HOKKAIDO UNIVERSITY

Title	Automatic Joint Space Narrowing Quantification in Radiographic Images with Sub-pixel Accuracy for Rheumatoid Arthritis
Author(s)	欧, 亜非
Degree Grantor	北海道大学
Degree Name	博士(工学)
Dissertation Number	甲第15536号
Issue Date	2023-03-23
DOI	https://doi.org/10.14943/doctoral.k15536
Doc URL	https://hdl.handle.net/2115/89709
Type	doctoral thesis
File Information	Yafei_Ou.pdf

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

令和四年度 博士学位論文

関節リウマチ診断に向けたX線画像における関節狭小化の自動定量化

Automatic Joint Space Narrowing Quantification in Radiographic Images with Sub-pixel Accuracy for Rheumatoid Arthritis

2023年1月

北海道大学大学院 情報科学院 情報科学専攻
欧 亜非

指導教員 池辺 将之 教授

Contents

Abstract

Rheumatoid Arthritis (RA) is a chronic autoimmune disease characterized by synovitis that primarily affects peripheral synovial joints, like fingers, wrists, and feet. The major radiographic changes on hand, wrist, and feet joints are cartilage damage and bone destruction (like bone erosion and Joint Space Narrowing (JSN)). Those damages and destruction typically lead to painful joints, progressive joint destruction, and deformity, followed by functional limitation and severe disability. There are substantial evidence that RA can be managed in a low level of disease activity and clinical remission with disease-modifying antirheumatic drugs. Early diagnosis by precise quantification of subtle radiographic changes is essential for successful treatment, as it can improve outcomes and effectively manage the progression of RA. Radiology plays a critical role in the diagnosis and monitoring of RA. Clinical radiologist/rheumatologist can assess the radiographic progression of RA by using the Sharp/van der Heijde (SvdH) scoring method. This method relies on scoring the radiographies by subjectively assessing JSN and bone erosion of 38 hand or foot joints. The joint space is one of the most important indicators for the diagnosis and monitoring of RA, and it has attracted the extensive attention of many researchers resulting in great efforts to study the automatic quantification of joint space in RA. According to the methodology and its output metric, those previous works on joint space quantification in RA can be grouped into two frameworks, classification based SvdH scoring method, and margin detection based Joint Space Width (JSW) quantification method.

However, classification based SvdH scoring method has only five levels, which limits the sensitivity and timeliness of the tool. Whereas, the margin detection based JSW quantification works can achieve only pixel-level accuracy. Limited by the current spatial resolution of radiographic imaging, JSN progression of RA can be less than one pixel per year with universal spatial resolution. Insensitive monitoring of JSN can hinder the radiologist/rheumatologist from making a definitive and timely clinical judgment.

In this thesis, we propose a novel and sensitive framework which aims to automatically quantify JSN progression in the early RA. The proposed framework implements fully automatic quantization of JSN progression in RA, which can be broadly grouped into two parts, joint position detection, and JSN progression quantification. For joint location detection, we propose and compare two methods, a classification based joint location detection method and a key point detection based joint location detection method. The two joint location detection methods show different strengths and weaknesses in our experiments. For joint space quantification, we propose a new framework by calculating the relative widths of the joint space based on an image registration method. Take a Metacarpophalangeal (MCP) joint as an example, our methodology can be understood in four steps. (i) Segment the proximal phalanx bone and metacarpal bone based on gradient information. (ii) Combining Discrete Fourier Transform (DFT) and the joint segmentation curve, segment the proximal phalanx bone and metacarpal bone in the phase spectrum. (iii) Measure the movement of the proximal phalanx bone and metacarpal bone between baseline and follow-up radiographs respectively by using the image registration algorithm. (iv) Calculate the displacement difference between the proximal phalanx bone and metacarpal bone to get JSN progression.

Also in this thesis, we have improved Full Image Phase Only Correlation (FIPOC) by adding a phase spectrum segmentation step, so that it can measure the displacements of multiple regions at the same time. We named the improved FIPOC algorithm as Partial Image Phase Only Correlation (PIPOC). As an image registration algorithm, PIPOC is used in JSN progression quantification. Compared with FIPOC, PIPOC can effectively avoid the impact of the segmentation and in-painting process and reduce the phase dispersion in the phase difference spectrum. In our phantom study, PIPOC has a much lower mean error than FIPOC. This work aims for computer-aided monitoring and diagnosis of RA. Our work measures JSN progression between a baseline and its follow-up finger joint images by using the phase spectrum in the frequency domain. In our phantom studies, and experiments using clinical dataset our algorithm can measure the displacements of upper and lower bones with sub-pixel accuracy. The measured mean error of our algorithm is in range of 11.9% - 35.0% in comparison to manual measurements using radiographic phantom datasets, in range of 6.0% - 16.1% when using tomosynthesis phantom datasets, and with a Standard Deviation (SD) of 0.0519 mm when using radiographic clinical dataset. Considering that previous works can best achieve only pixel-level accuracy, our extensive experiments demonstrate that the proposed image registration based JSN progression quantization framework promises high precision monitoring when compared to two mainstream related works; the margin detection based

JSW quantification framework and image classification based SvdH scoring framework.

These advantages indicate that our proposed framework can fill the gap in monitoring the JSN progression with high sensitivity. Thereby it has a broad application prospect in the monitoring of RA. With the sub-pixel accuracy far beyond usual manual measurements, we are optimistic that the proposed work is a promising scheme for automatically quantifying JSN progression. Currently, Machine Learning (ML) is applied to difficult tasks in medical image analysis. We anticipate future studies in this direction. Our experiments in this study have shown the superiority of image registration based JSN progression quantification framework in RA monitoring compared to the current most popular classification based SvdH scoring framework, and the margin detection based JSW quantification framework. Those ML-based image registration algorithms can be used for JSN quantization in RA. To address the posture (finger movement) related constraints and inconsistent joint angle which is likely to result in mismatched registration. Given the advantages of ML, it may be possible to achieve higher robustness (lower mismatching ratios) at the expense of a small amount of accuracy. Furthermore, we can quantify JSN by ML using the image features extracted by our work, this can improve the overall performance of the algorithm.

Key words

Rheumatoid Arthritis, Frequency Domain Analysis, Joint Space Narrowing, Tone mapping, Phantom Imaging, Radiology, Computer-aided Diagnosis.

Acknowledgement

A great number of people have helped and supported me during my doctoral research. While it is not possible to name everyone, I would like to express my sincere gratitude to all the people who have contributed to this thesis in various ways.

First, I am extremely grateful to my supervisor Prof. Masayuki Ikebe for his great guidance and advice for my study and research. It has been a pleasure to work with and learn from him and I would like to thank him for his unfailing patience. I would like to sincerely thank Prof. Tamotsu Kamishima of the Faculty of Health Sciences, Hokkaido University. His extensive knowledge of radiology and professional knowledge of rheumatoid arthritis is indispensable, and his guidance and support throughout my doctoral project have been invaluable. I would also like to thank Dr. Prasoon Ambalathankandy for his great guidance and advice. He has always been available to share experiences and discuss practicalities with me at length. I consider myself very lucky to have had the opportunity to be mentored by them. They help me have a deeper understanding and thinking in my research. My thesis would have been impossible without their help and support.

This work was supported by great collaborators, Prof. Shinya Takamaeda from the Department of Computer Science, The University of Tokyo, Prof. Masato Motomura from the Institute of Innovative Research, Tokyo Institute of Technology, and Prof. Tetsuya Asai from the Graduate School of Information Science and Technology, Hokkaido University. I would like to thank them for their valuable comments and suggestions in the writing of my papers. Prof. Kenichi Tamura Department of Mechanical Engineering, Nihon University, who provided the phantom material for the experiments, which made the experiments in the thesis more comprehensive.

I would like to sincerely thank Dr. Akira Sagawa from Sagawa Akira Rheumatology Clinic (Sapporo, Japan), Dr. Masaya Mukai from Sapporo City General Hospital (Sapporo, Japan) and Dr. kazuhide Tanimura from Hokkaido Medical Center for Rheumatic Diseases (Sapporo, Japan) for image data preparation. Thank you for your help and support to enable this work to be validated on clinical data.

I would like to sincerely thank my thesis examination committee of Prof. Akihisa Tomita, Prof. Tetsuya Asai, Prof. Junichi Motohisa, and Prof. Tamotsu Kamishima for their detailed review, constructive criticism and excellent advice during the preparation of this thesis.

Thanks also to members of the Functional Communication Devices and Circuits Laboratory, who have helped me in my research and provided me with a great environment to have many fantastic discussions. Thanks for their care during these five and half years. There are too many to name. I would like to thank Prof. Masamichi Akazawa, Mr. Yuri Kanazawa, Ms. Pitchayapatchaya Srikram, Mr. Takahiro Ikegami, Mr. Seiya Kawada, Mr. Ryunosuke Furuya, Mr. Naoto Goto, and Mr. Gen Ota. My initial training year was spent at Laboratories for Advanced LSI Engineering lead by Prof. Masato Motomura and Prof. Tetsuya Asai. I would like to thank my lab mates for the great effort they made to integrate me into a foreign environment. I would like to thank Prof. Shinya Takamaeda, Dr. Kodai Ueyoshi, Prof. Kota Ando, Mr. Seokjin Na, Mr. Takeshi Shimada, Ms. Sayuri Yokoyama, and Ms. Yuki Miura. I would also like to thank Ms. Cheiko Akiyama of RCIQE, Ms. Manami Ueno of IST for their timely administrative support.

Finally, and most importantly, I would like to thank my parents and all my friends for their unwavering support.

This work was financially supported, in part by JST SPRING under Grant JPMJSP2119, in part by the Japan Society for the Promotion of Science (JSPS) Grants-in-Aid for Scientific Research (KAKENHI) under Grant 18H0321309, Grant 18H05288 and Grant 21K07611, in part by JSPS Fellows under Grant 19J14105.

