características que a descrevam. A principal característica que será utilizada para a classificação de embarcação será a forma. Segundo Luo *et al.* 2006 [5], a forma é a característica mais importante para a detecção de embarcações.

Um método que será utilizado para a extração de características de embarcações será o histograma de gradientes orientados (HOG) [10]. A ideia deste método é que a forma dos objetos pode ser caracterizada pela direção e intensidade dos gradientes em uma imagem.

Esse método é implementado através da divisão da imagem em pequenas regiões (células). As células são divididas em blocos, e para cada bloco é computado o histograma de gradientes orientados sobre os pixels das células [12].

D. Classificação

Nesta etapa será utilizado o classificador SVM (Support Vector Machine) como foi usado em [12], [10], previamente treinado. A priori serão classificadas as embarcações voltadas para o turismo e para o transporte de carga por serem as que trafegam com mais frequência nas eclusas. Após a implementação das técnicas e treinamento dos classificadores, serão realizados os testes com as imagens e vídeos separados para esta finalidade.

IV. RESULTADOS PRELIMINARES

Até o presente momento a etapa A (Criação do Dataset) foi finalizada parcialmente e a etapa B (Segmentação) foi finalizada. As etapas C (Extração de Características) e D (Classificação) estão em desenvolvimento. A seguir os resultados preliminares serão apresentados.

A. Criação do Dataset

Foi construído um dataset a partir de imagens capturadas da eclusa de Nova Avanhandava. O dataset é composto de 5 vídeos de embarcações se aproximando da eclusa com uma média de 1280 frames por vídeo. Cada vídeo contém uma embarcação em um ambiente com carros, árvores, nuvens e água em movimento e variação de luminosidade ao longo do dia. Nestes vídeos foram realizadas as anotações de ground-truth em binarymask com o auxílio de uma ferramenta de anotação semi-automática desenvolvida para facilitar e acelerar este processo.

Esta ferramenta de marcação, primeiramente, cria os frames com uma pré-anotação feita automaticamente utilizando um simples algoritmo de subtração de background. Em seguida, cada frame é editado manualmente fazendo as correções necessárias sobre a pré-anotação. Para facilitar e acelerar o processo, a ferramenta conta com uma interface que auxilia a mudança rápida entre os frames, auxilia a verificação da anotação sobrepondo a imagem original e possui regulagem do marcador, além da utilização de uma mesa digitalizadora.

Este dataset foi utilizado para a avaliação dos algoritmos de background e para a realização do treinamento e validação do classificador de água.

B. Segmentação

Esta etapa foi dividida em duas sub-etapas, uma responsável pela identificação da água do rio (pré-processamento) e outra para a avaliação dos algoritmos de background e adaptação do melhor algoritmo encontrado.

1) *Identificação da água do rio:* Nesta etapa, primeiramente, foram recortados segmentos de imagens com exemplos positivos e negativos de água a partir de cada vídeo do dataset das eclusas. Com isso foi possível criar cinco conjuntos de treinamento e teste, onde cada conjunto de treinamento utiliza imagens de quatro vídeos e cada conjunto de teste utiliza imagens do vídeo restante, tornando os conjuntos mutualmente exclusivos.

Para a extração das características da água do rio, foi realizada em cada imagem dos conjuntos de treinamento e

Tabela I
RESULTADO DOS TESTES DE CLASSIFICAÇÃO DE ÁGUA.

	linear	polynomial	base radial	sigmoid
Data 1	98.0%	98.0%	97.6%	66.6%
Data 2	99.6%	99.6%	64.6%	66.6%
Data 3	100.0%	100.0%	100.0%	66.6%
Data 4	100.0%	98.3%	98.3%	66.6%
Data 5	94.6%	93.0%	92.0%	66.6%

teste a transformação do espaço de cor RGB para LAB e utilizadas as características A e B, as quais representam as cores, e ignorada a característica de luminosidade L.

O classificador utilizado foi o SVM e os resultados obtidos, variando-se o kernel em linear, polynomial, de base radial e sigmoid são mostrados na Tabela I. Através desta tabela pode-se verificar que o SVM com kernel linear obteve os melhores resultados, com uma média de acerto de 98.4% e por isso escolhido neste trabalho.

Por fim, foi realizada a segmentação de regiões similares no frame inicial de cada vídeo, utilizando um algoritmo de segmentação de regiões proposto em [13]. Em seguida cada região é classificada como água ou não água utilizando-se o classificador SVM com kernel linear, treinado previamente a partir de características de cor do espaço de cor LAB.

O pré-processamento para a identificação de água é realizado apenas no primeiro frame do vídeo, partindo da hipótese que a região de água não será alterada nos frames seguintes.

2) *Avaliação dos algoritmos de subtração de background:* Dado que para a realização da subtração de background é necessária uma câmera estática, esta fase é realizada apenas nos momentos em que a câmera está parada.

Tendo em vista a eficiência do threshold com hysteresis observado em [5], a avaliação dos algoritmos de subtração de background foi realizada visando a aplicação desta técnica no algoritmo selecionado. O melhor algoritmo foi encontrado através da média harmônica (HM) da taxa de verdadeiros positivos (TPR) e a taxa de verdadeiros negativos (TNR). O TPR é definido por $TP / (TP + FN)$, enquanto o TNR é definido por $TN / (FP + TN)$ e o HM é $2 * TPR * TNR / (TPR + TNR)$, onde TP é número de verdadeiros positivos, TN é o número de verdadeiros negativos, FN é o número de falsos negativos, e FP é o número de falsos positivos.

A partir dos dados extraídos das métricas de avaliação dos algoritmos, foram feitas comparações a fim de determinar qual algoritmo melhor se aplica ao projeto. Para cada um dos vídeos presentes no dataset foram feitas 27 avaliações referentes a cada algoritmo da biblioteca bgslibrary [11], utilizando a média harmônica da taxa de verdadeiros positivos e a taxa de verdadeiros negativos apresentada anteriormente. Dada a grande quantidade de algoritmos, os parâmetros dos mesmos foram mantidos como padrão da biblioteca bgslibrary.

Dentre os algoritmos testados, como pode ser visto na tabela II, o algoritmo DPEigenbackgroundBGS apresentado em [14] foi o que apresentou os melhores resultados, com HM de 0.95, sendo então o escolhido para a realização da segmentação de

Tabela II
MÉDIA HARMÔNICA (HM) DA TAXA DE VERDADEIROS POSITIVOS E A
TAXA DE VERDADEIROS NEGATIVOS PARA CADA ALGORITMO.

algoritmo	HM
AdaptiveBackgroundLearning	0.7124
DPAdaptiveMedianBGS	0.4712
DPEigenbackgroundBGS	0.9528
DPGrimsonGMMBGS	0.7078
DPMeanBGS	0.2112
DPPratiMediodBGS	0.5903
DPTexureBGS	0.8430
DPWrenGABGS	0.6530
DPZivkovicAGMMBGS	0.7853
FrameDifferenceBGS	0.3663
FuzzyChoquetIntegral	0.4131
FuzzySugenoIntegral	0.3797
GMG	0.8025
KDE	0.8120
LbpMrf	0.8253
MixtureOfGaussianV1BGS	0.2342
MixtureOfGaussianV2BGS	0.3457
MultiLayerBGS	0.3514
PixelBasedAdaptiveSegmenter	0.2889
StaticFrameDifferenceBGS	0.9167
T2FGMM_UM	0.1812
T2FGMM_UV	0.6851
T2FMRF_UM	0.1694
T2FMRF_UV	0.4589
VuMeter	0.2902
WeightedMovingMeanBGS	0.2595
WeightedMovingVarianceBGS	0.2890

embarcações em ambientes fluviais neste trabalho.

Sobre o algoritmo DPEigenbackgroundBGS foi aplicado o hysteresis. Na Figura 2 é possível observar os resultados para o reconhecimento de água (Figura 2b), o threshold alto (Figura 2c), o threshold baixo (Figura 2d), o resultado da aplicação do hysteresis (Figura 2e) sobre um frame de um dos vídeos do dataset, e o resultado final da segmentação (Figura 2f).

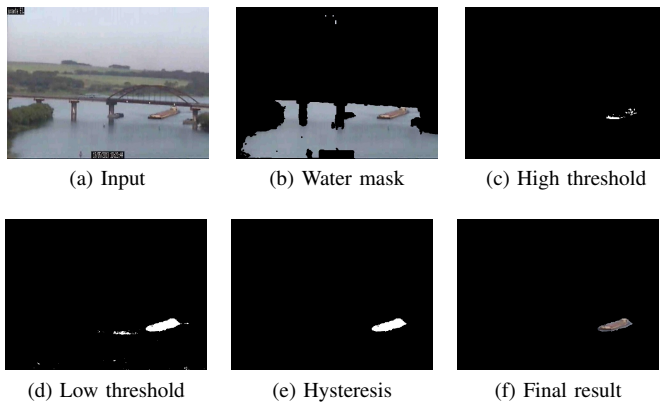


Figura 2. Hysteresis aplicado ao algoritmo DPEigenbackgroundBGS

V. CONCLUSÃO

Neste artigo foram apresentadas as etapas necessárias para a segmentação e classificação de embarcações em ambientes fluviais, bem como quais já foram concluídas parcialmente

ou completamente, e aquelas que ainda estão em desenvolvimento.

A etapa de segmentação, que é composta pela identificação da água do rio e pela avaliação dos algoritmos de background, apresentou como resultado: o classificador de água obteve uma média de acerto de 98.4% e o algoritmo de background mais indicado para a segmentação de embarcação foi o DPEigenbackgroundBGS, uma vez que através da avaliação dos algoritmos da biblioteca bgslibrary [11], este foi o que apresentou a maior média harmônica (0.95).

Com esses resultados, as etapas de extração de características e classificação de embarcações já começaram a ser desenvolvidas e a expansão do dataset (aumento do número de vídeos) também já está em andamento.

AGRADECIMENTOS

Os autores agradecem à FAPESB pelo apoio financeiro, à Allan Obrecht e Haroldo Silva da AES/Tietê pelo fornecimento dos vídeos e à Franklin Oliveira e Beatriz de Brito pelas anotações de groundtruth.

REFERÊNCIAS

- [1] (2014, Jun.) Normas de tráfego nas eclusas da hidrovía tietê-paraná e seus canais. [Online]. Available: www.ahrana.gov.br/down.php?download=22
- [2] (2014, Jun.) Diretrizes da política nacional de transporte hidroviário. [Online]. Available: www2.transportes.gov.br/Modal/Hidroviario/PNHidroviario.pdf
- [3] AES, "AES/tietê," www.aesbrasil.com.br. Acesso em: 13 de julho de 2013. [Online]. Available: <http://www.aesbrasil.com.br>
- [4] X. Bao, S. Zinger, R. Wijnhoven *et al.*, "Ship detection in port surveillance based on context and motion saliency analysis," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2013, pp. 86 630D–86 630D.
- [5] Q. Luo, T. Khoshgofaar, and A. Folleco, "Classification of Ships in Surveillance Video," in *2006 IEEE International Conference on Information Reuse & Integration*. IEEE, Sep. 2006, pp. 432–437. [Online]. Available: <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=4018530>
- [6] Z. L. Szpak and J. R. Tapamo, "Maritime surveillance: Tracking ships inside a dynamic background using a fast level-set," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6669–6680, Jun. 2011.
- [7] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, vol. 2. IEEE Comput. Soc, 1999, pp. 246–252.
- [8] S. Toral, M. Vargas, F. Barrero, and M. Ortega, "Improved sigma-delta background estimation for vehicle detection," *Electronics Letters*, vol. 45, no. 1, p. 32, Jan. 2009.
- [9] J. Alves, J. Herman, and N. C. Rowe, "Robust Recognition of Ship Types from an Infrared Silhouette," Jun. 2004. [Online]. Available: <http://calhoun.nps.edu/public/handle/10945/36813>
- [10] R. Wijnhoven, K. van Rens, E. Jaspers, and P. de With, "Online Learning for Ship Detection in Maritime Surveillance," pp. 73 – 80, 2010.
- [11] A. Sobral, "{BGSLibrary}: An OpenCV C++ Background Subtraction Library," in *IX Workshop de Visão Computacional (WVC'2013)*, Rio de Janeiro, Brazil, Jun. 2013.
- [12] N. Dalal and W. Triggs, "Histograms of Oriented Gradients for Human Detection," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR05*, vol. 1, no. 3, pp. 886–893, 2004.
- [13] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.
- [14] N. M. Oliver, B. Rosario, and A. P. Pentland, "A bayesian computer vision system for modeling human interactions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 831–843, 2000.

Multiscale Approach for Cluster Estimation and Image Segmentation

Alan M. Braga[†], Fátima N. S. Medeiros* and Regis C. P. Marques[†].

[†]Departamento de Telemática

Instituto Federal do Ceará, Fortaleza, Brazil

Email: regismarques@ifce.edu.br, alanmbraga@gmail.com

*Departamento de Engenharia de Teleinformática

Universidade Federal do Ceará, Fortaleza, Brazil

Email: fsombra@ufc.br

Abstract—Data clustering and related applications usually require the number of clusters which can be available or estimated. In the absence of such a priori information, an appropriate procedure is required to estimate it. Thus, this paper introduces a new approach for cluster estimation and image segmentation based on clustering. Our proposal consists of a scale-space technique that encompasses an undecimated wavelet decomposition to detect clusters in a multiscale analysis of image histogram. The results showed that the proposed technique improved classical image segmentation methods such as thresholding, K-means and level set.

Keywords—Multiscale histogram, cluster estimation, histogram filtering, image clustering.

I. INTRODUCTION

Clustering data plays a central role in digital image processing, more precisely in image segmentation applications. Moreover, several shape analysis and pattern recognition applications are highly dependent on robust segmentation methods, which definitely configure the success of posterior tasks. Clustering algorithms partition sample data into a predefined number of clusters. Actually, algorithms based on K-means [1] and level sets [2] for image segmentation may also require initial clusters which effectively influence the quality of results. However, in most cases the number of clusters is not known a priori and, furthermore, it configures a challengeable problem.

A current approach to solve this problem is the histogram analysis. In fact, image segmentation can be approached by clustering techniques. Several techniques extract histogram information for cluster estimation and to partition images into meaningful regions. In [3], algorithms for histogram specification are investigated in order to modify images for segmentation purpose. According to [4], the best number of clusters is determined by the number of gray histogram peaks. Nevertheless, data filtering may be required to achieve cluster estimation.

Multiresolution histogram [5] and hierarchical clustering [6] have been introduced as promising image segmentation techniques. The multiresolution histogram shares many desirable properties with the plain histogram including that they are both

computationally fast and robust to noise. Thus, multiresolution histograms can improve segmentation results.

Puzicha [7] proposed an unsupervised segmentation method based on histogram clustering introducing a statistical latent class model for probabilistic grouping of distributional and histogram data. Motivated by this challengeable problem, we introduce a novel approach to estimate the number of clusters and its centroids, based on a multiscale histogram analysis. Our proposal was inspired by [8], where multiscale analysis is employed to evoked potential signal filtering. By applying the *à trous* algorithm [9], the signal is decomposed into several redundant scales, without decimation, and the interscale correlation is computed for peak detection. Our scheme uses a histogram scale-space analysis and an interscale correlation is evaluated in the first level of wavelet decomposition. Thus, we extract the number of clusters and its centroids from the image histogram. Therefore, this information may be used by segmentation algorithms which rely on clusters information as input parameter.

II. THE MULTISCALE CLUSTERING ALGORITHM

Image segmentation algorithms based on K-means and level sets which encompass information about the number of clusters and their centroids are susceptible to provide different results, if the centroids are randomly selected. Thus, the most probable approach to achieve satisfactory segmentation results is to process the image for several rounds of clustering.

In order to solve this instability without choosing information about clusters heuristically, the proposed method estimates the number of clusters and its centroids according to the peaks which are identified in image histogram. Thus, our method correlates the smoothed histogram with its decomposed versions by using an undecimated wavelet. Additionally, it identifies peaks in different scales that point to the dominant clusters in images.

The algorithm consists of the following steps:

- 1) obtain the image histogram;
- 2) smooth the original histogram applying a Gaussian filter;
- 3) decompose the smoothed histogram by using the *à trous* algorithm and correlate signals at the first scale;
- 4) estimate the number of clusters and its centroids.

After the scale-space filtering step, the smoothed histogram W_0 is decomposed and correlated with the first level of wavelet detail coefficients W_1 . Here, we adopted the correlation formulation introduced in [8], which states that $corr_0(n) = W_0(n) \cdot W_1(n)$. Then, the correlation signal is compared to the smoothed histogram and the points that satisfy the condition $corr_0 > |W_0|$ are mapped into the filtered histogram.

Point groups which are in the neighborhood of a peak represent a cluster, and hence these groups are used to estimate the centroids. In fact, the histogram smoothing process displaces peaks and, furthermore, to handle this effect we estimate the centroids in the original histogram signal adopting the local maximum of each group.

A. The Multiscale Gaussian Filtering and Cluster Detection

The histogram smoothing process can be defined as a scale-space representation, where the standard deviation (σ) of the Gaussian filter is the scale parameter. The essential idea in the scale-space approach is to embed the original signal in a family of derived ones obtained by a Gaussian kernel convolution [10]. Actually, without the smoothing step, the original histogram is prone to false cluster detection. Hence, the proposed method overcomes this problem by applying the Gaussian filter to the histogram signal.

Fig. 1(a) shows an image histogram that is highly variable. And Fig. 1(b) and Fig. 1(c) illustrate the smoothed versions of this original histogram, which were originated by different smoothing levels ($\sigma = 2, L = 5$ and $\sigma = 5, L = 11$, respectively).

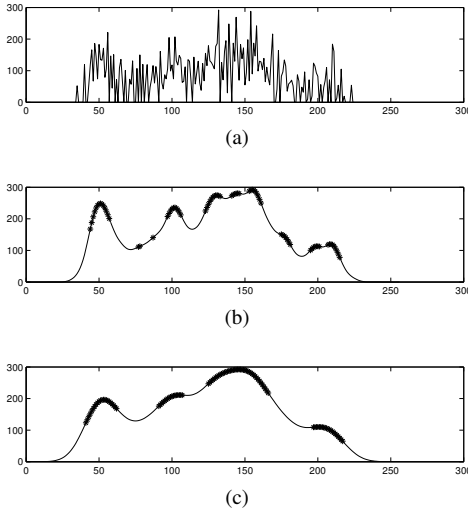


Fig. 1. Cluster detection for different smoothing levels: (a) original histogram and its smoothed versions using (b) $\sigma = 2, L = 5$ and (c) $\sigma = 5, L = 11$. The detected clusters are indicated by (*).

It is worth mentioning that different smoothing levels lead to different segmentation results, because the number of clusters differs for each level. Fig. 2 shows how the multiscale proposed method performs for different smoothing levels and it also illustrates that the standard deviation value (σ) determines the number of clusters. Moreover, this number of clusters

is dependent on the window size (L) and it tends to a constant value as the window size increases. Here, we have followed [11] and assumed $L = 6\sigma$. Thus, the scale-space representation has only a single parameter σ .

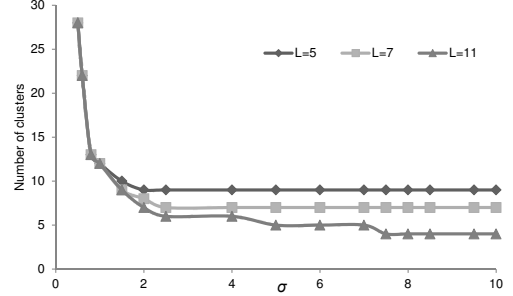


Fig. 2. Number of clusters x standard deviation.

III. IMAGE SEGMENTATION APPLICATION

The proposed method is able to improve segmentation techniques which are dependent on parameters related to clusters. Furthermore, it solves the uncertainty related to the number of clusters and the randomness of centroids and its effects over these techniques. Here, we performed segmentation experiments to achieve 2 clusters, i.e. binary results by applying thresholding, K-means and level set methods to a set of synthetic noisy images and by using the estimated clusters. In this section, we briefly describe these methods.

A. Classical Image Segmentation Methods

1) *Thresholding Method*: Thresholding is a simple but effective method for image segmentation and it achieves good results for images with bimodal histograms. Here, the main purpose of this method is to separate image background from foreground, which yields a binary result. In fact, thresholding is a well-researched field and hence there are many algorithms to determine an optimal thresholding [6].

2) *K-means algorithm*: K-means is an unsupervised classification algorithm based on clustering. It assigns image pixels to a predetermined class K on the basis of minimizing the error function [4]. A pixel is usually assigned to the cluster whose centroid is closest to it in the Euclidean sense. The clustering process stops when there is no more back and forth movements of the pixels from one cluster to another.

3) *Level set algorithm*: The level set theory formulated by Sethian [12] is based on Hamilton-Jacobi equation. It states that a closed curve $\vec{\phi}$ evolves with a velocity field $(\pm F\vec{\eta})$, where $(\pm\vec{\eta})$ is the unit vector that is perpendicular to $\vec{\phi}$.

The movement of front $\vec{\phi} : \mathbb{R}^2 \times \mathbb{R}^+ \rightarrow \mathbb{R}$ is a consequence of the surface $\Psi(\mathbf{x})$ motion, in which one is embedded in accordance with:

$$\vec{\phi}(s, t) = \{\mathbf{x} | \Psi(\mathbf{x}, t) = 0\}, \mathbf{x} \in \mathbb{R}^n, \quad (1)$$

where $s \in \mathbb{R}^2$ is the coordinate of the parametric curve $\vec{\phi}$ and $t \in \mathbb{R}^+$ is the evolution time. This process is expressed as [12]

$$\psi^{k+1} = \psi^k + \Delta t \times \psi_t \cdot |\nabla \psi^k|, \quad (2)$$

where ψ^k is the level set function, Δt is the time step, and ψ_t is the evolution function. The operator $|\nabla \psi^k|$ stands for the gradient magnitude.

The evolution function comprises a regularization term based on the curvature of ψ^k , a term of advection, and a propagation (or velocity) term F . Since the advection term is not commonly used in segmentation applications, the evolution function is defined as

$$\psi_t = F + \varepsilon \mathcal{K}, \quad (3)$$

where $\varepsilon \in \mathbb{R}$. The curvature \mathcal{K} is defined in [12].

The speed for a two-phase level set method, i.e., a binary image segmentation is given by

$$F = F_1(\vec{\eta}) + F_2(-\vec{\eta}). \quad (4)$$

In this paper, we have used the likelihood model based on the normal distribution [13].

B. Similarity Measure for Segmentation Evaluation

In order to provide a quantitative analysis of the proposed method, we have used the Relative Foreground Area Error (RAE) [14], which is given by

$$RAE = \begin{cases} \frac{A_0 - A_T}{A_0} & \text{if } A_T < A_0 \\ \frac{A_T - A_0}{A_T} & \text{if } A_T \geq A_0 \end{cases} \quad (5)$$

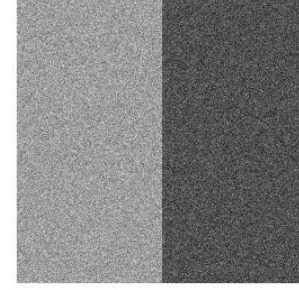
where A_0 is the area of the reference image and A_T is the area of the thresholded image. RAE measures the discrepancy of a thresholded image with respect to a reference image. In our experiments, we have adopted a manually segmented binary image as reference. When the segmented regions and the reference ones match, RAE is zero and it denotes a perfect matching. On the other hand, if these regions do not match, it means that there is no overlapping between these regions and the penalty reaches the maximum value, i.e. one. Thus, the similarity measure is straightforward

$$S_r(\%) = 100 \times (1 - RAE). \quad (6)$$

IV. EXPERIMENTAL RESULTS

To demonstrate the robustness of the proposed method, we have estimated the number of clusters and centroids in synthetic images contaminated with additive gaussian noise.

