



Skin Image Processing and Skin Characterizations

Wei Pan

**School of Engineering
London South Bank University**

Ph.D.

July 2017

To my Parents

Content

Content.....	1
Acknowledgement.....	1
Abstract.....	1
Chapter 1 Introduction	1
1.1 Skin Histology.....	2
1.1.1 The Epidermis.....	2
1.1.2 The Dermis.....	5
1.1.3 The Subcutaneous Tissue	6
1.1.4 Skin Functions	7
1.2 Skin Hydration	7
1.2.1 Skin Hydration.....	8
1.2.2 Water concentration profile and TEWL	9
1.2.3 The water in stratum corneum.....	9
1.3 Skin Measurement Techniques.....	9
1.3.1 Electrical Conductance	9
1.3.2 Electrical Capacitance.....	10
1.3.3 TEWL Measurement.....	12
1.3.4 Infrared Spectroscopy	16
1.3.5 Photoacoustic spectroscopy	16
1.3.6 Optothermal Infrared Spectroscopy	17
1.3.7 Magnetic Resonance Imaging.....	17
1.3.8 X-ray Diffraction.....	18
1.3.9 Ultrasound.....	18
1.3.10 Structured-light 3D Scanner.....	19
1.3.11 Optical Coherence Tomography.....	20
1.3.12 Diffusion Tensor Imaging	20
1.3.13 Confocal Raman Spectroscopy	21
1.4 Summary and Research Objectives.....	22
1.5 Organization of the Thesis	23
Chapter 2 Skin Measurement Instruments	24
2.1 OTTER.....	24
2.2 AquaFlux.....	29
2.3 Capacitive Contact Imaging Based on Fingerprint sensor	34
2.4 Corneometer.....	39
2.5 Digital Camera	41
2.6 ProScope HR2 Digital Microscope.....	42
2.7 Summary	43
Chapter 3 Skin Image Processing using MATLAB.....	45
3.1 Image processing Techniques.....	46
3.2 GUI Programme Implementation.....	66
3.3 Summary	83
Chapter 4 Skin Image Processing Using OpenCV.....	84
4.1 OpenCV	84
4.2 Microsoft Visual Studio	85
4.3 Image Processing Techniques.....	86
4.3.1 Face Detection	87
4.3.2 Image Stitching	91
4.3.3 Contours Detection	95
4.3.4 Real-time Video Processing.....	96
4.4 Summary	104

Chapter 5 Skin Characterization Measurements	106
5.1 Skin Over-Hydration Measurement	106
5.2 AquaFlux Occlusion Effect in Skin Hydration Measurement	110
5.3 Skin damage assessment	113
5.3.1 Tape stripping measurement	113
5.3.2 SLS Irritation Measurement 1	122
5.3.3 SLS Irritation Measurement 2	130
5.3.4 The Occlusion Effects in Capacitive Contact Imaging for In-vivo Skin Damage Assessments	133
5.4 Dermabrasion	145
5.5 Soap Drying Effect.....	150
5.6 Summary	158
Chapter 6 Conclusions and Future Work.....	160
6.1 Conclusions.....	160
6.2 Future Works	161
References	163
Publications	179
Appendix	181
QT GUI programme 1	181
QT GUI programme 2.....	186
Matlab GUI Programme.....	188

Acknowledgement

First, I wish to express my sincere thanks to Dr. Perry Xiao, my supervisor, for his guidance, support and patience during my PhD work. With his help, I obtained a lot of co-operation projects with the other researchers.

I would like to thank to my colleges in the group of South Bank University and Biox Systems Ltd for their help and technical support.

I would like also thank to my friends for their assistance.

Abstract

The skin hydration and skin Trans-epidermal water loss (TEWL) are of great importance in many skin research areas, such as dermatology, clinical analysis, pharmacology and cosmetic science etc. However, to measure them is not easy. Over the years, our research group has developed three novel technologies for such measurements: Opto-Thermal Transient Emission Radiometry (OTTER), AquaFlux and capacitive contact imaging based on the Fingerprint sensor. The aim of this research is to develop new skin image processing and data analysis techniques for capacitive contact images, as well as digital colour images, and to develop new methodologies for skin characterization by using the three technologies.

For skin image processing, a new GUI based MATLAB programme has been developed, which can be used for extracting and analysing the images from the result files created by the measurement instruments. The programme implements the skin image processing techniques such as image enhancement (i.e. brightness equalization, extraction of skin texture, hair removal), image stitching, image matching and skin surface 3D profiling etc. Another image processing programme based on OpenCV has also been developed, which is more suitable for real time video processing, including contour detection, texture extraction and face detection etc.

For the skin characterization, several experiments are conducted: skin over-hydration experiments; skin damage assessment including intensive washing, SLS irritations, and tape stripping; dermabrasion experiments; soap drying effect assessment. These experiments provide better understandings of the technologies. The occlusion effects in capacitive images shows good potential for skin damage assessment, as it can not only reflect the scale of damage, but also the types of damage.

Chapter 1 Introduction

The human skin is the largest organ of the body. It accounts for more than 12% total body weight and 18- 40% total water content in the body.

The water, found mainly in the dermis, is mobilised rapidly in stress situations for use in other organs in the body. However, the ability of the skin to hold water is primarily due to the stratum corneum (SC) and is critical for maintaining healthy skin (Blank, I.H 1952). The water content of the SC and Trans-epidermal water loss (TEWL) are of importance for its infection, dehydration, barrier function and mechanical properties.

There are many attempts which have been made for skin in-vivo measurement, such as electrical capacitance and electrical conductance measurements, evaporimetry, infrared spectrometry, ultrasound, magnetic resonance imaging and wide-angle x-ray diffraction.

In this research, three novel instruments are mainly used for skin measurements: Opto-Thermal Transient Emission Radiometry (OTTER), AquaFlux and capacitive contact imaging based on the Fingerprint sensor. There is a brief introduction of the instruments in Chapter 2.

The aim of this research is to develop new digital image processing and data analysis techniques for capacitive contact images from fingerprint sensor; to develop novel graphic user interface (GUI) based MATLAB software for digital image processing techniques; to develop new methodologies of skin measurement using our novel capacitive contact imaging, AquaFlux and OTTER, as well as other instruments.

This thesis can be divided into two parts, theoretical investigation (Chapter 3 and Chapter 4) and experimental work (Chapter 5).

In the theoretical part, a skin image processing algorithm has been developed for digital images, greyscale capacitive images and webcam. Relevant GUI has been developed to implement all the algorithm and experimental data analysis.

In the experimental work, various in-vivo measurements were performed in order to study skin hydration function using the three novel instruments.

In this chapter, a general introduction and review of skin structure and skin hydration will be presented.

1.1 Skin Histology

Skin is a barrier to provide protection against environmental hazards, such as radiological and microbial attacks, physical and chemical. It also works as a sensor to transmit environmental information such as pain, cool to the nerve centre and depot to store excess food as fat for future use (Swarbrick et al, 1995). For an average 70 kg human being, the surface area is about $1.8m^2$. A typical square centimetre covers 10 hair follicles, 12 nerves, 15 sebaceous glands, 100 sweat glands, 3 blood vessels with 0.92 m total length, 3.6 m of nerves and 3×10^6 cells.

Human skin may be subdivided into three mutually dependent layers: the fatty subcutaneous layer (hypodermis), the overlying dermis, and the epidermis, the outermost layer of the skin. Human skin displays two main types. Hairy skin encloses hair follicles and sebaceous glands, but there no encapsulated sense organs.

1.1.1 The Epidermis

The most superficial layer of human skin is the epidermis. The cells of the epidermis are produced from the basal layer and migrate to the exterior, undergoing keratinization to form the outermost layer, the stratum corneum. It

varies in thickness from 0.8 mm on the palm to 0.06 mm on the eyelids. The epidermis can be subdivided into five layers:

(1) The stratum basale (stratum germinativum).

The stratum basale is characterized by intense mitotic activity. The cell division occurs in this layer. New keratinocytes which move upwards pushing the older above towards the surface of the skin is generated by the basal keratinocytes.

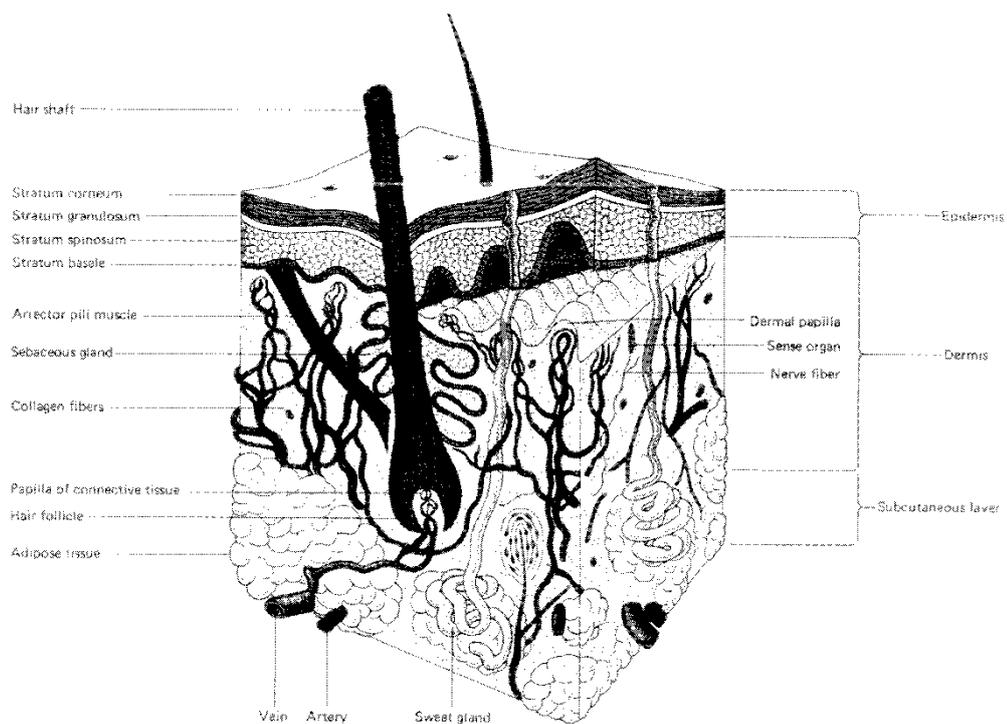


Figure 1.1 Structure of Human Skin, from Van De Graaff KM, Fox SI.

(2) The stratum spinosum.

The cells begin to flatten and their nuclei shrink in this layer. The

epidermis of areas subject to continuous friction and pressure (such as the sole of the feet) has a thicker stratum spinosum with more abundant tonofibrils and desmosomes.

(3) The stratum granulosum (granular layer).

It is characterized by 3-5 layers of flattened polygonal cells containing centrally located nuclei and cytoplasm. The shape is membrane-coating granule, ovoid or rodlike. The cells generate keratohyalin granules in this layer.

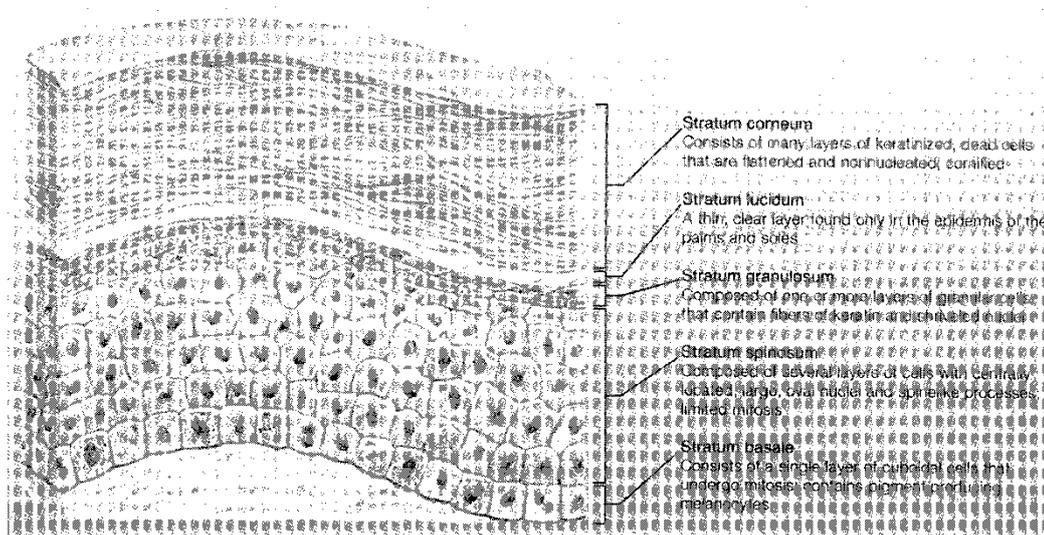


Figure 1.2 Structure of Epidermis, from Van De Graaff KM, Fox SI

(4) The stratum lucidum.

It is a thin translucent layer and is generated in the palm of the hand and the sole of the foot, an anatomically distinct, poorly staining hyaline zone.

(5) The stratum corneum (The horny layer).

It is the most superficial layer of the skin and comprises 10 to 15 layers of dead cells with approximately 10-20 μm thickness. The structure of the stratum corneum may be represented as a brick and mortar model

which provides two micropathways for the trans-epidermal drug diffusion (Figure 1.3, William AC and Barry BW, 1992). When the cells arrive at the SC, they are fully keratinized and dead.

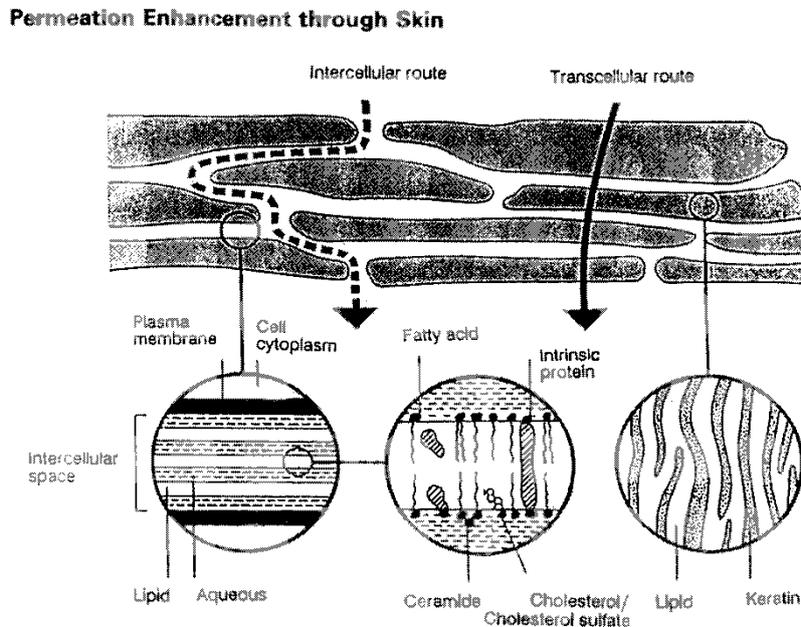


Figure 1.3 The “brick and mortar” structure of the stratum corneum (Diagram from William AC and Barry BW, 1992).

The epidermis contains three less abundant cell types: Melanocytes, the Langerhans Cells, and the Merkel Cells.

1.1.2 The Dermis

The dermis shown in Figure 1.4 is the second layer of the skin. The connective tissue that supports the epidermis and binds it to the subjacent layer forms the dermis. The thickness of the dermis varies depending upon the region of the body, reaching its maximum of 3 mm on the soles of the feet (Junqueira LC, 1977).

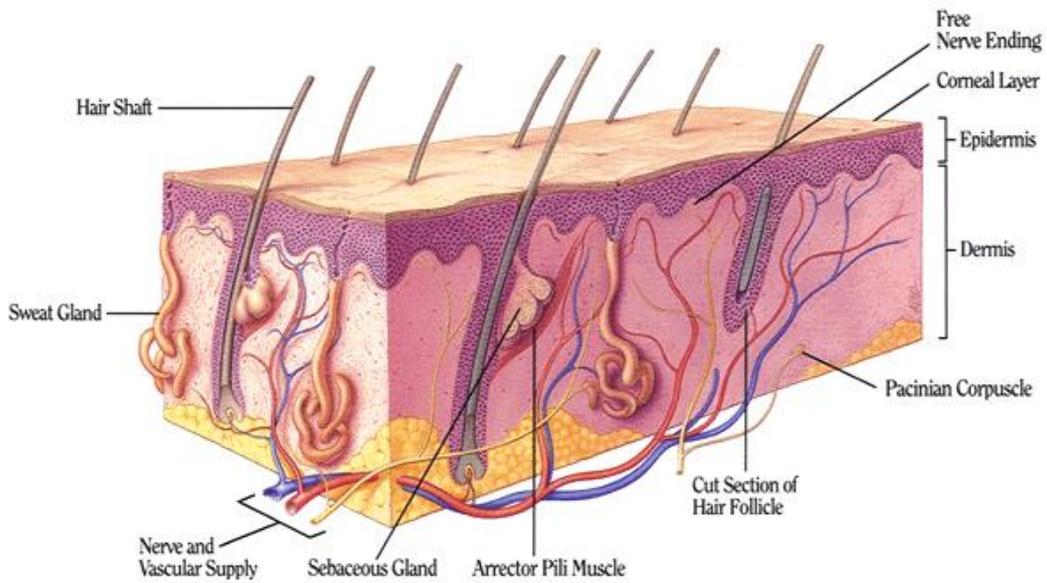


Figure 1.4 Structure of dermis

The dermis is composed of two layers. One is the immediately sub-epidermal papillary layer, and another is the deeper, more massive and denser the reticular layer. The dermis has a rich network of blood vessels which regulates temperature and pressure, delivers nutrients to the skin and removes waste products, mobilizes defence forces, and contributes to skin colour. The dermis also contains some epidermal derivatives, the hair follicles, sweat and sebaceous glands. A rich supply of nerves is found in the dermis, and the effect or nerves to the skin are postganglionic fibres of the ganglia of the paravertebral chain.

1.1.3 The Subcutaneous Tissue

The subcutaneous fat (hypodermis, subcutis) spreads all over the body. The size of this layer varies throughout the body and from person to person. The subcutis provides a thermal barrier and a mechanical cushion; it is a site of synthesis and depot of readily available high-energy chemicals (Barry BW 1983). It carries the major blood vessels and nerves to the skin and may contain sensory pressure organs (William AC, 1992).

1.1.4 Skin Functions

The skin performs many varied functions. The following presents a brief digest of its biological role (Barry BW 1983).

1. To contain body fluids and tissues --the mechanical function.
2. To protect from potentially harmful external stimuli--the protective or barrier function: (a) micro-organisms; (b) chemicals; (c) radiation; (d) heat; (e) electrical barrier; or (f) mechanical shock.
3. To receive external stimuli, i.e., to mediate sensation: (a) tactile (pressure) ; (b) pain; or (c) heat.
4. To regulate body temperature.
5. To synthesize and to metabolize compounds.
6. To dispose of chemical wastes.
7. To provide identification by skin variations.
8. To attract the opposite sex.
9. To regulate blood pressure.

1.2 Skin Hydration

A water profile in skin calculated from Energy Dispersive Spectrometry Spectra (Warner et al, 1988) is shown in Figure 1.5. The water in skin and its relationship to the skin function will be introduced in the following section.

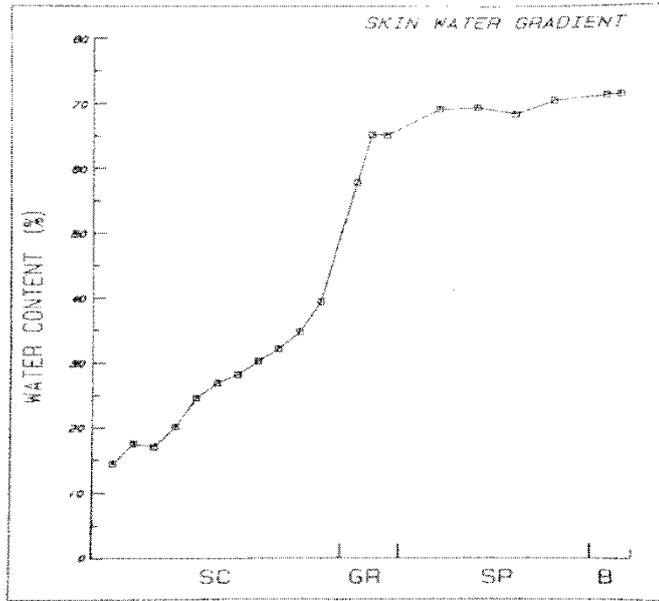


Figure 1.5 Calculated Water profile across human skin. The vertical scale is percent water expressed as grams of water per total grams (water plus dry mass) of tissue. SC stands for Stratum Corneum, GR stands for Stratum Granulosum, SP stands for Stratum Spinosum, and B stands for Stratum Basale.

1.2.1 Skin Hydration

The relationship between relative humidity and water contents of the stratum corneum is shown in Figure 1.6 (Spencer TS et al, 1975). According to Takenouchi M., at 0% Relative Humidity (RH), stratum corneum still holds about 0.05 g/g water.

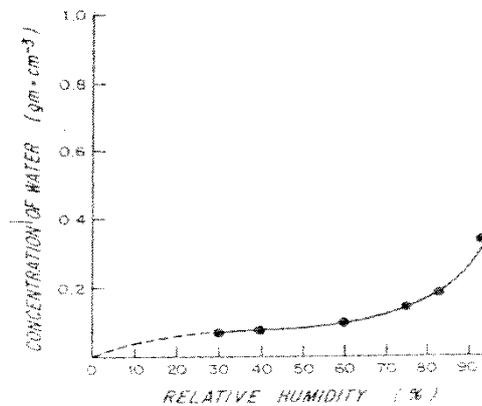


Figure 1.6 Concentration of water in stratum corneum in equilibrium with air at 30 °C as a function of relative humidity.

1.2.2 Water concentration profile and TEWL

Stratum corneum is dry at its surface because of the dry external environment and wet at its base where is contact with the deeper fully hydrated part of the epidermis (Blank IH, 1984). Therefore, a concentration gradient which results in a continuing diffusion of water from within the body through the skin to the environment, trans-epidermal water loss (TEWL) will generate within the stratum corneum. The TEWL for normal skin is about 4~8 g/m^2 per h (Tsai J et al, 1990).

1.2.3 The water in stratum corneum

In the stratum corneum, there are three types of water: tightly bound water, loosely bound water and free water. The experiments from (Anderson RL et al 1973) show that the hydration and dehydration rates of free water are lower than that of bound water. This suggests that free water is located primarily intracellular, with the cell envelope the major barrier to its loss, while the bound water is primarily intercellular or on the cell surface.

1.3 Skin Measurement Techniques

Numerous electronic, optical, magnetic and acoustic methods have been used in skin moisture content and skin state measurement (Potts RO, 1986, Berardesca E, 1995).

This thesis mainly focuses on the Fingerprint sensor, OTTER and TEWL measurements. This section presents the skin measurement techniques in order to figure out the strength of each technique.

1.3.1 Electrical Conductance

The physical principle of the conductance method is based on the changes in the electrical properties of the stratum corneum associated with skin hydration (Berardesca E, 1995).

For a conducting or semiconducting element, electrical resistance R and electrical conductance G are defined as

$$G = \frac{1}{R} = \frac{I}{V} \quad (1.1)$$

Where I is the electric current through the object and V is the voltage (electrical potential difference) across the object.

The electrical conductance of organic materials such as protein in the stratum corneum is greatly influenced by the water content. Human skin impedance falls when the frequency of applied alternating current raises and become easily measurable with dry electrodes. It is possible to assess the state of hydration of skin at 3.5 MHz (Tagami H. et al, 1994).

1.3.2 Electrical Capacitance

Water also affects the dielectric constant, which increases with increasing hydration of the stratum corneum. The capacitive imaging instrument (Fingerprint sensor), developed by our previous PhD student (Singh, H. 2010), see Figure 1.1, are based on the capacitance fringing field measurement principle. In this case, the parallel plates of the capacitor are split to a coplanar position; the fringing electric field will continue to create a field between the two plates. More details are described in Chapter 2.



Figure 1.1 The prototype of the Fingerprint sensor.

The Corneometer is one of the most commonly used capacitance measurement instruments. The measurement is based on capacitance measurement of a dielectric medium. The Corneometer® CM 825 measures the change in the dielectric constant due to skin surface hydration changing the capacitance of a precision capacitor. The measurement can detect even slight changes in the hydration level. More detail about the Corneometer® CM 825 is presented in Chapter 2.

Advantages of the Corneometer®

- Substances on the skin (e.g. salts or residues of topical applied products) have only minimal influence due to capacitance measurement.
- It allows very quick measurement (1 s) important to avoid occlusion.
- Continuous measurements over a longer time period can optionally be performed.
- The measurement depth is very small (10-20 μm of the Stratum corneum) to exclude the influence of deeper skin layers (e.g. from the blood vessels). Measurements on an only 15 μm thick foil show only 20 % of the original measurement value.
- The probe is small and lightweight for easy handling and measurement on all body sites (e.g. lips etc.).

- The spring in the probe head ensures constant pressure on the skin enabling exact, reproducible measurements.
- The accuracy of the Corneometer® probe and the penetration depth of the scatterfield can be checked easily anytime.
- The probe head can quickly be cleaned after each measurement.
- Worldwide established as "corneometry" with a broad range of studies. The Corneometer® has even been used on a space mission on the ISS.
- The Corneometer® CM 825 and its predecessor Corneometer® CM 820 are the only skin moisture measurement instruments of which the accuracy have ever been evaluated in a broad multicentric study.

1.3.3 TEWL Measurement

Measurement of transepidermal water loss (TEWL) is important for the infection, dehydration, barrier function and mechanical properties. The instrument (AquaFlux) details are described in Chapter 2.

TEWL is assessed by determining the vapour pressure gradient in the water boundary layer surrounding the skin. This gradient, which is stable in the absence of forced convection, is proportional to the amount of water vapour passing through the boundary layer per unit time and area by evaporation from the skin surface.

AquaFlux is a condenser-chamber based instrument. When the chamber is brought in contact with the skin surface, the chamber is sealed and it can protect the diffusion zone within it from ambient air movement. The natural convection and other bulk air movements are brought down because of the internal dimensions of the chamber being very small. In fluid dynamics terms, this requires the Rayleigh Number to be below the critical value for its geometry. Under these conditions, passive diffusion remains the only transport mechanism for the water vapour entering the chamber. The condenser controls the absolute humidity in the measurement chamber independently of ambient conditions. It is similar to a vapour sink by forming ice on its surface, thus creating a zone of low humidity in its immediate vicinity. By contrast, the test surface acts as a vapour source,

creating a zone of higher humidity in its immediate vicinity. This humidity difference causes water vapour to migrate from source to sink by passive diffusion and creates a diffusion vapour density gradient from which the flux density can be measured (Imhof, R.E. et al, 1999).

The Tewameter® TM 300 is the worldwide most accepted measuring device for the assessment of the TEWL. This is an indispensable parameter for the evaluation of the water barrier function of the skin and a basic measurement in all kind of applications. Even the slightest damage in the skin water barrier can be determined at an early stage. Several hundreds of international scientific studies (Literature List Tewameter®, www.courage-khazaka.de) demonstrate its importance in dermatological research and cosmetological fields.

A certain evaporation of water from the skin takes always place as part of the normal skin metabolism. As soon as the barrier function of the skin is slightly damaged, the water loss will increase (even with smallest damage invisible to the human eye). Therefore this measurement is a basis for all cosmetic and dermatological research. The Tewameter® probe measures the density gradient of the water evaporation from the skin indirectly by the two pairs of sensors (temperature and relative humidity) inside the hollow cylinder. This is an open chamber measurement. The open chamber measurement method is the only method to assess the TEWL continuously without influencing its micro environment. A microprocessor analyses the values and expresses the evaporation rate in g/h/m².

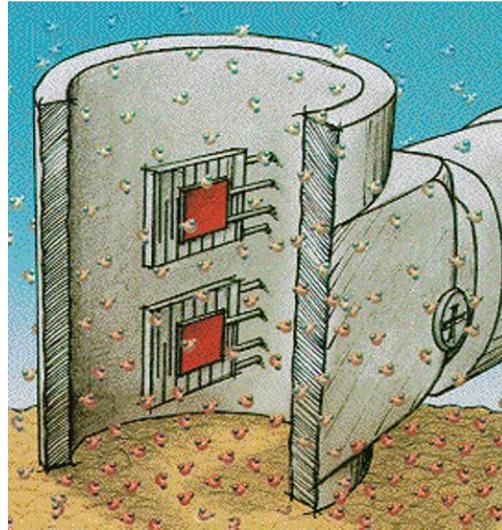


Figure 1.2 Cutout diagram of the probe head chamber of the Tewameter (<http://www.courage-khazaka.de/index.php/en/products/scientific/139-tewameter#tm1a>).

$$\frac{dm}{dt} = -D \cdot A \cdot \frac{dp}{dx} \quad (1.2)$$

Where A = surface [m²], m = water transported [g], t = time [h], D = diffusion constant [= 0.0877 g/m(h(mmHg))], p = vapour pressure of the atmosphere [mm Hg], x = distance from skin surface to point of measurement [m].

Advantages of the Tewameter®

- Extremely accurate and quick measurement of the TEWL without any influences of the micro climate of the skin (pressure, occlusion, temperature)
- The small size of the probe head minimizes the influence of air turbulence inside the probe.
- Its low weight has no influence on the skin structure and ensures easy handling.
- No waiting time between the measurements
- Traceable, sophisticated calibration of humidity, temperature and TEWL to g/h/m²
- A stable measurement is achieved quickly; continuous measurements over longer periods are also possible depending on the application.

- The development of the measurement values, average and standard deviation over time is always visible. Display of r.h and °C separately for upper and lower sensor
- Worldwide most used TEWL measurement method (even approved in space). Several hundreds of performed studies with the Tewameter® prove this fact.
- Recording of the ambient conditions with the Ambient Condition Sensor RHT 100 to help to compare measurements taken at different times
- Special calculations i.e. SSWL (Skin Surface Water Loss - assessment of skin's water holding capacity after occlusion) are possible
- With the open chamber measurement even higher water loss values can be detected accurately as no water is collected inside the probe.
- Accuracy of the probe can be checked anytime quickly
- Offset function to compensate "aging effects" of the probe
- Measurement with several probes at the same time possible (e.g. for the measurement of large areas or sweat studies)
- Accessories for measurements on small skin areas, special adapters to be hygienically cleaned etc. are available

The VapoMeter is the only fully portable instrument available for the measurement of TEWL values and evaporation rates.

The VapoMeter is battery-operated and portable. This offers great freedom of movement and the possibility of measuring difficult areas. Changeable and reusable adapters are available for different *in-vivo* and *in-vitro* user applications.

The core of the VapoMeter is a sensitive humidity sensor that is inside a cylindrical measurement chamber. This chamber is closed by the skin or other measurement surface during the measurement period and is unaffected by ambient airflows. The sensor monitors the increase of relative humidity (RH) inside the chamber during the measurement. The evaporation rate value (g/m²h) is automatically calculated

from the RH increase. The chamber is passively ventilated between measurements and the ventilation time is automatically controlled.

The VapoMeter is widely accepted as the most practical and efficient instrument for TEWL measurements.

1.3.4 Infrared Spectroscopy

The Infrared spectroscopy measurement (Osberghaus et al, 1978) is based upon human skin exhibiting two characteristic absorption bands at $1,645\text{ cm}^{-1}$ and at $1,545\text{ cm}^{-1}$. The first of these bands, amide I band, is greatly expanded by the presence of water, while the second band, the amide II band, is unaffected. The ratio between the two bands is therefore largely dependent upon the amount of water present in the horny layer (Gloor M et al, 1980, 1981).

Attenuated Total-reflectance (ATR) infrared spectroscopy was also used to measure the water content (Potts RO, 1985). A weak water-absorbance band near $2,100\text{ cm}^{-1}$ was detected in both in-vivo and in-vitro spectra. The significance of this band is that it occurs in a region of the mid-infrared where the SC and most topically applied substances show no absorption. In-vitro spectra obtained as a function of ambient relative humidity showed an increase in absorbance near $2,100\text{ cm}^{-1}$ with increasing water concentration in the SC. The combined in-vivo and in-vitro results lead to a quantitative assessment of water concentration in the uppermost layers of the SC. In addition, the ATR-IR technique holds promise for the measurement of water concentration at varying depths, since the penetration changes with the incident angle. Thus, optical sectioning of the SC may be achieved with variable-angle optics. The internal reflection element (IRE) usually used is a zinc sulfide trapezoid.

1.3.5 Photoacoustic spectroscopy

Photoacoustic spectroscopy has also been examined for measuring the concentration profile of water in human SC in-vivo (I. Simon et al, 1979). Photoacoustic spectroscopy measures the acoustic signals arising when modulated optical radiation is absorbed near a sample's surface. Absorption of optical

radiation by water within the stratum corneum results in localized heating in proportion to the amount of water present at any given depth. Because the incident radiation is modulated, the penetration depth of a periodic heat wave into the stratum corneum depends on its frequency.

1.3.6 Optothermal Infrared Spectroscopy

An Optothermal Infrared Spectroscopy (OTIS) method has been introduced for skin water measurement in-vivo (Frodin T et al, 1988). The technique is based on detection of heat generated in sample owing to absorption of periodic monochromatic radiation, where water has a highly specific absorption band. The heat is conducted to a thin sapphire plate in contact with the skin and transparent to the radiation directed to the test area. The subsequent thermal expansion of the plate is transformed to an electrical signal by an annular piezo-electric crystal cemented to the periphery of the plate. The signal is fed into a lock-in amplifier and recorded on a pen-recorder. The signal detection depth is depending on the chopping frequencies, the higher frequencies the shorter the detection depth, the lower frequencies the longer the detection depth.

1.3.7 Magnetic Resonance Imaging

Magnetic Resonance Imaging (MRI) is a technique for producing images with good spatial resolution and tissue differentiation. MRI has been used as a diagnostic technique since 1980 and is now frequently employed because of its non-invasive nature. The parameters which produce contrast between different tissues are essentially relaxation times (T1 and T2) and proton density. They can be measured in a localized area and are related to mobile water content. Transverse relaxation time (T2) is a more sensitive indicator of changes in the amount of mobile water than longitudinal relaxation time (T1). Proton density might seem to be a potentially superior index of skin hydration, but the measurement of this parameter is more difficult than that of T2, and, in addition, needs an external reference (Franconi F et al, 1995). In principle, the method can also be used to make absolute, not just relative, measurements of the water content of skin in-vivo. MRI can readily locate and distinguish water in body tissue. MRI enables one to

study proton exchange phenomena, making it possible to move beyond the simplistic concepts of “bound” water and “free” water (Ablett S et al, 1991, Duce SL et al, 1992). But MRI has a relatively low spatial resolution and sensitivity for SC because of its unusual dryness and low thickness (Elsner P et al, 1994).

1.3.8 X-ray Diffraction

The structure (lipid, keratin and protein) of SC has been investigated by Small-angle and Wide-angle X-ray Diffraction measurements (Swanbeck G, 1959, Bouwstra JA et al, 1992, Cornwell PA et al, 1994). The intercellular bilayers and intracellular keratin structures did not change with hydration level, this was taken to indicate that there is no water absorbed between the lamellar regions of human SC. A disordering of the lipid lamellar structure was observed when SC was heated to about 60°C. However, no changes in the protein structure were observed. Because water is a strong X-ray absorber, the overall intensity and resolution of the diffraction patterns decrease with increasing water content of the sample.

1.3.9 Ultrasound

A typical dermatological ultrasound scanner uses echo reflections of pulsed high frequency ultrasound (>10MHz) from a transducer. The time lag between emitted and reflected sound waves depends on the physical distance between the surface of the object and the different layers of the object, which might reflect the sound. In A-mode, scans produce intensity-time curves, with peaks representing echoes from different layers. The distance between peaks within the object is easily calculated when the intra-object velocity of sound is known. In B-mode scanning, the transducer is automatically moved tangentially over the object and a number of A-mode scans are depicted and processed electronically. In C-mode scanning, a horizontal picture is depicted. M-mode scanning is a special procedure in which such structures and their motility pattern may be characterized (Berardesca E et al, 1995).

1.3.10 Structured-light 3D Scanner

A structured-light 3D scanner is a 3D scanning device for measuring the three-dimensional shape of an object using projected light patterns and a camera system.

Structured-light 3D scanners project a pattern of light on the subject and look at the deformation of the pattern on the subject. The pattern is projected onto the subject using either an LCD projector or other stable light source. A camera, offset slightly from the pattern projector, looks at the shape of the pattern and calculates the distance of every point in the field of view.

Structured-light scanning is still a very active area of research with many research papers published each year. Perfect maps have also been proven useful as structured light patterns that solve the correspondence problem and allow for error detection and error correction (Morano, R., et al, 2002).

The advantage of structured-light 3D scanners is speed and precision. Instead of scanning one point at a time, structured light scanners scan multiple points or the entire field of view at once. Scanning an entire field of view in a fraction of a second reduces or eliminates the problem of distortion from motion. Some existing systems are capable of scanning moving objects in real-time. VisionMaster creates a 3D scanning system with a 5-megapixel camera – 5 million data points are acquired in every frame.

A real-time scanner using digital fringe projection and phase-shifting technique (certain kinds of structured light methods) was developed, to capture, reconstruct, and render high-density details of dynamically deformable objects (such as facial expressions) at 40 frames per second (Song Zhang, 2006). Recently, another scanner has been developed. Different patterns can be applied to this system, and the frame rate for capturing and data processing achieves 120 frames per second. It can also scan isolated surfaces, for example, two moving hands (Kai Liu, 2010). By utilizing the binary defocusing technique, speed breakthroughs have been made that could reach hundreds of (Song Zhang, 2010) to thousands of frames per second (Yajun Wang, 2011).

1.3.11 Optical Coherence Tomography

Optical coherence tomography (OCT) is an established medical imaging technique that uses light to capture micrometer-resolution, three-dimensional images from within optical scattering media (e.g., biological tissue). OCT is based on low-coherence interferometry, typically employing near-infrared light. The use of relatively long wavelength light allows it to penetrate into the scattering medium. Confocal microscopy, another optical technique, typically penetrates less deeply into the sample but with higher resolution.

Depending on the properties of the light source (superluminescent diodes, ultrashort pulsed lasers, and supercontinuum lasers have been employed), optical coherence tomography has achieved sub-micrometer resolution (with very wide-spectrum sources emitting over a ~100 nm wavelength range).

1.3.12 Diffusion Tensor Imaging

Diffusion tensor imaging (DTI) is a magnetic resonance imaging technique that enables the measurement of the restricted diffusion of water in tissue in order to produce neural tract images instead of using this data solely for the purpose of assigning contrast or colours to pixels in a cross sectional image. It also provides useful structural information about muscle—including heart muscle—as well as other tissues such as the prostate (Manenti et al, 2007).

In DTI, each voxel has one or more pairs of parameters: a rate of diffusion and a preferred direction of diffusion—described in terms of three-dimensional space—for which that parameter is valid. The properties of each voxel of a single DTI image is usually calculated by vector or tensor math from six or more different diffusion weighted acquisitions, each obtained with a different orientation of the diffusion sensitizing gradients. In some methods, hundreds of measurements—each making up a complete image—are made to generate a single resulting calculated image data set. The higher information content of a DTI voxel makes it extremely sensitive to subtle pathology in the brain. In addition, the directional

information can be exploited at a higher level of structure to select and follow neural tracts through the brain—a process called tractography (Filler, Aaron, 2009).

A more precise statement of the image acquisition process is that the image-intensities at each position are attenuated, depending on the strength (b-value) and direction of the so-called magnetic diffusion gradient, as well as on the local microstructure in which the water molecules diffuse. The more attenuated the image is at a given position, the greater diffusion there is in the direction of the diffusion gradient. In order to measure the tissue's complete diffusion profile, one needs to repeat the MR scans, applying different directions (and possibly strengths) of the diffusion gradient for each scan.

1.3.13 Confocal Raman Spectroscopy

Raman Spectroscopy is a non-destructive technique, based on scattering of light by molecules. The sample under investigation is illuminated with low power laser light. In a Raman scattering event, part of the energy of the incoming light is transferred to a molecule, thereby exciting one of the molecule's vibrational modes.

Every molecule contributes to the overall Raman spectrum of a cell or a tissue, the spectrum is, in fact, a direct representation of the overall molecular composition. As such the Raman spectra can be used as highly specific spectroscopic fingerprints, which enable identification or classification of cells and tissues. The technique is non-destructive and does not require sample preparation, nor reagents, dyes, labels or other contrast enhancing agents. It is perfectly suited for *in vivo* application. Measurements can be carried out in a matter of seconds or less.

The gen2-SCA has been custom-designed for rapid, non-invasive data collection from the skin, including the skin of the scalp and the axilla. It uses red and near-infrared laser light and complies with international standards for safe laser light exposure of the skin.

The gen2-SCA is a highly sensitive confocal Raman system for *in vivo* skin analysis.

- Determine molecular concentration profiles from the skin surface into the dermis, in seconds to minutes, with high spatial resolution.
- Obtain information about the concentration and distribution of amino acids, sweat constituents, lipids, proteins, water, and topically applied materials.
- Study penetration and transdermal delivery of topically applied materials.

Examples of applications are:

- Water concentration profiles from skin surface to dermis in minutes
- Determination of Natural Moisturizing Factor concentration in the stratum corneum
- Studies of the penetration of topically applied materials
- Optimization of formulations for actives delivery

1.4 Summary and Research Objectives

In summary, this chapter presents the basic introduction and review of skin characterization such as skin structure, functions, hydration and trans-epidermal water loss (TEWL). This chapter also describes the latest skin measurement technologies.

The literature reviews show that there are many exciting technologies that have been used for skin measurements, such as recently developed capacitive contact imaging etc. But there are still many questions unanswered, particularly in the areas related to skin hydration, skin TEWL, and skin damage assessments.

Therefore, the aims of this study are to develop new skin imaging analysis algorithm for capacitive contact imaging and other skin imaging, and to have a better understanding in characterization, i.e. skin hydration, skin TEWL and skin damage assessments.

1.5 Organization of the Thesis

Chapter 2 presents the instruments which were used in this thesis for skin hydration and TEWL measurements: OTTER, Aqua Flux, Fingerprint sensor, Corneometer, digital camera and ProScope HR2.

Chapter 3 and Chapter 4 present the skin image processing algorithms and relevant GUI programme such as image matching, stitching, enhancing etc.

Chapter 5 presents three experiments: Skin damage assessment including tape striping, Dermabrasion and SLS-irritation; Skin over-hydration measurements; Soap drying assessment.

Chapter 6 lists the conclusions and future works of the research.

Chapter 2 Skin Measurement Instruments

Measurements of skin hydration of Stratum Corneum (SC) and Trans-epidermal water loss (TEWL) are of great importance in many research areas such as dermatology, clinical analysis, pharmacology and cosmetic science.

Studies show that there are many techniques for skin in-vivo measurements, such as electrical capacitance measurements, evaporimetry, infrared spectrometry, ultrasound, magnetic resonance imaging and wide-angle x-ray diffraction.

In this thesis, three novel instruments are mainly used for skin hydration and TEWL measurements: Opto-Thermal Transient Emission Radiometry (OTTER), AquaFlux and capacitive contact imaging based on Fingerprint sensor. Another four instruments were used for performance assessment comparing with the data from Fingerprint sensor.

In this Chapter, a brief introduction of the instruments will be presented.

2.1 OTTER

Opto-thermal transient emission radiometry (OTTER), shown in Figure 2.1, is an infrared remote sensing technique based on photothermal radiometry (PTR) or Opto-thermal radiometry (OTR). OTTER is invented and developed by the research group. It has many advantages, such as non-contact, non-invasive, work on arbitrary surfaces, in-sensitive to colour and small movements. Due to the large size of the laser generator and cooling system, OTTER is not a portable instrument.



Figure 2.1 The OTTER apparatus.

Figure 2.2 shows a schematic diagram of the OTTER. A Q-switched pulsed laser as the excitation source is used to heat up the sample surface and then generate a heat radiation. Heat radiation, whose wavelength region is in the mid-infrared $\sim 5\text{-}15\mu\text{m}$, is focused by an aluminium ellipsoidal mirror from the sample onto a high speed, liquid nitrogen cooled Mercury Cadmium Telluride (MCT) detector, whose signal is captured by a digital oscilloscope. A PC, linked to the transient recorder as digital oscilloscope through a high-speed parallel interface, is used for signal averaging, data storage, display and analysis. The shape of the measured signal is dependent on the sample's optical and thermal properties, the thickness of the sample and its layer structure.

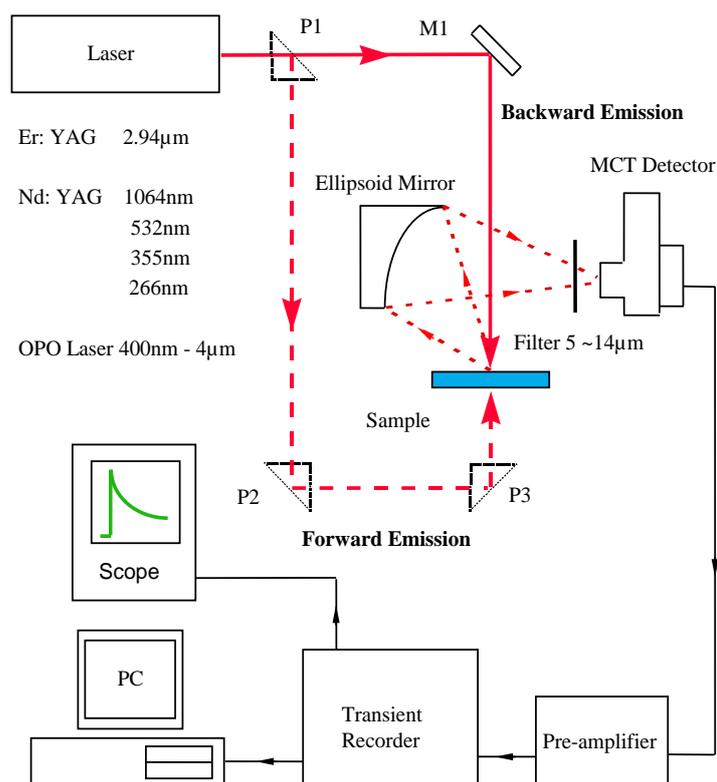


Figure 2.2 The schematic diagram of OTTER, where P1, P2, P3 are prisms, and M1 is a mirror.

In Opto-Thermal physical process, OTTER measurements can be modelled as three sequential physical processes as follows:

- (1) The near-surface region of the sample is heated by absorption of the incident laser energy. This establishes an initial temperature field at time $t=0$.
- (2) The absorbed energy diffuses within the sample and causes the temperature distribution to change with time (transient temperature field).
- (3) Due to the changes of the transient temperature field, the thermal radiation from the sample surface changes. This heat radiation can be detected, recorded and analysed as an opto-thermal signal.

Therefore, Opto-thermal signals can be modelled in three steps: (1) the initial temperature field; (2) the transient temperature distribution; (3) the infrared heat radiation. The main sample properties involved in these three steps: (1) the absorption coefficient for the laser radiation α ; (2) the thermal diffusivity D ; (3) the absorption coefficient for the emitted thermal radiation β .

When laser radiation is incident on a sample surface, some radiation will penetrate into sample as it is absorbed. The initial temperature field can be expressed as:

$$\theta(z, 0) = \frac{E_0 \alpha}{\rho C} e^{-\alpha z} \quad (2.1)$$

Where $\theta(z, 0)$ is the temperature at $t=0$ of sample at position z , with $z=0$ at the surface of the sample and increasing toward the inside of the sample. C is the sample's specific heat, ρ is the sample's density and E_0 is the energy density absorbed from the excitation pulse (Xiao, P, Imhof, R.E 1998).

The temperature re-distribution after the initial disturbance follows the diffusion law, which can conveniently be expressed in Green's function form as:

$$\begin{aligned} \theta(z, t) &= \int_0^t \int_0^\infty \frac{Q(z', t')}{\rho C} G(z, z'; t, t') dz' dt' \\ &= \int_0^\infty \theta(z', 0) G(z, z'; t, t') dz' \end{aligned} \quad (2.2)$$

Where $G(z, z'; t, t')$ is Green's function describing the temperature change in position z at time t due to a Dirac-Delta heat disturbance at position z' and time t' . Therefore the transient temperature field $\theta(z, t)$ can be found by integration of the heat source and Green's function in the semi-space of the sample.

The opto-thermal signal comes from the transient thermal radiation, which is caused by the time-dependent temperature field $\theta(z, t)$ above. The calculation of Opto-thermal signal is given by (Xiao, P, Imhof, R.E 1998):

$$S(t) = \zeta \int_0^{\infty} \beta(z) e^{-\int_0^z \beta(z) dz} \theta(z, t) dz \quad (2.3)$$

Where the parameter $\zeta = \zeta(\lambda_{em})$ includes factors that depend on the blackbody emission curve, detector sensitivity, focusing and alignment, but is independent of the properties of the sample.

By substituting Eq. (2.1) and (2.2) into Eq. (2.3), and using semi-opaque limit, $\alpha \gg \beta$, or $\beta \gg \alpha$, we have

$$S(t) = A_1 e^{t/\tau_{\hat{\alpha}}} \text{erfc} \sqrt{t/\tau_{\hat{\alpha}}} \quad (2.4)$$

Where $\hat{\alpha}$ is either α or β , $\tau_{\hat{\alpha}} = \frac{1}{\hat{\alpha}^2 D}$, $A_1 = \frac{\zeta E_0 \hat{\alpha}}{\rho C}$ is the amplitude of the signal and $\text{erfc}(x) = 1 - \text{erf}(x)$ is the complementary error function,

$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-y^2} dy$ is error function. The least-squares fitting of all signals of

OTTER measurements in this thesis is done by using Eq. (2.4).

OTTER has demonstrated the capability for measuring skin hydration, skin pigments, skin thickness and transdermal drug delivery. However, Skin measurement is not the only application of OTTER measurements. OTTER can also be used as a non-destructive evaluation technology for many industrial applications, such as (Xiao, P, Imhof, R.E 1998):

- (1) Measurement of thermal properties of materials.
- (2) Measurement of optical properties of materials.
- (3) Measurement of paint thickness.

(4) Measurement of thermal resistance of materials.

(5) Material characterization.

(6) Flaw detection and characterization.

2.2 AquaFlux

AquaFlux (Biox Systems Ltd), shown in Figure 2.3, originally developed by the research group, is a novel condenser-based, closed chamber technology for measuring water vapour flux density from arbitrary surfaces, including in-vivo measurements of trans-epidermal water loss (TEWL), skin surface water loss (SSWL) and perspiration.



Figure 2.3 AquaFlux

It uses a cylindrical measurement chamber, with one end open and attached to the sample surface, and another end closed and cooled down to below the water freezing point. Figure 2.4 shows a simple cutout diagram of the AquaFlux condenser-chamber. When the chamber is brought in contact with the skin surface, the chamber is sealed and it can protect the diffusion zone within it from ambient

air movement. The natural convection and other bulk air movements are brought down because of the internal dimensions of the chamber being very small. In fluid dynamics terms, this requires the Rayleigh Number to be below the critical value for its geometry. Under these conditions, passive diffusion remains the only transport mechanism for the water vapour entering the chamber.

