Introduction

Almost all computer science papers begin with a section entitled **Introduction**. Virtually every paper does this differently—it's individualized.

We introduce topics that are relevant, and identify how they are relevant. What information does the reader need to remember to understand the remainder of the paper?

- We include **a description of the bigger picture** but we pinpoint how this research fits in.
- We **make global definitions**: we give definitions and describe notation which are frequently used throughout the paper.
- We **cite relevant references**, so the reader knows where to learn more about related topics.
- We **discuss critical papers**: often papers are developed from one or a few other papers.
- We **describe what is new**.

We **don't assume the reader has read the abstract**.

Introduction: what not to include

- **Describe the new ideas** (what we do), **not the outcomes** (not: did it work?).
  - *Good*: The inherent fine-grain parallelism in [foo] motivates us to offload this task to a GPU which is suited to such computation, thereby relieving the CPU of this burden.
  - *Bad*: We offload [foo] onto a GPU, which reduces the overall run time by half.
- **Do not describe experimental results.** They belong in the section entitled "Experimental Results", and this is where the reader will look for them.
  (Exception: sometimes preliminary experimental results are part of the motivation.)

Introduction: caveats

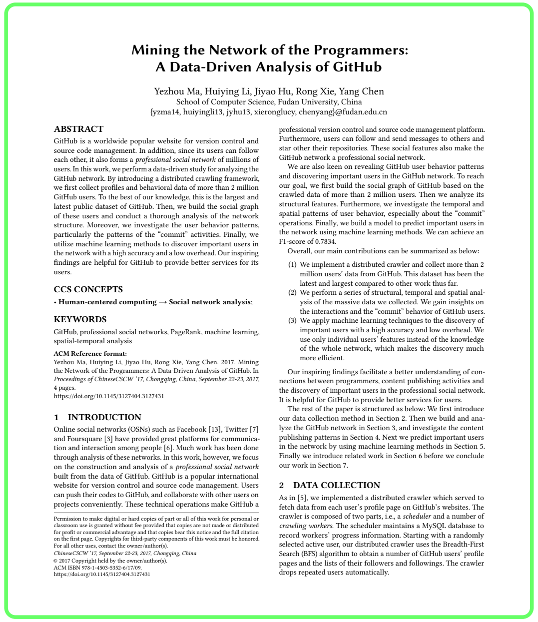
- The usual caveats apply:
  - different people (including referees) have different preferences, and
  - we break the rules if it helps communication.
- What to include in an Introduction, and how to write it is **mostly determined by what you want to say** (i.e., the content), and not by any template or algorithm.

General rules:

- The Introduction (as with everything else) should be **structured**.
  - Similar ideas are grouped together in paragraphs.
  - Related paragraphs occur next to one another, and may be grouped into sections or subsections.
- Generally, the Introduction (or a large part of it) should be **readable without much notation nor many definitions**.
- It should be designed so that **difficult paragraphs are identifiable and skippable**.

Again, I need to make a selection of topics to talk about.

Length of the introduction



Mining the Network of the Programmers:
A Data-Driven Analysis of GitHub

Yezhou Ma, Huiying Li, Jiyao Hu, Rong Xie, Yang Chen
School of Computer Science, Fudan University, China
{yzma14, huiyingli13, jyhu13, xieronglucy, chenyang}@fudan.edu.cn

ABSTRACT

GitHub is a worldwide popular website for version control and source code management. In addition, since its users can follow each other, it also forms a *professional social network* of millions of users. In this work, we perform a data-driven study for analyzing the GitHub network. By introducing a distributed crawling framework, we first collect profiles and behavioral data of more than 2 million GitHub users. To the best of our knowledge, this is the largest and latest public dataset of GitHub. Then, we build the social graph of these users and conduct a thorough analysis of the network structure. Moreover, we investigate the user behavior patterns, particularly the patterns of the "commit" activities. Finally, we utilize machine learning methods to discover important users in the network with a high accuracy and a low overhead. Our inspiring findings are helpful for GitHub to provide better services for its users.

CCS CONCEPTS

• **Human-centered computing** → **Social network analysis**;

KEYWORDS

GitHub, professional social networks, PageRank, machine learning, spatial-temporal analysis

ACM Reference format:
Yezhou Ma, Huiying Li, Jiyao Hu, Rong Xie, Yang Chen. 2017. Mining the Network of the Programmers: A Data-Driven Analysis of GitHub. In *Proceedings of ChineseCSCW '17, Chongqing, China, September 22-23, 2017,* 4 pages.
https://doi.org/10.1145/3127404.3127431

1 INTRODUCTION

Online social networks (OSNs) such as Facebook [13], Twitter [7] and Foursquare [3] have provided great platforms for communication and interaction among people [6]. Much work has been done through analysis of these networks. In this work, however, we focus on the construction and analysis of a *professional social network* built from the data of GitHub. GitHub is a popular international website for version control and source code management. Users can push their codes to GitHub, and collaborate with other users on projects conveniently. These technical operations make GitHub a professional version control and source code management platform. Furthermore, users can follow and send messages to others and star other their repositories. These social features also make the GitHub network a professional social network.