The segmentation result in Fig. 3(b) was accomplished by applying the Otsu's method and the similarity measure reached $S_r = 97.56\%$. The proposed method estimated the clusters and we assumed the threshold value as the minimum valley between two adjacent centroids. Moreover, it identified two clusters with a threshold $T = 124$, and the segmentation result



(a)

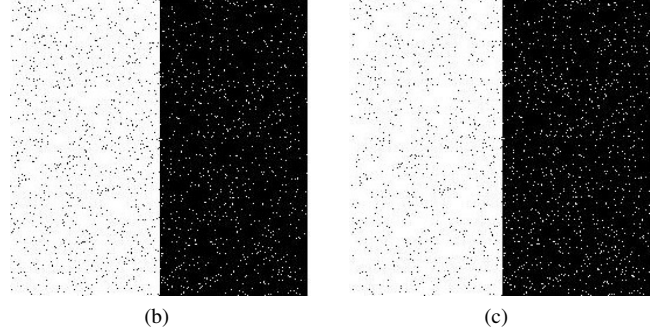


Fig. 3. Segmentation results: (a) noisy image, (b) Otsu's method result ($T = 126$) and (c) proposed method result ($T = 124$).

achieved $S_r = 98.10\%$ which denotes a good matching between the segmented regions and the reference ones. Fig. 3(c) illustrates this segmentation result.

Fig. 4 depicts an image processed similarly, and for which our method detected two dominant clusters and estimated their centroid values (76; 153). Fig. 5 shows the clusters and centroids.

Fig. 6 illustrates the results of segmentation methods applied to the noisy image presented in Fig. 4. The Otsu's threshold was $T = 113$ and the result (Fig. 6(a)) reached $S_r = 79.22\%$. Fig. 6(b) shows the classical K-means result and according to $S_r = 71.72\%$ value and visual analysis, this method performed worse than the others.

Fig. 6(c) to Fig. 6(e) correspond to results from the thresholding ($S_r = 82.98\%$), K-means ($S_r = 84.02\%$) and level set ($S_r = 99.25\%$) algorithms, respectively. Regarding our method, the result comprised regions with a foreground cluster (in white) and centroid value equal to 153, and a background cluster (in black) with centroid equal to 76.

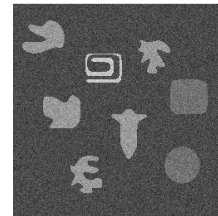


Fig. 4. Noisy image used in tests.

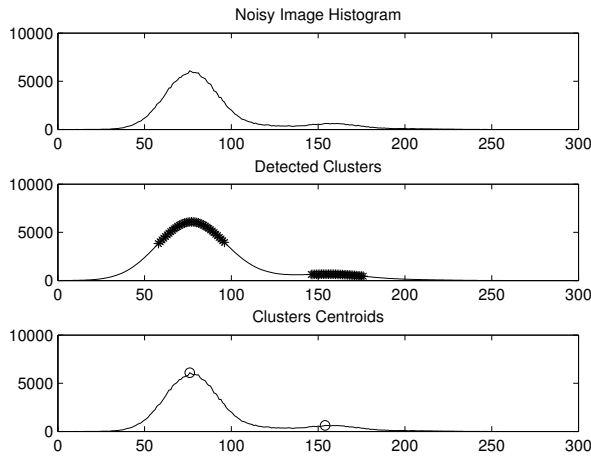


Fig. 5. Detected clusters in noisy image of fig. 4(top to down) noisy image histogram, detected clusters (*) and its centroids (o).

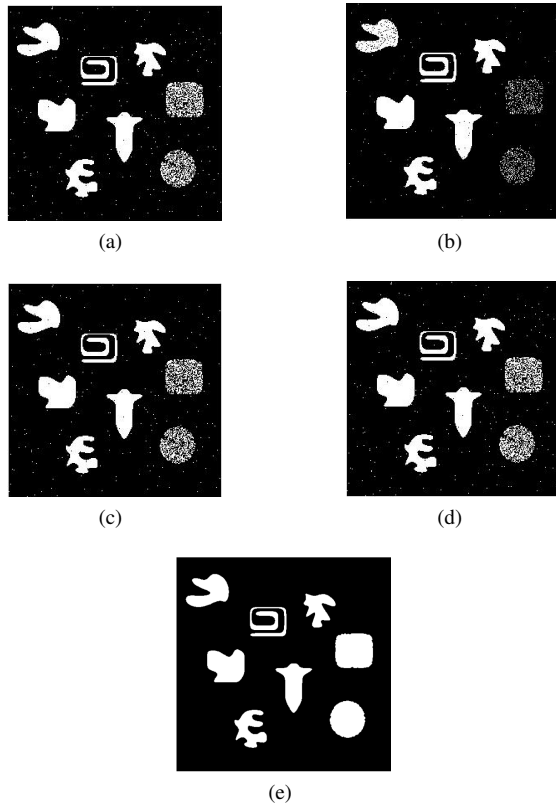


Fig. 6. Segmentation results of noisy image in Fig. 4: (a) Otsu's thresholding result, (b) Classical K-means result, and (c)-(e) results by using the proposed method for cluster estimation, (in sequence): thresholding, K-means and level set algorithms.

V. CONCLUDING REMARKS

In this paper, we proposed an approach to detect clusters that improved the performance of segmentation techniques and solved the uncertainty regarding the number of clusters and centroid parameters. Three segmentation methods (threshold-

ing, K-means and level set) which rely on these parameters were applied to noisy images in order to assess our multiscale approach for cluster estimation. It is noteworthy that the segmentation results of noisy images proved the effectiveness of proposed method to detect clusters in plain histograms.

The thresholding results obtained with the proposed method and Otsu's method achieved high accuracy values, as the similarity measure S_r indicated. Thus, we conclude from the experiments that the proposed approach is simple and effective for cluster estimation.

From the segmentation results of the K-means and level set algorithms using the proposed method, we conclude that it improved their performance and, thus, it is promising for cluster estimation.

Moreover, further work will investigate the applicability of the proposed method for multiple cluster detection (> 2). A multiphase level set implementation should also be investigated with multiple clusters.

ACKNOWLEDGMENT

Author would like to thank FUNCAP by financial support (#PJP72000910100/12) and CNPq (#301264/2013 – 9).

REFERENCES

- [1] C.-H. Lin, C.-C. Chen, H.-L. Lee, and J.-R. Liao, "Fast k-means algorithm based on a level histogram for image retrieval," *Expert Systems with Applications*, vol. 41, pp. 3276–3283, 2014.
- [2] Y. Wang, S. Xiang, C. Pan, L. Wang, and G. Meng, "Level set evolution with locally linear classification for image segmentation," *Pattern Recognition*, vol. 46, pp. 1734–1746, 2013.
- [3] G. Thomas, D. Flores-Tapia, and S. Pistorius, "Histogram specification: A fast and flexible method to process digital images," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 5, pp. 1565–1578, May 2011.
- [4] H. Yao, Q. Duan, D. Li, and J. Wang, "An improved k-means clustering algorithm for fish image segmentation," *Mathematical and Computer Modelling*, vol. 58, pp. 790–798, 2013.
- [5] E. Hadjidemetriou, M. Grossberg, and S. Nayar, "Multiresolution histograms and their use for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 831–847, July 2004.
- [6] A. Z. Arifin and A. Asano, "Image segmentation by histogram thresholding using hierarchical cluster analysis," *Pattern Recogn. Lett.*, vol. 27, no. 13, pp. 1515–1521, 2006.
- [7] J. Puzicha, T. Hofmann, and J. M. Buhmann, "Histogram clustering for unsupervised segmentation and image retrieval," *Pattern Recognition Letters*, vol. 20, pp. 899–909, 1999.
- [8] G. Sita and A. G. Ramakrishnan, "Wavelet domain nonlinear filtering for evoked potential signal enhancement," *Computers and Biomedical Research*, vol. 33, no. 6, pp. 431–446, December 2000.
- [9] M. J. Shensa, "The discrete wavelet transform: Wedding the trous and mallat algorithms," *IEEE Trans. Signal Processing*, vol. 40, no. 10, pp. 2464–2482, October 1992.
- [10] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, Jul 1990.
- [11] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. New Jersey: Prentice Hall, January 2002.
- [12] J. A. Sethian, *Level Set Methods and Fast Merging Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Comput. Vision and Materials Science*, 1st ed. Cambridge: Cambridge University Press, 1999.
- [13] A. Mitiche and I. B. Ayed, *Variational and Level Set Methods in Image Segmentation*. Berlin: Springer-Verlag, 2010.
- [14] S. H. Shaikh, A. Maiti, and N. Chaki, "Image binarization using iterative partitioning: A global thresholding approach," *IEEE International Conference on Recent Trends in Information Systems*, pp. 281–286, 2011.

Medição de Distâncias utilizando o Princípio de Visão Estereoscópica

Marco Antônio Dalcin, Julia Grando, Rafael Follmann Faccenda,
Simone Regina Ceolin, Andrei Piccinini Legg e Osmar Marchi dos Santos
Centro de Tecnologia – CT
Universidade Federal de Santa Maria – UFSM
Santa Maria, Brasil

Abstract—This paper refers about the measurement of distances based on the principle of stereoscopy with the objective of assisting the navigation of a quadcopter. The distance calculus is the first step in the development of this vehicle that will be used in forest mapping for future analyses of deforestation. The experiment used two cameras and the object detection was done with the image processing library OpenCV. The developed algorithm works in real time and has as input images captured from each camera and, using pixel disparity, results the distance to a certain object.

Keywords—Stereoscopy, OpenCV, Distance Measurement.

Resumo—Este paper trata o cálculo de distâncias baseando-se no princípio da estereoscopia com o objetivo de auxiliar na navegação de um quadricóptero. O cálculo da distância é o primeiro passo para o desenvolvimento desse veículo que será utilizado no mapeamento de matas para futuras análises de desmatamento. No experimento utilizou-se duas câmeras e a detecção do objeto foi realizada usando a biblioteca de processamento de imagens OpenCV. O algoritmo desenvolvido trabalha em tempo real e tem como entrada as imagens capturadas de cada câmera e, utilizando disparidade de pixels, resulta a distância a um determinado objeto.

Keywords—Estereoscopia, OpenCV, Cálculo de Distância.

I. INTRODUÇÃO

Acoplar câmeras a veículos não tripulados é uma poderosa ferramenta para análise territorial, mapeamento tridimensional de ambientes e exploração de lugares desconhecidos[1]. O objetivo deste trabalho é desenvolver uma visão computacional[2] para o sistema de navegação de um quadricóptero, tornando-o assim um veículo autônomo[3].

Uma das principais orientações necessárias para o veículo navegar sozinho é a noção de distância entre ele e obstáculos[2]. Existem vários modos de realizar o cálculo da distância, como utilizando sensores ativos (ondas de rádio, infravermelhas, ultrassônicas, etc.) que enviam sinais para identificar a posição do objeto. Além dos sensores ativos, têm-se também técnicas que apenas captam informações sobre o ambiente, tal como a estereoscopia[4].

A visão estereoscópica utiliza um conjunto binocular composto de duas câmeras alinhadas sobre o mesmo eixo e separadas a uma distância aproximada à dos olhos humanos. Essa é uma técnica que permite extrair informações de tamanho, distância e profundidade do meio a partir de uma perspectiva

tridimensional. Isso é possível devido à disparidade entre as imagens capturadas[5].

II. TRABALHOS RELACIONADOS

Existem estudos que analisam diferentes aspectos à respeito do uso da estereoscopia para o cálculo da distância até um determinado objeto. Mahammed *et al.*[6] utilizaram esse método comparando os resultados obtidos com os de um laser que mede distâncias e obtiveram um erro de no máximo 2% para distâncias de até 1m.

Mrovlje e Vrancic[4] focaram seu trabalho em distâncias mais elevadas, entre 10m e 60m, e também mediram os efeitos nos resultados conforme a variação da distância entre as duas câmeras. Com isso eles concluíram que é possível obter mais precisão com uma maior distância entre as câmeras.

Esteves *et al.*[7] utilizaram esse método para o controle de uma plataforma robótica, no entanto foram usadas imagens estáticas e distâncias de no máximo 50cm.

A diferença desse trabalho é que se trata de medição de distâncias entre 15cm e 4,5m utilizando imagem em tempo real e um algoritmo, escrito em C++, de processamento leve e ideal para sistemas embarcados.

III. TÉCNICA UTILIZADA

A visão estereoscópica é a sensação tridimensional que o ser humano tem a partir da combinação das imagens capturadas em cada olho e interpretadas pelo cérebro. Com as duas imagens combinadas é possível ter uma noção de profundidade, distância, tamanho e velocidade em um ambiente. Isso ocorre devido à disparidade binocular do sistema visual humano, que gera duas imagens ligeiramente diferentes quando uma cena é projetada nas retinas dos olhos[8].

Através da estereoscopia é possível calcular a distância D entre o par de câmeras e o objeto focado nas imagens. Por meio de derivações geométricas obtemos:

$$D = \frac{B}{\tan(\varphi_1)\tan(\varphi_2)} \quad (1)$$

Onde B é a distância entre as câmeras. φ_1 e φ_2 são os ângulos entre o eixo das câmeras e o objeto. Através da Figura 1 é possível encontrar:

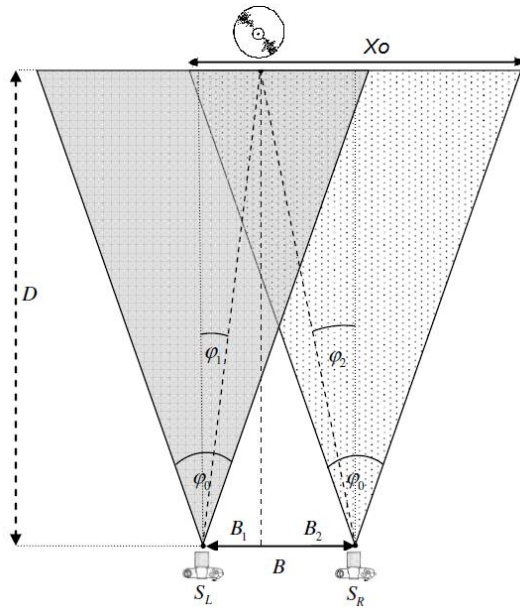


Figura 1. Esquema ilustrando a visão estereoscópica.

$$D = \frac{Bx_0}{2\tan\left(\frac{\varphi_0}{2}\right)(x_L - x_D)} \quad (2)$$

Onde x_0 representa o tamanho da abertura da câmera até o objeto, x_L e x_D representam o centro do objeto na imagem capturada e por substituições trigonométricas, troca-se φ_1 e φ_2 por φ_0 .

Nesse presente trabalho utiliza-se:

$$\Delta p = x_L - x_D \quad (3)$$

IV. IMPLEMENTAÇÃO

O algoritmo de detecção do objeto é basicamente composto pela função *HoughCircles*[9] contida na biblioteca do *OpenCV*, que percorre a imagem detectando e destacando elementos circulares. A vantagem de usá-la é que se trata de uma função de simples aplicação e, além de identificar os círculos nas imagens usadas como parâmetros, desenha o contorno deles e fornece as coordenadas do centro, estas que foram muito importantes para obter Δp .

Para que as imagens fossem processadas pela *HoughCircles*, elas foram primeiramente modificadas de coloridas para escala de cinza e passaram pela função *GaussianBlur*[10], que suaviza a imagem, reduzindo os ruídos[11]. Com essa função é possível evitar que círculos indesejados sejam detectados pela função *HoughCircles*, aumentando a eficiência desta e, consequentemente, aumentando a precisão do cálculo da distância.

As imagens foram obtidas através de um par de câmeras fixado sobre um eixo plano com marcações de distância, como mostra a Figura 3. O círculo a ser identificado era deslocado ao longo das medições.

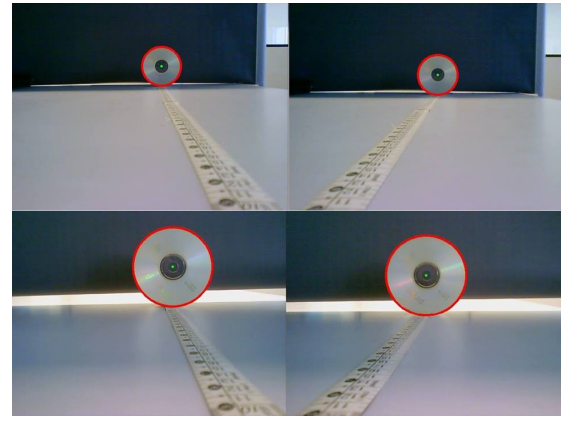


Figura 2. Demonstração da função *HoughCircles* em imagens capturadas pelas câmeras. As imagens superiores foram capturadas na distância de 100cm e as inferiores em 50cm.

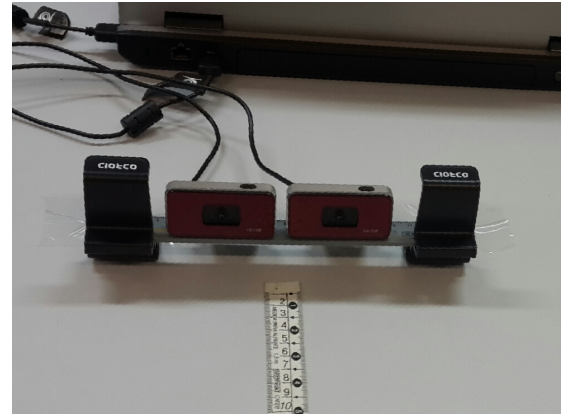


Figura 3. Configuração das câmeras.

V. EXPERIMENTO

Para comprovar a viabilidade do método proposto, o experimento foi realizado em duas partes.

A. Coleta de dados

A primeira parte consiste em obter a diferença de *pixels* (Δp) em variados pontos com distâncias reais conhecidas e a partir dos dados gerar uma função aproximada $f(\Delta p)$.

B. Análise de eficiência

Para comprovar a eficiência do método, coloca-se o objeto em 5 pontos diferentes (15cm, 45cm, 70cm, 115cm e 450cm) e calcula-se a distância a partir do algoritmo modificado, que agora tem implementado a função $f(\Delta p)$ obtida no passo de coleta de dados.

C. Material Utilizado

- Notebook com processador Intel® Core™ i7-2640M CPU @ 2.80GHz, RAM 6GB e GPU ATI® Radeon™ HD6630M 1GB
- Sistema Operacional Ubuntu 13.10
- OpenCV 2.4.6
- Câmeras USB Odroid 720p

VI. RESULTADOS

A parte de coleta de dados apresentada na Seção V foi realizada em uma resolução de 640x480 utilizando 28 pontos espaçados em 5cm, com o ponto de número 1 posicionado no centímetro 15 da régua de medição. A Tabela I e a Figura 4 apresentam os resultados.

Tabela I

DADOS OBTIDOS PELA COLETA. D REPRESENTA A DISTÂNCIA REAL ENTRE O OBJETO E AS CÂMERAS.

Linha	D (cm)	Δp	D (cm)	Δp
1	15	341	85	87
2	20	270	90	83
3	25	224	95	80
4	30	192	100	78
5	35	170	105	76
6	40	153	110	74
7	45	139	115	72
8	50	126	120	70
9	55	118	125	68
10	60	110	130	66
11	65	104	135	65
12	70	98	140	64
13	75	95	145	63
14	80	90	150	62

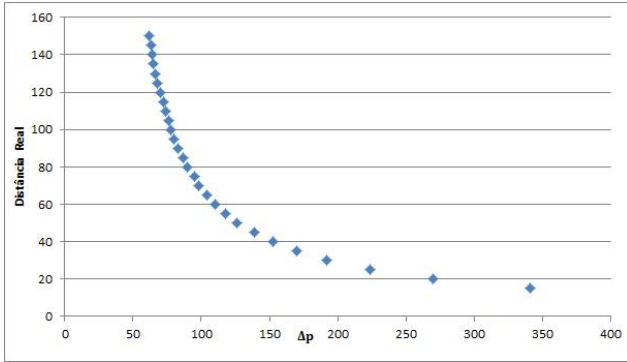


Figura 4. Representação gráfica dos dados obtidos.

Através da comparação das linhas 1 e 14 da Tabela I, percebe-se que quanto maior a distância entre o objeto e as câmeras, menor é a disparidade de *pixels* entre as imagens. Quando observada a plotagem desses pontos na Figura 4 se nota que esse decrescimento comporta-se de maneira exponencial.

A partir desses dados, utilizando o software *Matlab* foi obtida a função aproximada $f(\Delta p)$, expressada por 4.

$$f(\Delta p) = 1377e^{-0.04384\Delta p} + 101.4e^{-0.006907\Delta p} \quad (4)$$

A. Eficiência e Limitações

Seguindo o processo de análise de eficiência citado na Seção V, foram obtidos os resultados da Tabela II.

Observa-se que pontos fora das distâncias em que foi feita a coleta de dados não apresentam tanta precisão quanto os de dentro da área de medição, como mostra na última linha da tabela o maior erro na medição de 4,5 m.

Tabela II
RESULTADOS DA PERFORMANCE DA FUNÇÃO.

D (cm)	$f(\Delta p)(cm)$	Erro
15	14,8	1,3%
45	43,9	2,4%
70	70	0%
115	117,4	2,1%
450	435	3,3%

VII. CONCLUSÃO E TRABALHOS FUTUROS

Nesse método é possível medir a distância das câmeras até o objeto utilizando apenas a disparidade de *pixels* entre as imagens obtidas de cada câmera. Os resultados devem variar quando usadas diferentes câmeras e/ou resoluções.

O algoritmo apresentou erros entre 1-2% em distâncias menores que 1,5m e 3,3% de erro em uma distância de 4,5m. Isso ocorre pois quanto mais distante está o objeto, menor é a disparidade que as imagens apresentam, incidindo em uma perda gradativa de precisão. Os erros são aceitáveis para o propósito, pois não são significativos ao ponto de influenciar na tomada de decisões da inteligência artificial do quadricóptero.

Como trabalho futuro, objetiva-se testar o método desenvolvido com diferentes resoluções para avaliar variações na precisão do cálculo da distância. Além disso, pretende-se aprimorar o algoritmo de detecção de objetos, que atualmente é aplicado somente para círculos.

