

FORMING PROJECTION IMAGES FROM EACH LAYER OF RETAIN USING AN APPROACH OF IMAGE PAIR FUSION

T.Deepa¹, E.Nandakumar M.E.,(Ph.D)², M.Arulprakash M.Tech.,²

¹Department of Computer Science, Sri Subramanya College of Engineering and Technology, Palani

²Department of Computer Science, Sri Subramanya College of Engineering and Technology, Palani

¹deeparamanujam@gmail.com

Abstract : Recently, image pairs, such as noisy and blurred images or infrared and noisy images, have been considered as a solution to provide high-quality photographs under low lighting conditions. In this paper, a new method for decomposing the image pairs into two layers, i.e., the base layer and the detail layer, is proposed for image pair fusion. In the case of infrared and noisy images, simple naive fusion leads to unsatisfactory results due to the discrepancies in brightness and image structures between the image pair. To address this problem, a local contrast-preserving conversion method is first proposed to create a new base layer of the infrared image, which can have visual appearance similar to another base layer, such as the denoised noisy image. Then, a new way of designing three types of detail layers from the given noisy and infrared images is presented. To estimate the noise-free and unknown detail layer from the three designed detail layers, the optimization framework is modeled with residual-based sparsity and patch redundancy priors. To better suppress the noise, an iterative approach that updates the detail layer of the noisy image is

adopted via a feedback loop. This proposed layer-based method can also be applied to fuse another noisy and blurred image pair..

Keywords : Noisy, Infrared and restored images, Gradient Different Constraint, Artificial Blurred Image Generation, Multi Resolution.

1. Introduction

For noisy and blurred image pairs, models based on residual deconvolution (RD), multi-resolution (MR), gradient different constraint (GDC), and artificial blurred image generation (ABIG) have been used. The RD model identifies the unknown original image (x) by adding the lost detailed to the base layer, i.e., the denoised noisy image (x_{n,d}). This model is effective in restoring the fine details; however, it requires single-image-based deconvolution to identify the detail layer, and thus it is susceptible to ringing artifacts. In contrast, the GDC and ABIG models utilize the noisy image as the regularization and data fidelity terms, respectively, and then combine them in the transform or spatial

domain. Thus, the ringing artifacts that result from the deconvolution with the captured blurred image (x_b) and the inaccurate kernel (k) can be mitigated by fusing the sharp noisy image. The difference between GDC and ABIG is that the captured noisy image is used as the prior in the GDC model, whereas the captured image is used as the data fidelity term in the ABIG model. In other words, the GDC model makes the gradients of the unknown original image similar to those of the noisy image. However, the ABIG model produces two types of blurred images: one is the captured blurred image, and the other is the artificially created blurred image, which is obtained by denoising the noisy image (x_n, d).

The artificially created blurred image is updated with the corresponding estimated kernel (k_g) via a feedback loop. This is based on a related work, which showed that two blurred images yield a better restored image compared to the use of only one blurred image. In the GDC and ABIG models, the alpha value (α) indicating the prior can be fixed at 0.5 or 1.0, or it can be spatially varied. The MR method combines the wavelet coefficients of the noisy image, (x_n), with those of the blurred image, (x_b) based on the assumption that the absolute differences between two images are typically larger and closer to the edges of the images. This indicates that the value of λ is inversely proportional to the absolute value of (d). In Table I, λ can be a scalar value or a function depending on the model, and the value of λ can be constant or spatially varied. However, this assumption fails in the presence of severe noise in the captured noisy image. Thus, this MR model is more appropriate for high dynamic range (HDR) fusion,

infrared coloring, or dehazing, where the input images contain little noise.

1. System Analysis

1.1 Existing System

A novel contrast-based image fusion algorithm is used in the wavelet domain for noisy source images. Novel features of the fusion method are the noise reduction taking into consideration the linear dependency among the noisy source images and introducing an appropriate modification of the magnitude of the wavelet coefficients depending on the noise strength.

Region based multi-focus image fusion method using local spatial frequency first segments the average image of source images to get the region map and then calculates local spatial frequency for each pixel in source images from a local window. After this, regional spatial frequency is calculated for each region. Then a fused image is constructed from the selected regions according to the RSF calculated.