We are also keen on revealing GitHub user behavior patterns and discovering important users in the GitHub network. To reach our goal, we first build the social graph of GitHub based on the crawled data of more than 2 million users. Then we analyze its structural features. Furthermore, we investigate the temporal and spatial patterns of user behavior, especially about the "commit" operations. Finally, we build a model to predict important users in the network using machine learning methods. We can achieve an F1-score of 0.7834.

Overall, our main contributions can be summarized as below:

(1) We implement a distributed crawler and collect more than 2 million users' data from GitHub. This dataset has been the latest and largest compared to other work thus far.
(2) We perform a series of structural, temporal and spatial analysis of the massive data we collected. We gain insights on the interactions and the "commit" behavior of GitHub users.
(3) We apply machine learning techniques to the discovery of important users with a high accuracy and low overhead. We use only individual users' features instead of the knowledge of the whole network, which makes the discovery much more efficient.

Our inspiring findings facilitate a better understanding of connections between programmers, content publishing activities and the discovery of important users in the professional social network. It is helpful for GitHub to provide better services for users.

The rest of the paper is structured as below: We first introduce our data collection method in Section 2. Then we build and analyze the GitHub network in Section 3, and investigate the content publishing patterns in Section 4. Next we predict important users in the network by using machine learning methods in Section 5. Finally we introduce related work in Section 6 before we conclude our work in Section 7.

2 DATA COLLECTION

As in [5], we implemented a distributed crawler which served to fetch data from each user's profile page on GitHub's websites. The crawler is composed of two parts, i.e., a *scheduler* and a number of *crawling workers*. The scheduler maintains a MySQL database to record workers' progress information. Starting with a randomly selected active user, our distributed crawler uses the Breadth-First Search (BFS) algorithm to obtain a number of GitHub users' profile pages and the lists of their followers and followings. The crawler drops repeated users automatically.

ACM ISBN 978-1-4503-5352-6/17/09.
https://doi.org/10.1145/3127404.3127431

The Introduction should ordinarily be around **half a page to a page long**. (It varies.)

Moreover, it should be broken up (usually into Subsections, and additional Sections).

In this example, we can easily find:

(a) the **bigger picture** and (b) **what is new**.

Introduction

(Paragraph structure)

A typical well-structured first paragraph in a paper:

> Many security and counter-terrorism-related decision support applications need data mining techniques for identifying emerging behavior, link analysis, building predictive models, and extracting social networks. They often deal with multi-party databases/data-streams where the data are privacy sensitive. Financial transactions, health-care records, and network communication traffic are a few examples. Figure 1 depicts the data sources of a typical security screening application where the data may be privacy sensitive. Mining the data in such applications requires algorithms that are sensitive to privacy issues.
>
> — Dutta, et al., WPES, 2003.

- The first sentence gives **the broader picture**, which helps motivate the paper.
- The last sentence is what **what the reader needs to remember** from this paragraph.
- We put the details in between these two sentences. (But not too much.)

Paragraph structure

Most paragraphs in English writing have a particular structure (including in computer science).

- **Topic sentence.** (The first sentence.) What the paragraph is about.
- **Evidence and analysis.** (The middle bit—usually around 3 sentences.) The technical details, logical deductions, relevant references, and so on.
- **Conclusion.** (The last sentence.) This answers:
  - What is the final message?
  - Basically: *What does the reader need to remember?*

- *Poor writing*: **too-long paragraphs**. Lack of structure. (And they're really boring!!!)
- *Poor writing*: **too-short paragraphs**. Lack of substance.

How can we improve the topic sentence below?

> Testing a compiler traditionally relies on running it on a suite of hand-written test programs. This approach is problematic for two reasons. Firstly, because collecting a large number of suitable programs is difficult and such programs rarely cover all interesting cases of code [7, 10]. Secondly, because the compiler is tested against the same set of programs over and over again, which means that if a bug is not triggered by any of them, then it will never be found. Using random property-based testing is an alternative that could remedy both of these problems. However, this alternative requires automatic generation of test programs.
>
> — Pałka, et al., AST, 2010.

It does not explain that the paragraph talks about problems with the traditional method. We combine it with the second sentence:

> *Testing a compiler traditionally relies on running it on a suite of hand-written test programs, which is problematic for two reasons.*

We can likewise improve the conclusion.

What's wrong with starting a paragraph like this?

> However, most existing algorithms have one common character which leads to their common limitations. They detect the existence of frame deletion by an analyzing the side effect of it. Namely, the artifacts of RI. While the artifacts of RI are effective in some cases, they are not in the following situations. ...
>
> — Feng, et al., IH&MMSec, 2014.

Starting a sentence with "However, ..." implies it uses context from the prior sentence. In this case, that sentence is in an earlier paragraph. (There's also several grammar problems.)

We should restructure:

> Most existing algorithms detect frame deletion by analyzing its side effects, giving rise to a common limitation in the following situations. ...