REFERÊNCIAS

- [1] G. Borges, *Um sistema óptico de reconhecimento de trajetórias para veículos automáticos*. Copele, DEE/UFPB, Campina Grande, Paraíba, Brazil, 1998
- [2] J. A. de França, *Desenvolvimento de Algoritmos de Visão Estereoscópica para Aplicações em Robótica Móvel*. Ph.D. dissertation, Universidade Federal de Santa Catarina, Florianópolis, Brazil, 2005
- [3] K. Schauwecker, N. R. Ke, S. A. Scherer and A. Zell, *Markerless Visual Control of a Quad-Rotor Micro Aerial Vehicle by Means of On-Board Stereo Processing*. Department Cognitive Systems, Wilhelm-Schickard-Institute for Computer Science, University of Tübingen, Germany, 2012
- [4] J. Mrovlje and D. Vrancic, *Distance measuring based on stereoscopic pictures*. in 9th International PhD Workshop on Systems and Control: Young Generation Viewpoint 1.- 3. October 2008, Izola, Slovenia
- [5] M. E. Stivanello, *Desenvolvimento de uma Biblioteca para Sistemas de Visão Estereoscópica para Robótica Móvel*. Master's thesis, Universidade Federal de Santa Catarina, Florianópolis, Brazil, 2008
- [6] M. A. Mahammed, A. I. Melhum and F. A. Kochery, *Object Distance Measurement by Stereo VISION*. in International Journal of Science and Applied Information Technology (IJSAT), Vol. 2, No. 2, Pages: 05-08, 2013
- [7] G. R. P. Esteves, M. A. F. Feitosa and B. J. T. Fernandes, *Estereoscopia no cálculo de distância e controle de plataforma robótica*. in SIBGRAP 2012
- [8] A. B. Raposo, F. Szenberg, M. Gattas and W. Celes, *Visão Estereoscópica, Realidade Virtual, Realidade Aumentada e Colaboração*. Tecgraf – Grupo de Tecnologia em Computação Gráfica, Departamento de Informática, PUC-Rio, 2004
- [9] http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/hough_circle/hough_circle.html Accessed on May 4th, 2014
- [10] <http://docs.opencv.org/modules/imgproc/doc/filtering.html#gaussianblur> Accessed on May 4th, 2014
- [11] R. Leganière, *OpenCV 2 computer Vision Application Programming Cookbook*. 1. ed. Birmingham UK: Packt Publishing Ltd, 2011
- [12] G. Bradski and A. Kaebler, *Learning OpenCV*. 1. ed. Sebastopol USA: O'Reilly, 2008

Estereoscopia para Mapeamento 3D

Cristiano Santos*, Eduardo Speroni*, Marco Dalcin[†], Simone Ceolin[‡], Osmar Santos[‡], Andrei Legg[‡]

* Alunos de Mestrado UFSM, Emails: {cristiano.santus, edusperoni}@gmail

[†] Aluno de Graduação UFSM Email: marco_dalcin@hotmail.com

[‡] Professores Drs. UFSM, Emails: sceolin@redes.ufsm.br, osmar@inf.ufsm.br, andrei.legg@gmail.com

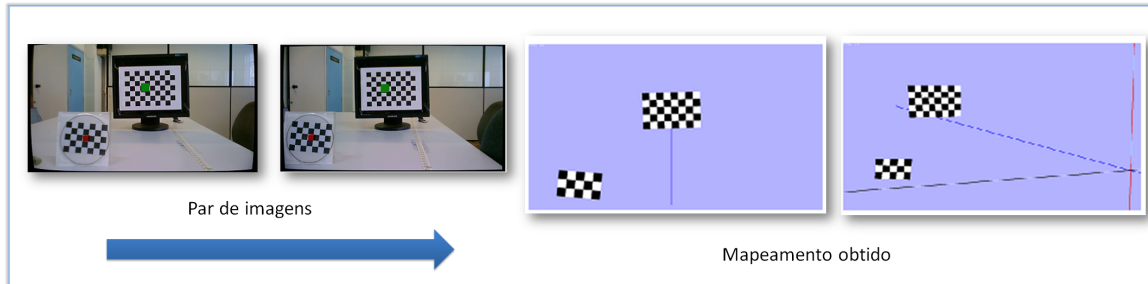


Figure 1. Resultado do mapeamento: O par de imagens de entrada á esquerda e imagens resultante do mapeamento á direita.

Abstract—This paper presents a study of stereoscopic vision with the purpose of mapping external environments. The initial proposal is to calibrate the system and generate a 3D model as starting point. A new approach is presented as an alternative to the use of disparity maps, with the goal of obtaining more precise measures from segmented regions while minimizing the computational cost.

Keywords—Stereoscopic; 3D Mapping; Computer Vision;

Resumo—Este artigo apresenta um estudo sobre visão estereoscópica com o propósito de realizar o mapeamento de ambientes externos. A proposta inicial se concentra em realizar a calibração do sistema e geração de um modelo 3D como ponto de partida. Uma nova abordagem é apresentada como alternativa ao uso do mapa de disparidades, com o propósito obter medidas mais precisas de regiões segmentadas buscando minimizar o custo de processamento.

Keywords—Estereoscopia; Mapeamento 3D; Visão Computacional;

I. INTRODUÇÃO

A estereoscopia é uma técnica de visão computacional, que torna possível por meio da aquisição de duas imagens de diferentes pontos de uma cena, aferir a profundidade de um determinado objeto. Em outras palavras, determinar a distância do observador ao objeto observado, ou seja, a profundidade da cena [1]. Também é possível mensurar distâncias entre objetos que compõem uma cena e dessa forma obter um modelo ou mapa tridimensional de um ambiente utilizando apenas duas câmeras.

Na robótica, a estereoscopia é instrumento de estudo, como forma de obter a autonomia na locomoção de robôs, onde destaca-se a utilização em ambientes industriais automatizando procedimentos e na exploração de locais inacessíveis ou considerados perigosos para os seres humanos [2].

O mapeamento de ambientes tem uma ampla aplicação, podendo ser utilizado no auxílio a topógrafos em mapeamento

de diferentes tipos de terrenos por meio de imagens aéreas, na elaboração de projetos nas engenharias por meio de aquisições de imagens 3D e até em missões militares como no caso de um possível mapeamento de território desconhecido [3].

Na literatura atual, são encontradas diferentes abordagens para o mapeamento e modelagem 3D, como a utilização sensores de profundidade e técnicas de tratamento de imagens [4] e [5].

Contribuições: Este trabalho utiliza uma nova abordagem como alternativa ao mapa de disparidades, como forma de obter maior precisão e ganho de desempenho. Nesse contexto, pretende-se desenvolver o mapeamento de ambientes externos com a utilização estereoscopia através de imagens obtidas de um quadricóptero. Dessa forma, objetiva-se contribuir para realização do mapeamento a fim de: detectar desmatamentos em regiões florestais, auxiliar em missões militares e até mesmo em projetos arquitetônicos.

Este artigo esta organizado da seguinte forma: Na Seção 2 são apresentados alguns trabalhos relacionados. A Seção 3 aborda uma visão geral de estereoscopia e a Seção 4 apresenta o detalhamento. A implementação é apresentada na Seção 5 e posteriormente na Seção 6 é relatada a conclusão do trabalho e o que pretende-se de trabalhos futuros.

II. TRABALHOS RELACIONADOS

Dentre os trabalhos que utilizam a técnica visão estéreo, destaca-se a proposta de Po-Kuan et al [6], que utiliza da técnica para reconhecimento de gestos de mão, onde é obtido um processamento de tempo real com resolução de até 1080px. Porém é importante ressaltar que para esse trabalho o ponto de interesse era o mapeamento somente da mão do usuário. Ulusoy et al [7], faz uma abordagem da visão estéreo com o objetivo de construir uma mapa 3D, sendo utilizado um ambiente que simula um ambiente do mundo real. Por fim, Scaramuzza et al [8] faz simultaneamente o mapeamento e localização através de quadricóptero e a utilização de um sistema monocular (uma única câmera), completamente em-

⁺ Esse trabalho é parcialmente financiado pela FAPERGS, CAPES e pelo Exército Brasileiro

barcado. Neste contexto, o trabalho em questão busca um mapeamento de ambiente assim como abordado por Scaramuzza et al [8], porém com a utilização de um sistema binocular, ou seja duas câmeras, dispensando dessa forma o uso de outros dispositivos.

III. VISÃO GERAL

Estereoscopia é um processo pelo qual por meio de duas imagens de um objeto, assumindo que essas sejam obtidas por duas câmeras idênticas e separadas por uma determinada distância, é possível adquirir a informação de tridimensionalidade da cena [1]. Para buscar o entendimento de visão estéreo e a geometria envolvida para obter as informações da cena, como a distância dos objetos observados, é interessante destacar a geometria envolvida em apenas uma câmera. O modelo geométrico de uma câmera é ilustrado na Figura 2, onde é chamado basicamente de pinhole (buraco de agulha).

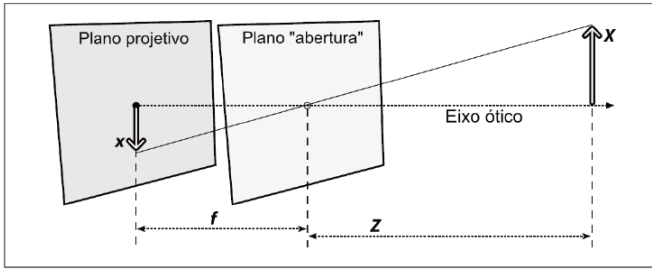


Figura 2. Modelo geométrico de câmera pinhole. Fonte: Bradski [10]

O principal parâmetro desse modelo é a distância focal [9], representada por f , esta é distância entre a abertura da câmera e o plano projetivo. O plano projetivo é o local onde a luz é projetada após ter passado pela abertura da câmera. A distância entre a abertura e o objeto é denotada por " Z ", " X " é o comprimento do objeto no mundo real e " x " na projeção. Sendo conhecida a distância focal é possível obter o comprimento de qualquer objeto no plano projetivo. É importante destacar que no plano projetivo a imagem encontra-se invertida. Também é importante ressaltar que a partir da geometria apresentada, não é possível determinada a distância entre um ponto do mundo e a câmera [10].

Sendo assim, por meio de um sistema canônico se torna possível aferir a distância de um determinado ponto P na cena até o sistema de câmeras. A Figura 3 ilustra o sistema.

Conforme a Figura 3, é possível por triangulação em um sistema ideal, obter a distância do ponto P até o sistemas de câmeras, ou seja, o valor de Z . Sendo assim, a disparidade ou d , é obtida pela equação: $d = x^e - x^d$, onde x^e e x^d correspondem as distâncias dos pontos entre a imagem da câmera da esquerda e direita no eixo x . Neste contexto, as equações presentes em (1), mostram a similaridade entre os triângulos $O_e P O_d$ e $x^e P x^d$ à esquerda e a parte da direita o calcula para distância Z [10].

$$\frac{T - (x^e - x^d)}{Z - T} = \frac{T}{Z} \Rightarrow Z = \frac{fT}{x^e - x^d} = \frac{fT}{d} \quad (1)$$

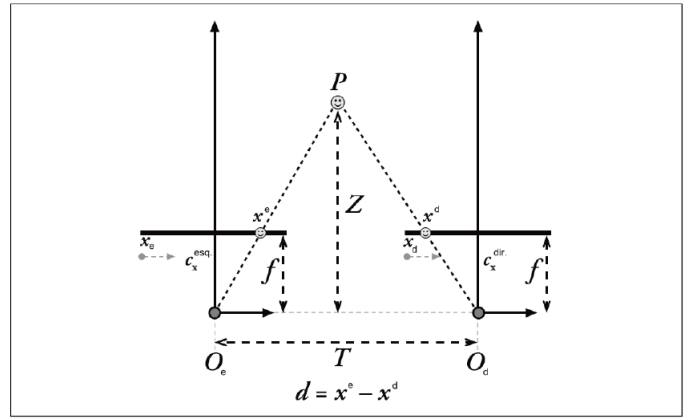


Figura 3. Modelo de Sistema Canônico com distância focal f e câmeras com deslocamento T . Fonte: Bradski [10]

No sistema canônico, a busca por pontos correspondentes poderia ser realizado horizontalmente, porém na prática sistemas estéreo possuem distorções e as câmeras muitas vezes não se encontram alinhadas de forma satisfatória. Diante disso, através da geometria epipolar, é possível fazer o alinhamento virtual das câmeras conforme ilustrado na Figura 4. A partir do alinhamento, torna-se possível reduzir o escopo da busca nas duas dimensões.

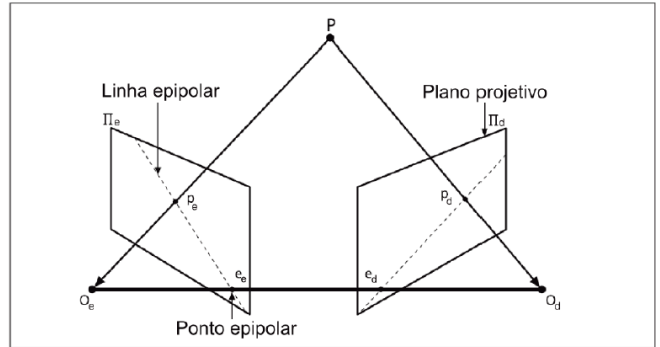


Figura 4. Modelo de geometria epipolar. Fonte: Bradski [10]

Conforme a Figura 4, o centro da projeção é representado por " O_e " e " O_d ", da imagem da direita e esquerda respectivamente. As intersecções que ligam os centros das projeções e os planos de projeção são chamados de pontos epipolares, são eles e_e e e_d , levando em consideração o ponto " P ". Dessa forma, se tem a intersecção do plano epipolar em cada plano projetivo. Assim como mencionado por Scaramuzza et al [9], quando torna-se necessário encontrar o ponto P na imagem da direita, a partir do conhecimento da projeção deste ponto na esquerda, ou seja, conhecendo " P_e " e " E_d ", é possível calcular a linha epipolar e restringir a busca ao ponto " P_d ".

A Correspondência é a área onde se encontram grande concentração de pesquisas, pois é a tarefa que mais dispensa processamento [2]. Além disso, as câmeras ainda possuem distorções, denominada distorção radial formada pelo formato imperfeito da lente [10]. Outra distorção é a tangencial pro-

vocada pelo alinhamento imperfeito da lente em relação ao sensor. Neste contexto o processo de calibração deve ser considerado.

IV. ESTEREOSCOPIA

A estereoscopia é composta por três etapas, sendo elas: Calibração, Emparelhamento, e a geração do Mapa de Disparidade [2].

- Calibração: Estima parâmetros intrínsecos e extrínsecos de cada câmera. Intrínsecos engloba a correção da matriz de distorção radial e determina a distância focal. Parâmetros extrínsecos são as Matrizes de rotação, translação, e fundamental;
- Emparelhamento : Determina a posição dos pontos de um objeto nas duas imagens, dessa forma, viabilizando o cálculo da distância relativa dos pontos;
- Mapa de Disparidades: é a geração de uma terceira imagem que armazena as informações de profundidade da cena, essa etapa necessita do emparelhamento. O fluxo das etapas são ilustradas na Figura 5 conforme segue.

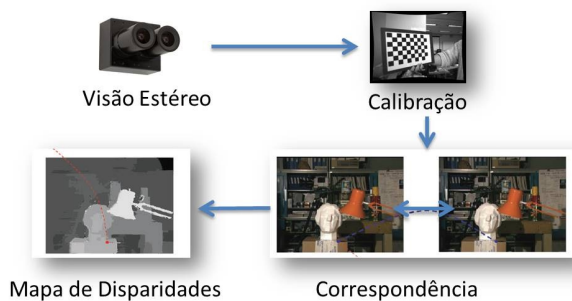


Figura 5. Etapas do sistema estéreo

A Figura 5, esboça as etapas para o sistema estéreo proposto, não considerando o mapeamento 3D, etapa realizada ao final destes procedimentos.

Neste contexto é interessante destacar os principais problemas em cada uma das etapas vista pela técnica de visão estéreo, com o objetivo de obter um melhor entendimento e clareza, os problemas bem como as técnicas sugeridas encontram-se separadas pelas as fases de calibração, emparelhamento e mapa de disparidade.

No processo de calibração, para uso em ambiente dinâmicos como quadricóptero se torna interessante o uso de técnicas mais autônomas com descrito por Orghidan et al [11].

Emparelhamento ou correspondência: Tem como objetivo gerar a disparidade de cada pixel, de certo modo o mapa de disparidade é o resultado da correspondência. A correspondência é a etapa de estereoscopia onde mais se dispensa processamento, por este motivo é a área que mais se concentram pesquisa. Baydoun et al [12] propõem uma nova abordagem que visa obter maior desempenho, nele ainda é destacado o maior problema na correspondência, a oclusão, onde alguns pixels aparecem em uma imagem porém na outra não. Outras abordagens que buscam solucionar desempenho

bem como resolver oclusões são descritas por Cyganek et al [13] e Ju et al, [14].

Mapa de disparidades: O mapa de disparidade pode ser classificado como denso ou esparsos. O denso é um mapa mais completo porém requer a avaliação de todos os pontos envolvidos no par de imagens previamente, o que ocasiona em um custo computacional muito elevado. Em contrapartida mapas esparsos são computacionalmente menos exigentes, porém são insuficientemente capazes de serem utilizados em mapeamento de ambientes. Abordagens como a proposta por Yao et al [15], buscam resolver este problemas com técnicas como interpolação de mapas intermediários. Alternativamente ainda podem ser usadas técnicas de segmentação como sugerido por Klaus et al [16].

V. IMPLEMENTAÇÃO

Em primeiro momento, foi realizada a calibração das câmeras com a utilização da biblioteca OpenCV, pois a mesma possui uma grande quantidade de funções úteis para a validação da implementação. Em seguida para realização da geração do modelo 3D utilizou-se a biblioteca OpenGL, onde o processo foi simplificado à importação das coordenadas relevantes para uma projeção em uma escala de proporcionalidade, resultando na geração de uma forma geométrica com o objetivo de representar o objeto a ser mapeado. Para esta tarefa foram utilizadas duas câmeras USB com resolução 720p (HD) em um ambiente Linux.

Por meio da calibração, foram obtidos os parâmetros intrínsecos e extrínsecos das câmeras. Os parâmetros intrínsecos são compostos pelas matrizes fundamentais das câmeras e dos coeficientes de distorção. Em contrapartida, os parâmetros extrínsecos são compostos pelas posições reais das câmeras na forma de matrizes de rotação e matrizes de projeção. De acordo com estes dados foram projetados raios óticos 3D a partir de pontos 2D da imagem, que foram usados para triangular a posição do objeto.

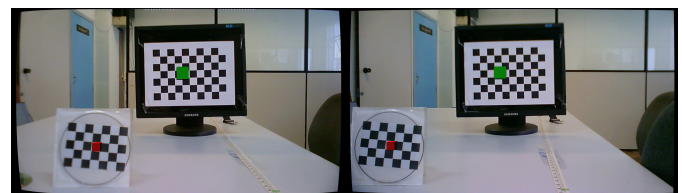


Figura 6. Resultado do mapeamento sobrepondo as imagens de entrada

A Figura 6 demonstra os resultados da projeção do centro dos objetos que compunham o ambiente do mundo real para um ambiente virtual 3D. Este ambiente virtual está sendo exibido por duas câmeras virtuais em posições equivalentes às posições das câmeras reais, com o centro dos objetos demonstrados pelos cubos de cores diferentes. Dois objetos identificados por padrões estavam dispostos sobre uma mesa em frente às duas câmeras. Nos resultados obtidos, os objetos possuem coordenadas x, y e z proporcionais a cena real, e, para distâncias de até 1,5m, o erro foi inferior a 1%.

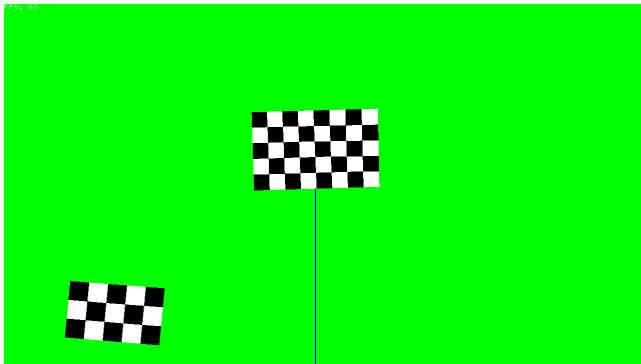


Figura 7. Resultado do mapeamento quando visto da origem

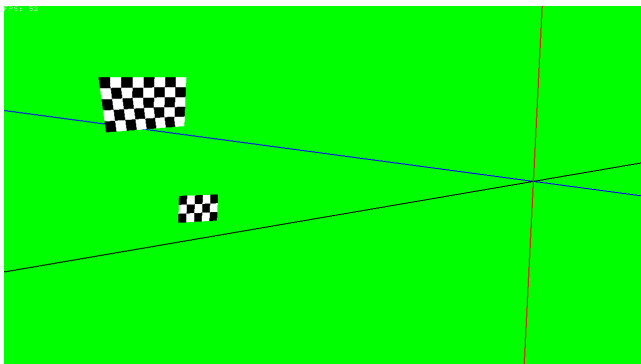


Figura 8. Resultado do mapeamento em perspectiva

A Figura 7 demonstra o resultado final do mapeamento dos dois objetos dispostos para o mundo virtual 3D, renderizado via OpenGL, quando visto da origem. A Figura 8 ilustra o mesmo ambiente virtual em outra perspectiva, demonstrando as proporções de distâncias obtidas. Posteriormente pretende-se transladar e rotacionar as câmeras para um mapeamento completo do mundo real, transladando e rotacionando o mundo virtual proporcionalmente.

Nesta abordagem não foi utilizado o mapa de disparidades devido a não haver necessidade de identificação da profundidade de todos os pontos das imagens e, principalmente, por demonstrar-se uma técnica com elevado custo computacional em nossos testes. No cenário utilizado foi constatado um alto custo computacional por frame processado, o que é inviável para uma aplicação que visa processamento em tempo-real. Considera-se também que o mapa de disparidades é dado por uma imagem em escala de cinza, o que pode limitar os valores que cada ponto pode assumir, e, deste modo, podendo ser menos preciso.

VI. CONCLUSÃO

Como parte inicial de um projeto com objetivo de realizar mapeamento de ambientes através de um quadricóptero, este artigo apresentou o ponto de partida para estudos mais aprofundados em busca de técnicas mais robustas. Os resultados iniciais mostraram-se preliminares, porém válidos para a proposta inicial de calibrar o sistema de câmeras, encontrar

as coordenadas e exporta-la para um ambiente computacional de visão tridimensional. Para trabalhos futuros, pretende-se realizar um procedimento de calibração mais autônomo, bem como uma técnica de correspondência estéreo robusta entre as duas imagens adquiridas, e, conseqüentemente, um mapeamento mais realístico com a utilização de texturas e maior fidelidade em relação aos objetos da cena.

REFERÊNCIAS

- [1] R. Gonzalez and R. Woods, *Processamento de imagens digitais*, 1st ed. Edgard Blucher, 2000.
- [2] D. M. S. Cardoso, "Uso de estereoscopia na movimentação autônoma de robôs," in *Workshops of SIBGRAPI*, A. C. N. J. D. Menotti, Ed., Ouro Preto, MG, Brazil, august 2012. [Online]. Available: <http://www.decom.ufop.br/sibgrapi2012/eproceedings>
- [3] M. B. Skouson, "How the air force should stay engaged in computer vision technology development," DTIC Document, Tech. Rep., April 2007.
- [4] D. Maier, A. Hornung, and M. Bennewitz, "Real-time navigation in 3d environments based on depth camera data," in *Humanoids*. IEEE, 2012, pp. 692–697. [Online]. Available: <http://dblp.uni-trier.de/db/conf/humanoids/humanoids2012.html#MaierHB12>
- [5] M. E. Stivanello and M. R. Stemmer, "Development of an active stereo vision system for robotics applications," *Induscon*, vol. 1, 2012. [Online]. Available: <http://www.eletrica.ufpr.br/anais/induscon/2012/Data/iREP1792.pdf>
- [6] P.-K. Huang, T.-Y. Lin, H.-T. Lin, C.-H. Wu, C.-C. Hsiao, C.-K. Liao, and P. Lemmens, "Real-time stereo matching for 3d hand gesture recognition," in *SoC Design Conference (ISOCC)*, Nov 2012, pp. 29–32. [Online]. Available: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6406917&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D6406917
- [7] I. Ulusoy, U. Halici, and K. Leblebicioglu, "3d cognitive map construction by active stereo vision in a virtual world," in *ISCIS*, C. Aykanat, T. Dayar, and I. Korpeoglu, Eds., vol. 3280. Springer, 2004, pp. 400–409. [Online]. Available: <http://dblp.uni-trier.de/db/conf/iscis/iscis2004.html#UlusoyHL04>
- [8] D. Scaramuzza, M. Achtelik, L. Doitsidis, F. Fraundorfer, E. Kosmatopoulos, A. Martinelli, M. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip *et al.*, "Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in gps-denied environments," *IEEE Robotics & Automation Magazine*, pp. 1–10, 2013. [Online]. Available: http://www.inf.ethz.ch/personal/hengli/papers/sfly_ram2013.pdf
- [9] C. C. T. Mendes, "Navegação de robôs móveis utilizando visão estéreo," Dissertação de Mestrado, Universidade de São Paulo, 2012. [Online]. Available: http://www.teses.usp.br/teses/disponiveis/55/55134/tde-18062012-162436/publico/dissertacao_rev_a4.pdf
- [10] A. Bradski, *Learning OpenCV, [Computer Vision with OpenCV Library ; software that sees]*, 1st ed. O'Reilly Media, 2008, gary Bradski and Adrian Kaehler.
- [11] R. Orghidan, J. Salvi, M. Gordan, C. Florea, and J. Batlle, "Structured light self-calibration with vanishing points," *Machine vision and applications*, vol. 25, no. 2, pp. 489–500, 2014.
- [12] M. Baydoun and M. A. Al-Alaoui, "Enhancing stereo matching with classification," *Access, IEEE*, vol. 2, pp. 485–499, 2014.
- [13] B. Cyganek and J. P. Siebert, *An introduction to 3D computer vision techniques and algorithms*. John Wiley & Sons, 2011.
- [14] M.-H. Ju and H.-B. Kang, "A new method for stereo matching using pixel cooperative optimization," in *Image Processing (ICIP), 16th IEEE International Conference on*. IEEE, 2009, pp. 2105–2108.
- [15] G. Yao, Y. Liu, B. Lei, and D. Ren, "A rapid stereo matching algorithm based on disparity interpolation," in *World Automation Congress (WAC)*. IEEE, 2012, pp. 5–10.
- [16] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *Pattern Recognition. ICPR. 18th International Conference on*, vol. 3. IEEE, 2006, pp. 15–18.