While it is challenging to enforce digital forensics analysis, Boneh and Waters introduced augmented broadcast encryption that is efficient for constructing traitor-tracing, and trace-and-revoke systems. The scheme is resistant to an arbitrary number of colluders and secure against adaptive adversaries. Attrapadung and Imai proposed a new cryptosystem, called Broadcast ABE (BABE), with direct revocation mechanism. Garg et al. presented a trace-and-revoke scheme based on prime order bilinear groups, and provided the first implementations of efficient fully collusion-resilient

traitor tracing scheme. Liu et al. also presented a blackbox traceable ABE that achieves the traceability in $O(\sqrt{n})$, where n is the number of users in the system. There have been some cryptosystems constructed on the partial order relation. Akl and Taylor put forward a simple scheme to solve multilevel security problem in 1982. Kim et al. proposed a new key management system for multilevel security using various one-way functions in 2005. Chung et al. proposed a hierarchy method based on the elliptic curve cryptosystem and one-way hash function to solve dynamic access problems in 2008. Another related field is hierarchical key management with time control. For example, Tzeng proposed a time-bound scheme based on Lucas function in 2002, but it is insecure against collusion attacks by Yi and Ye. Another similar schemes based on the tamper-resistant device and the hash function were proposed by Chien in 2004 and Bertino et al. in 2008, respectively. Santis et al. summarized and

provided several provably-secure hierarchical key assignment schemes based on an existing schemes in 2007. Peer et al. also propose an interesting method to represent data in multiple resolutions with each resolution secured with a different key in 1014. In all, these work cannot support common access control and digital forensics, but their techniques are worth learning for our construction.

Proposed System

The proposed image fusion algorithm directly fuses two source images of a same scene using weighted average. The proposed method differs from other weighted average methods in terms of weight computation and the domain of weighted average. Here, the weights are computed by measuring the strength of details in a detail image obtained by subtracting CBF output from original image. The weights thus computed are multiplied directly with the original source images followed by weight normalization.

3. System Design

3.1 Flow diagram

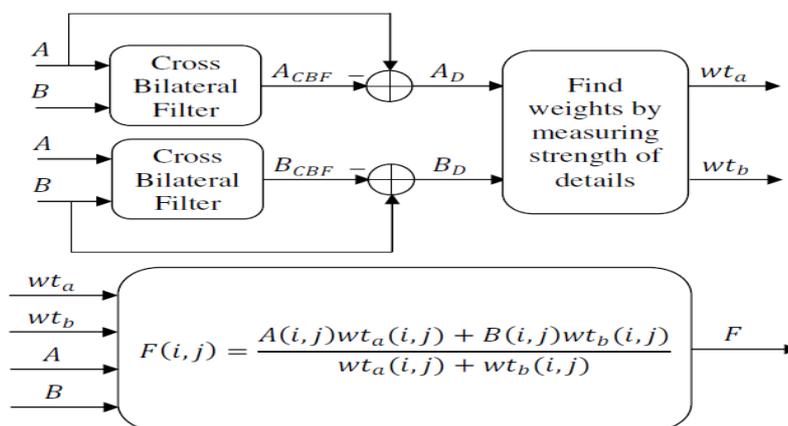
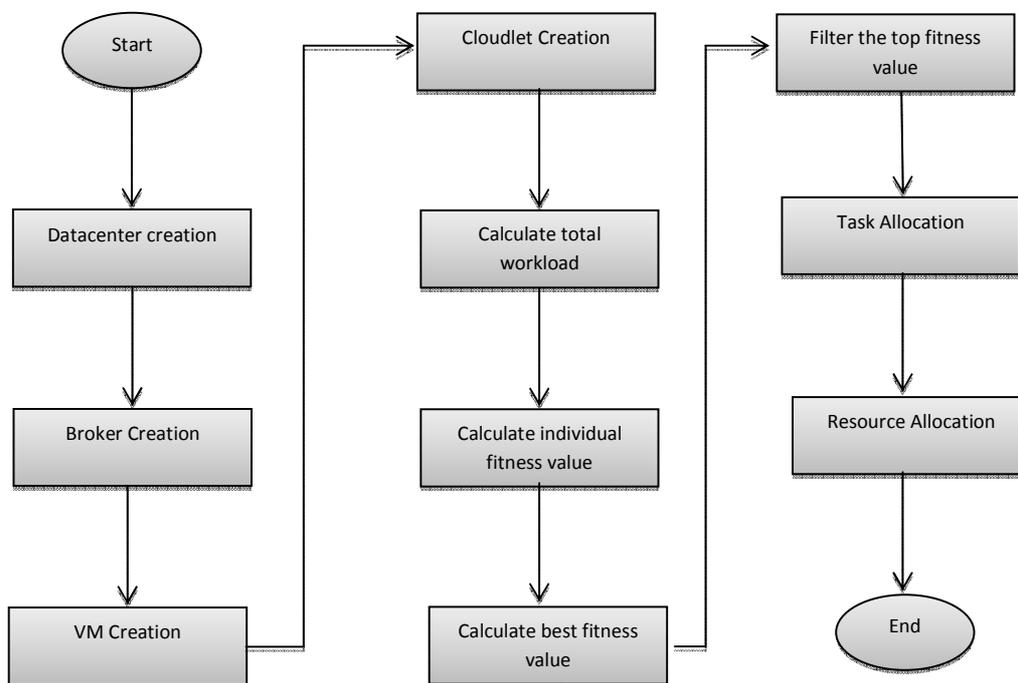


