Lan, Xiangqi, Jiang, Xiang, Blunt, Liam and Xiao, Shaojun

An adaptive system framework for surface characterisation.

Original Citation

Lan, Xiangqi, Jiang, Xiang, Blunt, Liam and Xiao, Shaojun (2009) An adaptive system framework for surface characterisation. In: Proceedings of Computing and Engineering Annual Researchers' Conference 2009: CEARC'09. University of Huddersfield, Huddersfield, pp. 142-146. ISBN 9781862180857

AN ADAPTIVE SYSTEM FRAMEWORK FOR SURFACE CHARACTERISATION

X. Lan, X. Jiang, L. Blunt And S. Xiao
Centre for Precision Technologies
Computing and Engineering, University of Huddersfield, Queensgate, Huddersfield HD1 3DH, UK

ABSTRACT

Surface features have been recognised to be significant in many fields and the characterisation techniques are being constantly improved. The surface characterisation systems are developed by various instrument companies and institutions. However, the lack of good extensibility, reusability and maintainability is a serious obstacle to the system innovation. Component based software development offers a promising solution to the surface characterisation system with better extendibility and maintainability. This paper presents an overview of surface characterisation techniques, analyzes current surface characterisation systems, and then proposes a component based architecture for the surface characterisation system.

Keywords Surface Characterisation, Component Based Software Development, Extensibility

1 INTRODUCTION

Surface texture and its measurement are becoming more and more critical and important in the field of high precision engineering and nano-technology (Jiang, Scott et al. 2007). Surface roughness is an important factor in determining the satisfactory performance of a workpiece (Whitehouse 1997), for example in tribology and coatings. Also it has been found to be useful in machine tool monitoring. Surface metrology is the science of measuring geometrical features on surfaces. From the occurrence which can be traced back to the 1930s up to now, the surface metrology has undergone a huge paradigm shift. The study of surface metrology is becoming more and more commonplace in industrial and research environments (Conroy and Armstrong 2005).

Most of current surface characterisation systems are implemented based on the object-oriented technologies. The reusable and extendible of the system are limited even the architecture is well designed. During the last decades, with the advancement of computer technology, the paradigm of software development has also shifted. From the beginning of process-oriented, through object-oriented later, to the current aspect-oriented and component-oriented, software engineers have been working to simplify the development, implementation, and maintenance of software applications (Papajorgji, Beck et al. 2004). At present, object-orient programming is still widely used in the software development. However, it is not suitable to establish a flexible architecture. Component based software development (CBSD) is the successor of object-oriented software development, and it has become an increasingly popular approach to facilitate the development of evolving systems as it promised to address some of the problems of object-oriented development technologies. Component based software development is adopted to establish an adaptive system framework for surface characterisation, that could satisfy the indefinitely requirements, and the frequent changes and extension without affecting the stability of the system.

2 SURFACE CHARACTERISATION TECHNIQUES

Surface analysis and characterisation are as important as the measurement. With the improvement of the precision and accuracy in the measurement, the characterisation technology of the surface must be in simultaneous evolution. The common surface analysis method is comprised of fitting, filtering, parameterisation and some special analysis such as bearing ratio, autocorrelation, power spectral and so on. All these analysis methods are aimed to find out the surface features which we are interested in and the analysis result could reflect the surface characteristics in some prospective.

A set of discrete data acquired from a metrology instrument is the concrete representation of the surface texture. Such original measurement data itself is considered to be representation of surface features. As the reconstruction of the surface is from the measurement data on the chart, the judgement can be made from the magnified representation of the surface. However, that may be too complex to characterise the measured surface, it utilises too many data to represent the surface and it is very difficult to make a judgement. On the other hand, this is a qualitative analysis method. The result of 'good or bad' is too simple, and does not provide enough assessment information. To represent the surface texture more accurately and effectively, quantitative analysis is in need. Parameterisation of the surface texture is much more convenient for evaluation.

Generally it is nonsense to calculate the parameters from the original measurement data. When measuring a surface, even after careful alignment with the measurement instrument, there is still some residual slope and slight curve. So fitting is supposed to be the first step of the process which can rectify the measurement setup. Figure 1 illustrates the first order polynomial fitting of a sample data.