Abbreviations

AdaBoost	Adaptive Boosting
AR	Augmented Reality
ASIC	Application-Specific Integrated Circuit
ASIP	Application-Specific Instruction set Processor
CMOS	Complementary Metal–Oxide–Semiconductor
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CT value	X-ray absorption coefficient
DFT	Discrete Fourier Transform
DICOM	Digital Imaging and Communications in Medicine
DIP	Distal Interphalangeal
DNN	Deep Neural Network
DR	Dynamic Range
FFT	Fast Fourier Transform
FIPOC	Full Image Phase Only Correlation
FPGA	Field-Programmable Gate Array
FPS	Frames Per Second
FT	Fourier Transform
GPU	Graphics Processing Unit

HDR	High Dynamic Range
HE	Histogram Equalization
HVS	Human Visual System
HW	Hardware
IDFT	Inverse Discrete Fourier Transform
IFFT	Inverse Fast Fourier Transform
IP	Interphalangeal
IQA	Image Quality Assessment
JSN	Joint Space Narrowing
JSW	Joint Space Width
LDR	Low Dynamic Range
LHE	Local Histogram Equalization
LUT	Lookup Table
MCP	Metacarpophalangeal
ML	Machine Learning
PC	Personal Computer
PIP	Proximal Interphalangeal
PIPOC	Partial Image Phase Only Correlation
POC	Phase Only Correlation
PSNR	Peak Signal to Noise Ratio
RA	Rheumatoid Arthritis
RIPOC	Rotation Invariant Phase Only Correlation
RMSD	Root-mean-square Deviation
RMSE	Root-mean-square Error
SD	Standard Deviation

SDR	Standard Dynamic Range
SoC	System on Chip
SP	Signal Processing
SSIM	Structural Similarity Index Measure
SvdH	Sharp/van der Heijde
SVM	Support Vector Machine
SW	Software
TM	Tone Mapping
TMA	Titanium Medical Apatite
TMO	Tone Mapping Operator
TMQI	Tone Mapped Image Quality Index
VR	Virtual Reality
VSoC	Vision System on Chip

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Rheumatoid arthritis

RA is a progressive, chronic autoimmune disease characterized by synovitis that can ultimately cause deformities and ankylosis in peripheral synovial joints and impair the movement and flexibility of digits, and as well as the patient's whole hand. The incidence of RA over the adult population in developed regions is relatively stable. Previous studies from Japan suggested a relatively higher incidence of the disease, with a prevalence rate of 0.3% [1, 2]. In Japan, there are about 0.6 to 1 million people with RA, 10 million osteoarthritis patients and 30 million potential arthritis patients [3]. Related works investigated the relationship between sex and incidence of RA [5–7]. The sex ratio varies in most studies from about 2:1 to about 3:1. RA patients

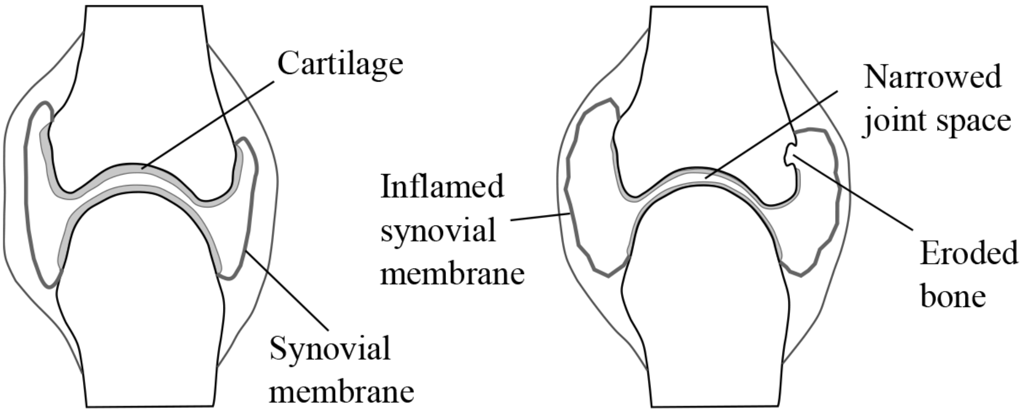


Figure 1.1: Schematic of a healthy joint (left) and a joint affected by RA (right). [4]

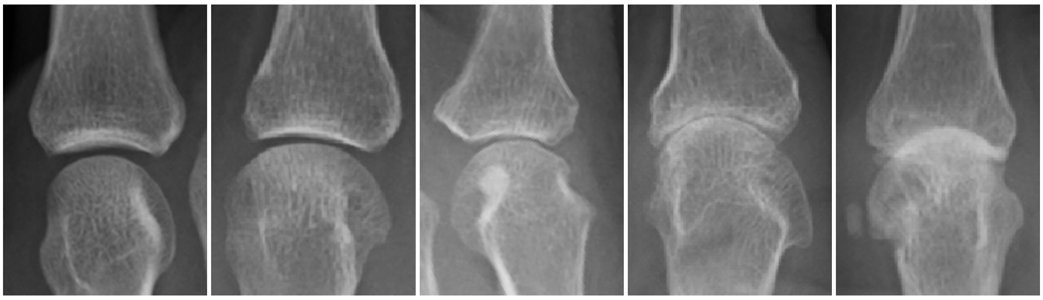


Figure 1.2: The progression of JSN in RA.

need at least one million Japanese yen for medicines every year [8]. RA can cause a huge economic burden for patients.

The major radiographic changes on hand, wrist and feet joints are cartilage damage and bone destruction (like bone erosion and JSN), as show in Fig. 1.1. With time progressing, the joint space of RA patient will gradually become narrow, as show in Fig 1.2. Narrowing and destruction typically lead to painful joints, progressive joint destruction, deformity, followed by functional limitation and severe disability [9, 10]. There are substantial evidence that RA can be managed in a low level of disease activity and clinical remission with disease-modifying antirheumatic drugs [11, 12]. Early diagnosis by precise quantification of subtle radiographic changes is essential for successful treatone mappingent, as it can improve outcomes and effectively manage the progression of RA [11, 12].

1.2 Medical image analysis in RA

Radiology plays a crucial role in diagnosis and monitoring of RA. Clinical radiologist/rheumatologist can assess the radiographic progression of RA by using SvdH scoring method. This method relies on scoring of the radiographies by subjectively assessing JSN and bone erosion of 38 hand or foot joints [13]. As one of the most important indicators for the diagnosis and monitoring of RA, the joint space has always attracted extensive attention of researchers. In recent years, researchers have invested great efforts to study automatic quantification of joint space in RA [14–29]. The JSN progression quantification pipline in radiographs is performed in two steps; joint position detection and joint space quantification.

1.2.1 Related works about finger joint position detection

The earliest studies about finger joint location detection were based on using pixel information. Those algorithms extracted the finger midlines based on ridge detection, thus, finger joint location can be detected according to the gradient or intensity information of finger midline [21]. However, these method may break finger midline at the MCP joint because of decrease in bone density. This may lead to mismatch of joint position for the following reasons: (i) Bone overlap caused by finger bending in the vertical plane. (ii) Marginal density decrease caused by ankylosis or complete luxation [21].

In recent years, ML based methods have become a very important tool to solve complex medical image processing tasks [30]. It is widely used in image segmentation [31], computer aided diagnosis [32], image registration [33] and others. For finger joint detection, there are some ML-based studies utilizing key point detection for CNN [17,24], Support Vector Machine (SVM) [22,26]

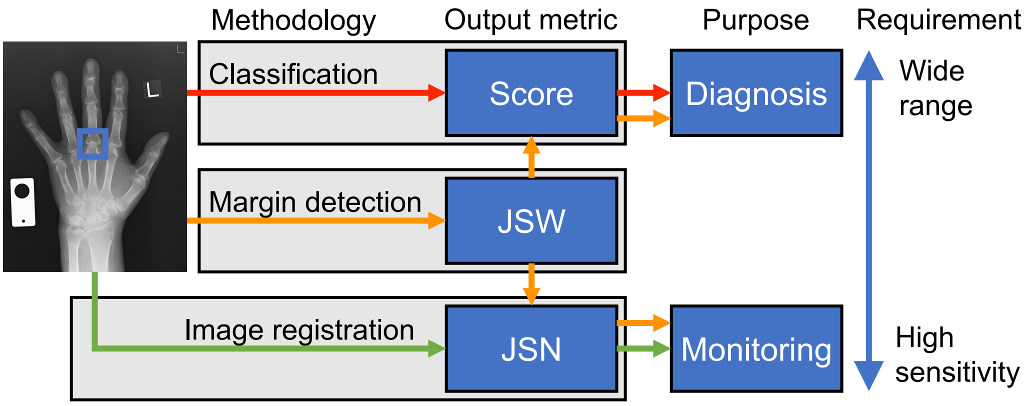


Figure 1.3: A schematic diagram of the classification of joint space quantification works in the computer-aided diagnosis and monitoring of RA. According to the methodology and its output metric, joint space quantification works can be broadly grouped into three groups, ML-based classification based SvdH scoring method (red lines), margin detection based JSW quantification method (orange lines) and image registration based JSN progression quantification method (green lines, the framework we proposed). JSW can be used to score SvdH score to qualitative diagnosis, or calculate the JSN to quantitative monitoring. Diagnosis methodologies should be used in all stages of RA, which have higher requirement on the detectable range. Meanwhile, monitoring methodologies require higher timeliness and sensitivity.

and Haar-like AdaBoost [25, 34].

1.2.2 Literature survey of joint space quantification in RA

As shown in Fig. 1.3, according to the nature of purpose, previous works on joint space quantification in RA can be grouped into two groups, qualitative diagnosis and quantitative monitoring. Diagnosis is a qualitative judgment of the RA stage, usually based on the absolute width of the joint space. Monitoring is the basis for RA patient Dynamic Range (DR) management, which requires high sensitivity quantitative quantification, usually based on the relative narrowing of the joint space.

In the literature, the earliest joint space quantification method detects the upper and lower bone margins to measure the JSW (orange lines in Fig. 1.3). According to the margin detection method, they can be broadly grouped into two groups; supervised ML-based [14, 17] and image features based [21], such as intensity, gradient, derivative or differential. Take Huo's work as an example [21] (Fig. 1.4 Left), this work can be performed as follow: (i) Detect bone margin by using intensity and gradient information. (ii) Fit polynomial functions to bone margin curves. (iii) Quantify JSW according to the distance between polynomial functions. Peloschek *et al* in [16, 17] train a point distribution model by active shape model, a kind of marching learning algorithm (Fig. 1.4 Right). And automatic joint space measurement is performed using key points of bone margin.