Comparison of Normalization Methods Applied on Face Recognition

Igor L. P. Andrezza, Erick V. C. L. Borges, Adriano S. Marinho and Leonardo V. Batista*

*Centro de Informática

Universidade Federal da Paraíba

João Pessoa, Paraíba, Brazil 58051-900

Email: {igorlpa90, erickvagnerr}@gmail.com, {adriano.marinho, leonardo}@ci.ufpb.br

Abstract—Biometric offers a reliable authentication mechanism that identifies the users through their physical and behavioral characteristics. The problem of face recognition isn't trivial because many factors can affect the face detection and recognition, for example, the lighting, face position, hair, beard, etc. This work proposes to analyze the effects of geometric and lighting normalization on face recognition techniques, aiming to adapt them to uncontrolled environments. Our results show that the lightning normalization improves the performance of the studied face recognition systems.

Keywords—Face Recognition; Normalization; Lighting

I. INTRODUCTION

Computer vision systems can extract information from images. These information can be used to recognize complex patterns such as textures, objects, text and even people [1]. Biometric recognition systems are examples of computer vision systems. They use unique human traits to recognize individuals. Fingerprints, voice and faces are examples of biometric traits [2] [3].

Traditional methods for establishing the identity of a user include knowledge-based mechanisms (e.g. passwords) and token based mechanisms (e.g. ID cards). However, such mechanisms can be lost, stolen or even manipulated in order to circumvent the system. Biometric authentication arises therefore as an alternative [4]. It allows us to establish the identity of a user based on who he is rather than for what he has or what he remembers. The human face is one of the least intrusive biometric traits.

Face recognition systems can operate in two modes: identification or verification [1] [5]. In identification, a face of a query person is compared to a face database in order to assign the person to a person of the database. In verification, a query face and its supposed owner are previously informed. Then, the system compares it with a face database and reports if the owner is correct [6].

Face recognition is not a trivial problem for there are many factors that influence the detection and recognition as, for example, lighting, position of the face in the image, the background image, facial expressions, variations of the beard and hair, skin tone, the use of piercings, glasses, makeup, hair color, etc [2] [7]. For this reason, it is common to find in the literature face recognition systems for controlled environments.

Face recognition methods are classified into holistic, non-holistic (also called analytical or geometric) and hybrid. The holistic methods analyze the face as a whole, typically creating models of faces and making comparisons between these models to perform face recognition. It is possible to perform holistic face recognition using the Discrete Cosine Transform (DCT) [8] [7] [9] [10].

This work analyzes the effects of normalization methods on face recognition systems in order to adapt these systems for uncontrolled environments. To do this, we created a normalization module that: a) normalizes the faces geometrically; b) uses different illumination normalization methods that are described in the literature; c) and uses different cuttings around the face in order to reduce the amount of information that is not part of the face. This module was tested in SDRF (from Portuguese, Sistema para Detecção e Reconhecimento de Face) [7].

II. TECHNICAL BACKGROUND

This section describes some topics that are related to our work.

A. Active Shape Models

Active shape models (ASMs) are statistical models of the shape of objects which iteratively deform to fit to an example of the object in a new image. The shape of the object is represented by a set of points [11]. An active shape model can be used to retrieve specific points of the human face.

B. Histogram Equalization

Filter whose function is to equalize the distribution of gray levels in the image, making its histogram more uniform [12].

C. Histogram Expansion

Operation that changes the histogram of an image as follows: the lowest intensity value is taken to zero and the highest intensity one is taken to the maximum possible value [12].

D. LogAbout

Method for normalize the illumination that consists in applying a high-pass filter and a logarithmic transformation on the image [13].

E. Retinex

Image Enhancement algorithm which is used to improve the contrast, brightness and sharpness through dynamic range compression. It removes the effects caused by different light sources in the scene [14].

III. FACE DATABASES

This section describes the face databases which were used in this work.

A. GTAV Face Database

Face database created to evaluate the robustness of face recognition algorithms in relation to pose and illumination variation. It has 44 individuals with a minimum of 27 images of each one [15]. In this work we used only the frontal images available in the database.

B. Yale Face Database A

The Yale Face Database has images of 15 individuals in 11 poses counting a total of 165 images. The images are all 8 bit gray scale and also vary in face expressions and illumination [16].

C. Extended Yale Face Database B

This version of the Yale Face Database has 16,128 images from 28 individuals in 9 different poses under 64 forms of illumination [16]. We selected ten frontal images from each individual to use in this work.

D. Private

We used a private database developed for this work. It has 84 images (six images for 14 individuals).

IV. IMPLEMENTATION

A. Geometrical Normalization

At first we had to normalize geometrically the faces. To do so, we used the Active Shape Models to retrieve the eye positions because the ASM can approximate the eye positions even when the eyes are occluded.

Once we had the eyes positions, we remove the inclination between the eyes by applying an rotation operation. After that we define the distance between the eyes. The chosen value was 48 pixels because the distance between eyes in 75% of the images was greater than it (see Fig. 1) and because it is better to reduce the images than to enlarge and add information to it.

The next step was to define the size of the images. The images need to be of the same size in order to be used in the SDRF. The chosen size was 124 x 147 pixels (see Fig. 2).

After that, two selections were used for each database before the lighting normalization. Both use the ASM face contour points to retrieve the face. In the first selection, the face image is normalized and put to the 124 x 147 size. In the second selection, the ASM points that define the contour of face were used as a boundary to paint the non-face pixels gray in order to reduce the information that don't belong to the face. These forms are illustrated in Fig. 3.

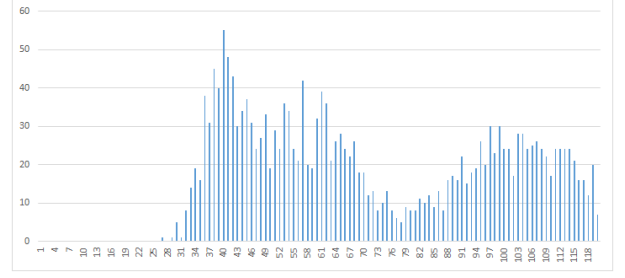


Fig. 1. Histogram of distances of eyes in the tested databases.

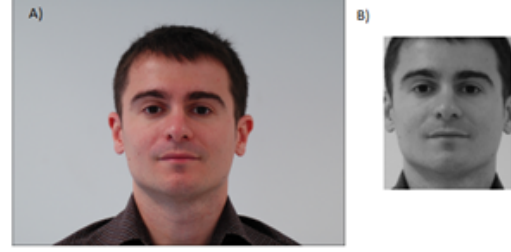


Fig. 2. Result of the geometric normalization. A) Original Image. B) Operation result.

B. Lighting Normalization

Six lighting normalization methods were implemented and tested in order to check the SDRF robustness in relation to lighting. They are: histogram equalization, histogram local equalization, logarithmic filter, LogAbout, Retinex and Retinex plus histogram expansion. They were applied in each selection of each database after the geometrical normalization.

V. RESULTS

This section describes the tests that we made in order to measure the impact of the normalization methods on the SDRF performance. We used 1-to-1 tests and 1-to-N tests. Each database was tested separated and the union of the databases was also tested.

A. 1-to-1 Tests

In this type of test, we compare all faces of a database with each other. This comparison yields a value and if this value is less than a given threshold, the faces are considered to be

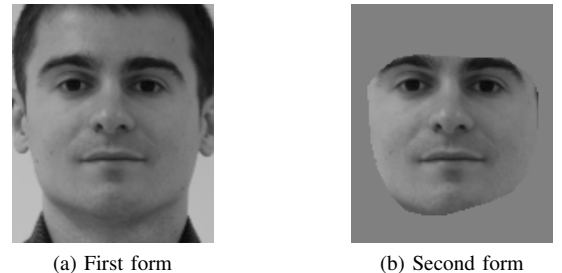


Fig. 3. Forms of selection.

TABLE I
EER RATE OF EACH NORMALIZATION METHOD

	GTAV	PRIVATE	YALE A	YALE B	ALL
Original	23%	10%	23%	6.8%	-
Normalized (1st selection)	34%	45%	37%	31%	28%
Log (1st selection)	31%	40%	32%	33%	28%
Logabout (1st selection)	33%	42%	33%	34%	27%
Equalization (1st selection)	34%	44%	35%	33%	28%
Local Equalization (1st selection)	37%	42%	40%	35%	36%
Retinex (1st selection)	35%	41%	37%	36%	38%
Retinex + expansion (1st selection)	33%	42%	41%	42%	33%
Normalization (2nd selection)	41%	46%	40%	37%	35%
Log (2nd selection)	39%	42%	39%	40%	37%
Logabout (2nd selection)	41%	46%	39%	40%	36%
Equalization (2nd selection)	42%	42%	43%	37%	34%
Local Equalization (2nd selection)	40%	39%	40%	36%	35%
Retinex (2nd selection)	40%	42%	41%	42%	40%
Retinex + Expansion (2nd selection)	43%	42%	42%	40%	33%

TABLE II
RECOGNITION RATE WITH LEAVE-ONE-OUT

	GTAV	PRIVATE	YALE A	YALE B	ALL
Original	86%	100%	90.9%	99.1%	-
Normalized (1st selection)	54.6%	56.4%	70.7%	74.1%	57.3%
Log (1st selection)	61.6%	57.7%	77.2%	70.4%	62.7%
Logabout (1st selection)	68%	61.5%	79.2%	73.7%	69.2%
Equalization (1st selection)	51.7%	59%	75.3%	76.2%	56.7%
Local Equalization (1st selection)	54%	58%	71.4%	70.8%	55.4%
Retinex (1st selection)	60.4%	60%	76%	65.4%	62.2%
Retinex + expansion (1st selection)	54.6%	48.7%	69.4%	64.6%	53.6%
Normalization (2nd selection)	34.8%	42.3%	57.8%	45%	37%
Log (2nd selection)	23.8%	41%	51.3%	35.4%	27.7%
Logabout (2nd selection)	24.4%	27%	53.2%	48.3%	31.5%
Equalization (2nd selection)	35.4%	34.6%	57.8%	56.2%	38.5%
Local Equalization (2nd selection)	26.1%	39.7%	51.9%	55.4%	34.6%
Retinex (2nd selection)	34.3%	30.7%	52%	40.8%	31.5%
Retinex + Expansion (2nd selection)	15.7%	24.3%	41.5%	27.5%	18%

of the same person. This test allow us to check the SDRF performance after each normalization method. We generated a graph showing the False Rejection Rate and the False Acceptance Rate by varying the threshold value. Fig. 4 shows this graph for the Yale Face Database A.

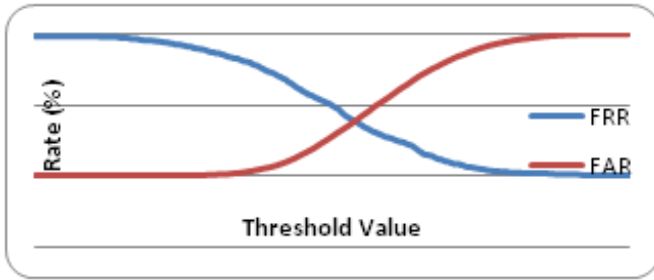


Fig. 4. FRR x FAR from the normalized Yale A face database

The graphs generated by all the normalization methods are similar: they differ only at the joint point of the curves. The joint point is the Equal Error Rate (EER). Table I shows the results of each test.

B. 1-to-N Tests

In each test the cross validation technique was used. Table II shows the results of the test using leave-one-out, where 1

photo of 1 subject is used for query and the remaining for training. Table III shows the results of the test using the inverse of leave-one-out, e.g., 1 photo of 1 subject is used for training. The later test is closer to real scenarios due to commercial systems use one sample for training only.

We used the nearest neighbour classifier in every test. We chose it because using a k-nearest neighbour classifier in the test where only one subject is used for training has no sense.

VI. DISCUSSION AND CONCLUSION

By observing Table I and the graphs generated by the 1 to 1 test, we can tell that the SDRF is not fit for real case scenarios. Removing information from the original image always makes the performance decay. Nevertheless, the lighting normalization improve the performance of the system when we compare the same form of selection. But the choice of a lighting normalization method depends on the environment.

The 1 to N tests (see Table II and Table III) presented the same behaviour of the 1 to 1 tests. As seen in [7], the more images we use in the training, the better the recognition result. As observed in the 1 to 1 tests, the choice of a lighting normalization method also depends on the environment. These conclusions apply when we compare results from the same image selection.

The obtained results show that when we use only information from the faces the SDRF performance decays, thus, we

TABLE III
RECOGNITION RATE WITH TRAINING WITH ONE ELEMENT

	GTAV	PRIVATE	YALE A	YALE B	ALL
Original	86%	100%	90.9%	99.1%	-
Normalized (1st selection)	54.6%	56.4%	70.7%	74.1%	57.3%
Log (1st selection)	61.6%	57.7%	77.2%	70.4%	62.7%
Logabout (1st selection)	68%	61.5%	79.2%	73.7%	69.2%
Equalization (1st selection)	51.7%	59%	75.3%	76.2%	56.7%
Local Equalization (1st selection)	54%	58%	71.4%	70.8%	55.4%
Retinex (1st selection)	60.4%	60%	76%	65.4%	62.2%
Retinex + expansion (1st selection)	54.6%	48.7%	69.4%	64.6%	53.6%
Normalization (2nd selection)	34.8%	42.3%	57.8%	45%	37%
Log (2nd selection)	23.8%	-	51.3%	35.4%	27.7%
Logabout (2nd selection)	24.4%	27%	53.2%	48.3%	31.5%
Equalization (2nd selection)	35.4%	34.6%	57.8%	56.2%	38.5%
Local Equalization (2nd selection)	26.1%	39.7%	51.9%	55.4%	34.6%
Retinex (2nd selection)	34.3%	30.7%	52%	40.8%	31.5%
Retinex + Expansion (2nd selection)	15.7%	24.3%	41.5%	27.5%	18%

conclude that [6], [7] and [9] use background information in the feature extraction process and hence have a better performance. The tested databases have the same conditions for different samples of a same subject, these information end up being used by the referenced recognition systems (unknown to it) to improve the recognition performance.

We conclude that the SDRF is not ready to real case scenarios. Nevertheless, it is a beta system and it can be improved. One possible improvement is to test other classification methods. More importantly, the tests on the SDRF point out that we can use lighting normalization methods to improve the recognition rate of a face recognition system.

For future work, we will experiment the lighting normalization methods on other face recognition systems in order to draw a better conclusion about its impact on the recognition rates.

ACKNOWLEDGMENT

We would like to thank CNPq for the financial support received.

REFERENCES

- [1] A. Jain, L. Hong, and S. Pankanti, "Biometric identification," *Commun. ACM*, vol. 43, no. 2, pp. 90–98, Feb. 2000. [Online]. Available: <http://doi.acm.org/10.1145/328236.328110>
- [2] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Comput. Surv.*, vol. 35, no. 4, pp. 399–458, Dec. 2003. [Online]. Available: <http://doi.acm.org/10.1145/954339.954342>
- [3] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, 2nd ed. Springer Publishing Company, Incorporated, 2009.
- [4] A. A. Ross, K. Nandakumar, and A. K. Jain, *Handbook of Multibiometrics (International Series on Biometrics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
- [5] A. F. Abate, M. Nappi, D. Riccio, and G. Sabatino, "2d and 3d face recognition: A survey," *Pattern Recogn. Lett.*, vol. 28, no. 14, pp. 1885–1906, Oct. 2007. [Online]. Available: <http://dx.doi.org/10.1016/j.patrec.2006.12.018>
- [6] F. M. d. S. Matos, "Reconhecimento de faces utilizando seleção de coeficientes da transformada cosseno discreta," Master's thesis, Universidade Federal da Paraíba, July 2008.
- [7] D. Omaia, "Um sistema para detecção e reconhecimento de face em vídeo utilizando a transformada cosseno discreta," Master's thesis, Universidade Federal da Paraíba, 2009.
- [8] S. A. Khayam, "The discrete cosine transform (dct): Theory and application. department of electrical and computing engineering," 2003.
- [9] A. d. S. Marinho, "Uma nova versão de um sistema de detecção e reconhecimento de face utilizando a transformada cosseno discreta," Master's thesis, Universidade Federal da Paraíba, July 2012.
- [10] Z. M. Hafed and M. D. Levine, "Face recognition using the discrete cosine transform," *Int. J. Comput. Vision*, vol. 43, no. 3, pp. 167–188, Jul. 2001. [Online]. Available: <http://dx.doi.org/10.1023/A:1011183429707>
- [11] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models—their training and application," *Comput. Vis. Image Underst.*, vol. 61, no. 1, pp. 38–59, Jan. 1995. [Online]. Available: <http://dx.doi.org/10.1006/cviu.1995.1004>
- [12] O. Marques Filho and H. Vieira Neto, *Processamento Digital de Imagens*. Rio de Janeiro, Brazil: Editora Brasport, 1999.
- [13] H. Liu, W. Gao, J. Miao, and J. Li, "A novel method to compensate variety of illumination in face detection." in *JCIS*, H. J. Caulfield, S.-H. Chen, H.-D. Cheng, R. J. Duro, V. Honavar, E. E. Kerre, M. Lu, M. G. Romay, T. K. Shih, D. Ventura, P. P. Wang, and Y. Yang, Eds. JCIS / Association for Intelligent Machinery, Inc., 2002, pp. 692–695. [Online]. Available: <http://dblp.uni-trier.de/db/conf/jcis/jcis2002.html#LiuGML02>
- [14] G. Hines, Z. ur Rahman, and G. Woodell, "Single-scale retinex using digital signal processors," in *Global Signal Processing Conference*, 2004.
- [15] F. Tarrs and A. Rama, "Gtav face database," [Online; accessed 25-May-2014]. [Online]. Available: <http://gps-tsc.upc.es/GTAV/ResearchAreas/UPCFaceDatabase/GTAVFaceDatabase.htm>
- [16] A. S. Georgiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001. [Online]. Available: <http://dx.doi.org/10.1109/34.927464>

Estudo da Viabilidade de Utilização de Placa ARM em um Sistema de Inspeção Óptica em Linha para a Produção Têxtil

Maurício Edgar Stivanello, Saulo Vargas
DAMM - Departamento de Metal-Mecânica
IFSC - Instituto Federal de Santa Catarina
Florianópolis, Brasil

E-mail: {mauricio.stivanello, saulo.vargas}@ifsc.edu.br

Juliano Emir Nunes Masson
Departamento de Automação e Sistemas
UFSC - Universidade Federal de Santa Catarina
Blumenau, Brasil

E-mail: juliano.masson@grad.ufsc.br

Abstract—The ARM architecture was originally popularized by its use on mobile devices, and now is being also used in different applications due to advantages such as low cost, small size, low power consumption, and competitive performance. The objective of this study is to evaluate the feasibility of using an ARM board for industrial applications involving complex tasks such as inspection by computer vision. Therefore, it was used as use case the automation of a step of inspection in a production line of textile industry. The given results demonstrate the feasibility of using the selected ARM board for the selected application. Due to the easy implementation and the obtained results it is concluded that this type of computer presents an interesting cost-benefit for industrial applications where the time requirement is met.

Keywords—ARM, automated inspection; computer vision, textile

Resumo—A arquitetura ARM, popularizada inicialmente por sua utilização em dispositivos móveis, tem começado a ser utilizada em diferentes aplicações em função de vantagens como baixo custo, tamanho reduzido, baixo consumo de energia, além de performance competitiva. O objetivo do presente trabalho é avaliar a viabilidade de utilização de uma placa ARM para aplicações industriais envolvendo tarefas complexas como inspeção através de visão computacional. Para isso, foi utilizado como caso de uso a automação da etapa de inspeção de tecidos em uma linha de produção da indústria têxtil. Os resultados apresentados demonstram a viabilidade de utilização da placa ARM avaliada na aplicação selecionada. Pela facilidade de desenvolvimento e pelos resultados obtidos conclui-se que este tipo de computador apresenta um interessante custo-benefício para aplicações industriais onde o requisito de tempo seja atendido.

I. INTRODUÇÃO

Uma verdadeira batalha foi travada nos anos 80 entre processadores RISC (Computador com um Conjunto Reduzido de Instruções) e CISC (Computador com um Conjunto Complexo de Instruções). Neste período a complexidade de circuitos eletrônicos miniaturizados e processadores era uma das principais restrições de projeto, e o panorama de computação era dominado por computadores de mesa e servidores [1].

Hoje, questões de energia e performance surgem como uma das principais restrições de projeto em um cenário muito diferente, incluindo computadores de diferentes portes, dispositivos móveis como *tables* e *smarthphones*. A utilização de

processadores ARM (RISC ISA), popularizados inicialmente por sua vasta utilização em dispositivos móveis, tem se expandido agora para áreas de aplicação antes dominadas pela arquitetura x86 (CISC ISA), como é o caso de virtualização, por exemplo [1].

A mesma disputa deve surgir na área de automação industrial e robótica, onde as aplicações precisam atender requisitos cada vez mais complexos e com alta integração, exigindo recursos computacionais não mais atendidos por microcontroladores ou arquiteturas tradicionalmente utilizadas. Neste domínio de aplicação, as placas ARM ou mini-pcs, recentemente popularizadas, tornam-se cada vez mais interessantes frente aos tradicionais computadores industriais, tendo em vista fatores como custo, tamanho reduzido, baixo consumo de energia, além de performance e conectividade cada vez mais competitivos [2], [3].

No presente trabalho é apresentado um estudo de viabilidade de utilização de uma placa ARM para uma aplicação industrial do ramo têxtil. O objetivo do estudo é avaliar a possibilidade de utilizar esta plataforma para embarcar um Sistema de Inspeção Óptica, que deve rodar em linha na produção de tecidos identificando defeitos de diversos tipos através de visão computacional.