How *not* to structure a paragraph

> **1. INTRODUCTION**
>
> Today's enterprises are facing serious cyber-threat posed by intrusive multi-step attacks such as Advanced Persistent Threat (APT) attacks. It takes much time for attackers to gradually penetrate into an enterprise, to understand its infrastructure, to own the high-value targets, and to steal important information [1, 6, 2, 9, 10, 12] or to sabotage mission critical infrastructures [11]. Compared with conventional attacks, sophisticated multi-step attacks such as APT attacks usually inflict much more severe damage upon the enterprises' business. The recent DARPA Transparent Computing (TC) program [13] emphasizes that the challenges of multi-step attacks come from modern computing systems being large, complex and opaque, and the threats can remain undetected for years if individual steps are buried in background "noise." Thus, to counter these sophisticated attacks, enterprises are in great need of solutions to "connect the dots" across multiple activities that individually might not be suspicious enough. Even though these attacks can be powerful and stealthy, one typical constraint from the attacker side is that such an attack needs multiple steps to succeed, such as infrastructure reconnaissance and target discovery, as illustrated in the cyber kill chain [5]. Therefore, multiple attack footprints might be left as "dots."
>
> — Xu, et al., CCS, 2016.

Identify the topic sentence.

Identify the conclusion.

It needs restructuring.

Should avoid metaphors (e.g. "connect the dots").

> 12:2 Y. Zhang and D.-Y. Yeung
>
> the source tasks have been learned using some symmetric multitask learning method. In this sense, asymmetric multitask learning is related to *transfer learning* [Pan and Yang 2010]; however, the major difference is that the source tasks are still learned simultaneously in asymmetric multitask learning but are learned independently in transfer learning.
>
> Major advances have been made in multitask learning over the past decade, although some preliminary ideas actually date back to much earlier work in psychology and cognitive science. Multilayered feedforward neural networks provide one of the earliest models for multitask learning. In a multilayered feedforward neural network, the hidden layer represents the common features for data points from all tasks, and each unit in the output layer usually corresponds to the output of one task. Similar to the multilayered feedforward neural networks, multitask feature learning (MTFL) [Argyriou et al. 2008a] also learns common features for all tasks but under the regularization framework. Unlike these methods, the regularized multitask support vector machine (SVM) [Evgeniou and Pontil 2004] enforces the SVM parameters for all tasks to be close to each other. Another widely studied approach for multitask learning is the task clustering approach [Thrun and O'Sullivan 1996; Bakker and Heskes 2003; Xue et al. 2007; Kumar and Daumé III 2012]. Its main idea is to group all tasks into several clusters and then learn similar data features or model parameters for the tasks within each cluster. An advantage of this approach is its robustness against outlier tasks, because they reside in separate clusters that do not affect other tasks. As different tasks are related in multitask learning, model parameters of different tasks are assumed to share a common subspace [Ando and Zhang 2005; Chen et al. 2009], and to deal with outlier tasks that are not related with other remaining tasks, other methods [Chen et al. 2010; Jalali et al. 2010; Chen et al. 2011] assumed that the model parameter matrix consists of a low-rank part to capture the correlated tasks and a structurally sparse part to model the outlier tasks. Moreover, some Bayesian models have been proposed for multitask learning by using Gaussian process (GP) [Yu et al. 2005; Bonilla et al. 2007], t process [Yu et al. 2007; Zhang and Yeung 2010b], Dirichlet process [Xue et al. 2007], Indian buffet process [Rai and Daumé III 2010; Zhu et al. 2011; Passos et al. 2012], and sparse Bayesian models [Archambeau et al. 2011; Titsias and Lázaro-Gredilla 2011]. Different from the preceding global learning methods, some multitask local learning algorithms are proposed in Zhang [2013] to extend the k-Nearest-Neighbor algorithm and the kernel regression method. Moreover, to improve the interpretability, the multitask feature selection methods [Obozinski et al. 2006; Obozinski et al. 2010; Zhang et al. 2010] are to select one subset of the original features by utilizing some sparsity-inducing priors (e.g., l_1/l_p norm ($p > 1$)). Most of the preceding methods focus on *symmetric* multitask learning, but there also exist some previous works that study *asymmetric* multitask learning [Xue et al. 2007] or transfer learning [Raina et al. 2006; Kienzle and Chellapilla 2006; Eaton et al. 2008; Zhang and Yeung 2010c, 2012].
>
> Since multitask learning seeks to improve the performance of a task with the help of other related tasks, a central issue is to characterize the relationships between tasks accurately. Given the training data in multiple tasks, there are two important aspects that distinguish between different methods for characterizing the task relationships. The first aspect is on *what* task relationships can be represented by a method. Generally speaking, there are three types of pairwise task relationships: positive task correlation, negative task correlation, and task unrelatedness (corresponding to outlier tasks). *Positive task correlation* is very useful for characterizing task relationships, because similar tasks are likely to have similar model parameters. For *negative task correlation*, since the model parameters of two tasks with negative correlation are more likely to be dissimilar, knowing that two tasks are negatively correlated
>
> ACM Transactions on Knowledge Discovery from Data, Vol. 8, No. 3, Article 12, Publication date: May 2014.
>
> — Zhang and Yueng, ACM T. Knowl. Discov. Data, 2014.