As shown in Fig. 1.3, margin detection based JSW quantification methods can combine both qualitative diagnosis and quantitative monitoring. Accord-

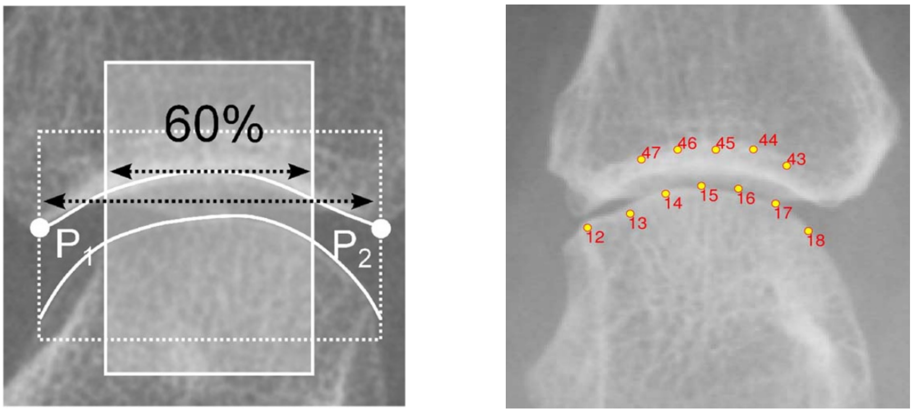


Figure 1.4: The algorithm principle of previous studies. Left: [21]. Right: [16, 17]

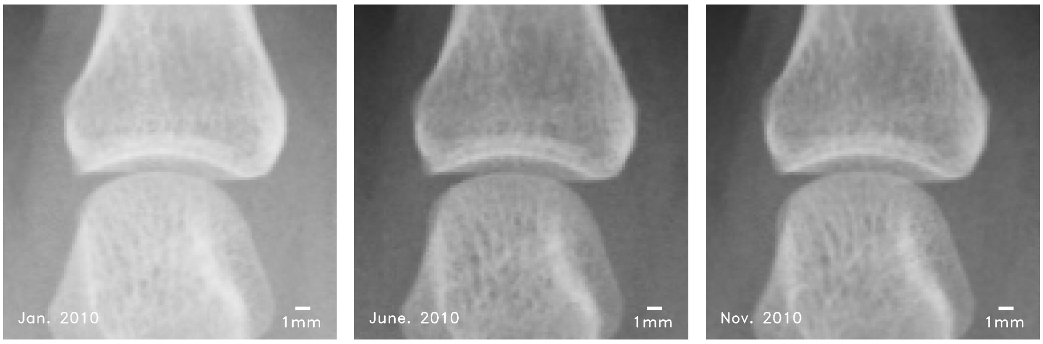


Figure 1.5: JSN progression of a MCP joint for little finger over a period of 10 months. From left to right the images are: baseline, five-month, and ten-month images (spatial resolution: 0.175 mm/pixel). Usually, JSN progression is less than one pixel per year, therefore, it is difficult for radiologist/rheumatologist to see. Then, operating with an algorithm with pixel level accuracy to quantify JSN progression over a period of one year can be ineffective. JSN progression measured for five and ten months X-rays relative to baseline using our method are -0.111 pixel and 0.213 pixel respectively.

ing to the quantified JSW combined with SvdH scoring standard, a qualitative diagnosis of the RA stage can be obtained. And the JSN can be obtained by calculating the difference of the JSW during the two imaging to realize the quantitative monitoring of RA. However, margin detection based JSW quantification works have three main limitations: (i) Margin detection based studies [14,17,21] can best achieve only pixel-level accuracy (please see § 5.2 for more details). Furthermore, limited by the current spatial resolution of radiographic imaging, JSN progression over a period of one year can be less than one pixel, as show in Fig. 1.5. This means that the pixel-level accuracy algorithm requires more time to detect for any changes in the joint space. Nevertheless, this can lead to insensitive monitoring of JSN progression, and this may hinder the radiologist/rheumatologist from making a proper monitoring in the “window of opportunity” [35–37]. (ii) Considering that margin detection require clear margin information, it cannot be used in the advanced RA when the bone margin is destroyed. (iii) Fundamentally detecting the upper bone margin accurately is a challenge, which is known to be affected by false edges [21].

At present, ML-based algorithms have increasingly become the mainstream of computer-aided diagnosis in RA [38, 39]. The popular research direction in joint space quantification is SvdH scoring method [13] based on ML image classification (red lines in Fig. 1.3). Those works can be performed as follow: (i) Manually classify the joint images into five levels from level 0 to level 4 (most severe level) according to the JSN based on SvdH scoring standard. (ii) Train a joint image classifier based on ML algorithms like CNN [24, 27] or SVM [26].

ML-based classification can quickly determine the RA condition in early RA and advanced RA. Considering that ML-based classification is proven now, these works can achieve very low false negative and false positive rates. Nevertheless, SvdH scoring standard with only five levels limits the sensitivity and timeliness of the tool. If it is not feasible to increase the number of levels in the scoring standard, that would make it difficult for radiologists/rheumatologists to make accurate scores for training data. But blindly increasing the number of levels will increase the difficulty and error rate for radiologists/rheumatologists to manually label data. It is precise because of this conflict that the upper limit of the sensitivity of these algorithms is severely limited. Compared to margin detection based JSW quantification works, these works greatly improve the detectable range at the expense of low sensitivity. And abandon the application of monitoring and strengthen the qualitative diagnosis in early RA and advanced RA.

We proposed a JSN progression quantification method by calculating the relative widths of the joint space based on an image registration method (green lines in Fig. 1.3). Take a MCP joint as an example, this work can be performed as follow: (i) Segment the proximal phalanx bone and metacarpal bone in the spatial domain. (ii) Fill vacant space by using an image inpainting algorithm. (iii) Measure the displacements of the proximal phalanx bone and metacarpal bone between a baseline and its follow-up finger joint images respectively by using an image registration algorithm. (iv) Calculate the displacement difference between the proximal phalanx bone and metacarpal bone to get JSN progression.

Our experiments show that the image registration based JSN progression quantification framework has the potential for higher sensitivity and lower mean error compared to the margin detection based JSW quantification framework and image classification based SvdH scoring framework. These advantages indicate that our proposed framework can fill the gap in monitoring the JSN progression with high sensitivity. Thereby it has a broad application prospect in the monitoring of RA. Nevertheless, the image registration based JSN progression quantification framework has two limitations: (i) This framework can only calculate the difference of JSW between the baseline and its follow-up joint image, which is a relative width of joint space. Although it can be used for RA monitoring with high sensitivity, it cannot provide absolute width information of joint space. It needs to cooperate with other algorithms for qualitative diagnosis in RA. (ii) Take a MCP joint as an example, this framework should segment the proximal phalanx bone

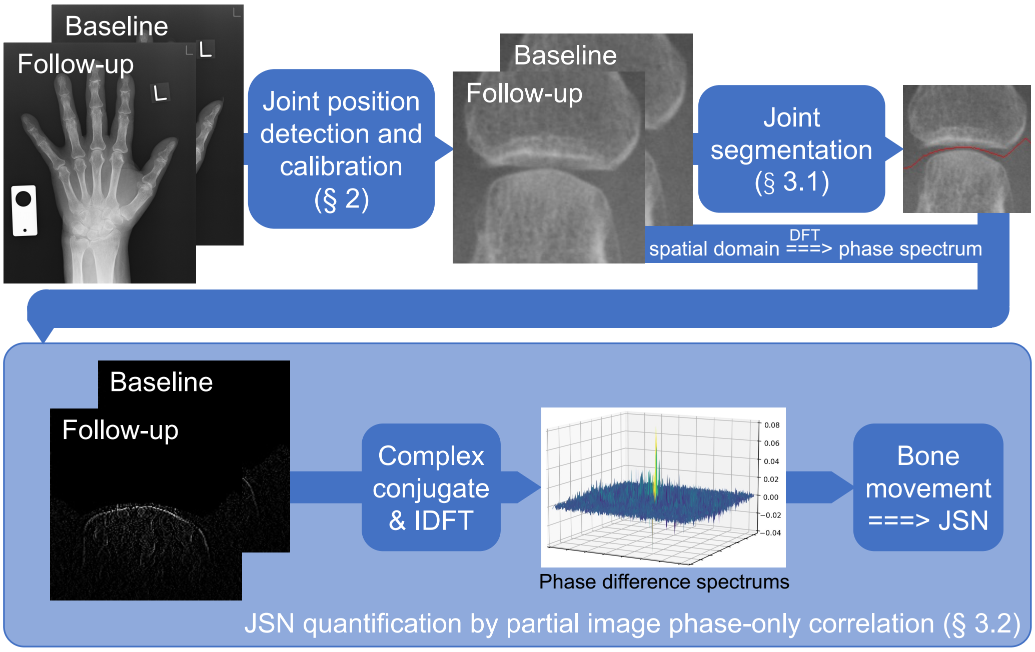

Baseline
Follow-up
Joint position detection and calibration (§ 2)
Baseline
Follow-up
Joint segmentation (§ 3.1)
spatial domain $\overset{\text{DFT}}{===>}$ phase spectrum
Baseline
Follow-up
Complex conjugate & IDFT
0.08
0.06
0.04
0.02
0.00
−0.02
−0.04
Phase difference spectrums
Bone movement ===> JSN
JSN quantification by partial image phase-only correlation (§ 3.2)

Figure 1.6: The overview of our proposed image registration based JSN progression quantification framework. Our framework can be understood in three steps. § 2 Use image processing algorithms to detect and calibrate joint positions. § 3 Segment the upper bone and lower bone based on gradient information. Measure the movement of the upper bone and lower bone between baseline and follow-up radiographs respectively by calculating the phase difference, thus resulting in JSN progression quantification.

and metacarpal bone. Segmentation requires a certain joint space, and this framework cannot be used in advanced RA when the joint space completely disappears.

1.3 Our contributions

In this work, we propose a JSN progression quantification method by calculating the relative widths of the joint space based on an image registration method and studying HW TM algorithms.

The main objective of the proposed JSN progression quantification algorithm is to improve sensitivity, accuracy, and robustness so that radiologist/rheumatologist can closely monitor the JSN progression in early RA. The schematic overview of this work is shown in Fig. 1.6.

The original contribution of this work can be summarized as follows:

1. Describe a detection method for finger midline and joint position (§ 2).
2. Propose an image segmentation algorithm to segment joint images (§ 3.1).
3. Present an improved Phase Only Correlation (POC) method named PIPOC to measure JSN progression in early RA (§ 3.2).
4. Automate the features listed in (1-3). Using our method, JSN progression can be measured from a group of input sequential radiographs. The proposed work can achieve sub-pixel accuracy on JSN progression measurement (§ 5.2).