Na seção II é apresentada uma descrição geral da arquitetura de placas ARM e suas aplicações. Na seção III é discutido o caso de uso selecionado para a avaliação, assim como é apresentada a técnica de automação através de visão computacional implementada para as avaliações realizadas. Na seção IV são discutidos os detalhes de implementação. Na seção V são apresentados e discutidos os resultados obtidos. Por fim, na seção VI são apresentadas as conclusões do trabalho.

II. PLACAS ARM E SUAS APLICAÇÕES

O termo ARM corresponde a uma família de arquiteturas para processadores de computador baseadas em um conjunto reduzido de instruções (RISC). Por serem baseadas em RISC, os processadores ARM requerem significativamente menos transistores do que os típicos processadores x86 baseados em CISC. O seu design simplificado facilita a utilização de

processadores com maior quantidade de núcleos a um baixo custo, menor consumo de energia e tamanho reduzido [1]. Os processadores ARM podem variar em performance, energia e preço, com frequências de 15MHz a mais de 1GHz.

Popularizados inicialmente pela ampla utilização em produtos manufaturados como *smartphones*, estes processadores estão sendo empregados nas mais diversas aplicações. Sua popularização nas áreas de automação e robótica tem sido impulsionada por sua disponibilização integrada a mini-pcs, ou placas integradas ARM (*Single Board Computers*). Um mini-pc é um computador completo construído em uma única placa de circuito onde estão o processador, memória RAM, dispositivos de entrada e saída e uma série de recursos como rede Wi-Fi, Bluetooth, GPS, dentre outros. Estas placas são completas, no sentido de que possuem todos os dispositivos e conectores necessários, e muitas vezes são comercializadas com um sistema operacional pré-instalado.

Diversos sistemas operacionais estão disponíveis para mini-pcs baseados na arquitetura ARM, incluindo diferentes versões de Linux, Android, dentre outros.

Tendo em vista as vantagens já descritas, somadas ao surgimento de uma série de kits de desenvolvimento e a facilidade de criação e implantação de aplicações, este tipo de plataforma tem sido cada vez mais utilizado em robótica [3], televisão digital [4], aplicações médicas [5], automóveis [6], domótica [7], dentre outros exemplos.

III. ESTUDO DE CASO: AUTOMAÇÃO DA ETAPA DE INSPEÇÃO DE TECIDOS ATRAVÉS DE VISÃO COMPUTACIONAL

A avaliação realizada no presente trabalho foi baseada na automação da etapa de inspeção presente na produção de tecidos na indústria têxtil. Uma série de defeitos podem surgir nos processos de fiação e tecelagem, como passamento errado de malha, furos, nós, dentre outros ilustrados na Figura 1. É essencial que estes defeitos sejam identificados sem que um material defeituoso seja repassado aos processos de maior valor agregado.

O procedimento tradicional de inspeção de tecidos ocorre retirando-se os rolos fabricados da máquina de tecelagem e desenrolando-os sobre uma mesa de inspeção a uma velocidade relativamente alta. Quando o inspetor humano percebe um defeito no tecido, ele para o motor que move o rolo, registra o defeito e sua localização, e inicia novamente o motor. Esta tarefa se mostra extremamente repetitiva e propensa a erros.

Visto que a presença dos defeitos a serem identificados provoca alterações visuais na superfície do tecido produzido, esta etapa pode ser automatizada através do uso de um sistema de inspeção por meios ópticos, ilustrado na Figura 2. Para atender ao requisito de inspeção existente é proposto o uso de visão computacional clássica, composto por um sistema de aquisição e um sistema de processamento [8]. O cenário de implantação é ilustrado na Figura 2.

Neste sistema é utilizado um programa de processamento de imagens que implementa um método estatístico por análise

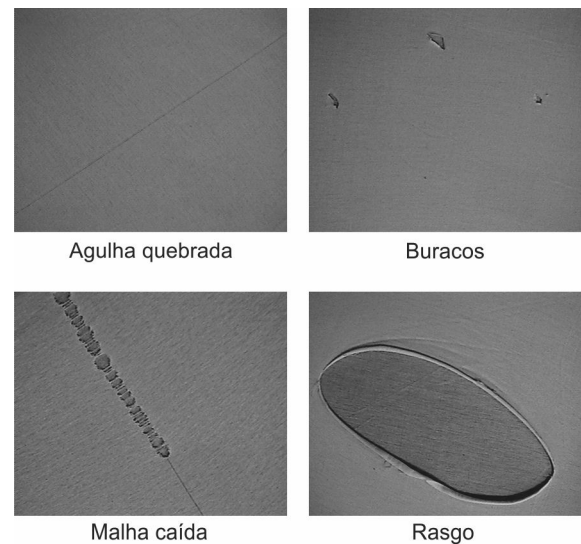


Figura 1. Defeitos em tecidos

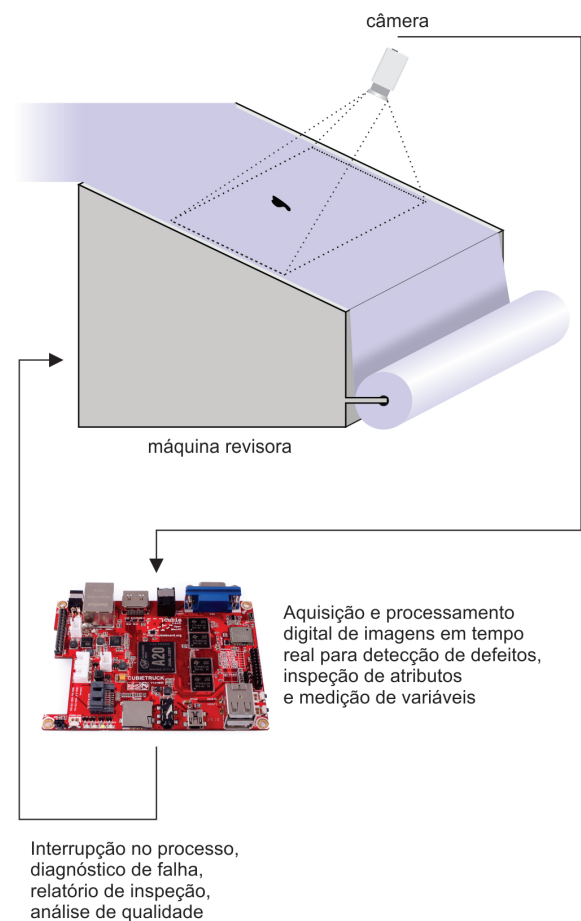


Figura 2. Ilustração da implantação em máquina revisora

de componentes conexos, descrito pelas etapas apresentadas na Figura 3.

Na etapa de *aquisição* é realizada a captura de imagens digitais dos tecidos de interesse. Ainda nesta etapa, podem

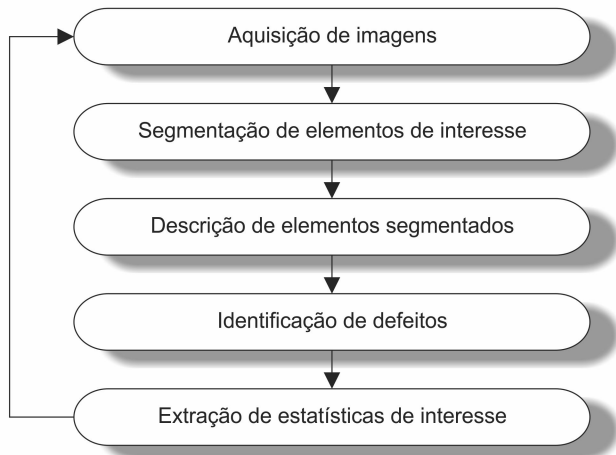


Figura 3. Fluxo de processamento proposto para identificação automática de defeitos

ser adicionados pré-processamentos, como a aplicação do filtro da mediana, por exemplo, para eliminar possíveis ruídos existentes.

Na etapa de *segmentação* é realizada a separação dos elementos de interesse da cena. Neste momento é necessário separar nas imagens a fase correspondente à textura original do tecido da fase correspondente aos defeitos presentes na amostra. Para isso, inicialmente é aplicada uma filtragem passa banda com o objetivo de remover a textura repetitiva obtida em função do entrelaçamento dos fios do tecido. Em seguida é empregada a técnica de limiarização seguida por uma sequência de filtros morfológicos.

Na etapa de *descrição* cada um dos elementos restantes na imagem são detectados através de um algoritmo de rotulação de componentes conexos. Neste momento é gerada uma lista que contém a posição de cada elemento, além da área calculada com base no número de pontos pertencentes ao mesmo.

Por sua vez, na etapa de *identificação* os defeitos são encontrados de forma direta, avaliando-se a área de cada elemento e desconsiderando aqueles com valores fora de uma faixa aceitável. Finalmente, a partir das informações dos defeitos devidamente identificados e descritos é possível levantar estatísticas ou mesmo enviar comandos a sistemas atuadores integrados à linha de produção.

Na Figura 4 são apresentadas as imagens obtidas nas principais etapas de processamento empregando o sistema implementado na análise de uma dada amostra de tecido.

IV. IMPLEMENTAÇÃO

O sistema de visão descrito na seção III foi implementado em duas plataformas, aqui denominadas como COMP-x86 e COMP-ARM. COMP-x86 corresponde a um computador padrão baseado na arquitetura x86. COMP-ARM corresponde a um mini-computador baseado na arquitetura ARM com dimensões de 8 X 11 cm. A descrição detalhada dos computadores utilizados pode ser visualizada na Tabela I.

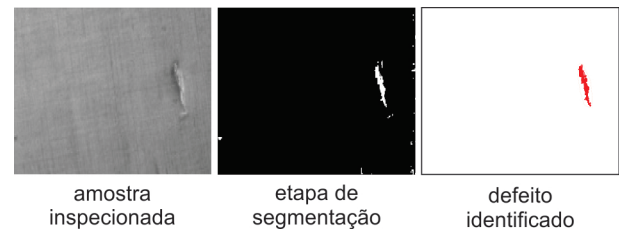


Figura 4. Ilustração do processamento empregado

Tabela I
DESCRIÇÃO DOS COMPUTADORES UTILIZADOS

Item	COMP-ARM	COMP-x86
Modelo	CubieBoard 3	Dell Latitude 3440
Processador	ARM Cortex A7 Dual-Core 1.00GHz	Intel Core i7 4500U 1.80GHz
Memória	2 GB	8GB
SO - Distribuição	Debian GNU/Linux 7.1	Ubuntu 14.04 LTS
SO - Kernel	Linux 3.4.75-sun7i	Linux 3.13.0-27

Para cada uma destas plataformas foram criados os binários da aplicação descrita na Seção III, a partir do mesmo código-fonte, empregando o compilador GCC 4.8 e a biblioteca de visão computacional OpenCv 2.4.

A aquisição das imagens utilizadas no processamento foi realizada por uma câmera digital Integriss modelo WC360 de 1.3MP, utilizando a interface USB.

V. EXPERIMENTOS, RESULTADOS E DISCUSSÃO

Utilizando as plataformas descritas na seção IV, os tempos de execução de uma série de tarefas foram estimados, de modo a subsidiar a avaliação frente ao domínio de aplicação. Tanto os resultados como a discussão são apresentados na sequência.

Execução da Aplicação de Visão Computacional: Inicialmente, uma bateria de testes foi realizada utilizando as plataformas testadas com o objetivo de estimar o tempo necessário para a execução do ciclo de inspeção descrito na seção III.

Na Tabela II são apresentados os tempos médios obtidos na inspeção de 1000 quadros.

Tabela II
TEMPOS MÉDIOS DE PROCESSAMENTO - APLICAÇÃO (ms)

Etapa	COMP-ARM	COMP-x86
Aquisição e pré-processamento	39,27	8,24
Segmentação	3,17	0,10
Descrição, identificação e extração de dados	24,52	2,09
Fluxo geral	66,96	10,43

Avaliação de Desempenho através de Ferramenta de Benchmark: Para subsidiar a avaliação, também foram estimados os tempos de execução de tarefas genéricas através da ferramenta de *Benchmark* HardInfo. Esta ferramenta é comumente utilizada para levantamento de performance e perfil em sistemas Linux.

Na Figura III são apresentados os resultados obtidos para algumas das tarefas avaliadas pela ferramenta.

Tabela III
TEMPOS DE PROCESSAMENTO (s) - HARDINFO

Teste	COMP-x86	COMP-ARM
CPU Blowfish	4,33	55,64
CPU Fibonacci	1,39	15,80
FPU FFT	0,852	73,94
FPU Raytracing	3,779	70,71

Avaliação Geral dos Resultados: De mãos dos tempos de processamento consumidos para a realização das várias tarefas executadas, podemos avaliar a viabilidade de aplicação da plataforma ARM frente ao domínio de aplicação. O sistema proposto, descrito pelo ciclo ilustrado na Figura 3 será integrado a uma máquina revisora ou tear em uso na indústria, conforme ilustrado na Figura 2, de modo a automatizar a etapa de inspeção de defeitos.

Se considerarmos a utilização de câmeras em uma configuração cujo campo de visão seja de 40 cm no sentido longitudinal do rolo de tecido sendo inspecionado, e considerando que o tempo necessário para a realização de todo o ciclo de inspeção pela plataforma COMP-ARM foi de 66,96 ms, temos que o sistema pode operar a no máximo 14 Hz. Nesta taxa de operação, o sistema pode inspecionar até 6 metros de tecido por segundo. As máquinas de inspeção de malha manual operam a velocidades entre 8 a 20 metros por minuto [9]. Desta forma, os resultados obtidos permitiriam ainda a utilização de várias câmeras no sistema posicionadas no sentido transversal do tecido de modo a cobrir larguras maiores do que o campo de visão de uma única câmera, mantendo ainda uma velocidade de inspeção superior quando comparada à operação manual.

É importante observar que nestas mesmas condições, a plataforma COMP-x86, cujo tempo necessário para a realização do ciclo de inspeção é de 10,43 ms, poderia operar a uma taxa de até 95 Hz. Mesmo diante do fato de que o tempo de processamento de diferentes etapas da aplicação implementada tenha sido consideravelmente mais lenta quando executadas no sistema ARM, considera-se que pelo seu tamanho reduzido, custo, baixo consumo de energia, facilidade de uso e pela presença de uma série de interfaces adequadas a integração com sensores, atuadores, e outros dispositivos, o seu emprego apresenta um excelente custo-benefício.

É importante considerar que as operações de processamento de imagens presentes em um sistema de visão computacional podem variar muito em complexidade, em função do problema a ser tratado. Se observarmos os resultados obtidos através da ferramenta HardInfo, especificamente no que se refere a Transformada Rápida de Fourier (FFT), comumente utilizada no processamento de imagens no domínio da frequência, podemos concluir que requisitos de tempo mais rígidos não serão atendidos se não forem explorados outros recursos de hardware.

VI. CONCLUSÕES

No presente trabalho foi apresentado um estudo de viabilidade de utilização de uma placa ARM para implementação de um sistema de inspeção automatizada de tecidos. Os resultados obtidos pela execução da aplicação de visão computacional em plataformas de arquitetura ARM e x86 são estimados e comparados. Em seguida, uma avaliação dos resultados levantados são avaliados frente as restrições do domínio da aplicação.

Os resultados obtidos na avaliação são promissores, ao permitir que o sistema inspecione até 6 metros de tecido por segundo. Mesmo diante do fato de que a taxa de operação do sistema pela plataforma ARM tenha sido consideravelmente mais lenta quando comparada com a plataforma x86, considera-se que pelo seu tamanho reduzido, custo, baixo consumo de energia, interfaces de comunicação e facilidade de uso o seu emprego apresenta um excelente custo-benefício para aplicações de automação.

Como trabalho futuro podemos citar a exploração de diferentes recursos de hardware da plataforma e a utilização de um sistema operacional de tempo real de modo a reduzir o custo computacional. Além disso, uma avaliação de outros aspectos da aplicação deste tipo de computador em aplicações industriais, considerando fatores como interferência eletromagnética e temperatura associadas ao meio de operação, será bastante adequada.

AGRADECIMENTOS

Os autores gostariam de agradecer ao CNPq pelo suporte financeiro provido para a realização do projeto na qual este estudo foi realizado.

REFERÊNCIAS

- [1] P. Varanasi and G. Heiser, "Hardware-supported virtualization on arm," in *Proceedings of the Second Asia-Pacific Workshop on Systems*, 2011.
- [2] N. Alee, M. Rahman, and R. Ahmad, "Performance comparison of single board computer: A case study of kernel on arm architecture," in *Computer Science Education (ICCSE), 2011 6th International Conference on*, 2011.
- [3] M. Bin Ayob and M. Zakaria, "3wd omni-wheeled mobile robot using arm processor for line following application," in *Industrial Electronics and Applications (ISIEA), 2011 IEEE Symposium on*, Sept 2011, pp. 410–414.
- [4] W. Junhua and G. Hengshan, "Development of digital television network tester based on dual cpu architecture," in *Intelligent Networks and Intelligent Systems, 2009. ICINIS '09. Second International Conference on*, Nov 2009, pp. 8–10.
- [5] T. Chakravorty, D. N. Sonawane, S. Sharma, and T. Patil, "Low-cost subcutaneous vein detection system using arm9 based single board computer," in *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, vol. 2, April 2011, pp. 339–343.
- [6] J. Yang and G. Feng, "Key technology of micro navigation system for automobile based on arm-linux platform," in *Machine Vision and Human-Machine Interface (MVHI), 2010 International Conference on*, April 2010, pp. 428–431.
- [7] Q. Du, H. Qin, J. Tang, and X. Li, "Design of the arm based remote surveillance system," in *System Science, Engineering Design and Manufacturing Informatization (ICSEM), 2012 3rd International Conference on*, vol. 1, Oct 2012, pp. 336–338.
- [8] R. C. Gonzalez and R. E. Woods, *Processamento Digital de Imagens*. Pearson, 2010.
- [9] A. Kumar, "Computer-vision-based fabric defect detection: A survey," *Industrial Electronics, IEEE Transactions on*, vol. 55, no. 1, pp. 348–363, Jan 2008.

Visualizações de Matrizes Origem-Destino no Cenário do Tráfego Urbano

H. W. A. S. Gondim , H. A. D. do Nascimento
Instituto de Informática
Universidade Federal de Goiás, Campus II
Goiânia - Goiás, Brasil
email: {halleygondim, hadn}@inf.ufg.br

D. Reilly
Faculty of Computer Science
Dalhousie University
Halifax - Nova Scotia, Canada
email: reilly@cs.dal.ca

Abstract—Traffic problems are frequent in most large cities, and actions to minimize them have become ever more necessary. This paper discusses and proposes information visualizations for the representation of origin-destination matrices. Despite the importance of such matrices for the study of urban traffic, this is an aspect still little explored by the information visualization area and for which significant contributions can be made.

Keywords-urban traffic, traffic congestion, information visualization, O-D matrix

Abstract—Problemas relacionados ao tráfego urbano são frequentes na maioria das grandes cidades, sendo que ações no sentido de minimizar os mesmos têm se tornado cada vez mais necessárias. O presente trabalho discute e propõe visualizações de informação para a representação de matrizes origem-destino. Apesar da importância de tais matrizes para o estudo do tráfego urbano, esse é um elemento ainda pouco explorado pela área de visualização de informações e para o qual contribuições importantes podem ser feitas.

Keywords-tráfego urbano, congestionamento de veículos, visualização da informação, matriz Origem-Destino.

I. INTRODUÇÃO

O congestionamento de veículos e outros problemas de tráfego são recorrentes em grandes centros urbanos [1]. Diversas soluções têm sido propostas para resolver ou atenuar tais problemas, contudo, as pesquisas sobre o trânsito, em sua maioria, demandam esforço e tempo consideráveis, dado o complexo dinamismo que envolve o mesmo [1], [2]. Em um estudo recente, Gondim et al [3] destacaram a importância de utilizar técnicas de Visualização da Informação (VI) para a análise de dados e a tomada de decisões em problemas de tráfego urbano. Entretanto, eles identificaram uma escassez na aplicação de técnicas de VI's em certos problemas de tráfego. Um exemplo dessa escassez, que é o foco do presente trabalho, é na visualização de matrizes O-D, as quais visam, basicamente, representar o volume de deslocamentos de veículos entre zonas de uma região ¹.

Matrizes O-D são de grande importância para estudo do tráfego [4], [5], sendo sua construção prevista, inclusive, no método clássico de Quatro Etapas [6] de engenharia do

tráfego. Porém, a escassez de representações visuais efetivas de tais matrizes deve-se à dificuldade de proporcionar imagens adequadas que abordem os três seguintes aspectos [3], [7]: o volume de veículos de acordo com o horário e o dia da semana (aspecto temporal), os diversos modos de viagem (multimodalidade) e a exibição de muitos dados. A visualização da variação dos dados com tempo tem relação com identificar os padrões de mudanças do tráfego ao longo do dia e da semana. Já uma visualização que contemple os modos de transporte deve destacar os diversos meios de transporte (carro, ônibus, etc.) utilizados para locomoção em uma rede viária. A capacidade de mostrar muito dados mas sem sobrecarga visual, por sua vez, é um aspecto importante para subsidiar a tomada de decisão em diversos problemas. Com o intuito de contribuir para essa área, o presente trabalho propõe um conjunto visualizações de matrizes O-D que auxiliam na percepção da demanda de viagem de veículos com variações de horário, modo de transporte e suporte a maior volume de dados, de maneira intuitiva.

II. TRABALHOS RELACIONADOS

Matrizes Origem-Destino são utilizadas para sintetizar e quantificar a mobilidade associada a pessoas e bens, fornecendo o número de viagens realizadas entre cada zona de origem e destino em um determinado período de tempo [8], [9].

A sua obtenção pode ser oriunda de métodos tradicionais como: pesquisas e entrevistas com motoristas, *car-following* e contagens volumétricas de fluxos nas vias [8], [10], [11], [12].

Comumente, matrizes O-D são apresentadas na forma de uma tabela (veja um exemplo na Tabela I). Nessa visualização, linhas e colunas representam zonas de tráfego e suas interseções contêm a quantidade de veículos que viajam entre as mesmas. Outra visualização comum usa mapas para demonstrar a demanda de deslocamento de veículos entre zonas por meio de arcos ². Contudo, tais representações visuais apresentam problemas na proporção em que o volume de dados aumenta, como: dificuldade na observação dos pares O-D devido ao adensamento de arcos e perda visual de

¹Zona é a definição de uma área, podendo ser de baixa ou alta granularidade (bairros até países). Zonas podem ser definidas por critérios sociais e econômicos ou aglomerações populacionais.

²Essa visualização é frequente na ilustração de linhas de voo por companhias aéreas. Ela também foi usada em outras situações como, por exemplo, para demonstrar chamadas telefônicas entre regiões de um país.

informações pela grande quantidade de linhas e colunas de uma tabela.

Table 1
MATRIZ O-D.

O/D	1	2	...	m	
1	d_{11}	d_{12}	...	d_{1m}	P_1
...
n	d_{n1}	d_{n2}	...	d_{nm}	P_n
	A_1	A_2	...	A_m	A/P

Dois outros trabalhos se destacam na visualização de matrizes O-D. Boyandin et al [7] apresentaram o uso de dois mapas distintos, um para origem e outro para destino, com um *heat map* separando-os. As demandas de pares OD foram expressas por meio da coloração do *heat map*, sendo estes agrupados de acordo com o tempo. Já Gunay et al[13] transformaram informações de matrizes O-D em ondas *wavelets*, conseguindo, com isso, destacar a variação do volume de veículos ao longo da malha viária.

Apesar das contribuições destes trabalhos, informações como multimodalidade e metáforas de fácil percepção não foram exploradas. Na verdade, a falta de outras informações nessas representações visuais e até de novas visualizações para demonstrar matrizes O-D no cenário do tráfego urbano podem ter sua causa na existência de um grande número de atributos associados aos dados. Esses atributos competem pela percepção do observador e dificultam representações visuais mais adequadas. Chen et al [14] e outros autores [15], [16] comentam sobre essas dificuldades na construção de representações visuais em geral.