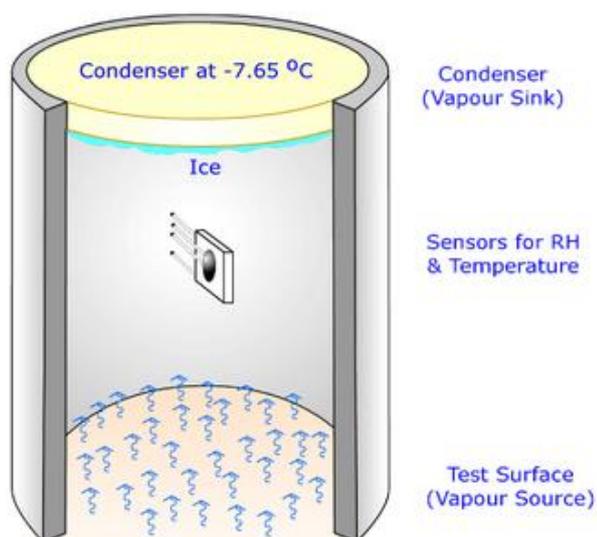


Figure 2.4 A simple cut-out diagram of the AquaFlux condenser-chamber from (AquaFlux, Biox systems Ltd)

The condenser controls the absolute humidity in the measurement chamber independently of ambient conditions. It is similar to a vapour sink by forming ice on its surface, thus creating a zone of low humidity in its immediate vicinity. By contrast, the test surface acts as a vapour source, creating a zone of higher humidity in its immediate vicinity. This humidity difference causes water vapour to migrate from source to sink by passive diffusion and creates a diffusion vapour density gradient from which the flux density can be measured (Imhof, R.E. et al, 1999).

Since the size of sample is normally much greater than the measurement orifice, the measurement principles are discussed under one dimension, and the Cartesian coordinate system is used, shown in Figure 2.5.

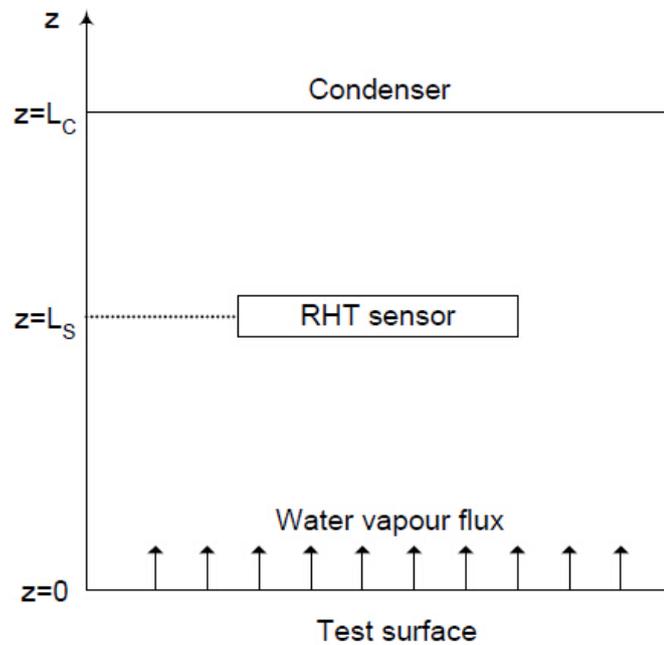


Figure 2.5 Geometry of 1-D AquaFlux modeling from (Imhof, R.E. et al, 1999)

As mentioned above, the internal space of the measurement chamber is small enough to suffocate natural convection and other bulk air movement. Therefore, diffusion is the only transport mechanism available, so bulk air movement is ignored, and thus, the migration of the water molecules in the chamber can be calculated using Fick's first law of diffusion:

$$J = -D_{VA} \frac{\partial H}{\partial z} \quad (2.5)$$

Where J is the water vapour flux density from the sample surface, D_{VA} is the molecular diffusion coefficient for water vapour in air ($D_{VA} = 2.42 \times 10^{-5} \text{ m}^2 / \text{s}$) at room temperature 20°C , and H is the water vapour humidity.

Similarly, the temperature transfer is expressed as:

$$Q = -K_A \frac{\partial \theta}{\partial z} \quad (2.6)$$

Where Q is the heat flux density, K_A is the thermal conductivity of air, and θ is the temperature distribution in the chamber.

Immediately after the vapour enters the chamber from the sample, it will be removed by the condenser. The steady-state vapour distribution is established. The steady-state solution is characterized by the constant water vapour humidity gradient $\frac{\partial \theta}{\partial z}$ in the chamber, whence the water vapour flux density from the sample surface. Therefore,

$$J = D_{VA} \frac{H(z) - H(L_C)}{L_C - z} \quad (2.7)$$

In the AquaFlux measurements, the water vapour humidity $H(z)$ is calculated using the following equations.

$$H(z) = \chi \rho_{VE}(\theta_z) \quad (2.8)$$

Where χ is the fractional RH value, $\rho_{VE}(\theta)$ is the equilibrium vapour density at temperature θ , and θ_z is the temperature at position z .

Then Eq. (2.8) can be rewritten as:

$$J = \frac{D_{VA}}{L_C - z} [\chi_z \rho_{VE}(\theta_z) - \chi_C \rho_{VE}(\theta_C)] \quad (2.9)$$

Where χ_z and χ_C are the fractional RH values at z and at the condenser respectively, θ and θ_z are the temperature at position z are the temperatures at z and at the condenser respectively, and $\rho_{VE}(\theta_z)$ and $\rho_{VE}(\theta_C)$ are the equilibrium vapour densities at z and at the condenser respectively.

Due to the thermodynamic equilibrium between the vapour phase near the condenser and the solid phase at the condenser (ice), $\chi_C = 1$ at the condenser. The Eq. (2.9) can be simplified as:

$$J = \frac{D_{VA}}{L_C - z} [\chi_z \rho_{VE}(\theta_z) - \rho_{VE}(\theta_C)] \quad (2.10)$$

Some extreme cases of the above equation are important for the AquaFlux measurements. When $z = L_S$, Eq. (2.9) is rewritten as:

$$J = \frac{D_{VA}}{L_C - L_S} [\chi_S \rho_{VE}(\theta_S) - \rho_{VE}(\theta_C)] \quad (2.11)$$

Where θ_S and χ_S are the temperature and the fractional RH value at the RHT sensor respectively, and above equation allows the flux density to be determined from measurements of, χ_S , θ_S and θ_C .

When $z=0$, Eq. (2.10) becomes:

$$J = \frac{D_{VA}}{L_C} [\chi_0 \rho_{VE}(\theta_0) - \rho_{VE}(\theta_C)] \quad (2.12)$$

Where θ_S and χ_S are the temperature and the fractional RH value at the sample surface.

The above equation specifies the microclimate immediately above the sample surface. With J calculated from Eq. (2.11), the fractional RH value at the sample surface can be calculated.

Comparing with other water vapour flux density measurement instruments, such as Tewameter (Courage + Khazaka Electronic GmbH, Germany), DelfinVapoMeter (Delfin Technologies, Finland) and Evaporimeter (Cortex Technology ApS, Denmark), AquaFlux has many advantages, for example, high sensitivity, low noise, super reliability, excellent calibration ability and comparability, and above all, the measurement results are independent of external environment. AquaFlux can be used in the following areas (Imhof, R.E. et al 1999):

- (1) Skin barrier function, skin recovery, skin disease etc., based on TEWL measurements.
- (2) Membrane penetration and permeation.
- (3) Material absorption/desorption.

2.3 Capacitive Contact Imaging Based on Fingerprint sensor

The water content within human skin is very important for its cosmetic properties and its barrier functions; however, to measure it is very difficult. There are some disadvantages of the commercial skin hydration measurement devices, such as, poor repeatability, reproducibility and difficult to calibrate. To address these issues, a novel hand-held probe for in-vivo skin hydration imaging using capacitive Fingerprint sensors has been developed by the research group. It is based on capacitance measurement principle of capacitors.

Commonly, a capacitor is used to store (hold) the electrical charge. A capacitor is constructed of two conductive parallel plates separated by a non-conductive

medium or the dielectric region. Assuming that the parallel plates are charged with $+Q$ and $-Q$ that the voltage between the plates is V , the capacitance can be shown as:

$$C = \frac{Q}{V} \tag{2.13}$$

The unit of measurement used for measuring capacitance is farad, where 1 farad is equal to 1 coulomb per volt (Xiang, Ou, 2014).

The capacitor stores energy which is equal to the energy required to charge it and thus electrical energy is not dissipated.

If a small element of charge dq is moved across the two plates against the potential difference $V = q/C$ it will require the amount of energy dW :

$$dW = \frac{q}{V} dq \tag{2.14}$$

This energy fluctuates depending on the interference of the dielectric medium.

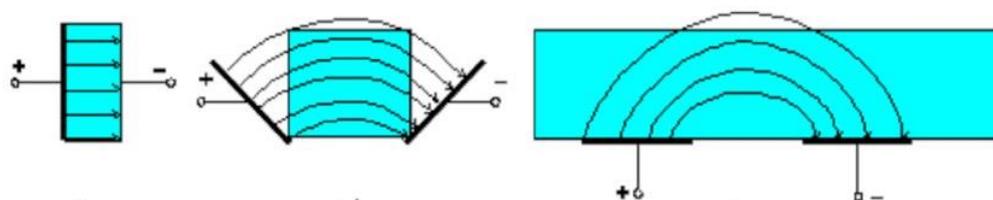


Figure 2.6 Two parallel capacitor plates are moved into coplanar position with fringing electric fields (Li Xiaobei, 2002).

Fingerprint sensors are based on the capacitance fringing field measurement principle (Singh, H. 2010). The parallel plates of the capacitor are split to a

coplanar position; the fringing electric field will continue to create a field between the two plates, see Figure 2.6 (Li Xiaobei, 2002).

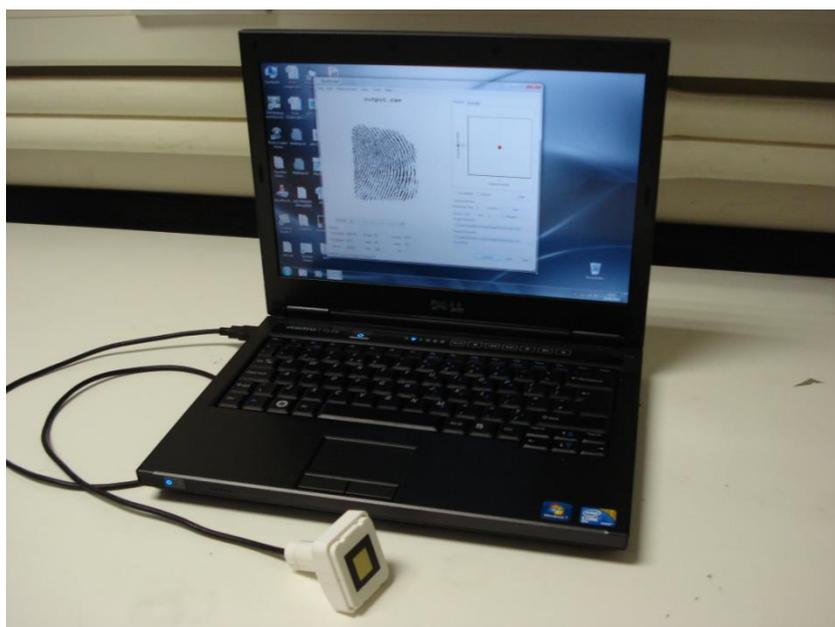
Each set of a parallel plate is regarded as a cell and each cell in turn contains a feedback capacitance known as a sensor plate. After a side-by-side array of alternating charged and sensor conductor plates are created, the alternating plates form the two plates of the capacitor. Meanwhile, any interference in between the plates causes the dielectric fields to change between the conductor plates. One set of conductor plates makes a single cell whose size is larger than the width of one ridge on the skin. The print image is created when the sensor detects variance in dielectric constants between the two.

Each set of the conductor plates (or cells) contains their own electrical circuit which the electrical circuit creates an inverting operational amplifier that alters the voltage being supplied thus making them tiny groups of semiconductor chips. The altered voltage is connected to the inverting terminal input and the non-inverting terminal input is relative to a reference voltage supply and a feedback loop. This feedback loop is in turn connected to the amplifier output which includes two conductor plates.

The Fingerprint sensor is non-invasive, quick, simple to use, imaging based and is based on latest fingerprint sensor technologies and using well researched mathematical imaging processing algorithms. Compared with thermal sensors, capacitor technology offers better image quality with a wider range of operating conditions. It can measure any part of the body, and generate 256×300 pixels black and white skin capacitance images with a $50 \mu m \times 50 \mu m$ special resolution. In each image, each pixel is represented by an 8 bit greyscale value, 0~255, with 0 represents white (i.e. dry skin - low capacitance) and 255 represents black (i.e. wet skin- high capacitance).

The prototype of capacitive contact imaging based on Fingerprint sensor shown in Figure 2.7(a) has many functions:

- (1) By processing those capacitive skin images using a dedicated software programme with purposely designed mathematical algorithms, we can produce skin hydration images, skin micro relief images and skin 3D surface profile images.
- (2) It can also be used for study soft materials.
- (3) It can also be used to study solvent membrane penetrations. Apart from water,
- (4) Fingerprint sensor is also sensitive to many the solvents using in pharmaceutical studies.



(a) The prototype of capacitive contact imaging based on Fingerprint sensor.



(b) *Epsilon permittivity imaging system.*

Figure 2.7 Development of capacitive imaging system.

By using AquaImager software, developed specifically for the Fingerprint Sensor by the research group, the skin site is visualized as shown in Figure 2.7(a). It produces the raw images of the skin site which can be processed by MATLAB programme into grey images. To study the difference between digital image and grey images, the instrument is very important and useful.

Based on the Fingerprint sensor, the new Epsilon permittivity imaging system has been developed (Imhof R. E. and Xiao P., 2012), as shown in Figure 2.7(b). The Epsilon differs from other similar systems in its calibrated, linear response to near-surface dielectric permittivity. The linear response is important because hydration is linearly related to permittivity (Pan. W. et al, 2014).

The calibration ensures consistency from instrument to instrument and from time to time. With calibrated Epsilon imaging systems, we can measure the absolute dielectric permittivity of the material (Zhang. X. et al, 2016).

Epsilon shown in Figure 2.7(b) has many functions:

- (1) By processing those images which generated from Epsilon using a dedicated software programme with purposely designed mathematical

algorithms, we can produce skin hydration images, skin micro relief images and skin 3D surface profile images.

- (2) It can also be used for study materials.
- (3) With Epsilon, we can also study membrane penetration. Apart from water, Epsilon is also sensitive to most the solvents using in pharmaceutical studies.

2.4 Corneometer

The Corneometer measures changes in capacitance using a contacting probe that forms a capacitor with the skin under the test. Since the dielectric permittivity of water is high compared with those of other skin constituents, the hydration of the skin in contact with the probe affects the capacitance sensed by the probe (Courage W, 1994).

The instrument is made up of a hand-held probe and a central processing unit, as shown in Figure 2.8 below (O`Driscoll. D., 2000).



Figure 2.8 View of Corneometer CM825

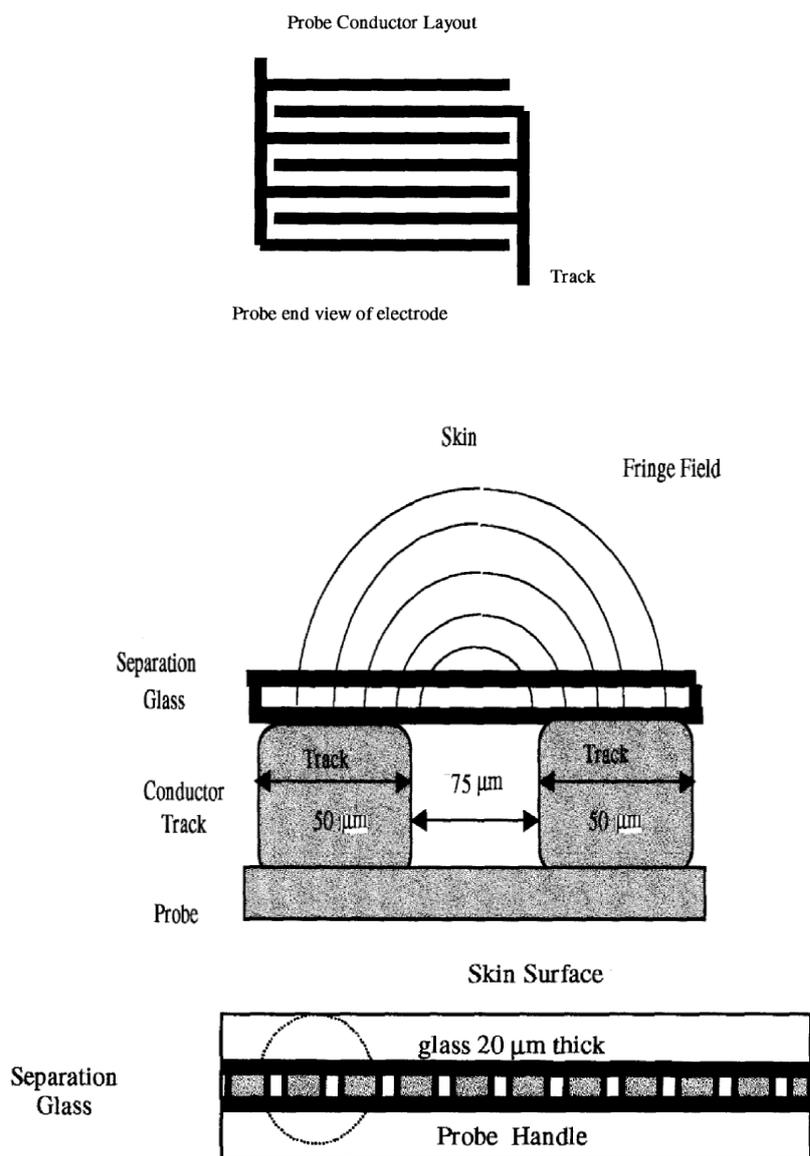


Figure 2.9 Corneometer probe head design conductor track and cross section (reproduced from O'Driscoll, D., 2000).

The probe head is made-up of a 7x7-mm size ceramic base with gold-plated conductor lines forming an interlaced grid pattern with spacing of 75 μm and a surface area of 0.5 cm^2 . A view of the probe head with a cross section view is shown in Figure 2.9. The conductor tracks are protected by a separation glass where the test skin surface will be in contact. The glass cover is of 20 μm thickness.

Due to the small size and lightweight of the probe, it allows quick measurement and continual measurement important to avoid occlusion, easy handling and cleaning, and measuring on all body sites.

The measurement depth is 10-20 μ m of the SC, to avoid the influence of dermis layer. Measurements on an only 15 μ m thick foil show only 20% of the original measurement value. The Corneometer measures in arbitrary units from 0 to 120. The units have been very well established as so called "Corneometer units".

2.5 Digital Camera

For this research, the digital camera used is SONY DSC-W55. The camera has a 7.2 Mega Pixels with 3X optical zoom shown in Figure 2.10.



Figure 2.10 View of the SONY DSC-W55 camera.

In this thesis, the digital images captured by the camera were used for colour image processing.

2.6 ProScope HR2 Digital Microscope

The view of ProScope HR2 shows in figure 2.11 which has larger pixels that take in more light than other instruments on the market. The Bodelin engineering team customized this imager to allow extensive imaging adjustment and LED intensity control. The most impressive feature is the ability to actually stream true pixels at higher resolutions. The total effect in the ProScope HR2 is live video that appears to be 4 to 5 megapixels in clarity while only using an actual 2 megapixel imager. The ProScope products are currently used in tens of thousands of schools, universities, law enforcement forensic labs, manufacturing quality control, medical and tele-medicine since 2001. The model used in this study uses an optical lens of 30x zoom with cross polarized light that can take images from both skin surface and underneath the skin surface.

(<http://www.bodelin.com/proscope/proscope-hr2>)



Figure 2.11 View of ProScopeHR2 digital microscope.

In this thesis, to figure out the difference between colour image processing and grey image processing, the images taken from the ProScope HR2 are compared with Fingerprint sensor.

2.7 Summary

In summary, all the skin measurement instruments which were used in the thesis are discussed in this chapter. It has to be admitted that although all of them have great performance in skin in-vivo measurement, there are some weakness by their own.

OTTER is an infrared remote sensing technique for the Non-Destructive Examination (NDE) of sample surfaces, through their optical and thermal properties. It has advantages such as non-contact, non-invasive, work on arbitrary surfaces, in-sensitive to colour and small movements. However, due to the large size of the laser and cooling system, OTTER is not a portable instrument.

AquaFlux is a novel technology for measuring water vapour flux density from arbitrary surfaces. The closed chamber design perfectly cuts off the external facts and occlusion which influence the TEWL result on the experimental skin site. It allows quick measurement and continual measurement, easy handling, and measuring on all body sites. It is an excellent instrument of TEWL measurement with better repeatability and better precision.

Capacitive contact imaging based on Fingerprint sensor is a novel fringing field capacitive imaging technique, which can be used for skin hydration measurements. It is a quick measuring, high resolution imaging, easy handling and cleaning, and visual imaging tool for skin in-vivo measurements. It produces the raw images of the skin site which can be processed by MATLAB programme into grey images in order to figure out the difference between digital image and grey images. The only weakness is that the occlusion of skin site highly affects the skin hydration during the measurement.

Corneometer is made up of a hand-held probe and a central processing unit. Due to the small size of the probe, it allows quick measurement and continual measurement easy to avoid occlusion and measuring on all body sites. The Corneometer measures in arbitrary units from 0 to 120 in a small area 7x7-mm. It is not suitable for measuring the large skin sites because of the limited size of the probe head.

Digital camera and ProScope HR2 microscope were used for capturing digital images. ProScope is preferred to measure the images by the 30x zoomed camera for comparing with the raw images from Fingerprint sensor.

Chapter 3 Skin Image Processing using MATLAB

Image processing is very important in our skin research, as several skin imaging techniques are used in our group, such as the digital camera, ProScope HR2 digital microscope, and the latest capacitive contact imaging. This chapter first gives an overview of the imaging processing techniques and the MATLAB software, then describes the image processing work carried out in this project using MATLAB software.

MATLAB (MATrix LABoratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. A proprietary programming language developed by MathWorks, MATLAB allows.

- (1) Math and computation
- (2) Algorithm development
- (3) Modelling, simulation, and prototyping
- (4) Data analysis, exploration, and visualization
- (5) Scientific and engineering graphics
- (6) Application development, including Graphical User Interface building.

In computer science, a graphical user interface (GUI), is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation. Its goal is to allow users implement MATLAB functions without learning any programming skills.

There are the techniques in the GUI programme presented in this thesis for image processing including:

- (1) Noise generator: Gaussian constant, Poisson, salt&pepper, speckle.
- (2) Noise filters: Gaussian, average, median, unsharp, motion.
- (3) Equalization: RGB components, contrast, brightness.
- (4) Digital image processing: RGB components separating, image converting, image enhancing, image 3-D plot.
- (5) Edge detection: Canny, Sobel, log, Prewitt, Roberts.
- (6) Feature enhancement: Gabor feature detection and matching, dilation and erosion.

All of the functions above can be used on any types of the images.

3.1 Image processing Techniques

In imaging science, the image processing is processing of images using mathematical operations by using any form of signal processing for which the input is an image, a series of images, or a video, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image (Rafael C et.al.2008).

3.1.1 Filters

In image processing, filtering is a process that removes some unwanted components or features from the images such as background, brightness, fog and smoke etc.

To achieve this, relevant filter functions have been added to the GUI programme. The images also need to be processed into numerical or matrix. It is well known that the Fourier transform has been the most commonly used tool for analysing frequency properties of a given signal. However, it has some limitations, where the information about time is lost.

In mathematics, a wavelet series is a representation of a square-integrable (real- or complex-valued) function by a certain orthonormal series generated by a wavelet. Nowadays, wavelet transformation is one of the most popular of the time-frequency-transformations.

Wavelet transform could extract both the time (spatial) and frequency information from a given signal, and the tuneable kernel size results in different time-frequency resolution pairs. In other words, the image can be seen under the lens with a magnification given by the scale of a wavelet. This great character also makes wavelet transform suitable for applications such as image edge detection, filter design, and some kinds of image object recognition, etc.

Wavelets, as the name suggests, are ‘little waves’. If a single real valued function $\psi(t) \in L^2(\mathbb{R})$, namely, $\psi(t)$ must satisfy the admissibility condition:

$$\int_0^{\infty} \frac{|\psi(\omega)|^2}{\omega} d\omega < \infty \quad (3.1)$$

Where $\psi(\omega)$ denotes the Fourier transform of $\psi(t)$ and $L^2(\mathbb{R})$ represents the space of finite- energy (also known as square – integral) functions.

Then the function $\psi(t)$ is called the basic wavelet or mother wavelet.

Once one has a mother wavelet, one can then generate wavelets by the operations of dilation and translation as follow:

$$\begin{aligned} \psi_{s,\tau}(t) &= s^{-1/2} \psi\left(\frac{t-\tau}{s}\right) \\ s, \tau &\in \mathbb{R} \quad s > 0 \end{aligned} \quad (3.2)$$

Where s is dilation parameter and τ is translation parameter.

There are five different filters in the GUI programme presented in this chapter: Gaussian filter, average filter, median filter, unsharp filter and motion filter.

(1) Gaussian filter

In electronics and signal processing, a Gaussian filter is a filter whose impulse response is a Gaussian function (or an approximation to it). Gaussian filters have the properties of having no overshoot to a step function input while minimizing the rise and fall time. This behaviour is closely connected to the fact that the Gaussian filter has the minimum possible group delay. Mathematically, a Gaussian filter modifies the input signal by convolution with a Gaussian function.

The one-dimensional Gaussian filter has an impulse response given by

$$g(x) = \sqrt{\frac{a}{\pi}} e^{-a|x|^2} \quad (3.3)$$

and the frequency response is given by the Fourier transform

$$\hat{g}(f) = e^{-\frac{x^2 f^2}{a}} \quad (3.4)$$

with f the ordinary frequency. These equations can also be expressed with the standard deviation as parameter

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (3.5)$$

and the frequency response is given by

$$\hat{g}(f) = e^{-\frac{f^2}{2\sigma_f^2}} \quad (3.6)$$

By writing a as a function of σ with the $g(x)$ two equations for $\hat{g}(f)$ and as a function of σ_f with the two equations for $\hat{g}(f)$ it can be shown that the product of the standard deviation and the standard deviation in the frequency domain is given by

$$\sigma \cdot \sigma_f = \frac{1}{2\pi} \quad (3.7)$$

where the standard deviations are expressed in their physical units, e.g. in the case of time and frequency in seconds and Hertz. In two dimensions, it is the product of two such Gaussians, one per direction:

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.8)$$

Where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution.

(2) Gabor filter

In image processing, a Gabor filter, named after Dennis Gabor, is a linear filter used for edge detection. Frequency and orientation representations of Gabor filters are similar to those of the human visual system (Kruizinga P, 1999), and they have been found to be particularly appropriate for texture representation and discrimination. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave (Marčelja S.1980).

Simple cells in the visual cortex of mammalian brains can be modelled by Gabor functions (Daugman. J.G., 1985). Thus, image analysis with Gabor filters is thought to be similar to perception in the human visual system. Gabor wavelet

texture feature (Yang. J. 2003) are the conventional methods used to describe texture feature. Gabor wavelet transform has been used in optical character recognition, iris recognition and fingerprint recognition.

The typical 2D Gabor function can be expressed as the product of Gaussian function and sinusoidal function (Fischer S, 2001):

$$g(x, y) = \left[\frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)} \right] \times (e^{2\pi jfx}) \quad (3.9)$$

Where σ_x and σ_y are the Gaussian variance, which describe the spreads of the Gaussian function, j is the imaginary part of complex number, f is the frequency of the sinusoidal function. Using Eq. 3.1 as the mother function, we can generate a set of child functions, called Gabor wavelets.

$$g_{mn}(x, y) = a^{-m} g(x', y') \quad a > 1, m, n = \text{Integer}$$

$$x' = x \cos \theta + y \sin \theta$$

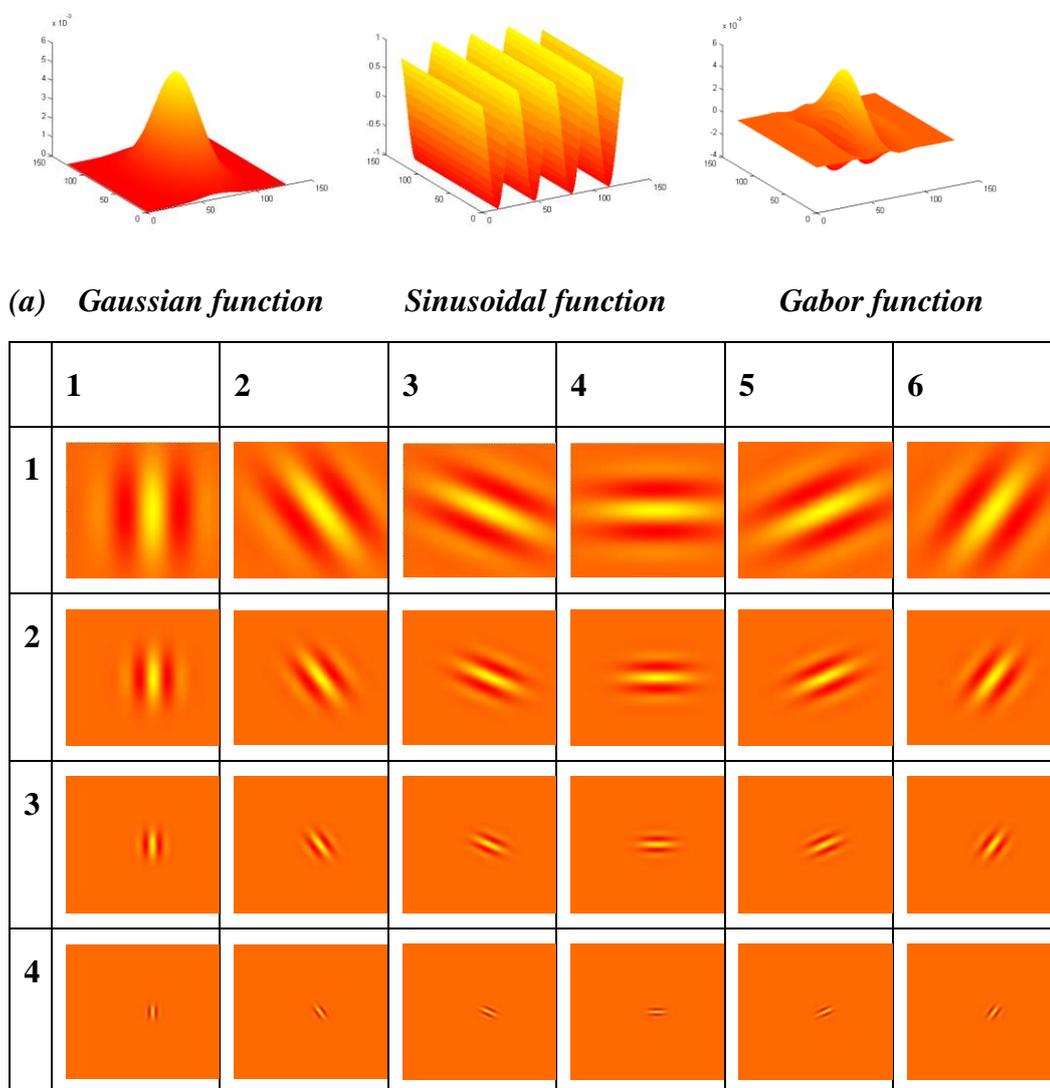
$$y' = -x \sin \theta + y \cos \theta \quad (3.10)$$

Where $\theta = n\pi/K$, $n = 0, 1, \dots, K-1$, and K is the total number of the directions which specifies the orientation of a Gabor function; $m = 0, 1, \dots, S-1$, and S is the number of scales which specifies the amplitude of a Gabor function. If we use (U_l, U_h) to denote the lower and upper centre frequency of the sinusoidal function, we have

$$a = \left(\frac{U_h}{U_l} \right)^{\frac{1}{s-1}}$$

$$f = U_h \quad (3.11)$$

Figure 3.1 shows typical 2D Gabor wavelet profile with different directions and scales.



(b) Views of 2D Gabor wavelet profiles at 6 different directions (columns) and 4 scales (rows).

Figure 3.1 Views of 2D Gabor wavelet profile with different directions and scales.

Image feature extraction based on 2D Gabor wavelet transform is introduced as follows (Yu L, 2008)

Gabor wavelet transform can be considered as a wavelet transform whose mother wavelet is Gabor function. For a given image $I(x, y)$ with $M \times N$ pixels, its Gabor wavelet transform is defined as follows:

$$W_{mn}(x, y) = \int I(x_1, y_1) g_{mn}^*(x - x_1, y - y_1) d_{x_1} d_{y_1} \quad (3.12)$$

Where * represents the complex conjugate.

In this chapter, U_l and U_h used are 0.05 and 0.4, the total number of directions (K) and scales (S) and have chosen to be $K = 6$ and $S = 4$, respectively, which is resulting $4 \times 6 = 24$ Gabor wavelet filters to filter the images.

An array of magnitudes, which represent the energy content at different scales and directions of the image, can be obtained after applying Gabor wavelet filters on the image with different directions and different scale,

$$E(m, n) = \sum_x \sum_y |W_{mn}(x, y)| \quad (3.13)$$

The mean μ_{mn} and the standard deviation σ_{mn} of the magnitude of the transform coefficients are used to indicate the region for classification and retrieval purposes:

$$\mu_{mn} = \frac{\sum_x \sum_y |W_{mn}(x, y)|}{M \times N}$$

$$\sigma_{mn} = \sqrt{\frac{\sum_x \sum_y (|W_{mn}(x, y)| - \mu_{mn})^2}{M \times N}} \quad (3.14)$$

A feature vector F is created using μ_{mn} and σ_{mn} as the feature components. In this paper, scales S and directions K are set to 4 and 6, so the feature vector defined as follows:

$$\bar{f} = [\mu_{00}\sigma_{00}\mu_{01}\dots\mu_{35}\sigma_{35}] \quad (3.15)$$

Distance measurement and correlation measurement are the two main image similarity measurements. In this chapter, distance measurement is used only. The distance between the query images i and the target image j in the database is defined by:

$$d(i, j) = \sum_m \sum_n d_{mn}(i, j) \quad (3.16)$$

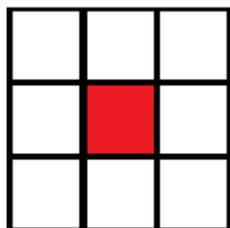
where

$$d_{mn}(i, j) = \left| \frac{\mu_{mn}^{(i)} - \mu_{mn}^{(j)}}{\alpha(\mu_{mn})} \right| + \left| \frac{\sigma_{mn}^{(i)} - \sigma_{mn}^{(j)}}{\alpha(\sigma_{mn})} \right| \quad (3.17)$$

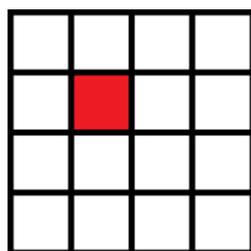
$\alpha(\mu_{mn})$ and $\alpha(\sigma_{mn})$, which are used to normalize the individual feature components, are the standard deviations of the respective features over the entire database.

(3) Median filter

In signal processing, it is often desirable to be able to perform some kind of noise reduction on an image or signal. The median filter is a nonlinear digital filtering technique, often used to remove noise. Such noise reduction is a typical pre-processing step to improve the results of later processing (for example, edge detection on an image). Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise



(a) Median filter block with centre



(b) Median filter block without exact centre

Figure 3.2 Views of Median filter blocks

For 2D (or higher-dimensional) signals such as images, more complex window patterns are possible (such as "box" or "cross" patterns). Figure 3.2 shows the Median filter blocks.

The Median Filter block replaces the central value of an M-by-N neighbourhood with its median value. If the neighbourhood has a centre element as shown in figure 3.2(a), the block places the median value there. Otherwise, without an exact centre as it shown in figure 3.2(b), the block has a bias toward the upper-left corner and places the median value there.

(4) Average filter

As it shows in figure3.5, the Average Filter Block using the same block model. The centre value is replaced by the average filter block.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Figure 3.3 View of Average filter block

(5) Unsharp filter

The unsharp filter is a simple sharpening operator which derives its name from the fact that it enhances edges (and other high frequency components in an image) via a procedure which subtracts an unsharp, or smoothed, version of an image from the original image. The unsharp filtering technique is commonly used in the photographic and printing industries for crispening edges (Fisher R, 2000).

Unsharp masking produces an edge image $g(x, y)$ from an input

image $f(x, y)$ via

$$g(x, y) = f(x, y) - f_{smooth}(x, y) \quad (3.18)$$

where $f_{smooth}(x, y)$ is a smoothed version of $f(x, y)$.

$$f_{sharp}(x, y) = f(x, y) + k * g(x, y) \quad (3.19)$$

Where k is scaling constant. Reasonable values for k vary between 0.2 and 0.7, with the larger values providing increasing amounts of sharpening.

3.1.2 Edge Detection

Edge detection includes a variety of mathematical methods that aim at identifying points in a digital image at which the image brightness changes sharply or, more

formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. The same problem of finding discontinuities in 1-D signals is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction (Umbaugh, Scott E, 2010).

(1) Sobel

The Sobel operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives – one for horizontal changes, and one for vertical (Schar, Hanno, 2000). If we define A as the source image, and G_x and G_y are two images which at each point contain the horizontal and vertical derivative approximations respectively, the computations are as follows:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A \quad \text{and} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * A \quad (3.20)$$

Where * here denotes the 2-dimensional signal processing convolution operation.

Since the Sobel kernels can be decomposed as the products of an averaging and a differentiation kernel, they compute the gradient with smoothing. For example, G_x can be written as

$$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & +1 \end{bmatrix} \quad (3.21)$$

The x-coordinate is defined here as increasing in the "right"-direction, and the y-coordinate is defined as increasing in the "down"-direction. At each point in the

image, the resulting gradient approximations can be combined to give the gradient magnitude, using:

$$G = \sqrt{G_x^2 + G_y^2} \quad (3.22)$$

Using this information, we can also calculate the gradient's direction:

$$\Theta = \text{atan} \left(\frac{G_y}{G_x} \right) \quad (3.23)$$

(2) Roberts

By using the Equations in Sobel model, with different kernels in:

$$\begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}.$$

$$G(x, y) = \sqrt{G_x^2 + G_y^2}.$$

$$\Theta(x, y) = \arctan \left(\frac{G_y(x, y)}{G_x(x, y)} \right). \quad (3.24)$$

(3) Prewitt

By using the Equations in Sobel model, with different kernels in:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * A \quad \text{and} \quad G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix} * A$$

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\Theta = \text{atan} 2(G_y, G_x) \quad (3.25)$$

(4) Canny

Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed. It has been widely applied in various computer vision systems. Canny has found that the requirements for the application of edge detection on diverse vision systems are relatively similar. Thus, an edge detection solution to address these requirements can be implemented in a wide range of situations. The general criteria for edge detection include:

Detection of edge with low error rate, which means that the detection should accurately catch as many edges shown in the image as possible; The edge point detected from the operator should accurately localize on the centre of the edge; a given edge in the image should only be marked once, and where possible, image noise should not create false edges.

The Process of Canny edge detection algorithm can be broken down to 5 different steps (Moeslund, T, 2009)

1. Apply Gaussian filter to smooth the image in order to remove the noise by Eq.3.8
2. Find the intensity gradients of the image. Eq 3.22, Eq 3.25.
3. Apply non-maximum suppression to get rid of spurious response to edge detection
4. Apply double threshold to determine potential edges
5. Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

3.1.3 Morphology

Morphology is a technique which most commonly applied to digital images for the analysis and processing of geometrical structures, based on set theory, lattice theory, topology, and random functions, but it can be employed as well on graphs, surface meshes, solids, and many other spatial structures.

The basic morphological operators are erosion, dilation, opening and closing. In this thesis, erosion and dilation are used.

Erosion is one of two fundamental operations (the other being dilation) in morphological image processing from which all other morphological operations are based. It was originally defined for binary images, later being extended to greyscale images, and subsequently to complete lattices.

In greyscale morphology, images are functions mapping an Euclidean space or grid E into

$$\mathbb{R} \cup \{\infty, -\infty\} \quad (3.26)$$

where \mathbb{R} is the set of reals, ∞ is an element larger than any real number, and $-\infty$ is an element smaller than any real number.

Greyscale structuring elements are also functions of the same format, called "structuring functions". Denoting an image by $f(x)$ and the structuring function by $b(x)$, the greyscale dilation of f by b is given by

$$(f \oplus b)(x) = \sup_{y \in E} [f(y) + b(x - y)] \quad (3.27)$$

Where "sup" denotes the supremum. Similarly, the erosion of f by b is given by

$$(f \ominus b)(x) = \inf_{y \in E} [f(y) - b(y - x)] \quad (3.28)$$

It is common to use flat structuring elements in morphological applications. Flat structuring functions are functions $b(x)$ in the form

$$b(x) = \begin{cases} 0, & x \in B, \\ -\infty, & \text{otherwise} \end{cases} \quad (3.29)$$

Where $B \subseteq E$. In this case, the dilation and erosion are greatly simplified, and given respectively by

$$(f \oplus b)(x) = \sup_{z \in B^s} f(x + z) \quad \text{and} \quad (f \ominus b)(x) = \inf_{z \in B} f(x + z) \quad (3.30)$$

In the bounded, discrete case (E is a grid and B is bounded), the supremum and infimum operators can be replaced by the maximum and minimum. Thus, dilation and erosion are particular cases of order statistics filters, with dilation returning the maximum value within a moving window (the symmetric of the structuring function support B), and the erosion returning the minimum value within the moving window B .

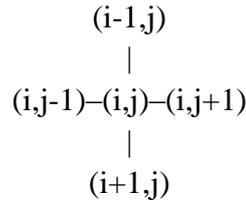
In the case of flat structuring element, the morphological operators depend only on the relative ordering of pixel values, regardless their numerical values.

3.1.4 3-D Image Profiling

Abstractly, a parametric surface is parameterized by two independent variables, i and j , which vary continuously over a rectangle; for example, $1 \leq i \leq m$ and $1 \leq j \leq n$. The three functions $x(i,j)$, $y(i,j)$, and $z(i,j)$ specify the surface. When i and j are integer values, they define a rectangular grid with integer grid points. The

functions $x(i,j)$, $y(i,j)$, and $z(i,j)$ become three m -by- n matrices, X , Y , and Z . Surface colour is a fourth function, $c(i,j)$, denoted by matrix C .

Each point in the rectangular grid can be thought of as connected to its four nearest neighbours.



This underlying rectangular grid induces four-sided patches on the surface. To express this another way, $[X(:) Y(:) Z(:)]$ returns a list of triples specifying points in 3-space. Each interior point is connected to the four neighbours inherited from the matrix indexing. Points on the edge of the surface have three neighbours; the four points at the corners of the grid have only two neighbours. This defines a mesh of quadrilaterals or a quad-mesh.

Surface colour can be specified in two different ways: at the vertices or at the centres of each patch. In this general setting, the surface need not be a single-valued function of x and y . Moreover, the four-sided surface patches need not be planar. For example, you can have surfaces defined in polar, cylindrical, and spherical coordinate systems.

The shading function sets the shading. If the shading is `interp`, C must be the same size as X , Y , and Z ; it specifies the colours at the vertices. The colour within a surface patch is a bilinear function of the local coordinates. If the shading is `faceted` (the default) or `flat`, $C(i,j)$ specifies the constant colour in the surface patch:

$$\begin{array}{ccc} (i,j) & - & (i,j+1) \\ | & C(i,j) & | \\ (i+1,j) & - & (i+1,j+1) \end{array}$$

In this case, C can be the same size as X , Y , and Z and its last row and column are ignored. Alternatively, its row and column dimensions can be one less than those of X , Y , and Z .

3.1.5 Histogram Equalization

This method usually increases the global contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values.

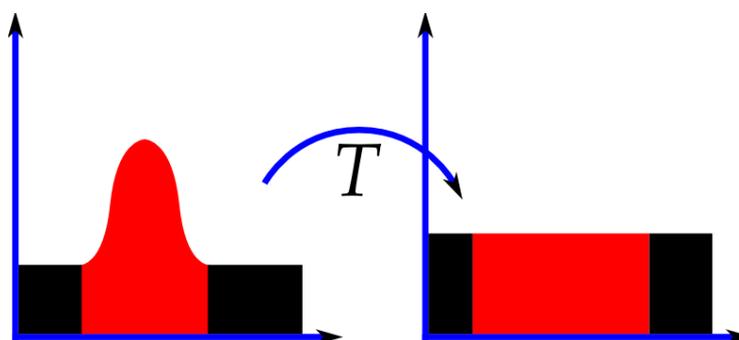


Figure 3.4 Histogram equalization before and after

The method is useful in images with backgrounds and foregrounds that are both bright or both dark. In particular, the method can lead to better views and detail in photographs that are over or under-exposed. A key advantage of the method is that it is a fairly straightforward technique and an invertible operator. So in theory, if the histogram equalization function is known, then the original histogram can be recovered. The calculation is not computationally intensive. A disadvantage of the

method is that it is indiscriminate. It may increase the contrast of background noise while decreasing the usable signal.

3.1.6 Colour Image Processing

Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of multidimensional systems.

The RGB colour model is an additive colour model in which red, green and blue light are added together in various ways to reproduce a broad array of colours. The name of the model comes from the initials of the three additive primary colours, red, green and blue.

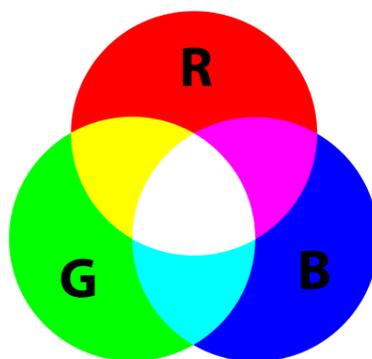


Figure 3.5 Views of RGB colour mixing

Figure 3.5 shows RGB mixing where adding red to green yields yellow; adding red to blue yields magenta; adding green to blue yields cyan; adding all three primary colours together yields white (Charles A, 2003).

Digital image processing mainly focusses on the RGB components analyze.

- (1) RGB components separating.
- (2) RGB converting to Binary
- (3) RGB converting to Grey level
- (4) Histogram equalization
- (5) Edge detection
- (6) Enhancement including brightness, contrast and RGB components.

All the functions mentioned above can be implemented by the GUI programme.

3.1.7 8-Connected Pixel Connectivity

In image processing and image recognition, pixel connectivity is the way in which pixels in 2-dimensional (or voxels in 3-dimensional) images relate to their neighbors.

8-connected pixels are neighbors to every pixel that touches one of their edges or corners. These pixels are connected horizontally, vertically, and diagonally. In 8-connected pixels mode, each pixel with coordinates $(x \pm 1, y \pm 1)$ or $(x \pm 1, y \mp 1)$ is connected to the pixel at (x, y) .

Figure 3.6 shows the 8-connected block model, the pixel marked in red has 8 neighbor pixels for checking if the red pixel is any part of the connected regions.

This is a fast and very simple method to implement and understand. It is based on graph traversal methods in graph theory. In short, once the first pixel of a connected component is found, all the connected pixels of that connected component are labeled before going onto the next pixel in the image. This

algorithm is part of Vincent and Soille's watershed segmentation algorithm (Vincent, Luc et al, 1991), other implementations also exist (Abubaker, Aet al, 2007).

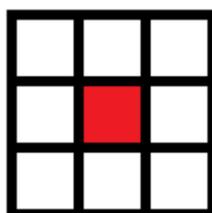


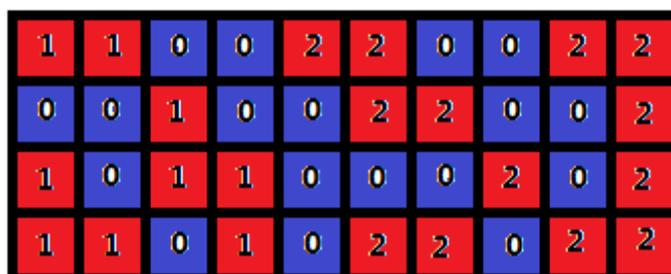
Figure 3.6 8-connected model for connectivity detection.

It is assumed that the input image is a binary image, with pixels being either background or foreground and that the connected components in the foreground pixels are desired.

Figure 3.7 shows the two-pass algorithm in graphs.

1	1	0	0	2	2	0	0	3	3
0	0	1	0	0	2	2	0	0	3
4	0	1	1	0	0	0	2	0	3
4	1	0	1	0	5	2	0	2	2

(a) First pass labeling



(b) Second pass labeling

Figure 3.7 Example of two-pass algorithm in image pixels.

As it shown in Figure 3.7 (a), red pixels are foreground and blue pixels are background labeled as 0. The first pass uniquely labels the non-zero pixels which have no neighbor pixels from left to right by row and from top to bottom by column. If the pixel has any neighbor pixels, label the current pixel with the smallest label in the neighbor pixels.

After fully processed by the first pass, the second pass shows in figure 3.7(b) is applied on the result in order to re-label the pixels with the lowest equivalent label.

The maximum label is the number of objects and the number of red pixels is total area of the object in the image. The results can be further used in object detection, cell counting and edge detection programme.

3.2 GUI Programme Implementation

This section presents the GUI programme implementation of the image processing function in this thesis.