1.4 Thesis outline

The thesis is organized as follows: § 1 introduces the application of medical image processing algorithms in RA and TM in pre-processing of medical image processing. § 2 Use image processing algorithms to detect and calibrate joint positions. And proposes a joint position detection method based on ML, its calibration method based on phase domain analysis, and a JSN progression quantification framework based on image registration. § 3 Combining the DFT and the joint segmentation curve, segment the proximal phalanx bone and metacarpal bone in the phase spectrum. Then, Measure the movement of the proximal phalanx bone and metacarpal bone between baseline and follow-up radiographs respectively by calculating the phase difference, thus resulting in JSN progression quantification. In § 4, we provide information on the datasets used in this work; including phantom study, and clinical datasets. § 5, presents the joint position detection results using clinical data and the JSN progression quantification evaluation results for both the phantom study and clinical study. § 6 presents a detailed discussion with concluding remarks. Finally, considering that the current popular research direction in ML-based joint space quantification of RA is SvdH scoring method based on image classification, we present an account of the future research directions for the computer-aided monitoring in RA, especially the JSN progression quantification based on ML-based image registration.

In the appendix, § A covers HDR imaging and HW TM algorithms. We present figures of merit for TM implementations, thus throwing light on design trade-offs, and discuss the design issues related to the implementation of TM HW architectures. Additionally, we review various image quality metrics used in HW TM algorithms.

Bibliography

[1] Kanji Shichikawa, Koji Inoue, Shigenaga Hirota, Akira Maeda, Hiroshi Ota, Masami Kimura, Toshio Ushiyama, and Masaki Tsujimoto. Changes in the incidence and prevalence of rheumatoid arthritis in kamitonda, wakayama, japan, 1965–1996. *Annals of the rheumatic diseases*, 58(12):751–756, 1999.

[2] Yannis Alamanos and Alexandros A Drosos. Epidemiology of adult rheumatoid arthritis. *Autoimmunity reviews*, 4(3):130–136, 2005.

[3] Tetsuji Sawada, Shigeko Inokuma, Takeo Sato, Takeshi Otsuka, Yukihiko Saeki, Tsutomu Takeuchi, Takemasa Matsuda, Tamiko Takemura, and Akira Sagawa. Leflunomide-induced interstitial lung disease: prevalence and risk factors in japanese patients with rheumatoid arthritis. *Rheumatology*, 48(9):1069–1072, 2009.

[4] YH Huo. *Automated measurement of joint space width in early rheumatoid arthritis hand radiographs.* PhD thesis, Utrecht University, 2017.

[5] TOVE Riise, BJARNE K Jacobsen, and JT Gran. Incidence and prevalence of rheumatoid arthritis in the county of troms, northern norway. *The Journal of rheumatology*, 27(6):1386–1389, 2000.

[6] Sherine E Gabriel, Cynthia S Crowson, and Michael O'Fallon. The epidemiology of rheumatoid arthritis in rochester, minnesota, 1955–1985. *Arthritis & Rheumatism: Official Journal of the American College of Rheumatology*, 42(3):415–420, 1999.

[7] Kimmo Aho, Oili Kaipiainen-Seppänen, Markku Heliövaara, and Timo Klaukka. Epidemiology of rheumatoid arthritis in finland. In *Seminars in arthritis and rheumatism*, volume 27, pages 325–334. Elsevier, 1998.

[8] Eiichi Tanaka, Eisuke Inoue, Ajitha Mannalithara, Mihoko Bennett, Shigeo Kamitsuji, Atsuo Taniguchi, Shigeki Momohara, Masako Hara, Gurkirpal Singh, and Hisashi Yamanaka. Medical care costs of patients with rheumatoid arthritis during the prebiologics period in japan: a large prospective observational cohort study. *Modern rheumatology*, 20(1):46–53, 2010.

[9] John J Cush, Arthur Kavanaugh, and Michael E Weinblatt. *Rheumatoid arthritis: early diagnosis and treatment.* Professional Communications, 2010.

[10] A Young, J Dixey, N Cox, P Davies, J Devlin, P Emery, S Gallivan, A Gough, D James, P Prouse, et al. How does functional disability in early rheumatoid arthritis (ra) affect patients and their lives? results of 5 years of follow-up in 732 patients from the early ra study (eras). *Rheumatology*, 39(6):603–611, 2000.

[11] Kenneth G Saag, Gim Gee Teng, Nivedita M Patkar, Jeremy Anuntiyo, Catherine Finney, Jeffrey R Curtis, Harold E Paulus, Amy Mudano, Maria Pisu, Mary Elkins-Melton, et al. American college of rheumatology 2008 recommendations for the use of nonbiologic and biologic disease-modifying antirheumatic drugs in rheumatoid arthritis. *Arthritis Care & Research: Official Journal of the American College of Rheumatology*, 59(6):762–784, 2008.

[12] Vikas Majithia and Stephen A Geraci. Rheumatoid arthritis: diagnosis and management. *The American journal of medicine*, 120(11):936–939, 2007.

[13] DMFM Van der Heijde. How to read radiographs according to the sharp/van der heijde method. *The Journal of rheumatology*, 27(1):261, 2000.

[14] J Duryea, Y Jiang, M Zakharevich, and HK Genant. Neural network based algorithm to quantify joint space width in joints of the hand for arthritis assessment. *Medical physics*, 27(5):1185–1194, 2000.

[15] Jane Angwin, Geoff Heald, Andrew Lloyd, Kate Howland, MARIA Davy, and Michael F James. Reliability and sensitivity of joint space measurements in hand radiographs using computerized image analysis. *The Journal of rheumatology*, 28(8):1825–1836, 2001.

[16] Philipp Peloschek, Georg Langs, Michael Weber, Johannes Sailer, Michael Reisegger, Herwig Imhof, Horst Bischof, and Franz Kainberger. An automatic model-based system for joint space measurements on hand radiographs: initial experience. *Radiology*, 245(3):855–862, 2007.

[17] Georg Langs, Philipp Peloschek, Horst Bischof, and Franz Kainberger. Automatic quantification of joint space narrowing and erosions in rheumatoid arthritis. *IEEE transactions on medical imaging*, 28(1):151–164, 2008.

[18] R Van't Klooster, EA Hendriks, I Watt, M Kloppenburg, JHC Reiber, and BC Stoel. Automatic quantification of osteoarthritis in hand ra-

diographs: validation of a new method to measure joint space width. *Osteoarthritis and Cartilage*, 16(1):18–25, 2008.

[19] Andrzej Bielecki, Mariusz Korkosz, and Bartosz Zieliński. Hand radiographs preprocessing, image representation in the finger regions and joint space width measurements for image interpretation. *Pattern Recognition*, 41(12):3786–3798, 2008.

[20] Bartosz Zieliński. Hand radiograph analysis and joint space location improvement for image interpretation. *Schedae Informaticae*, 17, 2009.

[21] Yinghe Huo, Koen L Vincken, Désirée van der Heijde, Maria JH De Hair, Floris P Lafeber, and Max A Viergever. Automatic quantification of radiographic finger joint space width of patients with early rheumatoid arthritis. *IEEE Transactions on Biomedical Engineering*, 63(10):2177–2186, 2015.

[22] Kento Morita, Patrick Chan, Manabu Nii, Natsuko Nakagawa, and Syoji Kobashi. Finger joint detection method for the automatic estimation of rheumatoid arthritis progression using machine learning. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1315–1320. IEEE, 2018.

[23] Kazuki Kato, Nobutoshi Yasojima, Kenichi Tamura, Shota Ichikawa, Kenneth Sutherland, Masaru Kato, Jun Fukae, Kazuhide Tanimura, Yuki Tanaka, Taichi Okino, et al. Detection of fine radiographic progression in finger joint space narrowing beyond human eyes: Phantom experiment and clinical study with rheumatoid arthritis patients. *Scientific reports*, 9(1):1–10, 2019.

[24] Toru Hirano, Masayuki Nishide, Naoki Nonaka, Jun Seita, Kosuke Ebina, Kazuhiro Sakurada, and Atsushi Kumanogoh. Development and validation of a deep-learning model for scoring of radiographic finger joint destruction in rheumatoid arthritis. *Rheumatology advances in practice*, 3(2):rkz047, 2019.

[25] Yafei Ou, Prasoon Ambalathankandy, Takeshi Shimada, Tamotsu Kamishima, and Masayuki Ikebe. Automatic radiographic quantification of joint space narrowing progression in rheumatoid arthritis using poc. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 1183–1187. IEEE, 2019.

[26] Kohei Nakatsu, Kento Morita, Naomi Yagi, and Syoji Kobashi. Finger joint detection method in hand x-ray radiograph images using statistical

shape model and support vector machine. In *2020 International Symposium on Community-centric Systems (CcS)*, pages 1–5. IEEE, 2020.

[27] Kemal Üreten, Hasan Erbay, and Hadi Hakan Maraş. Detection of rheumatoid arthritis from hand radiographs using a convolutional neural network. *Clinical rheumatology*, 39(4):969–974, 2020.

[28] Krzysztof Maziarz, Anna Krason, and Zbigniew Wojna. Deep learning for rheumatoid arthritis: Joint detection and damage scoring in x-rays. *arXiv preprint arXiv:2104.13915*, 2021.

[29] Aimi Taguchi, Shun Shishido, Yafei Ou, Masayuki Ikebe, Tianyu Zeng, Wanxuan Fang, Koichi Murakami, Toshikazu Ueda, Nobutoshi Yasojima, Keitaro Sato, et al. Quantification of joint space width difference on radiography via phase-only correlation (poc) analysis: a phantom study comparing with various tomographical modalities using conventional margin-contouring. *Journal of Digital Imaging*, 34(1):96–104, 2021.

[30] Miles N Wernick, Yongyi Yang, Jovan G Brankov, Grigori Yourganov, and Stephen C Strother. Machine learning in medical imaging. *IEEE signal processing magazine*, 27(4):25–38, 2010.

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[32] Norah Asiri, Muhammad Hussain, Fadwa Al Adel, and Nazih Alzaidi. Deep learning based computer-aided diagnosis systems for diabetic retinopathy: A survey. *Artificial intelligence in medicine*, 99:101701, 2019.

[33] Grant Haskins, Uwe Kruger, and Pingkun Yan. Deep learning in medical image registration: a survey. *Machine Vision and Applications*, 31(1):1–18, 2020.

[34] Sungmin Lee, Minsuk Choi, Hyun-soo Choi, Moon Seok Park, and Sungroh Yoon. Fingernet: Deep learning-based robust finger joint detection from radiographs. In *2015 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4. IEEE, 2015.

[35] Sytske Anne Bergstra, Joy A Van Der Pol, Naghmeh Riyazi, Yvonne PM Goekoop-Ruiterman, Pit JSM Kerstens, Willem Lems, Tom WJ

Huizinga, and Cornelia F Allaart. Earlier is better when treating rheumatoid arthritis: but can we detect a window of opportunity? *RMD open*, 6(1):e001242, 2020.