Na próxima seção, propomos visualizações da informação que consideram variáveis de tempo, modalidade de transporte e exibição de maior volume de dados na representação de matrizes O-D. Obviamente, os problemas de percepção são potencializados quando mais atributos dinâmicos são inseridos simultaneamente na visualização. Tentamos minimizar isso através do uso de um mapeamento visual mais efetivo.

III. VISUALIZAÇÕES PROPOSTAS

As visualizações descritas a seguir combinam técnicas de VI já existentes com diferentes abordagens de mapeamento visual a fim de facilitar a agregação de diversas informações. Elas contemplam os seguintes itens: pontos de origem e respectivos destinos, tipos de transporte envolvidos, horários e dias da semana, capacidade de síntese de grande volume de dados, quantidade de demanda, áreas saturadas, etc. Um resumo do mapeamento visual e do funcionamento de cada visualização é apresentado. Alguns elementos com indicadores de interação foram omitidos a fim de simplificar a visualização como: o uso de barras de rolagem e de recursos de zoom para o controle da variação temporal e de navegação. Itens enumerados, dentro de cada visualização, são exibidos com o intuito de facilitar a explicação de cada detalhe.

A Visualização apresentada na Figura 1-(A) utiliza um gráfico de barras espelhado a fim de refletir a respectiva zona de Origem (Produção) x Destino (Atração) (itens 1 e 2). A

modalidade de veículos pode ser observada em zoom no item 3, onde uma barra menor exibe os tipos de veículos através da coloração ao longo dos horários do dia. Uma *table lens*, item 4, auxilia na exploração de uma grande quantidade de dados, enquanto que interação com a visualização permite que, ao se selecionar uma origem, seus respectivos destinos fiquem em destaque. Este recurso também é recíproco quando se seleciona um destino. E uma outra interação proposta é a navegação pelos dias da semana por meio da barra de tempo localizada na parte superior da visualização. Nas próximas visualizações o recurso de barra de tempo será omitido para simplificação do texto.

A Figura 1-(B) apresenta uma VI clássica de matrizes de origem e destino com o uso do *treemap* como detalhe. A exposição modal e temporal são visualizadas no *treemap*, onde é possível observar os diversos tipos de transporte (item 3), como: veículos de grande porte GP, pequeno porte PP e motociclistas em geral MO, bem como a respectiva demanda ao longo dos dias da semana. A coloração permite destacar o volume dominante de demanda em cada célula (item 2).

A Figura 1-(C) é uma variação da abordagem de Boyandin et al [7] com o seguinte diferencial: a utilização de um grafo bipartido ao centro indicando as ligações entre os pares O-D (item 1). Dois mapas, um origem e o outro o destino, foram utilizados a fim de minimizar o número de cruzamento de linhas (itens 2 e 3). A espessura das linhas e a área dos círculos indicam o volume da demanda. Regiões podem ser selecionadas (item 4) a fim de exibirem seus respectivos pares O-D (itens 5). *Edges bundles* podem ser utilizadas quando a quantidade de pares O-D aumentam.

A Figura 1-(D) utiliza um *diagrama de arcos* para apresentar as demandas entre os pares O-D. A visualização dispõe de todos os pontos O-D ordenados por regiões, representados por círculos, pela coloração e proximidade. A atração e produção de demandas podem ser observadas em um mesmo arco (item 1). Abaixo de cada rótulo, item 2, tem-se a representação modal (item 3) por coloração dos diversos tipos de veículos. A interação permite reordenar regiões, bem como selecionar pontos para destaque dos respectivos pares O-D.

A Figura 1-(E) exibe um *Radial Table* com o uso de *Edge bundles* para sintetizar as ligações entre os pares O-D (item 1). Há uma hierarquia que é apresentada no item 2, sendo a mesma dividida em regiões de alta granularidade, como bairros, até granularidades mais baixas, como cidades. A interação permite selecionar regiões (item 3), para que os tons das linhas mudem indicando a predominância de um determinado tipo de transporte, bem como adição e remoção dos níveis hierárquicos.

A Figura 1-(F) utiliza a técnica *Overview and Detail* com mapas. As origens podem ser identificadas na parte inferior do mapa, em pequenos quadrantes, e os destinos são distribuídos no mapa maior (itens 1). A espessura das linhas indicam a demanda, enquanto que as barras posicionadas abaixo das origens, item 2, representam os diversos meios de transporte. *Edge bundles* podem adicionar uma melhor visualização a medida que o volume de pares O-D aumentem.

Table II
MAPEAMENTOS USADOS NAS VISUALIZAÇÕES.

Vis.	Técnica VI	Tempo	Modo	Volume
(A)	Table lens	barra/rolagem	coloração em barras agregadas	altura da barra
(B)	Treemap e matriz	treemap	treemap	coloração
(C)	Mapas e grafos	barra/rolagem		espessura de linhas/círculos
(D)	Arc Graph	barra/rolagem	coloração em barras agregadas	espessura de linhas/círculos
(E)	Radial Table	barra/rolagem	coloração em área selecionada	espessura de linhas/arcs
(F)	Mapa e grafos.	barra/rolagem	coloração em barras agregadas	espessura de linhas
(G)	Mapas e grafos	barra/rolagem	coloração das linhas	espessura de linhas

Table III
VANTAGENS E DESVANTAGENS DAS VISUALIZAÇÕES.

Vis.	Prós	Contras
(A)	O-D são espelhados.	Pares O-D exibidos apenas se selecionados.
(B)	Todos os pares O-D são exibidos.	Treemap exibido apenas ao selecionar uma célula na tabela.
(C)	Mapas são usados para detalhes de pares O-D.	Não apresenta modalidade
(D)	Regiões saturadas são destacadas facilmente.	Pares O-D destacados apenas se selecionados.
(E)	A visualização apresenta alto poder de sintetização de dados	Os pontos O-D são difíceis de se observar.
(F)	Cruzamento de linhas são minimizados.	Os mini mapas apresentam perda de contexto.
(G)	Cruzamento de linhas são minimizados.	Os pontos O-D são difíceis de observar.

Por fim, a Figura 1-(G) apresenta as relações origem-destino através de dois mapas em perspectiva, sendo um o reflexo do outro. A origem é retratada no mapa inferior, enquanto que o destino posiciona-se no mapa superior. A espessura e a coloração das linhas indicam respectivamente a quantidade de demanda, bem como os diversos tipos de veículos (item 1). Essa abordagem visual foi criada com o intuito de minimizar a sobreposição de linhas quando há muitos pares O-D a serem visualizados.

Um resumo das estratégias usadas nas visualizações é apresentado na Tabela II. Para cada visualização proposta, têm-se os seguintes itens: técnicas de VI's utilizadas; representação temporal (hora, dia e semana); representação de modos de transporte e, por fim, capacidade de destacar grande volume de dados.

De acordo com a Tabela II, as representações foram satisfatórias, pois atenderam boa parte dos aspectos (modo, tempo e exibição massiva de dados) críticos.

Os prós e contras de cada visualização são evidenciados na Tabela III.

Apesar de uma variedade de representações propostas não

temos uma definição de qual visualização é melhor. Isso depende de quais dados e quais aspectos de interesse serão considerados.

IV. CONCLUSÃO

As representações propostas utilizaram diversos mapeamentos e técnicas de VI. Através das mesmas é possível observar vários tipos de informação como: as zonas de origem e seus respectivos destinos, zonas que possuem maior e menor demanda de tráfego, horários de saturação do tráfego e principais meios de transportes utilizados em cada zona.

No momento, estamos implementando as visualizações usando tecnologias como Three.js, d3.js e JavaScript e planejando a validação das mesmas com usuários. Estamos também projetando novas visualizações que consideram outros tipos de dados associados a matrizes O-D.

REFERENCES

- [1] R. Berkowicz, O. Hertel, S. E. Larsen, N. N. Sorensen, and M. Nielsen, "Modelling traffic pollution in streets," *Roskilde, Ministry of Environment and Energy, National Environmental Research Institute*, 1997.
- [2] K. J. Button, *Handbook of transport systems and traffic control*. Emerald Group Publishing, 2001, vol. 3.
- [3] H. A. D. d. N. H. W. A. S. Gondim and D. Reilly, "Visualizing large scale vehicle traffic network data: a survey of the state-of-the-art," *International Conference on Information Visualization Theory and Applications (IVAPP 2014)*, 2014.
- [4] H. Dezani, R. D. Bassi, N. Marranghello, L. Gomes, F. Damiani, and I. Nunes da Silva, "Optimizing urban traffic flow using genetic algorithm with petri net analysis as fitness function," *Neurocomputing*, vol. 124, pp. 162–167, 2014.
- [5] Y. N. Wang and B. F. Si, "Estimation of origin-destination demand matrix for the urban multimodal traffic network," *Applied Mechanics and Materials*, vol. 511, pp. 963–970, 2014.
- [6] M. G. McNally, "The four-step model," *Handbook of transport modelling*, vol. 1, pp. 35–41, 2000.
- [7] I. Boyandin, E. Bertini, P. Bak, and D. Lalanne, "Flowstrates: An approach for visual exploration of temporal origin-destination data," in *Computer Graphics Forum*, vol. 30, no. 3. Wiley Online Library, 2011, pp. 971–980.
- [8] S. Bera and K. Rao, "Estimation of origin-destination matrix from traffic counts: the state of the art," 2011.
- [9] T. Pohlmann and B. Friedrich, "A combined method to forecast and estimate traffic demand in urban networks," *Transportation Research Part C: Emerging Technologies*, vol. 31, pp. 131–144, 2013.
- [10] E. Cascetta, "Estimation of trip matrices from traffic counts and survey data: A generalized least squares estimator," *Transportation Research Part B: Methodological*, vol. 18, no. 4, pp. 289–299, 1984.
- [11] H. D. Sherali, A. Narayanan, and R. Sivanandan, "Estimation of origin-destination trip-tables based on a partial set of traffic link volumes," *Transportation Research Part B: Methodological*, vol. 37, no. 9, pp. 815–836, 2003.
- [12] C. Xie, K. M. Kockelman, and S. T. Waller, "Maximum entropy method for subnetwork origin-destination trip matrix estimation," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2196, no. 1, pp. 111–119, 2010.
- [13] B. Gunay, "Using wavelet transforms for better interpretation of traffic simulation," *Traffic Engineering & Control*, vol. 50, no. 10, 2009.
- [14] C. Chen, "Top 10 unsolved information visualization problems," *Computer Graphics and Applications, IEEE*, vol. 25, no. 4, pp. 12–16, 2005.
- [15] C. R. Johnson and A. Sanderson, "A next step: Visualizing errors and uncertainty," *Computer Graphics and Applications, IEEE*, vol. 23, no. 5, pp. 6–10, 2003.
- [16] C. Johnson, "Top scientific visualization research problems," *Computer graphics and applications, IEEE*, vol. 24, no. 4, pp. 13–17, 2004.

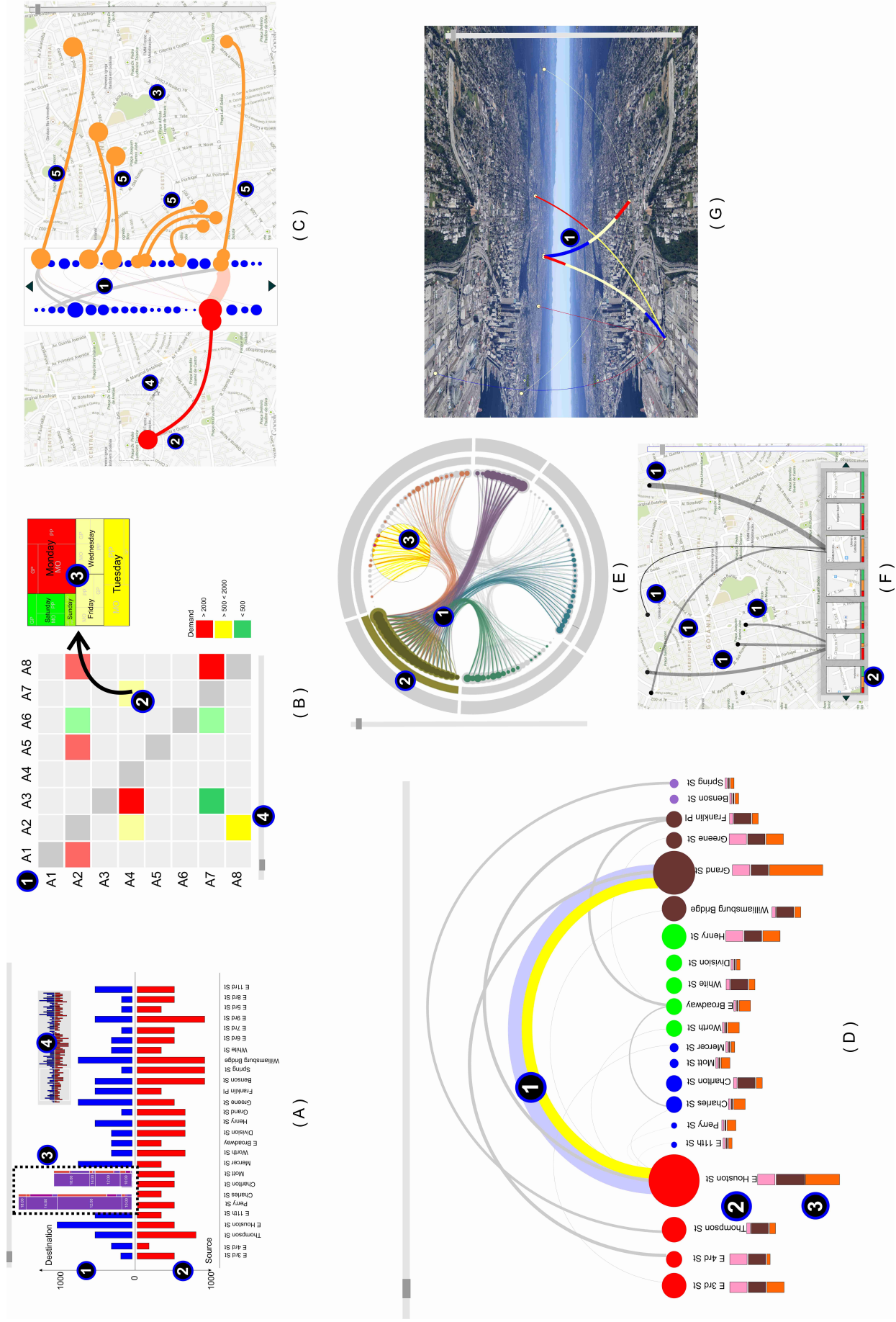


Figure 1. Visualizações - Quadro 1.

Extracting and Rendering Primitives from Point Clouds

Thiago Barroso Perrotta, Daniel Pinto Coutinho, Ricardo Guerra Marroquim
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
email: perrotta.thiago@poli.ufrj.br, {dpcoutinho, marroquim}@cos.ufrj.br

Abstract—In this work we provide a method to extract and render geometric primitives from point clouds, without the need to convert to polygon meshes. The extraction is realized in two steps: first the basic shapes are detected, then their boundaries are analyzed and stored in a convenient spatial data-structure. Finally, a ray-tracer is employed to directly visualize the extracted primitives.

Keywords—point cloud; primitive detection; ray tracing

I. INTRODUCTION

Due to the popularization of acquisition devices and techniques, such as 3D scanners and photogrammetry, point clouds are becoming a common representation in many application domains. Their main advantage is the simplicity of the data, since no adjacency information is stored. However, this lack of connectivity also renders point clouds challenging in some important aspects, such as reconstruction and visualization.

An important challenge in this field is the extraction of structured information from the point sets. Even though scanners are able to produce datasets in the order of billions of points, often a more compact representation is possible and desired. For example, the set of points representing a plane can be highly redundant, and may be traded for a simple definition of the plane and its physical boundaries.

Another matter with this representations is visualization, since graphics hardware are tailored to deal with triangle meshes. Directly rendering point clouds has also been the focus of many research efforts, to avoid converting the point set to a mesh for example. Nevertheless, our interest lies on the direct visualization of the extracted primitives.

In this work, we propose a primitive extractor from point clouds, and a coupled ray-tracer to directly render our compact structure. An important step in our algorithm is the detection of geometric primitives, such as planes, spheres, cones, cylinders and tori. Having extracted these primitives we delimit their shapes and extract their boundaries to respect as closely as possible the point set. The shape representation is directly visualized using a ray-tracer, to avoid conversions.

II. PRIMITIVE EXTRACTION FROM POINT CLOUDS

Our algorithm is based on the Random Sample Consensus (RANSAC) paradigm [1], a way to fit a model into experimental data. Traditional methods for model fitting try to use most of the available points, where the RANSAC paradigm works in

the opposite way, using only the minimum number of points to describe a model (ex. three points completely define a plane).

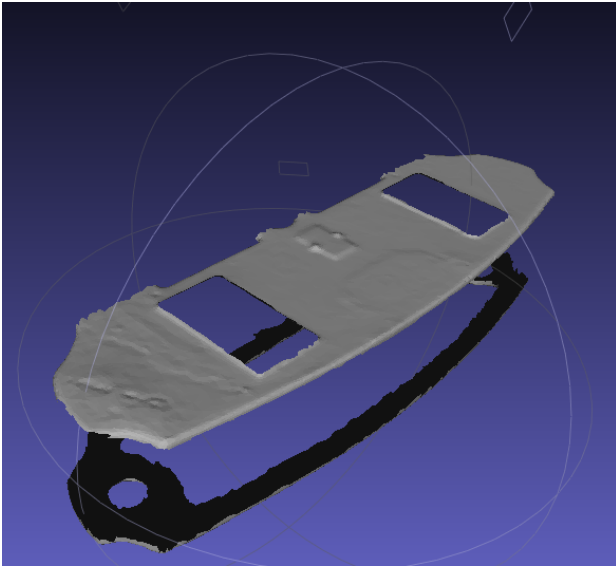
Briefly, the procedure works in the following way: given a number of points P and a minimal number of points n that describe a model, we randomly select a subset of n points from P and create a model M_1 . Subsequently, we validate M_1 by checking how many n_1 points from P are within an error tolerance ϵ to M_1 . If n_1 is larger than a given threshold t , we accept M_1 and use the selected n_1 points to calculate a new refined model M_1^* . Otherwise, we start the process again by choosing new random points and creating a new model. If, after some trials, no better consensus has been found, we either choose the one with largest consensus during previous iterations or we terminate the algorithm with failure (in case there was no large enough consensus).

A. Detecting shapes

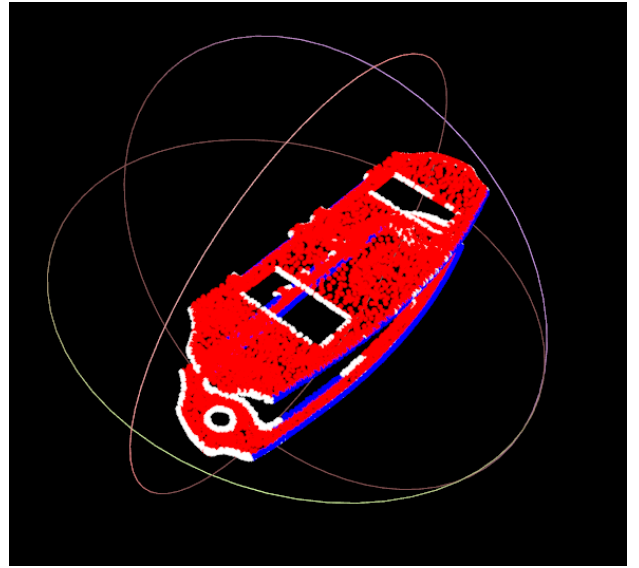
We detect shapes in points clouds by using the algorithm developed by Schabel [2]. The algorithm works by using the RANSAC procedure where in each iteration a minimal number of points is randomly selected to create one of the possible primitives: sphere, plane, torus, cylinder and cones. When a model of a primitive is accepted, the candidates that fall on the primitive are extracted from the point cloud. The method continues until there are no points left, or no primitive can be extracted given the thresholds. Fig. 1 shows an example of an input point cloud to the algorithm.

Even though this basic method is sound, without further optimizations it suffers heavily of performance issues. However, under the assumption that shapes are a local phenomena, we do not need to create a minimal set by choosing randomly from all the points, we choose points that are spatially close. Exploiting this assumption, we use an octree to establish spatial proximity between points, so we only form a candidate set from points of the same cell.

Every time a primitive is extracted we perform a refitting step using non-linear least squares. Using Newton's algorithm, we refine the extracted primitive model by using all points that were considered to belong to that primitive, and not only the initial random candidates. Finally, we further refine our set of points that belong to the primitive as those that lie at most at a distance of 3ϵ , where ϵ is the distance threshold.



(a) Input mesh.



(b) Detected primitives.

Fig. 1: A real case example, from the top and bottom sections of a scanned mechanical instrument. Points in red belong to planes; points in blue belong to cylinders; and white points were not assigned to any primitive. Even though the input is noisy, our extractor was able to retrieve most primitives.

B. Detecting boundaries

Mathematically, most of our primitives are infinite in space. However, to be able to effectively extract the model represented by the point set, we need to know their limits, e.g., the boundaries of plane in our points cloud. An assumption we made is that all of our inputs are closed models, meaning that every primitive is enclosed by others. From this assumption we can infer that the intersection points between the primitives are part of their boundary. Even so, a shape may still contain "holes", see Fig. 1a. Therefore, we not only need to detect outer boundaries, but possibly inner boundaries as well. We approach this issue by doing the following:

- Calculating the intersection points;
- Classifying the region in space of one primitive in respect to the other primitives.

Since we are dealing with any given type of intersection, many can be very challenging to solve analytically, such as the torus-torus intersection. Thus, we create a spatial data structure in order to detect the intersection points. More specifically, a KD-Tree is generated where a subdivision occurs every time a node contains more than one primitive. This process continues until the cell is small enough so that its center might be considered as a close approximation to the intersection point between the shapes.

Finally, we classify the region of one primitive. The idea behind our approach derives from the work of Tilove [3] and the main point is to create a series of membership classification (*MC*) functions for each primitive. Let P_i and P_j be primitives, then we can define a MC function $MC(P_i, P_j)$ which "classifies" P_i with respect to P_j . For example, if P_i is a cylinder and P_j is a plane, there are three possible

classification of the cylinder with respect to the plane:

- The cylinder lies in front of the plane (according to the plane's normal);
- The cylinder lies behind the plane;
- The plane passes through the cylinder.

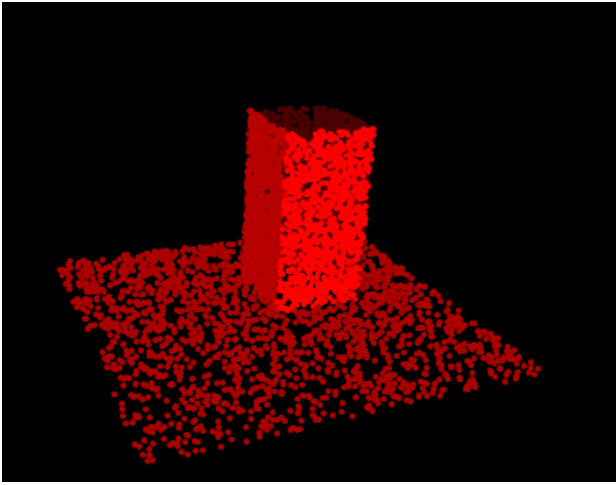
Note that in this last case, we can say that the cylinder is both in front and behind the plane. We follow this last approach in our work.