This is a page of the Introduction of this paper.

The paragraph is gigantic!! It takes up nearly the whole page.

It desperately needs structure! It should be **redesigned to be read by a human**.

> Many LP detection algorithms have been proposed in the past. Most of these methods make use of LP features, like shape [14], background color [16], or characters [15]. And the biggest difference among these methods are feature representations [such] as saliency-related features [10], MSER [5], CSER [12], wavelet [11] etc.. Some of them use rule based method to locate the LP with those features and other use machine learning based, like SVM [7], Adaboost [19], neural network [13] etc.. The precision of the resulting bounding box with ground truth, reported in [10] with 75% and [8] with 77%, has significant influence on performance of character segmentation and recognition in LPR system. However it is difficult to get precise bounding box of LP.
>
> — Cao, et al., ICIMCS, 2014.

(Here LP = "license plate".)

- *Good*: Clear topic sentence and conclusion.
- *Good*: Tangentially related references described succinctly.
- *Poor*: Too precise for the introduction: "75%".
- *Grammar quibble*: Ending a sentence with "etc..". Either write "and so on" or end with "etc." (One period.)

What's wrong with this in the Introduction?

> To validate the efficacy of our proposed approach, we conduct a comprehensive evaluation on real-world auditing systems using log traces of more than one month. Our evaluation results demonstrate that the CPR algorithm is general and does not lose any accuracy by design, which can improve the capacity of a big-data system by up to 3.4 times. In other words, the same system can store 3.4 times more data without affecting storage and data processing efficiency. With a trade off of introducing very few false positives (in our evaluation at the rate of 0.2%), the PCAR algorithm can enlarge the capacity of the system by up to 5.6 times. We also compared our approach with a naive time-window-based data aggregation. We show that the naive approach introduces many more false positives at the rate of 13.9%. Without considering manually tuning our approach for each application in an enterprise environment, we achieve a similar reduction ratio as LogGC, but our solution can be combined with LogGC to achieve more significant data reduction.
>
> — Xu, et al., CCS, 2016.

We **don't present experimental results** in the Introduction. Delete! (If we must say something: "Experimental results are given in Section 5.")

> While SRAM-based FPGA suffer from long configuration-loading time and excessive leakage power during stand-by, FPGAs using non-volatile memories (NVM) have emerged as a promising alternative [1]. Non-volatile memory eliminates the necessity of loading configuration from off-chip storage, for it preserves the configuration information stored on-chip while powered off, allowing the devices to immediately run at power-up. In addition, the immunity to soft errors also makes such NVM-based FPGA attractive for mission-critical aerospace applications [1]. Consequently, FPGAs using flash to store configuration bits have already been in the market by Actel and Lattice [1]. However, low logic density, inadequate performance, and the lack of bit-level programmability prevent flash memory from being the universal memory replacement in FPGAs. Recently, new non-volatile memory technologies including magnetic RAM (MRAM), Ferroelectric RAM (FeRAM), and phase change RAM (PCRAM) have been intensively explored [2]. These candidates generally provides high logic density and moderate to high performance compared with existing technologies. However, the manufacture of these new memories usually requires new materials and separate processes which complicate the fabrication of FPGAs. A low-cost integration technique is thus needed to incorporate these new non-volatile memories in conventional FPGAs.
>
> — Chen, et al., ISLPED, 2010.

Long, boring paragraphs need to be skippable. Here, the reader can read the topic sentence and conclusion, and skip the middle part.

Introduction

(Reference-dense sentences)

Reference-dense sentences from an earlier example (Cao, et al., 2014):

> Most of these methods make use of LP features, like shape [14], background color [16], or characters [15]. And the biggest difference among these methods are feature representations [such] as saliency-related features [10], MSER [5], CSER [12], wavelet [11] [and so on]. Some of them use rule based method to locate the LP with those features and other use machine learning based, like SVM [7], Adaboost [19], neural network [13] [and so on].

(Note: "[...]" indicates a correction.)

I encourage writing **some reference-dense sentences** like this in the Introduction (or Background, or Related Work):

- It demonstrates the authors are aware of related literature. (Breadth.)
- It indicates how the paper fits into the literature.
- It's more informative than saying "relevant references are [5,7,12,15]".
- The reader might be particularly interested in one of these papers.
- It's written so that the disinterested reader can easily skip ahead.
- Each citation adds links to your paper in databases. (SEO.)

Critique these reference-dense sentences

> Prior work [15], [6], [7], [13] tend to associate big data with big science, which is also characterized with big organization and big budget.
> — Xie, et al., JCDL, 2015.

It gives the reader no idea what the papers "[15], [6], [7], [13]" are about. (And they're out of order.)

> This technique has been extended to handle different related problems, including database services in wireless broadcast environments [44,45]; high-dimensional query evaluation [7]; continuous location-based services [4,32,43]; and virus spread analysis among mobile devices [41].
> — Xie, et al., VLDB J., 2013.

Good use of reference-dense sentences: it's a list of succinctly described examples with concrete references.