[36] Tore K Kvien, Till Uhlig, SIGRID ØDEGÅRD, and Marte S Heiberg. Epidemiological aspects of rheumatoid arthritis: the sex ratio. *Annals of the New York Academy of Sciences*, 1069(1):212–222, 2006.

[37] Tomohiro Shimizu, Ana Cruz, Matthew Tanaka, Kenji Mamoto, Valentina Pedoia, Andrew J Burghardt, Ursula Heilmeier, Thomas M Link, Jonathan Graf, John B Imboden, et al. Structural changes over a short period are associated with functional assessments in rheumatoid arthritis. *The Journal of rheumatology*, 46(7):676–684, 2019.

[38] Mengdi Jiang, Yueting Li, Chendan Jiang, Lidan Zhao, Xuan Zhang, and Peter E Lipsky. Machine learning in rheumatic diseases. *Clinical Reviews in Allergy & Immunology*, 60(1):96–110, 2021.

[39] Kathryn M Kingsmore, Christopher E Puglisi, Amrie C Grammer, and Peter E Lipsky. An introduction to machine learning and analysis of its use in rheumatic diseases. *Nature Reviews Rheumatology*, 17(12):710–730, 2021.

Chapter 2

Joint position detection and calibration

For joint location detection, we propose and compare two methods, a classification based joint detection method and a key point detection based joint location detection method.

2.1 Classification based joint detection

As shown in Fig. 2.1, the pipeline of joint position detection and calibration can be briefly explained as follows: (i) Obtain the approximate estimates of the finger midlines in binary image. (ii) Detect joint positions by using a ML-based joint classifier. (iii) Calibrate the relative position deviations in joint windows.

2.1.1 Finger midline detection

Finger position estimation can significantly reduce the potential region, thus reduces the calculation of joint detection. The scheme of finger midline detection is shown in Fig. 2.1. The approximate area and angle of fingers are estimated using the binary images obtained from respective hand radiographs.

Given a radiograph, we binarize the X-ray using Otsu's method [1], and smooth its margin by using morphological opening and closing [2], as shown in Fig 2.2 (b). From our experiments we found that using polygonal approximation can significantly improve robustness when searching for extrema, that one could obtain using pure margin [3] (Fig. 2.2 (c)).

Then, we obtain the local maxima (red point) and local minima (green

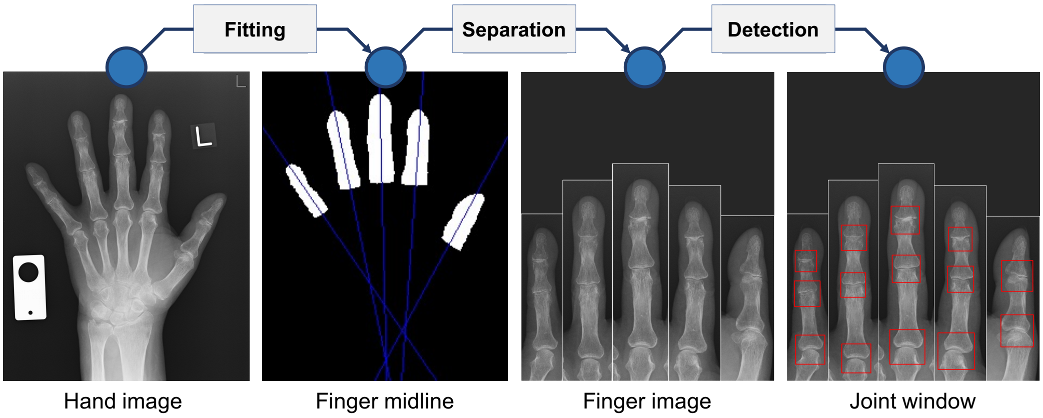


Figure 2.1: Schematic overview of joint position detection. The approximate regions of fingers are obtained according to each pair of local maxima and local minima in binary image. Then, the finger midlines (blue lines) are calculated by fitting to each region. Finally, an AdaBoost based joint classifier is used to detect the joint positions (red windows).

point) of hand margins as shown in Fig. 2.3 (a). Next, the approximate region of fingers are obtained according to each pair of local maxima and minima, which can calculate the midline of each finger by fitting to each finger region based on least squares method [4] (Fig. 2.3 (b)). Finally, the width of each finger can be calculated according to the size of the finger region. Thus, the location, angle and width of other fingers can obtained by using the same method. The midline of each finger as show in Fig 2.3 (c).

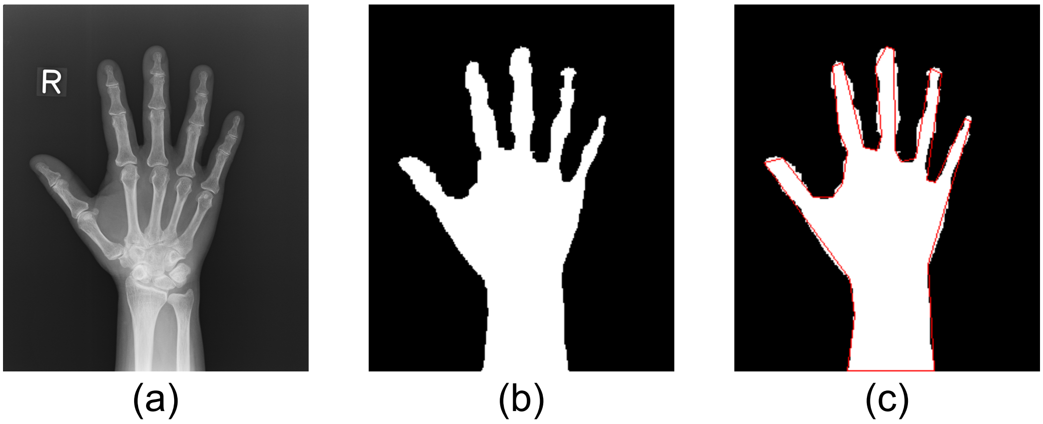


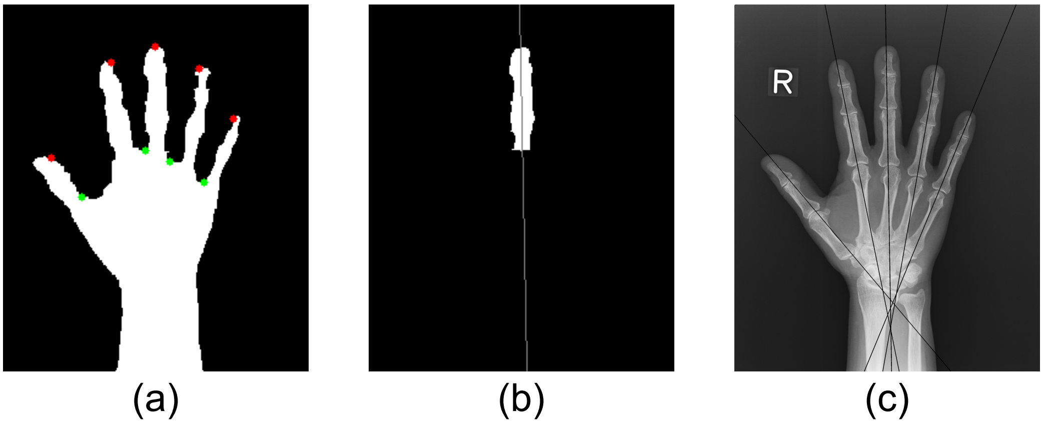
Figure 2.2: (a) A hand radiographic image. (b) The binary image of hand region. (c) The polygonal approximate edge of hand region.

Then, from a hand radiographic image (Fig 2.4 (a)), the fingers area can be cut out (Fig 2.4 (b)) to detect the position of joints. From our experiments and analysis we found that reducing the width and height of the binary hand images to one-fifth does not significantly effect the accuracy of the finger midline detection, and this results in accelerating the detection process (17.7 × faster).

2.1.2 Joint position detection

As shown in Fig. 2.1, the position, angle, and size of each finger image is obtained according to the finger midline and its region in the binary image. Then, the joint windows are detected in finger images with a joint classifier which is trained by using haar-like feature based AdaBoost (Adaptive Boosting) [5–7].

AdaBoost is a cascade classifier algorithm. Through learning a large number of positive and negative samples, a large number of weak classifiers with less than 50% false detection rates are generated. A weighted superposition of these weak classifiers is used to obtain a strong classifier with a low false detection rate. The theory proves that the false detection rate of the strong classifiers will approach zero, when the number of weak classifiers approaches infinity and the false detection rate of each weak classifier is less than 50%. Papageorgiou *et al.* in [8] proposed a face detection algorithm based on AdaBoost algorithm. A group of weak classifiers was trained based on Haar-like features and the integral graph is used to accelerate the training process [5]. The cascade structure can improve the detection accuracy of the



(a) (b) (c)

Figure 2.3: (a) Grooves and fingertips of hand. (b) The area and midline of finger. (c) Finger midlines in the radiographic hand image.

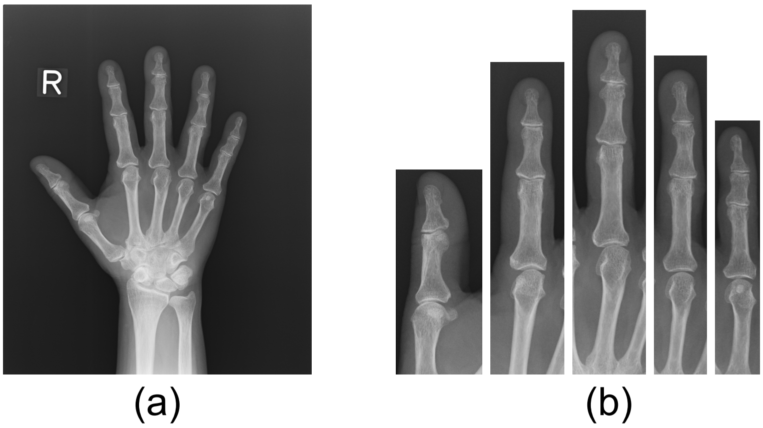


Figure 2.4: (a) A hand radiographic image. (b) Finger images from hand radiographic image.

classifier.

Haar-like feature also called rectangular feature, which reflects the gray change of the image and local gray distribution. The application of Haar-like features to face detection was proposed by Papageorgiou *et al.* in [8]. Paul Viola *et al.* in [5] proposed quickly calculating Haar-like features by using integral graphs. Haar-like features can divide into three categories: Edge, line, and center-surround features [8], as show in Fig, 2.5.