So, for each primitive we create a CSG tree where each node represents a classification function of the current primitive and another detected one. Each time we create a new node to add in the tree we classify our current primitive with respect to the new one generating two more nodes. Clearly, the height of our tree is $O(n)$, where n is the number of primitives.

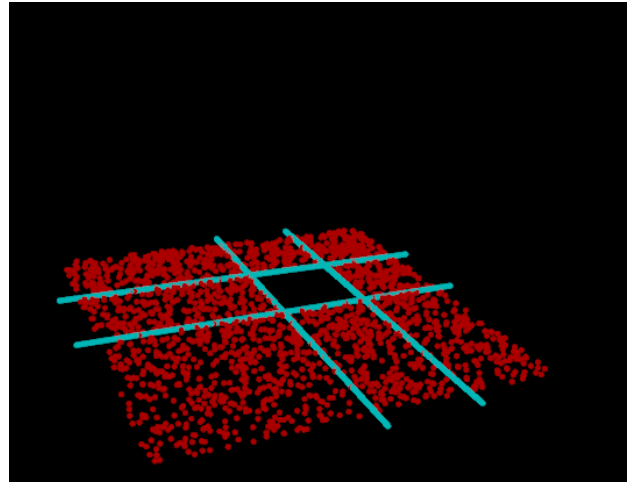
In Fig. 2 depicts a simple case to illustrate the method. We can see the lines of intersection points resulting from each vertical plane. These lines illustrate the Membership Classification function as well. Each vertical plane divides the bottom plane in two regions, or patches. Taking into account all four vertical planes, a total of eight patches is generated, each of these representing a leaf node in the CSG tree. All plane's points of interest are contained in these patches, so we use this information in the Ray Tracing algorithm.

In Fig. 3 an example of a cube is shown. In this case we can clearly see the intersection points delimiting the region of the plane.

Since each primitive requires a very reduced set of parameters to define it, a CSG tree of primitives is indeed very low memory consuming, even in the face of a large number of shapes. With our representation a primitive can be completely described by its boundary points and its CSG tree.

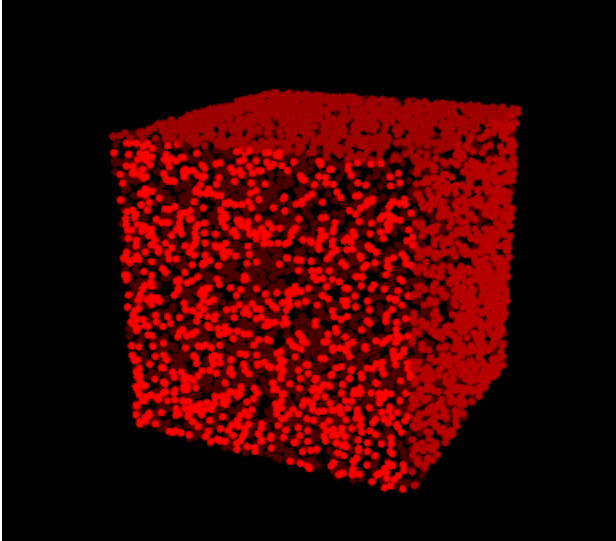


(a) Point cloud containing five planes.

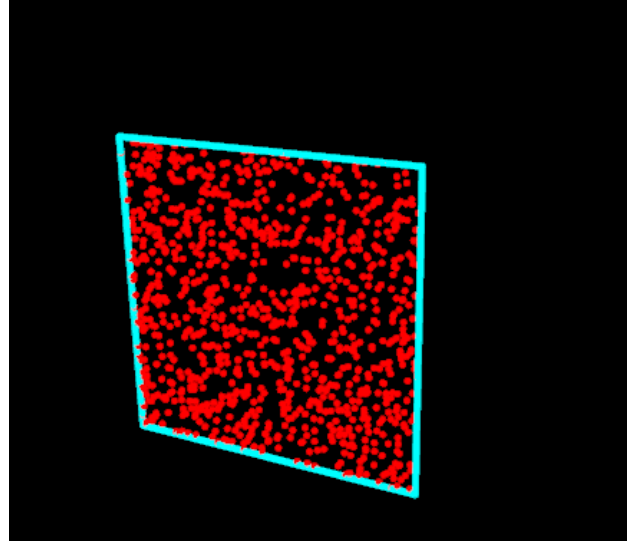


(b) The bottom plane with the intersection lines drawn in cyan.

Fig. 2: A simple point cloud generated from five planes. This division creates eight patches for the bottom plane and we use this information to determine the plane's boundaries for the ray tracing algorithm.



(a) Point cloud of the cube.



(b) Front side of the cube with the intersections.

Fig. 3: A point cloud representing a cube. Each edge represented by the intersection of the adjacent plane.

Even though the CSG tree alone can describe the spatial boundary of the primitive needed for the ray tracing algorithm, we also detect the intersection points because they may be necessary for other types of operations other than visualization.

III. RAY TRACING

Ray tracing is a common technique used in computer graphics for rendering images by tracing the path of light sources through pixels in a viewplane, and simulating the effects of its intersections with virtual objects within a scene.

Its basic operation is to intersect rays with objects, such as the point cloud primitives extracted by our algorithm, then finding the closest hit point p from a given perspective. After finding p , it is only a matter of computing its correct color,

primarily by analyzing the light sources around it. Although lights are not strictly needed – that is, we can just assign a simple shading to every point –, they are important for creating some realism in scenes. Another noticeable step is sampling, where several rays are shot through the same pixel of the viewplane to be interpolated later on. This process usually improves the overall anti-aliasing of scenes, and it is specially important when shooting rays into our primitives extracted from point clouds, instead of objects defined by implicit equations.

Fig. 4 illustrates an overview of our algorithm. It works like a conventional ray-tracer; the most remarkable difference is the intersection of rays using the structures from the extraction algorithm. First, a hit function computes the point

of intersection between the primitives and a given ray. Then, the hit point is traversed through the CSG tree. If it reaches a leaf node that is not null, then the point has hit a patch and makes part of our primitive, on the contrary, we discard this point since it is outside the boundaries.

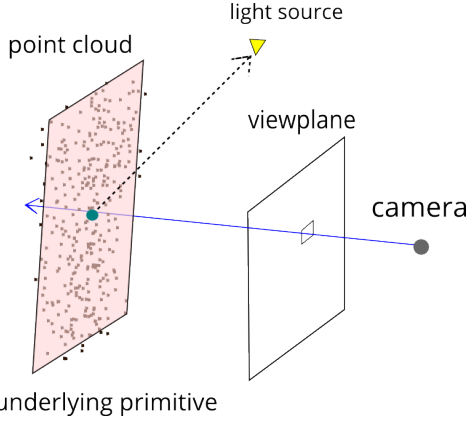


Fig. 4: The ray tracing algorithm builds an image by shooting and extending rays through the scene.

IV. RESULTS

All code was developed in the C++ programming language, and the ray-tracer was primarily based on the work of Suffern[4], although we introduced some small differences in our design.

Fig. 5 shows the result of submitting the extracted primitives from a cube point cloud with 6000 points to our ray-tracer. The extraction process took about one second, while the ray tracing took over two seconds, using, however, ten samples per pixel. Even though the results are still preliminary, the extraction, boundary retrieval, and rendering were accurately achieved with this simple example.

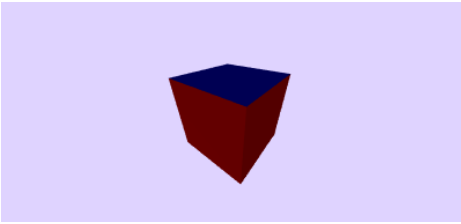


Fig. 5: The cube point cloud, rendered with the ray-tracer.

V. CONCLUSIONS

Once the primitives are extracted and their boundaries computed, we achieve a very compact representation that can replace the entire point cloud in many situations. This is specially interesting in some fields such as mechanical pieces, where the primitives can describe the objects in a very compact and precise way. Even more, this leads to a possibly noise free representation that is useful in many domains. The coupled ray-tracer brings even more power

to our compact representation, since conversion to triangle meshes, for example, becomes unnecessary.

At this point, the strongest limitation is that our algorithm is not scalable to a large number of primitives, since the memory to calculate all intersections becomes impracticable.

VI. FUTURE WORK

We list some important aspects that we would like to improve in our approach:

- improve the overall performance of the primitive extraction. Although it is able to handle small datasets without significant noise, its accuracy drops considerably in real case scenarios;
- due to the lack of accuracy in more complex models, we are not yet able to produce good CSG trees for all primitives. Consequently, the regions are not well delimited, causing erroneous renderings with the ray tracing. This is the reason that only a simple primitive – the cube – was shown here as a result within the complete extraction and rendering pipeline. We are currently working on solving this issue to work with more complex point clouds.
- store and use only information from the boundary points and the CSG tree to improve memory efficiency;
- improve the overall performance and efficiency of the ray-tracer, for example, by parallelizing it using multiple processing cores, and by implementing some acceleration schemes, such as regular grids.

REFERENCES

- [1] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: <http://doi.acm.org/10.1145/358669.358692>
- [2] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, Jun. 2007.
- [3] R. Tilove, H. Voelcker, and A. Requicha, "A study of geometric set-membership classification," Ph.D. dissertation, 1977.
- [4] K. G. Suffern and K. Suffern, *Ray Tracing from the Ground up*. AK Peters, 2007.

Interactive Timeline Visualization of Documents

Leandro S. Guedes, Rafael Garcia, Bruno Pagno, Luciana Nedel, João Comba and Carla M.D.S. Freitas

Universidade Federal do Rio Grande do Sul (UFRGS)

Porto Alegre, Brazil

{lsguedes, rgarcia, blpagno, nedel, comba, carla}@inf.ufrgs.br

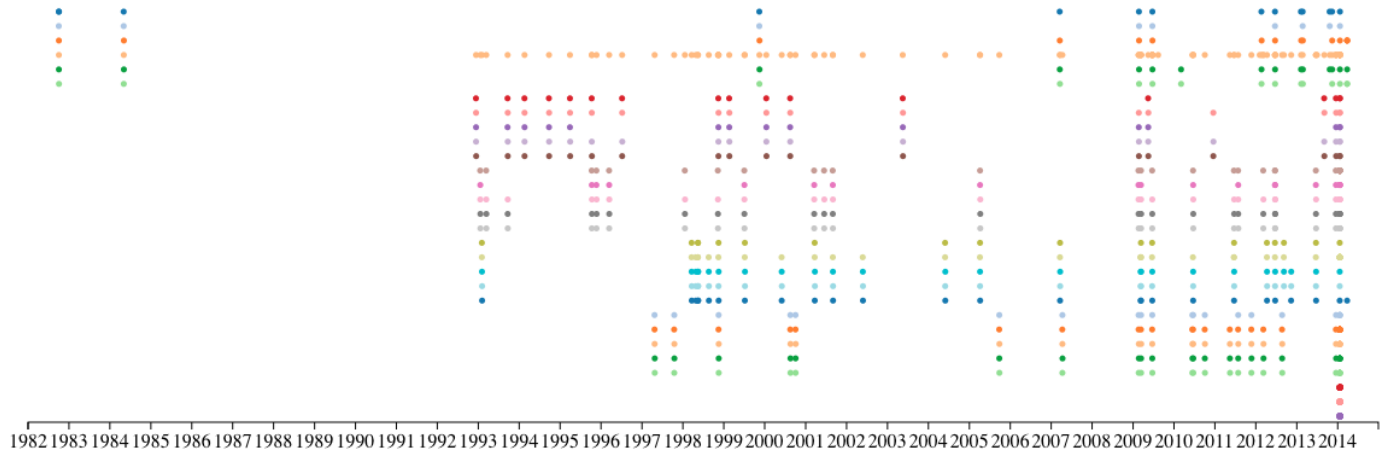


Figure 1. Timeline overview from 1982 to 2014, in this timeline, each line means a different source of documents.

Abstract—In this paper we present an approach for exploring a set of varied documents by means of a data visualization method aiming at enabling the discovery of specific facts for which there might be hints in some of the documents. The approach consists of two phases, the mining of relevant tags in the set of available documents and the actual visualization of the large amount of blog, magazines and newspapers reports. The mining phase transforms raw data into an organized file with tags related to the facts one wants to discover. The visualization phase is based on an interactive timeline that shows the documents over the years. Interactive features allow to query specific periods in the timeline as well as to filter the articles according to the tags found in the previous phase, providing a faster and efficient way of observing the presented data, and facilitating the recognition of the articles that have some interconnection, either by tags in common, source or time proximity.

Keywords—timeline visualization; documents visualization.

I. INTRODUCTION

This work was motivated by the Mini-Challenge 1 proposed as part of the VAST Challenge 2014[1]. The challenge consisted in responding a series of questions about the disappearance of a group of employees of a powerful fictional company called GASTech. The scenario is the fictional country Kronos, and the main suspect in the kidnapping is an organization known as POK (Protectors of Kronos). In order to corroborate the answers, we needed to create visualizations, which of course also allowed the analysis of the data provided within the challenge proposal.

The available documents for solving the problem were a map of Kronos, a chart describing the company organization, a spreadsheet containing the GASTech employees records, a text file containing the headers of all e-mails sent by the employees in the last weeks, short resumes and biographies of some GASTech employees, a document with historical data about Kronos and a large set (844 articles) of news reports of the last decades containing relevant information about the involved organizations.

We tackled the problem in two phases, the first for mining specific data in the set of news articles, and the second performing the exploratory task using the visualization we created. The visualization is basically an interactive timeline showing all documents along the years, separated by source (Figure 1). As we will describe later on, the visualization can be modified by the user through zooming, panning and filtering. We also show a map of the tags relevant for the problem, which were found in all documents, and details on demand in a way similar to a tooltip as well as the content of a selected document in a separate view.

Although they were created for and used specifically in this challenge, the methods presented herein are fully adaptable to any similar problem involving a large amount of data comprising a wide period of time.

In the next sections we briefly comment related work (section II), present our approach (section III) and discuss our results (section IV). Finally, in section V we draw some conclusions and comment on future work.

II. RELATED WORK

Background concepts and techniques for our work can be divided into two classes: document and text mining and visualization of temporal data. However, since our text mining is very simple up to now, we restrain ourselves to comment only visualization of temporal data, more specifically the timeline-based visualizations.

There are several works that use a timeline based visualization for solving specific problems. The need of representing temporal data is quite old. For example, in 1812, Charles Minard made a visual history of Napoleon's Russian Campaign where he illustrated its disastrous result in a graph. This classic example and other historical visualizations can be seen on [2] and on Tufte's classical book [3].

Another well-known example is Themeriver [4], which is a visualization of the thematic variations over time across a collection of documents. The flow is shown within the context of a timeline and a corresponding textual presentation of external events.

In a rather different context, Wongsuphasawat et al. [5] developed LifeFlow, where they introduced an interactive visual overview of event sequences applied to health research. In their tests, novice users were able to rapidly answer questions about the prevalence and temporal characteristics of sequences, find anomalies, and gain significant insight from the data.

Two more recent examples of timeline visualizations are by Craig and Roa-Seiler [6] and Wang and Yuan [7]. Craig and Roa-Seiler [6] used a vertical timeline to support the analysis of human-computer dialogue data, while Wang and Yuan [7] employ a timeline visualization for comparing urban trajectories.

A work closer to ours is by Pak et al. [8]. They also use data mining applied to huge datasets for finding topics that occur in a corpus along seven years. The timeline visualization helps finding sequential patterns formed by temporal sequence of those topics. Unlike our work, however, they show the appearance of topics along time while we decided to represent the news sources along the timeline.

III. DOCUMENTS TEMPORAL VISUALIZATION

The raw dataset provided within the VAST challenge proposal comprises articles with different structures. Most of the files have a header with some information like title, date and source. However, in this dataset, such information is not easily recognized due to the articles not necessarily have all of them. Also, the order and the format in which these items appear in the documents are variable, requiring a more sophisticated parsing strategy. Although all documents are written in english, a reasonable amount of them were originally written in other languages and translated by an automatic translator, making the parsing even more difficult, as this may cause the sentences have a confusing meaning due to translation errors.

The first step was to standardize the articles content, so the timeline visualization could easily handle these files. Although the files have different formats, it was possible to recognize

that they were divided into few groups of articles with similar formats. So, we first manually recognized these formats and implement parsers to read and standardize these documents. They were organized by "content", "id", "authors", "title", "source", "date" and the most important, the "tags" we were in need for solving the motivating problem.

To find the set of tags of each article, we looked for the names of all known people related with some organization (POK and GAS tech in our scenario). These names were taken from the spreadsheet of employee records, from the csv files with the e-mail headers, from the historical reports and from the biographies of the employees. The tags of each article represent all known people mentioned in its content.

By the end of this phase, we have a JSON file as a ready-to-use database.

A. Timeline-based visualization

Our dataset contains a reasonably large amount of articles separated by date, which fits perfectly in a timeline. Each article is represented on the timeline as a circular marker, with the x coordinate being used for its temporal position in the timeline, and the y coordinate serving to separate the articles' sources, facilitating the identification of related reports, since it was known that likely the same source would contain several reports about the same subject, especially in a short period of time.

The colors of the circles also facilitate the identification of the report source, having been assigned a different color for each source. The overview of this timeline is presented in Figure 1. Also, there is an option to show all the tagged articles. In this case, the articles with no associated tags were presented in light blue, and the tagged ones in dark blue. When there are more than one tagged article in the same point, the tagged one comes first. This feature can be observed in Figure 2.

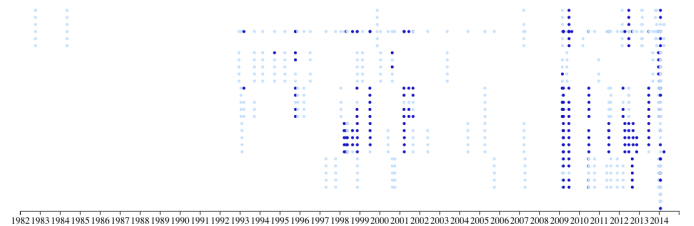


Figure 2. Showing all the tagged articles.

As the articles are distributed over a relatively long period of time, a static timeline would become cluttered, with many elements sharing similar positions. To avoid this, we built a timeline that allows zooming, causing only reports of a given year remain visible. This zoom can be extended, allowing a monthly visualization of the articles. The different zooming levels can be observed in Figure 3.

The identification of the article content is possible by hovering the mouse over the corresponding marker, showing the item number, the name of the newspaper in which it was

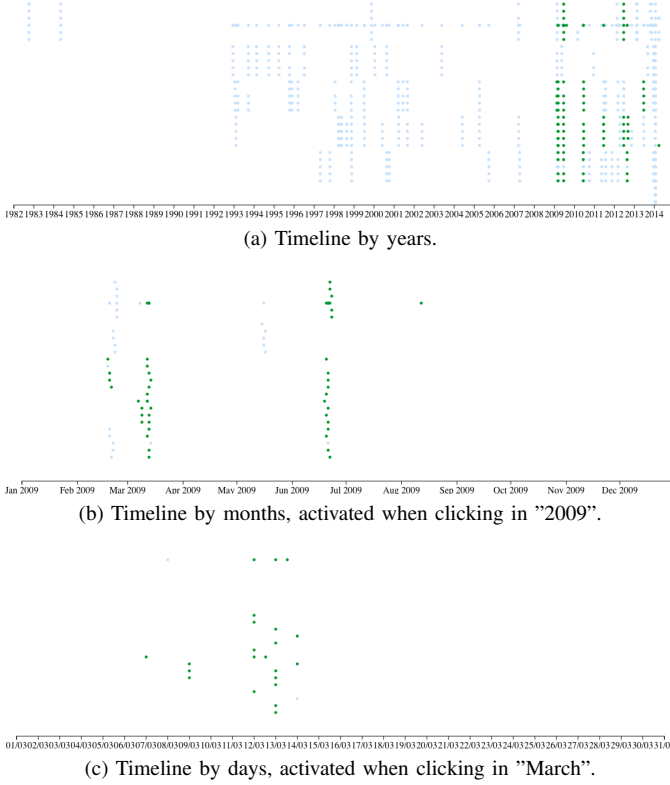


Figure 3. Visualizing articles mentioning "Elian Karel".

published, its tags and the amount of articles posted by the same source on the same day. This feature can be seen in Figure 4.



Figure 4. Identification of the article shown with the mouse over its marker.

B. Map of tags

Since there is a large number of tags, we implemented a map of tags, which is actually a set of buttons. They allow filtering all the reports, making it possible to easily identify those that contain keywords in common. Clicking on the marker of an article causes its content to appear in an information

field on the same page but without disturbing the timeline visualization. The whole page can be observed in Figure 5.

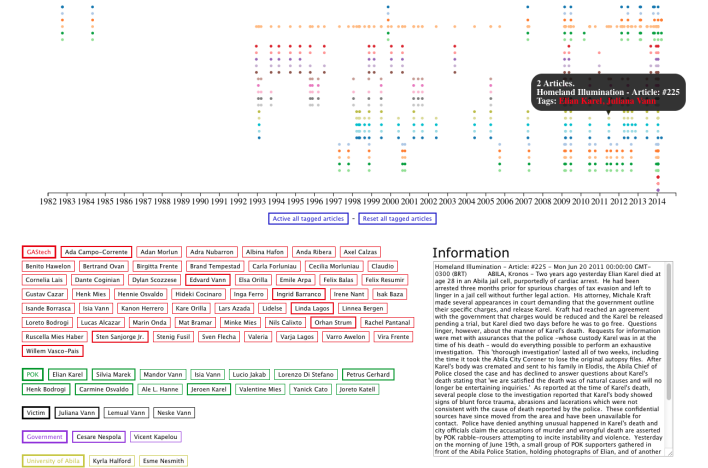


Figure 5. Page overview. Timeline is on top, buttons on the lower left side and article content on the lower right.

Such approach facilitate the observation of key points in the sequence of events of the proposed problem, allowing to quickly identify related articles, and navigate among them, also enabling a better understanding of the events reported in these articles without the need of reading all of them.

IV. RESULTS AND DISCUSSION

Our technique was implemented in Javascript and HTML5, the database being a JSON file. As for the visualization, we adopted the Data-Driven Documents (D3) library [9].

We discuss our results using the motivating problem for showing the positive aspects that the visualization provided in the analysis of the 844 articles.

The articles distribution along the timeline shows that from 1985 to 1992 no articles were produced, as well as in 1983, 2006 and 2008. Also, the first group of articles was published in October 1982. A quick look at the source of these reports shows that all of them were published in international news, allowing the understanding that this is a relevant issue even out of Kronos. When checking the contents of these reports, it is noticeable that they are emphasizing the growth and the economic power of GASTech, crucial information to understand the background kidnapping of its employees. A second group of articles appear after two years, allowing the user to quickly obtain the information that GASTech uses a controversial gas extraction technique, which can contaminate the water.

Filtering the articles by the tags on the map also greatly facilitates decision making, since it allows the user to decide reading only the reports of his/her interest. When selecting items with the tag of some GASTech employees, relevant information can be easily found, as for example, suspicion that, among the kidnapped employees, are Ingrid Barranco, Orhan Strum and Ada Campo-Corrente. An important information is

that the security Edvard Vann went through a long interrogation due to suspicion of involvement with the kidnapping, since he has the same surname of several known POK members.

Clearly, by looking at and interacting with the visualization, we were able to conclude some important facts that would only be possible by reading all the articles to make the adequate connections.

V. FINAL COMMENTS

In this paper, we employed a timeline visualization technique for showing documents containing news articles, which underwent a preprocessing procedure for extracting relevant data, and identifying whether they mention a specific content. Articles were tagged based on this specific content. Our timeline visualization allows to instantly recognize periods with a large quantity of reports, and the associated map of tags allows to filter articles per tag retaining in the timeline only those that comply with the selected tags. Other interactive features provide access to information and content of a selected article.

