

# 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

O *software* livre é uma realidade que existe desde os primórdios da computação. Baseando-se na ideia básica de que o código fonte deve ser público, o movimento do *software* livre gerou, e ainda gera, bastante polêmica dentre a comunidade da tecnologia da informação, especialmente quando o assunto tange as grandes corporações que lucram com a venda de *softwares* proprietários. Como movimento, iniciou em 1983 com um americano chamado Richard Stallman, que liderou o desenvolvimento de um sistema operacional baseado totalmente nas ideias do *software* livre.

Para ser considerado livre, um *software* deve seguir determinadas "leis", que definem como ele deve ser publicado. Para facilitar a publicação, foram criadas licenças genéricas que servem para qualquer *software*.

Um dos principais questionamentos quando o assunto é tratado é como empresas podem faturar fabricando código aberto. Como será exposto posteriormente, existem diversos modelos de negócios que podem ser abordados para este fim.

Este artigo seguirá a seguinte estrutura: na seção 2, será mostrada a motivação para este estudo. Na seção 3, o tema *software* livre será abordado de maneira mais detalhada, bem como modalidades que onde este é encontrado. Na seção 4, será realizado um breve estudo sobre as principais licenças de publicação. Finalmente, na seção 5, será tratado como empresas podem fazer o uso de *software* livre: tanto no lado cliente quanto no lado empresas produtoras. A seção 6 fará uma discussão sobre o futuro da distribuição dos *softwares* e como o *software* livre se encaixa nessa realidade futura.

## 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 [1]).

### 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 (SaaS), *software* como um serviço (SaaS) e os componentes da produção de *software*.

### 3.1 Software as a Product

*Software as a Product* é a referência ao *software* oferecido como produto. Pensando-se em *software* proprietário, um exemplo seria a compra de uma licença de uso de um produto como o Microsoft Office, em que a compra proporciona o uso do *software* sem custos adicionais, nem assinatura mensal. É a política do "comprei, é meu", limitada a um *software* que não tem código aberto.

De volta ao contexto de *software* livre, como exemplo de produto, há o Ubuntu, distribuição Linux desenvolvida pela Canonical. Ao instalar o sistema, o usuário tem o produto por quanto tempo quiser. Adaptando a política citada acima, para o *software* livre, o certo a dizer é então "baixei, é meu".

O Ubuntu é apenas um exemplo de produto. Há vários outros, como a maior parte das distribuições Linux, *softwares* comuns como o Firefox (Mozilla), e ferramentas avançadas de desenvolvimento, como Gimp (GNU), Eclipse e Netbeans (Oracle). Esses programas são de código aberto, e o usuário escolhe a versão que deseja, bem como as atualizações. No entanto, há o outro lado dos *softwares*: os que são feitos para estarem constantemente integrados e atualizados. Daí vem o conceito de *software* como um serviço.

### 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 SaaS

Ao pensar em *software* livre como um produto, do ponto de vista comercial, leva ao questionamento a respeito dos custos que uma empresa pode ter, e de como arrecadar fundos para cobri-los. Como já citado, gratuidade não é um pré-requisito de *software* livre. Sendo assim, há três principais formas de arrecadação das empresas no mercado atual.

A venda de softwares é uma das alternativas que desenvolvedoras como RedHat e Suse, por exemplo, adotaram para obtenção de lucros. O RedHat Enterprise Linux é a distribuição Linux desenvolvida pela empresa de seu nome. Possui foco empresarial, e como há todo um suporte exclusivo para isso, os custos são elevados. Sendo assim, é uma distribuição paga. O mesmo acontece com o Suse Linux.

Há também aquelas que se mantêm através de doações. A ideia é continuar oferecendo *software* livre de forma gratuita. No entanto, deve haver alguma forma para o sustento da empresa. Desenvolvedoras como a Canonical, por exemplo, adotaram então o sistema de doações. Ao fazer o *download* do Ubuntu, o usuário é direcionado a uma página sugerindo uma doação com o valor que achar que justo à empresa. Logicamente, isso é opcional.

A terceira alternativa de sustentação de uma empresa que lida com *software* livre, é o conjunto publicidade e parcerias. O correto a dizer seria que a publicidade vem através de parcerias. Essas últimas são comuns já que as desenvolvedoras costumam colocam *softwares* terceiros integrados aos seus, como uma forma de promovê-los e receber patrocínio em troca. É o caso por exemplo da Mozilla, que em 2011, recebeu *royalties*\*referencia aqui\* da Google por adotar o buscador como padrão do Firefox.

### 5.2 Software as a service

aqui vc faz

### 5.3 core

## 6 Tendências

olar

### Referências

1. 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)
2. 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)
3. Diday, E.: Introduction à l'analyse des données symboliques. Research report, Institut National de Recherche en Informatique et en Automatique, INRIA (1989)
4. Lutkebohle, I.: BWorld Robot Control Software. <http://www.nlm.nih.gov/research/visible/> - accessed 19-July-2008
5. 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.



**Italo Menezes** nasceu em Recife, capital de Pernambuco, Brasil. Graduando do curso de Ciência da Computação pela Universidade Federal de Campina Grande, participou do Programa de Iniciação Científica Jr. pela OBMEP em parceria com o IMPA e o CNPq. Fez curso técnico em Desenvolvimento para Web. Atualmente, é membro do projeto de capacitação da Sony do Laboratório de Sistemas Embarcados e Computação Pervasiva (Embedded Lab). Tem interesse em Engenharia de Software e Desenvolvimento para Sistemas Embarcados, com foco em Mobile e Web.