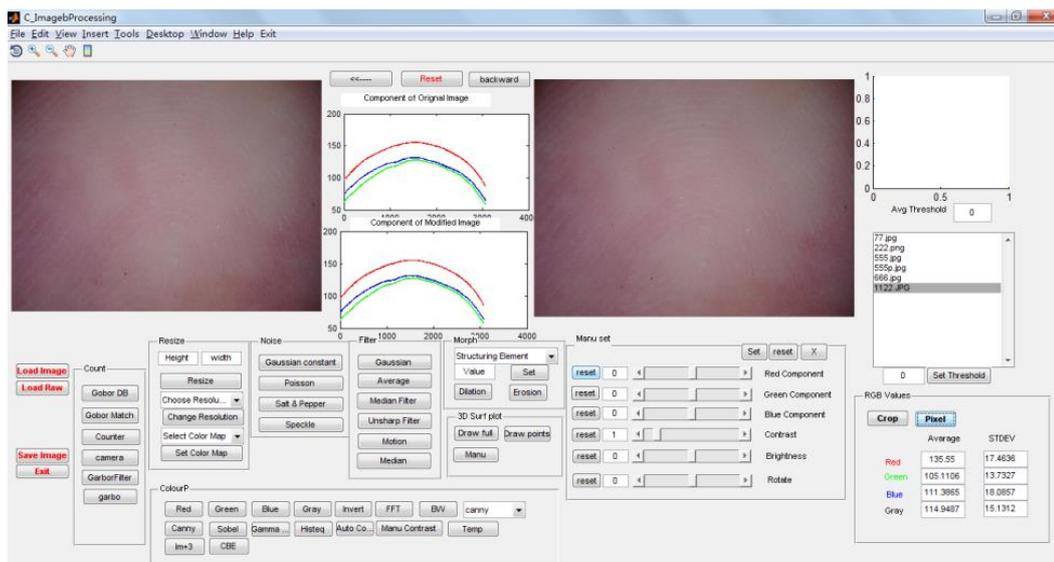


Figure 3.8 Views of the main GUI window of the programme

Figure 3.8 shows the colour images processing using GUI programme. On top-left there is the axes for displaying the original image, the one on the right is the processed image; two histogram axes in the middle shows the RGB components of the images with red, green and blue colour, if it is grey image or binary image, the colour will automatically change into black. On the top right, there is an axis for displaying the average grey thresh level of the images. Additionally, the list box on the right side will display the images which the user chosen, and easy to open those images just by clicking the filename in the list box. The bottom- right stands a table for calculating the RGB component of the whole image, or any customized area of the image by click crop button to select the interest area. All the processing functions were programmed as button clicking function. There is no need for customers to learn any programming knowledge, just by clicking the button with relevant processing tools' name on it.

Figure 3.9 shows the raw images from Fingerprint sensor can be loaded and processed by the GUI program.

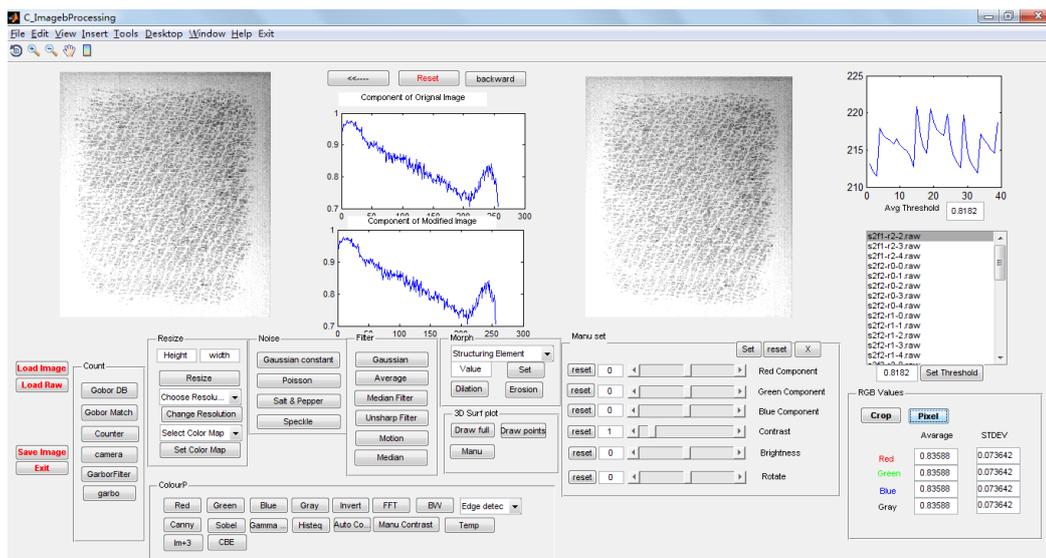


Figure 3.9 Raw images from Fingerprint sensor loading and processing by using GUI

The raw images are loaded in an array, then rebuild as matrix for display as an image by using GUI programme. After that, the raw images can be processed like the digital images. As it shown in Figure 3.9, the ‘average threshold’ chart on the top-right shows the average grey level of all input raw images, the line chart shows the hydration differences among the images. By clicking the ‘set threshold’ button, GUI set the global grey level by user input or by calculating itself.

This section presents the GUI image processing functions.

Figure 3.10 shows the results after applying filters on the image.



(a) Original

(b) Gaussian



(c) Average

(d) Motion

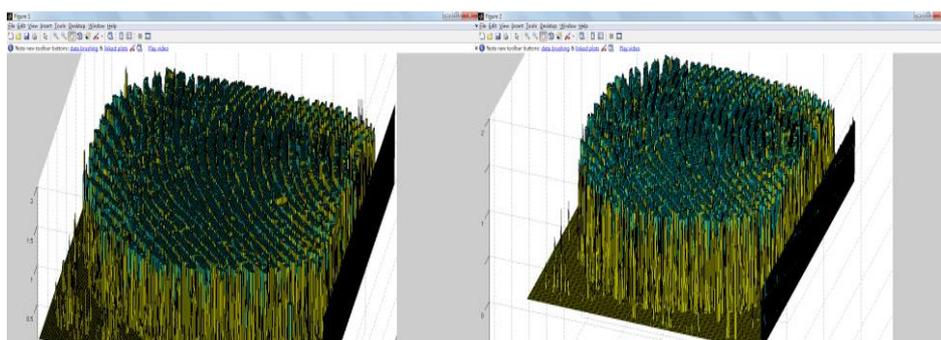


(e) Median

(f) Unsharp

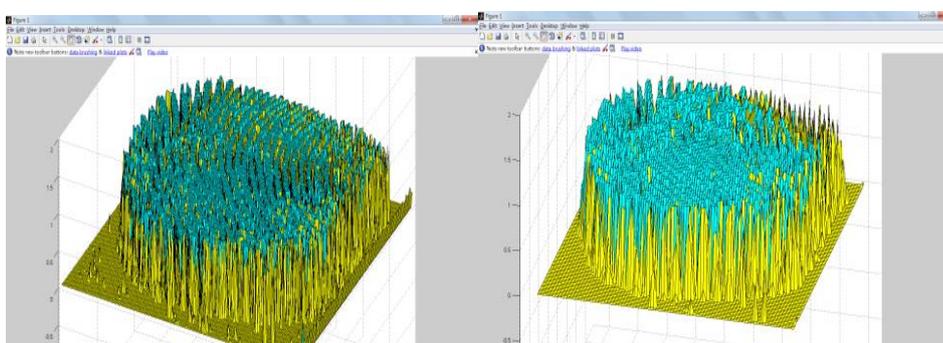
Figure 3.10 Filters performance on digital images.

Figure 3.10 shows the performance of the six filters. All the filters except the unsharp filter focused on smoothing the images in order to remove redundancy.



(a)1.0

(b)0.8



(c) 0.4

(d)0.2

Figure 3.11 3-D profiling of different resolutions of the image

Figure 3.11 shows the 3-D plot function using different resolutions for plotting a fingerprint image. The results show the different resolution of the input image, the plots show less bars but faster, the bad thing is the image lost more information of the fingerprint.

Figure 3.12 shows the colour image processing tools in the GUI. Figure 3.13 shows the results of GUI colour image processing tools including RGB components separate, grey and binary convert and edge detection.

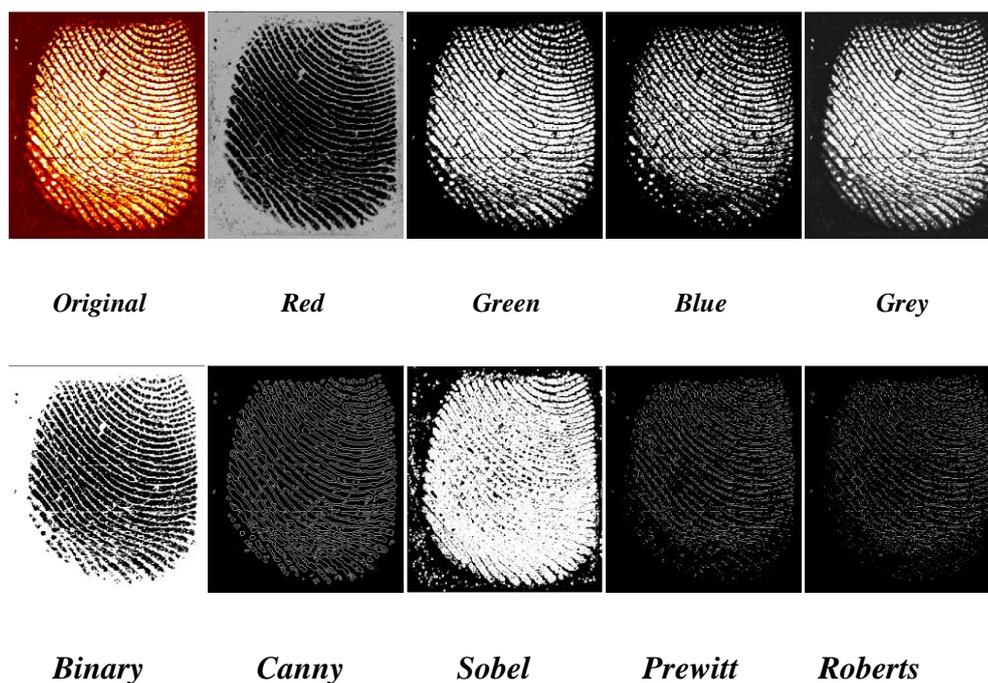
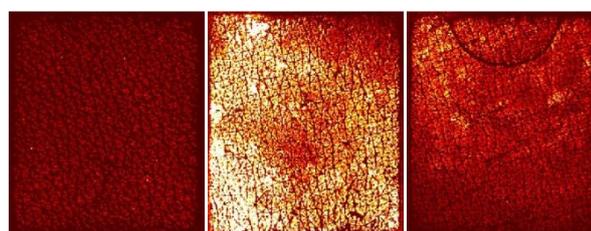


Figure 3.12 Results of colour image processing tools in the GUI



(a) Original images



(b) Processed images

Figure 3.13 Sample view of the extraction of the skin image from fingerprint sensor.

Skin image processing using image brightness equalization extract skin texture for further investigating the skin characteristics.



(a) Original digital image

(b) Processed digital image

Figure 3.14 Views of the extraction of the skin image from digital images.

Figure 3.14 shows the GUI feature extraction function for digital skin image. The result shows the hair extraction function on hairy skin. The extraction using histogram equalization to ignore the brightness and scope shadow, with the combination of filters and Colour image Brightness Equalization (CBE button) functions to extract the hair from skin images.

There is a function for SC cell counting in the GUI programme, which shows SC cells in Figure 3.15.

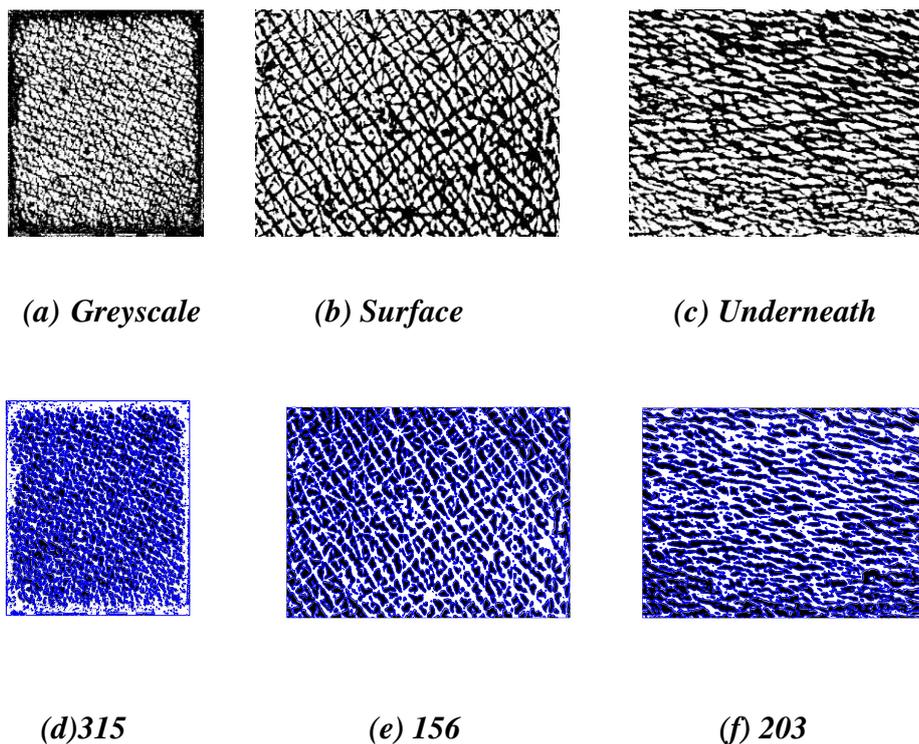
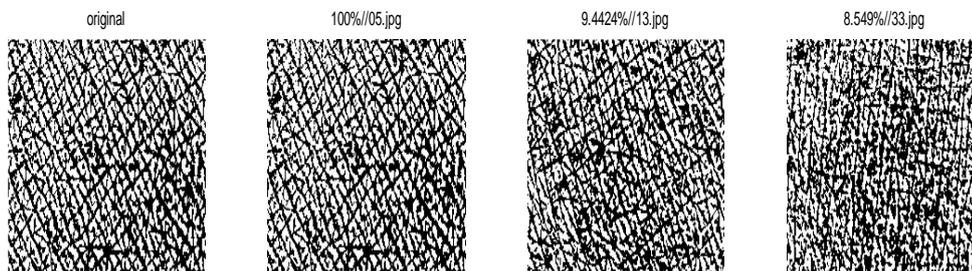


Figure 3.15 Processed skin images for SC cells counting function

The quality of the skin image affects the result that the clearer of the skin surface, the clearer texture it can be processed. Figure 3.15 shows three processed skin images using SC cell counting programme. The results of three images are 315, 156 and 203. In Figure 3.15(e), the texture is clearer than (d) and (f) that the SC cells has 15~17 rows and 8~10 columns more or less. It shows a matched result as well as (d) and (f). The programme convert skin images into greyscale images and further processed into binary images. The SC cells are separated by skin texture which can be marked by 8-connected object detecting method. To identify the cells, each pixel needs to be compared with 8-neighbor pixels until skin texture was detected. Then calculate the closed area numbers which is equal to the number of SC cells.

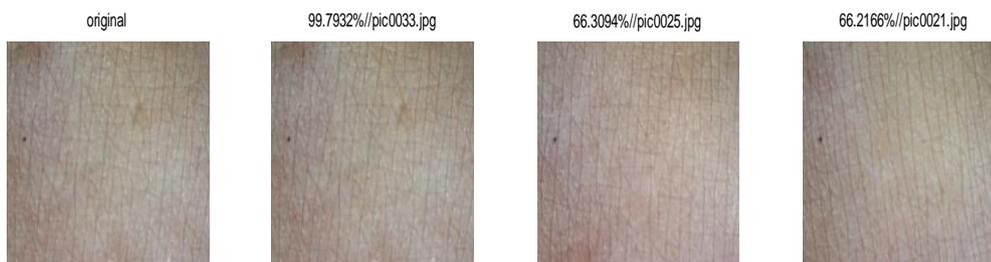
Figure 3.16 shows the Gabor feature detection and matching with the images in database.



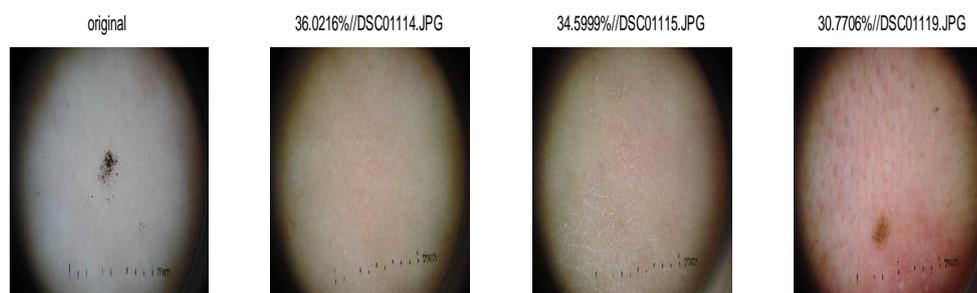
(a) Processed binary images



(b) Underneath skin images with different hydration level



(c) Surface skin images with different hydration level

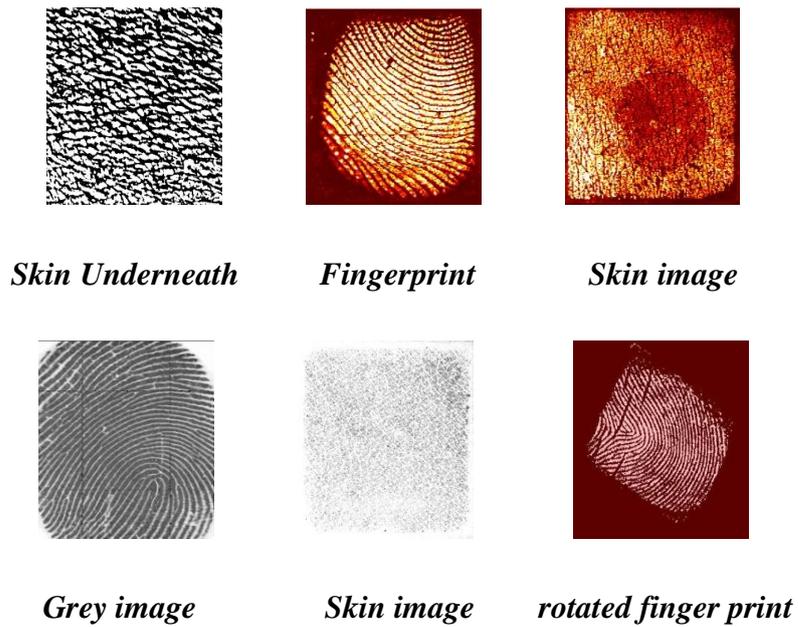


(d) Skin images with scope shadow

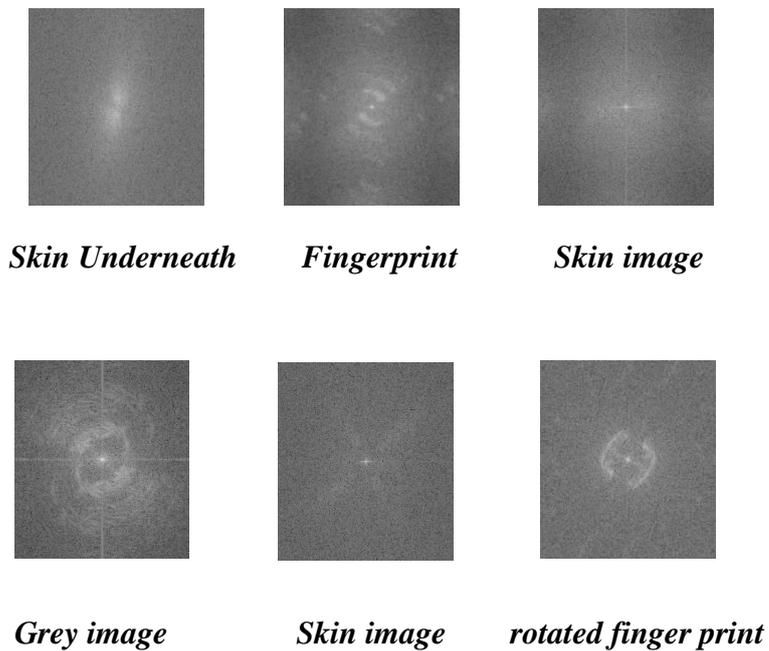
Figure 3.16 Gabor feature matching.

Figure 3.16 shows the matched results in the database. First, by setting 200 images in the database of the skin, face, and other none related images, using Gabor feature detecting program. Then match the input image with the database. The result shows the matched results where it matches itself at nearly 100% rate, 60~70% with different hydration level. It has 30% match rate with scope shadow.

The Gabor feature extraction cuts off image into small pieces, such as 4x4, 8x8 and 16x16. The smaller it is, the better result is determined, but need more time to generate the Gabor features. Then the programme compares the features with the images in the database to calculate the matching rate.



(a).Original images



(b) FFT images



Skin Underneath Fingerprint Skin image



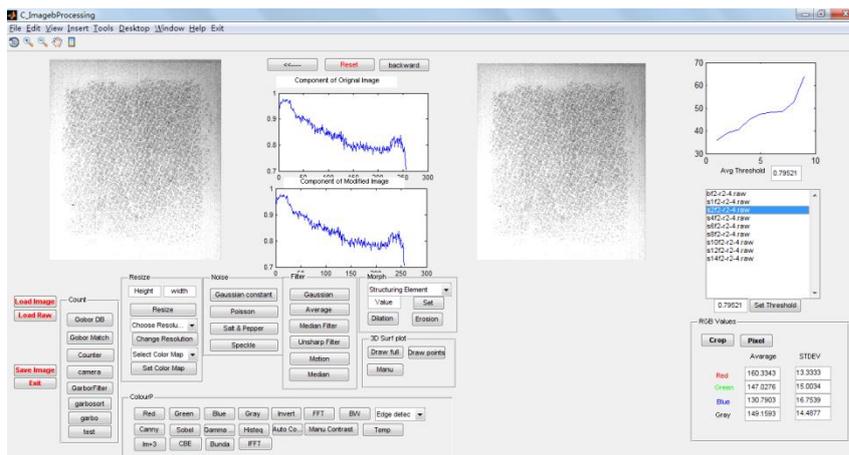
Grey image Skin image rotated fingerprint

(c).IFFT images

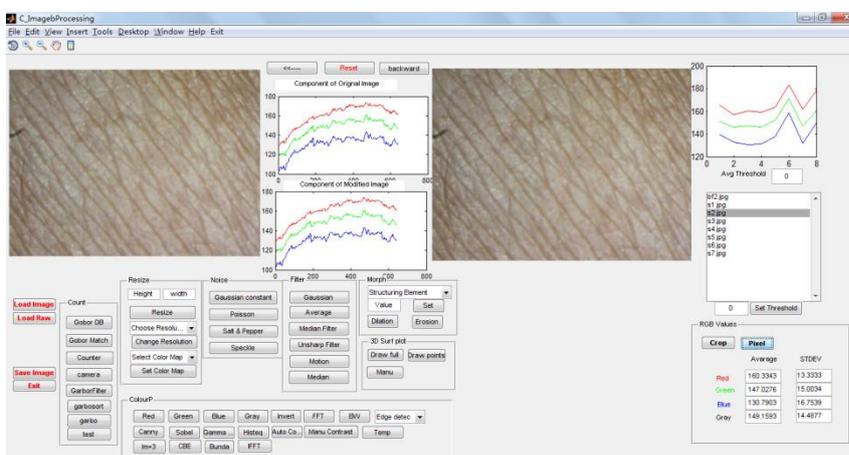
Figure 3.17 Views of the FFT and IFFT of skin images.

Figure 3.17 shows the fast Fourier transform (FFT) and inverse fast Fourier transform (IFFT). In figure 3.17(b), it shows the shifted 2-D FFT results which move the zero-frequency to the center of the array by swapping top-left and bottom-right blocks as well as the top-right and bottom-left block. The components which are more far away from the center, the more details they have. The IFFT results show that the images lost detail after IFFT.

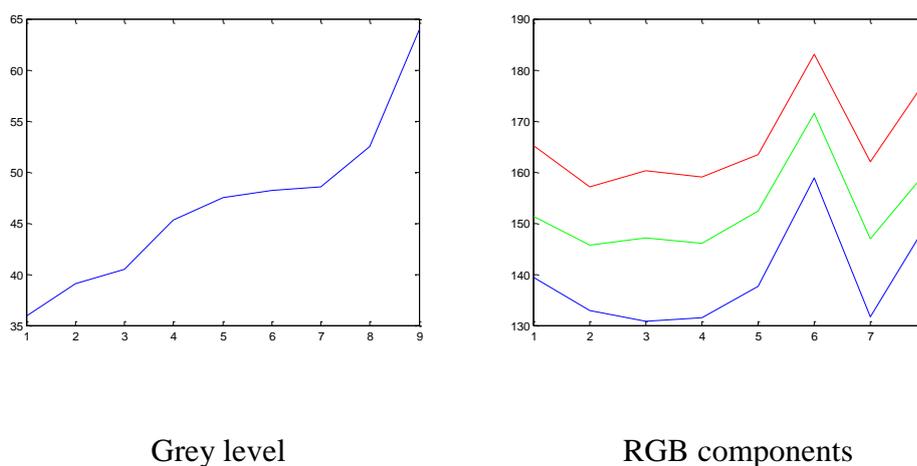
Figure 3.18 shows some analysis results of digital images and raw images.



(a) Greyscale Raw images processing



(b) Digital colour images processing



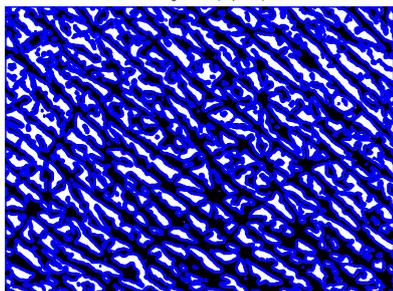
(c) *Greyscales of raw images and RGB components of digital images*

Figure 3.18 Results of skin images using GUI programme.

Figure 3.18 shows the results of GUI programme. The images are from baseline data to the last strip, 8 images in total. The charts in Figure 3.18(c) show the average greyscales level and RGB components of all the input images. It clearly shows the greyscale values increased steadily as the tape strips increased. The RGB values also increased but with huge fluctuations, this is likely to the fluctuation of LED lighting during the measurements. As it presented in chapter 2, the Proscope HR has independent LED lighting source on the camera.

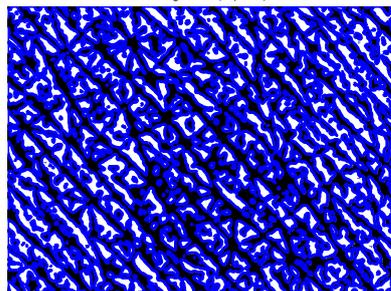
By using texture extraction function to calculate the SC cell area and skin texture area from baseline to S7:

cells : 233//Average area (in pixels):652.5751



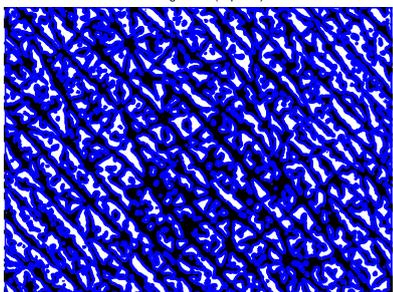
(a)Baseline

cells : 278//Average area (in pixels):549.9137



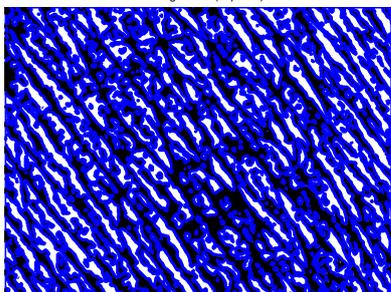
(b)S1

cells : 289//Average area (in pixels):524.5363



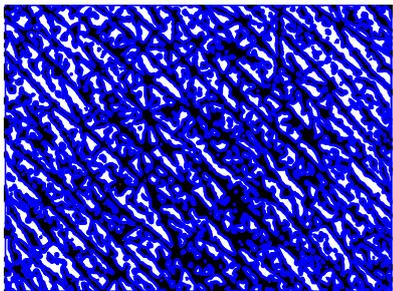
(c)S2

cells : 272//Average area (in pixels):560.4485



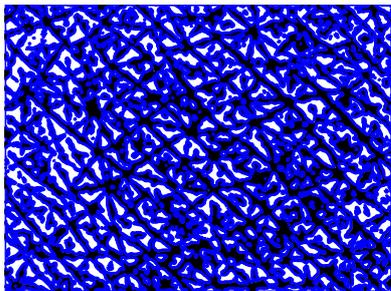
(d)S3

cells : 322//Average area (in pixels):472.0093



(e)S4

cells : 343//Average area (in pixels):442.1224



(f)S5

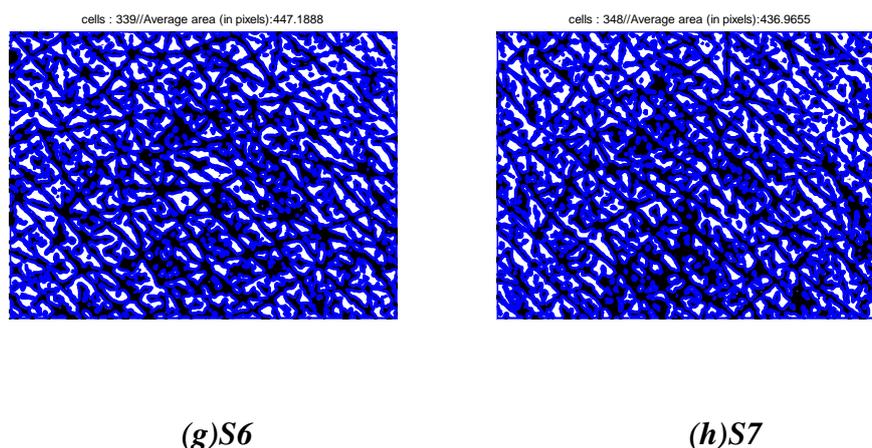
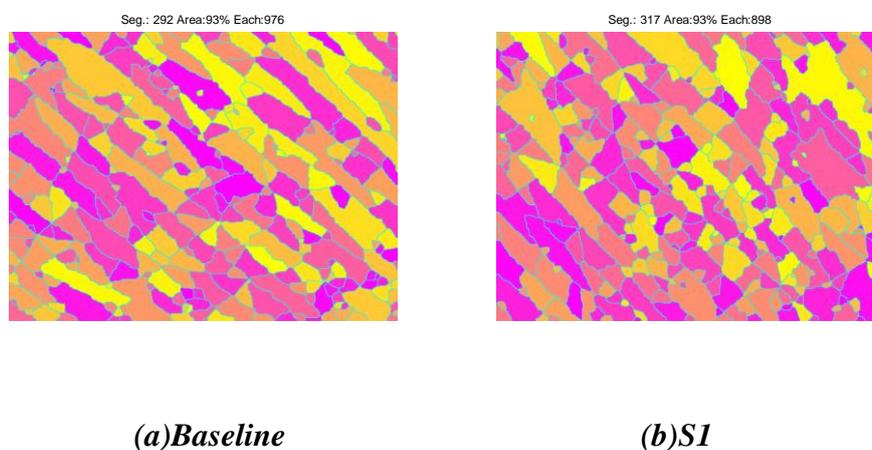


Figure 3.19 Result of SC counting programme for cells and skin texture area calculation.

Figure 3.19 shows the result of SC counting programme for tape stripping images from baseline to S7. The number of SC cells and the average area of the cells are shown on the top of the images. The SC cells: 233, 278,289, 272, 322, 343, 339, 348 and 433.

The skin texture areas: 652,549, 525,560,472,442,447 and 436 in pixels. The SC cells increased and the skin texture area decreased by strips. In other word, the skin texture gets sharp and thin.

Figure 3.20 shows the Watershed algorithm (see section 3.1.7, 8-connected pixel connectivity) for segment detection (Vincent, Luc et al, 1991).



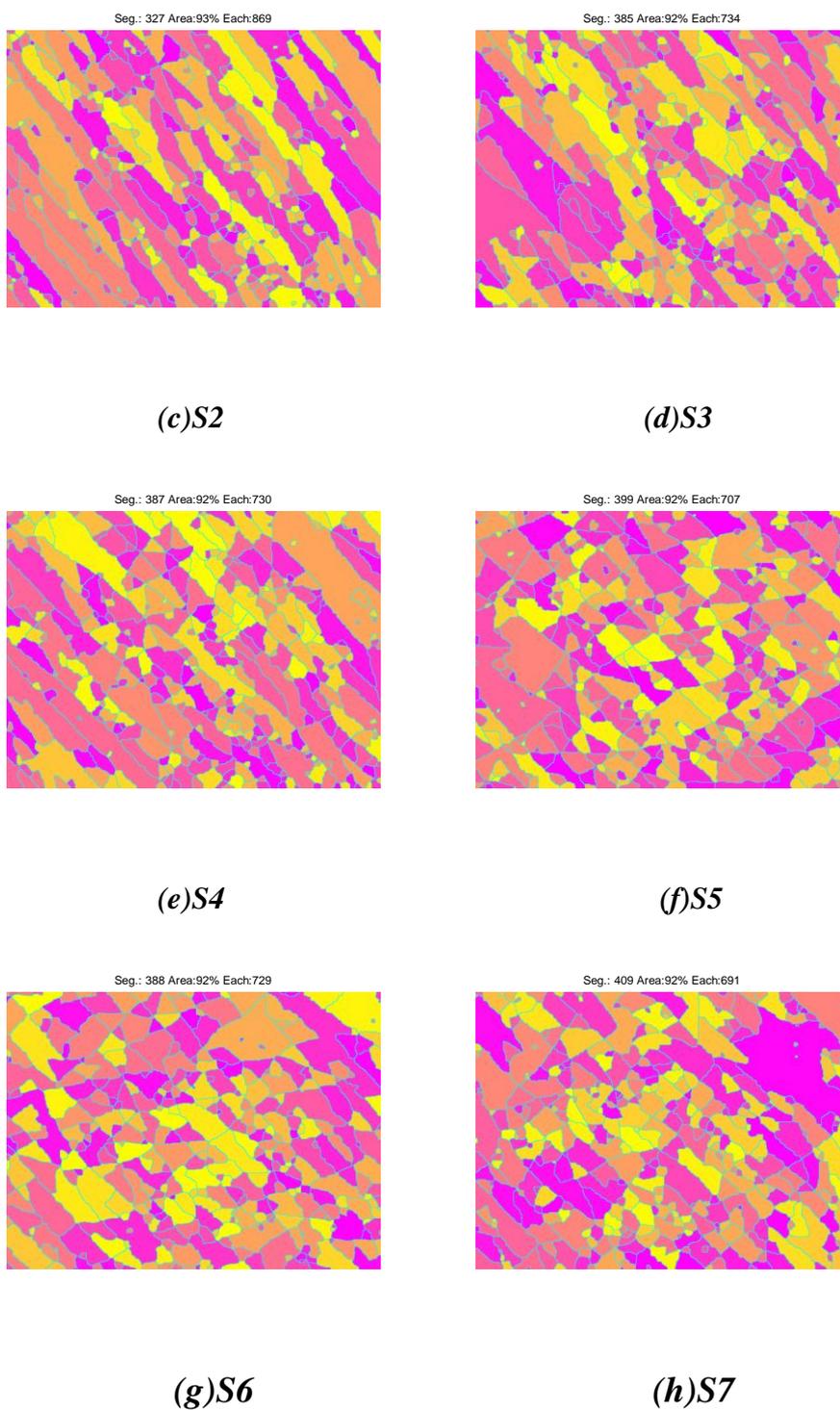


Figure 3.20 Result of Watershed algorithm for SC cell segment counting in tape stripping images from baseline to S7.

Figure 3.20 shows the Watershed algorithm using for SC segment counting. It detects and marks the SC cells by independent labels and colours based on 8-

connected detecting method. The cells are random coloured for better recognizing. The SC cells increase and the area decrease.

3.3 Summary

In summary, an effective and comprehensive skin image processing GUI based MATLAB programme has been developed. It can be used for analyzing both digital colour skin images and greyscale capacitive skin images. Apart from conventional digital image processing functions, such as filters, edge detections, morphology, 3D image profiling, histogram equalization etc., a set of new feature extraction functions have also been developed.

One of the useful feature extractions is to identify hairs in skin images, which allows us to analyze the skin and the hairs separately.

The results of Gabor feature matching shows that Gabor filter can be used for extracting texture features for skin images, despite the high degree of similarity among the images. This could be potentially used for skin disease or skin cancer detections. It also shows potential in human face recognition, facial expression recognition.

The FFT, cell counting, and Watershed algorithm also show interesting results for processing greyscale skin capacitive images of the human skin, which has very rich texture to analyze.

Chapter 4 Skin Image Processing Using OpenCV

MATLAB is a powerful tool in matrix operating. It has a lot of advantages described in Chapter 3. However, to implement real-time functions like camera real-time processing, digital image stitching, feature detecting etc., it is too slow. To find out another effective tool for enhancing the weak part or just replace them is very important in improving the performance of the GUI programme.

This Chapter presents the OpenCV library study which includes image processing functions can be implemented in MATLAB GUI. Microsoft visual studio also was used in dynamic link library (DLL) design for calling at GUI programme.

4.1 OpenCV

Open Source Computer Vision (OpenCV) is a library of programming functions, mainly aimed at real-time computer vision (Puli karli, 2012). Originally developed by intel's research centre in Nizhnv Novgorod (Russia), it was later supported by Willow Garage and is now maintained by Itseez. The library is cross-platform and free for use under the open source BSD license.

OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. The library can take advantage of multi-core processing. There are bindings in Python, Java and MATLAB/OCTAVE. OpenCV was designed for computational efficiency and with a strong focus on real-time applications.

OpenCV's application areas include:

- 2D and 3D feature toolkits
- Egomotion estimation
- Facial recognition system
- Gesture recognition
- Human–computer interaction (HCI)
- Mobile robotics
- Motion understanding
- Object identification
- Segmentation and recognition
- Stereopsis stereo vision: depth perception from 2 cameras
- Structure from motion (SFM)
- Motion tracking
- Augmented reality

All of the developments and algorithms in OpenCV are now developed in the C++ interface.

4.2 Microsoft Visual Studio

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio supports different programming languages and allows the code editor and debugger to support programming language. Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level

debugger and a machine-level debugger. Other built-in tools include a forms designer for building GUI applications, web designer, class designer and database schema designer.

The Microsoft visual C++ only used for implementing the OpenCV functions design for MATLAB GUI calling.

Dynamic link library

Dynamic-link library (or DLL) is Microsoft's implementation of the shared library concept in the Microsoft Windows and OS/2 operating systems. These libraries usually have the file extension DLL. Data files with the same file format as a DLL, but with different file extensions and possibly containing only resource sections, can be called resource DLLs. Examples of such DLLs include icon libraries, sometimes having the extension ICL, and font files, having the extensions FON and FOT.

It is not possible to directly execute a DLL since it requires an EXE for the operating system to load it through an entry point. Additionally, DLLs provide a mechanism for shared code and data, allowing a developer of shared code/data to upgrade functionality without requiring applications to be re-linked or re-compiled.

In this thesis, the use of DLL is to be linked into MATLAB GUI program in order to call OpenCV image processing function in GUI.

4.3 Image Processing Techniques

OpenCV has its power in real-time analysing and detecting functions.

4.3.1 Face Detection

Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images. Face detection also refers to the psychological process by which humans locate and attend to faces in a visual scene (Tutorial OpenCV). Face detection is very important for our research, as one of the applications is to automatically analyse the facial skin, which is of great interest to the cosmetic industry. This requires identifying the face in an image, and subsequently the locations of different skin areas.

Face detection can be regarded as a specific case of object-class detection. In object-class detection, the task is to find the locations and sizes of all objects in an image that belong to a given class. Examples include upper torsos, pedestrians, and cars.

Face-detection algorithms focus on the detection of frontal human faces. It is analogous to image detection in which the image of a person is matched bit by bit. Image matches with the image stored in the database. Any facial feature changes in the database will invalidate the matching process.

Traditional

Some facial recognition algorithms identify facial features by extracting landmarks, or features, from an image of the subject's face. For example, an algorithm may analyse the relative position, size, and/or shape of the eyes, nose, cheekbones, and jaw. These features are then used to search for other images with matching features (Bonsor, K, 2008). Other algorithms normalize a gallery of face images and then compress the face data, only saving the data in the image that is useful for face recognition. A probe image is then compared with the face data (Smith, Kelly, 2008). One of the earliest successful systems (R. Brunelli and T. Poggio, 1993) is based on template matching techniques (R. Brunelli, 2009)

applied to a set of salient facial features, providing a sort of compressed face representation.

Recognition algorithms can be divided into two main approaches, geometric, which looks at distinguishing features, or photometric, which is a statistical approach that distils an image into values and compares the values with templates to eliminate variances.

Popular recognition algorithms include principal component analysis using eigenfaces, linear discriminant analysis, elastic bunch graph matching using the Fisherface algorithm, the hidden Markov model, the multilinear subspace learning using tensor representation, and the neuronal motivated dynamic link matching (Devi K. and Dimple, 2015).

3-dimensional recognition

A newly emerging trend, claimed to achieve improved accuracies, is three-dimensional face recognition. This technique uses 3D sensors to capture information about the shape of a face. This information is then used to identify distinctive features on the surface of a face, such as the contour of the eye sockets, nose, and chin (Williams, Mark, 2008).

One advantage of 3D facial recognition is that it is not affected by changes in lighting like other techniques. It can also identify a face from a range of viewing angles, including a profile view (Williams, Mark, 2008). Three-dimensional data points from a face vastly improve the precision of facial recognition. 3D research is enhanced by the development of sophisticated sensors that do a better job of capturing 3D face imagery. The sensors work by projecting structured light onto the face. Up to a dozen or more of these image sensors can be placed on the same

CMOS chip—each sensor captures a different part of the spectrum (Crawford, Mark, 2011).

Even a perfect 3D matching technique could be sensitive to expressions. For that goal a group at the Technion applied tools from metric geometry to treat expressions as isometries (Kimmel, Ron, 2005). A company called Vision Access created a firm solution for 3D facial recognition. The company was later acquired by the biometric access company Bioscrypt Inc. which developed a version known as 3D Fast Pass.

A new method is to introduce a way to capture a 3D picture by using three tracking cameras that point at different angles; one camera will be pointing at the front of the subject, the second one to the side, and the third one at an angle. All these cameras will work together so it can track a subject's face in real time and be able to face detect and recognize.

Skin texture analysis

Another emerging trend uses the visual details of the skin, as captured in standard digital or scanned images. This technique, called skin texture analysis, turns the unique lines, patterns, and spots apparent in a person's skin into a mathematical space (Bonsor, K, 2008).

Tests have shown that with the addition of skin texture analysis, performance in recognizing faces can increase 20 to 25 percent (Bonsor, K, 2008).

Thermal cameras

A different form of taking input data for face recognition is by using thermal cameras, by this procedure the cameras will only detect the shape of the head and it will ignore the subject accessories such as glasses, hats, or makeup. A problem with using thermal pictures for face recognition is that the database for face

recognition is limited. Research of the use of thermal face recognition in real life, and operation sceneries, and at the same time build a new database of thermal facial images. The research uses low-sensitive, low-resolution ferro-electric electrics sensors that are capable of acquire long wave thermal infrared (LWIR). The results show that a fusion of LWIR and regular visual cameras has the greater results in outdoor probes. Indoor results show that visual has a 97.05% accuracy, while LWIR has 93.93%, and the Fusion has 98.40%, however on the outdoor proves visual has 67.06%, LWIR 83.03%, and fusion has 89.02%. The study used 240 subjects over the period of 10 weeks to create the new database. The data was collected on sunny, rainy, and cloudy days (Diego Socolinsky, and Andrea Selinger, 2004).

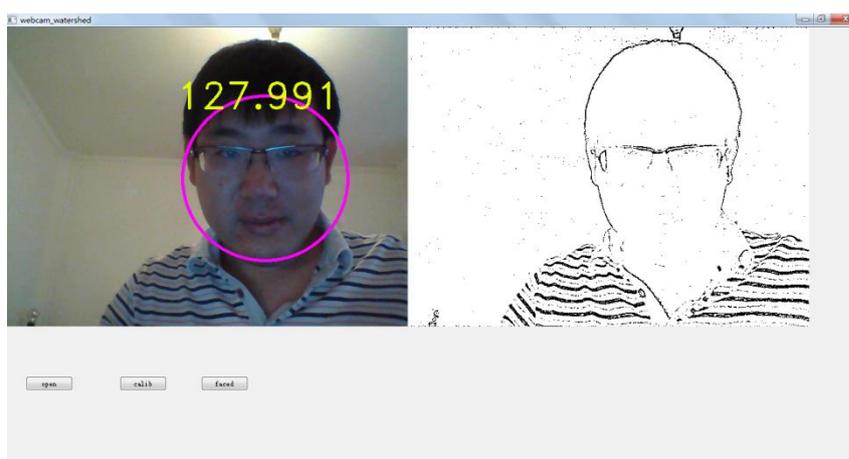


Figure 4.1 Face detection function result

Figure 4.1 shows the result of face detection function of the programme, the face is marked with a circle and the average grey level of the face areas is calculated and displayed in the window.

4.3.2 Image Stitching

Image stitching or photo stitching is the process of combining multiple photographic images with overlapping fields of view to produce a segmented panorama or high-resolution image. Commonly performed through the use of computer software, most approaches to image stitching require nearly exact overlaps between images and identical exposures to produce seamless results, (Mann, S, 1994) although some stitching algorithms actually benefit from differently exposed images by doing HDR (High Dynamic Range) imaging in regions of overlap (Mann S, 1993). Some digital cameras can stitch their photos internally. Image stitching is widely used in today's world in applications such as

- Medical Imaging.
- Multiple image super-resolution.
- Video Stitching (Breszcz, M, 2015)
- Object Insertion.

A general overview of image stitching algorithms firstly determines the appropriate mathematical model relating pixel coordinates in one image to pixel coordinates in another. Then estimate the correct alignments relating various pairs (or collections) of images. Next, combine direct pixel-to-pixel comparisons with gradient descent (and other optimization techniques) can be used to estimate these parameters. Distinctive features can be found in each image and then efficiently matched to rapidly establish correspondences between pairs of images. When multiple images exist in a panorama, techniques have been developed to compute a globally consistent set of alignments and to efficiently discover which images overlap one another.

The image stitching process can be divided into three main components - image registration, calibration and blending.

Registration

Image registration involves matching features (Szeliski, Richard, 2005) in a set of images or using direct alignment methods to search for image alignments that minimize the sum of absolute differences between overlapping pixels (Suen. S. et al, 2007). When using direct alignment methods one might first calibrate one's images to get better results. Additionally, users may input a rough model of the panorama to help the feature matching stage, so that - for example - only neighbouring images are searched for matching features. Since there are smaller group of features for matching, the result of the search is more accurate and execution of the comparison is faster.

To estimate a robust model from the data, a common method used is known as RANSAC. The name RANSAC is an abbreviation for "RANdom SAMple Consensus". It is an iterative method for robust parameter estimation to fit mathematical models from sets of observed data points which may contain outliers. The algorithm is non-deterministic in the sense that it produces a reasonable result only with a certain probability, with this probability increasing as more iterations are performed. It being a probabilistic method means that different results will be obtained for every time the algorithm is run. The RANSAC algorithm has found many applications in computer vision, including the simultaneous solving of the correspondence problem and the estimation of the fundamental matrix related to a pair of stereo cameras. The basic assumption of the method is that the data consists of "inliers", i.e., data whose distribution can be explained by some mathematical model, and "outliers" which are data that do not fit the model. Outliers are considered points which come from noise, erroneous measurements, or simply incorrect data. For the problem of homography estimation, RANSAC works by

trying to fit several models using some of the point pairs and then checking if the models were able to relate most of the points. The best model, i.e., the homography which produces the highest number of correct matches, is then chosen as the answer for the problem thus if the ratio of number of outliers to data points is very low the RANSAC outputs a decent model fitting the data.

Calibration

Image calibration aims to minimize differences between an ideal lens models and the camera-lens combination that was used, optical defects such as distortions, exposure differences between images, vignetting (Pablo d'Angelo, 2007), camera response and chromatic aberrations. If feature detection methods were used to register images and absolute positions of the features were recorded and saved, stitching software may use the data for geometric optimization of the images in addition to placing the images on the panosphere. Panotools and its various derivative programs use this method.

Blending

Image blending involves executing the adjustments figured out in the calibration stage, combined with remapping of the images to an output projection. Colours are adjusted between images to compensate for exposure differences. If applicable, high dynamic range merging is done along with motion compensation and deghosting. Images are blended together and seam line adjustment is done to minimize the visibility of seams between images.

The seam can be reduced by a simple gain adjustment. This compensation is basically minimizing intensity difference of overlapping pixels. Image blending algorithm allots more weight to pixels near the centre of the image. Gain compensated and multi band blended images compare the best. IJCV 2007. Straightening is another method to rectify the image. Matthew Brown and David

G. Lowe in their paper ‘Automatic Panoramic Image Stitching using Invariant Features’ describe methods of straightening which apply a global rotation such that vector u is vertical (in the rendering frame) which effectively removes the wavy effect from output panoramas

Even after gain compensation, some image edges are still visible due to a number of unmodelled effects, such as vignetting (intensity decreases towards the edge of the image), parallax effects due to unwanted motion of the optical centre, mis-registration errors due to mismodelling of the camera, radial distortion and so on. Due to these reasons they propose a blending strategy called multi band blending.

Figure 4.2 shows the stitched image from three parts of the fingerprint.

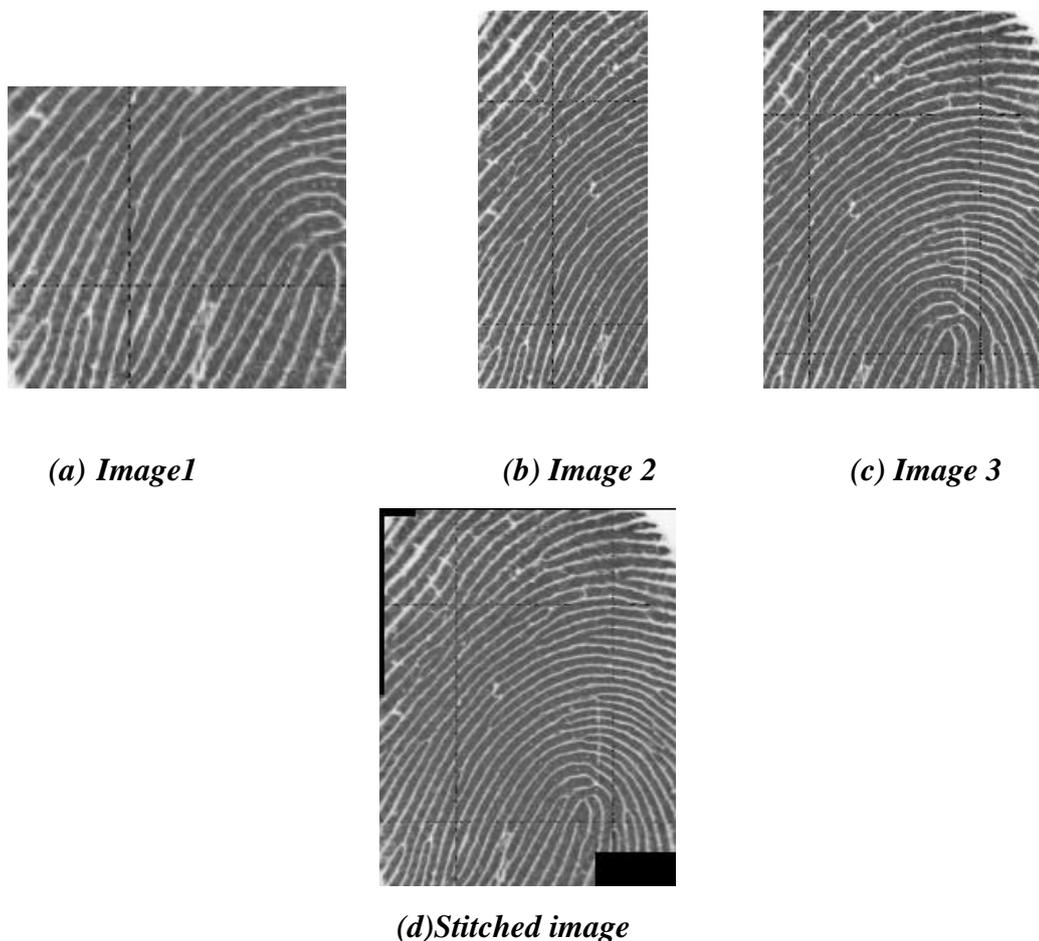


Figure 4.2 Image stitching result using fingerprint images.