After pre-processing by fitting, the subsequent step is filtering. Filtering is very important in surface analysis. The action of filtering is to extract the significant or momentous information from the measurement data for further analysis (Jiang, Scott et al. 2007). The primary surface topography can be divided into form, waviness and roughness by filtering; an example is shown in figure 2. As the surface feature of interest may be different and the requirements of the extraction can also be diverse, taking advantage of the suitable filter with right parameters can contribute to effective extraction.

Parameter calculation is based on processed data by fitting and filtering. Most of the 2D parameters are defined in ISO 4287, while the 3D parameters are defined in ISO 25178 (Blateyron 2006). Different parameters reflect the surface texture from different perspective. There is no universal parameter that could represent the surface texture quite well in all cases, but there are numerous parameters that could take effect in particular applications.

Apart from parameterisation, there are some other analysis algorithms for the surface. For example, the bearing curve analysis is used for the lubrication and material analysis of surfaces. The autocorrelation function is used to reveal changes in random processes such as grinding. Power spectral analysis can be used in processes which are fundamentally periodic or repetitive, such as in turning and milling. Figure 3 illustrates an example of autocorrelation analysis.

3 CURRENT SURFACE CHARACTERISATION SYSTEMS

The first tools of analysing measurements are bound to instruments and are developed by the instrument manufacturers themselves. Until now, most of the surface characterisation software systems are still developed by the instrument companies (e.g. Taylor Hobson, Digital Surf and Veeco), and are combined to the instruments. These systems are mainly designed to satisfy the functions of the instruments. They are used to control the measurement and provide some tools to analyze the measured data. Apart from instrument companies, there are also many institutes researching on the surface characterisation and analysis. They develop their own characterisation systems for investigation and application, and append the outgrowth of the research to the characterisation system, so that the new methods and algorithms could be promoted to be used quickly. For example, NIST (National Institute of Standards and Technology) has developed an internet based surface metrology algorithm test system (Bui, Muralikrishnan et al. 2005).

Generally, all these characterisation systems have implemented those common analyses for surfaces. Each system has its own strengths and weaknesses compared with others. It is hard to decide if any one system is much better than the others. In the realm of surface metrology, all the strengths should be combined together rather than each organization keep the superiority on their own characterisation system. They should throw themselves on the characterisation technologies but the implementation of the software application. The disadvantages of these systems are classified as follows:

- Extensibility: Usually the characterisation system is implemented as a chunk of binary code, and each functional module is tightly coupled in the system framework. Modifying some existing code or appending some new code is required when extending the system. Thus, a great deal of detailed study is required to find out where the amendment is needed.

- Reusability: It is greatly limited to reuse the functional modules as the fine-grained (class) reuse, and the client have to be familiar with the inner structure. On the other hand, application bound and language restriction leads to the narrow scope of reuse.

- Maintainability: Excluding the system errors is especially difficult for the maintainers, they need to have intimate knowledge of the system architecture and modify the source code at the proper place.

Any code modification would cause the system application to be compiled, built and distributed again. Furthermore, with the increasing modification, the system may become instable.

4 COMPONENT BASED CHARACTERISATION SYSTEM

Although object-orient programming is still widely used in the software development, it is not suitable to establish an adaptive system application. Component based software development (CBSD) as the successor of object-oriented software development has become an increasingly popular approach to facilitate the development of evolving systems as it promised to address some problems of object-oriented development technologies (Grundy and Hosking 2002). And it can greatly improve the implementation efficiency and ease the extensibility and maintenance of large engineering software.

A component is an isolated element, a member of an external distributed composite system, and it interacts with system through a standard set of interfaces. Components which are reusable pieces of software code that serve as building blocks within an application framework should be designed to be independent units, much like Lego building blocks. They can be easily added to the system, and existing components can be detached and plugged into other systems. The services offered by components are made public by publishing their interfaces and contracts.

Component based software development has already been supported by commercial component frameworks such as COM (Component Object Model), CORBA (Common Object Request Broker Architecture) and Java/RMI (Remote Method Invocation) (Dolenc 2004). COM which is adopted to implement the characterisation system is a binary interface standard for software component introduced by Microsoft Corporation. The essence of COM is a language-neutral way of implementing objects that can be used in environments different from the one they are created in, even across machine boundaries.

There are many advantages to implement the characterisation system by using component based software development:

- Reduce the development time and cost. Component, as an independent module, is designed and implemented separately. Loosely coupled relationship between the components and mainframe allows the parallel development of the system. It is no doubt that the system development cycle would be shortened. On the other hand, many system functions can be achieved by reusing existing components rather than designing entirely new ones.