The haar-like feature is mainly responsible for training weak classifiers. In

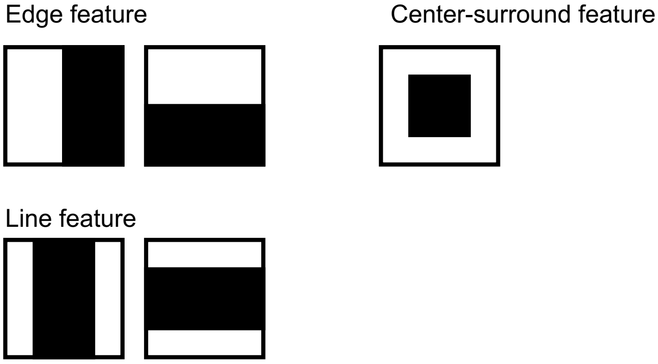


Figure 2.5: Haar-like features

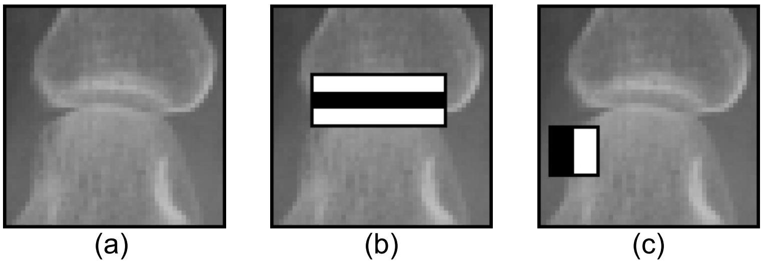


Figure 2.6: Haar-like features in a joint image.

general, the Haar-like feature is composed of 2 to 4 rectangles (Fig 2.5). These Haar-like features represent some location features like edge (edge feature), line (line feature), or point (center-surround feature). And the difference between the gray values of all pixels in the range of these rectangles is the Haar-like feature value. The Haar-like feature value of a kind of feature represents the strength of the feature at the point.

As shown in Fig. 2.6 (a), a joint positive sample is downsampled. The Haar-like feature of Fig. 2.6 (b) is composed of three vertically arranged matrices, and the Haar-like feature of Fig. 2.6 (c) is composed of two horizontally arranged matrices. The Haar-like feature template can be located at any position in the image. As shown in Fig 2.6, the two Haar-like features value are the difference of the grayscale between the white matrix region and black matrix region. The Haar-like feature can show the grayscale changes of the image or local. For example, in Fig 2.6, the joint space area is darker than the upper and lower bones, and the line feature value will be bigger than in other places. And the bone area is brighter than the background so that the edge feature of points on the bone edge will be bigger than others. Through the value of the line feature, the probability that the point is a joint space can be determined. Same to the edge feature, the probability of bone edge also can be determined. Only two regions cannot reflect all the features of the joint. However a large number of weak classifiers trained by Haar-like features can classify joint windows with high accuracy.

Integral image

is proposed by Paul Viola *et al.* to reduce the repeated steps in the calculation of the sum of gray values in [5]. The integral image is an image representation method. It represents the sum of the gray values of all pixels in the upper left corner of each pixel of the original image, as shown in Fig 2.7. The integral image can convert the integral calculation of each gray value into the addition and subtraction calculation in the integral image, which can greatly reduce the time of training. The integral image $SAT(x, y)$ obtained from the original gray image $f(x, y)$ is defined as follows:

$$SAT(x, y) = \sum_{x'=0,y'=0}^{x'<x,y'<y} f(x', y') \tag{2.1}$$

The following formula is used for quick calculation of the integral image.

$$s(x, y) = s(x, y - 1) + f(x, y) \tag{2.2}$$

$$SAT(x, y) = SAT(x - 1, y) + s(x, y) \tag{2.3}$$

$s(x, y)$ represents the column integral, $f(x, y)$ is the gray value of the original gray image, and $SAT(x, y)$ represents the integral image.

Calculate Haar-like feature by integral image

The gray value integration of the pixels contained in any matrix can be completed in several addition and subtraction calculations. Taking Fig. 2.8 (a) as an example, calculate the gray value integral $RecSum(r)$ in the black area r as:

$$\begin{aligned} RecSum(r) &= \sum_{x'=A(x),y'=A(y)}^{x'<D(x),y'<D(y)} f(x', y') \\ &= SAT(A) - SAT(B) - SAT(C) + SAT(D) \end{aligned} \tag{2.4}$$

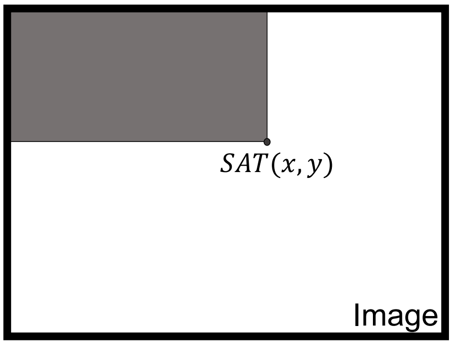


Figure 2.7: Integral image

Haar-like features consist of 2 to 4 rectangles, and this method can also be used to obtain the Haar-like feature value within a few steps of addition and subtraction. Taking Fig. 2.8(b) as an example, the feature value F of this edge feature is calculated as:

$$\begin{aligned} F &= RecSum(r_1) - RecSum(r_2) \\ &= SAT(A) - SAT(B) - 2SAT(C) + 2SAT(D) + SAT(E) - SAT(F) \end{aligned} \tag{2.5}$$

Weak classifiers

Each Haar-like feature corresponds to a weak classifier. Each weak classifier h_j is composed of feature value f_j, threshold value θ_j, and sign p_j (indicating the direction of the inequality, which can only take ±1). The formula is:

$$h_j = \begin{cases} 1 & p_j f_j(x) \leq p_j \theta_j \\ 0 & otherwise \end{cases} \tag{2.6}$$

Each weak classifier depends on the feature value f_j and the threshold θ_j. The training of the weak classifier is to determine the best value of the threshold θ_j under the current weight so that the classifier has the smallest classification error value.

For each Haar-like feature, the feature values are calculated for all positive and negative training samples, and the classification error corresponding to each feature value is calculated. Then, the classification with the smallest

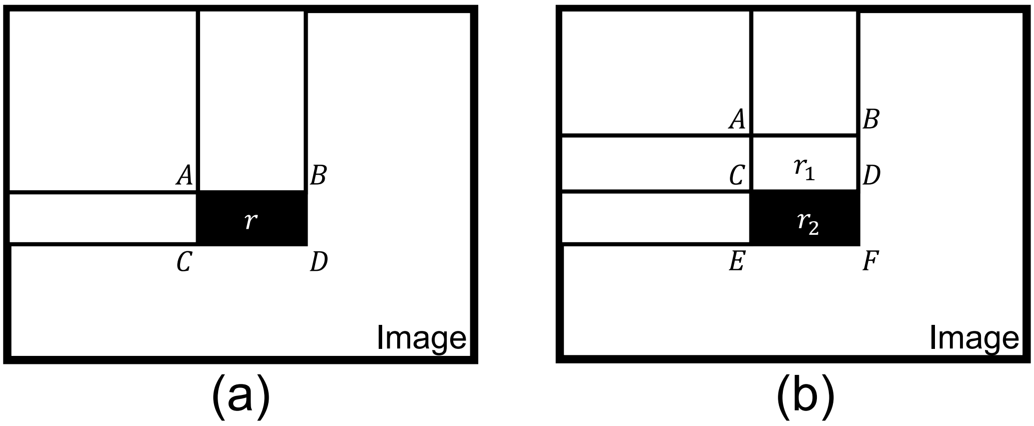


Figure 2.8: Calculate Haar-like feature by integral image

error is selected and its corresponding threshold is the optimal threshold of the feature. Thus, a weak classifier can be trained. The formula is as follows.

The classification error value e_j^i at a certain threshold θ_j^i and sign p_j^i is:

$$e_j^i = min((S^+ + (T^- - S^-)), (S^- + (T^+ - S^+))) \tag{2.7}$$

S^+ represents the weighted integral of all samples judged to be positive by this weak classifiers, and S^- represents the weighted integral of all samples judged to be negative by this weak classifiers. T^+ represents the weighted integral of all positive samples, and T^- represents the weighted integral of all negative samples.

When the optimal threshold, the classification error value e_j has a minimum value:

$$e_j = min(e_j^i) \tag{2.8}$$

At this time, the corresponding threshold value θ_j^i and sign p_j^i are obtained for the weak classifier.

Complex classifiers

Considering there are a large number of Haar-like features and a large number of weak classifiers derived from Haar-like features. To balance detection accuracy and detection efficiency, weak classifiers with high classification accuracy are selected to generate complex classifiers with higher classification accuracy.

Among all the classifiers, several weak classifiers h_1 $h_2 ... h_n$ with the highest classification accuracy are selected, and complex classifier H_k obtained according to the following formula:

$$H_k = \begin{cases} 1 & \sum_{j=1}^{N} a_j h_j \geq \frac{1}{2} \sum_{j=1}^{N} a_j \\ 0 & otherwise \end{cases} \tag{2.9}$$

a_j is the weight of the weak classifier h_j determined by the classification accuracy. When a complex classifier performs a classification operation on an image, it is equivalent to all the weak classifiers of this complex classifier performing weighted voting according to their error rate.

As the number of weak classifiers increases, the error rate of the strong classifiers decreases rapidly. For example, a complex classifier consisting of 400 weak classifiers with an error rate of 40% has an error rate of less than 1%. Considering a large number of weak classifiers, complex classifiers can have a high detection accuracy.

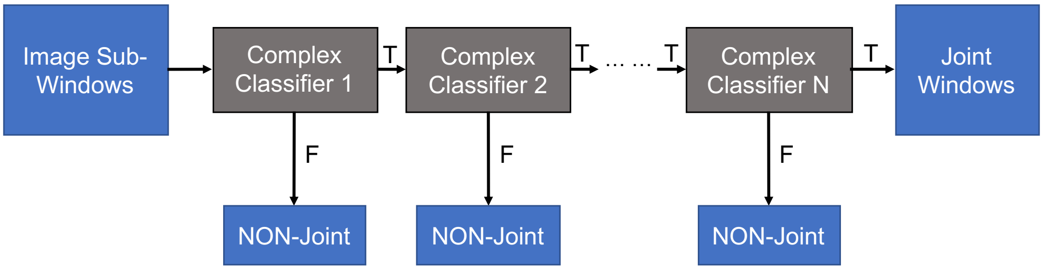


Figure 2.9: Cascade structure

Cascade structure

Although the complex classifier has high detection accuracy. However, considering that there are a large number of sub-windows in each image, it will have a large amount of computation and cost a lot of time, if the complex classifier with the strongest classification ability is used to detect each sub-window. Therefore, the cascade structure is introduced to improve the operational efficiency of the detection.