Fig 3.1 Flow Diagram

3.2. System Architecture



3.2 System Architecture

3.3 Use case diagram

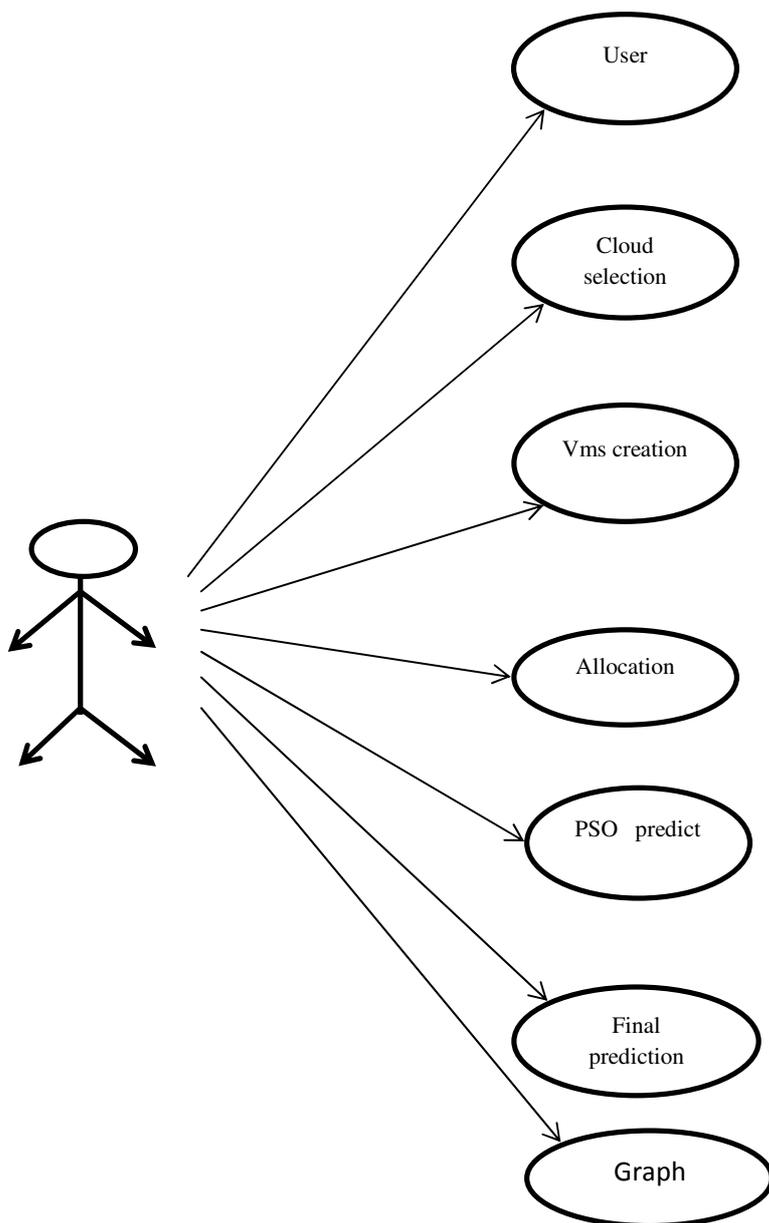


Fig 3.3 Use Case Diagram

3.4 Class Diagram

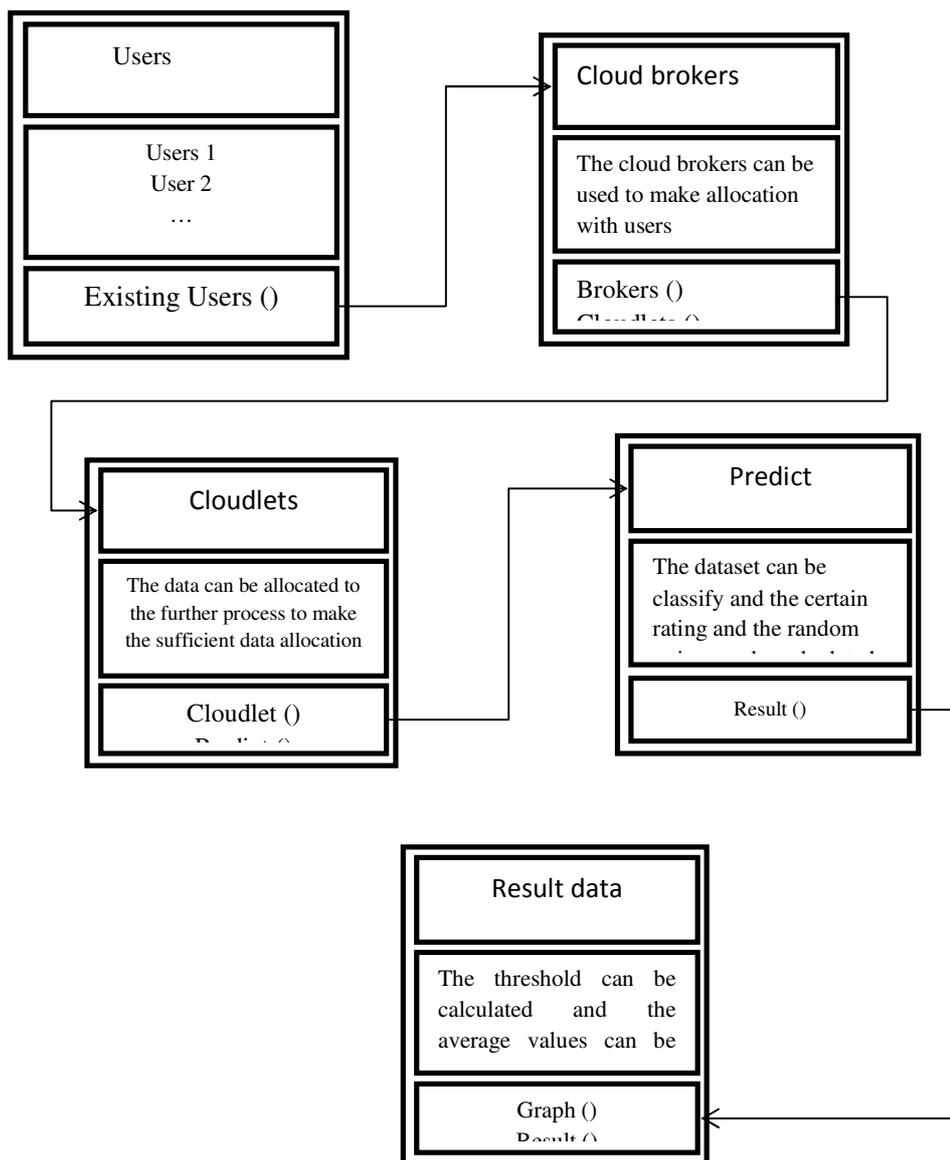


Fig 3.4 Class Diagram

3.6.ERDiagram

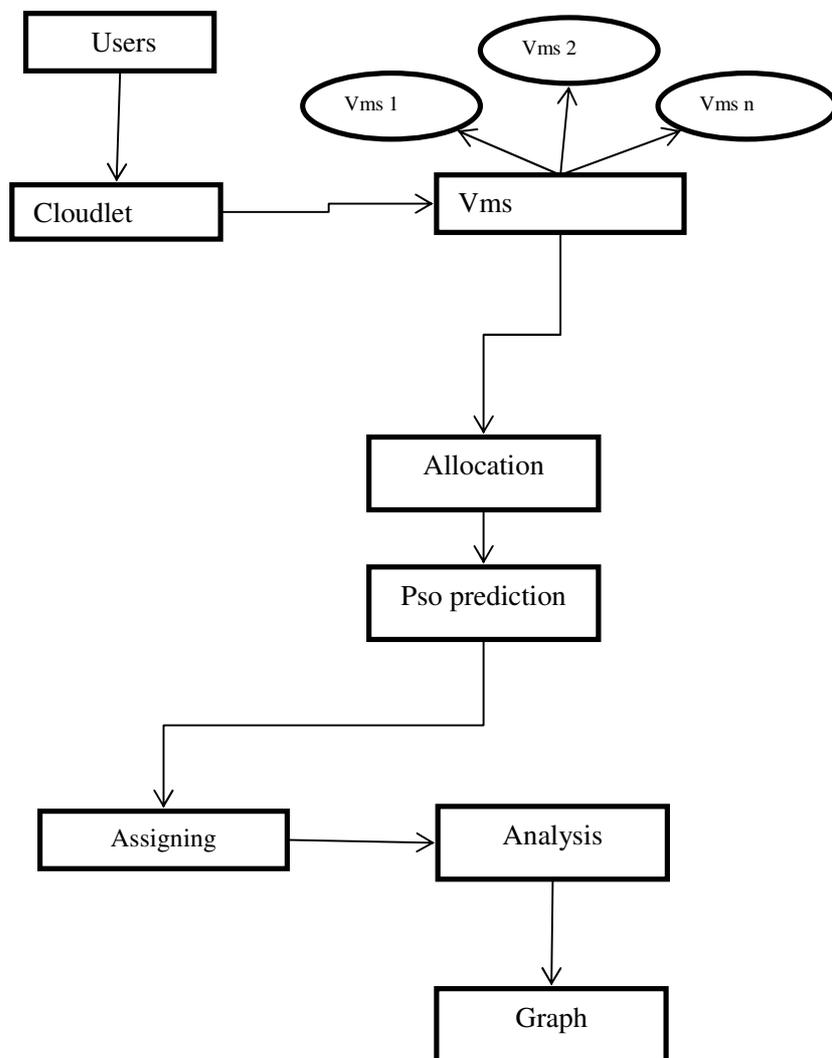


Fig 3.6 ER Diagram

4. System Testing

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. The candidate system is subject to variety of tests-on-line response, Volume Street, recovery and security and usability test. A series of tests are performed before the system is ready for the user acceptance testing. Any engineered product can be tested in one of the following ways. Knowing the specified function that a product has been designed to from, test can be conducted to demonstrate each function is fully operational. Knowing the internal working of a product, tests can be conducted to ensure that “al gears mesh”, that is the internal operation of the product performs according to the specification and all internal components have been adequately exercised.

4.1 Types of Testing

4.1.1 Unit Testing

Unit testing is the testing of each module and the integration of the overall system is done.

Unit testing becomes verification efforts on the smallest unit of software design in the module. This is also known as ‘module testing’. The modules of the system are tested separately. This testing is carried out during the programming itself. In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module. There are some validation checks for the fields. For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included. It is very easy to find error and debug the system.

4.1.2 Integration Testing

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function. Integrated testing is systematic testing that can be done with sample data. The need for the integrated test is to find the overall system performance. There are two types of integration testing. They are:

- i) Top-down integration testing.
- ii) Bottom-up integration testing.

4.1.3 White Box Testing

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases. Using the white box testing methods, we derived test cases that

guarantee that all independent paths within a module have been exercised at least once.

