

Software Livre para Empresas

Antunes Dantas da Silva^{*}, Gabriel Silva Vinha, Italo M. de Lima Poroca e Valter V. M. de Lucena

^{*}Correspondência:

antunes.dantas@ccc.ufcg.edu.br
Universidade Federal de Campina
Grande, Centro de Engenharia
Elétrica e Informática,
Departamento de Sistemas e
Computação, Rua Aprígio Veloso,
882, Bairro Universitário,
58429-140, Campina Grande,
Brasil.

Resumo

Escrever no máximo 150 palavras no resumo do trabalho. Exemplo: The objective of this work is to determine if people are interacting in TV video by detecting whether they are looking at each other or not. We determine both the temporal period of the interaction and also spatially localize the relevant people. We make the following four contributions: (i) head detection with implicit coarse pose information (front, profile, back); (ii) continuous head pose estimation in unconstrained scenarios (TV video) using Gaussian process regression; (iii) propose and evaluate several methods for assessing whether and when pairs of people are looking at each other in a video shot; and (iv) introduce new ground truth annotation for this task, extending the TV human interactions dataset. The performance of the methods is evaluated on this dataset, which consists of 300 video clips extracted from TV shows. Despite the variety and difficulty of this video material, our best method obtains an average precision of 87.6% in a fully automatic manner.

Keywords: Escreva; algumas; palavras-chaves; aqui!

1 Introdução

Escreva introdução e motivação do seu trabalho. Tente convencer o leitor da importância da sua pesquisa. Exemplo: If you read any book on film editing or listen to a director's commentary on a DVD, then what emerges again and again is the importance of eyelines. Standard cinematography practice is to first establish which characters are looking at each other using a medium or wide shot, and then edit subsequent close-up shots so that the eyelines match the point of view of the characters. This is the basis of the well known 180° rule in editing.

The objective of this paper is to determine whether eyelines match between characters within a shot—and hence understand which of the characters are interacting [1]. The importance of the eyeline is illustrated by the three examples of Figure 1 - one giving rise to arguably the most famous quote from Casablanca, and another being the essence of the humour at that point in an episode of Fawlty Towers. Our target application is this type of edited TV video and films. It is very challenging material as there is a wide range of human actors, camera viewpoints and ever present background clutter. The thirty Brodatz textures used are shown in Figure 1.

begin

Gray level co-occurrence matrix (GLCM) [2] describes the relative frequencies with which two pixels separated by a distance d under a specified angle occur on the image. Then, the GLCM matrices are pre-processed in order to obtain input data for



Figura 1 - Exemplo de Figura 01.

the clustering or classification modules. In the Clustering module the SOM neural network organises and extracts prototypes from the processed matrices, which ends the learning stage. The classification module receives a pre-processed query image and compares it with the prototypes (representations of clusters) obtained in the clustering module. The final result is a list of images belonging to a few number of clusters considered to be the nearest to the user's query. Figure 2 shows these building blocks.

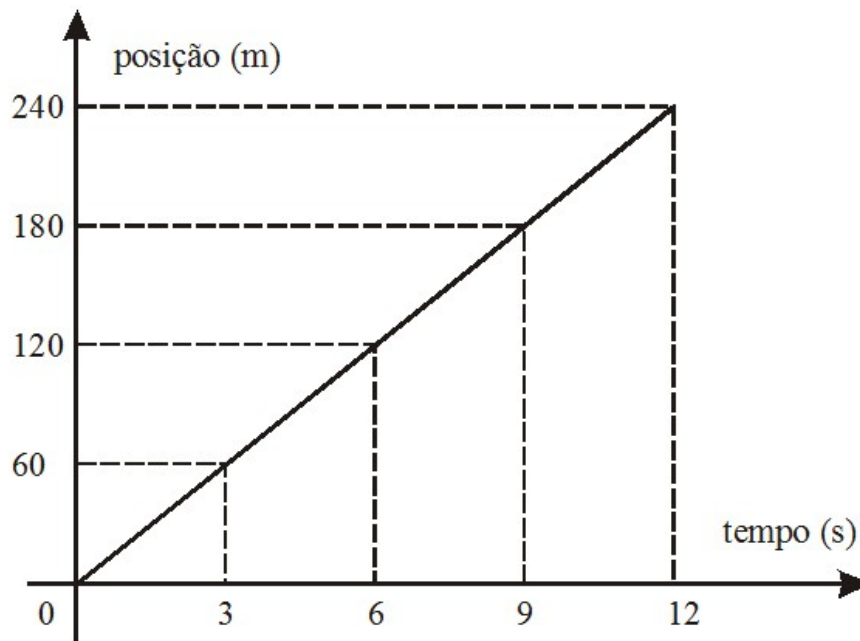


Figura 2 - Exemplo de Figura 02.