A cascade structure is a combination of a series of complex classifiers. As shown in Fig. 2.9, the cascade structure is a structure that connects several complex classifiers in series. First, the non-joint sub-windows are removed by the complex classifier with a fast detection speed, then uses a stronger complex classifier to classify the sub-windows that the previous classifier cannot distinguish. During the detection process, only the joint sub-window and a very small number of negative sub-windows through all the complex classifiers in the cascade structure, and most of the non-joint sub-windows will be gradually eliminated during the detection process. Because a large number of non-joint sub-windows are removed from the first few strong classifiers of the cascade structure, only a small number of sub-windows that are difficult to distinguish enter the subsequent strong classifiers. Therefore, this type of classifier structure can greatly reduce the calculation and improve the detection speed.

Joint classifier

Figure 2.10(a) is the detection result by the classifier trained by AdaBoost. Combining finger location and angle detection algorithm and the joint location detection algorithm, the location of joints in hand radiographic images can be detected. The location of each joint in the hand radiographic image is shown in Fig. 2.10(b).

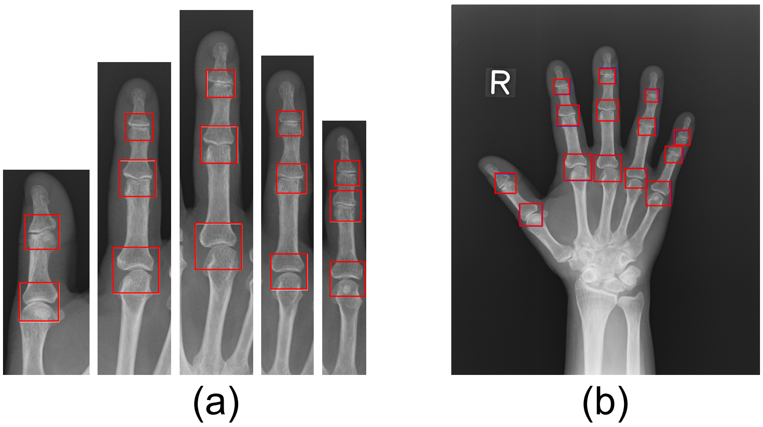


Figure 2.10: (a) Joint location. (b) Joint location in hand radiographic image.

2.1.3 Joint position calibration

Actually, the location of joint windows obtained by using a classifier usually has pixel-level deviation. This subsection aims to align two joint images. As shown in Fig. 2.11, (a) is two PIP joint images of the little finger. (b) is the margin of the PIP radiograph in (a) before position calibration (red: baseline radiograph, cyan: its follow-up radiograph). From sub-images (a) and (b),

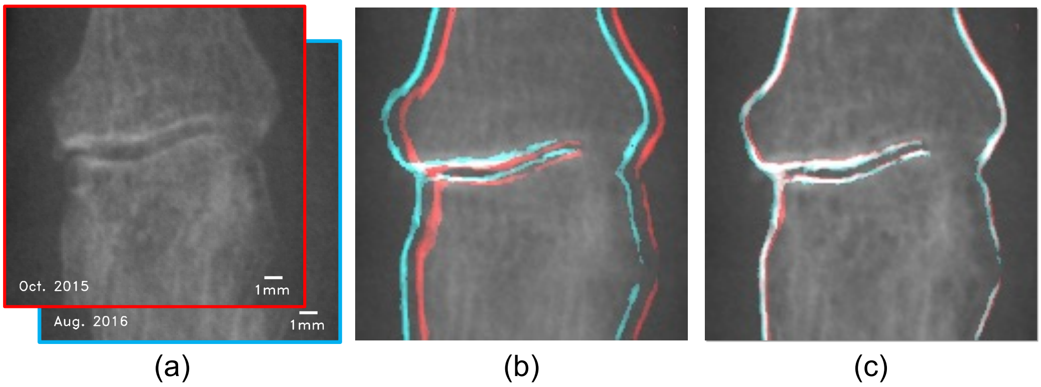


Figure 2.11: Results of joint position calibration: (a) A PIP joint of the little finger in baseline radiograph (red border) and its follow-up radiograph (cyan border). (b) The margin of the PIP radiograph in (a) before position calibration (red: baseline radiograph, cyan: its follow-up radiograph, white: overlap). (c) The margin information after position calibration.

we can easily see the location difference between the two joint windows. We propose a low computational solution based on FIPOC to calibrate the joint position, a detailed discussion of FIPOC implementation is presented in § 3.2.

Usually, the bone surface texture is frequently changing and varies when the bone is growing, and these irregular variation will cause phase dispersion in phase difference spectrum when using POC method. Our experiments show that margin preserving filter can significantly reduce mismatch error of POC, which can suppress bone surface texture information while preserving the bone margin. In our work, we preprocess the image using a median filter to calibrate position deviation [9].

As show in Fig. 2.11 (c) the joint position calibration which relies on FIPOC cannot reduce the deviation with ground truth. It can limit relative position deviation between baseline and follow-up joint windows within one pixel.

2.2 Key point detection based joint location detection

As shown in Fig. 2.12, the proposed joint location detection architecture consists of three consecutive CNNs. These networks sequentially implement the following functions: normalizing the input hand radiography, detecting rough joint positions and correcting joint position. The important properties of the CNNs are given in Table 2.1.

Because the left hand and right hand radiographic images are mirror-symmetrical, left hand radiographic images are mirrored horizontally for standardization. An AlexNet-like network is implemented to classify radiographic images [10]. To reduce training and detection time, we cropped out the part without joint information in the lower half of the input single hand radiographic image from 1430×1722 pixels to 1430×1430 pixels, and down-scaled from 1430×1430 pixels to 100×100 pixels.

In the joint position detection network and the joint position correction network, the CoordConv layer is used to improve the accuracy of key point detection [11]. The joint position detection network can detect the rough position of 14 joints from a right hand radiographic image (28 coordinate data in total). Then, a square area centered on the the rough position of each joint was cropped from original radiographic image as the input of the joint position correction network. Considering that the size of different joints are different, we used different size of cropping windows for different joints based on our experience. The size of Distal Interphalangeal (DIP)

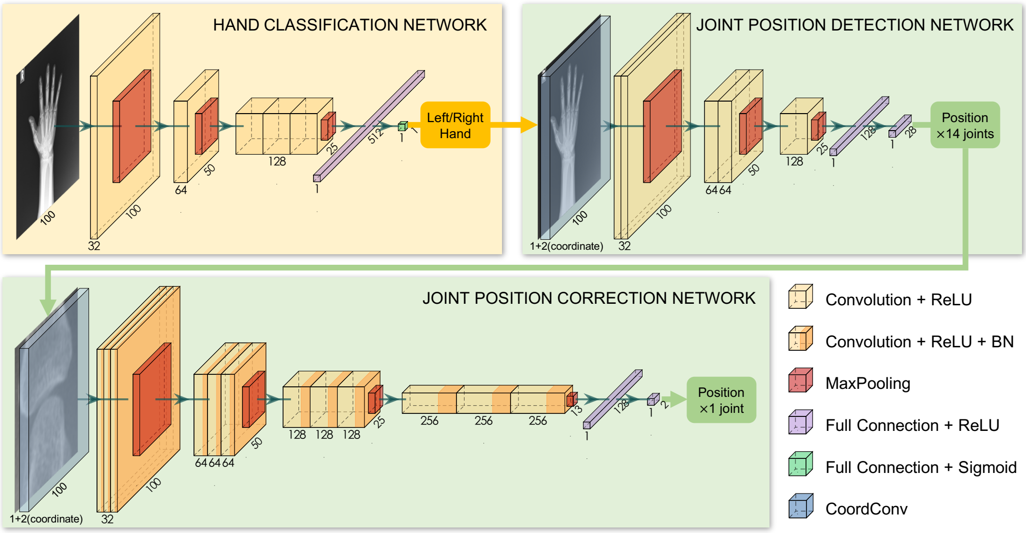


Figure 2.12: The joint location detection method consists of three consecutive CNNs. The first network is used to classify left or right hand, and mirror left hand radiography horizontally for standardization. The second network can detect the rough position of 14 joints, and the CoordConv layer is introduced to improve the accuracy of key point detection. Finally, a deeper network is used to correct the position of each joint.

Table 2.1: The CNN design properties of joint location detection method.

CNN	**Hand classification**	**Joint position detection**	**Joint position correction**
Input layer size	100×100×1	100×100×3	100×100×3
Output layer size	1	28	2
Number of layers	7	7	14
Activation	ReLU	ReLU	ReLU
Optimizer	Adam	Adam	Adam
Loss	binary cross entropy	MSE*	MSE*
Number of epochs	10	100	100

* MSE: mean squared error.

joint, PIP joint, Interphalangeal (IP) joint and MCP joint are 60×60 pixels, 80×80 pixels, 80×80 pixels and 100×100 pixels respectively. And cropped images were scaled to 100×100 pixels for standardization (spatial resolution: 0.175mm/pixel).

Compared to the joint position detection network, the joint position correction network has a deeper structure to achieve higher accuracy. The loss of the two networks were optimized with the ground truth separately.

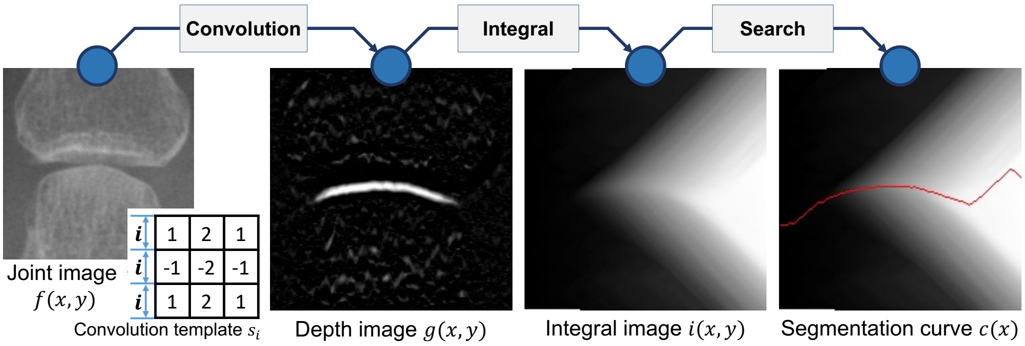
Chapter 3

Joint space quantification

In our joint space quantification, we segment the upper and lower bones by gully detection and then used phase only correlation-like algorithm to calculate the movement of upper and lower bones, respectively. Then, the JSN progression can be calculated according to the bone movements.