4.1.4 Black Box Testing

- Black box testing is done to find incorrect or missing function
- Interface error
- Errors in external database access
- Performance errors
- Initialization and termination errors

In 'functional testing', is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called 'black box testing'. It tests the external behavior of the system. Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

4.1.5 Validation Testing

After the culmination of black box testing, software is completed assembly as a package, interfacing errors have been uncovered and corrected and final series of software validation tests begin validation testing can be defined as many, but a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

4.1.6 User Acceptance Testing

User acceptance of the system is the key factor for the success of the system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required.

4.1.7. Output Testing

After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format. The output displayed or generated by the system under consideration. Here the output format is considered in two ways. One is screen and the other is printed format. The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs. For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system.

5. System Implementation

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended users and the operation of the system. The people are not sure that the software is meant to make their job easier.

- The active user must be aware of the benefits of using the system
- Their confidence in the software built up
- Proper guidance is impaired to the user so that he is comfortable in using the application

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not running on the server, the actual processes will not take place.

5.1 User Training

To achieve the objectives and benefits expected from the proposed system it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for education and training is more and more important.

Education is complementary to training. It brings life to formal training by explaining the background to the resources for them. Education involves creating the right atmosphere and motivating user staff. Education information can make training more interesting and more understandable.

5.2 Training on the Application Software

After providing the necessary basic training on the computer awareness, the users will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design, type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the data entered. This training may be different across different user groups and across different levels of hierarchy.

5.3 Operational Documentation

Once the implementation plan is decided, it is essential that the user of the system is made familiar and comfortable with the environment. A documentation providing the whole operations of the system is being developed. Useful tips and guidance is given inside the application itself to the user. The system is developed user friendly so that the user can work the system from the tips given in the application itself.

5.4 System Maintenance

The maintenance phase of the software cycle is the time in which software performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is to make adaptable to the changes in

the system environment. There may be social, technical and other environmental changes, which affect a system which is being implemented. Software product enhancements may involve providing new functional capabilities, improving user displays and mode of interaction, upgrading the performance characteristics of the system. So only thru proper system maintenance procedures, the system can be adapted to cope up with these changes. Software maintenance is of course, far more than “finding mistakes”.

5.4.1 Corrective Maintenance

The first maintenance activity occurs because it is unreasonable to assume that software testing will uncover all latent errors in a large software system. During the use of any large program, errors will occur and be reported to the developer. The process that includes the diagnosis and correction of one or more errors is called Corrective Maintenance.

5.4.2 Adaptive Maintenance

The second activity that contributes to a definition of maintenance occurs because of the rapid change that is encountered in every aspect of computing. Therefore Adaptive maintenance termed as an activity that modifies software to properly interfere with a changing environment is both necessary and commonplace.