- Reduce the maintenance cost. Because of the encapsulation of the component, it is possible to add new components or replace or remove the existing ones without affecting all the system that relies on them. Regardless of any changes in the components, it is not necessary to recompile and reconfigure the system.

- Enhance the extendibility and diversity. Instead of being combined into the system statically, the component could be connected to the system framework dynamically. It means that the component can be plugged in and play at once and there is no need to take system architecture into account. Thus, the users could customize and update the system as they expected.

With the development of the measurement technology, the characterisation and analysis technology are innovating. The most important task is to take advantage of these analysis technologies to assess the surface texture. Based on these nature functional requirements, the entire system should be comprised of data access, analysis and process, surface display, system development kit and the system framework. Figure 5 is the roughly operation flow of the analysis.

Data access, analysis, display is the most important functional components of the whole system. They are designed to be independent pieces that implement the standard system interface by which all these components could be connected dynamically. The system invokes the functions in the component through the interface, thus, the inner implementation is invisible for the system framework. In other words, the component could be replaced by another component as long as they implement the same interface without affecting other parts of the system. For instance, as data access interface has been defined, all the data access components are the same for the system framework. Regardless of which file format would be supported by the system, developing a new component that implement the data access interface is the only work to do.

5 CONCLUSIONS

Surface metrology as a discipline is undergoing a rapid development. More and more analysis and evaluation methods are used to find out the surface features, and additional parameters are used to indicate these characteristics. An adaptive framework will promote the prompt and wide use of advance analysis technologies. Component based development technology as a successor to object-oriented is aimed to achieve coarse-grained (component) reuse. It does enable the development of extendible system although it is relatively more difficult. Further work will focus on the research of surface analysis technologies, the implementation of functional components and the realisation of system framework.

REFERENCES

Blateyron, F. (2006). "New 3D parameters and filtration techniques." Proceedings of the JSPE.

Bui, S. H., B. Muralikrishnan, et al. (2005). "A framework for Internet-based surface texture analysis and information system." Precision Engineering 29(3): 298-306.

Conroy, M. and J. Armstrong (2005). "A comparison of surface metrology techniques." Journal of Physics: Conference Series 13: 458-465.

Dolenc, M. (2004). "Developing extendible component-oriented finite element software." Advances in Engineering Software 35(10-11): 703-714.

Grundy, J. and J. Hosking (2002). "Developing adaptable user interfaces for component-based systems." Interacting with Computers 14(3): 175-194.

Jiang, X., P. J. Scott, et al. (2007). "Paradigm shifts in surface metrology. Part I. Historical philosophy." Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science 463(2085): 2049-2070.

Papajorgji, P., H. W. Beck, et al. (2004). "An architecture for developing service-oriented and component-based environmental models." Ecological Modelling 179(1): 61-76.

Whitehouse, D. J. (1997). "Surface metrology." Measurement Science and Technology 8(9): 955–972.