Introduction

(Major contributions)

Some introductions contain a paragraph which lists of **major contributions**.

> The main contributions of the paper include: (1) According to the characteristics of web services, through the analysis of business processes' structure, we find exchangeable relationship between requests; (2) We describe how to use pipeline concurrency and exchangeability between requests to improve the Paxos's performance; (3) We have implemented some experiments and experimental results show its effectiveness and performance advantages compared with other replication methods.
> — Zhang and Wang, Internetware, 2013.

I feel like this is a trade-off:

- *Bad*: It's **a meta paragraph**. We describe what the paper contains, rather than just giving the contents.
- *Good*: It's helps the reader **identify what is new**. (Particularly helpful for referees.)
- *Good*: It's **easily skipped** if the reader does not want to read it.

Consequently, I feel the positives outweigh the negatives.

> *as the popularity of BDS related jobs reaches the peak in the 2nd half of 2014, and decreases in 2015.* Generally, the contributions of this paper can be summarized as follows.
>
> - To the best of our knowledge, this paper is the first attempt to leverage unsupervised learning approach for automatically modeling the trend of recruit market. This work provides a new research paradigm for recruitment market analysis.
> - We propose a sequential latent variable model, named MTLVM, for learning the latent recruitment states, demands, and topics simultaneously. Particularly, MTLVM can dynamically generate recruitment topics by integrating hierarchical dirichlet processes.
> - We develop a prototype system for empirically evaluate our approach. Indeed, by visualizing the results obtained from MTLVM, we can successfully observe many interesting and useful findings.
>
> — Zhu, et al., KDD, 2016.

An example of a "major contributions" summary.

- *Good*: "To the best of our knowledge". (Better is: "To the best of the present authors' knowledge". [Who is "our"?])
- *Good*: Bullet points give it structure—easy to understand.
- *Poor*: Acronym MTLVM exceeds margin.
- *Grammar quibbles*: "leverage unsupervised learning approach", and "Particularly" and "for empirically evaluate".

Introduction

(Table-of-contents paragraph)

Many computer science papers end the Introduction with e.g.:

> The rest of the paper is organized as follows. In Section 2, we provide an overview of related work on Wikipedia contribution motivations, contribution behaviors, and article quality. The two datasets used in this paper are described in Section 3. The network evolution in Wikipedia compared to TKB is presented in Section 4, followed by the analysis of Wikipedia contribution pattern in Section 5. Section 6 concludes this paper.
>
> — Zhang, et al., JCDL, 2010.

- This paragraph functions as a *table of contents*.
- Personally, I think writing this is pointless—we should instead choose meaningful section titles.
  - "Section 6 concludes this paper." Two points: (a) "Concludes" also means "brings to an end" (not necessarily "gives conclusions"). (b) It's named "Conclusions"!

  However...
- I've encountered reviewers who objected to it being omitted.

Thus, I (begrudgingly) recommend including it. It's easily skipped.

What's the pros and cons of this paragraph?

> In the remainder of the paper, we describe the compiler architecture in §2, optimizations in §3, evaluation results in §4, and related work in §5.
>
> — Wu, et al., CGO, 2016.

- *Con*: It's a one-sentence paragraph, and "in the remainder of the paper" is redundant (where else is e.g. Section 2 going to be?).
- *Pro*: It's succinct!
- *Con*: We should avoid notation "§2", etc., in favor for words "Section 2"—the reader might not know what § means.

Introduction

(Background)

In the introduction, we might give information such as the following:

- References to and descriptions of other approaches.
- Definitions of terms, notation, acronyms, and so on, that are used throughout the paper.
- A table of notation, if the paper contains a lot.

How to structure this varies; if there's a lot of material, it's probably best to create a separate sections or sub-sections, such as:

- Related Work
- Terminology
- Background
- Definitions
- Preliminaries
- Notation (Not "notations".)
- Or some combination of these, e.g. "Background and Notation" or "Definitions and Notation".

Real-time Motion Effect Enhancement
Based on Fluid Dynamics in Figure Animation

Tian-Chen Xu*‡ En-Hua Wu†‡§ Mo Chen‡ Ming Xie‡

‡Faculty of Science and Technology, University of Macau, Macao, China
§State Key Lab of CS, Institute of Software, Chinese Academy of Sciences, Beijing, China

Abstract

In fast figure animation, motion blur is often employed to generate fantastic effects of figure motion, for exaggerating the atmosphere one wants to convey. In the previous works for long time, the solution based on certain kind of image blending in terms of hardware or software, was simply adopted. In order to provide high standard of motion blur effect, methods based on 3D geometry of the motion figure with global illumination become gradually in demand. However, the computation cost to meet such demand is very high and it is hard to achieve real time rendering.