5.4.3 Perceptive Maintenance

The third activity that may be applied to a definition of maintenance occurs when a software package is successful. As the software is used, recommendations for new capabilities, modifications to existing functions, and general enhancement are received from users. To satisfy requests in this category, Perceptive maintenance is performed. This activity accounts for the majority of all efforts expended on software maintenance.

5.4.4 Preventive Maintenance

The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability, or to provide a better basis for future enhancements. Often called preventive maintenance, this activity is characterized by reverse engineering and re-engineering techniques.

6 Modules

- Cloud Parameter Creation
- Job Scheduler
- Applying QPSO
- VM Scheduler

6.1 Cloud parameter creation

6.1.1 User Registration and policy

User, first register their details and login into the process. Depending upon the policy given,

user send request to the SLA Manager. User requirement is matching with the resources in the cloud service providers.

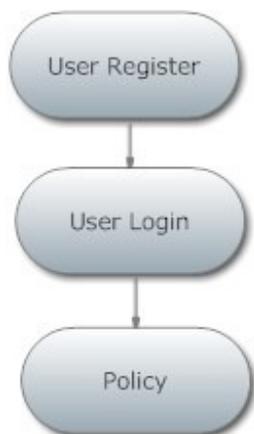


Fig 6.1 Parameter Creation

6.1.2 Vms scheduler

SLA Manager

SLA stands for Service Level Agreement. SLA Manager receive the request from the user. They handover the request to the broker for further process. The service user negotiates with the service broker on the service-level agreement (SLA) details. The service user negotiates with the service broker on the service-level agreement (SLA) details. They eventually prepare an SLA contract. According to this contract, the broker selects, and then presents highly trusted resources to users from the trusted resource pool.

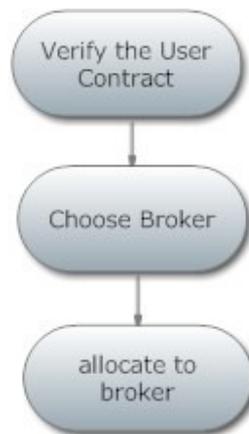


Fig 6.2 Virtual Machine

Allocation

6.1.3 Job scheduler

Resource Registration

Each resources in the cloud are register their details and resources details in the cloud for resource allocation. It manages and indexes all the resources available from multiple cloud providers, and obtains information from each particular cloud resource, acting as pricing interface for users, and updating the data-base when new information is available.

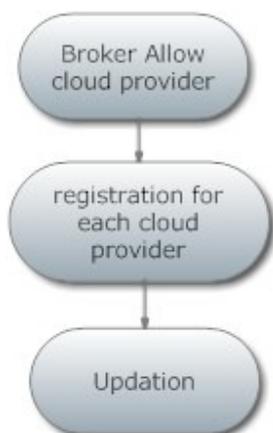


Fig 6.3 Job Scheduling

6.1.4 Trust Evaluation

Each cloud manager registers its service resources through the cloud broker. The service user negotiates with the service broker on the service-level agreement (SLA) details. They eventually prepare an SLA contract. According to this contract, the broker selects, and then presents highly trusted resources to users from the trusted resource pool.

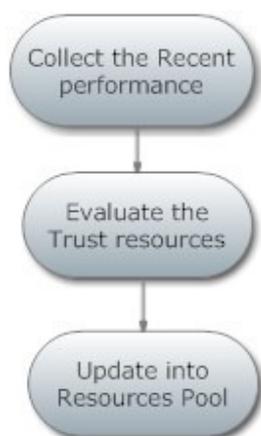


Fig 6.4 Trust Evaluation

6.1.5 Evaluate Trusted Resource

RTD is used to evaluate recent cloud resource service operators, and RTD is evaluated by knowledge of a resource's quality of service. When matchmaking a resource for users, the cloud broker must first consider whether the resource has the required capabilities. The first of these considerations can be evaluated by the resource's availability, which can determine whether a resource has the required capability or not. The second consideration mainly focuses on the reliability and security of the resource, which can be evaluated by the resource's service.

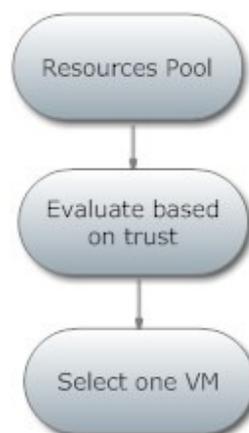


Fig 6.5 Evaluation

6.1.6 Applying qpso

Allocation of User Policy

Allocate the resources to the user depends upon the trust evaluation and trust resources in the cloud. Finally, Trusted resources is chosen and send to the broker. Then broker allocate the resources to the user through SLA Manager.