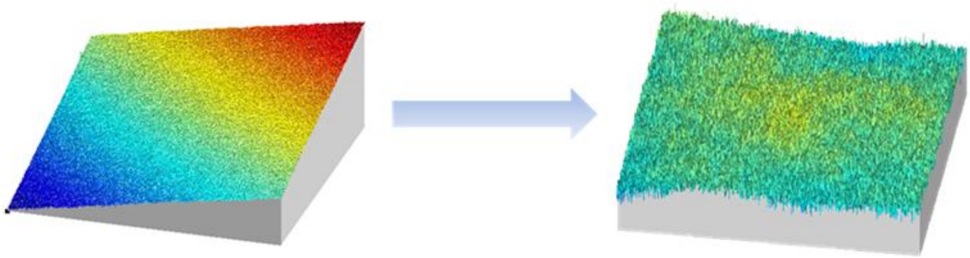

Figure 1: Fitting

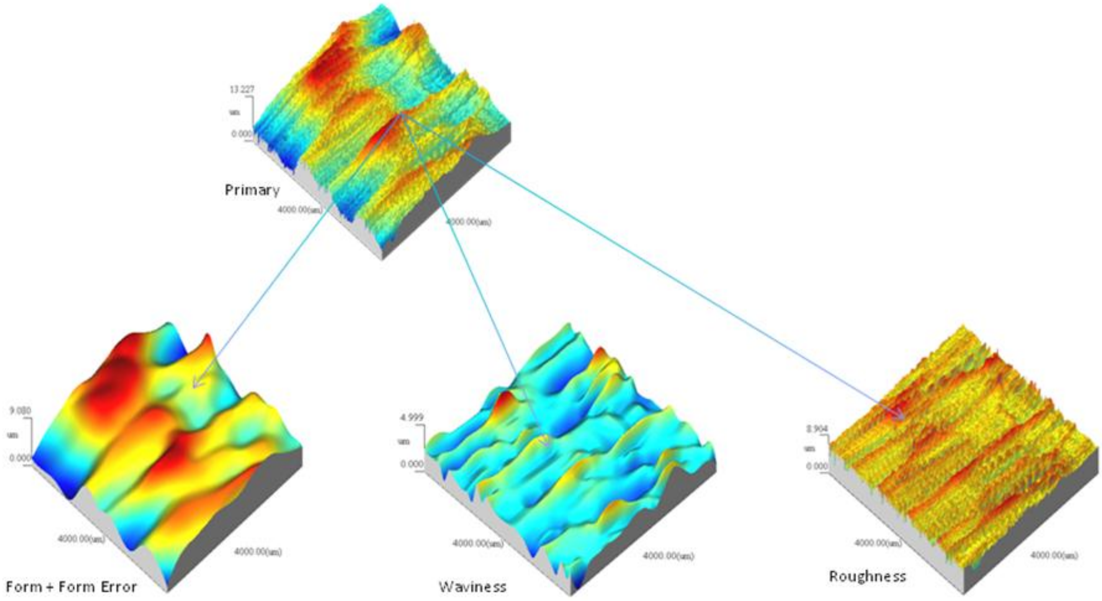

Figure 2: Filtering

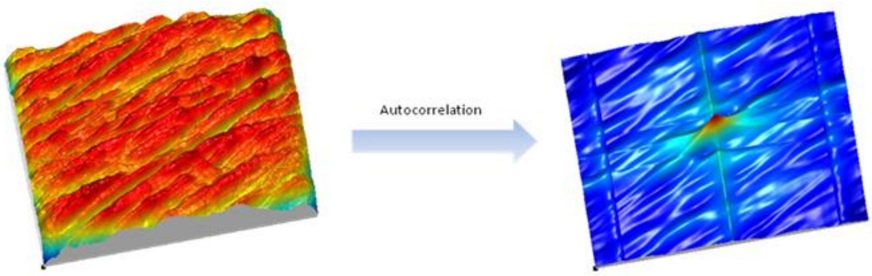

Figure 3: Autocorrelation

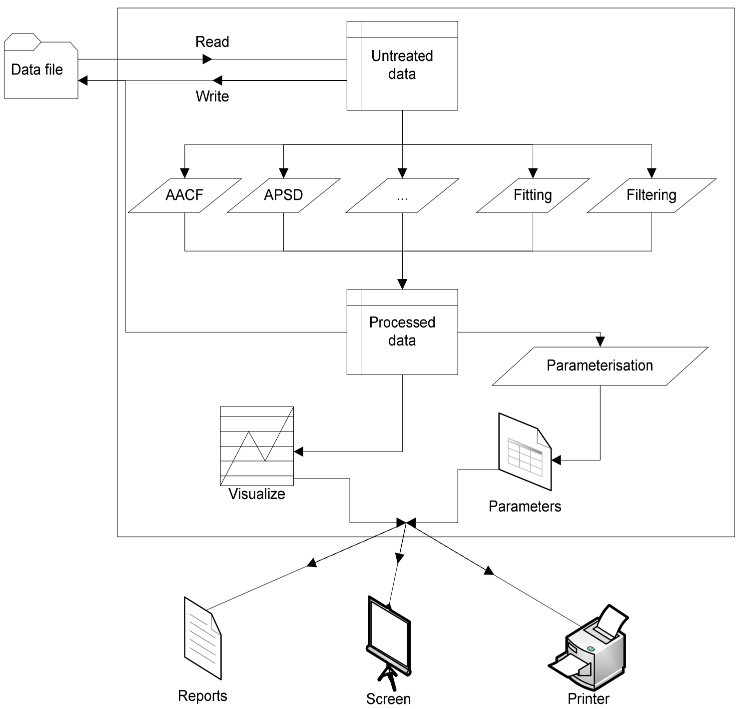

Figure 4: Operation flow of the system framework