In our work, a novel 3D geometric approach of real-time motion effect is proposed. By the approach, a special effect along the motion trajectory based on fluid simulation is combined with the volumetric motion blur. Besides, in order to avoid the redundant calculation of each frame and break the limitation of trajectory generation, we decompose the motion trajectory and employ multi-pass geometry rendering to achieve geometry instancing for reuse. In the pipeline, we separate motion tracking and fluid solution, to support various fluid effects flexibly. As our algorithm is constructed in the context of GPU geometry shading in parallel, high efficiency of computation is guaranteed while realizing gorgeous rendering. As a result, real time rendering including the motion blur effect is achieved.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Radiosity;

Keywords: motion blur, skeletal animation, fluid dynamics, GPU geometric processing

1 Introduction

When the pursuit of realistic detail and artistic quality becomes even stronger in figure animation, motion blur becomes an indispensable part of animation special effect. In various kinds of rapidly moving environments like the extreme speed of a racing car, the vibration of an explosion airstream or just the dizziness one character feels, motion blur can make them all more expressive with convincing particulars. With the help of GPU computation, the trails of moving objects can be analyzed to provide better continuity and smoothness for live-action footage.

Making fast-moving scene more realistic has always been the main purpose of motion blur technique. Originally, it was not realized by simply inserting data into adjacent frames, but by merging individual frame with its prior frame. In using the most popular tools of DirectX 9 and OpenGL 2.0, the outcome did not altogether satisfy the viewers: the moving objects and their background are often mixed together, leading to the result of image quality downgrading. Furthermore, its efficiency is also rather poor. Due to all these reasons, motion blur is not quite as popular.

In our paper, a novel approach of real-time motion effects based on geometry volume is proposed, which presents motion blur with well quality of illumination in 3D space, by taking advantage of modern GPU capability. To prevent from the over-duplicated calculation of each frame, we decompose the trajectory into segments, and employ multi-pass geometry rendering to achieve geometry instancing for reuse. Besides, a special calculation in real time based on fluid dynamics is performed to enhance the post-motion effect.

The structure of our paper is as follows: after briefly discussing related works on the topics concerning our study interest in the Section 2, we show the main principles of our method, with descriptions of the overall algorithms in Section 3. Some key solutions for implementation and the testing results in varying complexity will be presented in Section 4&5. Finally, we conclude in Section 6 with some discussions of the future work.

2 Related Work

In the early days, motion blur is often achieved with the help of OpenGL accumulation buffer. Haeberli and Akeley [1990] provided in-depth analysis towards the use of this architecture. Using ray tracing to perform the Monte Carlo evaluation of integrals [Cook et al. 1984] in the rendering equations, the problems of motion blur, depth of field, and penumbra can be solved. As the accumulation buffer is separated from the normal rendering hardware, the performance of the hardware was unable to achieve the requirement as demanded in this aspect. However, since GPU has come to our sight nowadays, accumulation buffer was gradually abandoned.

Brostow and Essa [2001] present a postprocess approach to simulate motion blur automatically. They first track the motion within the image plane, and then integrate the changing scene as the time elapses. In this manner, a better support for live-action footage and smoothness can be accomplished. The problem is, image-based motion blur can only provide the most basic effect, without sufficient flexibility. Some powerful or special effect such as the fluid dynamics we proposed in this work cannot be integrated with it.

A framework for elliptical weighted average (EWA) surface splatting with time-varying scenes was introduced by Heinzle et al. [2010]. They use a piecewise linear approximation to construct a rendering algorithm for point-based objects. In the context of point-sample geometry, 3D Gaussian kernel rather than 2D Gaussian kernel is employed to unify a spatial and temporal component for motion-blurred images, and the change makes sure the continuity in both space and time.

Since Microsoft released a sample "Motion Blur 10" in DirectX 2007, a geometric approach came into the world with the birth of

*e-mail: X-Universe@live.com
†e-mail: EHWu@umac.mo

VRCAI 2011, Hong Kong, China, December 11 – 12, 2011.
© 2011 ACM 978-1-4503-1060-4/11/0012 $10.00

This is an example where the Introduction is short.

We have a Related Work section—it's still introductory material, but it's bundled together based on commonality.

What's important is:

(a) there's **structure**, i.e., it's designed to be read, and

(b) What's **new** is made obvious.

Related Work

Different approaches to the same problem:

- Use **at most one sentence** per reference; sometimes they can share sentences.
- **Describe their ideas**, not their performance. (Do not restate their experimental results! Do not restate their salesmanship!)
- Write in **simple past tense** to describe prior work.
- **Group them together** based on commonality (e.g., hardware solutions vs. software solutions).
- Most importantly, **describe how each group is related to and differs from your work!!**
  ("We instead use [blah] for [blah].")

Related approaches to the same problem:

- We **describe these in detail**: maybe a few sentences to a whole paragraph.
- **Identify the differences** between these references and your work.

Example Related Work paragraph:

2 Related Work

Fully automated related work summarization is significantly different from traditional summarization. While there are no existing studies on this specific problem, there are closely related endeavors. The iOPENER[2] project works towards automated creation of technical surveys, given a research topic (Mohammad et al., 2009). Standard generic multi-document summarization algorithms **were applied** to generate technical surveys. They **showed** that citation information was effective in the generation process. This was also **validated** earlier in (Nakov et al., 2004), which **showed** that the citing sentences in other papers can give a useful description of a target work.

— Hoang and Kan, COLING, 2010.