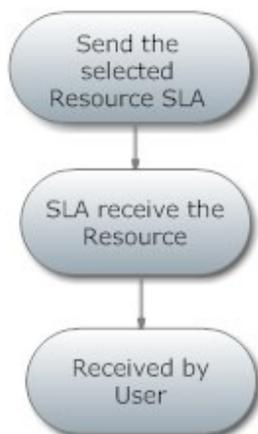


Fig 6.6 Allocation

7 SYSTEM REQUIREMENTS

7.1 Software Requirements

- O/S : Windows XP.
- Technology Used : MATLAB

7.2 Hardware Requirements

- System : Pentium IV 2.4 GHz
- Hard Disk : 160 GB
- Monitor : 15 VGA color
- Mouse : Logitech.

- Keyboard : 110 keys enhanced
- Ram : 4GB

7.3 MatLab

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems.

In 2004, MATLAB had around one million users across industry and academia. MATLAB users come from various backgrounds of engineering, science, and economics. MATLAB is widely used in academic and research institutions as well as industrial enterprises.

It is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are

expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including Graphical User Interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar no interactive language such as C or Fortran.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects, which together represent the state-of-the-art in software for matrix computation.

MATLAB features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

7.3.1 MatLab System

The MATLAB system consists of five main parts:

7.3.2 The MATLAB language

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

7.4 The MATLAB working environment

This is the set of tools and facilities that you work with as the MATLAB user or programmer. It includes facilities for managing the variables in your workspace and importing and exporting data. It also includes tools for developing, managing, debugging, and profiling M-files, MATLAB's applications.

7.5 Handle Graphics

This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It

also includes low-level commands that allow you to fully customize the appearance of graphics as well as to build complete Graphical User Interfaces on your MATLAB applications.

7.6 The MatLab mathematical function library

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

7.7 The MatLab Application Programming Interface(API)

This is a library that allows you to write C and Fortran programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

7.10 MySQL

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

Free-software-open source projects that require a full-featured database management system often use MySQL. Applications which use MySQL

databases include: TYPO3, Joomla, WordPress, phpBB, Drupal and other software built on the LAMP software stack.

7.11 Platforms and interfaces

Many programming languages with language-specific APIs include libraries for accessing MySQL databases. These include MySQL Connector/Net for integration with Microsoft's Visual Studio (languages such as C# and VB are most commonly used) and the JDBC driver for Java. In addition, an ODBC interface called MyODBC allows additional programming languages that support the ODBC interface to communicate with a MySQL database, such as ASP or ColdFusion. The MySQL server and official libraries are mostly implemented in ANSI C/ANSI C++.

7.12 FEASIBILITY STUDY

The feasibility study is carried out to test whether the proposed system is worth being implemented. The proposed system will be selected if it is best enough in meeting the performance requirements.

The feasibility carried out mainly in three sections namely.

- Economic Feasibility
- Technical Feasibility
- Behavioral Feasibility

Economic Feasibility

Economic analysis is the most frequently used method for evaluating effectiveness of the proposed system. More commonly known as cost benefit analysis. This procedure determines the benefits and saving that are expected from the system of the proposed system. The hardware in system department if sufficient for system development.

Technical Feasibility

This study center around the system's department hardware, software and to what extend it can support the proposed system department is having the required hardware and software there is no question of increasing the cost of implementing the proposed system. The criteria, the proposed system is technically feasible and the proposed system can be developed with the existing facility.

Behavioral Feasibility

People are inherently resistant to change and need sufficient amount of training, which would result in lot of expenditure for the organization. The proposed system can generate reports with day-to-day information immediately at the user's request, instead of getting a report, which doesn't contain much detail.

8 SYSTEM IMPLEMENTATION

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended users and the operation of the system.

The people are not sure that the software is meant to make their job easier.

- The active user must be aware of the benefits of using the system
- Their confidence in the software built up
- Proper guidance is impaired to the user so that he is comfortable in using the application

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not running on the server, the actual processes will not take place.

8.1 User Training

To achieve the objectives and benefits expected from the proposed system it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for education and training is more and more important. Education is complementary to training. It brings life to formal training by explaining the background to the resources for them. Education involves creating the right atmosphere and motivating user staff. Education information can make training more interesting and more understandable.

8.2 Training on the Application Software

After providing the necessary basic training on the computer awareness, the users will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design, type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the data entered. This training may be different across different user groups and across different levels of hierarchy.

8.3 Operational Documentation

Once the implementation plan is decided, it is essential that the user of the system is made familiar and comfortable with the environment. A documentation providing the whole operations of the system is being developed. Useful tips and guidance is given inside the application itself to the user. The system is developed user friendly so that the user can work the system from the tips given in the application itself.