2 Motivação

If we assume that sensitive cells follow a deterministic decay $Z_0(t) = xe^{\lambda_0 t}$ and approximate their extinction time as $T_x \approx \frac{1}{\lambda_0} \log x$, then we can heuristically estimate the expected value as:

$$E[Z_1(vT_x)] = \frac{\mu}{r} \log x \int_0^1 x^{1-u} du \quad (1)$$

$$E[Z_1(vT_x)] = \frac{\mu}{r} x^{1-\lambda_1/\lambda_0 v} \log \quad (2)$$

$$1 = 10 \quad (3)$$

$$E[Z_1(vT_x)] = \frac{\mu}{r} \log x \int_0^1 x^{1-u} du \quad E[Z_1(vT_x)] = \frac{\mu}{r} x^{1-\lambda_1/\lambda_0 v} \log \quad (4)$$

Thus we observe that this expected value is finite for all $v > 0$ (also see [3]).

2.1 Exemplo de Sub-Seção

In this section we examine the growth rate of the mean of Z_0 , Z_1 and Z_2 . In addition, we examine a common modeling assumption and note the importance of considering the tails of the extinction time T_x in studies of escape dynamics. We will first consider the expected resistant population at vT_x for some $v > 0$, (and temporarily assume $\alpha = 0$).

$$E[Z_1(vT_x)] = \mu T_x \int_0^{\inf} \lambda_1 T_x(v-u) du \quad (5)$$

If we assume that sensitive cells follow a deterministic decay $Z_0(t) = xe^{\lambda_0 t}$ and approximate their extinction time as $T_x \approx -\frac{1}{\lambda_0} \log x$, then we can heuristically estimate the expected value as.

3 Software Livre

Por *Software Livre* entende-se aquele que respeita a liberdade e o censo de comunidade do usuário. Isto é, todo o *software* que pode ser usado, copiado, estudado, modificado e redistribuído sem restrições.

Durante a década de 60, quando os computadores eram mais utilizados em empresas e instituições governamentais, não havia a ideia de *software* e *hardware* como algo separado, do ponto de vista comercial. Em geral, o *software* era entregue junto com o código-fonte, ou apenas este último era entregue. Devido a isso, grupos e comunidades de usuários que trocavam informações e compartilhavam código eram comuns. A partir daí, pode-se afirmar que o *software* era livre, em suas origens.

Ainda nessa mesma década, sistemas operacionais e compiladores de linguagens de programação começaram a evoluir, aumentando drasticamente seus custos. Assim, uma indústria pequena e crescente começava a surgir, competindo diretamente com os *softwares* entregues juntos ao *hardware*. Em 1970, a IBM, líder do mercado de computadores da época, anunciou que a partir daquele ano passaria a vender parte de seus programas separada das máquinas. Com isso, a indústria de *software* tomou um rumo em que restrições de acesso e de compartilhamento de código entre desenvolvedores ficaram cada vez mais comuns.

Em 1978, Donald Knuth professor da Universidade de Stanford, começou a trabalhar no TeX, sistema de tipografia popular até hoje no meio acadêmico, que foi distribuído com a ideia de que qualquer um pudesse usá-lo sem restrições (seu código-fonte estava em uma seção do volume 2 do seu livro *The Art of Computer Programming*). A partir daí, a ideia base do *software* livre como conhecido hoje começou a surgir.

Em 1983, Richard Stallman, funcionário do MIT, teve uma experiência negativa com *software* comercial, e deu origem ao Projeto GNU. Durante o período que estava no MIT, identificou uma falha no *software* de uma impressora. Ao tentar corrigi-lo, a empresa se negou disponibilizar o código-fonte. Isso o motivou a criar um mecanismo legal de garantia para que todos pudessem desfrutar dos direitos de copiar, redistribuir e modificar *Software*, dando origem à licença GPL. Para institucionalizar o Projeto GNU, Stallman fundou a Free Software Foundation. Nasce assim o Movimento do *Software* Livre.