As future work, we intend to tag and display the employees' e-mails, so we would have better insights without the need of navigating through different visualizations at the same time. This will impose the need of designing a way of showing in the same visualization existing relations between e-mails and news, as for example, co-occurrence of tags and temporal sequence of tags in different sources, just to name a few possibilities.

We also intend to perform tests with users since up to now we were the only users of our technique for solving the problem of the VAST challenge. Another possibility of future work is to apply the technique with a different set of news sources incorporating more sophisticated data mining techniques.

ACKNOWLEDGMENTS

We are grateful to CAPES, CNPq and FAPERGS for funding our current studies.

REFERENCES

- [1] Vast challenge 2014. [Online]. Available: <http://vacommunity.org/VAST+Challenge+2014>
- [2] Timelines and visual histories. [Online]. Available: <http://www.datavis.ca/gallery/timelines.php>
- [3] E. R. Tufte, *The Visual Display of Quantitative Information*. Cheshire, CT, USA: Graphics Press, 1986.
- [4] S. Havre, B. Hetzler, and L. Nowell, "Themeriver: visualizing theme changes over time," in *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*, 2000, pp. 115–123.
- [5] K. Wongsuphasawat, J. A. Guerra Gómez, C. Plaisant, T. D. Wang, M. Taieb-Maimon, and B. Shneiderman, "Lifeflow: Visualizing an overview of event sequences," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 1747–1756. [Online]. Available: <http://doi.acm.org/10.1145/1978942.1979196>
- [6] P. Craig and N. Roa-Seiler, "A vertical timeline visualization for the exploratory analysis of dialogue data," in *Information Visualisation (IV), 2012 16th International Conference on*, July 2012, pp. 68–73.
- [7] Z. Wang and X. Yuan, "Urban trajectory timeline visualization," in *Big Data and Smart Computing (BIGCOMP), 2014 International Conference on*, Jan 2014, pp. 13–18.
- [8] P. C. Wong, W. Cowley, H. Foote, E. Jurrus, and J. Thomas, "Visualizing sequential patterns for text mining," in *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*, 2000, pp. 105–111.
- [9] Data-driven documents. [Online]. Available: <http://d3js.org/>

Visualização de Incertezas para a Melhoria da Consciência de Situação em Sistemas de Comando e Controle de Conflitos Urbanos

Natália Oliveira², Leonardo Botega^{1,2}, Claudia Berti¹ e Regina Araújo¹

¹Wireless Networks and Distributed Interactive Simulations Lab (WINDIS)

Departamento de Computação - Universidade Federal de São Carlos (UFSCar) - São Carlos - São Paulo - Brasil

²Computing and Information Systems Research Lab (COMPSI)

Centro Universitário Eurípides de Marília (UNIVEM) - Marília - São Paulo - Brasil

natoliiveira@gmail.com, {leonardo_botega, claudia.berti, regina}@dc.ufscar.br

Abstract— *Reaching the situation awareness (SAW) state is a challenging process. Systems that aim such objective, as the military command and control (C2), need to acquire, process and present trustful information for the decision making in critical scenarios. SAW can be damaged by problems referring to data quality such as uncertain data, aggregated to the process during acquisition, processing and cognition routines. To mitigate such problems and to avoid that they harm decision making, the visualization in C2 systems support operators to perceive and comprehend the environment in a more reliable fashion. In this paper it is presented visualization techniques for uncertain data for the C2 domain, inspired by methods found in literature. Later, it is shown the first results of the application of such techniques for reaching SAW in a C2 case study on the management of urban conflicts.*

Keywords - Uncertainty Visualization; Situation Awareness; Command and Control Systems.

Resumo— *Atingir o estado de consciência da situação (SAW) é um processo desafiador. Sistemas que o visam tal objetivo, como os de comando e controle (C2) militar, necessitam adquirir, processar e apresentar informações confiáveis para a tomada de decisão em cenários críticos. SAW pode ser degradada por problemas de qualidade, tais como dados incertos, que podem ser agregados ao processo durante rotinas de aquisição, processamento e cognição. Para mitigar estes problemas e evitar que prejudiquem a tomada de decisão, visualizações em sistemas de C2 dão suporte aos operadores para que possam perceber e compreender o ambiente de forma mais confiável. Neste trabalho são apresentadas técnicas de visualização de incerteza para o domínio de C2, inspiradas por métodos encontrados na literatura. Posteriormente, são apresentados os primeiros resultados da aplicação de tais técnicas para atingir SAW em um estudo de caso de C2 para a gestão de conflitos urbanos.*

Palavras-chave - Visualização de Incertezas; Consciência de Situação, Sistemas de Comando e Controle.

I. INTRODUÇÃO

Representação de dados e informações através de visualizações é um estudo desafiador, devido à necessidade de fornecer aos operadores tomadores de decisão, uma compreensão da realidade de determinado cenário. Tal cenário pode ser reproduzido fidedignamente ou possuir em sua

constituição problemas de qualidade adquiridos em processamento e interpretação da informação.

Em [1], Pang *et al* descrevem incerteza como um conceito complexo que aborda diversas dimensões, como inconsistência, dúvida, confiabilidade, imprecisão e erro. A mesma também pode incluir variações estatísticas, diferenças, ruídos ou dados ausentes.

A incerteza é um elemento que está associado à formação e composição de dados e informações, e a mesma pode surgir desde a aquisição até o processamento e cognição dos mesmos. No domínio de C2, o processo pode ser prejudicado devido a uma decisão imprecisa originária da má interpretação de dados.

Kostin em [2], define C2 como o processo de autoridade e direção exercido por comandantes militares e gerentes civis sobre sua equipe e seus recursos materiais, com o intuito de alcançar objetivos estratégicos e táticos.

Neste contexto, os objetivos estratégicos e táticos do C2 são alcançados quando há a definição de processos na tomada de decisão, os quais abrangem desde a decisão até a execução final, passando pela correta divulgação da informação. A principal função dos comandantes e gerentes civis é criar um modelo estruturado de operações militares, desde a criação da estratégia até o recurso executante, atingindo assim, o estado de SAW [2].

Deste modo, o presente artigo discute técnicas para a representação visual da qualidade de dados e informações no domínio de C2. Tais dados são provenientes de fontes heterogêneas, tais como sensores físicos, redes sociais, serviços de vigilância e câmeras de segurança. Ao final, são apresentadas visualizações em um estudo de caso para representar dimensões de qualidade de dados como a integridade, consistência e dimensões de tempo, visando habilitar a eficiente cognição do usuário que busca tomar decisões conscientes da situação de um conflito urbano.

II. TRABALHOS RELACIONADOS

Fricker e Macklin [3], apresentam uma exposição baseada em mapas onde são representados dados provenientes de bases militares distintas. Toda a informação é apresentada em

relação à quantidade de recursos e nível de potência em relação às demais bases. Neste exemplo, cores, formas e tamanho são utilizadas para auxiliar a representação, o que permite ao operador do sistema, facilidade no entendimento situacional.

Em [3], Fricker e Macklin mapeiam rotas para conduzir os recursos a partir de bases aliadas para bases inimigas. A informação é apresentada em relação à criticidade, quantidade de recursos, nível de potência de cada base e se a mesma está em funcionamento. Este exemplo transmite mais informações ao tomador de decisões, pois utiliza cores na estrutura terrestre pra apresentar o grau de segurança de cada região.

Gouin e Evdokiou [4] apresentam uma visualização avançada desenvolvida para o programa do Posto de Comando do Futuro (CPOF), chamado *Circular Blobs*. Em um terreno tridimensional, círculos representam a força de implantação. A espessura da linha representa a resistência da força e o diâmetro do círculo mostra a gama de armas. Neste exemplo, recursos como cores, tamanho, forma, bordas e espessura tornam-se fatores positivos para a representação das informações e sua qualidade associada.

III. CONSCIÊNCIA DA SITUAÇÃO NO DOMÍNIO MILITAR

Segundo Endsley [5], SAW é a percepção de elementos no ambiente em um volume e espaço de tempo, a compreensão do seu significado e a projeção de seu estado em um futuro próximo. Atingir o estado de consciência de situação é um processo que ocorre na mente do humano e é de suma importância entender o nível de conhecimento obtido pelo mesmo na utilização do sistema, para que falhas na cognição não prejudiquem o processo.

Para que os comandantes militares adquiriram SAW de determinada missão por meios visuais, faz-se necessário a utilização de mapas, gráficos, sobreposições e distorções visuais em sua composição, para que as visualizações sejam integradas ao processo, e desta forma, as ações possam ser priorizadas, fazendo com que as visualizações sejam úteis à tomada de decisão reduzindo o tempo de análise de informações relevantes.

A análise visual combina técnicas avançadas de mineração de dados com a interface de sistemas que provêm SAW para fornecer informações que suportem raciocínios complexos e destaquem informações significativas de grandes conjuntos de dados.

Sempre que os dados são processados, transformados e mapeados em representações visuais, a incerteza pode ser agravada, o que torna difícil manter a qualidade do dado ao longo do processo [6]. Com as visualizações e a incerteza disposta de forma explícita em todo o processo, o tomador de decisões terá como base níveis de confiança de dados, obtendo informações que possam auxiliar na tomada de decisão.

As técnicas apresentadas para a criação das representações baseiam-se em estudos perceptivos e cognitivos de Tufte [7] [8], Chambers [8][9] e Bertin [8][10], tais como métodos intrínsecos, como uso de parâmetros gráficos de cor, transparência, textura e glifos que codificam a incerteza. A literatura de SAW como a de Endsley [11] também suporta as decisões de representação com o foco em SAW.

Adicionalmente, foram consideradas as dimensões de qualidade de dados como precisão, integridade, consistência, dimensões relacionadas a tempo [12] e taxonomias das causas do conhecimento imperfeito como as descritas por Gershon em [13], que envolvem dados e informações corrompidas, incompletas, inconsistentes, de difícil compreensão e incerteza propriamente dita.

IV. TÉCNICAS DE VISUALIZAÇÃO DE INCERTEZAS PARA O DOMÍNIO DE COMANDO E CONTROLE MILITAR

Representações de informações de um determinado objeto, tais como incerteza, imperfeição e outras, podem ser abordadas de duas maneiras: intrínseca ou extrínseca [13]. As Intrínsecas são utilizadas para representar algo que faz parte do objeto, como variações na aparência. Já as extrínsecas, são utilizadas para mostrar o que não faz parte do objeto, apresentando novas estruturas associadas para identificar novos atributos [14].

Variáveis gráficas, como as usadas em cartografia (ex.: tamanho, cor, localização, valor, forma, textura e orientação), quando sofrem adição de saturação de cor, podem ser mapeadas para diferentes tipos de incerteza nos dados: estas manipulações são consideradas métodos intrínsecos. Quando comparadas os dois tipos de visualizações, as intrínsecas têm um grande potencial de processamento visual e redução de demandas de atenção visual, pois a representação da incerteza poderia ser sobreposta ou integrada ao objeto. Já para as extrínsecas serem vantajosas devem estar contidas de maneira adicional, pois estas não poderiam ser integradas à visualização, e sim utilizadas de modo classificativo [13].

Visto que a tomada de decisão militar pode demandar o conhecimento de dados incertos, faz-se necessário tratá-la na visualização. Em [8], os autores apresentam uma classificação para técnicas de representação de incerteza: utilização de glifos, adição e modificação da geometria, de atributos, animação, sonorização e técnicas psicos-visuais, as quais permitem maior cognição por parte dos usuários. As técnicas para a representação de incerteza estudadas em [15] não se limitam a um domínio específico. Aplicando-as em C2, objetiva-se obter facilidade na compreensão e apresentação das informações que remetam a situações que acontecem em determinado cenário. As técnicas descritas são:

- Variáveis gráficas livres: cor, tamanho, posição, foco, clareza, imprecisão, saturação, transparência e borda, conforme a Figura 1.

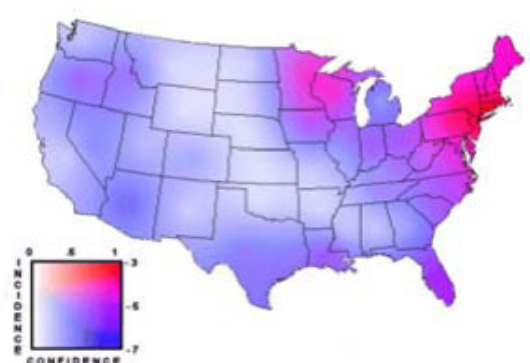


Figura 1. Visualização em mapa com a utilização de variação de cores, variação da escala de cor e geometria [8].

- Objetos adicionais: rótulos, imagens ou símbolos. Sua abordagem é incluir a incerteza na magnitude, direção e comprimento em hieróglifos [8].
- Animação: a incerteza é mapeada para parâmetros de animação como a velocidade ou duração, borrão de movimento, alcance ou extensão do movimento.
- Representação interativa: por exemplo, incerteza pode ser descoberta pela interação do mouse [8].
- Sonificação e técnicas psico-visuais: incorporação de acústica, mudanças de tom, volume, ritmo, vibração ou piscar mensagens textuais [8].

V. VISUALIZAÇÃO DE INFORMAÇÕES PARA COMANDO E C2 MILITAR EM CONFLITOS URBANOS

O estudo de caso para este artigo lida com C2 em um cenário de grande evento, mais especificamente sobre os incidentes de desordem social dentro de uma estação de metrô após um grande evento (jogo de futebol), em um dos grandes estádios na cidade de São Paulo. Este contexto, também apresenta a preocupação em manter a ordem e a segurança, dada à restrição de elevada densidade populacional.

Vigilância pública foi escolhida para exemplificar os processos de um sistema que utiliza fusão de dados semi-automatizados. Este trabalho enquadra-se como um dos passos finais de tal sistema, próximos do usuário e da tomada de decisão. Frequentemente, identificação e rastreamento de pessoas e objetos perigosos envolvem vigilância pública.

A fim de identificar infratores há uma série de recursos de vigilância a serem utilizados, tais como câmeras, sistemas de identificação automática e rádio. Para interação com o sistema do C2, é fornecido uma interface gráfica (tangível/natural) de usuário (GUI), que permite tanto a interação entre os operadores do sistema, bem como a possibilidade de exibir a saída de identificação e rastreamento de criminosos.

Tendo por base o cenário e as técnicas de visualizações apresentados, foi construída uma visualização que identifica os tipos de dados obtidos desta ocorrência no centro de C2 militar, a fim de que as mesmas fossem mapeadas e destacadas de forma clara e concisa ao operador do sistema.

A visualização é composta por dados que costumam ser obtidos e analisados em casos de grandes conflitos urbanos, a fim de identificação da desordem social e seus causadores, para que a consciência da situação por parte dos usuários seja concreta e a tomada de decisão rápida e precisa. Dados e informações de chamadas de emergência (190), SMS de reclamação, *posts* de redes sociais, sensores de áudio, câmeras de segurança e depoimentos de oficiais presentes no local foram aplicadas a um mapa para representar a situação e exemplificar sua utilização. Estas informações são recolhidas e analisadas para fins de monitoramento e avaliação de risco frente às dificuldades de alto fluxo de usuários do metrô.

A Figura 2 representa um incidente ocorrido numa estação de metrô na cidade de São Paulo. O diâmetro demarcado em vermelho representa o local da ocorrência, já os pontos das ocorrências e informações recebidas sobre o incidente são representados no mapa através de círculos e marcadores, estas

informações são representadas utilizando as técnicas de representação de incerteza que foram apresentadas neste artigo, como cores, adição de transparência, bordas e glifos. No caso de chamadas de emergência (190), as mesmas são representadas no mapa por círculos azuis e viaturas como marcadores representam a ocorrência. Quando há um relato de algum oficial presente no local o mesmo é mostrado na visualização através de círculos marrons, somados à imagem de um oficial.



Figura 2. Visualização de informações de uma ocorrência em uma estação de metrô após um grande evento.

Quando a sala de controle e vigilância interna do metrô disponibiliza imagens e informações como as de câmeras de segurança e arquivos de sensores de áudio. Tais dados são representados na visualização através de um círculo laranja e o desenho de uma câmera, e um círculo cor de rosa com um alto-falante como marcador respectivamente. Posts de redes sociais e SMS de reclamação também são analisados. Esta visualização é representada por círculos vermelhos com a imagem da rede social *Facebook*. Círculos verdes são utilizados para representar as mensagens de texto, juntamente com balões de conversa.

Nesta visualização, ainda é possível notar uma área circular mais escura em relação ao mapa. Esta representa o raio de abrangência de informações que o usuário deseja alcançar para a análise. Interagindo sobre os marcadores é possível obter informações adicionais sobre a ocorrência, como data e hora, a identificação de quem gerou a informação, o conteúdo da mesma, localização e grau de confiabilidade.

O objetivo da utilização de marcadores na visualização é que o uso de glifos (objetos adicionais) para o tratamento de dados incertos representam mais facilmente uma informação, o que torna todo o processo mais rápido e intuitivo. Glifos são símbolos utilizados para representar dados através de propriedades visuais utilizadas em interface de usuário, objetivando rápida compreensão de um processo [15]. A utilização dos mesmos contempla as técnicas de Tufte [7][8], Chambers [8][9] e Bertin [8][10] previamente anunciadas.

VI. RESULTADOS PRELIMINARES

Na Tabela 1 são apresentados os dados e elementos visuais utilizados na composição da visualização.

Estudos de Jiang *et al* [16] testaram a utilização de cinco diferentes tipos de escalas de cores para a representação de

incerteza em dados. Suas conclusões indicaram que quanto mais clara a cor utilizada maior o nível de alerta, já os tons mais escuros como cinza, preto e cores vibrantes como vermelho, causam estresse visual. Em [17], o autor conclui que a utilização de transparência facilita a percepção, por ter valor híbrido (codificação redundante).












Visualização das Informações descritas			
Informação	Tratamento da informação		
	Marcador	Cor	Descrição
Chamada de Emergência	Viatura 		Azul – Informação de grande confiabilidade. Cor forte: causa estresse visual – Maior atenção do usuário [16].
SMS de reclamação	SMS 		Verde – Informação com nível de confiabilidade mediana, incerteza tratada com a adição de transparência [17].
Posts de Redes Sociais	Facebook 		O ícone remete à rede social. O vermelho gera estresse visual [16], e a transparência evidencia a incerteza.
Relato de oficial	Oficial 		Informação confiável, utilizada com cor neutra que remete confiabilidade ao usuário [16].
Sensor de áudio	Alto-falante 		Roxo: cor escura que chama atenção por gerar estresse visual. A transparência é utilizada por não poder reconhecer e identificar com clareza o som e seu interlocutor.
Câmeras internas de segurança	Câmera 		O amarelo foi escolhido, pois gera contraste em relação ao marcador, cor forte, com a adição de transparência representando incerteza, visto que as imagens, não identificam as pessoas [17][16]

Tabela 1. Classificação dos recursos aplicados na visualização

De acordo com a literatura apresentada, as cores utilizadas na visualização causam certo estresse visual, o que chama a atenção do operador para informações importantes para a tomada de decisão. Exemplos são: vermelho, amarelo, azul escuro. O uso de transparência se torna indispensável visto que estudos de [17] afirmam que transparência é entendida como incerteza, e que quanto maior o nível de transparência aplicado, maior a incerteza na cognição.

VII. CONCLUSÕES PRELIMINARES

Com a aplicação das técnicas utilizadas, conclui-se que o uso de glifos permite com maior facilidade à cognição de qual tipo de informação é apresentado. Quanto à incerteza contida no dado, a transparência é a técnica que trata a informação com mais clareza e facilidade.

Com o uso de técnicas intrínsecas, como a utilização de cores, bordas, transparência, tamanhos e posição, somadas ao uso de glifos, tais como legendas e o uso de marcadores (extrínsecos), se tornou possível atingir SAW. Novos estudos serão efetuados pelos autores, a fim de classificar o tempo

gasto e a utilização de elementos visuais, como sonificação, animação e interação.

Como resultados preliminares, constatou-se que as técnicas intrínsecas como variação de cor, aplicação de transparência, bordas e utilização de glifos, quando aplicadas ao domínio de C2, permitem maior e mais rápido entendimento situacional, permitindo melhor a análise do processo para que a tomada de decisão seja baseada no nível de criticidade da incerteza da informação.

AGRADECIMENTOS

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo auxílio financeiro. Processo número 144495/2013-9.

REFERÊNCIAS

- Pang, A. T., Wittenbrink, C. M., & Lodha, S. K. (1997). Approaches to uncertainty visualization. *The Visual Computer*, 13(8), 370-390.
- Kostin, S. (2001). Modelo para gerenciamento de dados em um ambiente de comando e controle. PUC/RJ.
- Fricker, L., & Macklin, C. (2004). Improving Campaign Assessment and Decision Making in Command and Control Through the Use of Visualisation Techniques. QINETIQ LTD FARNBOROUGH (UNITED KINGDOM) CENTRE FOR HUMAN SCIENCES.
- Gouin, D., & Evdokiou, P. (2004). A showcase of visualization approaches for military decision makers. DEFENCE RESEARCH AND DEVELOPMENT CANADAVALCARTIER (QUEBEC).
- ENDSLEY, M. R. Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37(1), 32-64. 1995.
- Correa, C., Chan, Y. H., & Ma, K. L. (2009, October). A framework for uncertainty-aware visual analytics. In *Visual Analytics Science and Technology*, 2009. VAST 2009. IEEE Symposium on (pp. 51-58).
- Tufte, E. R., & Rifkin, S. (1997). Visual Explanations: Images and Quantities, Evidence and Narrative. *ISIS-International Review Devoted to the History of Science and its Cultural Influence*, 88(4), 748-748.
- Riveiro, M. (2007, July). Evaluation of uncertainty visualization techniques for information fusion. In *Information Fusion*, 2007 10th International Conference on (pp. 1-8). IEEE.
- J. M. Chambers and W. S. Cleveland and B. Kleiner, and P. A. Tukey, *Graphical Methods for Data Analysis*. New York: Chapman and Hall, 1983.
- Bertin, J. , *Graphics and Graphics Information Processing*. Berlin-New York: Walter de Gruyter, 1981, (trans. W. Berg and P. Scott).
- Endsley, M. R., & Connors, E. S. (2008, July). Situation awareness: State of the art. In *Power and Energy Society General Meeting- Conversion and Delivery of Electrical Energy in the 21st Century*, 2008 IEEE (pp. 1-4). IEEE.
- Mecella, M., Scannapieco, M., Virgillito, A., Baldoni, R., Catarci, T., & Batini, C. (2002). Managing data quality in cooperative information system. *Lecture Notes in Computer Science*, 2519, 486-502.
- Gershon, Nahum. "Visualization of an imperfect world." *Computer Graphics and Applications*, IEEE 18.4 (1998): 43-45.
- Ann M. Bisantz, (2013) *Uncertainty Visualization and Related Techniques*. The Oxford Handbook of Cognitive Engineering.
- Wittenbrink, C. M., Pang, A. T., & Lodha, S. K. (1996). Glyphs for visualizing uncertainty in vector fields. *Visualization and Computer Graphics*, IEEE Transactions on, 2(3), 266-279.
- B. A. Jiang B. and F. J. Ormeling, "Some perceptual aspects of colouring uncertainty," *Advances in GIS Research* II, pp. 477-490, 1996.
- T. Zuk and S. Carpendale, "Theoretical analysis of uncertainty visualizations," in *Visualization and Data Analysis*, Proc. of SPIE-IS&T Electronic Imaging, SPIE, R. F. Erbacher, J. C. Roberts, M. T. Gröhn, and K. Borne, Eds., vol. 6060, 2006.