4.3.3 Contours Detection

Contours detection is based on canny edge detection algorithm. The function first labels the boundaries of the objects in the image, then draw the contours of the objects using the coordinates of the edges which has been labelled by canny edge detection programme. Figure 4.3 shows an example of boundary.

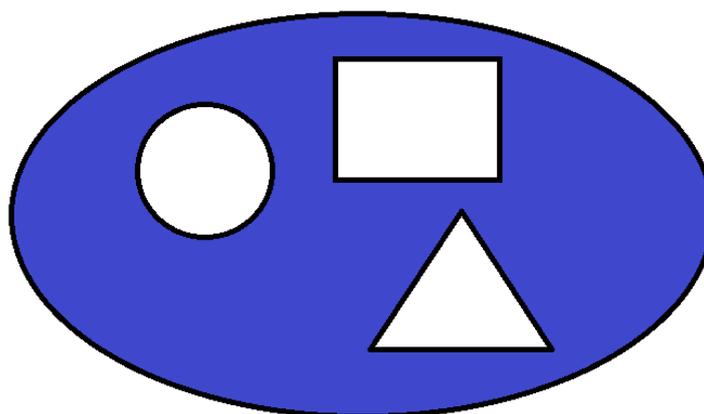
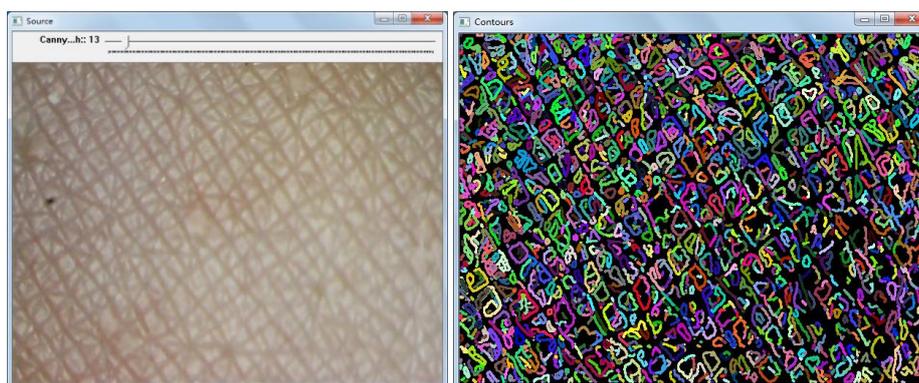


Figure 4.3 Example of boundaries of objects in image.

As it shown in figure 4.3, the blue part is the background pixels of the image with value 0, and black lines are the boundaries of the objects in the image.

Figure 4.4 shows the contours detection on skin images.



(a) Skin surface image

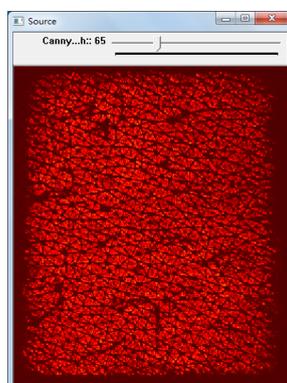
(b) Contours



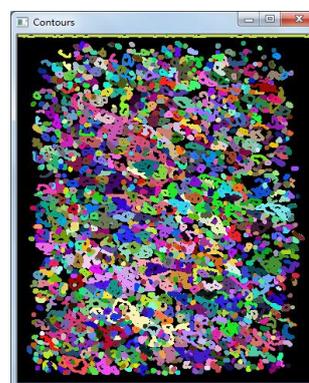
(c) Grey image



(d) Contours



(e) Fingerprint sensor image



(f) Contours

Figure 4.4 Results of contours detection programme on different skin images.

It is clear in Figure 4.4 that the contours detection algorithm is strong in extracting edges of the objects. It also can be used for object detection by calculating the number of the closed contours.

4.3.4 Real-time Video Processing

A video camera can feed or stream its images in real time to or through a computer to computer network. To implement the real time video processing, each frame needs to be processed and then displayed by the GUI programme.

In this section, for the real time processing part, four different edge detection algorithms are applied to the frames from the webcam and ProScope HR2 digital microscope for the real-time processing. There are Canny, Sobel, Laplace and morphology operators.

Laplace Operator

As we know, in human visual, the edge can be recognized by eye because the edge is the boundary of an object which has significant changes from the environment or background. In image processing, the changes are considered as the pixel intensity changes. A way to express changes is by using derivatives. A high change in gradient indicates a major change in the image.

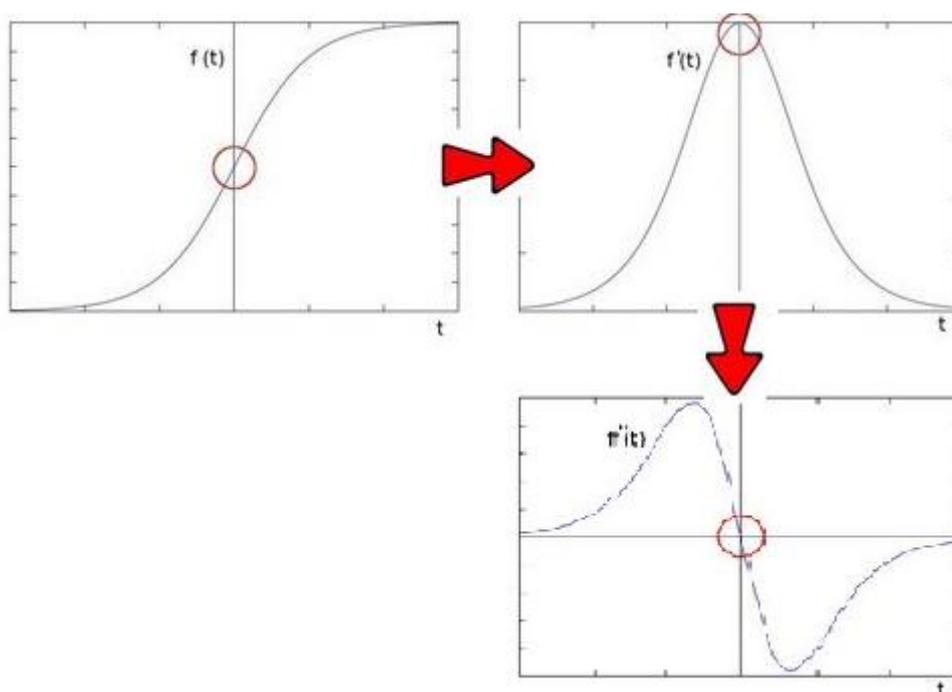


Figure 4.5 Example of twice derivative on image pixel intensity.

In figure 4.5, it is observed that an edge is characterized by a maximum by the first derivative of the intensity; second derivative is zero. It can also use this criterion to attempt to detect edges in an image. However, the zeros will not only appear in

edges they can actually appear in other meaningless locations. It can be solved by applying filtering where needed.

Since images are 2-D, the derivative needs to be applied in both dimensions. The Laplacian operator is defined by:

$$Laplace(f) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (4.1)$$

Morphological Transformation

Morphology operator includes two operations: erosion and dilation which were introduced in Chapter 3.

The morphological transformation includes five different operators: Opening, Closing, Morphological Gradient, Top Hat and Black Hat.

Opening uses erosion first on the image then further applies dilation on the result. It is commonly used for removing small object in the image.

Closing is the inverse way of opening.

The Morphological Gradient is the difference between dilation and erosion of the image using Morph=Dilation-Erosion.

In mathematical morphology and digital image processing, top-hat transform is an operation that extracts small elements and details from given images. There exist two types of top-hat transform: The white top-hat transform is defined as the difference between the input image and its opening by some structuring element; The black top-hat transform is defined dually as the difference between the closing and the input image. Top-hat transforms are used for various image

processing tasks, such as feature extraction, background equalization, image enhancement, and others.

Let $f : E \mapsto R$ be a greyscale image, mapping points from the Euclidean space or discrete grid E into the real line. Let $b(x)$ be a greyscale structuring element.

Then, the white top-hat transform of f is given by:

$$T_w(f) = f - f \circ b \quad (4.2)$$

where \circ denotes the opening operation.

The black top-hat transform of f (Tcheslavski, Gleb V, 2010) is given by:

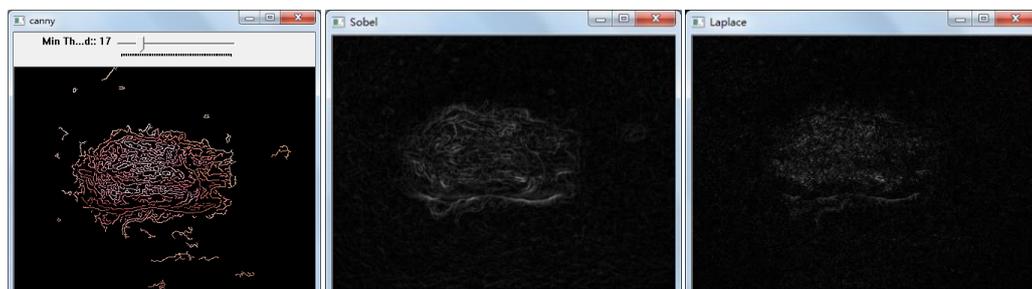
$$T_b(f) = f \bullet b - f \quad (4.3)$$

where \bullet denotes the closing operation.

Webcam processing results

This section presents some webcam processing results including edge detection, face detection and skin image processing.

Figure 4.6 shows the results of the four operators on scar image.



(a) *Canny*

(b) *Sobel*

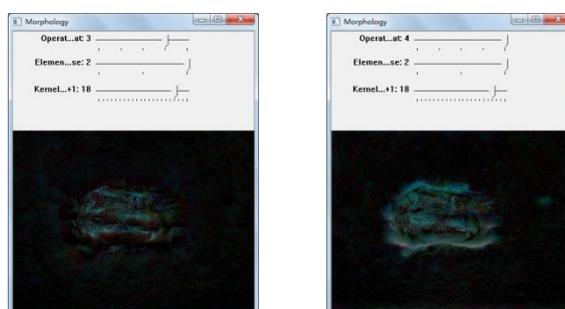
(c) *Laplace*



(d) Opening

(e) Closing

(f) Morphological Gradient

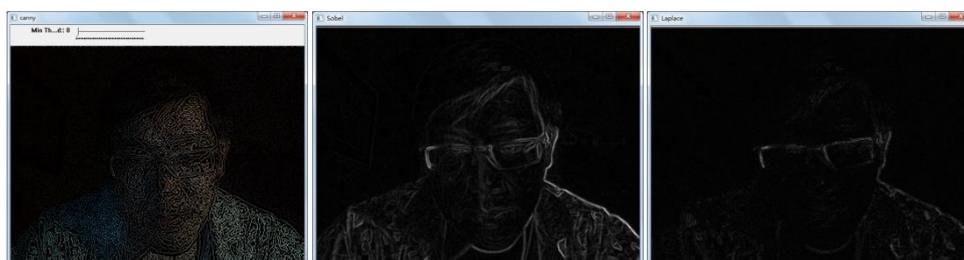


(g) Top hat

(h) Black hat

Figure 4.6 Results of canny, sobel, morphology and laplace operators on scar image.

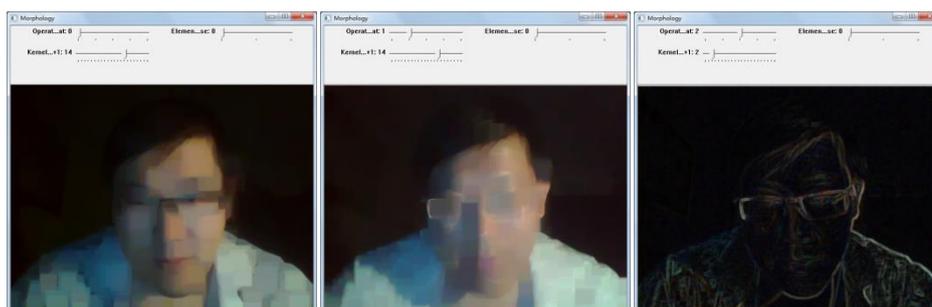
Figure 4.6 shows the results of scar image with image processing tools. It proved that more details can be easily and visually obtained by using these programme.



(a) Canny

(b) Sobel

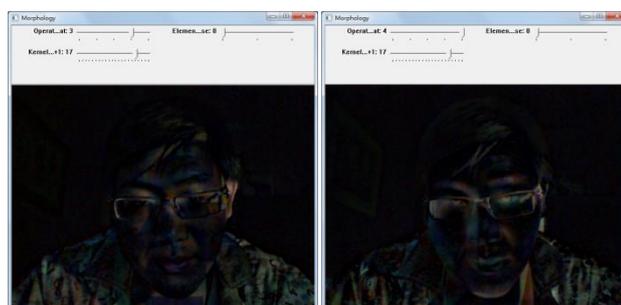
(c) Laplace



(d) Opening

(e) Closing

(f) Morphological Gradient



(g) Top hat

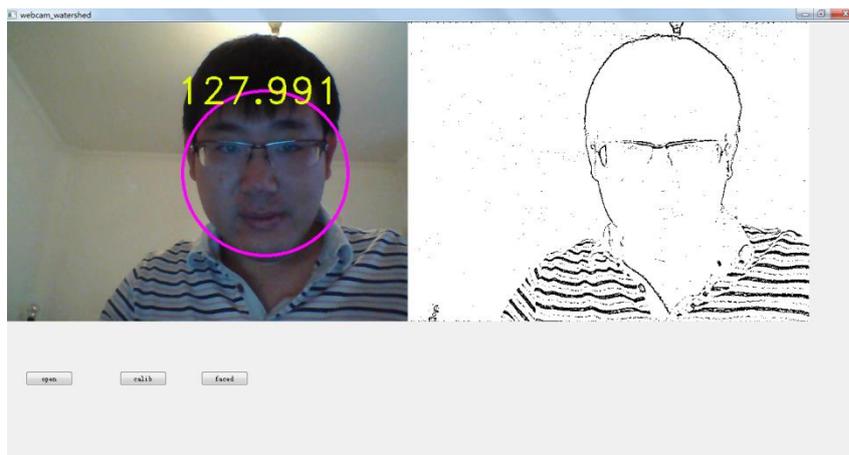
(h) Black hat

Figure 4.7 Results of the operators on webcam frames real-time processing programme.

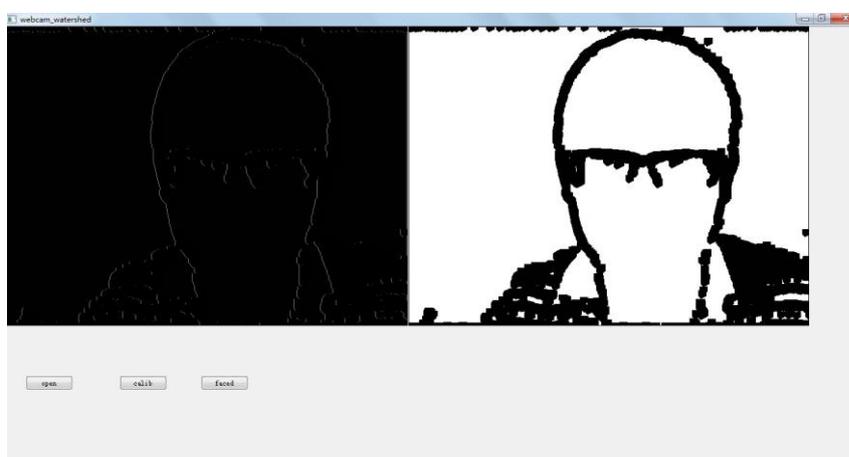
Figure 4.7 shows the results of human face image using webcam image processing tools.

Real-time processing

The webcam captures the image with a frequency depends on the quality of the webcam. Human eyes can recognize up to 60 FPS (frames per second). To implement the real-time processing function, the main point is to process the webcam frame by frame.



(a).Face detection and average grey level of the face



(b).Face contours separation

Figure 4.8 Webcam real-time processing results of face detection.

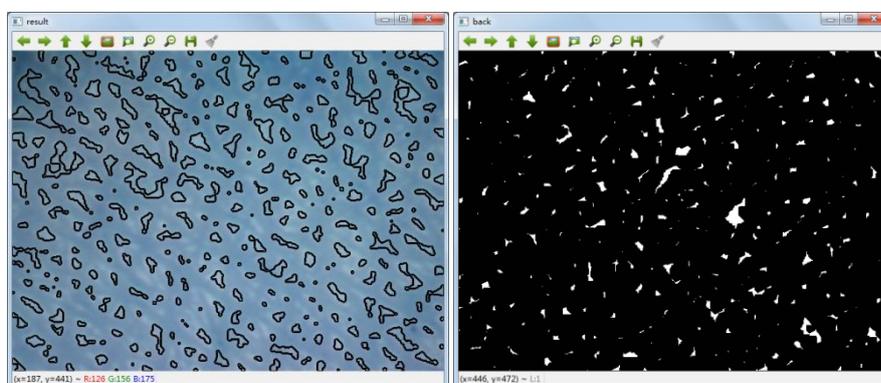
Figure 4.8 shows the webcam real-time face detection. Figure 4.8 (a) shows the grey level result of the human face in the webcam. Figure 4.8 (b) shows contours of the face which is used for face recognizing and grey level calculation.

ProScope HR2 results

The ProScope HR2 is a high resolution digital microscope which can be used for capturing the skin image as source images for the real-time processing part.



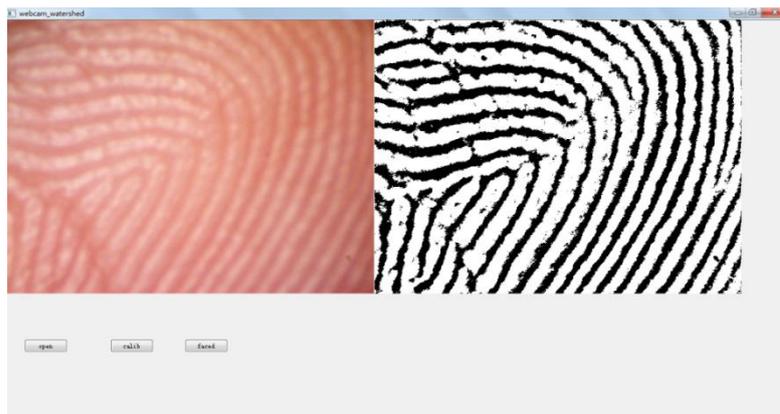
(a).Skin texture extraction



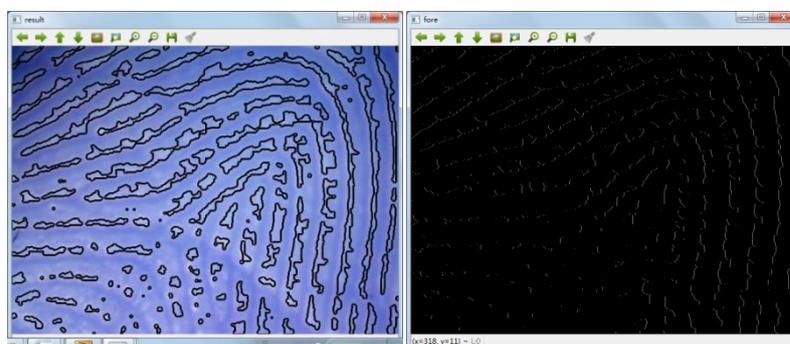
(b).SC cell contours using watershed

Figure 4.9 Webcam real-time processing results of skin using watershed.

Figure 4.9 (a) shows the skin image texture extraction. Figure 4.9 (b) shows the SC cell contours using the watershed function in order to locate the SC cells for further calculation.



(a).Finger print extraction



(b).Finger print contours using watershed

Figure 4.10 ProScope HR2 real-time processing results of finger print using watershed.

Figure 4.10 shows the results of finger print. Figure 4.10(a) shows the extraction of finger print and 4.10(b) shows the finger print texture contours using watershed.

4.4 Summary

In summary, a new C++ skin image processing programme based on OpenCV has been developed. Comparing with MATLAB, OpenCV is a royalty free open source image processing library, and very efficient for processing real time video images.

The OpenCV is interesting software. There are many examples in OpenCV, which can be implemented directly for better understanding those algorithms and procedure of every function without learning another programming language.

The DLL design has been more challenging due to the difference between M code (MATLAB) and C++ code. The function implemented worked well. Using DLL makes the real-time video processing faster than GUI loading it by its own. It also can be used for easy implement the face detection and image stitching.

Above all, the OpenCV library is very useful tool in image processing due to the distinct interface to other compile language. The real-time video programme is effective for detecting edges, extracting skin textures, ROI calculations, as well as face detection. Face detection opens a door for the future application in automatic facial skin analysis.

Chapter 5 Skin Characterization Measurements

Measurements of skin hydration of the Stratum Corneum (SC) and Trans-epidermal water loss (TEWL) are of great importance in skin characteristics research. One of the aims of this research is to develop new methodologies for such skin measurement using the three novel technologies developed by the research group: Opto-Thermal Transient Emission Radiometry (OTTER), AquaFlux and capacitive contact imaging based on the Fingerprint sensor. To achieve this aim, six types of experiments are presented in this chapter including skin over-hydration measurement, skin damage assessment, dermabrasion, and soap drying effect. This work has attracted a lot of interests, and as a result, most of the experiments are carried out through collaborations with other universities and companies.

It is important to point out that during the course of this project work, capacitive contact imaging has gone through from a prototype device based on a bare sensor (referred as Fingerprint Sensor), which is uncalibrated and generates only greyscale images, to a fully calibrated permittivity imaging system (referred as Epsilon), which generates full colour RGB images. Experiments in section 5.1 - 5.3 are done by using the uncalibrated Fingerprint Sensor, while experiments in sections 5.4 – 5.5 use calibrated Epsilon permittivity imaging system.

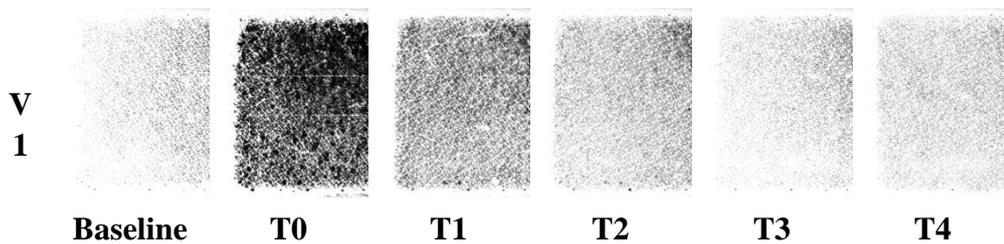
5.1 Skin Over-Hydration Measurement

This experiment is carried out through collaborations with the Continence and Skin Technology Group of UCL. The purposes of this experiment are: to investigate the

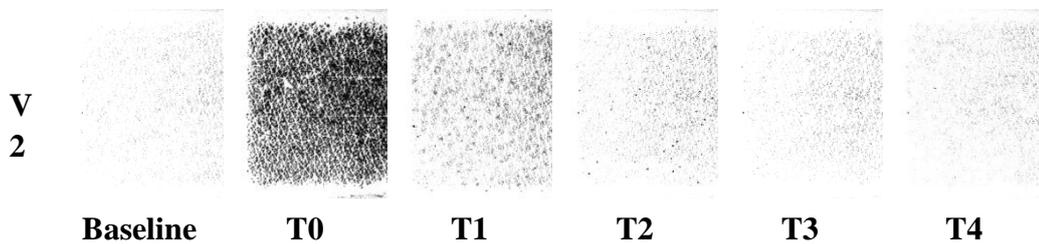
difference between over-hydrated skin and normal skin; to find out the correlations between hydration and TEWL results; to find out the characteristics of the skin during over-hydration.

The ambient temperature is 23.0°C and related humidity (RH) 49 %, the skin site on left volar forearm (F1) was chosen on two volunteers (V1 and V2, aged 20-30, one female V1) to perform an over-hydrated area for the measurement. To make the skin over-hydrated in experimental method, a tissue paper soaked with water was used for soaking skin site. F1 covered by a film for 20minutes while soaking. Hydration and TEWL were measured by Fingerprint sensor (3-times average) and AquaFlux, before soaking baseline, immediately after soaking and every 5 minutes up to 20 minutes, denoted as Baseline, T0, T1~T4.

Figure 5.1 shows the greyscale images from Fingerprint sensor of two volunteers.



(a) Images from Fingerprint sensor of V1 from baseline to T4



(b) Images from Fingerprint sensor of V2 from baseline to T4

Figure 5.1 Greyscale images from Fingerprint sensor of two volunteers from Baseline to T4

The images of both V1 and V2 get extremely dark on T0 and then recovered to grey on T1 rapidly, finally force down to the baseline level steady and slow from T2 to T4. All the images of V1 are darker than V2 in all time slots.

Figure 5.2 shows the Hydration and TEWL results.

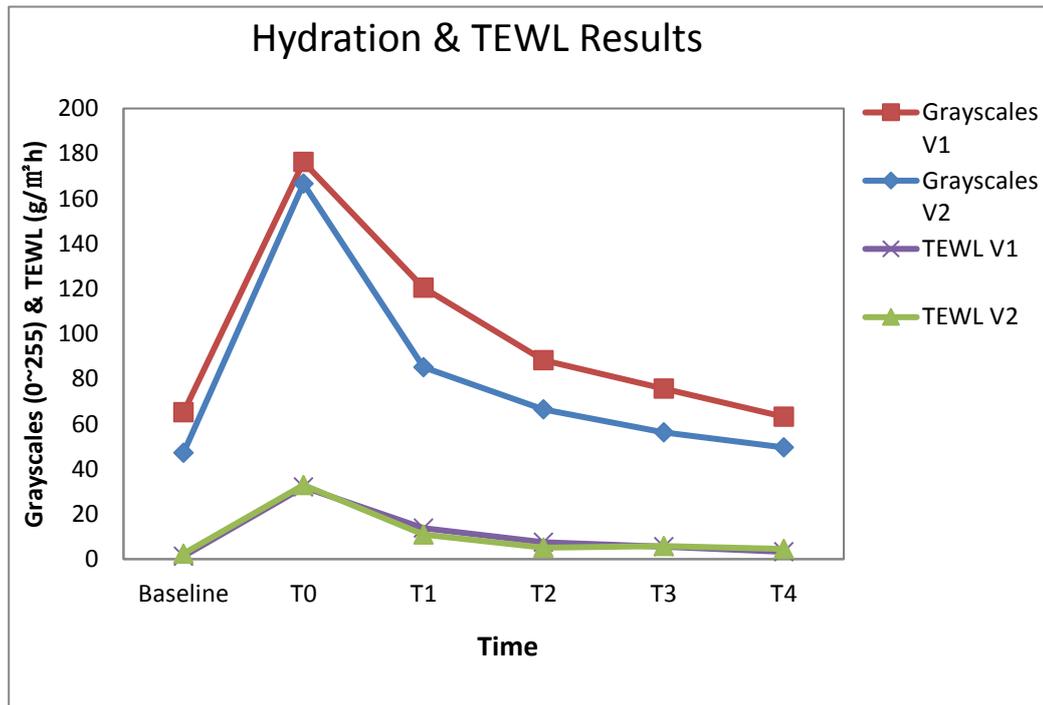


Figure 5.2 Hydration and TEWL results of two volunteers from Baseline to T4

The results clearly show the variation of hydration and TEWL. At T0, both TEWL and Hydration reached a cap value, and then fell down rapidly to a lower level at T1. Finally, the hydration and TEWL return to the baseline value from T2 to T4. Additionally, Hydration of V1 is higher than V2 at all time, as well as the TEWL.

Figure 5.3 shows the hydration and TEWL results of two volunteers.

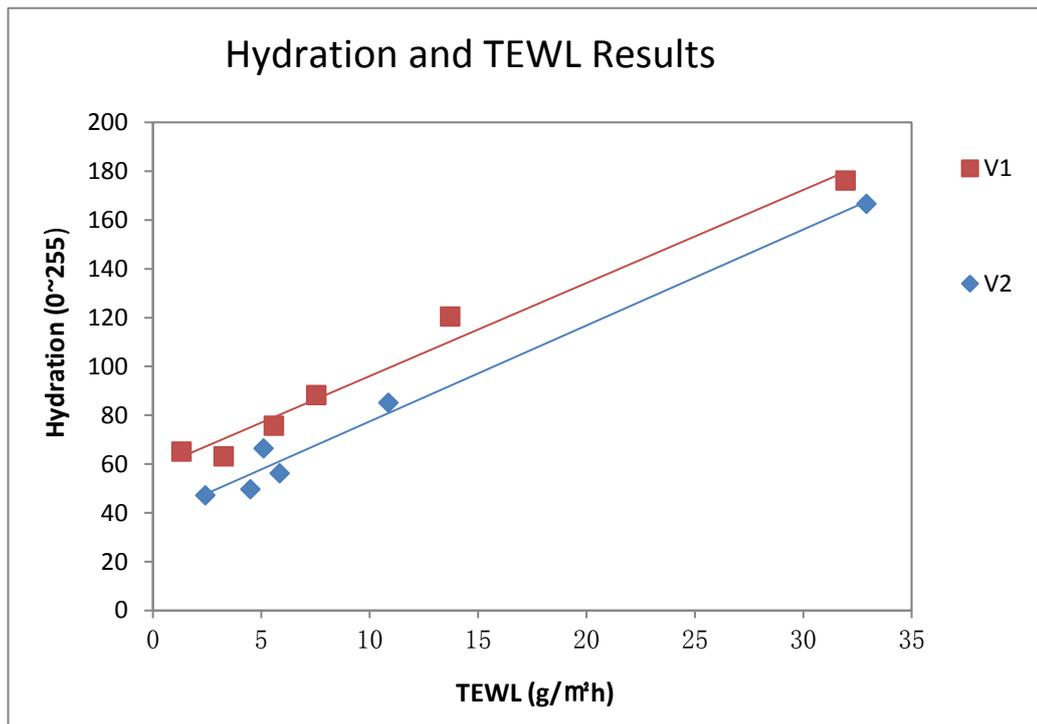


Figure 5.3 Hydration and TEWL results of two volunteers

The correlation shows positive of Hydration and TEWL results.

According to the hydration results in Figure 5.2, the hydration data shows a good match with the darkness of images in Figure 5.1: More black spots in the image, the higher hydration there is. In other word, V1 has higher hydration than V2 during the measurement. However, at T1, the hydration of V2 returns to a lower level than V1 which means V2 lost more water from T0 to T1. The TEWL of V2 at T0 higher than V1 but lower at T1, and the hydration shows an opposite relation. In this experiment, the over-hydrated skin keeps losing water to reach the normal level, rapidly at the beginning but slowly after 5 minutes. The speed depends on the skin recovering function: the more hydration, the more TEWL of the skin.

It also shows a nice correlation of Hydration and TEWL in Figure 5.3 that the hydration and TEWL are two important status of the skin.

Above all, the hydration status varies from people in different ages, sexuality and health status. Hydration and TEWL are two important measurements for skin characteristic studies. Fingerprint sensor produces the images of skin hydration visually in order to further understanding the hydration variation in experimental skin research. Also, AquaFlux is a novel instrument for TEWL measurement with high performance.

5.2 AquaFlux Occlusion Effect in Skin Hydration

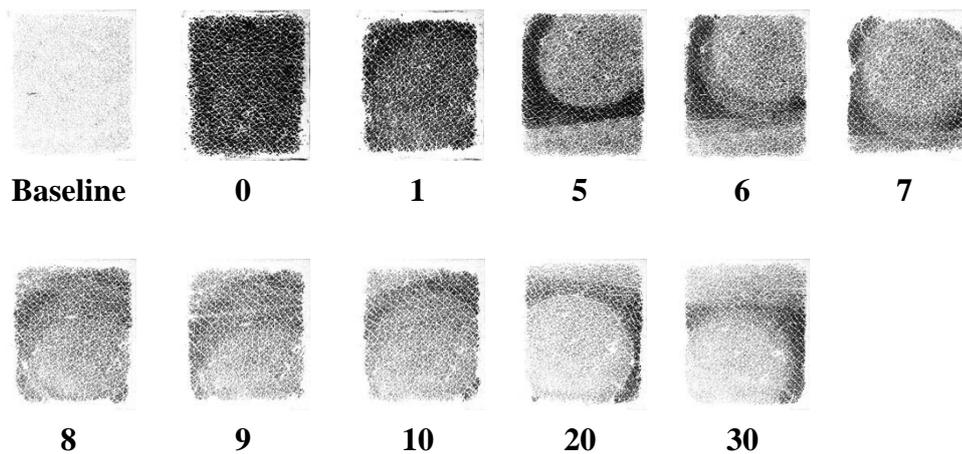
Measurement

During the above skin over-hydrated measurements, it was found that AquaFlux often left a ring mark on the skin site after measurements, which appeared as a dark ring in the consequent capacitive contact imaging measurements, as shown in Figure 5.4.

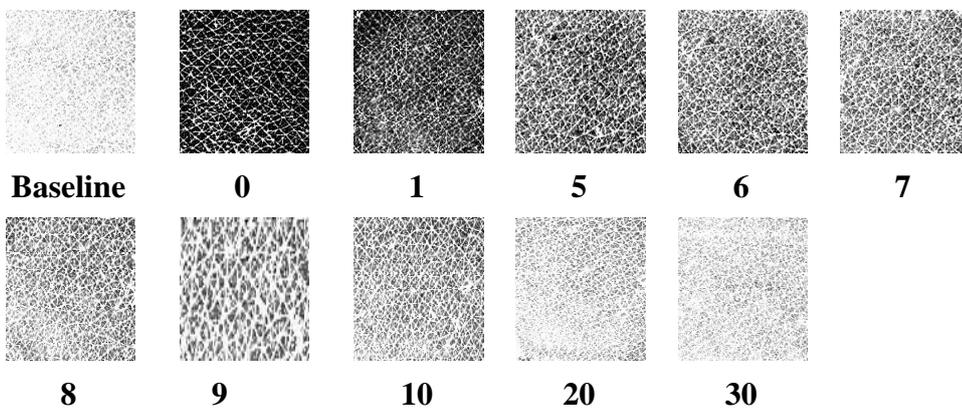
This is because the AquaFlux's cylindrical probe edge that is in contact with skin blocked the skin water loss, whilst skin areas within the cylindrical probe can lose water freely. Therefore, from the capacitive contact images, a dark ring appeared after the AquaFlux measurement. This mark will obviously affect the skin hydration results.

It is interesting to point out that this dark ring occlusion effect only happens to over-hydrated skin. For normal skin and dry skin, it was never observed.

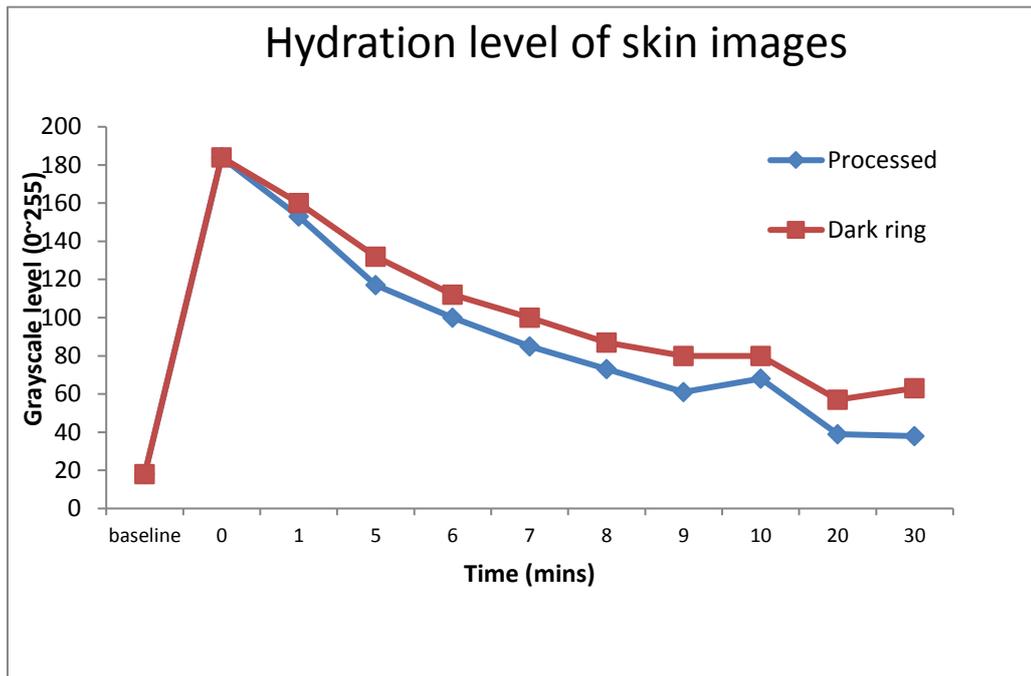
Figure 5.4 shows the dark ring in the skin hydration images.



(a) Skin hydration images with dark rings



(b) Processed skin images which ignore the dark rings



(c) Line chart of the comparison of skin images.

Figure 5.4 *The comparison of skin hydration images and chart with dark rings and processed images without dark rings.*

Figure 5.4 shows the comparison of the images with dark rings and the processed images without dark rings. 5.4(a) and (b) clearly show the dark rings in skin hydration images. 5.4(c) shows the comparison of the dark ring images and processed images. The dark ring slightly affects the data from 1 min to 10 min but significantly affect the data in 0 min and 30 min. In other word, the dark ring has affected skin hydration measurements which increase the skin hydration.

This demonstrated the importance of image processing in the capacitive contact imaging measurements. This also gives the capacitive contact imaging an advantage comparing to other skin hydration technologies, which are non-imaging based, and the results are just the average of entire probe contact area.

5.3 Skin damage assessment

As the outermost layer of the skin, the SC layer plays a pivotal role in protecting the body from the environment and limiting the amount of water lost from the epidermis.

Healthy, problem-free skin is even in colour, smooth in texture, well hydrated and appropriately sensitive to touch, pressure and temperature. When skin's natural barrier is disturbed, its protective function and healthy appearance are compromised. It loses moisture and elasticity; getting dry, rough and cracked, furthermore, it becomes increasingly sensitive to external influences (such as sun and temperature changes) and is particularly prone to infection. Infected skin can become inflamed as inflammatory immune cells move in to try and repair the damaged barrier and heal the infection. The skin recovers without scarring only if the SC layer gets damaged.

In this section, three experiments for skin damage assessment will be presented in order to figure out the functions and characteristics of SC layer.

5.3.1 Tape stripping measurement

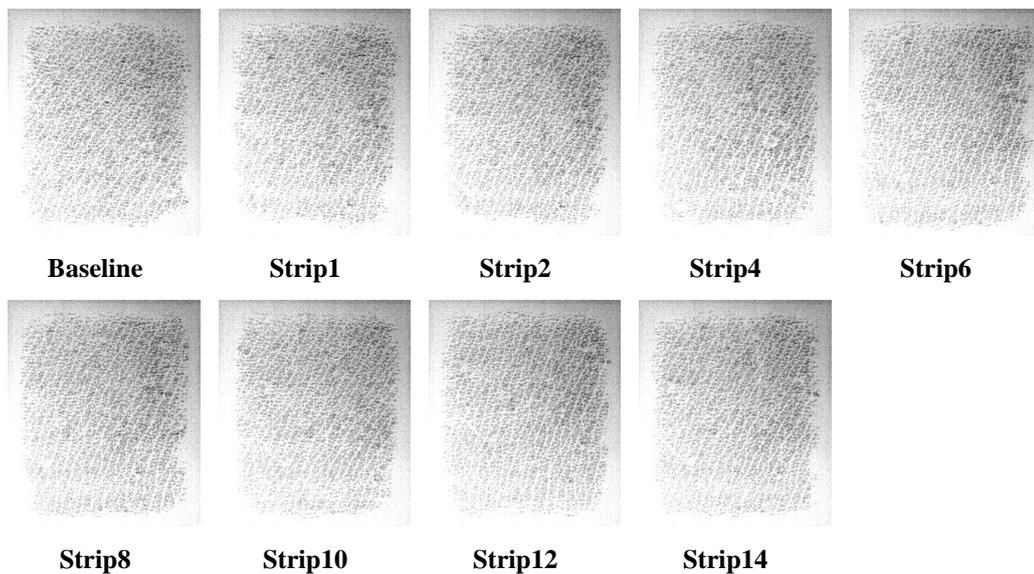
Tape stripping is the process in which tape is applied to skin then lifted to remove the top layer of skin. The main purpose is to figure out the hydration of the different stripped skin. It is a very useful experiment for investigating skin hydration and solvent penetration.

The aim of this experiment is to find out the skin characteristics of the stripped volar forearm with the measurements of the skin hydration by Fingerprint sensor, TEWL by AquaFlux, digital images by ProScopeHR2 and Corneometer.

Additionally, compare with the measurements and find out the correlation between TEWL and Hydration of stripped skin.

In this measurement, the ambient temperature is 23.0°C and related humidity (RH) 49 %, which were set up by the air conditioning. Two skin sites marked as controlled site (F1) and stripping site (F2) were chosen on a female volunteer (aged 20-30, 60kg) left volar forearm near the elbow where it has less hair. The hydration and TEWL in the measurement conditions on the sites of volunteer's both forearms are measured by Fingerprint sensor and AquaFlux. The hydration measurement was repeated three times, and the TEWL and Corneometer only measured once for every two strips up to 14 strips on both sites. Before every single measurement, digital pictures of both sites were taken by ProScope HR2.

Figure 5.5 shows the grey images and digital images produced by Fingerprint sensor and ProScopeHR2.



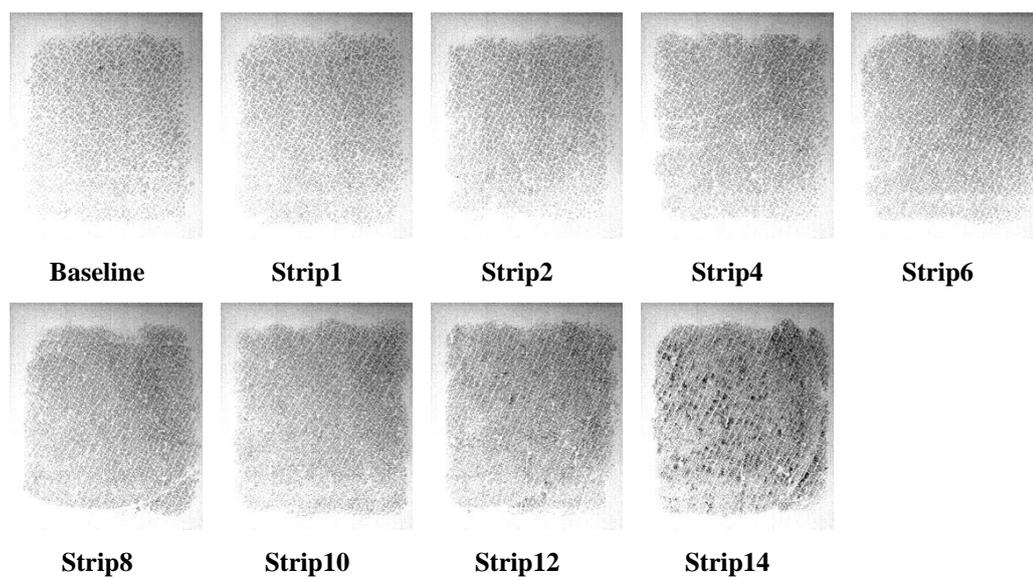
(a).Greyscale images from baseline to 14th strips on F1



(b). Digital images from baseline to 14th strips on F1 underneath.



(c). Digital images from baseline to 14th strips on F1 surface.



(d). Greyscale images from baseline to 14th strips on F2.



(e). Digital images from baseline to 14th strips on F2 underneath.



(f). Digital images from baseline to 14th strips on F2 surface.

Figure 5.5 Grey Images and digital images from baseline to 14 strips on skin sites

Figure 5.5 shows the images captured by Fingerprint sensor and ProScopeHR2 of two skin sites. By comparing the images of F1 with F2 (Figure 5.5(a) and Figure 5.5(d)), it shows clearly that the grey images (skin hydration) get darker and more black spots show up with the increased strips. In other word, the water content of F2 increased after each strip. In the meantime, the skin colour has slightly changed during each strip as shown in Figure 5.5(b), 5.5(c), 5.5(e) and 5.5(f). However, to identify the changes by human eyes is not the best choice, as human visual is affected by the brightness. By using MATLAB programme, it is easy to equalize

the brightness of the images and process the images into numerical and matrix format and calculate the colour components of the skin images.

Figure 5.6 shows the line chart of the Hydration and TEWL results of F1 and F2.

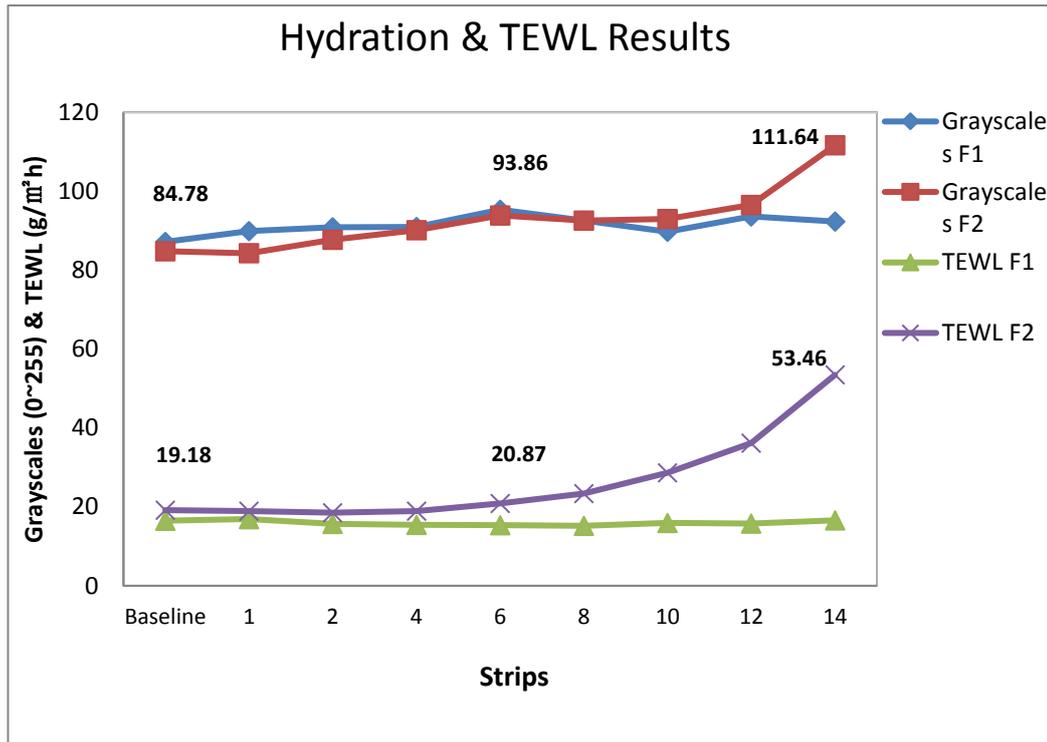


Figure 5.6 Hydration and TEWL results of the tape stripping measurement.

The results of skin hydration and TEWL on F1 and F2 by using Fingerprint sensor and AquaFlux show that both hydration and TEWL get increased slowly at early strips but rapidly after 8th-strip up to 14thstrips. The TEWL has more than twice as much as the 1st strip and hydration increased nearly 40%.

Figure 5.7 shows the Corneometer results of F1 and F2.