9 Conclusion

A new layer-based image pair fusion method was proposed. To solve the discrepancy problem between noisy and infrared images, a local contrast-preserving conversion method was developed, thereby enabling the appearance of the base layer of the infrared image to be close to that of the noisy image. In addition, a new way of designing three

types of detail layers that contain different information, e.g., highlights, edges, or textures, was introduced. This detail layer design leads to improvement in the edge and texture representations. Also, the proposed residual-based sparse coding effectively suppresses the noise in the detail layer of the noisy image through a feedback loop. Furthermore, the experimental results showed that the proposed layer-based method can also be applied to noisy and blurred image pairs.

10 References

- [1] Azzedin .F and Ridha .A. "Feedback behavior and its role in trust Assessment for peer-to-peer systems,"Telecommunication. System., vol. 44, No. 3-4, pp. 253–266, 2010.
- [2] Cayman .S, Gales .A, Chapman .C, Tofutti .G, Rodeo-Merino .L,Vaquero .L.M, K. Nagging and B. Rochwerger, "Monitoring service Clouds in the future internet," in Future Internet Assembly. Amstardam, the Netherlands: IOS Press, 2010, pp. 115–126.
- [3] Chaves .S.A, Urinate R.B ,septal .C.W, "Toward an architecture for monitoring private clouds," IEEE Communication Mag.,vol.49, no. 2, pp. 130–137, 2011.
- [4] Dragon .N, "A survey on trust-based web service provision Approaches," in Proc. 3rd Int. Conf. Dependability, 2010, pp. 83–99.
- [5] Ghosh .D, Canon .R.S, and Ramakrishnan .L. "I/O performance Of virtualized cloud environments," in Proc. 2nd Int. Workshop

- Data Intensive Comput. Clouds, 2011, pp. 71–80.
- [6] Habib .S.M, Rise .S, and Muhlhauser .M, “Towards a trust man-agreement system for cloud computing,” in Proc. IEEE 10th Int.Conf. Trust, Security Privacy Comput. Commun. 2011, pp. 933–939.
- [7] Liang.Z and Shi .W, “A reputation-driven scheduler for auto-gnomic and sustainable resource sharing in grid computing,” Parallel Diatribe. Comput. vol. 70, no. 2, pp. 111–125, 2010.
- [8] Hoffman .L.J, Lawson-Jenkins .K, and Blum .J, “Trust beyond Security: An expanded trust model,” Communication. ACM, vol. 49, No. 7, pp. 95–101, 2006.
- [9] Hwang .K and Li .D, “Trusted cloud computing with secure Resources and data coloring,”IEEE Internet Comput. , vol. 14, no. 5, pp. 14–22, Sep. /Oct. 2010.
- [10] Jackson .K.R, Morikis .K, Ramakrishnan .L, K. J. Range, and R. C.Thomas, “Performance and cost analysis of the Supernova factory On the Amazon AWS cloud,” Sci. Program. , vol. 19, no. 2-3, pp. 107–119, 2011.
- [11] Khan .K.M and Mallei .Q, “Establishing trust in cloud Computing,” IT Prof., vol. 12, no. 5, pp. 20–27, Sep. /Oct. 2010.
- [12] Kim .H, Lee .H, Kim .W, and Kim .Y, “A trust evaluation model For QoS guarantee in cloud systems,” Int. J. Grid Diatribe. Comput. , vol. 3, no. 1, pp. 1–10, 2010.
- [13] Legrand .I, Voicu .R, Cirstoiu .C, Grigoras .C, Betev .L, Costan .A, “Monitoring and control of large systems with MonALISA,” Com-mun. ACM, vol. 52, no. 9, pp. 49–55, 2009.
- [14] Li .X and Yang .Y, “Trusted data acquisition mechanism for cloud Resource scheduling based on distributed agents,” China Commun. , vol. 8, no. 6, pp. 108–116, 2011.
- [15] Li .X, Zhou .F, and Yang .X, “Scalable feedback aggregating (SFA) Overlay for large-scale P2P trust management,” IEEE Trans. Parallel Diatribe. Syst., vol. 23, no. 10, pp. 1944–1957, Oct. 2012.
- [16] Manuel .P.D, S. Tamara Salve, and Barr .M .I .A .E, “Trust manegement system for grid and cloud resources,” inProc. 1st Int.Conf. Adv. Comput., Dec. 2009, pp.176–181.
- [17] Shafer .J, “I/O virtualization bottlenecks in cloud computing Today,” in Proc. 2nd Conf. I/O Virtualization, 2010, pp. 1–7.
- [18] Woolskin .R, Obertelli G, Nurmi .D, Osman .S, Yousef .L, and Zagorodnov .D, “The eucalyptus open-source cloud-computing



system,” in Proc. 9th IEEE/ACM Int. Symp.

Cluster Com-put. Grid, 2009, pp.124–131.