We identify:

- More emphasis on iOPENER than Nakov et al., 2004.
- Simple past tense for describing prior work. (Except for iOPENER.)
- (The [2] is a footnote; it's confusing to write "iOPENER[2]".)

The next paragraph:

Other studies focus mainly on single-document scientific article summarization. The pioneers of automated summarization (Luhn, 1958; Baxendale, 1958; Edmundson, 1969) ~~had~~ **envisioned** their approaches being used for the automatic creation of scientific summaries. They **examined** various features specific to scientific texts (e.g., frequency-based, sentence position, or rhetorical clues features) which **were** ~~**proven**~~ effective for domain-specific summarization tasks.

— Hoang and Kan, COLING, 2010.

We identify:

- How the present paper differs from the cited papers.
- The paragraph groups together references by some commonality; in this case, being "pioneering" works.
- We don't describe performance—we describe the relevant ideas.
- Simple past tense to describe prior work ("were proven" essentially means "were proved").

The next paragraph:

Further, Mei and Zhai (2008) and Qazvinian and Radev (2008) utilized citation information in creating summaries for a single scientific article in computational linguistics domain. ~~Also,~~ Schwartz and Hearst (2006) also utilized the citation sentences to summarize the key concepts and entities in bioscience texts, and argued that citation sentences may contain informative contributions of a paper that complement its original abstract.

— Hoang and Kan, COLING, 2010.

We identify:

- At most one sentence per tangentially related papers.
- References are grouped together based on commonality; this time the utilization of citation information.
- Only the ideas are described (not the performance).

The next paragraph (and the last paragraph):

These works all center on the role of citations and their contexts in creating a summary, using citation information to rank content for extraction. However, they did not study the rhetorical structure of the intended summaries, targeting more on deriving useful content. For working along this vein, we turn to studies on the rhetorical structure of scientific articles. Perhaps the most relevant is work [is] by (Teufel, 1999; Teufel and Moens, 2002) who defined and studied argumentative zoning of texts, especially ones in computational linguistics. While they studied the structure of an entire article, it is clear from their studies that a related work section would contain general background knowledge (BACKGROUND zone) as well as specific information credited to others (OTHER and BASIS zones). This vein of work has been followed by many, including Teufel et al. (2009); Angrosh et al. (2010).

— Hoang and Kan, COLING, 2010.

We identify:

- How the present paper relates to the cited papers.
- More about Teufel et al.: more discussion $\Longleftrightarrow$ more important.

Introduction

(Definitions and notation)

Sometimes we create a separate section or subsection for definitions and notation. Either way, the reader needs to know **where to look for definitions**.

For example:

> It is important to understand the key *figures of merit* (i.e., the objectives and constraints in the optimization) and the *knobs* (the variables) that can be used in disk drive optimization. The figures of merit include performance (both throughput and latency), power, form factor, capacity, cost etc. Of the available knobs, some are usable at design time (static knobs) and others can potentially be varied at runtime (dynamic knobs). Static knobs include the number of platters and their size, the characteristics of the spindle motor (SPM) and the voice coil motor (VCM). Dynamic knobs include the voltages for the SPM and VCM, which can be used to trade off performance and power by slowing down or speeding up the platter rotation and the seek time.
>
> — Zhang, et al., DAC, 2007.

Here, when the reader later forgets what "figures of merit" or "knobs" means, they can flick back and easily find where its defined.

Acronyms don't need to be put in italics when they're defined—they stand out already. For example

> However, factors other than complete disk failures influence the reliability of data, often expressed as the mean time to data loss (MTTDL).
>
> — Bairavasundaram, et al., SIGMETRICS, 2007.

Here, the acronym is "MTTDL".

Proper nouns don't need to be put in italics when they're defined—they stand out already. For example

> We propose MetroSense, a network architecture for urban-scale people-centric sensing with a design goal of broad application and sensor heterogeneity support. MetroSense provides the network architecture that is lacking in current urban-scale pervasive systems. ...
>
> — Campbell, et al., WICON, 2006.

Here, the proper noun is "MetroSense".

What's wrong with this?

> We also systematically analyze risks that originate with NVIDIA GPUs, where the GPU serves as a host for *stealthy*, *long-lived* malicious code. It is difficult to detect the execution of GPU-hosted malware and in certain cases, it is even difficult to detect its presence. We demonstrate attack code running on the NVIDIA GPU that reads secrets from CPU memory and corrupts the memory state of CPU computations by leveraging GPU Direct Memory Access (DMA) capabilities.
>
> — Zhu, et al., GPGPU, 2010.

What's in italics is not to highlight something being defined. It's not the appropriate use for italics in a paper.

What's wrong with this?

> Heterogeneous systems, comprising a CPU (central processing unit) and a GPU (graphics processing unit), that are both capable of general-purpose computations have started to emerge. A matching heterogeneous workload could take advantage of both the high computing power of GPUs for SIMD-friendly multithreaded computation and the high single-thread performance of CPUs.
>
> — Zakharenko, et al., DATE, 2013.