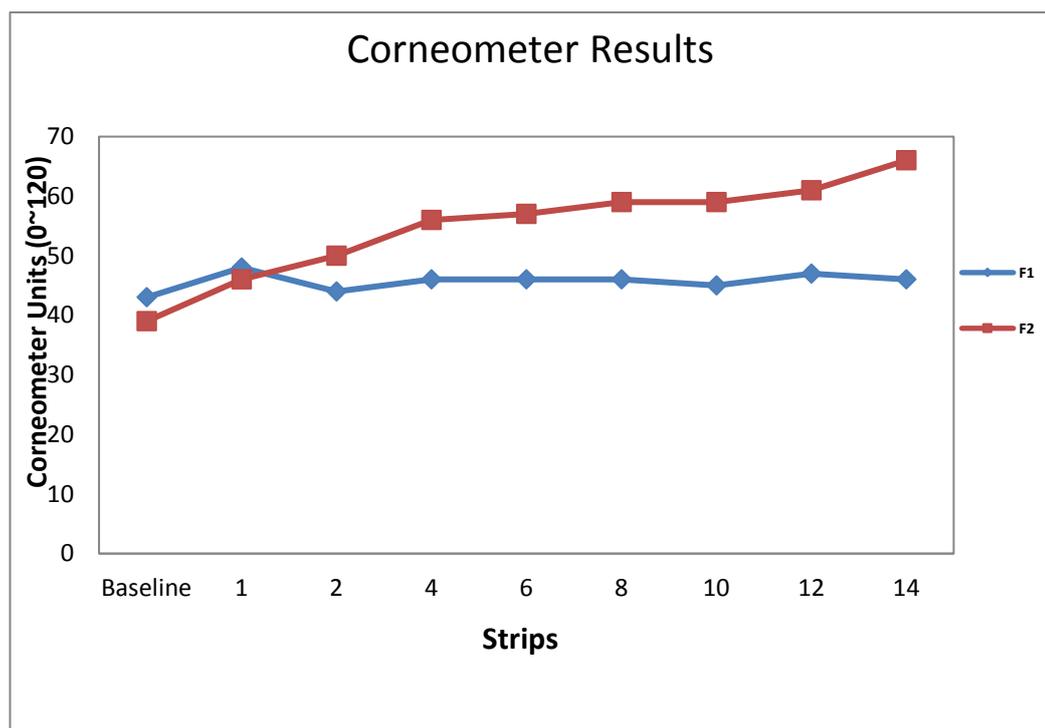
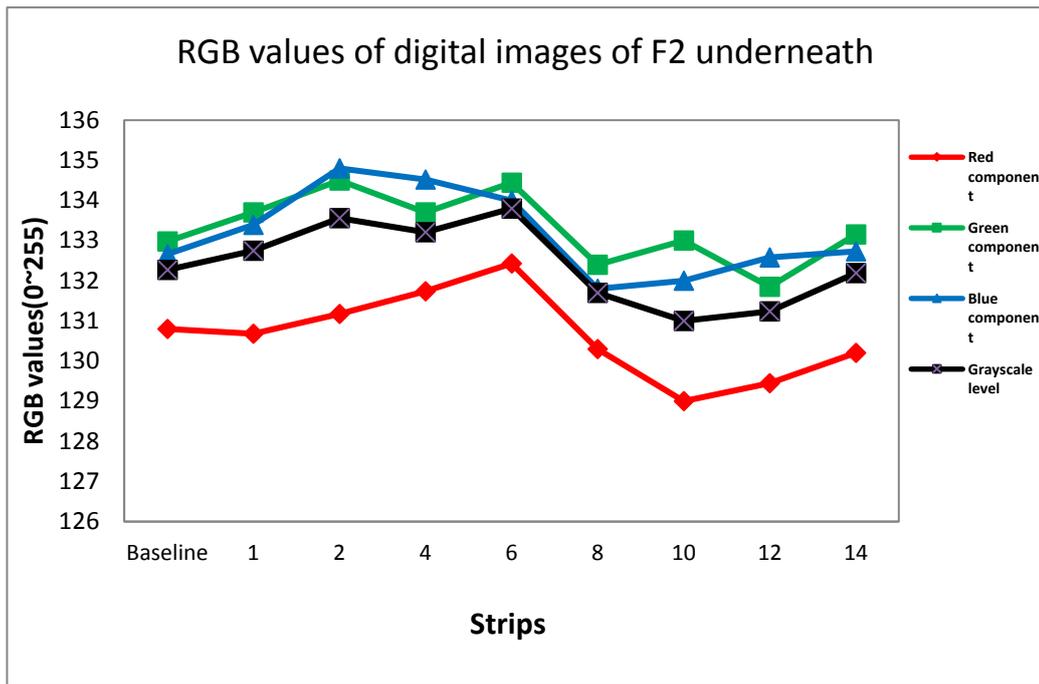


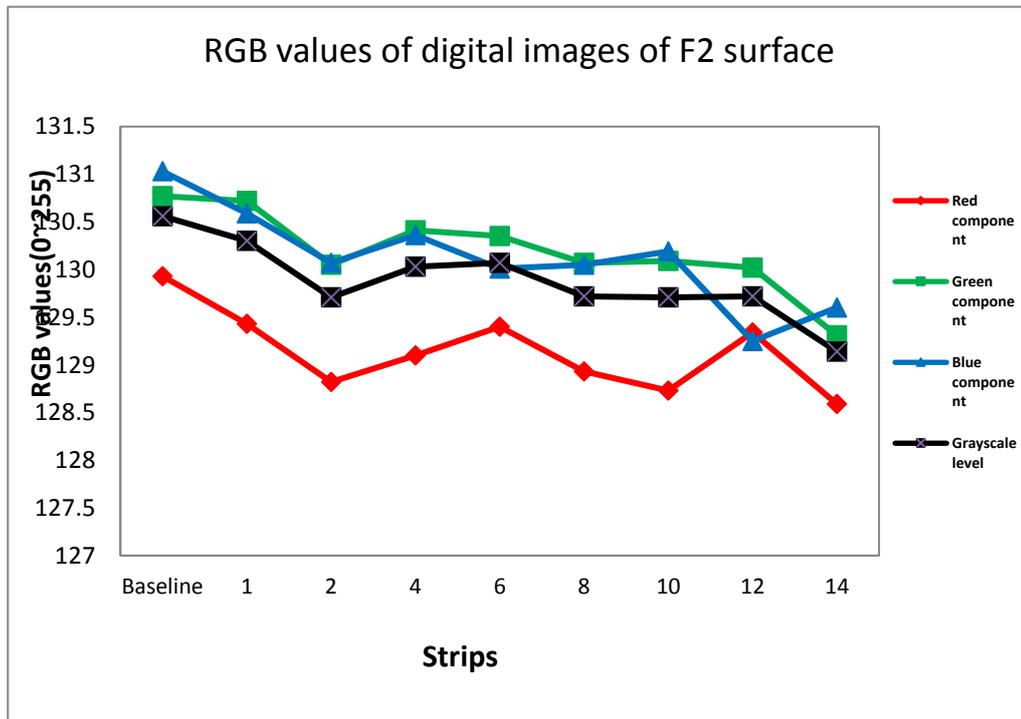
Figure 5.7 Hydration measurement using Corneometer.

The value of F2 in figure 5.7 is increased from baseline 40 up to 66 with the number of strips. Comparing with the Fingerprint sensor results in Figure 5.6, Fingerprint sensor result increased from 84 to 111.

Figure 5.8 (a) shows the processed RGB values of digital images of F2 from underneath. The RGB values fluctuated with no consistent trend.



(a).RGB values of the digital images of F2 underneath



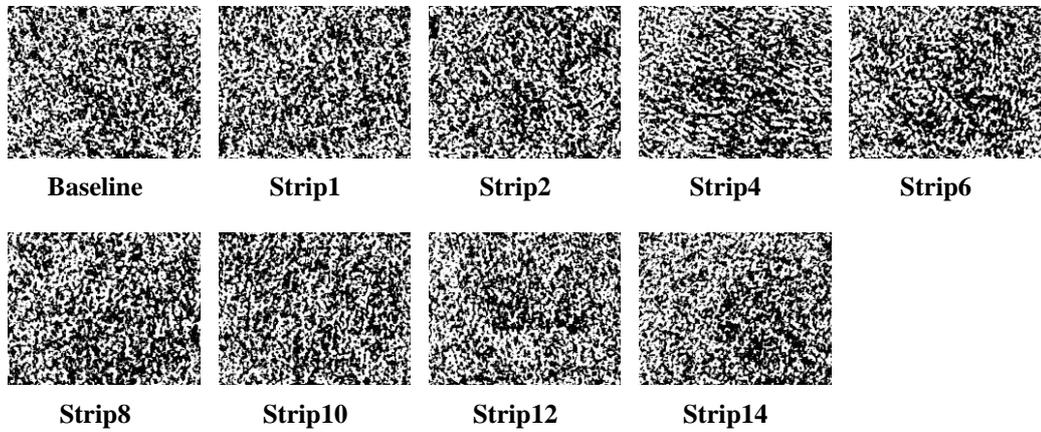
(b).RGB values of the digital images of F2 surface

Figure 5.8 RGB values of the digital images of F2 surface

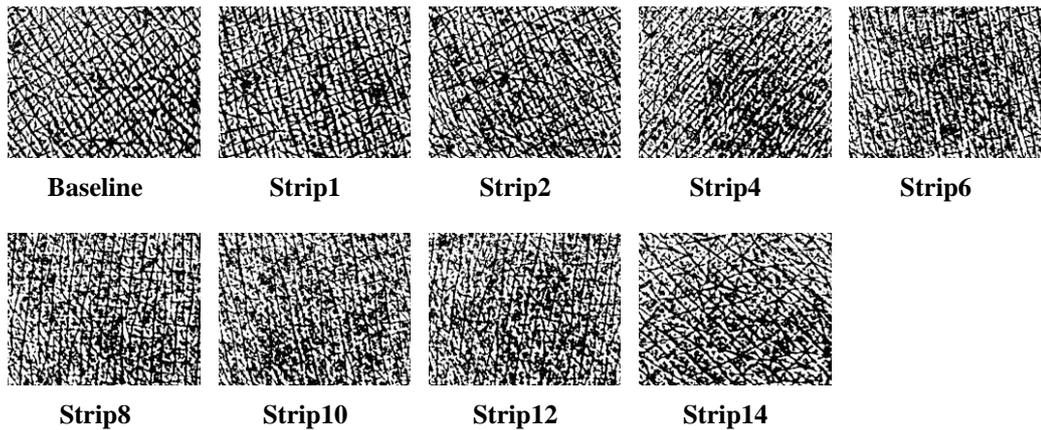
Figure 5.8(b) shows the processed RGB values of the digital images of F2 surface. The RGB values gradually go down slightly.

Although there is LED light source on the head of the ProScope instrument for avoiding environment illuminate inference, results are not consistent. They cannot be used directly without image processing such as the brightness equalization.

Compared with Figure 5.1(a) and 5.1(f), Figure 5.9 shows the processed images of digital images of F2.



(a) Processed binary images from digital images of F2 underneath



(b) Processed binary images from digital images of F2 surface

Figure 5.9 Processed binary images from digital images of F2.

The digital images, processed into binary format with less redundancy in brightness and skin colour, can be used to further study in skin microrelief investigation and demantoglyph extraction.

Figure 5.10 shows the Hydration and TEWL results of two skin sites.

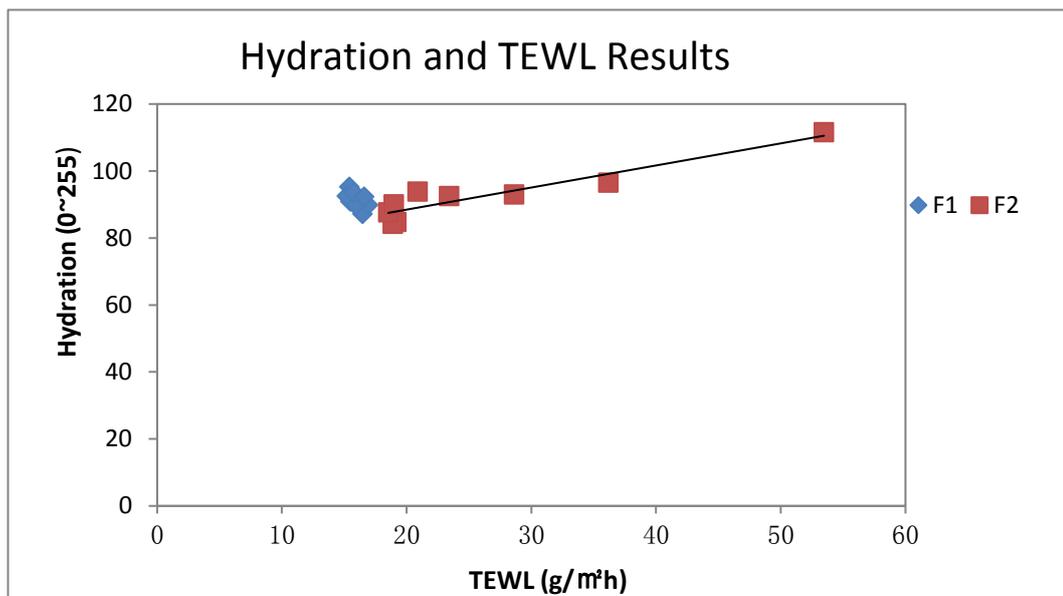


Figure 5.10 Hydration and TEWL results of two skin sites F1 and F2

In summary, Figure 5.5 shows the raw images from Fingerprint sensor, the digital images from ProScope HR2. It is easy to see that the colour of the images of F2 get darker by the number of strips, in other word, hydration of F2 is increased.

Figure 5.6, 5.7 and 5.8 show the results of the TEWL, Hydration and RGB results from AquaFlux, Fingerprint sensor, Corneometer and Proscope HR2. Comparing with the correlation chart in Figure 5.10, the result shows that the TEWL and Hydration has a positive correlation.

However, in Figure 5.8, the RGB values decreased slightly by the number of strips. To identify the changes by human eyes is not a smart choice. Human visual is

affected by the brightness. By using MATLAB programme as shown in Chapter 3, it is easy to equalize the brightness of the images, process the images into numerical and matrix format and calculate the colour components.

5.3.2 SLS Irritation Measurement 1

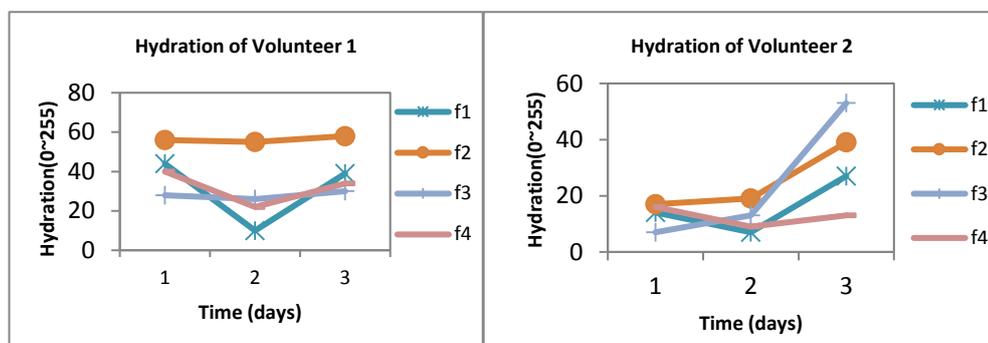
This work is carried out in collaboration with Department of Dermatology, University of Lübeck, Germany, and was funded by an EU short-term scientific mission (STSM) grant. The main object of the experiment is to investigate the effectiveness of measuring skin SLS (Sodium Lauryl Sulphate) irritation with multiple instruments including fingerprint sensor, Corneometer, Tewameter, chroma meter, and PH meter.

SLS (Sodium Lauryl Sulphate), in science also referred to as sodium dodecyl sulfate (SDS), is used in laboratory cleaning procedures and is commonly used to aid lysing cells during DNA extraction. SDS is a common component of many domestic cleaning, personal hygiene and cosmetic, pharmaceutical, and food products, as well as of industrial and commercial cleaning and product formulations.

The measurements were performed under normal ambient laboratory conditions, of 20°C, and 50% RH. Five healthy volunteers (female, aged 40-50) were chosen for the experiments. SLS irritation was achieved by applying 0.5, 1 and 2% SLS water solution (v/v, volume ratio) on three skin sites (F1, F2 and F3) of volunteer's volar forearm. All the volunteers were asked following the SLS irritation instruction: The volunteers need to wash three skin sites for four days, three washes a day, 150 times per wash by using a roller brush on each site, no bathing, no food or drink before measurement. All the measurements were repeated 3 times

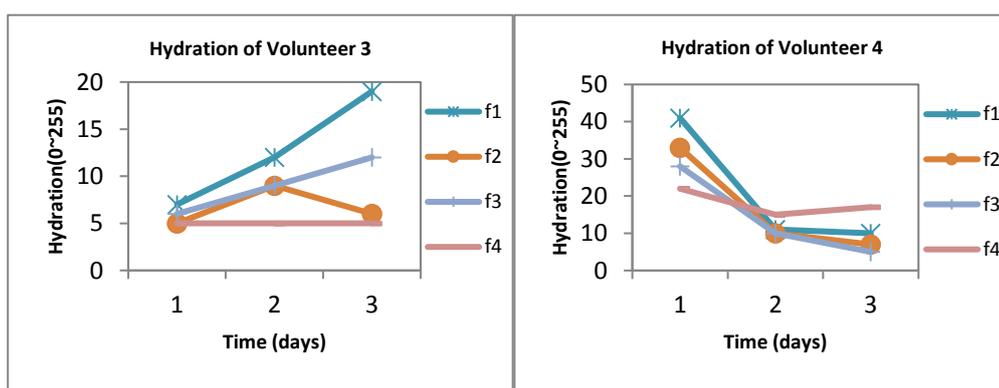
on each site to generate average value except the Corneometer which was repeated 5 times; TEWL measurement performs 2-minutes continual measuring.

Figure 5.11 shows the result of hydration using the fingerprint sensor.



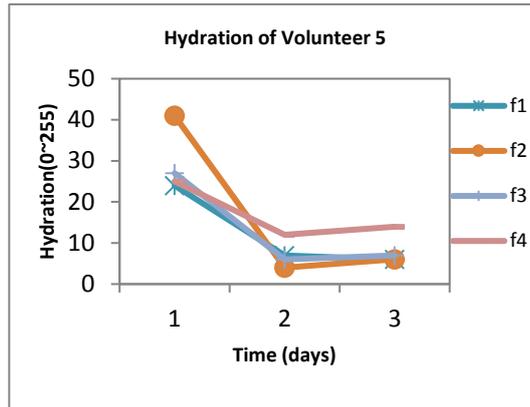
(a)V1

(b)V2



(c)V3

(d)V4

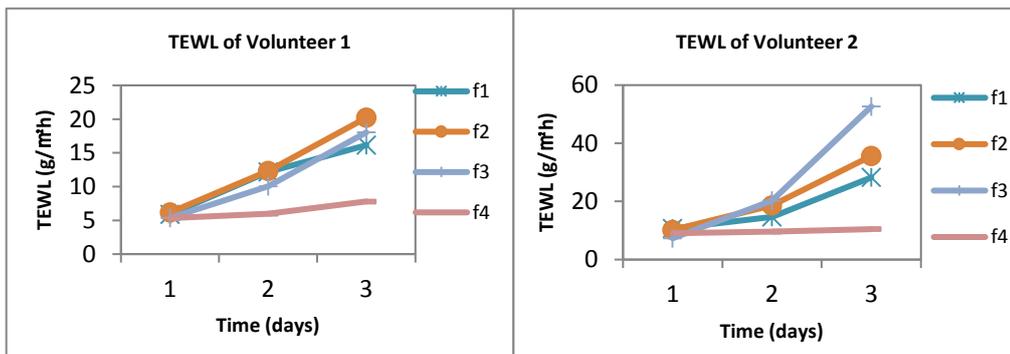


(e)V5

Figure 5.11 Results of hydration using fingerprint sensor

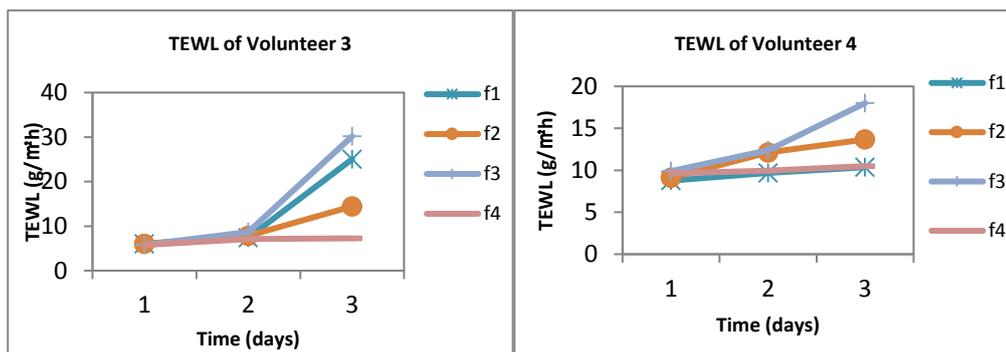
Figure 5.11 shows the hydration results from fingerprint sensor of five volunteers. The hydration of V2 and V3 increased after SLS irritation but V4 and V5 decreased. The inconsistent results are likely due to the poor contact between the sensor and skin, as skin is getting dryer.

Figure 5.12 shows the TEWL results.



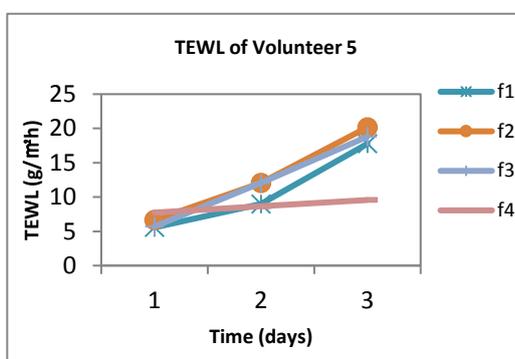
(a)V1

(b)V2



(c)V3

(d)V4

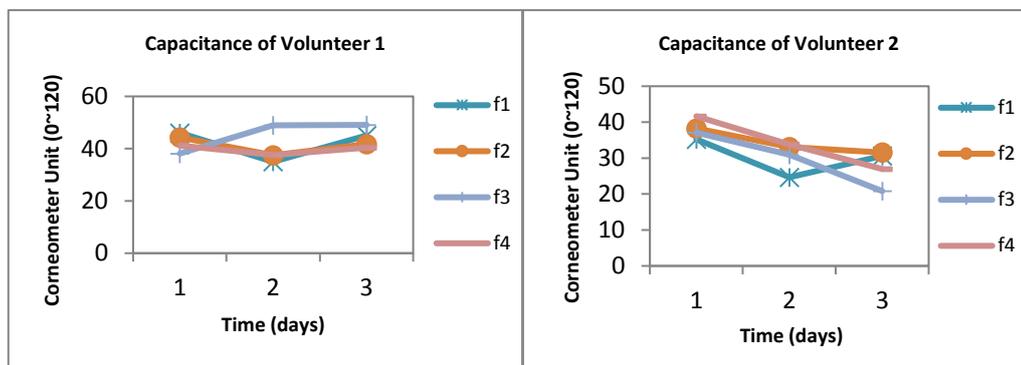


(e)V5

Figure 5.12 TEWL results using Tewameter

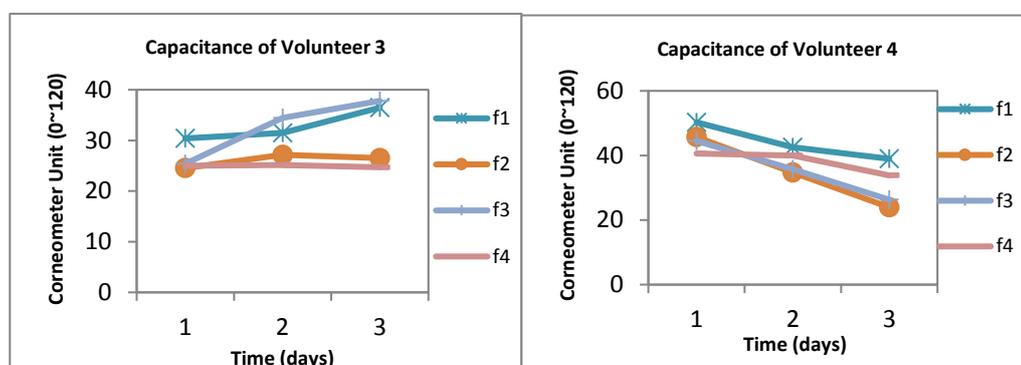
Figure 5.12 shows the TEWL results of volunteers. The results show that the TEWL increase after SLS irritation.

Figure 5.13 shows the Corneometer results.



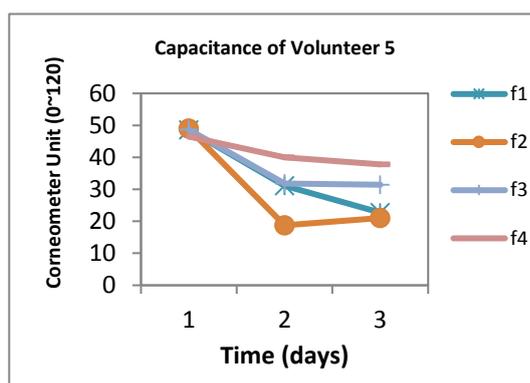
(a)V1

(b)V2



(c)V3

(d)V4

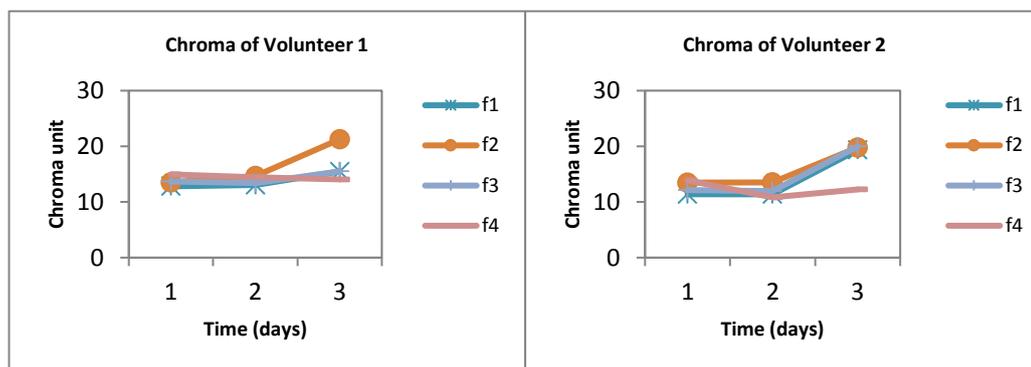


(e)V5

Figure 5.13 Corneometer results of volunteers

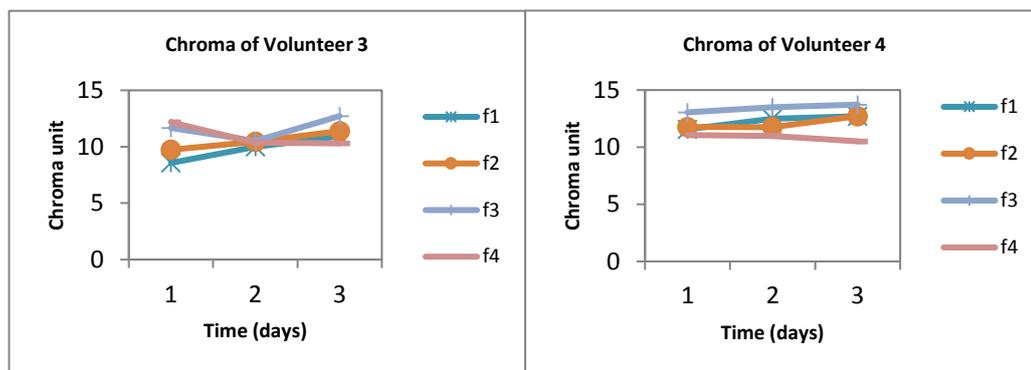
V2, V4 and V5 results in Figure 5.13 show that the capacitance of the skin is decreased after SLS irritation. However, V1 and V3 are not decreased. Again, the inconsistent results here are likely due to the poor contact between the instrument and skin.

Figure 5.14 shows the results of chroma meter.



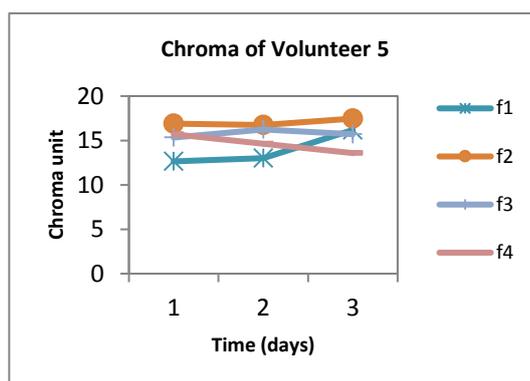
(a) V1

(b) V2



(c) V3

(d) V4

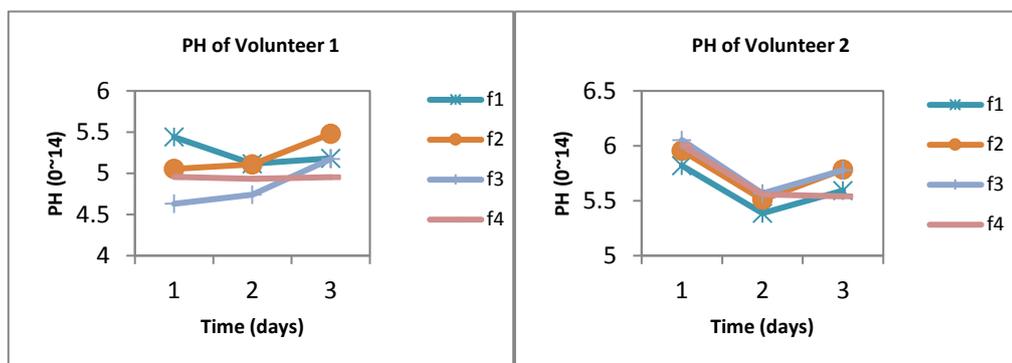


(e)V5

Figure 5.14 Results of Chroma meter

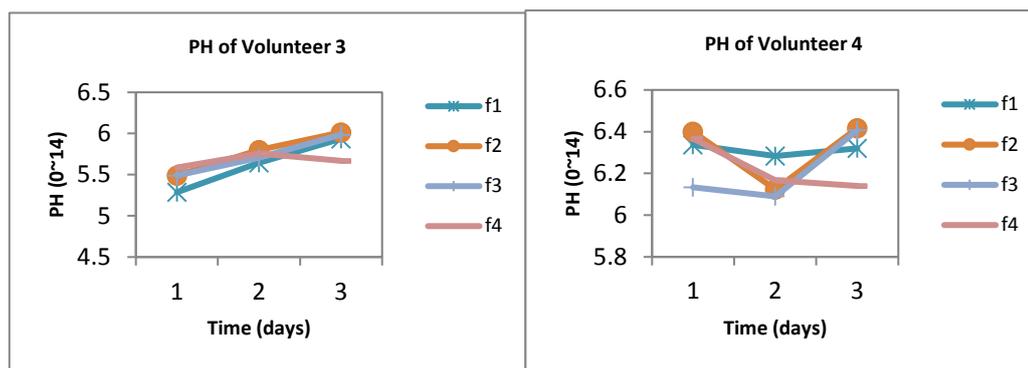
The chroma meter results in figure 5.14 show that the chroma unit of volunteers have a slightly increase after SLS irritation.

Figure 5.15 shows the results of PH meter.



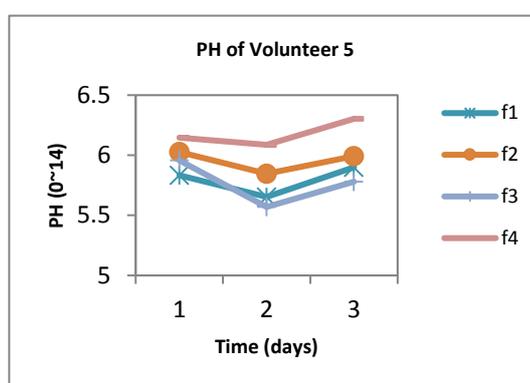
(a)V1

(b)V2



(c) V3

(d) V4



(e) V5

Figure 5.15 Results of PH meter

The PH meter results in Figure 5.15 show that the PH of the skin decrease on day-1 and then increase to a higher level but not the same level as the baseline.

In summary, the hydration of the skin is decreased after SLS irritation and the TEWL is increased; the chroma unit has slightly increase; the PH of the skin decrease to 5.5 and then increase to a higher level; the capacitance of the skin decreased. However, there are some inconsistent results, which is likely due to the poor contact between the instruments and skin.

5.3.3 SLS Irritation Measurement 2

In order to address the issue of inconsistent results mentioned above, the experiment was repeated, but with fewer volunteers and different procedures. This time, the aim is to investigate the skin function and characteristics under a repetitive damage by using different skin measurement instrument. AquaFlux, fingerprint sensor (3 times, average) and Corneometer (10 times, average) were used for the measurement.

In this experiment, 2% SLS solution is mixed by 1g powder and 49ml water. Two skin sites are chosen on two healthy volunteers (V1 and V2, aged 20-30, male) volar forearm, control site (F1) and applying site (F2) with less hair. By using cotton buds, washes must be last one minute and three times per wash. Measurement on day-1 starts at baseline, immediately after first wash (T0), 30 minutes later (T1), another wash and 30minutes later (T2). On day-2 and day-3, only take one wash and measure the baseline and T1 data.

Figure 5.16 shows the Hydration and TEWL results from three instruments.

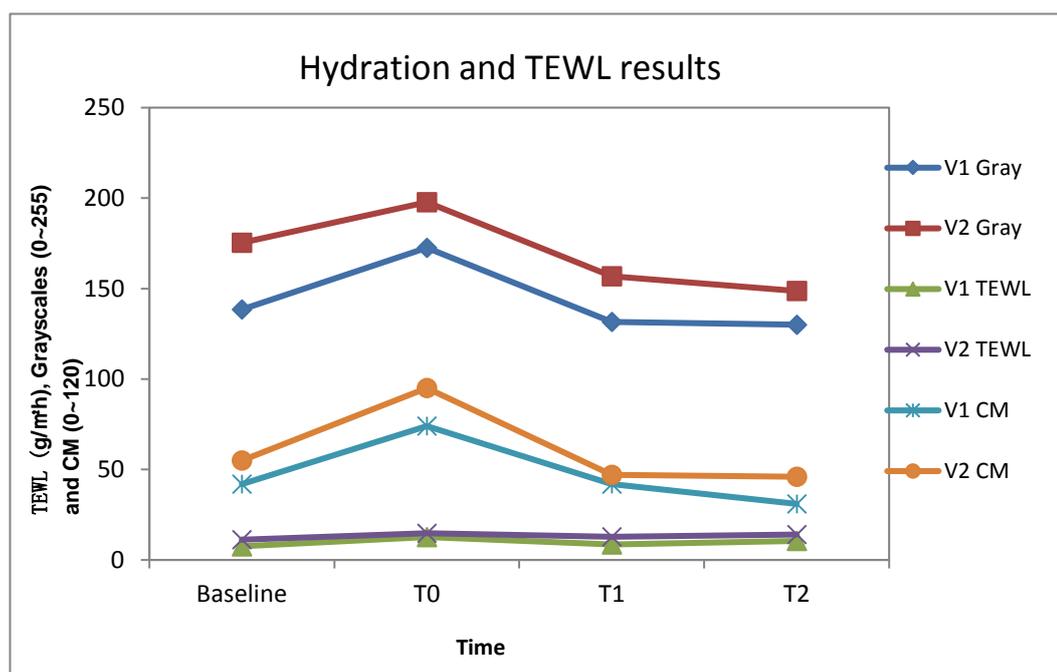
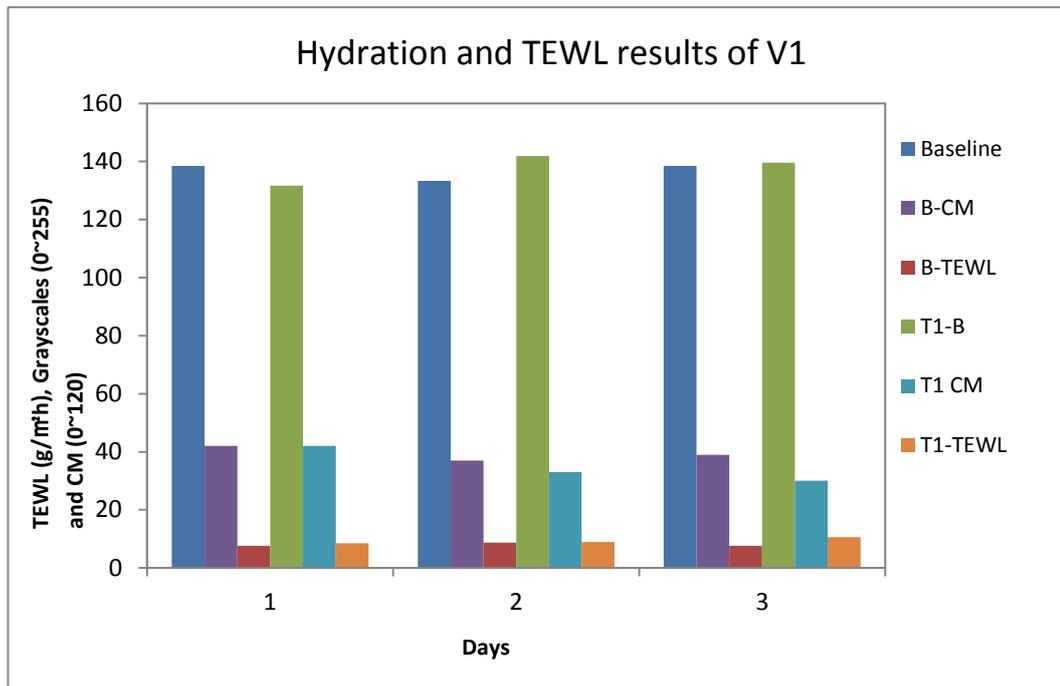


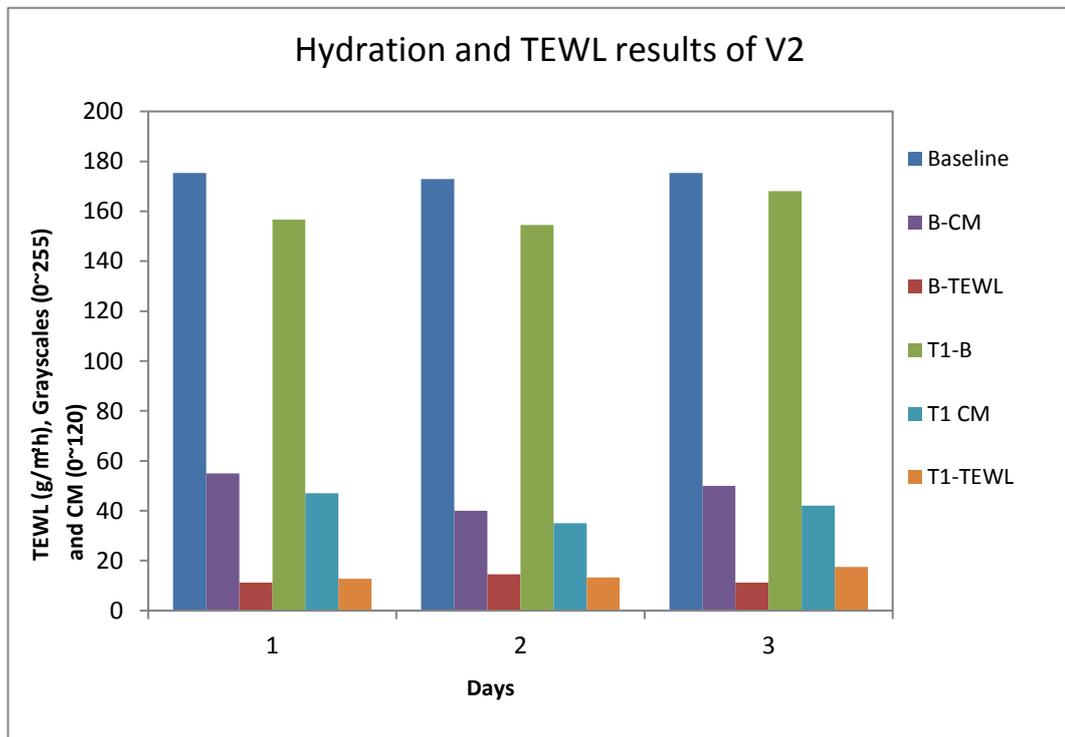
Figure 5.16 Hydration and TEWL results from three instruments of F2 of two volunteers.

The chart shows day-1 result that hydration of F2 increased after SLS irritation at T0, decrease rapidly at T1 and slowly down to T2 which is lower than baseline data. However, TEWL increase lightly, and they are higher than baseline data.

The following Figure 5.17 shows the three-days data on F2 of two volunteers of baseline and T1.



(a). Bar chart of 3-days results of Hydration and TEWL of V1.



(b). Bar chart of 3-days results of Hydration and TEWL of V2.

Figure 5.17 Hydration and TEWL results from three instruments of F2 of two volunteers.

By the hydration results from Figure 5.17(a), the hydration of V1 decreased on day-1 then return to the same level in two days. The TEWL increased slightly. According to the data in Figure 5.11(b), V2 has different results from V1 where the hydration is decreased after every wash and then attempt to recover to the base level. However, the hydration is lower than the baseline data. The TEWL on day-2 is lower than day-3 but higher than day-1 as well as the hydration results.

In this experiment, SLS irritation is applied on human skin. It has been proved that the 2% SLS solution irritates the skin which temporary influences the skin hydration and TEWL. The hydration gets lower and TEWL increases, that makes the skin drier and loses more water from the body. The results shows that the concentration of SLS solution is 2% which not that strong to leave a critical damage on skin.

In addition, the skin function recovered in one or two days. According to the results of two volunteers, V2 has sensitive skin and better recovering function than V1. Although V1 has some resistance against the SLS irritation which makes the skin hydration in a normal level, the skin was damaged according to the increased TEWL and lower hydration on day-3. In other word, the recover function is weaker than V2.

5.3.4 The Occlusion Effects in Capacitive Contact Imaging for In-vivo Skin Damage Assessments

Skin damage, including mechanical injury, chemical hazard and bacterial invasion, is a very important issue but hard to be assessed quantitatively. Commonly, skin damage assessments are largely done through visual assessments, which can be subjective and difficult to quantify. The water in SC plays an important role in

skin's cosmetic properties as well as its barrier functions, and SC water concentration and TEWL are two key indexes for skin characterizations. There is a need to develop a new, quantitative, and simple methodology that can quantify the skin damage assessments. As it presented in Chapter 2, capacitive contact imaging based fingerprint sensors, originally designed for biometric applications, has shown potential for skin hydration imaging, surface analysis, 3D surface profile, skin micro-relief as well as solvent penetration measurements (Ou X., 2013, Xiao P., 2012, Singh H., 2008, Batisse D., 2006, Bevilacqua A., 2005, Leveque J.L., 2003). With the capacitive contact imaging, the skin surface water concentration distribution map can be measured. By occluding the skin with capacitive imaging sensor over a period of time, as water dynamically builds up underneath the sensor surface due to the blockage of TEWL, it can also generate time-dependent skin occlusive hydration curves. The purpose of this study is to develop a new methodology for skin damage assessments, which can even differentiate different types of damage, by using skin capacitive contact imaging occlusion measurements, as well as the TEWL measurements.

According to diffusion theory, the skin occlusion can be described by following one dimensional diffusion equation with following initial condition and boundary condition.

$$\left\{ \begin{array}{l} D(H) \frac{\partial^2 H}{\partial z^2} = \frac{\partial H}{\partial t}, 0 \leq z \leq L \\ H(z, 0) = f(z) \\ H(L, t) = H_1 \\ -D \frac{\partial H}{\partial z} \Big|_{z=0} = 0 \end{array} \right. \quad (5.1)$$

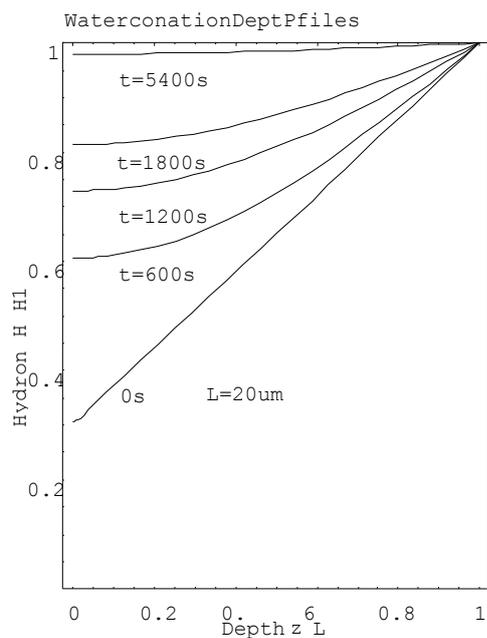
where $H(z,t)$ is the skin water content at depth z and time t , L is SC thickness, $D(H)$ is the SC water diffusion coefficient, which is a function of water content $H(z,t)$,

$f(z)$ is the initial skin water distribution within SC. In this case, we can assume it is a linear distribution, defined by

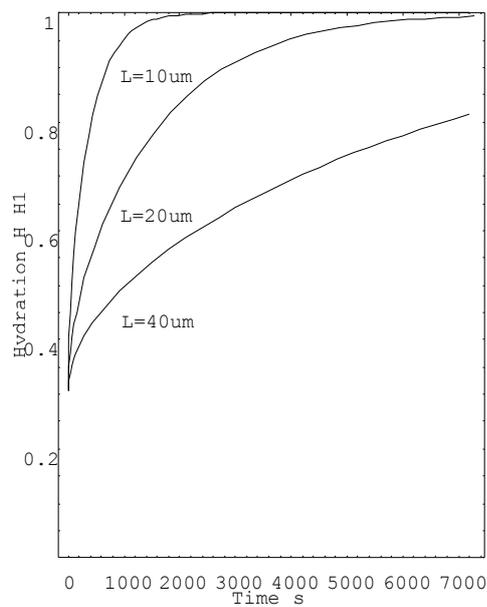
$$f(z) = H_0 + \frac{H_1 - H_0}{L} \times z \quad (5.2)$$

Where H_0 is the SC surface water concentration, and H_1 is the SC bottom water concentration. In Eq.(5.1), at the skin surface ($z=0$), there is zero flux due to occlusion, and at the SC bottom ($z=L$), we assume there is a constant water concentration H_1 . We can solve the Eq.(5.1) by substituting Eq.(5.2) into Eq.(5.1), and the solution can be expressed as,

$$H(z,t) = H_1 + \frac{2}{L} \sum_{n=0}^{\infty} \left(e^{-\frac{D(2n+1)^2 \pi^2 t}{4L^2}} \times \cos \frac{(2n+1)\pi z}{2L} \times \left(\frac{2L(-1)^{n+1} H_1}{(2n+1)\pi} + \frac{2L(H_1 (sn+1)\pi \cos(n\pi) + 2(H_1 - H_0)(1 + \sin(n\pi)))}{(2n+1)^2 \pi^2} \right) \right) \quad (5.3)$$



(a)



(b)

Figure 5.18 The SC normalized water concentration depth profiles at different time during the occlusion with $L=20\mu\text{m}$ (a), and the time dependent normalized surface water concentration levels of three different SC thickness (b).

Figure 5.18 shows results of above solution, the left plot shows the SC water concentration depth profiles at different time during the occlusion, using normalized the depth (z/L , $L=20\mu\text{m}$) and normalized water concentration (H/H_1 , $H_1=80\%$ $H_0=24\%$), and right plot shows the time dependent normalized surface water concentration (H/H_1 , $H_0=80\%$ $H_0=24\%$) levels of three different SC thickness ($L=10\mu\text{m}$, $20\mu\text{m}$, $40\mu\text{m}$).

The results show that different SC thicknesses have different times to reach steady state, for the SC with $20\mu\text{m}$ thickness, which is typical SC thickness in volar forearm. It is about 30 minutes to reach 80% of H_1 and about 2 hours to reach the steady state, i.e. 100% of H_1 .

In this experiment, skin sites on volar forearms of three healthy volunteers, aged 25 -45, were chosen for the measurements. The skin test sites were deliberately damaged by intensive washes, tape stripping or sodium lauryl sulfate (SLS) irritation. Three measurements were performed at the different period of time, and only one test site was chosen on each forearm, to avoid cross interference. The measurements were repeated a few times, to make sure the results are repeatable, but only one set of the data was presented. Intensive washing used room temperature running water and washing-up liquid, rubbing the site gently for 3 minutes with a finger. After each wash, the skin site was carefully patted dry with a tissue before the measurements. Tape stripping was performed 20 times per site by the use of standard stripping tape. SLS irritation was achieved by applying 2% SLS water solution (v/v, volume ratio) on skin. Capacitive contact imaging measurements and TEWL measurements were performed both before and immediately after the skin was damaged. They were both on the same skin site,

with capacitive contact imaging measurements performed after the TEWL measurements, which took about 1 minute. The skin occlusion measurements using capacitive contact imaging to occlude the skin test sites for a period of one minute. The skin capacitance images were recorded continuously. The average greyscale values of the images were then calculated at different times during occlusion. Since greyscale values are proportional to SC hydration, the plots of greyscale value against time, can be interpreted as SC hydration against time.

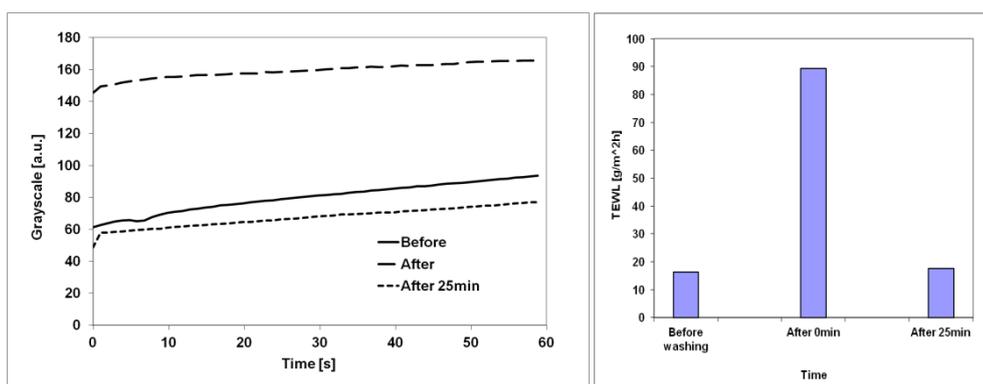
All the measurements were performed under normal ambient laboratory conditions, of 20-21°C, and 40-50% RH. The volar forearm skin sites used were initially wiped clean with ETOH/H₂O (95/5) solution. The volunteers were then acclimatized in the laboratory for 20 minutes prior to the experiments.

Figure 5.19 shows capacitive contact imaging occlusion curves and corresponding TEWL results of intensive washes, tape stripping and SLS irritation measurements. The intensive washes produced small changes in the shapes of the contact imaging occlusion curves. The general higher greyscale values of the occlusion curve immediately after the washes indicate general higher SC hydration levels, which may be caused by two factors, namely (i) superficial absorption of the water used in the washes; (ii) the removal of superficial SC cells during washing. After 25 minutes recovery time, the average greyscale values were found to have returned to near-normal level. However, there is an undershoot, which suggests a dehydration after the intensive washes, possibly due to the removal of some superficial SC cells and the resultant loss of some SC barrier function. The TEWL results followed a similar trend, and also confirmed the undershoot.

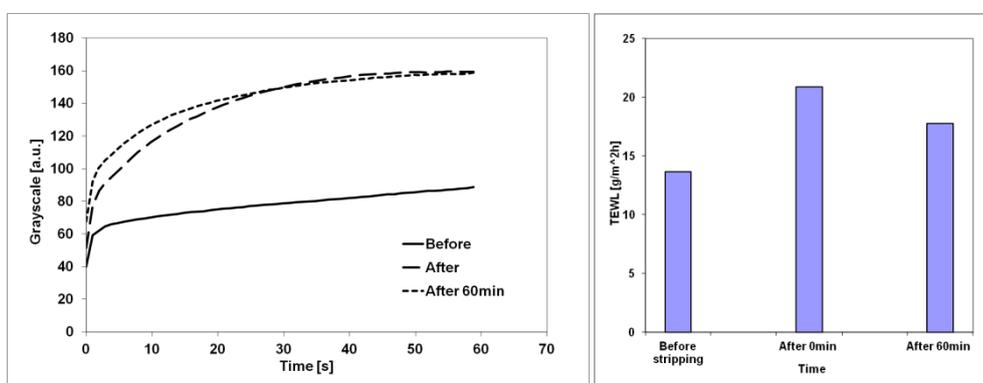
In tape stripping, the time dependent contact imaging occlusion curves show a significant difference in shape (i.e. more curvature) between normal skin and

damaged skin. This curvature change reflects the SC structure change due to tape stripping. Even after 60 minutes, the contact imaging occlusion curves were found to be still significantly different from those of normal skin, indicating that SC was still damaged. The TEWL values, however, have started returning to its normal value after 60 minutes, indicating that although SC is still damaged, it starts to recover.