Em julho de 1991, Linus Torvalds, estudante da Universidade de Helsinki - Finlândia, divulgou nota com menções sobre seu projeto de construir um núcleo operacional livre, similar ao Minix, e obteve ajuda de vários desenvolvedores ao redor do mundo. Em setembro do mesmo ano, Linus lançou a versão oficial do que hoje é o Linux. Centenas de desenvolvedores se juntaram ao projeto para integrar todo o sistema GNU (compilador, editor de textos, shell, etc) em torno do núcleo do Linux. Nasce então, sob a licença GPL, o sistema operacional GNU/Linux.

Após isso, o movimento do *software* livre vem crescendo com grandes projetos, tais como todas as distribuições do Linux, o OpenStack, o Eclipse, e empresas, como a RedHat, Canonical, Free Software Foundation – como já citada –, entre outras.

Algo a ser esclarecido é que *software* livre é diferente de *software* em domínio público e de *software* gratuito. Em domínio público, significa que seu autor abriu mão dos seus direitos autorais. E quanto a ser gratuito, pode citar os serviços de *cloud-computing* da RedHat, e a distribuição Suse Linux, com foco empresarial, que não são gratuitos.

Entram então alguns conceitos importantes a respeito de *software* livre, tais como *software* como um produto (SaaP), *software* como um serviço (SaaS) e os componentes da produção de *software*.

3.1 Software as a Product

Software as a Product//citação aqui?? – ou *Software* como um Produto, em tradução livre –

3.2 Software as a Service

olar

3.3 Componentes da Produção de Software

acesso ao software

4 Licenças de Publicação

tarara

5 Software Livre Para Empresas

5.1 estatísticas de mercado para saap
oi

5.2 Software as a service
aqui vc faz

5.3 core

6 Tendências

olar

Referências

1. Pressman, R.: SOFTERRR Engineering: A Practitioner’s Approach, 6th edn. McGraw-Hill, Inc., New York, NY, USA (2005)

2. Ferris, C., Farrell, J.: What are web services? Commun. ACM **46**(6), 31 (2003)

3. Rosenfeld, A., Troy, E.B.: Visual Texture Analysis. In: A Symposium on Feature Extraction and Selection in Pattern Recognition, vol. 1, pp. 115–124 (1970)

4. de Oliveira Domingues, M.A.: Métodos robustos de regressão linear para dados simbólicos do tipo intervalo. PhD thesis, Universidade Federal de Pernambuco (UFPE), Recife (2010)

5. Diday, E.: Introduction à l’analyse des données symboliques. Research report, Institut National de Recherche en Informatique et en Automatique, INRIA (1989)

6. Lutkebohle, I.: BWorld Robot Control Software. <http://www.nlm.nih.gov/research/visible/> - accessed 19-July-2008

7. de tal, F.: Livro de Eng. Civil, 6th edn. McGraw-Hill, Inc., New York, NY, USA (2005)



Fulano de Tal was born in India. She received the B.S. degree in computer science from Kurukshetra University, Kurukshetra, India and the M.Phil. and Ph.D. degrees from the University of Exeter, Exeter, UK in 1999, 2001 and 2004, respectively. Her Ph.D. was in the area of machine learning for image analysis in aviation security. Her main research interests include image processing, natural scene analysis, video analysis, and neural networks. She has published more than 30 papers in the area of machine learning for image analysis in peer reviewed journals and conferences. Currently she is a Senior Research Fellow at Loughborough University leading the project on imaging for road transport applications.



Fulano de Tal was born in India. She received the B.S. degree in computer science from Kurukshetra University, Kurukshetra, India and the M.Phil. and Ph.D. degrees from the University of Exeter, Exeter, UK in 1999, 2001 and 2004, respectively. Her Ph.D. was in the area of machine learning for image analysis in aviation security. Her main research interests include image processing, natural scene analysis, video analysis, and neural networks. She has published more than 30 papers in the area of machine learning for image analysis in peer reviewed journals and conferences. Currently she is a Senior Research Fellow at Loughborough University leading the project on imaging for road transport applications.