- The acronym "CPU" is expanded. Already familiar to the reader.
- The acronym "SIMD" is not expanded. (Less familiar than "GPU" and "CPU", which are expanded.)
- It is usual to write "graphics processing unit (GPU)" rather than "GPU (graphics processing unit)".

Introduction

(Words to learn and unlearn)

Learn this word: Problematic

"Problematic" is a gentle word for pointing out there is a problem.

- We don't outright say something is "wrong", "faulty", and so on.
- We don't specify the problem (useful when it's complicated).

Examples:

> Interacting with the World Wide Web is often **problematic** for many users that struggle, amongst many others, with unmet expectations, information overload, poor information architectures and substandard usability.
>
> — Vigo and Harper, WebSci, 2013.

> However, when examining existing embodied learning systems closely, we find a large breadth of designs that result in seemingly unrelated systems (see Figure 1). This becomes **problematic** when trying to understand where and how embodiment occurs in these systems, and which design elements help to facilitate embodied learning. The problem is further aggravated by ...
>
> — Melcer and Isbister, CHI EA, 2016.

Unlearn this word: Meanwhile

"Meanwhile" means "at the same time" (literally). I would go so far as to say it emphasizes two things happening at the same time.

How it might be used correctly:

> The CPU processes [something]. **Meanwhile**, the GPU processes [something else].

This has the same meaning as:

> The CPU processes [something]. **At the same time**, the GPU processes [something else].

Here, it's important emphasize that two things are happening in parallel.

Unlearn this word: Meanwhile (cont.)

The word "meanwhile" was used repeatedly for jokes in the US TV series *The Nanny*.

For example:

> **Fran**: *Ma, she worked in the lunch room. And she got fired for giving free lunches to a lady who tried to pass herself off as an 8th-grader.*
>
> **Sylvia**: *Meanwhile, with my little pigtails and my little plaid skirt, I was adorable.*

Here, the word "meanwhile" makes it clear that **Sylvia** was the "lady who tried to pass herself off as an 8th-grader". (This is the joke.)

Unlearn this word: Meanwhile (cont.)

> Lately, several approaches have been presented for detecting frame deletion. The most fundamental work was done by Wang and Farid[17]. They pointed out that after frame deletion, periodical spikes of the relocated I-frame (RI) will be observed on the mean motion residual sequence. ~~**Meanwhile**~~ **At the same time**, the magnitude of the Fourier transformation of the sequence can be inspected for peaks in the middle frequency.
>
> — Feng et al., IH&MMSec, 2014.

Here, I replace "Meanwhile" with "At the same time" to emphasize what it says:

- Wang and Farid pointed out [foo], and
- "the [blah] can be inspected".

We are not only saying this is true (which it's not), we're also saying it is important that they happen at the same time (which it's not).

Here "meanwhile" should be "moreover".

Be careful with this phrase: Previous works

Is this correct?

> In most of **the previous works**, researchers only focused on the popularity variation between days.
>
> — Wang, et al., HotPOST, 2016.

No! Here, "the previous work" includes everything—it's uncountable. Correct is:

> In most **previous work**, ...

Unlike "work" by itself, the phrase "work on this topic" can be countable or uncountable:

> In most of **the previous works on this topic**, ...

> In most of **the previous work on this topic**, ...

Be careful with this phrase: Previous works (cont.)

Is this correct?

> It should be noted that **many previous works on spammer detection** [Chirita et al. 2005; Cao et al. 2012; Xue et al. 2013] assume that legitimate users favor only other legitimate users.
>
> — Fu, et al., ACM T. Intel. Syst. Tech., 2017.

Yes! Here, "works on spammer detection" is countable.

However, this is also correct (and has the same meaning):

> It should be noted that **most previous work on spammer detection** [...] assumes ...

Previous work is most likely correct; "previous works" is sometimes wrong. A straightforward alternative is **previous studies**.

Unlearn this word: Besides (instead of "Moreover")

There's nuance to starting a sentence with the word "Besides".

> *I don't have time to go to the dentist. Besides, I'll be in Singapore on that date.*

Starting a sentence with "Besides, ..." is like starting a sentence with "Anyway, ...": it's often interpreted as rendering the previous sentence moot.

- ABC. Moreover XYZ. (Both ABC and XYZ are important.)
- ABC. Besides XYZ. (XYZ is sufficient: ABC is less significant.)

"Besides" is informal, and more suitable in conversation.

I recommend to simply not use the word "besides". Think of "Moreover" as the default. (And "Further" or "Furthermore", and "In addition to" or "Additionally", as alternatives.)

(If you Google this, you'll find a zillion contradictory explanations.)

Unlearn this word: Besides (cont.)

> It can interact with users in real-time to suggest several candidate APIs by analyzing their entered content. **Besides**, a list of frequently used APIs is provided for fast selection.
>
> — Chen, et al., VLDB, 2009.

The closest synonym for "besides" here is "moreover", but it's unnatural.

> It interacts with users in real-time to suggest several candidate APIs by analyzing their entered content, **and** it provides a list of frequently used APIs for fast selection.

Why not just use "and"? (This also resolves the passive second sentence.)