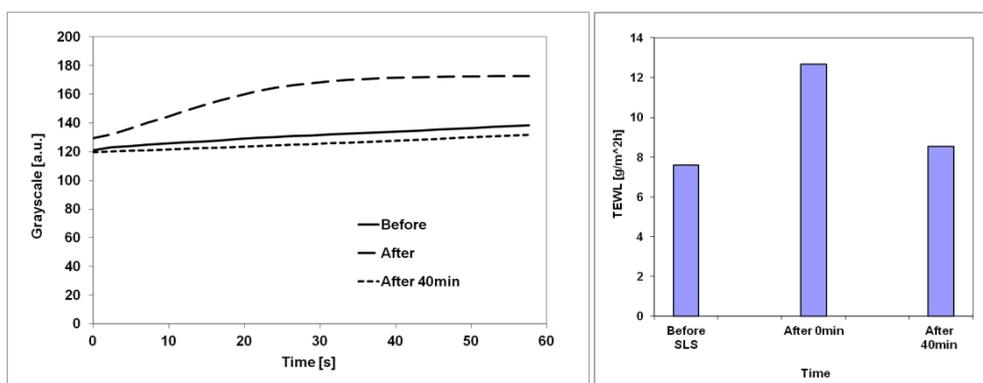
This curvature change can be confirmed by more detailed, longer time scale tape stripping measurements, as shown in Figure 5.20. As the number of tape strips increased, not only the level of contact imaging occlusion curves changed, but also the curvature as well, with after 20 tape strips most significant, as shown in Figure 5.20 (left). Then as the skin recovery in the following days, the curvature of the contact imaging occlusion curves gradually recovered to its normal level, as shown in Figure 5.20 (right).



(a) Intensive washing



(b) Tape stripping



(c) SLS irritation

Figure 5.19 Skin capacitive contact imaging occlusion curves and corresponding TEWL results of intensive washing; tape stripping; and SLS irritation.

Capacitive contact imaging measurements were taken after the TEWL measurements, which were taken immediately after the damage, and each TEWL measurement lasts about 1 minute.

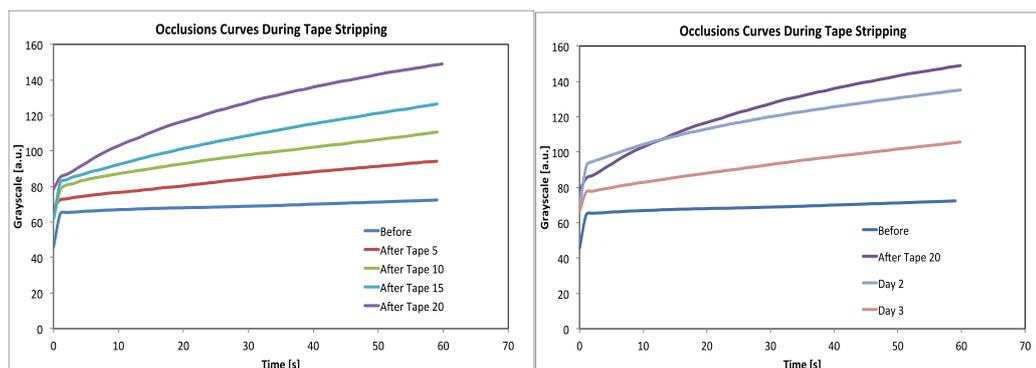
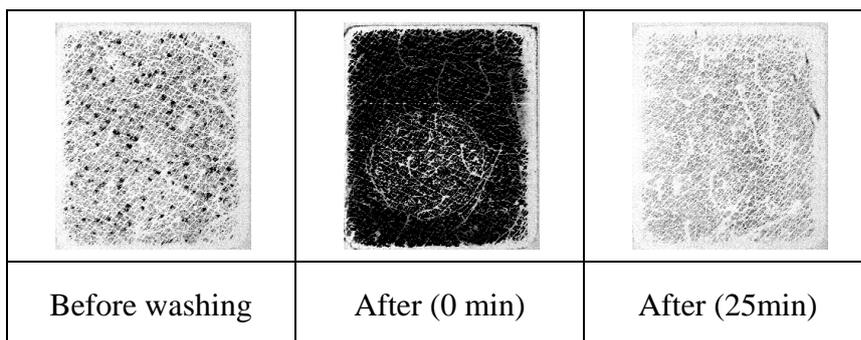


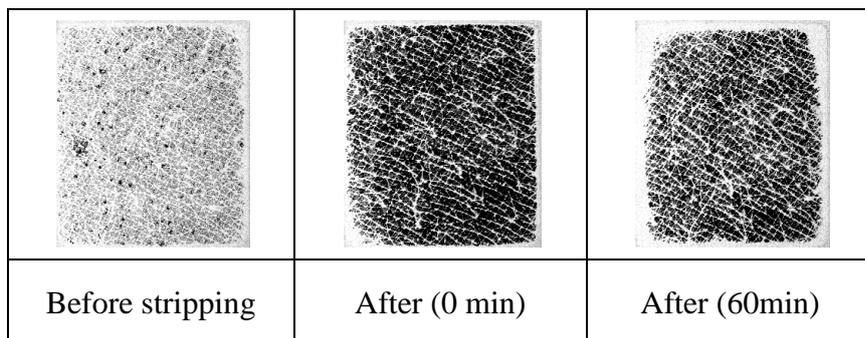
Figure 5.20 Skin capacitive contact imaging occlusion curves during tape stripping (left) and days after tape stripping (right).

Back to Figure 5.19, in SLS irritation, both the contact imaging occlusion curve and TEWL value changed after irritation, but largely recovered after 40 minutes. It is interesting to point out that the three types of skin damage produce three distinctive occlusion curves, which indicates that, according to our theoretical modelling, the SC surface hydration and SC structure are quite different under the different types of skin damage. This suggests that the shapes of capacitive contact imaging occlusion curves can provide extra information about skin damage. The different types of damages changed skin surface hydration and structure differently and therefore generated different types of occlusion curves. The results also show that TEWL results can reflect the skin damage, but cannot differentiate the damages. Therefore, the combination of capacitive contact imaging occlusion measurements and TEWL measurements can provide more detailed, comprehensive information about skin damage.

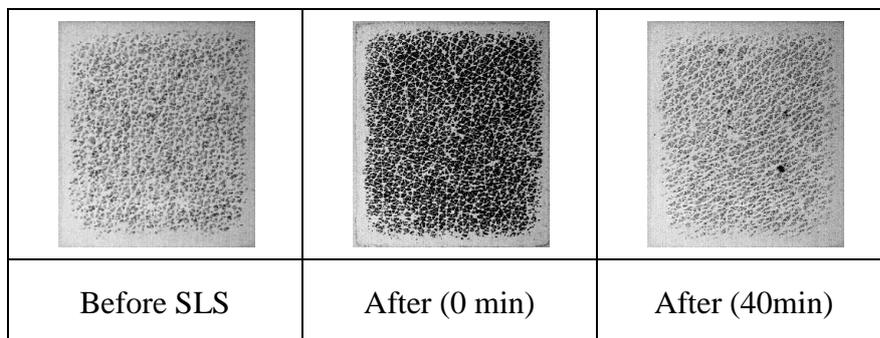
Figure 5.21 shows corresponding capacitive contact images of intensive washes, tape stripping and SLS irritation measurements. The skin images are generally getting darker after damage, which indicates higher water content in SC. In both intensive washes and SLS irritation, the lighter recovery skin images indicate there is a drying effect after the damage. The lighter areas in the images immediately after the intensive washing are imprints from the TEWL measurement head.



(a) intensive washing



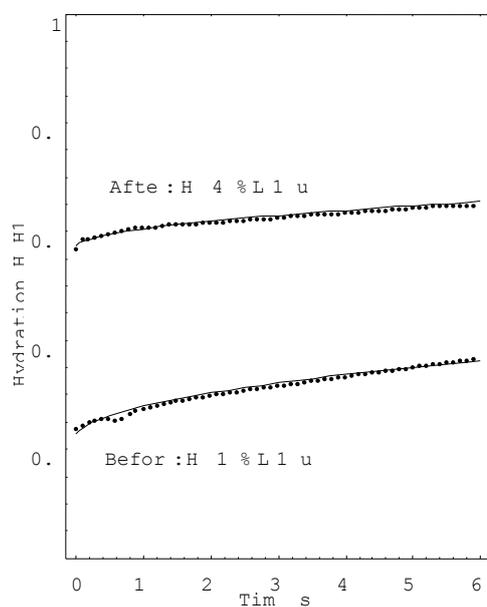
(b) Tape stripping



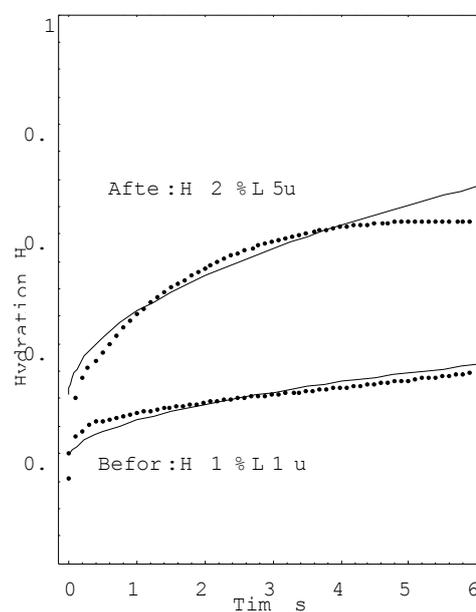
(c) SLS irritation

Figure 5.21 Skin capacitive contact images of intensive washing; tape stripping; and SLS irritation.

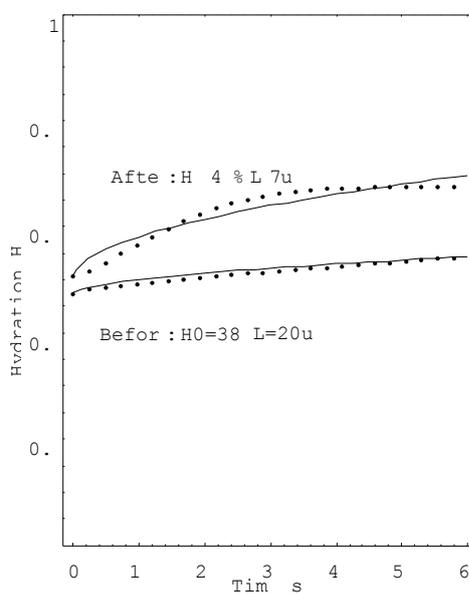
If we assume the maximum greyscale representing 100% water content, and zero greyscale represent 0% water content, then we can compare the theoretical results using Eq.(5.3) with above experimental results, see Figure 5.22. The comparison results show that the intensive washing has significantly increased the SC surface water content, but only slightly reduced the SC thickness, whilst the 20 tape stripping only slightly increase the SC surface water content, but significantly reduced the SC thickness. It is worth mentioning that the reduced SC thickness in theoretical modeling data after SLS irritation is more likely to reflect the changes of water distribution in SC, rather than the changes of SC structure. The more pronounced occlusion effect in Figure 5.22(b) is likely due to the significant skin structure changes due to tape stripping. Overall, the theoretical data matches better with normal skin data, the significant mismatch of theoretical data and the data after 20 tape stripping, indicate that tape stripping has significantly changed the structure of the SC. Clearly, when the capacitive contact imaging is calibrated, we will be able to get the SC surface content and SC thickness values by analysing the experimental results using the mathematical model described in Eq.(5.3).



(a) Intensive washing



(b) Tape stripping



(c) SLS irritation

Figure 5.22 The comparison of theoretical results and experimental results, the intensive washing (a), the tape stripping (b), and SLS irritation (c).

The results show that the shapes of the capacitive contacting imaging occlusion curves can be related to skin conditions, and different types of skin damage have different shapes of occlusion curves. The TEWL measurements can reflect the skin damage but cannot differentiate different types of damages. Therefore, the combination of skin occlusions using capacitive contact imaging and TEWL measurements can provide useful, complementary information about skin damage, and have potential as a new methodology for in-vivo skin damage assessments. The comparison of theoretical data and experimental data shows that the intensive washes change more of the SC surface water content, and the tape stripping changes more of the SC thickness.

5.4 Dermabrasion

This experiment is carried out at Sinlen Beauty Clinic, London. Dermabrasion is a standard cosmetic treatment to improve the look of the skin. Dermabrasion typically removes the top to deeper layers of the epidermis, and extend into the reticular dermis, there is always minor skin bleeding. The procedure carries risks of scarring, skin discolouration, infections, and facial herpes virus (cold sore) reactivation. In aggressive dermabrasion cases, there is often tremendous skin bleeding and spray during the procedure that has to be controlled with pressure. Afterward, the skin is normally very red and raw-looking. Depending on the level of skin removal with dermabrasion, it takes an average of 7–30 days for the skin to fully heal (re-epithelialize). Often this procedure was performed for deeper acne scarring and deep surgical scars.

In this experiment, only SC layer needs to be removed.

The aim of this experiment is to investigate the effect of dermabrasion using AquaFlux and Epsilon. To further study the SC characteristics, weight the removed SC in order to calculate the thickness of removed SC layer.

The ambient temperature is 23.0°C and related humidity (RH) 49 %, the skin site on right volar forearm (F1) was chosen to perform a standard dermabrasion on three volunteers (V1, V2 and V3, aged 20-30, one female V3) using AquaFlux and Epsilon to measurement Hydration and TEWL changes before and after every 10 minutes (Baseline, T0, T1 and T2). The removed SC is weighted for further investigating the thickness of removed SC layer.

The following Figure 5.23 shows the images from Epsilon.

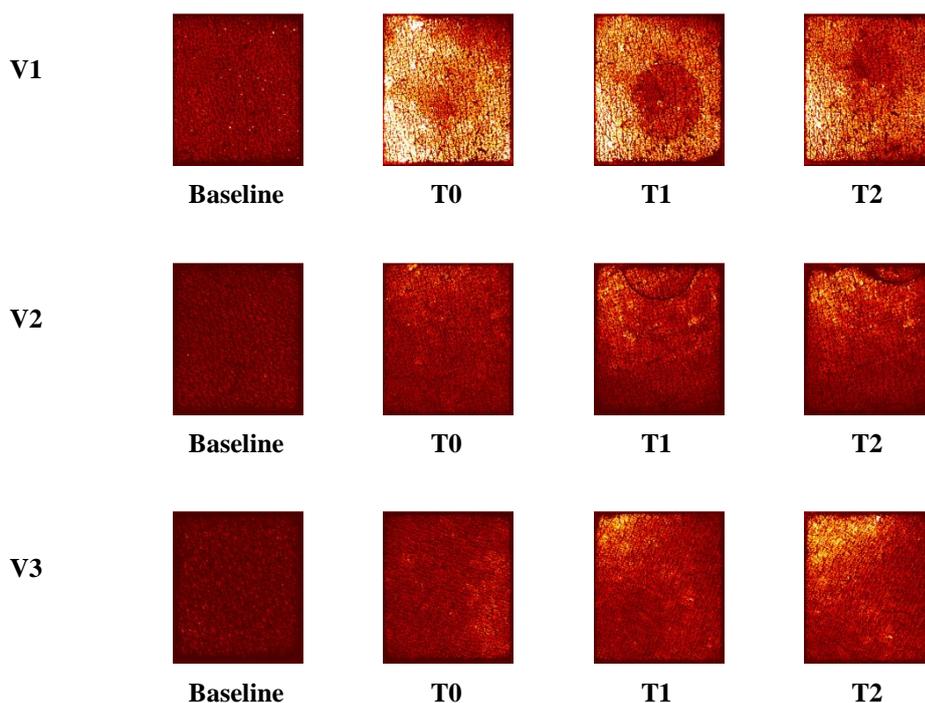


Figure 5.23 Images of hydration measurement using Epsilon on three volunteers before and after dermabrasion treatment from T0 to T2 every 10-minute slot.

Figure 5.23 shows the Greyscale images of hydration measurement using Epsilon on three volunteers before and after dermabrasion treatment from T0 to T2. The

baseline images are darker than those in T0 which were using dermabrasion treatment on the skin, the colour gets bright much more than the baseline images. At T2, the V1 image gets slightly darker but for V2 and V3 they get brighter than T0 and T1.

Figure 5.24 shows the TEWL results from AquaFlux.

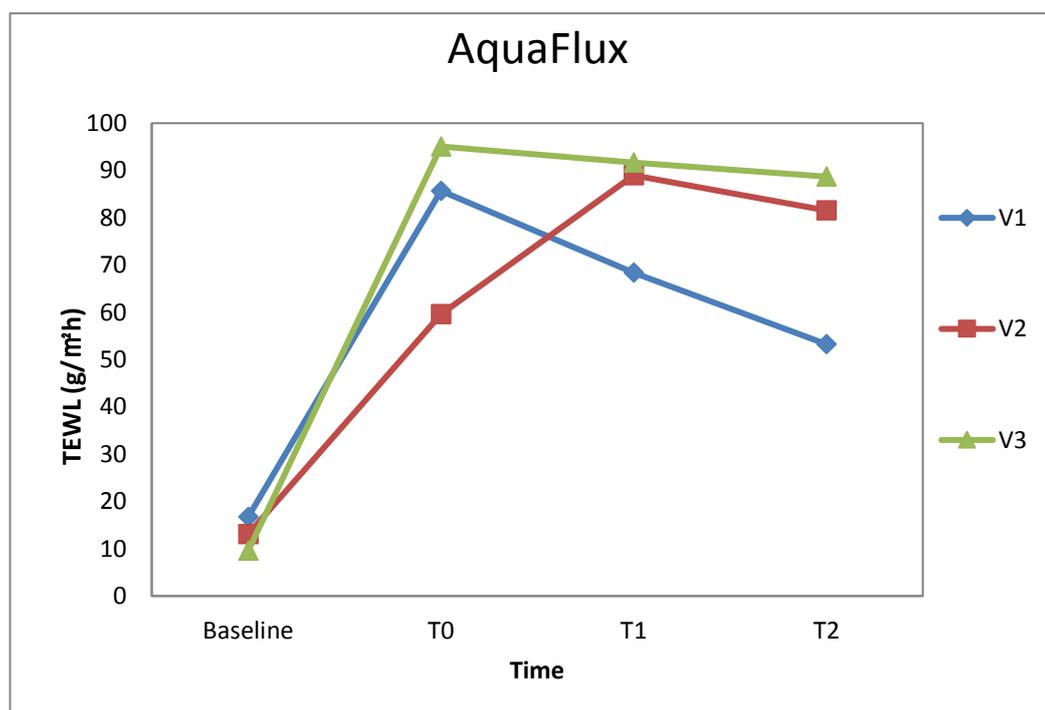


Figure 5.24 TEWL measurement of three volunteers of F1 by AquaFlux

The TEWL results of V1 and V3 reached an extremely high level from 10 upto 90 after dermabrasion treatment on F1 and then force down to a lower level by time.

Figure 5.25 shows the hydration measurement of three volunteers.

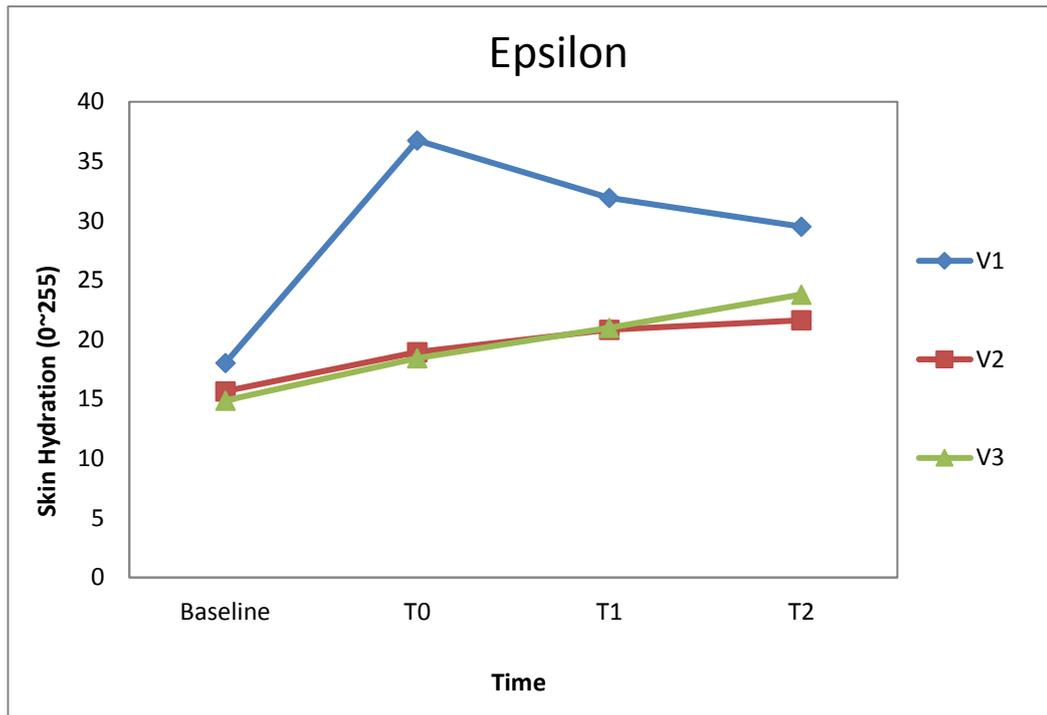


Figure 5.25 Hydration measurement of three volunteers of F1 by Epsilon

The skin hydration of V2 and V3 is increasing by time. But in the result of V1, the hydration reached at a high level after dermabrasion treatment and then force down to a lower level almost the same amount of V2 and V3.

Table 5.1 shows the weight of the removed SC of three volunteers before and after dermabrasion.

	Before (mg)	After (mg)	SC (mg)
V1	15.3	19.0	3.7
V2	17.6	24.0	6.4
V3	21.6	28.0	6.4

Table 5.1 The weight of the removed SC of three volunteers.

The removed skin is cylinder which has a radius r and height h , then its volume V is given by

$$V = \pi r^2 h \quad (5.4)$$

By using the weight of removed skin m divided by skin density $\rho(1.33g / cm^3)$, the volume V also can be given by

$$V = \frac{m}{\rho} \quad (5.5)$$

By using formula 5.4 and 5.5, the estimated thickness of removed skin is given by

$$h = \frac{m}{\rho\pi r^2} \quad (5.6)$$

Where r is the radius of the removed skin area (0.752cm), m is the weight of the removed SC.

By using formula 5.3, the thickness of the three volunteers are given (Table 5.2)

Volunteers	Thickness (μm)
V1	15.75
V2	27.24
V3	27.24

Table 5.2 thickness of SC layers of different volunteers.

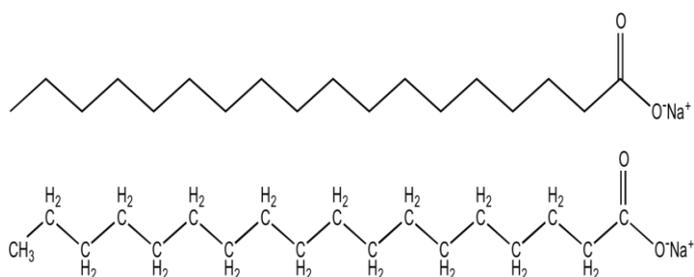
With considerations of the errors in weighing, there is some uncollectable SC losing during dermabrasion treatment. The estimated amount of the thickness of the removed SC could be 30 to 50 μm .

In summary, human skin has protecting function against water loss and recovering function for damaged skin. The skin hydration and TEWL increased due to the dermabrasion on the skin. Additionally, the skin recovering function depends on the damage level of SC. The Epsilon measures greyscale images in figure 5.7 for understanding the skin hydration visually.

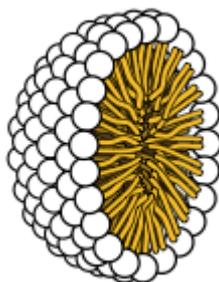
Table 5.1 and 5.2 show the weight and the thickness of removed SC of three volunteers which proved that V1 lost less SC than V2 and V3. Compare with the Aquaflux and Epsilon results which shown in figure 5.8 and figure 5.9, both TEWL and Hydration of V1 recovered faster than V2 and V3. In other word, V1 has better hydration and less TEWL than V2 and V3. This result proved that the SC is very important for protecting function against water loss from the body.

5.5 Soap Drying Effect

This experiment is carried out in collaboration with ITC Ltd, India, which is interested in the soap drying effect on the skin. In chemistry, soap is a water-soluble salt of a fatty acid (IUPAC 1997). Household uses soaps for washing, bathing, and other types of housekeeping, where soaps act as surfactants, emulsifying oils to enable them to be carried away by water. When used for cleaning, soap allows insoluble particles to become soluble in water, so they can then be rinsed away. In industry, they are also used in textile spinning and are important components of some lubricants.



(a) Two images of the chemical structure of sodium stearate, the main ingredient in soaps



(b) Structure of a micelle

Figure 5.26 Structure of a micelle, a cell-like structure formed by the aggregation of soap subunits (such as sodium stearate): The exterior of the micelle is hydrophilic (attracted to water) and the interior is lipophilic (attracted to oils).

For example: oil/fat (David. J 2006) is insoluble in water, but when a couple of drops of dish soap are added to the mixture, the oil/fat solubilizes into the water. The insoluble oil/fat molecules become associated inside micelles, tiny spheres formed from soap molecules with polar hydrophilic (water-attracting) groups on the outside and encasing a lipophilic (fat-attracting) pocket, which shields the oil/fat molecules from the water making it soluble. Anything that is soluble will be washed away with the water.

This section presents the soap drying effect assessment with three brands of soaps. OTTER, AquaFlux and Epsilon were used for skin hydration and TEWL measurements.

As the background introduction mentioned above, the use of soap can wash away the oil and fat and makes the skin dry and sensitive to the environment. The aim of this experiment is to assess the drying effect after using different brands of soaps in the market.

The ambient temperature is 23.0°C and related humidity (RH) 49 %, three skin sites near the elbow on left forearm (A, B and C) were chosen on two volunteers

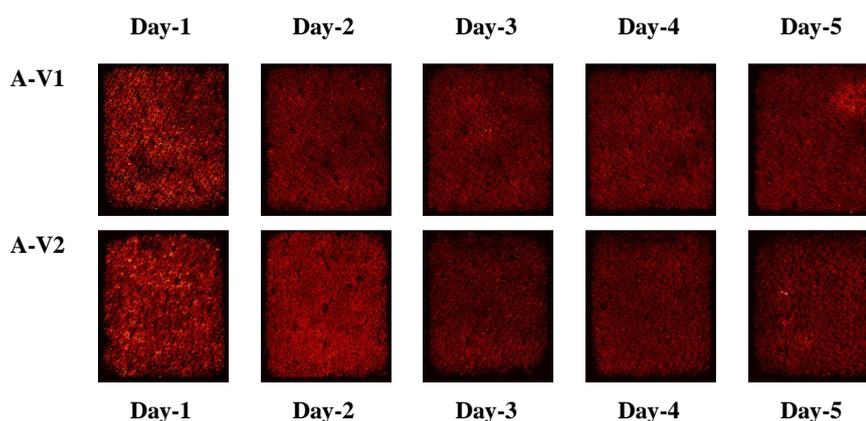
(V1 and V2, aged 20-30, male). Three brands of soaps were used for washing the three skin sites. There is need to wash each site for 3 minutes using rubber gloves. A new pair of gloves needs to be changed before every wash in order to avoid the cross contact with other brands of soaps. Repeat all the steps from day-1 to day-5.

For the hydration measurement, Epsilon needs to be fully contacted on the skin site for about 60 seconds. There is a need to mark the skin site as a 4x4 matrix in order to measure the average values of the full area of the skin site, as shown in Figure 5.27.

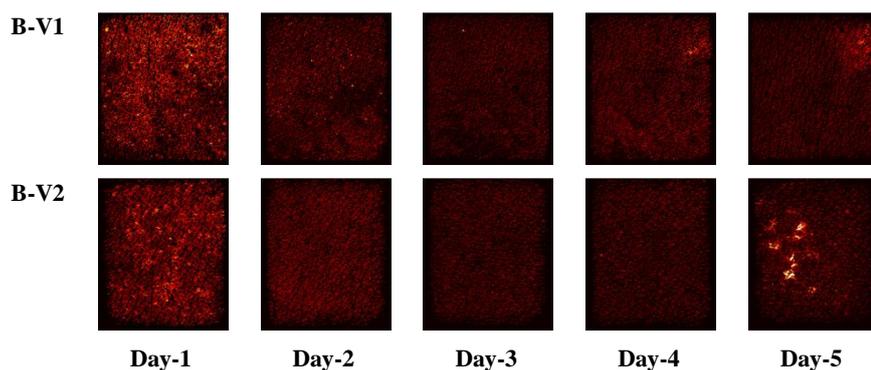


Figure 5.27 OTTER measurement matrix on skin site.

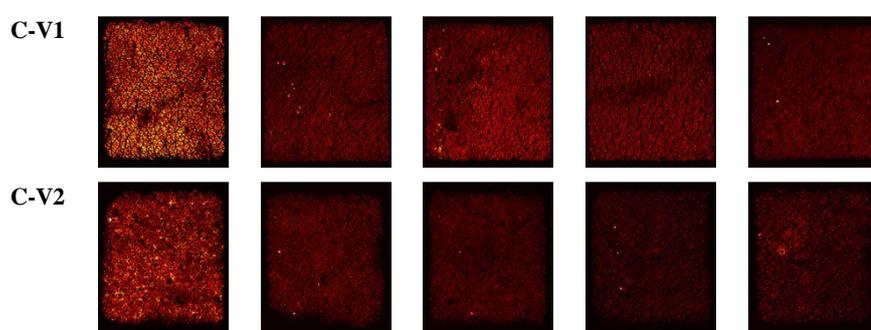
Figure 5.28 shows the 5-days image results from Epsilon.



(a) Hydration Images of soap A on two volunteers



(b) Hydration Images of soap B on two volunteers



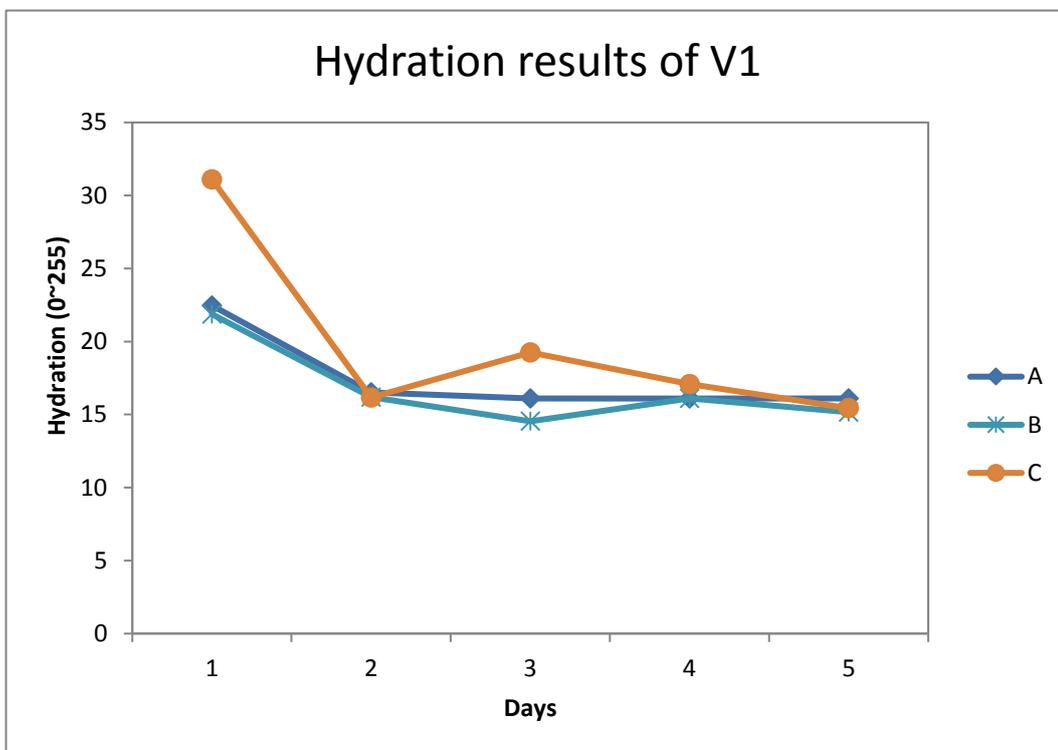
(c) Hydration Images of soap C on two volunteers

Figure 5.28 Hydration Images of three brand of soaps on two volunteers.

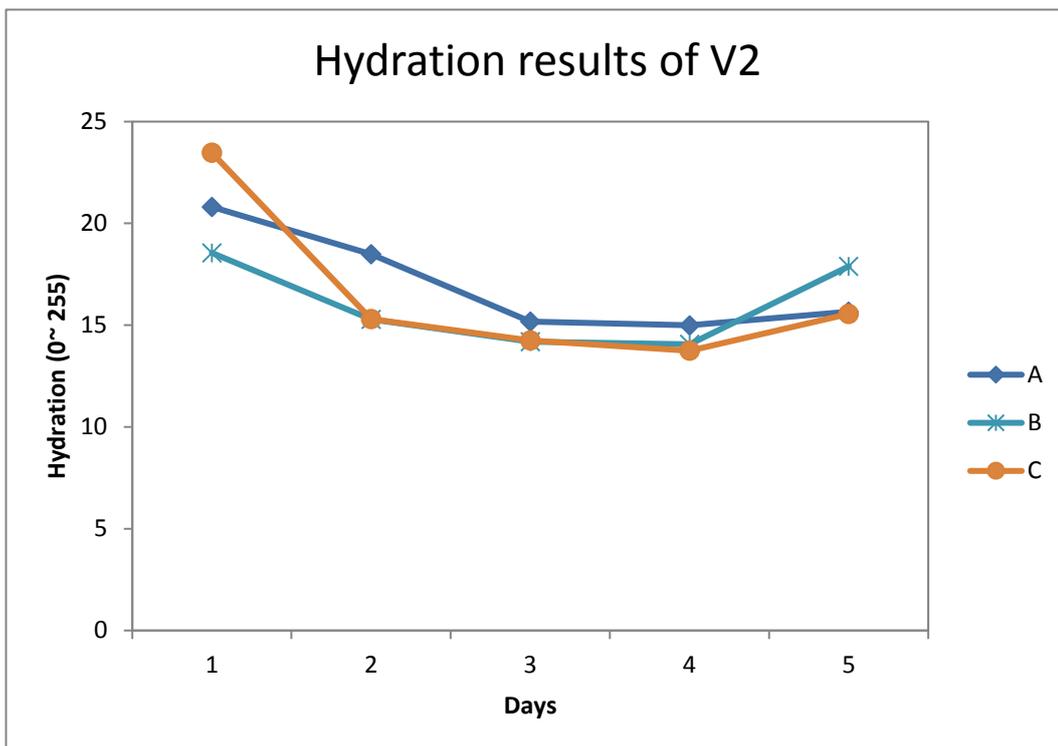
On day-1, all skin sites are relatively bright, showing a good water content, then as the washes go on, the skin images are getting darker and darker, indicating there is a consistent drying effect due to soap washing. On day-5, soap A shows the least drying, soap B shows the worse drying effect, and soap C is somewhere in the middle. For soap B on day -5, the bright spots on V2 are due to skin damage, where the barrier is gone, and underneath wet layers are shown.

The results show that capacitive imaging is a powerful tool for skin soap drying effect study. It can measure the subtle differences of the different soaps.

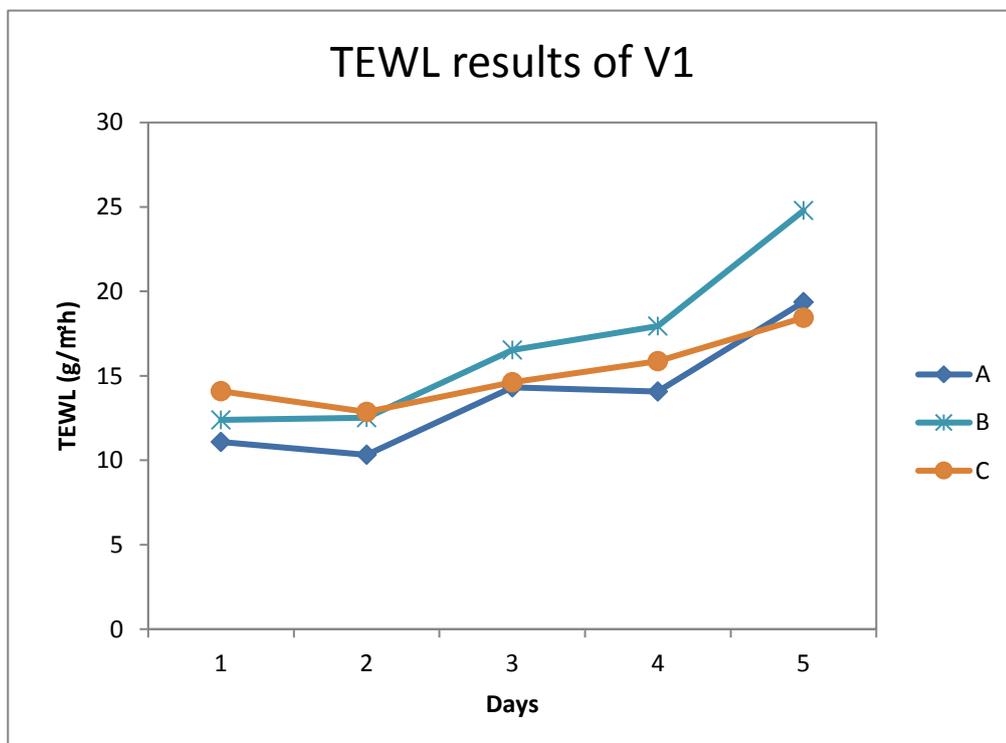
Figure 5.29 shows the corresponding hydration/TEWL results of two volunteers in 5-days measurement.



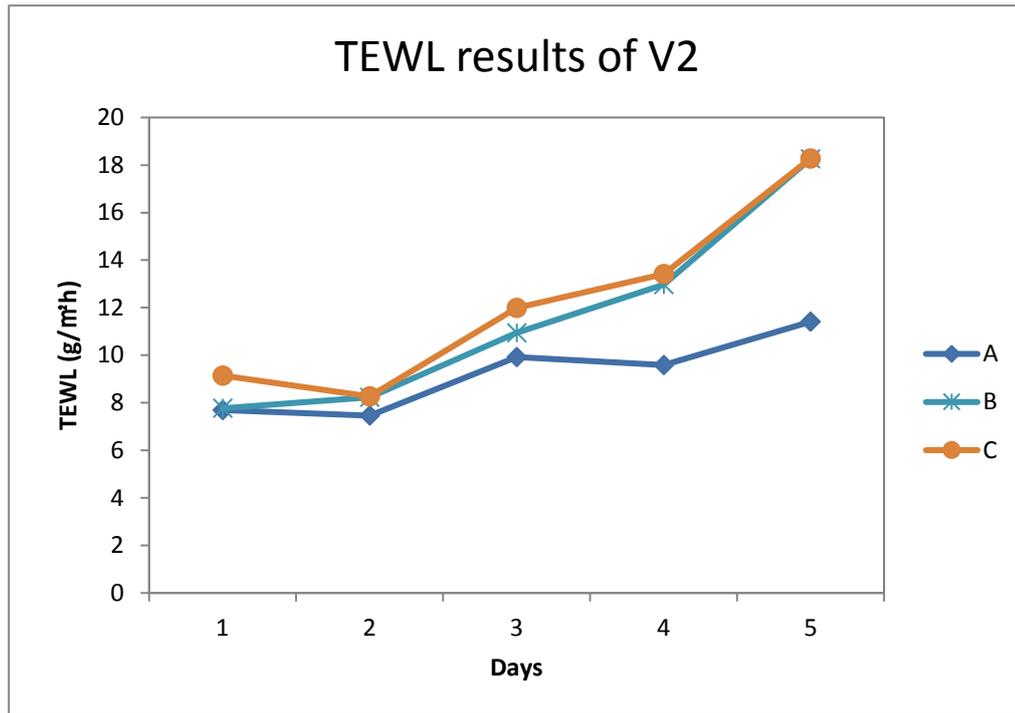
(a) Hydration results of V1



(b) Hydration results of V2



(c) TEWL results of V1



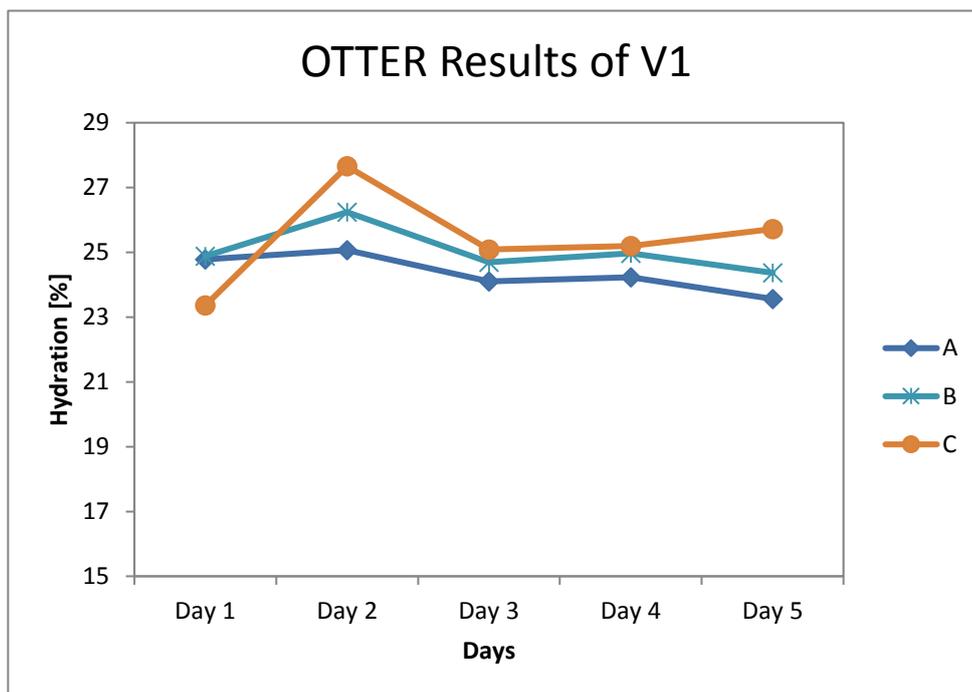
(d) TEWL results of V2

Figure 5.29 Line chart of the Hydration and TEWL results of three sites in 5-days measurement

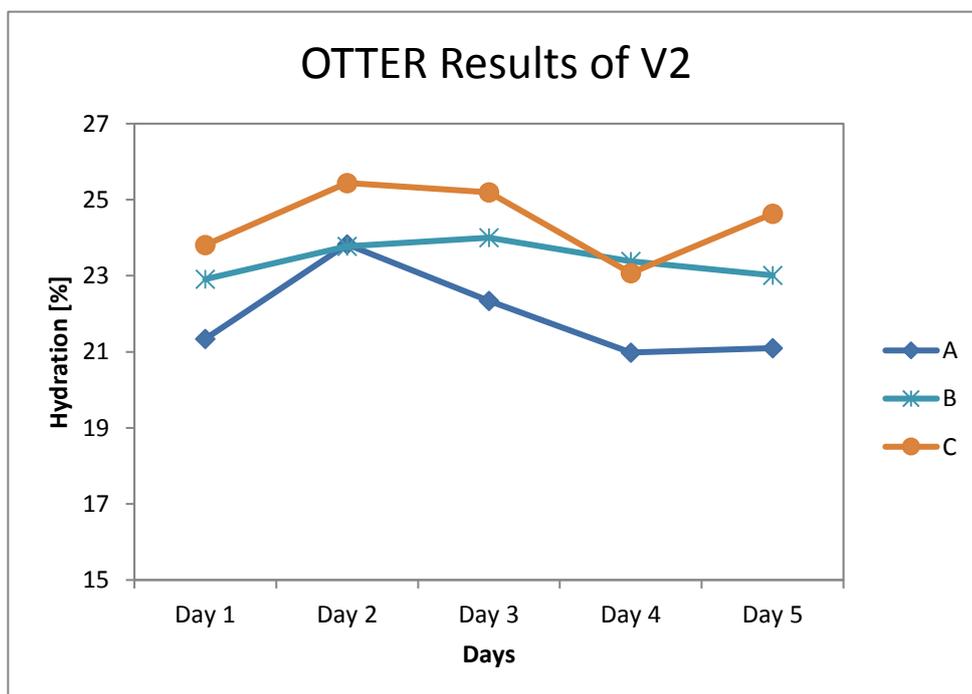
The hydration of site C on day-1 increased extremely high and then decreased rapidly on day-2. Additionally, V1 jumped up a little bit on day-3, and then decreased down to a lower level on day-5. The site B data of V2 returned to a higher level than the other on day-5.

According to the TEWL data in Figure 5.29(c) and (d), TEWL increased continually from day-1 to day-5 especially the site B in both V1 and V2. Site B and C of V2 has more TEWL than site A.

The Figure 5.30 shows the results of OTTER measurement.



(a) OTTER results of V1



(b) OTTER results of V2

Figure 5.30 Line Chart of OTTER results of two volunteers

The OTTER results show that the hydration data gradually goes down over the days, from day-1 to day-5. This agrees well with the other two measurements, indicating that skin is getting dryer during the course of soap washing. The initial increase in day-2 is likely due to the soap washing has left some residue on the skin. Therefore the effect of soap washing should only be observed from day-2 to day-5. The sudden jumps of soap C in day-5 in both volunteers are more likely due to the skin damage, as also shown in Figure 5.28.

By looking at the overall results, at day-5, the end of the study, soap C has the highest TEWL and lowest water content, therefore it damaged the skin the most, whilst soap A has the lowest TEWL and highest water content, therefore it damaged the skin the least. Soap B is somewhere in between.

The results show that the hydration and TEWL are important measurements for skin characteristics research. To identify the drying effect of the soaps, hydration and TEWL data need to be considered. By combining both hydration and TEWL results it is possible measure the soap washing effects even with subtle differences.

5.6 Summary

In summary, six types of experiments are presented in this chapter for developing new methodologies for skin characterization measurements by using OTTER, AquaFlux and Epsilon.

Epsilon is a useful tool for skin hydration measurement. It is also a small and easy instrument to operate. The results of the experiments in the first section show that it is not only a measuring tool, it also an imaging device for skin image capturing in order to study skin characteristics visually. Due to its imaging capabilities, the greyscale images measured in the experiments were clear and with high resolution

for further processing such as skin microrelief research and skin 3D profiling. The skin over-hydration experiment results (section 5.1) shows skin recovering function strength on different volunteers. It also shows a good correlation of TEWL and Hydration of the skin which can be used for skin damage and drying assessment.

The results show that the time-dependent occlusion curves of capacitive contact imaging can reflect the types of damages, and by analyzing the shapes of the curves we can get information about the skin surface water content level and stratum corneum thickness. The results also show that the combination of capacitive contact imaging and condenser-TEWL method gives extra information about the skin damage, such as the types of the damages and status of the damages.

The experiments mentioned in the last section was preformed drying effect assessment using OTTER, AquaFlux and Epsilon. The drying effect of three brands of soaps was assessed by the hydration and TEWL level measured by the three instruments. It shows the matched results which concluded by the Epsilon and AquaFlux. Additionally, OTTER has been proved that it has good performance in skin hydration measurement.

Chapter 6 Conclusions and Future Work

6.1 Conclusions

This thesis achieved the original aims of the research: new skin image processing and data analysis techniques for capacitive contact images from fingerprint sensor and images from digital camera; a novel graphic user interface (GUI) software for implement skin image processing techniques; in-vivo skin characterization measurements using Fingerprint sensor/Epsilon, AquaFlux and OTTER.

In the image processing work, a new GUI based MATLAB programme has been developed, which can be used for extracting and analysing images from the result files created by the measurement instruments. The programme implemented skin image processing techniques such as image enhancement (i.e. brightness equalization, skin texture extraction, hair removal), image stitching, image matching and skin surface 3D profiling etc. This has greatly enhanced the image processing functionalities and improved image processing efficiency. A new GUI using open source library - OpenCV - has also been developed. It can be used for real time video image processing including functions such as contours detections, texture extractions, and face detections.

In the skin characterisation work, six types of experiments were conducted: skin over-hydration experiments; skin damage assessment including intensive washing, tape stripping, and SLS irritation; dermabrasion experiments; soap drying effect assessment.

Skin Over-hydration experiment shows a good result of TEWL measurements and hydration measurements using capacitive contact imaging. An AquaFlux occlusion effect was discovered, and therefore care needs to be taken in image analysis in order to get more accurate results.

The skin damage assessment experiments also show very interesting results. The time dependent capacitive contact imaging occlusion curve shows good potential as a new methodology for skin damage assessment. It can not only reflect the scale of the damage, but also possibly reflect the types of damages.

Dermabrasion is a treatment to remove dead skin on outer layer to further improve the skin status such as the scar, dry skin scale, heavy texture and wrinkle. The results show that hydration and TEWL measurements could be used to identify the effect of the dermabrasion. The removed skin thickness has been calculated.

Soap drying effect assessment presents the experiment of assessing soap drying effect by using Epsilon, AquaFlux and OTTER. The results show that soap washing affects both the hydration and TEWL status of skin. Therefore in order to assess the soap drying effect both the hydration results and TEWL results should be considered. The Epsilon, AquaFlux and OTTER results generally agreed with each other, and can differentiate the differences of different soaps. The capacitive contact imaging shows the most interesting results, as it can visualize the changes caused by the soap washing.

6.2 Future Works

For the future work, the following are suggested:

A: Improvements for Capacitive Contact Imaging Measurement

It is not easy to measure the skin hydration without a consistent contact pressure for each measurement. Although the device is spring loaded, it is still difficult to maintain the exact same contact pressure. Sometimes the contact pressure is a little bit too strong, but sometimes is a little bit too weak, which makes the result unreliable. To reduce this influence, a force sensitive resistor sensor can be

implemented into the device, by calibrating the results against contact forces, the measurement results can be corrected against the contact forces.

B: Improvement for AquaFlux measurement

One of the problems is the dark ring on the skin over-hydration images captured by the fingerprint sensor. It is unavoidable that the AquaFlux probe head left the ring mark on the skin. It is not obvious though naked eyes, but it causes a problem in hydration measurement due to the large measurement area of the fingerprint sensor. To minimize this problem, different sizes and different shapes (such as square or rectangle) of AquaFlux measurement caps are recommended. A shorter measurement time, using fast predict algorithm, could also be very useful.

C: New challenge for skin image processing with OpenCV

In this project, two sets of skin imaging processing software programmes have been developed by using MATLAB and the open source OpenCV library separately. Through dynamic-link library (DLL) is possible to merge the two programmes into one, i.e. using C++ GUI programme to call MATLAB DLLs.

The real-time video image processing functionality can also be implemented in Epsilon to analyse the capacitive contact images.

D: Automatic Facial Skin Analysis

With the development of face detection, now it is possible to identify the different locations of the facial skin, and analyse them automatically. Facial skin has a great cosmetic importance. The facial skin automatic analysis can have huge potential for commercial applications.

References

Ablett S, Lillford P (1991), "Water in foods", Chem. Br. 27, 1024, 1991.

Abu. S. (2012), "Bandit Detection using Colour Detection Method", Procedia Engineering29, 1259–1263.

Abubaker. A.; Qahwaji. R.; Ipson. S.; Saleh. M. (2007). "One Scan Connected Component Labelling Technique". Signal Processing and Communications, 2007. ICSPC 2007. IEEE International Conference: 1283. doi: 10. 1109/ ICSPC. 2007. 4728561. ISBN 978-1-4244-1235-8.

Anderson, R.L, Cassidy, J.M., Hansen, J.R. and Yellin. W. (1973), "Hydration of Stratum Corneum. Biopolymers", 12, pp. 2789-2802.

AquaFlux, <http://www.biox.biz/Products/AquaFlux/AF200Page01.php> [access date: 12/ 09/ 2013]

Berardesca. E., Elsner. P., Wilhelm. K.P., Maibach. H. (1995), "Bioengineering of the Skin: Methods and Instrumentation", CRC press Inc.

Bevilacqua. A., Gherardi. A. (2008), Characterization of a capacitive imaging system for skin surface analysis, First International Workshops on Image Processing Theory, Tools& Applications, 23-26.

Baker. H. A., Kligmann. M., (1967), "Measurement of Trans epidermal Water Loss by Electrical Hygrometry. Instrumentation and Responses to Physical and Chemical Insults", Arch. Dermatol., 96, 441.

Barry, B.W (1983), "Dermatological Formulations (Percutaneous Absorption)", New York: Taylor & Francis.

Batisse. D., Giron. F., and Leveque. J.L. (2006). Capacitance imaging of the skin surface.

Blank. I.H. and Finesinger. J.E. (1946). Electrical resistance of the skin. Archives of Neurology and Psychiatry, 56, 544-557.

Blank. I.H. (1953), "Further Observations on Factors Which Influence the Water Content of Stratum Corneum", Invest. Dermatol, 21, 259.

Blank. I.H., Moloney. J., Emslie. A.G., Simon. I., Apt. C. (1984), "The diffusion of water across the stratum corneum as a function of its water content", J. Invest. Dermatol., 82, pp. 188-194.

Bonsor. K. (2008). "How Facial Recognition Systems Work". Retrieved 2008-06-02.

Bouwstra. J.A., Gooris. G.S., Sakmomns-de Vries. M.A., Van der spek. J.A., Bras W. (1992), "Structree if human stratum corneum as a function of temperature and hydration: A wide-angle X-ray diffraction study". International Journal of Pharmaceutics, 84 205-216.

Breszcz. M.; Breckon. T.P. (2015). "Real-time Construction and Visualization of Drift-Free Video Mosaics from Unconstrained Camera Motion" (PDF). IET J. Engineering. IET. 2015 (16): 1–12.

Cavitch. S. M. (1994). The Natural Soap Book. Storey Publishing. ISBN 0-88266-888-9.

Chang. T. (1980), "Texture Analysis of Digitized Fingerprints for Singularity Detection," Proc. Fifth ICPR, pp.478-480.

Charles. A. P. (2003). Digital Video and HDTV: Algorithms and Interfaces. Morgan Kaufmann.

Ciortea. L.I., (2004), "Development of Water Migration Measurement Techniques in Soft and Non-Compact Materials", PhD Thesis.

Cornwell. P.A., Barry. B.W., Stoddyart. C.P., Bouwstra. J.A. (1994), "Wide-angle X-ry diffraction of Human Stratum Corneum: Effects of Hydration and Terpene Enhancer Treatment". J. Pharm. Pharmacol., 46 938-950.

Crawford, Mark (2011). "Facial recognition progress report". SPIE Newsroom. Retrieved 2011-10-06.

CTPA (2014), "The Cosmetic, Toiletry & Perfumery Association".

[<http://www.ctpa.org.uk/>, Assess date: July 2014]

Daugman. J.G. (1985), "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two dimensional visual cortical filters", Journal of The Optical Society of America, 2(7), pp.1160-1169.

David. J., Anneken, Both. S., Christoph. R., Fieg. G., Steinberner. U., Westfechtel. A. (2006) "Fatty Acids" in Ullmann's Encyclopedia of Industrial Chemistry, Wiley-VCH, Weinheim.

Devi. K., Dimple (2015), "Scope of Face Recognition Techniques", International Journal of Computer Science and Information Technology Research ISSN 2348-120X, Vol. 3, Issue 2, pp: (892-896), Month: April - June 2015.

David. Z. (2008), "window manager Definition". PC Magazine. Ziff Davis Publishing Holdings Inc. Retrieved 12 November 2008.

Denda. M. (2000), "Influence of dry environment on epidermal function", Journal of Dermatological Science, pl.1S22–S28.

Deng. Y. (1999), "A Region Representation for Image and Video Retrieval". PhD thesis, University of California, Santa Barbara.

Deepak Prasanna.R. (2012), "Enhancement of vein patterns in hand image for biometric and biomedical application using various image enhancement techniques", Procedia Engineering 381174–1185.

Diego A.S. and Selinger A., (2004). "Thermal face recognition in an operational scenario". In Proceedings of the 2004 IEEE computer society conference on Computer vision and pattern recognition (CVPR'04). IEEE Computer Society, Washington, DC, USA, 1012-1019.

Duce. S.L., Carpenter. T.A., Hall. L.D., Hills. B.P. (1992), "An investigation of the origins of contrast in NMR spin echo images of plant tissue", Magn. Reson. Imaging, 10,289.

Dimai. A. (1999), "Rotation Invariant Texture Description using General Moment Invariants and Gabor Filters". Proc. Of the 11th Scandinavian Conf. on Image Analysis. I, 6, pp.391-398,

Extraction, Computer Systems& Applications, (2010), 19(6), pp. 195-198.

Eisner. P.E., Berardesca. H.I., Maibach (eds), (1994), Bioengineering of the Skin: Water and the Stratum Corneum, pp.60-64, CRC.Press, BocaRaton.

Edelberg. R. (1966). Response of cutaneous water barrier to ideational stimulation: a GSR component. *Journal of Comparative and Physiological Psychology*, 61, 28-33.

Edelberg. R. (1967), Electrical properties of the skin. In: Brown,C.C. (Ed.) *Methods in Psychophysiology*. Williams and Wilkins: Baltimore, 1-53.

Edelberg. R. (1968). Bio potentials from the skin surface: the hydration effect. *Annals of the New York Academy of Sciences*, 148, 252-262.

Edelberg. R. (1970). The information content of the recovery limb of the electro dermal response. *Psychophysiology*, 6, 527-539.

Edelberg. R. (1971). *Electrical and Biophysical Properties of the Skin*. Vol. I. Wiley: New York, 513-550.

Elias, P.M., 1991. Epidermal barrier function: intercellular lamellar lipid structures, origin, composition and metabolism. *Journal of Controlled Release*, 15. pp199-208.

Fischer, S (2001), "Minimum entropy transform using Gabor wavelets for image compression", 11th International Conference on Image Analysis and Processing and Proceedings, 9,428 – 433, 2001.

Fisher. R., Perkins. S., Walker. A. and Wolfart. E. (2000), *Hypermedia Image Processing Reference (HIPR2)*.

Morano. R. (2002) "Structured Light Using Pseudorandom Codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Filler, Aaron (2009). "The History, Development and Impact of Computed Imaging in Neurological Diagnosis and Neurosurgery: CT, MRI, and DTI". *Nature Precedings*. doi:10.1038/npre.2009.3267. ISSN 1756-0357.

Fluhr, J.P., Eisner, E., Berardesca, and H.I. Maibach (eds) (2005), *Bioengineering of the Skin, Water and the Stratum Corneum*, 2nd ed., pp 105-110, CRC Press, Boca Raton.

Foreman, M.I. (1976), "A Proton Magnetic Resonance Study of Water in Human Stratum Corneum", *Biochem. Biophys. Acta.* 437, 599.

Franconi, F., Akoka, S., Guesnet, J., Baret, J.M., Dersigny, D., Breda, B., Muller, C., Beau P (1995), "Measurement of epidermal moisture content by magnetic resonance imaging assessment of a hydration". *British Journal of Dermatology*, 132 913-917.

Frodin, T., Helander, P., Molin, L., Skogh, M. (1988), "Hydration of Human stratum corneum studied in-vivo by optothermal infrared spectrometry, electrical capacitance measurement, and evaporimetry". *Acta Derm Venereol (Stikh)*, 68 461-467.

Ghosh, A. (2011), "Assessment of Heat Affected Zone of Submerged Arc Welding Process through Digital Image Processing", *Procedia Engineering* 10, 2782-2785. *Geoscience Electronics, IEEE Transactions on.* 11(4), pp. 171-177.

Gloor, M., Heymann, B., Stuhler, T. (1981), "Infrared-Spectroscopic Determination of the Water content of the Horny layer in Healthy subjects and in patients suffering from atopic dermatitis". *Arch Dermatol Resm* 271, 429-436.

Gloor, M., Willebrandt, U., Thomer, G., Kupferschmid (1980), "Water content of the Horny Layer and Skin Surface lipids". *Arch Dermatol*, 268, 221-223.

Goering, Richard (2004). "MATLAB edges closer to electronic design automation world". EE Times.

Hugo. G., Espinosa, (2013), "Sports monitoring data and video interface using a GUI auto generation MATLAB tool", *Procedia Engineering*60,243–248.

Hussin. R. (2012), "Digital Image Processing Techniques for Object Detection from Complex Background Image", *Procedia Engineering*41,340–344.

Haralick. R.M. (1973). *Computer Classification of Reservoir Sandstones*.

Huang. T.H. (2011), "A High-Performance Skin Segmentation Method", *Procedia Engineering* 15,608–612.

Heinrich, U., U. Koop, et al.(2003): Multicentre comparison of skin hydration interms of physical-,physiological-and product-dependent parameters by the capacitance method (CorneometerCM825), *International Journal of Cosmetic Science* ,25,45-51.

Imhof. R.E. (2011),"AquaFlux Familiarisation", LSBU and Biox Limited, presentation material.

Imhof RE, Birch DJS, Thornley FR, Gilchrist JR and strivens TA, (1984),"Opto-thermal transient emission radiometry", *J.Phys.E: Sci. Instrum*.

Imhof RE, O'Driscoll Don, XiaoP and Berg E, (1998)," New Method of Measuring Trans-epidermal Water Loss", Internal Report based on a Poster Presentation at the SCII Conference, Cardiff.

Imhof, R.E., O' Driscoll. D., Xiao, P. and Berg, E (1999), "New sensor for water vapour flux", *Sensors and their Applications X*, pp. 173-177, IoP Publishing.

Imhof. R. E. and Xiao. P., Biox Epsilon - A New Permittivity Imaging System, Exhibitors' Workshop Presentation, ISBS World Congress, Copenhagen 2012.

Imhof. R.E., Zhang. B. and Birch. D.J.S., (1994), "Progress in Photothermal and Photoacoustic Science and Technology. ed Mandelis A (Englewood Cliffs USA: PTR Prentice Hall), pp.185-236.

Imhof. R.E, Zhang. B. and Birch. D.J.S. (1994), "Photothermal radiometry for NDE", in Mandelis A(ed): Progress in Photothermal and Photoacoustic Science and Technology. PTR Prentice Hall, Englewood Cliffs (USA), vol. II.

IUPAC (1997). "IUPAC Gold Book – soap" Compendium of Chemical Terminology, 2nd ed. (the "Gold Book"). Compiled by A. D. McNaught and A. Wilkinson. Blackwell Scientific Publications, Oxford.

Jacques. S.L. (1979), "A Linear Measurement of the Water Content of Human Skin Using a Microwave Probe", Proc. IEEE-EMBS Conf, Denver Colorado, 1.

Joseph P. Hornak (2002), Encyclopedia of Imaging Science and Technology.

Junqueira. L.C., Carneiro. J. (2009), "Basic Histology", 12th edition, Prentice Hall, USA.

Kai Liu; Yongchang Wang; Daniel L. Lau; Qi Hao; Laurence G. Hassebrook (2010). "Dual-frequency pattern scheme for high-speed 3-D shape measurement" (PDF). Optics Express. 18 (5): 5229–5244.

Kamarainen. J.K., Kyrki. V. (2006), "Invariance Properties of Gabor Filter-Based Features- Overview and Applications", IEEE Transactions on Image Processing.

Kavitha, A.R. (2010), "Implementation of Grey-level Clustering Algorithm for Image Segmentation", *Procedia Computer Science* 2, 314–320.

Kasting. G.B. and N.D. Barai (2003),"Equilibrium Water Sorption in Human Stratum Corneum", *J. Pharm. Sci.*, vol92, pp1624-1631.

Kirk, R.E. (1968). *Experimental Design: Procedures for the Behavioural Sciences*. Brooks Cole: Belmont, California.

Kimmel, R., (2005). "Three-dimensional face recognition"(PDF). Retrieved 2005-01-01.

Khan. M-H.M. (2014), "Analyzing Factors Affecting Hand Biometrics during Image Capture", *Procedia Computer Science* 32521–528.

Kinnaree. P. (2011), "RGB Colour Correlation Index for Image Retrieval", *Procedia Engineering* 8,36–41.

Kruizinga. P., Petkov. N., and Grigorescu. S.E. (1999), "Comparison of texture features based on Gabor filters", *Proceedings of the 10th International Conference on Image Analysis and Processing*, pp.142-147.

Larkins. B.D. (2012)," Introductory computational science using MATLAB and image processing", *Procedia Computer Science* 1,913–919.

Leveque. J.L..and Querleux. B. (2003), *Skin Chip a new tool for investigating the skin surface in-vivo*. *Skin Research and Technology* 9,343-347.

Lindeberg. T. (1998) "Edge detection and ridge detection with automatic scale selection", *International Journal of Computer Vision*, 30, 2, pages 117–154.

Liu. Z. (2012), "An Improved Face Recognition Method Based on Gabor Wavelet Transform and SVM", Fifth International Symposium on Computational Intelligence and Design (ISCID), 10, pp.378– 381.

Lykken. D.T., (1968), "euro psychology and psychophysiology in personality research." PartII. Psychophysiological techniques and personality theory. In: Borgatta. E.F. and Lambert. W.W. (Eds.) Handbook of Personality Theory and Research. Rand McNally:Chicago, 454509.

Mamishev A., Xiaobei Li, Kishore Sundara Rajan, (2002), "Measurement of Moisture Distribution in Organic Materials, Sensors", Energy and Automation Laboratory Research Paper, University of Washington.

Manenti; Carlani, M; Mancino, S; Colangelo, V; Di Roma, M; Squillaci, E; Simonetti, G (2007). "Diffusion tensor magnetic resonance imaging of prostatecancer". Investigative radiology. 42(6): 412–9.doi: 10.1097/01. rli. 0000264059. 46444. bf. PMID 17507813.

Martinsen. G.S., Grimnes. J.K., Nilsen. C., Tronstad. W. Yang, H. Kim and K. Shin, (2008), "Gravimetric Method for in-vitro Calibration of Skin Hydration Measurements", IEEE Trans. Biomed. Eng.,55(2),728.

Munsters. J. (2012), "Skin conductance measurements as pain assessment in new born infants born at 22–27 weeks gestational age at different postnatalage", Early Human Development, Volume 88, Issue1, Pages21-26.

Marčelja. S. (1980)." Mathematical description of the responses of simple cortical cells". Journal of the Optical Society of America. 70 (11): 1297–1300.

Nilsson. G.E. (1997), "Measurement of Water Exchange Through Skin" *Med. Biol Eng. Comput.*, 15, 209.

Moeslund. T. (2009, March 23). Canny Edge Detection.

Nuutinen. J., Alanen. E., Autio. P., Lahtinen. M.R., Harvima. I. and Lahtinen. T. (2003), "A Closed unventilated chamber for the measurement of transepidermal water loss", *Skin Research and Technology*, 9, pp. 85-89.

Pattern Analysis and Machine Intelligence, *IEEE Transactions on. PAMI-2(3)*, pp. 204–222.

Pablo d'Angelo (2007). "Radiometric alignment and vignetting calibration" (PDF).

Pinnagoda. J., R.A. Tupker, P.J. Coenraads, and J.P. Nater, (1981), "Transepidermal Water Loss: With and Without Sweat Gland Inactivation", *Contact Dermatitis*, 21,16.

Pinnagoda, J.R.A. Tupker,T Agner, and J.Serup,(1990)," Guidelines for Transepidermal Water Loss (TEWL) Measurement",*Contact Dermatitis*, 22,164.

Potts. R.O., D.B. Guzek, R.R. Harris and J.E. McKie (1985),"A Non-invasive in vivo Technique to Quantitatively Measure Water Concentration of the Stratum Corneum Using Attenuated Total Reflectance Infrared Spectroscopy", *Arch. Dermatol Res.*,277(6),489.

Potts. R.O. (1985), "In-vivo measurement of water content of the stratum corneum using infrared spectroscopy: A review". *Cosmetics and Toiletries*, 100, 27-31.

Pulli. Kari; Baksheev, Anatoly; Korniyakov, Kirill; Eruhimov, Victor (2012). "Realtime Computer Vision with OpenCV". *Queue*. pp. 40:40–40:56.

Rafael C. Gonzalez; Richard E. Woods (2008). Digital Image Processing. Prentice Hall. pp. 1–3.

R. Brunelli and T. Poggio (1993), "Face Recognition: Features versus Templates", IEEE Trans. on PAMI, (15)10:1042-1052

R. Brunelli (2009), Template Matching Techniques in Computer Vision: Theory and Practice, Wiley, ISBN 978-0-470-51706-2.

R. Fisher; S. Perkins; A. Walker; E. Wolfart (2003). "Connected Component Labeling".

Scharr, Hanno (2000), Dissertation (in German), Optimal Operators in Digital Image Processing.

Scott, R.C., G.J. A Oliver, PH. Dugard, and H.J. Singh, (1982), "A Comparison of Techniques for the Measurement of Transepidermal Water Loss", Arch. Derm. Res., 274, 57-64.

Simon. I., Emslie. A.G., Apt C.M., Blank I.H., Andersin R.R. (1979), "Determination of in-vivo water concentration profile in human stratum corneum by a photo acoustic method". IEEE Transactions on Biomedical Engineering. BME-26, 187-195.

Singh. H., Xiao. P., Berg. E.P., and Imhof. R.E. (2008), Skin Capacitance Imaging for Surface Profiles and Dynamic Water Concentration Measurements, ISBS Conference, Seoul, Korea.

Smith, J.R (1997), "Integrated Spatial and Feature Image System: Retrieval, Analysis and Compression". Ph.D thesis, Columbia University.

Smith, Kelly (2008), "Face Recognition" (PDF). Retrieved 2008-06-04.

Song Zhang; Peisen Huang (2006). "High-resolution, real-time 3-D shape measurement". *Optical Engineering*: 123601.

Song Zhang; Daniel van der Weide; James H. Oliver (2010). " Superfast phase-shifting method for 3-D shape measurement". *Optics Express*. 18:9684–9689.doi:10.1364/OE.18.009684.

Spencer. T.S., Linamen. C.E., Akers. W.A., Jones. H.E. (1975), "Temperature dependence of water content of stratum corneum. *Br J Dermatol*", 93, pp. 159-164.

Steven. M., (2000), *Water Vapor Myths: A Brief Tutorial*.

Steve Mann and R. W. Picard (1994) "Virtual bellows: constructing high-quality images from video.", In *Proceedings of the IEEE First International Conference on Image Processing Austin, Texas, November 13–16*.

Steve Mann (1993). "Compositing Multiple Pictures of the Same Scene", *Proceedings of the 46th Annual Imaging Science & Technology Conference, May 9–14, Cambridge, Massachusetts*.

Swanbeck. G., (1959), "Macromolecular organization of epidermal keratin: an X-ray diffraction study of the horny layer from normal, ichthyotic and psoriatic skin", *Acta Derm Venereol*. 39, 5-37..

Swarbrick. J. and Boylan (1995), "Permeation Enhancement through Skin", *Encyclopedia of Pharmaceutical Technology*, 11, pp. 449-492.

Szeliski, Richard (2005). "Image Alignment and Stitching" (PDF). Retrieved 2008-06-01.

S. Suen; E. Lam; K. Wong (2007). "Photographic stitching with optimized object and colour matching based on image derivatives". *Optics Express*. 15 (12): 7689–7696. doi:10.1364/OE.15.007689. PMID 19547097.

Tagami. H. (1994),” Quantitative Measurements of water concentration of the stratum corneum In vivo by high frequency current”. *Acta Derm Venereol* (Stockh), 74 29-33.

Tcheslavski, Gleb. V. (2010). "Morphological Image Processing: Grey-scale morphology" (PDF). Retrieved 4 November 2013.

Thangairulappan. K., Jeyasingh. J.B.V. (2012), “Face Representation Using Combined Method of Gabor Filters, Wavelet Transformation and DCV and Recognition Using RBF”, *Journal of Intelligent Learning Systems and Applications*, 4, pp.266-273.

Tsai. J., Weiner. N.D., Flynn. G.L. and Ferry. J. (1991), “Properties of adhesive tapes used for stratum corneum stripping”, *International Journal of Pharmaceutics*, 72, pp. 277 - 231.

Umbaugh, Scott E (2010). *Digital image processing and analysis: human and computer vision applications with CVIP tools* (2nd ed.). Boca Raton, FL: CRC Press. ISBN 978-1-4398-0205-2.

Vincent, Luc; Soille, Pierre (June 1991). "Watersheds in digital spaces: an efficient algorithm based on immersion simulations". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 13 (6): 583. doi:10.1109/34.87344.

William, A.C., Barry, B.W (1992), “Skin Absorption Enhancers”, *Critical Reviews in Therapeutic Drug Carrier Systems*, 9(3,4), pp. 305-353.

Williams, Mark (2008). "Better Face-Recognition Software". Retrieved, 2008-06-02.

Xiao. P., Singh. H., Zheng. X., Berg. E.P., and Imhof. R.E. (2007), In-vivo Skin Imaging for Hydration and Micro Relief Measurements, Stratum Corneum V conference, July11-13, Cardiff, UK.

Xiao. P, Lane. M.E. (2012), Membrane Solvent Penetration Measurement using Contact Imaging, Stratum Corneum VII Conference, Cardiff, UK.

Xiao. P., Imhof. R.E. (1998), "Opto-thermal measurement of water distribution within the stratum corneum", Skin Bioengineering Techniques and Applications in Dermatology and Cosmetology, CurrProbl Dermatol, Basel: Karger, 26, pp.48-60.

Xiao. P., Ou. P., Ciortea. L.I., Berg. E.P., and Imhof. R.E. (2012), "In-vivo Skin Solvent Penetration Measurements Using Opto-thermal Radiometry and Fingerprint Sensor", International Journal of Thermophysics 33: 1787-1794, DOI10.1007/s10765-012-1318-6.

Xiao. P., R.E. Imhof (1998), "Opto-Thermal Skin Water Concentration Gradient Measurement", School of EE&IE, South Bank University.

Li, Xiaobei, (2002),

<https://www2.ee.washington.edu/research/seal/projects/moisture%20sensing/project%20overview.htm> (accessed July, 2017)

Yajun Wang; Song Zhang (2011). "Superfast multifrequency phase-shifting technique with optimal pulse width modulation". Optics Express. 19: 9684–9689.doi:10.1364/OE.19.005149.

Zhang. X., Bontozoglou. C., Xiao. P. (2016), "Capacitive Imaging for Skin Characterization and Solvent Penetration", Skin Forum 16th Annual Meeting 21 - 22 June 2016, UCL, London, UK.

Publications

1. X. Ou, W. Pan, X. Zhang, P. Xiao, Skin image retrieval using Gabor wavelet texture feature, *International Journal of Cosmetic Science*, 01 Jan 2016. DOI: 10.1111/ics.12332
2. W. Pan, X. Zhang, M.E. Lane and P. Xiao, "The Occlusion Effects in Capacitive Contact Imaging for In-vivo Skin Damage Assessments", *International Journal of Cosmetic Science*, 2015, 37, 395–400, doi: 10.1111/ics.12209.
3. W. Pan, X. Zhang, E. Chirikhina, C. Bontozoglou and P. Xiao, Measurement of Skin Hydration with a Permittivity Contact Imaging System, *IFSCC Conference 2015*, Zürich, Switzerland, 21–23, September, 2015.
4. W. Pan, X. Ou, X. Zhang and P. Xiao, "Skin Damage Assessment by Using Capacitive Imaging and Condenser-TEWL Method", *Skin Forum 14th Annual Meeting* 4 to 5 September 2014, Prague, Czech Republic.
5. W. Pan, X. Ou, X. Zhang and P. Xiao, "Skin Damage Assessment by Using Capacitive Imaging and Condenser-TEWL Method", *Perspectives in Percutaneous Penetration Conference*, 23 - 25 April 2014, La Grande Motte, France.
6. W. Pan, X. Zhang, E. Chirikhina and P. Xiao, "Skin Hydration Measurement Using Contact Imaging", *SCC Showcase*, New York, December 11 - 12, 2014.
7. P. Xiao, W. Pan and X. Ou, "Digital Imaging Processing for in-vivo Skin Measurements", *Intelligent Signal Processing (ISP) Conference 2013*,

DOI: 10.1049/cp.2013.2044, London, 2-3 December 2013.

8. X. Ou, W. Pan and P. Xiao, In-Vivo Skin Capacitive Imaging Analysis by using Grey Level Co-occurrence Matrix (GLCM), *International Journal of Pharmaceutics*, Volume 460, Issues 1–2, 2 January 2014, Pages 28–32, ISSN 0378-5173, <http://dx.doi.org/10.1016/j.ijpharm.2013.10.024>.
9. P. Xiao, W. Pan, X. Ou, “Digital Imaging Processing for in-vivo Skin Measurements”, *International Journal of Cosmetic Science*, Volume 34, 2012, Pages 387-387.

Appendix

QT GUI programme 1

```
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/core/core.hpp>
#include <opencv2/objdetect/objdetect.hpp>
#include "webcam_watershed.h"
#include "ui_webcam_watershed.h"
#include <iostream>
#include <sstream>
#include <string>
using namespace cv;
webcam_watershed::webcam_watershed(QWidget *parent) :
    QWidget(parent),
    ui(new Ui::webcam_watershed)
{
    ui->setupUi(this);
    timer = new QTimer(this);
    connect(timer, SIGNAL(timeout()), this, SLOT(display()));
    if(cap.open(0))
    {
        frame = Mat::zeros(cap.get(CV_CAP_PROP_FRAME_HEIGHT),
cap.get(CV_CAP_PROP_FRAME_WIDTH), CV_8UC3);
        timer->start(33);
    }
    CascadeClassifier cascade, nestedCascade;
    connect(ui->pushButton, SIGNAL(clicked()), this, SLOT(opencam()));
    connect(ui->pushButton_2, SIGNAL(clicked()), this, SLOT(display()));
}

webcam_watershed::~webcam_watershed()
{
    delete ui;
}

 QImage Mat2QImage(const cv::Mat& mat)
```

```
{
    if(mat.type() == CV_8UC1)
    {
        QImage image(mat.cols, mat.rows, QImage::Format_Indexed8);
        image.setColorCount(256);
        for(int i = 0; i < 256; i++)
        {
            image.setColor(i, qRgb(i, i, i));
        }
        uchar *pSrc = mat.data;
        for(int row = 0; row < mat.rows; row ++)
        {
            uchar *pDest = image.scanLine(row);
            memcpy(pDest, pSrc, mat.cols);
            pSrc += mat.step;
        }
        return image;
    }
    else if(mat.type() == CV_8UC3)
    {
        const uchar *pSrc = (const uchar*)mat.data;
        QImage image(pSrc, mat.cols, mat.rows, mat.step, QImage::Format_RGB888);
        return image.rgbSwapped();
    }
    else if(mat.type() == CV_8UC4)
    {
        const uchar *pSrc = (const uchar*)mat.data;
        QImage image(pSrc, mat.cols, mat.rows, mat.step, QImage::Format_ARGB32);
        return image.copy();
    }
    else
    {
        return QImage();
    }
}

cv::Mat QImage2cvMat(QImage image)
{
    cv::Mat mat;
    switch(image.format())
    {
        case QImage::Format_ARGB32:
```

```
    case QImage::Format_RGB32:
    case QImage::Format_ARGB32_Premultiplied:
        mat = cv::Mat(image.height(), image.width(), CV_8UC4, (void*)image.constBits(),
image.bytesPerLine());
        break;
    case QImage::Format_RGB888:
        mat = cv::Mat(image.height(), image.width(), CV_8UC3, (void*)image.constBits(),
image.bytesPerLine());
        cv::cvtColor(mat, mat, CV_BGR2RGB);
        break;
    case QImage::Format_Indexed8:
        mat = cv::Mat(image.height(), image.width(), CV_8UC1, (void*)image.constBits(),
image.bytesPerLine());
        break;
    }
    return mat;
}
int i = 0;
double t = 0;
vector<Rect> faces, faces2;
const static Scalar colors[] = { CV_RGB(0,0,255),
    CV_RGB(0,128,255),
    CV_RGB(0,255,255),
    CV_RGB(0,255,0),
    CV_RGB(255,128,0),
    CV_RGB(255,255,0),
    CV_RGB(255,0,0),
    CV_RGB(255,0,255) } ;
vector<Rect> nestedObjects;
Point center;
Scalar color = colors[i%8];
int radius;
double scale=5;
using ContoursType = std::vector<std::vector<cv::Point>>;

void webcam_watershed::paintEvent(QPaintEvent *e)
{
    cascade.load("C:/Qt/proj/test/haarcascade_frontalface_alt.xml");
    nestedCascade.load("C:/Qt/proj/test/haarcascade_eye_tree_eyeglasses.xml");
    img=frame;
    Mat element = getStructuringElement(MORPH_RECT, Size(50, 50));
```

```
cv::Mat image;
Mat gray;
cv::cvtColor(img, image, CV_BGR2GRAY);
cv::cvtColor(img, gray, CV_BGR2GRAY);
Mat smallImg( cvRound (image.rows/scale), cvRound(image.cols/scale),CV_8UC1 );
cv::resize( image, smallImg, smallImg.size(), 0, 0, INTER_LINEAR );
Mat qq;
qq=image;
cascade.detectMultiScale(smallImg,faces,1.1, 2, 0|CV_HAAR_SCALE_IMAGE,
Size(30, 30) );
for( i = 0; i < faces.size(); i++ )
{
    Point center( faces[i].x*scale + faces[i].width*0.5*scale, (faces[i].y +
faces[i].height*0.5)*scale );
    ellipse( img, center, Size( faces[i].width*0.5*scale,
faces[i].height*0.5*scale), 0, 0, 360, Scalar( 255, 0, 255 ), 4, 8, 0 );
    Mat oo;
    oo=img(faces[i]);
    cv::Scalar o = cv::mean( oo );
    double ooo = o.val[0];
    std::stringstream ss;
    string str;
    int i = 1;
    float f = 10.1;
    double d = 100.11;
    ss << ooo << std::endl;
    ss >> str;
    putText(img, str, Point(faces[i].x*scale,faces[i].y*scale),
FONT_HERSHEY_DUPLEX, 2, Scalar(230, 255, 0), 2);
}
cv::Mat binary;
cv::adaptiveThreshold(image,
binary,255,CV_ADAPTIVE_THRESH_GAUSSIAN_C, CV_THRESH_BINARY, 51, 5);
auto const structure = cv::getStructuringElement(cv::MORPH_ELLIPSE, cv::Size(4,
4));
cv::Mat fore_ground;
cv::erode(binary, fore_ground, structure, cv::Point(-1, -1), 4);
ContourType contours;
cv::findContours(fore_ground, contours, CV_RETR_EXTERNAL,
CV_CHAIN_APPROX_NONE);
fore_ground.setTo(0);
```

```
cv::drawContours(fore_ground, contours, -1, cv::Scalar(255), CV_FILLED);
cv::Mat back_ground;
cv::dilate(binary, back_ground, structure, cv::Point(-1, -1), 4);
cv::threshold(back_ground, back_ground, 1, 128, cv::THRESH_BINARY_INV);
cv::findContours(fore_ground, contours, CV_RETR_EXTERNAL,
CV_CHAIN_APPROX_NONE);
back_ground.setTo(1);
cv::drawContours(back_ground, contours, -1, cv::Scalar(255), CV_FILLED);
cv::Mat markers = back_ground + fore_ground;
cv::Mat mask;
markers.convertTo(mask, CV_32S);
cv::watershed(img, mask);

mask.convertTo(mask, CV_8U);
    cv::adaptiveThreshold(mask,mask,255,CV_ADAPTIVE_THRESH_GAUSSIAN
        _C, CV_THRESH_BINARY, 7, 5);
cv::Mat result;
img.copyTo(result, mask);
    QImage image1 = QImage((uchar*)(img.data), frame.cols,
frame.rows,QImage::Format_RGB888);
    QImage image2 = QImage((uchar*)(binary.data), frame.cols, frame.rows,
QImage::/*Format_Grayscale8*/Format_Indexed8/*Format_RGB888*/);
    ui->label_2->setPixmap(QPixmap::fromImage(image2));
    ui->label_2->resize(image2.size());
    ui->label_2->show();
    ui->label->setPixmap(QPixmap::fromImage(image1));
    ui->label->resize(image1.size());
    ui->label->show();
    imshow("result",result);
    imshow("fore",fore_ground);
    imshow("back",back_ground);
}
void webcam_watershed::display()
{
    cap>>frame;
    if(frame.data)
    {
        cvtColor(frame, frame, CV_BGR2RGB);
        this->update();
    }
}
```

QT GUI programme 2

```
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/core/core.hpp>
#include <opencv2/objdetect/objdetect.hpp>
//#include <QDebug>
#include <iostream>
#include <sstream>
#include <string>
using namespace cv;
void detectAndDraw( Mat& img, CascadeClassifier& cascade,
                  CascadeClassifier& nestedCascade,
                  double scale, bool tryflip );

int main()
{
    VideoCapture cap(1);
    if(!cap.isOpened())
    {
        return -1;
    }
    Mat frame;
    Mat edges;
    CascadeClassifier cascade, nestedCascade;
    bool stop = false;
    cascade.load("C:/Qt/proj/test/haarcascade_frontalface_alt.xml");
    nestedCascade.load("C:/Qt/proj/test/haarcascade_eye_tree_eyeglasses.xml");
    while(!stop)
    {
        cap>>frame;
        detectAndDraw( frame, cascade, nestedCascade,2,0 );
        if(waitKey(30) >=0)
            stop = true;
    }
    return 0;
}

void detectAndDraw( Mat& img, CascadeClassifier& cascade,
                  CascadeClassifier& nestedCascade,
                  double scale, bool tryflip )
```

```
{
Mat qq=img;
Mat element = getStructuringElement(MORPH_RECT, Size(25, 25));
cvtColor( qq, qq, CV_BGR2GRAY );
equalizeHist(qq,qq);
cv::Mat binary;
morphologyEx(qq,qq, MORPH_TOPHAT, element);
vector<vector<Point> > contours;
vector<Vec4i> hierarchy;
cv::findContours(qq,contours,hierarchy, CV_RETR_CCOMP,
CV_CHAIN_APPROX_SIMPLE);
// watershed
Mat contoursImage(qq.size(),CV_8U,Scalar(255));
int count=0;
for(int i=0;i<contours.size();i++)
{
count++;
if(hierarchy[i][3]!=-1) drawContours(contoursImage,contours,i,Scalar(0),3);
using ContoursType = std::vector<std::vector<cv::Point> >;
cv::Mat image;
cv::cvtColor(img, image, CV_BGR2GRAY);
equalizeHist(image,image);
imshow("waters",contoursImage);
imshow("orin",img);
}
```

Matlab GUI Programme

```
function varargout = C_ImagebProcessing(varargin)
% C_IMAGEBPROCESSING MATLAB code for C_ImagebProcessing.fig
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @C_ImagebProcessing_OpeningFcn, ...
                  'gui_OutputFcn',  @C_ImagebProcessing_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before C_ImagebProcessing is made visible.
function C_ImagebProcessing_OpeningFcn(hObject, eventdata, handles, varargin)
    handles.output = hObject;
    initb;
    guidata(hObject, handles);

function varargout = C_ImagebProcessing_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function updateg3(handles)

    r=handles.img(:,:,1);
    g=handles.img(:,:,1);
    b=handles.img(:,:,1);
    [m n c]=size(handles.img);
if c==3
    r=handles.img(:,:,1);
    g=handles.img(:,:,2);
```

```
b=handles.img(:,:,3);
end
x=size(r); x=(1:x(1,2));
r=mean(r); g=mean(g); b=mean(b);
axes(handles.g3); plot(x,r,'r');
hold on
plot(x,g,'g'); plot(x,b,'b'); hold off;

function updateg4(handles)
    r=handles.img(:,:,1);
    g=handles.img(:,:,1);
    b=handles.img(:,:,1);
    [m n c]=size(handles.img);
    if c==3
        r=handles.img(:,:,1);
        g=handles.img(:,:,2);
        b=handles.img(:,:,3);
    end
    x=size(r); x=(1:x(1,2));
    r=mean(r); g=mean(g); b=mean(b);
    axes(handles.g4); plot(x,r,'r');
    hold on
    plot(x,g,'g'); plot(x,b,'b'); hold off;

function load_Callback(hObject, eventdata, handles)

    [file,path]=uigetfile('*.*','MultiSelect','on');
    if (length(file)==0)
        warndlg('You did not selected any file ');
    end
    files=cellstr(file);
    l=length(files);
    inita;

    for ii=1:l
        fi{ii}=strcat(path,'\',files{ii});
        PP{ii}=imread(fi{ii});
        inew1=PP{ii};
        inew(:,1)=inew1(:,1);
        inew(:,2)=inew1(:,2);
        inew(:,3)=inew1(:,3);
```

```
        PP{ii}=inew;
        [x y z]=size(PP{ii});
    if z==3
        r=inew(:,1);
        g=inew(:,2);
        b=inew(:,3);
        rl(ii)=mean(mean(r),2); gl(ii)=mean(mean(g),2); bl(ii)=mean(mean(b),2);
    end

    if z==4
        r=inew(:,1);
        g=inew(:,2);
        b=inew(:,3);

        rl(ii)=mean(mean(r),2); gl(ii)=mean(mean(g),2); bl(ii)=mean(mean(b),2);
    end

    if z==1
        r=inew(:,1);
        g=inew(:,1);
        b=inew(:,1);

        rl(ii)=mean(mean(r),2); gl(ii)=mean(mean(g),2); bl(ii)=mean(mean(b),2);
    end

end

x=1:l;
axes(handles.axes6); plot(x,rl,'r');
hold on
plot(x,gl,'g'); plot(x,bl,'b'); hold off;
handles.PP=PP;
handles.l=l;
axes(handles.g1)
handles.img=PP{ 1 };
imshow(handles.img);
axes(handles.g2)
imshow(handles.img);
updateg3(handles);
updateg4(handles);
set(handles.listbox1,'string',files);
set(handles.edit33,'string',num2str(0));
set(handles.edit34,'string',num2str(0));
```

```
guidata(hObject,handles);
```

```
function save_Callback(hObject, eventdata, handles)
```

```
    [file path]= uiputfile('* .jpg','Save Image as');  
    axes(handles.g2)  
    sa=getimage(gca);  
    save=[path file]; imwrite((sa),save,'jpg');
```

```
function slider1_Callback(hObject, eventdata, handles)
```

```
r1=get(handles.slider1,'Value');  
    g1=get(handles.slider2,'Value');  
    b1=get(handles.slider3,'Value');  
    c1=get(handles.slider4,'Value');  
    br1=get(handles.slider5,'Value');  
    ro1=get(handles.slider6,'Value');  
    x=r1;  
    r=handles.img(:,:,1);  
    g=handles.img(:,:,2); b=handles.img(:,:,3);  
    r1=r+r1;  
    g1=g+g1;  
    b1=b+b1;  
    bcon=cat(3,r1,g1,b1);  
    img=bcon;  
    img=(bcon).*c1+br1);  
    axes(handles.g2); cla; imshow(imrotate(img,ro1))  
    set(handles.rv,'String',num2str(x));  
    handles.img=img;  
  
    r=handles.img(:,:,1);  
    g=handles.img(:,:,2);  
    b=handles.img(:,:,3);  
    x=size(r); x=(1:x(1,2));  
    r=mean(r); g=mean(g); b=mean(b);  
    axes(handles.g4); plot(x,r,'r');  
    hold on  
    plot(x,g,'g'); plot(x,b,'b'); hold off;
```

```
function slider1_CreateFcn(hObject, eventdata, handles)
```

```
    if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))  
        set(hObject,'BackgroundColor',[.9 .9 .9]);  
    end
```

```
function rv_Callback(hObject, eventdata, handles)
function rv_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end
```

```
function slider2_Callback(hObject, eventdata, handles)
```

```
    r1=get(handles.slider1,'Value');
    g1=get(handles.slider2,'Value');
    b1=get(handles.slider3,'Value');
    c1=get(handles.slider4,'Value');
    br1=get(handles.slider5,'Value');
    ro1=get(handles.slider6,'Value');
    x=g1;
    r=handles.img(:,:,1);
    g=handles.img(:,:,2); b=handles.img(:,:,3);
    r1=r+r1;
    g1=g+g1;
    b1=b+b1;
    bcon=cat(3,r1,g1,b1);
    img=bcon;
    img=(bcon).*c1+br1;
    axes(handles.g2); cla; imshow(imrotate(img,ro1))
    set(handles.gv,'String',num2str(x));
    handles.img=img;
    updateg4(handles)
```

```
function slider2_CreateFcn(hObject, eventdata, handles)
```

```
    if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor',[.9 .9 .9]);
    end
```

```
function gv_Callback(hObject, eventdata, handles)
```

```
function gv_CreateFcn(hObject, eventdata, handles)
```

```
    if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end
```

```
function slider3_Callback(hObject, eventdata, handles)
```

```
    r1=get(handles.slider1,'Value');
    g1=get(handles.slider2,'Value');
    b1=get(handles.slider3,'Value');
    c1=get(handles.slider4,'Value');
    br1=get(handles.slider5,'Value');
    ro1=get(handles.slider6,'Value');
    x=b1;
    r=handles.img(:,:,1);
    g=handles.img(:,:,2); b=handles.img(:,:,3);
    r1=r+r1;
    g1=g+g1;
    b1=b+b1;
    bcon=cat(3,r1,g1,b1);
    img=bcon;
    img=((bcon).*c1+br1);
    axes(handles.g2); cla; imshow(imrotate(img,ro1))
    set(handles.bv,'String',num2str(x));
    handles.img=img;
    updateg4(handles)
```

```
function slider3_CreateFcn(hObject, eventdata, handles)
```

```
    if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor',[.9 .9 .9]);
    end
```

```
function bv_Callback(hObject, eventdata, handles)
```

```
function bv_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function slider4_Callback(hObject, eventdata, handles)
```

```
    r1=get(handles.slider1,'Value');
    g1=get(handles.slider2,'Value');
    b1=get(handles.slider3,'Value');
```

```
c1=get(handles.slider4,'Value');
br1=get(handles.slider5,'Value');
ro1=get(handles.slider6,'Value');
x=c1;
r=handles.img(:,:,1);
g=handles.img(:,:,2); b=handles.img(:,:,3);
r1=r+r1;
g1=g+g1;
b1=b+b1;
bcon=cat(3,r1,g1,b1);
img=bcon;
img=(bcon).*c1+br1;
axes(handles.g2); cla; imshow(imrotate(img,ro1))
set(handles.cv,'String',num2str(x));
handles.img=img;
updateg4(handles)
```

```
function slider4_CreateFcn(hObject, eventdata, handles)
```

```
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor',[.9 .9 .9]);
end
```

```
function cv_Callback(hObject, eventdata, handles)
```

```
function cv_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```

```
function slider5_Callback(hObject, eventdata, handles)
```

```
r1=get(handles.slider1,'Value');
g1=get(handles.slider2,'Value');
b1=get(handles.slider3,'Value');
c1=get(handles.slider4,'Value');
br1=get(handles.slider5,'Value');
ro1=get(handles.slider6,'Value');
x=br1;
r=handles.img(:,:,1);
g=handles.img(:,:,2); b=handles.img(:,:,3);
```

```
r1=r+r1;
g1=g+g1;
b1=b+b1;
bcon=cat(3,r1,g1,b1);
img=bcon;
img=(bcon.*c1+x);
axes(handles.g2); cla; imshow(imrotate(img,ro1))
set(handles.bbv,'String',num2str(x));
handles.img=img;
updateg4(handles)
```

```
function slider5_CreateFcn(hObject, eventdata, handles)
```

```
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor',[.9 .9 .9]);
end
```

```
function bbv_Callback(hObject, eventdata, handles)
```

```
function bbv_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```

```
function reset_Callback(hObject, eventdata, handles)
```

```
axes(handles.g2);
handles.img=handles.img2;
imshow(handles.img);
updateg4(handles);
s=num2str(size(handles.img));
guidata(hObject,handles);
```

```
function set1_Callback(hObject, eventdata, handles)
```

```
r1=0;
set(handles.slider1,'value',0);
g1=get(handles.slider2,'Value');
b1=get(handles.slider3,'Value');
c1=get(handles.slider4,'Value');
br1=get(handles.slider5,'Value');
ro1=get(handles.slider6,'Value');
```

```
x=r1;
r=handles.img(:,:,1);
g=handles.img(:,:,2); b=handles.img(:,:,3);
r1=r+r1;
g1=g+g1;
b1=b+b1;
bcon=cat(3,r1,g1,b1);

set(handles.rv,'String',num2str(0));
img=bcon;
img=((bcon.*c1+br1));
axes(handles.g2); cla; imshow(imrotate(img,ro1))
handles.img=img;
r=handles.img(:,:,1);
g=handles.img(:,:,2);
b=handles.img(:,:,3);
x=size(r); x=(1:x(1,2));
r=mean(r); g=mean(g); b=mean(b);
axes(handles.g4); plot(x,r,'r');
hold on
plot(x,g,'g'); plot(x,b,'b'); hold off;
```

```
function set2_Callback(hObject, eventdata, handles)
```

```
g1=0;
set(handles.slider2,'value',0);
r1=get(handles.slider1,'Value');
b1=get(handles.slider3,'Value');
c1=get(handles.slider4,'Value');
br1=get(handles.slider5,'Value');
ro1=get(handles.slider6,'Value');
x=r1;
r=handles.img(:,:,1);
g=handles.img(:,:,2); b=handles.img(:,:,3);
r1=r+r1;
g1=g+g1;
b1=b+b1;
bcon=cat(3,r1,g1,b1);
set(handles.gv,'String',num2str(0));
img=bcon;
img=((bcon.*c1+br1));
```

```
axes(handles.g2); cla; imshow(imrotate(img,ro1))
handles.img=img;
updateg4(handles);
```

function set3_Callback(hObject, eventdata, handles)

```
b1=0;
set(handles.slider3,'value',0);
r1=get(handles.slider1,'Value');
g1=get(handles.slider2,'Value');
c1=get(handles.slider4,'Value');
br1=get(handles.slider5,'Value');
ro1=get(handles.slider6,'Value');
x=r1;
r=handles.img(:,:,1);
g=handles.img(:,:,2); b=handles.img(:,:,3);
r1=r+r1;
g1=g+g1;
b1=b+b1;
bcon=cat(3,r1,g1,b1);
set(handles.bv,'String',num2str(0));
img=bcon;
img=((bcon.*c1+br1));
axes(handles.g2); cla; imshow(imrotate(img,ro1))
handles.img=img;
updateg4(handles);
```

function set4_Callback(hObject, eventdata, handles)

```
c1=1;
set(handles.slider4,'value',1);
r1=get(handles.slider1,'Value');
g1=get(handles.slider2,'Value');
b1=get(handles.slider3,'Value');
br1=get(handles.slider5,'Value');
ro1=get(handles.slider6,'Value');
x=r1;
r=handles.img(:,:,1);
g=handles.img(:,:,2); b=handles.img(:,:,3);
r1=r+r1;
g1=g+g1;
```

```
b1=b+b1;
bcon=cat(3,r1,g1,b1);
set(handles.cv,'String',num2str(1));
img=bcon;
img=((bcon.*c1+br1));
axes(handles.g2); cla; imshow(imrotate(img,ro1))
handles.img=img;
updateg4(handles);
```

```
function set5_Callback(hObject, eventdata, handles)
```

```
br1=0;
set(handles.slider5,'value',0);
r1=get(handles.slider1,'Value');
g1=get(handles.slider2,'Value');
b1=get(handles.slider3,'Value');
c1=get(handles.slider4,'Value');
ro1=get(handles.slider6,'Value');
x=r1;
r=handles.img(:,:,1);
g=handles.img(:,:,2); b=handles.img(:,:,3);
r1=r+r1;
g1=g+g1;
b1=b+b1;
bcon=cat(3,r1,g1,b1);
set(handles.bbv,'String',num2str(0));
img=bcon;
img=((bcon.*c1+br1));
axes(handles.g2); cla; imshow(imrotate(img,ro1))
handles.img=img;
updateg4(handles);
```

```
function e1_Callback(hObject, eventdata, handles)
```

```
function e1_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```

```
function e2_Callback(hObject, eventdata, handles)
```

```
function e2_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```

```
function resize_Callback(hObject, eventdata, handles)
```

```
h=str2num(get(handles.e1,'String'));
w=str2num(get(handles.e2,'String'));
handles.img=imresize(handles.img, [h w]);
axes(handles.g2); cla; imshow(handles.img);
updateg4(handles);
s=num2str(size(handles.img));
guidata(hObject,handles);
```

```
function chres_Callback(hObject, eventdata, handles)
```

```
val=get(hObject,'Value');
switch val
case 1
    warndlg('Please select a resolution value');
case 2 % 256x256
    handles.res=handles.img;
    axes(handles.g2); cla;
    handles.res=imresize(imresize(handles.res,.5),1);
    imshow(handles.res);
    guidata(hObject, handles);
    handles.img=handles.res;
    updateg4(handles);

case 3 % 128X128
    handles.res=handles.img;
    axes(handles.g2); cla;
    handles.res=imresize(imresize(handles.res,1/2),2);
    imshow(handles.res);
    guidata(hObject, handles);
    handles.img=handles.res; updateg4(handles);

case 4 % 64x64
    handles.res=handles.img;
    axes(handles.g2); cla;
```

```
handles.res=imresize(imresize(handles.res,1/4),4);
imshow(handles.res);
guidata(hObject, handles);
handles.img=handles.res; updateg4(handles);
```

case 5 % 32x32

```
handles.res=handles.img;
axes(handles.g2); cla;
handles.res=imresize(imresize(handles.res,1/8),8);
imshow(handles.res);
guidata(hObject, handles);
handles.img=handles.res; updateg4(handles);
```

case 6 % 16x16

```
handles.res=handles.img;
axes(handles.g2); cla;
handles.res=imresize(imresize(handles.res,1/16),16);
imshow(handles.res);
guidata(hObject, handles);
handles.img=handles.res; updateg4(handles);
```

case 7 % 8x8

```
handles.res=handles.img;
axes(handles.g2); cla;
handles.res=imresize(imresize(handles.res,1/32),32);
imshow(handles.res);
guidata(hObject, handles);
handles.img=handles.res; updateg4(handles);
```

end

function chres_CreateFcn(hObject, eventdata, handles)

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```

function sres_Callback(hObject, eventdata, handles)

```
handles.img=handles.res;
axes(handles.g2); cla; imshow(handles.img);
s=num2str(size(handles.img));
```

```
guidata(hObject,handles);
update4(handles);
```

```
function slider6_Callback(hObject, eventdata, handles)
```

```
    r1=get(handles.slider1,'Value');
    g1=get(handles.slider2,'Value');
    b1=get(handles.slider3,'Value');
    c1=get(handles.slider4,'Value');
    br1=get(handles.slider5,'Value');
    ro1=get(handles.slider6,'Value');
    x=br1;
    r=handles.img(:,:,1);
    g=handles.img(:,:,2); b=handles.img(:,:,3);
    r1=r+r1;
    g1=g+g1;
    b1=b+b1;
    bcon=cat(3,r1,g1,b1);
    img=bcon;
    img=(bcon.*c1+x);
    set(handles.rrv,'string',num2str(ro1));
    handles.rot=img;
    handles.rot=imrotate(handles.rot,ro1);
    axes(handles.g2); cla; imshow(handles.rot);
    guidata(hObject,handles)
```

```
function slider6_CreateFcn(hObject, eventdata, handles)
```

```
    if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor',[.9 .9 .9]);
    end
```

```
function rrv_Callback(hObject, eventdata, handles)
```

```
function rrv_CreateFcn(hObject, eventdata, handles)
```

```
    if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end
```

```
function set6_Callback(hObject, eventdata, handles)
```

```
ro1=0;
set(handles.slider6,'value',0);
r1=get(handles.slider1,'Value');
g1=get(handles.slider2,'Value');
b1=get(handles.slider3,'Value');
br1=get(handles.slider5,'Value');
c1=get(handles.slider4,'Value');
x=r1;
r=handles.img(:,:,1);
g=handles.img(:,:,2); b=handles.img(:,:,3);
r1=r+r1;
g1=g+g1;
b1=b+b1;
bcon=cat(3,r1,g1,b1);
set(handles.rrv,'String',num2str(0));
img=bcon;
img=((bcon.*c1+br1));

axes(handles.g2); cla; imshow(imrotate(img,ro1))
handles.img=img;
updateg4(handles);
```

```
function p1_Callback(hObject, eventdata, handles)
```

```
h = waitbar(0,'Please wait...');
```

```
steps = 100;
```

```
for step = 1:steps
```

```
    waitbar(step / steps)
```

```
end
```

```
close(h)
```

```
imageinfo(handles.file);
```

```
function p2_Callback(hObject, eventdata, handles)
```

```
    if (get(hObject,'Value')==get(hObject,'Max'))
```

```
        axes(handles.g2); imdistline
```

```
    elseif (get(hObject,'Value')==get(hObject,'Min'))
```

```
        axes(handles.g2); imshow(handles.img)
```

```
    end
```

```
function p3_Callback(hObject, eventdata, handles)
```

```
impixelinfo;
```

```
function p4_Callback(hObject, eventdata, handles)
```

```
    if (get(hObject,'Value')==get(hObject,'Max'))  
        axes(handles.g2); impixelregion  
    elseif (get(hObject,'Value')==get(hObject,'Min'))  
        axes(handles.g2); imshow(handles.img)  
    end
```

```
function p5_Callback(hObject, eventdata, handles)
```

```
    imscrollpanel(handles.g2,handles.img);
```

```
function p6_Callback(hObject, eventdata, handles)
```

```
    imoverview(handles.image)
```

```
function p7_Callback(hObject, eventdata, handles)
```

```
function selectcm_Callback(hObject, eventdata, handles)
```

```
handles.cm=handles.img;
```

```
val=get(hObject,'Value');
```

```
switch val
```

```
    case 1
```

```
        helpdlg('Please first select a color map','Color Map help');
```

```
    case 2
```

```
        handles.cm=rgb2ind(handles.cm,jet);
```

```
        handles.cm=ind2rgb(handles.cm,jet);
```

```
        axes(handles.g2); cla; imshow(handles.cm);
```

```
        guidata(hObject,handles);
```

```
        handles.img=handles.cm;
```

```
        updateg4(handles);
```

```
    case 3
```

```
        handles.cm=rgb2ind(handles.cm,HSV);
```

```
        handles.cm=ind2rgb(handles.cm,HSV);
```

```
        axes(handles.g2); cla; imshow(handles.cm);
```

```
        guidata(hObject,handles);
```

```
        handles.img=handles.cm;
```

```
        updateg4(handles);
```

case 4

```
handles.cm=rgb2ind(handles.cm,hot);
handles.cm=ind2rgb(handles.cm,hot);
axes(handles.g2); cla; imshow(handles.cm);
guidata(hObject,handles);
handles.img=handles.cm;
updateg4(handles);
```

case 5

```
handles.cm=rgb2ind(handles.cm,cool);
handles.cm=ind2rgb(handles.cm,cool);
axes(handles.g2); cla; imshow(handles.cm);
guidata(hObject,handles);
handles.img=handles.cm;
updateg4(handles);
```

case 6

```
handles.cm=rgb2ind(handles.cm,spring);
handles.cm=ind2rgb(handles.cm,spring);
axes(handles.g2); cla; imshow(handles.cm);
guidata(hObject,handles);
handles.img=handles.cm;
updateg4(handles);
```

case 7

```
handles.cm=rgb2ind(handles.cm,summer);
handles.cm=ind2rgb(handles.cm,summer);
axes(handles.g2); cla; imshow(handles.cm);
guidata(hObject,handles);
handles.img=handles.cm;
updateg4(handles);
```

case 8

```
handles.cm=rgb2ind(handles.cm,autumn);
handles.cm=ind2rgb(handles.cm,autumn);
axes(handles.g2); cla; imshow(handles.cm);
guidata(hObject,handles);
handles.img=handles.cm;
updateg4(handles);
```

case 9

```
handles.cm=rgb2ind(handles.cm,winter);
handles.cm=ind2rgb(handles.cm,winter);
axes(handles.g2); cla; imshow(handles.cm);
guidata(hObject,handles);
handles.img=handles.cm;
```

```
        updateg4(handles);
    case 10
        handles.cm=rgb2ind(handles.cm,gray);
        handles.cm=ind2rgb(handles.cm,gray);
        axes(handles.g2); cla; imshow(handles.cm);
        guidata(hObject,handles);
        handles.img=handles.cm;
        updateg4(handles);
    case 11
        handles.cm=rgb2ind(handles.cm,bone);
        handles.cm=ind2rgb(handles.cm,bone);
        axes(handles.g2); cla; imshow(handles.cm);
        guidata(hObject,handles);
        handles.img=handles.cm;
        updateg4(handles);
    case 12
        handles.cm=rgb2ind(handles.cm,copper);
        handles.cm=ind2rgb(handles.cm,copper);
        axes(handles.g2); cla; imshow(handles.cm);
        guidata(hObject,handles);
        handles.img=handles.cm;
        updateg4(handles);
    case 13
        handles.cm=rgb2ind(handles.cm,pink);
        handles.cm=ind2rgb(handles.cm,pink);
        axes(handles.g2); cla; imshow(handles.cm);
        guidata(hObject,handles);
        handles.img=handles.cm;
        updateg4(handles);
    case 14
        handles.cm=rgb2ind(handles.cm,lines);
        handles.cm=ind2rgb(handles.cm,lines);
        axes(handles.g2); cla; imshow(handles.cm);
        guidata(hObject,handles);
        handles.img=handles.cm;
        updateg4(handles);
end

function selectcm_CreateFcn(hObject, eventdata, handles)

    if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
```

```
set(hObject,'BackgroundColor','white');  
end
```

```
function setcm_Callback(hObject, eventdata, handles)
```

```
handles.img=handles.cm;  
axes(handles.g2); cla; imshow(handles.img);  
updateg4(handles);  
guidata(hObject,handles);
```

```
function invc_Callback(hObject, eventdata, handles)
```

```
handles.img=imcomplement((handles.img));  
axes(handles.g2); imshow((handles.img));  
updateg4(handles);  
guidata(hObject,handles);
```

```
function autoa_Callback(hObject, eventdata, handles)
```

```
handles.img= imadjust(handles.img,[.2 .3 0; .6 .7 1],[[]]);  
axes(handles.g2); cla; imshow(handles.img);  
guidata(hObject,handles);  
updateg4(handles);
```

```
function rgbo_Callback(hObject, eventdata, handles)
```

```
r=handles.img(:,:,1);  
axes(handles.g2);  
imshow(r);  
r=handles.img(:,:,1);  
g=handles.img(:,:,1);  
b=handles.img(:,:,1);  
x=size(r); x=(1:x(1,2));  
r=mean(r); g=mean(g); b=mean(b);  
axes(handles.g4); plot(x,r,'r');
```

```
function rgbm_Callback(hObject, eventdata, handles)
```

```
g=handles.img(:,:,2);  
axes(handles.g2);  
imshow(g);
```

```
r=handles.img(:,:,2);
g=handles.img(:,:,2);
b=handles.img(:,:,2);
x=size(r); x=(1:x(1,2));
r=mean(r); g=mean(g); b=mean(b);
axes(handles.g4);
plot(x,g,'g');
```

```
function exit_Callback(hObject, eventdata, handles)
```

```
    close all;
```

```
function autocon_Callback(hObject, eventdata, handles)
```

```
    r=handles.img(:,:,1);  r=histeq(r);
    g=handles.img(:,:,2);  g=histeq(g);
    b=handles.img(:,:,3);  b=histeq(b);
    handles.img=cat(3,r,g,b);
    axes(handles.g2); cla; imshow(handles.img);
    guidata(hObject,handles);
    updateg4(handles);
```

```
function n1_Callback(hObject, eventdata, handles)
```

```
    handles.img = imnoise(handles.img,'gaussian');
    axes(handles.g2); cla; imshow(handles.img);
    guidata(hObject,handles);
    updateg4(handles);
```

```
function n3_Callback(hObject, eventdata, handles)
```

```
h = waitbar(0,'Please wait...');
steps = 200;
for step = 1:steps
    waitbar(step / steps)
end
close(h)
handles.img = imnoise(handles.img,'poisson');
axes(handles.g2); cla; imshow(handles.img);
guidata(hObject,handles);
updateg4(handles);
```

```
function n4_Callback(hObject, eventdata, handles)
```

```
handles.img = imnoise(handles.img,'salt & pepper',0.02);
axes(handles.g2); cla; imshow(handles.img);
guidata(hObject,handles);
updateg4(handles);
```

```
function n5_Callback(hObject, eventdata, handles)
```

```
handles.img = imnoise(handles.img,'speckle',0.04);
axes(handles.g2); cla; imshow(handles.img);
guidata(hObject,handles);
updateg4(handles);
```

```
function f1_Callback(hObject, eventdata, handles)
```

```
h=fspecial('average');
handles.img=imfilter(handles.img,h,'replicate');
axes(handles.g2); cla; imshow(handles.img)
guidata(hObject,handles);
updateg4(handles);
```

```
function crop_Callback(hObject, eventdata, handles)
```

```
imcrop(handles.g2);
handles.img=getimage(gca); close;
axes(handles.g2); imshow(handles.img);
inew=handles.img;
[x y z]=size(handles.img);
```

```
if z==3
```

```
gray=rgb2gray(handles.img);
set(handles.edit25,'string',num2str(mean(mean(inew(:,1),2))));
set(handles.edit28,'string',num2str(sqrt(var(double(inew(:,1))))));
set(handles.edit26,'string',num2str(mean(mean(inew(:,2),2))));
set(handles.edit29,'string',num2str(sqrt(var(double(inew(:,2))))));
set(handles.edit27,'string',num2str(mean(mean(inew(:,3),2))));
set(handles.edit30,'string',num2str(sqrt(var(double(inew(:,3))))));
set(handles.edit31,'string',num2str(mean(mean(gray,2))));
set(handles.edit32,'string',num2str(sqrt(var(double(gray)))));
```

```
s=num2str(size(handles.img));
update4(handles);
end

if z==1
    gray=(handles.img);
    set(handles.edit25,'string',num2str(mean(mean(inew(:,:,1),2))));
    set(handles.edit28,'string',num2str(sqrt(var(double(inew(:,:,1))))));
    set(handles.edit26,'string',num2str(mean(mean(inew(:,:,1),2))));
    set(handles.edit29,'string',num2str(sqrt(var(double(inew(:,:,1))))));
    set(handles.edit27,'string',num2str(mean(mean(inew(:,:,1),2))));
    set(handles.edit30,'string',num2str(sqrt(var(double(inew(:,:,1))))));
    set(handles.edit31,'string',num2str(mean(mean(gray,2))));
    set(handles.edit32,'string',num2str(sqrt(var(double(gray)))));

    s=num2str(size(handles.img));

    r=handles.img(:,:,1);
    g=handles.img(:,:,1);
    b=handles.img(:,:,1);
    x=size(r); x=(1:x(1,2));
    r=mean(r); g=mean(g); b=mean(b);
    axes(handles.g4); plot(x,r,'r');
    hold on
    plot(x,g,'g'); plot(x,b,'b'); hold off;
end

function dia_Callback(hObject, eventdata, handles)

h = waitbar(0,'Please wait...');
    steps = 100;
for step = 1:steps
    waitbar(step / steps)
end

close(h)
handles.mori=handles.img;
str=strel(handles.strel,handles.value);
handles.mori=imdilate(handles.mori,str);
axes(handles.g2); cla; imshow(handles.mori)
guidata(hObject,handles);
handles.img=handles.mori;
```

```
updateg4(handles);
```

```
function f3_Callback(hObject, eventdata, handles)
```

```
r=medfilt2(handles.img(:,:,1)); g=medfilt2(handles.img(:,:,2)); b=medfilt2(handles.img(:,:,3));  
handles.img=cat(3,r,g,b);  
axes(handles.g2); cla; imshow(handles.img);  
guidata(hObject,handles);  
updateg4(handles);
```

```
function f2_Callback(hObject, eventdata, handles)
```

```
hsize=[8 8]; sigma=1.7;  
h=fspecial('gaussian',hsize,sigma);  
handles.img=imfilter(handles.img,h,'replicate');  
axes(handles.g2); cla; imshow(handles.img);  
guidata(hObject,handles);  
updateg4(handles);
```

```
function f4_Callback(hObject, eventdata, handles)
```

```
h=fspecial('unsharp');  
handles.img=imfilter(handles.img,h,'replicate');  
axes(handles.g2); cla; imshow(handles.img);  
guidata(hObject,handles);  
updateg4(handles);
```

```
function ero_Callback(hObject, eventdata, handles)
```

```
h = waitbar(0,'Please wait...');  
steps = 100;  
for step = 1:steps  
    waitbar(step / steps)  
end  
close(h)  
handles.mori=handles.img;  
str=strel(handles.strel,handles.value);  
handles.mori=imerode(handles.mori,str);  
axes(handles.g2); cla; imshow(handles.mori);  
guidata(hObject,handles);  
handles.img=handles.mori;
```

```
updateg4(handles);
```

```
function strel_Callback(hObject, eventdata, handles)
```

```
val=get(hObject,'Value');
```

```
switch val
```

```
case 1
```

```
helpdlg('Please first select Structring Element and value ','Morphological Operations');
```

```
case 2
```

```
handles.strel='diamond';
```

```
case 3
```

```
handles.strel='disk';
```

```
case 4
```

```
helpdlg('Value must be a nonnegative multiple of 3.','Octagon Structure Help');
```

```
handles.strel='octagon';
```

```
end
```

```
guidata(hObject,handles);
```

```
function strelv_Callback(hObject, eventdata, handles)
```

```
handles.value=str2num(get(hObject,'String'));
```

```
guidata(hObject,handles);
```

```
function strels_Callback(hObject, eventdata, handles)
```

```
h = waitbar(0,'Please wait...');
```

```
steps = 100;
```

```
for step = 1:steps
```

```
waitbar(step / steps)
```

```
end
```

```
close(h)
```

```
handles.img=handles.mori;
```

```
axes(handles.g2); cla; imshow(handles.img)
```

```
guidata(hObject,handles);
```

```
updateg4(handles);
```

```
function dfh2_Callback(hObject, eventdata, handles)
```

```
helpdlg('Click and draw a free hand on MODIFIED image & then press ENTER. ');
```

```
axes(handles.g2); improfile;grid on;
```

```
function ex_Callback(hObject, eventdata, handles)
```

```
close;
```

```
function pushbutton52_Callback(hObject, eventdata, handles)
```

```
axes(handles.g2)
handles.img=getimage(gca);
updateg3(handles);
updateg4(handles);
s=num2str(size(handles.img));
guidata(hObject,handles);
```

```
function pushbutton53_Callback(hObject, eventdata, handles)
```

```
pathname=uigetdir(cd);
```

```
files=dir(strcat(pathname, '*jpg'));
```

```
len=length(files);
```

```
for i=1:len
```

```
    fname{i}=strcat(pathname, '\',files(i).name);
```

```
end
```

```
    sz = 512;
```

```
    out_fn = outputfiles;
```

```
    stage = 4;
```

```
    orientation = 6;
```

```
    N = 128;
```

```
    freq = [0.05 0.4];
```

```
    flag = 0;
```

```
    j = sqrt(-1);
```

```
for s = 1:stage,
```

```
    for n = 1:orientation,
```

```
        [Gr,Gi] = Gabor(N,[s n],freq,[stage orientation],flag);
```

```
        F = fft2(Gr+j*Gi);
```

```
        F(1,1) = 0;
```

```
        GW(N*(s-1)+1:N*s,N*(n-1)+1:N*n) = F;
```

```
    end;
```

```
end;
```

```
% -----
```

```
height = 7;
width = 7;
c=cell(1,len);
fk=0;
for i = 1:len,
    w = waitbar(fk/len, 'Gabor extracting ... Please wait ...');
    close(w);
    texture = loading(fname{i},sz);
    A = zeros(stage*orientation*2,height*width);
    for h = 1:height,
        for w = 1:width,
            [i h w];
            img = texture((h-1)*64+1:(h-1)*64+128, (w-1)*64+1:(w-1)*64+128);
            F = Fea_Gabor_brodatz(img, GW, N, stage, orientation);
            A(:,(h-1)*width+w) = [F(:,1); F(:,2)];
        end;
    end;
    output = [ extractName(out_fn(i,:)) '.fea'];
    fid = fopen(output, 'w');
    fwrite(fid, A, 'float');
    fclose(fid);
    fname{i}=strrep(fname{i},pathname,"");
    ffname{i}=strrep(ffname{i},'\','');
    c{i}=A;
    fk=fk+1;
end
save 'db\\database.mat'
function pushbutton54_Callback(hObject, eventdata, handles)

    axes(handles.g2);
    handles.img=getimage(gca);
    axes(handles.g2);
    inew=handles.img;
    [x y z]=size(handles.img);
    if z==3
        gray=rgb2gray(handles.img);
        set(handles.edit25,'string',num2str(mean(mean(inew(:,1),2))));
        set(handles.edit28,'string',num2str(sqrt(var(double(inew(:,1))))));
        set(handles.edit26,'string',num2str(mean(mean(inew(:,2),2))));
        set(handles.edit29,'string',num2str(sqrt(var(double(inew(:,2))))));
        set(handles.edit27,'string',num2str(mean(mean(inew(:,3),2))));
```

```
set(handles.edit30,'string',num2str(sqrt(var(double(inew(:,2,3))))));
set(handles.edit31,'string',num2str(mean(mean(gray,2))));
set(handles.edit32,'string',num2str(sqrt(var(double(gray)))));
s=num2str(size(handles.img));
update4(handles);
```

```
end
```

```
if z==1
```

```
gray=(handles.img);
set(handles.edit25,'string',num2str(mean(mean(inew(:,1,2)))));
set(handles.edit28,'string',num2str(sqrt(var(double(inew(:,1))))));
set(handles.edit26,'string',num2str(mean(mean(inew(:,1,2)))));
set(handles.edit29,'string',num2str(sqrt(var(double(inew(:,1))))));
set(handles.edit27,'string',num2str(mean(mean(inew(:,1,2)))));
set(handles.edit30,'string',num2str(sqrt(var(double(inew(:,1))))));
set(handles.edit31,'string',num2str(mean(mean(gray,2))));
set(handles.edit32,'string',num2str(sqrt(var(double(gray)))));
s=num2str(size(handles.img));
r=handles.img(:,1);
g=handles.img(:,2);
b=handles.img(:,3);
x=size(r); x=(1:x(1,2));
r=mean(r); g=mean(g); b=mean(b);
axes(handles.g4); plot(x,r,'r');
hold on
plot(x,g,'g'); plot(x,b,'b'); hold off;
```

```
end
```

```
function pushbutton55_Callback(hObject, eventdata, handles)
```

```
set(handles.uipanel2,'Visible','off');
set(handles.pushbutton55,'Visible','off');
```

```
function pushbutton56_Callback(hObject, eventdata, handles)
```

```
set(handles.uipanel2,'Visible','on');
set(handles.pushbutton55,'Visible','on');
set(handles.cv,'string',num2str(1));
set(handles.slider4,'value',1);
```

```
function pushbutton57_Callback(hObject, eventdata, handles)
```

```
c1=0;
set(handles.slider1,'value',0);
set(handles.slider2,'value',0);
set(handles.slider3,'value',0);
set(handles.slider4,'value',1);
set(handles.slider5,'value',0);
set(handles.slider6,'value',0);
r=handles.img(:,:,1);
g=handles.img(:,:,2); b=handles.img(:,:,3);
bcon=cat(3,r,g,b);
img=bcon;
set(handles.cv,'String',num2str(1));
set(handles.rv,'String',num2str(0));
set(handles.gv,'String',num2str(0));
set(handles.bv,'String',num2str(0));
set(handles.bbv,'String',num2str(0));
set(handles.rrv,'String',num2str(0));
axes(handles.g2); cla; imshow(bcon)
handles.img=img;
updateg4(handles);
```

function pushbutton58_Callback(hObject, eventdata, handles)

```
r1=get(handles.slider1,'Value');
g1=get(handles.slider2,'Value');
b1=get(handles.slider3,'Value');
c1=get(handles.slider4,'Value');
br1=get(handles.slider5,'Value');
ro1=get(handles.slider6,'Value');
r=handles.img(:,:,1);
g=handles.img(:,:,2); b=handles.img(:,:,3);
r1=r+r1;
g1=g+g1;
b1=b+b1;
bcon=cat(3,r1,g1,b1);
% img=imrotate((bcon.*c1+br1),ro1);
img=bcon;
img=((bcon).*c1+br1);
axes(handles.g2);
imshow(imrotate(img,ro1))
```

```
handles.img=img;
axes(handles.g1);
imshow(imrotate(img,ro1))
handles.img=img;
guidata(hObject,handles);
```

```
function pushbutton59_Callback(hObject, eventdata, handles)
```

```
bw=handles.img;
[labeled,numObjects] = bwlabel(bw,8);
set(handles.edit33,'string',num2str(numObjects));
dataa=regionprops(labeled,'basic');
for i=1:numObjects
    CC(i)=dataa(i).Area;
end
avarea=(sum(CC))/numObjects;
axes(handles.g1)
iimg=(getimage(gca));
figure
imshow(iimg);
title(['cells : ',num2str(numObjects),'/', 'Average area (in pixels):',num2str(avarea)])
```

```
function pushbutton60_Callback(hObject, eventdata, handles)
```

```
[M,N,ttt] = size(handles.img);
RUN = 1;
while (RUN==1)
    prompt = {'Enter Median Row Factor (0-5%):','Enter Median Column Factor (0-5%):'};
    dlg_title = 'Enter Median Parameters: ';
    num_lines = 1;
    def = {'2','2'};
    answer = inputdlg(prompt,dlg_title,num_lines,def);
    if (isempty(answer))
        return;
    end
    M1 = str2num(answer{1})/100;
    M2 = str2num(answer{2})/100;
    if ((str2num(answer{1})>=0) & (str2num(answer{1})<=5) & ((str2num(answer{2})>=0) &
(str2num(answer{2})<=5))
        RUN = 0;
    end
```

```
end

M1 = round(M1 * M);
M2 = round(M2 * N);

w = waitbar(0, 'Median filtering ... Please wait ...');
handles.img(:,:,1) = medfilt2(handles.img(:,:,1),[M1 M2]);
waitbar(1/3, w);
handles.img(:,:,2) = medfilt2(handles.img(:,:,2),[M1 M2]);
waitbar(2/3, w);
handles.img(:,:,3) = medfilt2(handles.img(:,:,3),[M1 M2]);
close(w);
axes(handles.g2); imshow(handles.img);
updateg4(handles);
guidata(hObject, handles);
function pushbutton61_Callback(hObject, eventdata, handles)

[M,N,ttt] = size(handles.img);
D = max(M,N);
RUN = 1;
while (RUN==1)
    prompt = {'Enter Motion Length (0-15%):','Enter Motion Angle (0-360):'};
    dlg_title = 'Enter Motion Parameters: ';
    num_lines = 1;
    def = {'1','0'};
    answer = inputdlg(prompt,dlg_title,num_lines,def);
    if (isempty(answer))
        return;
    end
    M1 = str2num(answer{1})/100;
    M2 = str2num(answer{2});
    if ((str2num(answer{1})>=0) & (str2num(answer{1})<=15))
        RUN = 0;
    end
end
end

H = fspecial('motion',D * M1,M2);
w = waitbar(0, 'Motion filtering ... Please wait ...');
handles.img(:,:,1) = imfilter(handles.img(:,:,1),H,'replicate');
waitbar(1/3, w);
handles.img(:,:,2) = imfilter(handles.img(:,:,2),H,'replicate');
waitbar(2/3, w);
handles.img(:,:,3) = imfilter(handles.img(:,:,3),H,'replicate');
close(w);
```

```
axes(handles.g2); imshow(handles.img);  
updateg4(handles);  
guidata(hObject, handles);
```

```
function pushbutton64_Callback(hObject, eventdata, handles)
```

```
ac=handles.img;  
gray=rgb2gray(ac);  
f=gray;  
[m,n]=size(f);  
F=fft2(f,m,n);  
F2=fftshift(F);  
axes(handles.g2);  
figure  
imshow(log(1+abs(F2)),[]);  
updateg4(handles)
```

```
function pushbutton65_Callback(hObject, eventdata, handles)
```

```
b=handles.img(:,:,3);  
axes(handles.g2);  
imshow(b);  
r=handles.img(:,:,3);  
g=handles.img(:,:,3);  
b=handles.img(:,:,3);  
x=size(r); x=(1:x(1,2));  
r=mean(r); g=mean(g); b=mean(b);  
axes(handles.g4); plot(x,r,'r');  
hold on  
plot(x,g,'g'); plot(x,b,'b'); hold off;
```

```
function pushbutton66_Callback(hObject, eventdata, handles)
```

```
gr=rgb2gray(handles.img);  
gr(:,:,1)=gr(:,:,1);  
gr(:,:,2)=gr(:,:,1);  
gr(:,:,3)=gr(:,:,1);  
axes(handles.g2);  
imshow(gr);  
r=handles.img(:,:,3);  
g=handles.img(:,:,3);
```

```
b=handles.img(:,:,3);
x=size(r); x=(1:x(1,2));
r=mean(r); g=mean(g); b=mean(b);gr=mean(gr);
axes(handles.g4); plot(x,gr(:,1),'black');
guidata(hObject,handles);
```

```
function pushbutton67_Callback(hObject, eventdata, handles)
```

```
ac=handles.img;
[m n c]=size(ac);
ed=ac;
if c==3
gray=rgb2gray(ac);
ed=gray;
end
edec=edge(ed,'canny');
axes(handles.g2);
imshow(imoverlay(ed,edec,[0,0,1]));
```

```
function pushbutton68_Callback(hObject, eventdata, handles)
```

```
[file,path]=uigetfile('*.*raw','MultiSelect','on');
inita;
files=cellstr(file);
l=length(files);
for ii=1:l
    fi{ii}=strcat(path,'\',files{ii});
    fid=fopen(fi{ii},'rb');
    a=fread(fid);
    status=fclose(fid);
m=256;
n=300;
I=zeros(n,m);
k=1;
    for i=1:n
        for j=1:m
            I(i,j)=a(k);
            k=k+1;
        end
    end
end
J(:,1)=I/255;
```

```
J(:,2)=I/255;
J(:,3)=I/255;
PP{ii}=J;
alevel(ii)=graythresh(I/255);
mmm(ii)=255-mean(a);
end
handles.PP=PP;
bb=mean(alevel);
mm=mean(mmm);
handles.l=l;
handles.mm=mm;
axes(handles.g1)
handles.img=PP{1};
imshow(handles.img);
axes(handles.g2)
imshow(handles.img);
axes(handles.axes6)
plot(1:l,mmm(:));
figure
plot(1:l,mmm(:));
set(handles.listbox1,'string',files);
set(handles.edit33,'string',num2str(bb));
set(handles.edit34,'string',num2str(bb));
    updateg3(handles);
    updateg4(handles);
guidata(hObject,handles);
```

```
function listbox1_Callback(hObject, eventdata, handles)
```

```
ValueNum = get(handles.listbox1,'value');
    cla(handles.g1);
    axes(handles.g1);
PP=handles.PP;
l=handles.l;
    for i=1:l
        switch ValueNum
            case i
                [m n c]=size(PP{i});
                imshow(PP{i})
                handles.img=PP{i};
                handles.img2=PP{i};
```

```
        axes(handles.g2)
        imshow(PP{i})
    end
end
updateg3(handles);
updateg4(handles);
guidata(hObject,handles);
```

```
function pushbutton69_Callback(hObject, eventdata, handles)
```

```
gray=handles.img;
input_image=gray;
output_image=imresize(input_image,1);
[m,n,c]=size(output_image);
x=1:n;
y=1:m;
z=((double(imadjust(output_image(:,:,1),[0 1]))));
z=z/255;
[a,b]=size(z);
figure
surf(x',y',z);
zlim([-2 2])
```

```
function pushbutton70_Callback(hObject, eventdata, handles)
```

```
img=handles.img;
ac=handles.img;
axes(handles.g2)
[m,n,c]=size(img);
if c==3
    gray=rgb2gray(ac);
    z=coloredges(img);
    imshow(z);
else
    imshow(coloredges(ac));
end
```

```
function pushbutton71_Callback(hObject, eventdata, handles)
```

```
ac=handles.img;
[m n c]=size(ac);
```

```
axes(handles.g2)
if c==3
    ac(:,:,1)=histeq(ac(:,:,1));
    ac(:,:,2)=histeq(ac(:,:,2));
    ac(:,:,3)=histeq(ac(:,:,3));
    imshow(ac);
    handles.img=ac;
else
    histgray=histeq(ac);
    imshow(histgray);
    handles.img=histgray;
end
    updateg4(handles);
    guidata(hObject,handles);
```

```
function pushbutton72_Callback(hObject, eventdata, handles)
```

```
    x=get(handles.edit34,'string');
    xx=str2num(x);
    gray=handles.img;
    [m,n,c]=size(gray);
    if c==3
        gray=rgb2gray(gray);
    end
    axes(handles.g2)
    [m,n,c]=size(gray);
    grayy=zeros(m,n);
    grayy=gray(:,:,1);
    for i=1:m
        for j=1:n
            if grayy(i,j)<xx;
                grayy(i,j)=xx;
            end
        end
    end
    imshow(grayy);
    guidata(hObject,handles);
```

```
function pushbutton73_Callback(hObject, eventdata, handles)
```

```
    gray=handles.img;
```

```
input_image=gray;
RUN=1;
    while (RUN==1)
        prompt = {'Enter Motion Length (0-15%):','Enter Motion Angle (0-360):','Resolutions
(prefer 0-1)'};
        dlg_title = 'Enter Motion Parameters: ';
        num_lines = 1;
        def = {'0.4','0.6','0.2'};
        answer = inputdlg(prompt,dlg_title,num_lines,def);
        if (isempty(answer))
            return;
        end
        M1 = str2num(answer{1});
        M2 = str2num(answer{2});
        M3 = str2num(answer{3});
        if ((str2num(answer{1})>=0) & (str2num(answer{1})<=1)&(str2num(answer{2})>=0) &
(str2num(answer{2})<=1)&(str2num(answer{3})>0))
            RUN = 0;
        end
    end
output_image=imresize(input_image,M3);
[m,n,c]=size(output_image);
x=1:n;
y=1:m;
z=((double(imadjust(output_image(:,:,1),[M1 M2]))));
z=z/255;
[a,b]=size(z);
figure
    surf(x',y',z);
    colormap([1 1 0; 0 1 1])
    zlim([-2 2])
```

```
function info_Callback(hObject, eventdata, handles)
```

```
h = waitbar(0,'Please wait...');
steps = 100;
for step = 1:steps
    waitbar(step / steps)
end
close(h)
img=axes(handles.g2);
```

```
figure, imshow(img);  
imageinfo(handles.g2);
```

function pushbutton75_Callback(hObject, eventdata, handles)

```
axes(handles.g2)  
handles.img=getimage(gca);  
axes(handles.g1)  
imshow(handles.img);  
updateg3(handles);  
updateg4(handles);  
s=num2str(size(handles.img));  
guidata(hObject,handles);
```

function pushbutton76_Callback(hObject, eventdata, handles)

```
cla;axes(handles.g2);  
handles.img=handles.PP{ 1 };  
imshow(handles.img);  
axes(handles.g1);  
imshow(handles.img);  
updateg3(handles);  
updateg4(handles);  
s=num2str(size(handles.img));  
guidata(hObject,handles);
```

function pushbutton77_Callback(hObject, eventdata, handles)

```
load 'db\database.mat'  
[filenamec,pathnamec]=uigetfile(strcat(pathname,'\*.jpg'));  
fnamec=strcat(pathnamec,'\',filenamec);  
sz = 512;  
out_fn = outputfiles;  
stage = 4;  
orientation = 6;  
N = 128;  
freq = [0.05 0.4];  
flag = 0;  
j = sqrt(-1);  
for s = 1:stage,  
    for n = 1:orientation,  
        [Gr,Gi] = Gabor(N,[s n],freq,[stage orientation],flag);
```

```
F = fft2(Gr+j*Gi);
F(1,1) = 0;
GW(N*(s-1)+1:N*s,N*(n-1)+1:N*n) = F;
end;
end;
height = 7;
width = 7;
BB=cell(1,1);
for i = 1:1,
    texture = loading(fnamec,sz);
    A = zeros(stage*orientation*2,height*width);
    for h = 1:height,
        for w = 1:width,
            [i h w];
            img = texture((h-1)*64+1:(h-1)*64+128, (w-1)*64+1:(w-1)*64+128);
            F = Fea_Gabor_brodatz(img, GW, N, stage, orientation);
            A(:,(h-1)*width+w) = [F(:,1); F(:,2)];
        end;
    end;
    output = [ extractName(out_fn(i,:)) 'fea'];
    fid = fopen(output, 'w');
    fwrite(fid, A, 'float');
    fclose(fid);
BB=A;
end
currenim=BB;
for i=1:len
    if i==1
        cttotal=c{i};
    end
    if i>1
        cttotal=cttotal+c{i-1};
    end
end
cavg=cttotal/len;
zz=ones(size(A));
for i=1:len
    for j=1:48
        for q=1:49
            cfinal{i}(j,q)=abs((currenim(j,q)-c{i}(j,q)))/cavg(j,q);
        end
    end
end
```

```
        end
    end
    for i=1:len
        for j=1:24
            for q=1:49
                distancei{i}(j,q)=abs(cfinal{i}(j+24,q));
                distancej{i}(j,q)=abs(cfinal{i}(j+24,q));
            end
        end
        distance{i}=distancei{i}+distancej{i};
        finalans{2,i}=double(sum(sum(distance{i})));
        finalans{1,i}=fname{i};
        finalans{2,i}=(100/(100+finalans{2,i}))*100;
        xx(i)=finalans{2,i};
        finalans{2,i}=(strcat(num2str(finalans{2,i}),'%'));
    end
    [b,lx]=sort(xx);
    for i=1:len
        xxx{1,i}=finalans{1,lx(len+1-i)};
        xxx{2,i}=b(len+1-i);
        xxx{2,i}=(strcat(num2str(b(len+1-i)),'%'));
    end
    figure
    subplot(2,5,1);imshow(imread(fnamec));
    title('original')
    subplot(2,5,2);imshow(imread(strcat(pathname,'\',xxx{1,1})));
    title([xxx{2,1}','/',xxx{1,1}])
    subplot(2,5,3);imshow(imread(strcat(pathname,'\',xxx{1,2})));
    title([xxx{2,2}','/',xxx{1,2}])
    subplot(2,5,4);imshow(imread(strcat(pathname,'\',xxx{1,3})));
    title([xxx{2,3}','/',xxx{1,3}])
    subplot(2,5,5);imshow(imread(strcat(pathname,'\',xxx{1,4})));
    title([xxx{2,4}','/',xxx{1,4}])
    subplot(2,5,6);imshow(imread(strcat(pathname,'\',xxx{1,5})));
    title([xxx{2,5}','/',xxx{1,5}])
    subplot(2,5,7);imshow(imread(strcat(pathname,'\',xxx{1,6})));
    title([xxx{2,6}','/',xxx{1,6}])
    subplot(2,5,8);imshow(imread(strcat(pathname,'\',xxx{1,7})));
    title([xxx{2,7}','/',xxx{1,7}])
    subplot(2,5,9);imshow(imread(strcat(pathname,'\',xxx{1,8})));
    title([xxx{2,8}','/',xxx{1,8}])
```

```
subplot(2,5,10);imshow(imread(strcat(pathname,'\,xxx{1,9})));
title([xxx{2,9}','/',xxx{1,9}])
xxx
```

```
function pushbutton78_Callback(hObject, eventdata, handles)
```

```
bw=handles.img;
bw=im2bw(bw);
axes(handles.g2)
imshow(bw)
```

```
function pushbutton79_Callback(hObject, eventdata, handles)
```

```
axes(handles.g1)
handles.img=getimage(gca);
axes(handles.g2)
imshow(handles.img);
updateg4(handles);
guidata(hObject,handles);
```

```
function pushbutton72_ButtonDownFcn(hObject, eventdata, handles)
```

```
function popuptmenu5_Callback(hObject, eventdata, handles)
```

```
ac=handles.img;
ed=ac;
[m n c]=size(ac);
if c==3
gray=rgb2gray(ac);
ed=gray;
end
val=get(hObject,'Value');
```

```
switch val
```

```
case 1
```

```
    warndlg('Please select a method: ');
```

```
case 2    %canny
```

```
    edec=edge(ed,'canny');
    axes(handles.g2);
    imshow(edec);
    updateg4(handles);
```

```
case 3 % sobel
    axes(handles.g2)
    [m,n,c]=size(ac);
    if c==3
        gray=rgb2gray(ac);
        z=coloredges(ac);
    %     z=z/max(z(:));
        imshow(im2bw(z));
    else
        edec=edge(ed,'sobel');
        imshow(edec);
    end
    updateg4(handles);
case 4 % prewitt
    edec=edge(ed,'prewitt');
    axes(handles.g2);
    imshow(edec);
    guidata(hObject, handles);
    updateg4(handles);

case 5 % roberts
    edec=edge(ed,'roberts');
    axes(handles.g2);
    imshow(edec);
    guidata(hObject, handles);
    updateg4(handles);

case 6 % log
    edec=edge(ed,'log');
    axes(handles.g2);
    imshow(edec);
    guidata(hObject, handles);
    updateg4(handles);

case 7 % 8x8
    handles.res=handles.img;
    axes(handles.g2); cla;
    handles.res=imresize(imresize(handles.res,1/32),32);
    imshow(handles.res);
    guidata(hObject, handles);
```

```
handles.img=handles.res; updateg4(handles);  
end  
  
function pushbutton80_Callback(hObject, eventdata, handles)  
  
    i=handles.img;  
    I=rgb2gray(i);  
    BW=im2bw(I);  
    figure,imshow(BW)  
    [n1 n2]=size(BW);  
    r=floor(n1/10);  
    c=floor(n2/10);  
    x1=1;x2=r;  
    s=r*c;  
  
    for i=1:10  
        y1=1;y2=c;  
        for j=1:10  
            if (y2<=c | y2>=9*c) | (x1==1 | x2==r*10)  
                loc=find(BW(x1:x2, y1:y2)==0);  
                [o p]=size(loc);  
                pr=o*100/s;  
                if pr<=100  
                    BW(x1:x2, y1:y2)=0;  
                    r1=x1;r2=x2;s1=y1;s2=y2;  
                    pr1=0;  
                end  
            end  
            y1=y1+c;  
            y2=y2+c;  
        end  
        x1=x1+r;  
        x2=x2+r;  
    end  
    L = bwlabel(BW,8);  
    BB = regionprops(L, 'BoundingBox');  
    BB1=struct2cell(BB);  
    BB2=cell2mat(BB1);  
    [s1 s2]=size(BB2);  
    mx=0;
```

```
for k=3:4:s2-1
    p=BB2(1,k)*BB2(1,k+1);
    if p>mx & (BB2(1,k)/BB2(1,k+1))<1.8
        mx=p;
        j=k;
    end
end
figure,imshow(I);
hold on;
rectangle('Position',[BB2(1,j-2),BB2(1,j-1),BB2(1,j),BB2(1,j+1)],'EdgeColor','r')
```

```
function pushbutton81_Callback(hObject, eventdata, handles)
```

```
    camera;
```

```
function pushbutton82_Callback(hObject, eventdata, handles)
```

```
    axes(handles.g2);
    handles.img=imtophat(handles.img,strel('disk',10));
    imshow(handles.img)
```

```
function pushbutton83_Callback(hObject, eventdata, handles)
```

```
    ori=handles.img;
    [m n c]=size(handles.img);
    if c==3
        ori=rgb2gray(ori);
    end
    gim=im2double(ori);
    ag=get(handles.slider8,'Value')
    [Eim,Oim,Aim]=spatialgabor(gim,3,ag,0.5,0.5,1);
    axes(handles.g2)
    imshow(Aim);
```

```
function pushbutton84_Callback(hObject, eventdata, handles)
```

```
    [m n c]=size(handles.img);
    img=handles.img;
    img(:,:,1)=handles.img(:,:,1);
    img(:,:,2)=handles.img(:,:,1);
    img(:,:,3)=handles.img(:,:,1);
    handles.img=img;
```

```
guidata(hObject, handles);
    updateg4(handles);
    updateg3(handles);
function pushbutton85_Callback(hObject, eventdata, handles)

bw=handles.img;
[labeled,numObjects] = bwlabel(bw,8);
set(handles.edit33,'string',num2str(numObjects));
dataa=regionprops(labeled,'basic');
for i=1:numObjects
    CC(i)=dataa(i).Area;
end
avarea=(sum(CC))/numObjects;
BW = handles.img;
[B,L,N] = bwboundaries(BW);
axes(handles.g2)
figure;imshow(BW); hold on;
for k=1:length(B),
    boundary = B{k};
    if(k > N)
        plot(boundary(:,2),...
            boundary(:,1),'b','LineWidth',4);
        title(['cells : ',num2str(numObjects),'/','Average area (in pixels):',num2str(avarea)])
        set(gca,'fontsize',15)
    else
        plot(boundary(:,2),...
            boundary(:,1),'b','LineWidth',4);
        title(['cells : ',num2str(numObjects),'/','Average area (in pixels):',num2str(avarea)])
        set(gca,'fontsize',18)
    end
end
guidata(hObject, handles);

function pushbutton86_Callback(hObject, eventdata, handles)

ac=handles.img;
gray=rgb2gray(ac);
f=gray;
[m,n]=size(f);
F1=abs(fft2(f));
F2=fftshift(F1);
```

```
axes(handles.g2);  
figure  
imshow(uint8(log(1+abs((F2)))),[]);  
guidata(hObject, handles);  
    updateg4(handles);  
    updateg3(handles);
```

```
function pushbutton87_Callback(hObject, eventdata, handles)
```

```
    load 'db\database.mat'  
    [filenamec,pathnamec]=uigetfile(strcat(pathname,'\*.jpg'));  
    fnamec=strcat(pathnamec,'\',filenamec);  
    sz = 512;  
    out_fn = outputfiles;  
    stage = 4;  
    orientation = 6;  
    N = 128;  
    freq = [0.05 0.4];  
    flag = 0;  
    j = sqrt(-1);  
    for s = 1:stage,  
        for n = 1:orientation,  
            [Gr,Gi] = Gabor(N,[s n],freq,[stage orientation],flag);  
            F = fft2(Gr+j*Gi);  
            F(1,1) = 0;  
            GW(N*(s-1)+1:N*s,N*(n-1)+1:N*n) = F;  
        end;  
    end;  
    height = 7;  
    width = 7;  
    BB=cell(1,1);  
    for i = 1:1,  
        texture = loading(fnamec,sz);  
        A = zeros(stage*orientation*2,height*width);  
        for h = 1:height,  
            for w = 1:width,  
                [i h w];  
                img = texture((h-1)*64+1:(h-1)*64+128, (w-1)*64+1:(w-1)*64+128);  
                F = Fea_Gabor_brodatz(img, GW, N, stage, orientation);  
                A(:,(h-1)*width+w) = [F(:,1); F(:,2)];  
            end;  
        end;  
    end;
```

```
end;
output = [ extractName(out_fn(i,:)) 'fea'];
fid = fopen(output, 'w');
fwrite(fid, A, 'float');
fclose(fid);
BB=A;
end
currentim=BB;
for i=1:len
if i==1
    ctot=c{i};
end
if i>1
ctotal=ctotal+c{i-1};
end
end
cavg=ctotal/len;
zz=ones(size(A));
for i=1:len
    for j=1:48
        for q=1:49
            cfinal{i}(j,q)=abs((currentim(j,q)-c{i}(j,q))/cavg(j,q));
        end
    end
end
for i=1:len
    for j=1:24
        for q=1:49
            distancei{i}(j,q)=abs(cfinal{i}(j+24,q));
            distancej{i}(j,q)=abs(cfinal{i}(j+24,q));
        end
    end
    distance{i}=distancei{i}+distancej{i};
    finalans{2,i}=double(sum(sum(distance{i})));
    finalans{1,i}=fname{i};
    finalans{2,i}=(100/(100+finalans{2,i}))*100;
    xx(i)=finalans{2,i};
    finalans{2,i}=(strcat(num2str(finalans{2,i}),'%'));
end
xxx=finalans;
```

figure

```
subplot(2,5,1);imshow(imread(fnamec));
title('original')
subplot(2,5,2);imshow(imread(strcat(pathname,'\',xxx{1,1})));
title([xxx{2,1}','/',xxx{1,1}])
subplot(2,5,3);imshow(imread(strcat(pathname,'\',xxx{1,2})));
title([xxx{2,2}','/',xxx{1,2}])
subplot(2,5,4);imshow(imread(strcat(pathname,'\',xxx{1,3})));
title([xxx{2,3}','/',xxx{1,3}])
subplot(2,5,5);imshow(imread(strcat(pathname,'\',xxx{1,4})));
title([xxx{2,4}','/',xxx{1,4}])
subplot(2,5,6);imshow(imread(strcat(pathname,'\',xxx{1,5})));
title([xxx{2,5}','/',xxx{1,5}])
subplot(2,5,7);imshow(imread(strcat(pathname,'\',xxx{1,6})));
title([xxx{2,6}','/',xxx{1,6}])
subplot(2,5,8);imshow(imread(strcat(pathname,'\',xxx{1,7})));
title([xxx{2,7}','/',xxx{1,7}])
subplot(2,5,9);imshow(imread(strcat(pathname,'\',xxx{1,8})));
title([xxx{2,8}','/',xxx{1,8}])
subplot(2,5,10);imshow(imread(strcat(pathname,'\',xxx{1,9})));
title([xxx{2,9}','/',xxx{1,9}])
xxx;
```

function pushbutton88_Callback(hObject, eventdata, handles)

```
axes(handles.g1);
handles.img=getimage(gca);
figure
imshow(handles.img);
axes(handles.g2);
handles.img=getimage(gca);
figure
imshow(handles.img);
axes(handles.g3);
figure
r=handles.img(:,:,1);
g=handles.img(:,:,1);
b=handles.img(:,:,1);
[m n c]=size(handles.img);
if c==3
```

```
    r=handles.img(:,:,1);
    g=handles.img(:,:,2);
    b=handles.img(:,:,3);
end
x=size(r); x=(1:x(1,2));
r=mean(r); g=mean(g); b=mean(b);
plot(x,r,'r');
hold on
plot(x,g,'g'); plot(x,b,'b'); hold off;
axes(handles.g4);
figure
r=handles.img(:,:,1);
g=handles.img(:,:,1);
b=handles.img(:,:,1);
[m n c]=size(handles.img);
if c==3
    r=handles.img(:,:,1);
    g=handles.img(:,:,2);
    b=handles.img(:,:,3);
end
x=size(r); x=(1:x(1,2));
r=mean(r); g=mean(g); b=mean(b);
    plot(x,r,'r');
hold on
plot(x,g,'g'); plot(x,b,'b'); hold off;
(handles.axes6);
l=handles.l;
PP=handles.PP;
figure
for ii=1:l
    inew=PP{ii};
    [x y z]=size(PP{ii});
    if z==3
        r=inew(:,:,1);
        g=inew(:,:,2);
        b=inew(:,:,3);
        rl(ii)=mean(mean(r),2); gl(ii)=mean(mean(g),2); bl(ii)=mean(mean(b),2);
    end

    if z==1
        r=handles.img(:,:,1);
```

```
g=handles.img(:,:,1);
b=handles.img(:,:,1);
rl(ii)=mean(mean(r),2); gl(ii)=mean(mean(g),2); bl(ii)=mean(mean(b),2);
end

end

x=1:l;
plot(x,rl,'r');
hold on
plot(x,gl,'g'); plot(x,bl,'b'); hold off;
```

```
function pushbutton89_Callback(hObject, eventdata, handles)
```

```
I=handles.img;
[m n c]=size(size(I));
if c==3
I = rgb2gray(I);
end

level = graythresh(I);
I = im2bw(I,level)*255;
I2 = imcomplement(I);
I3 = imhmin(I2,20);
L = watershed(I3);
figure;imshow(I);
figure;imshow(label2rgb(L,'spring','c','shuffle'))
s=max(max(L));
area=0;
for i=1:s
    n=size(find(L ==i));
    area = area+n(1,1);
end
totalarea=int16(area*100/numel(L));
eacharea=int16(area/s);
set(gca,'fontsize',15)
title(['Seg.: ' num2str(s) ' Area:' num2str(totalarea) '% Each:' num2str(eacharea)]);
stats = regionprops(L, 'Area', 'Orientation');
area = [stats.Area];
orient = [stats.Orientation];
figure;plot(area, orient, 'b*');
title('Relationship of Skin Orientation to Area');
```

```
xlabel('Skin Segment area (pixels)');  
ylabel('Skin Segment orientation (degrees)');
```

```
function slider8_Callback(hObject, eventdata, handles)
```

```
    ori=handles.img;  
    [m n c]=size(handles.img);  
    if c==3  
        ori=rgb2gray(ori);  
    end  
    gim=im2double(ori);  
    ag=get(handles.slider8,'Value');  
    [Eim,Oim,Aim]=spatialgabor(gim,3,ag,0.5,0.5,1);  
    axes(handles.g2)  
    imshow(Aim);
```

```
function slider8_CreateFcn(hObject, eventdata, handles)
```

```
    equal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor',[.9 .9 .9]);  
end
```