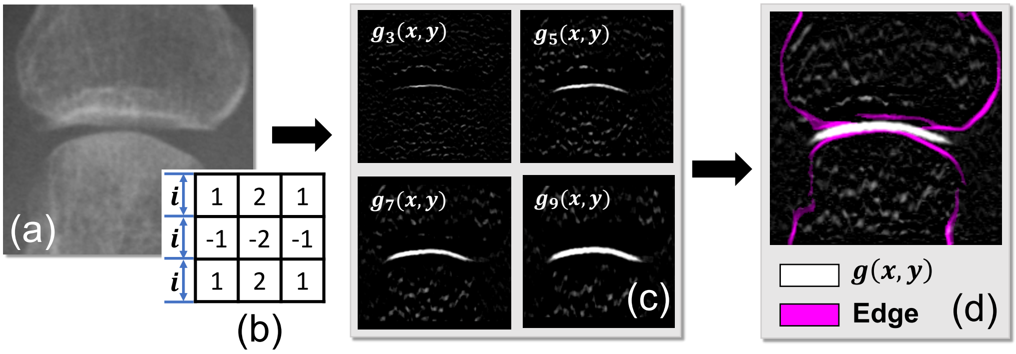
3.1 Joint segmentation

Take a MCP joint as an example, the proximal phalanx bone and metacarpal bone are segmented from the joint image, based on gradient information, so that the displacements of the proximal phalanx bone and metacarpal bone can be measured separately.



i	1	2	1
i	-1	-2	-1
i	1	2	1

Figure 3.1: Overview of joint segmentation based on gradient information. Gully depth map $g(x, y)$ is calculated to define the depth feather. Independent margin filtering on the upper and lower side determines the pixel depth. Height-adjustable convolution template S_i ensures that a given range of gully can be detected. The integral image $i(x, y)$ is calculated to search the segmentation curve $c(x)$ with the maximum depth-sum.



i	1	2	1
i	-1	-2	-1
i	1	2	1

Figure 3.2: (a) Joint image $f(x,y)$. (b) Convolution template S_i. (c) Gully depth map with any width $g_i(x,y)$. (d) Gully depth map $g(x,y)$ and bone margin.

3.1.1 Depth image

The depth map is used to gauge the depth of each pixel within a given range of width. Only the vertical depth is detected in this work, because all joint images are arranged vertically. Nevertheless, the depth of any direction can be detected with a customized convolution template S_i. The detailed explanation is shown in Fig. 3.1.

In order to detect depth within a range of width, a height-adjustable convolution template S_i is used (i is odd) to calculate the depth of i pixels gully height. Consider a joint image $f(x,y)$ with M pixel width and N pixel height. The convolution of $f(x,y)$ can be formulated as shown in Eq. 3.1.

$$\begin{aligned} g_{ia}(x,y) &= \sum_{k=-1}^{1} \sum_{l=-(i-1)/2}^{(3i-1)/2} s_i(k,l) f(x+k, y+l) \\ g_{ib}(x,y) &= \sum_{k=-1}^{1} \sum_{l=-(3i-1)/2}^{(i-1)/2} s_i(k,l) f(x+k, y+l) \end{aligned} \tag{3.1}$$

$g_{ia}(x,y)$ represents the gradient above, and $g_{ib}(x,y)$ represents the gradient below. The smaller gradient is defined as the depth, as shown in Eq. 3.2.

$$g_i(x,y) = \min(g_{ia}(x,y), g_{ib}(x,y)) \tag{3.2}$$

The depth images $g_i(x,y)$ of the joint image $f(x,y)$ are shown in Fig. 3.1 when i is 3, 5, 7 or 9. The i represents the height of the height-adjustable convolution template S_i. In depth map $g_i(x,y)$, narrow gullies can have high intensity when i is small, conversely, wide gullies have high intensity when i is large.

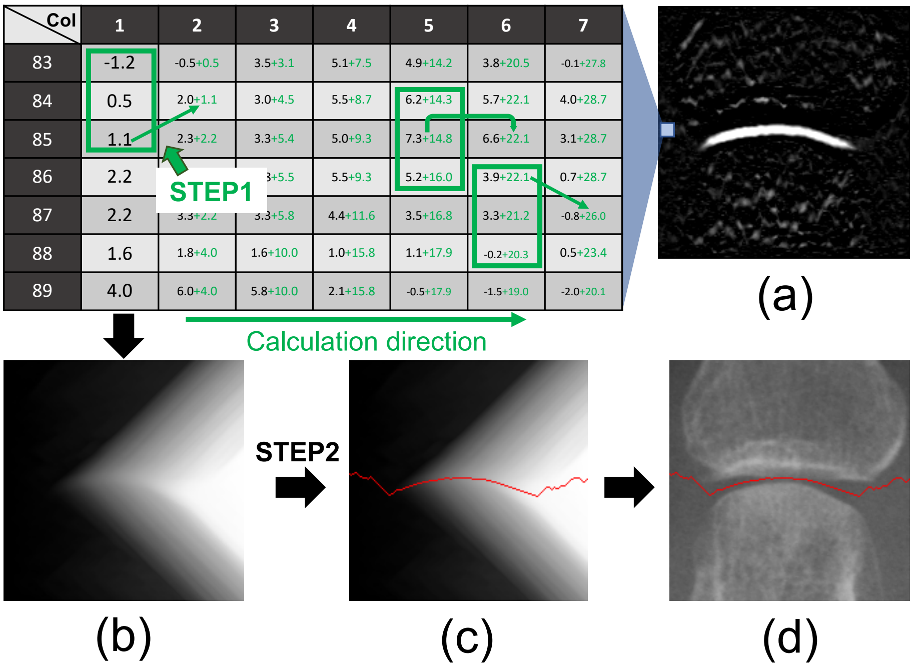
A max-pooling is performed on depth map $g_i(x, y)$ to down-sample, as shown in Eq. 3.3.

$$g(x, y) = \max(g_{i_{min}}(x, y), \cdots, g_{i_{max}}(x, y)) \tag{3.3}$$

The $[i_{min}, i_{max}]$ represents the range of the height of convolution template S_i. This range is positively correlated with the spatial resolution. In our experience this range is defined as $[i_{min} = 1, i_{max} = 9]$ when the spatial resolution is 0.175 mm/pixel.

3.1.2 Integral image

The integral image $i(x, y)$ is an intermediate matrix, which is used to find the segmentation curve with the maximum depth-sum. It can be expressed



Col	1	2	3	4	5	6	7
83	-1.2	-0.5+0.5	3.5+3.1	5.1+7.5	4.9+14.2	3.8+20.5	-0.1+27.8
84	0.5	2.0+1.1	3.0+4.5	5.5+8.7	6.2+14.3	5.7+22.1	4.0+28.7
85	1.1	2.3+2.2	3.3+5.4	5.0+9.3	7.3+14.8	6.6+22.1	3.1+28.7
86	2.2	[illegible]	3+5.5	5.5+9.3	5.2+16.0	3.9+22.1	0.7+28.7
87	2.2	3.3+2.2	3.3+5.8	4.4+11.6	3.5+16.8	3.3+21.2	-0.8+26.0
88	1.6	1.8+4.0	1.6+10.0	1.0+15.8	1.1+17.9	-0.2+20.3	0.5+23.4
89	4.0	6.0+4.0	5.8+10.0	2.1+15.8	-0.5+17.9	-1.5+19.0	-2.0+20.1

Figure 3.3: (a) Gully depth map $g(x, y)$. (b) Weight-sum map $r(x, y)$. (c) Segmentation result. (d) Result in the joint image. 3. Step 1: Add the local maximum in the left column to the current point. Step 2: Following the maximum value from right to left leads to obtaining the segmentation curve.

as the local maximum in the left column plus depth map $g(x, y)$, as shown in Eq. 3.4.

$$i(x,y) = \begin{cases} g(x,y) & x = 0 \\ \max(i(x-1,y-1), i(x-1,y), i(x-1,y+1)) + g(x,y) & x > 0 \end{cases} \tag{3.4}$$

$g(x, y)$ is the gully depth map calculated in the previous subsection, and $i(x, y)$ is the weight-sum matrix.

In our experiment, we use morphological dilation method [2] to speed up computing, as shown in Eq. 3.5 and Eq. 3.6.

$$r_{(x)} = \begin{cases} g_{(x)} & x = 0 \\ r_{(x-1)} \bigoplus D + g_{(x)} & x > 0 \end{cases} \tag{3.5}$$

$$D = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{3.6}$$

$g(x)$ is the column x of the gully depth map, r_x is the column x of the weight-sum matrix. Matrix D is the kernel of morphological dilation method. Dilation is used only in vertical direction as show in Eq. 3.6.

3.1.3 Segmentation curve

The segmentation curve with the maximum depth-sum can be determined from integral image $i(x, y)$ as follows. First, determine the maximum value of

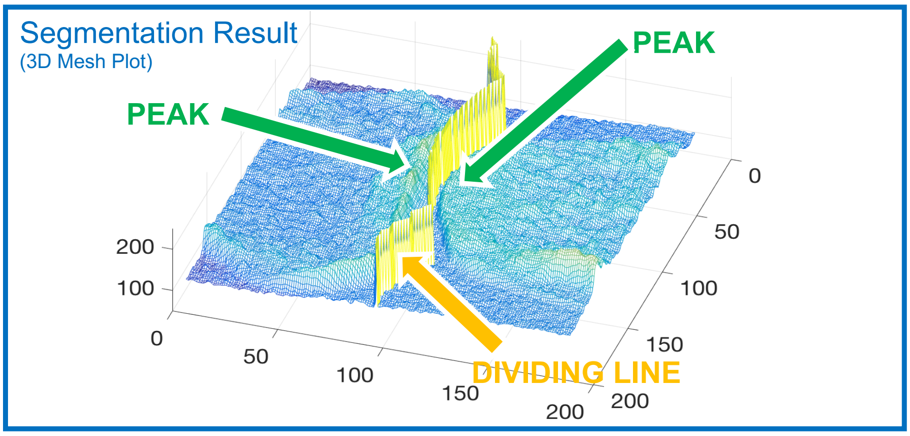


Figure 3.4: The 3D segmentation result of a MCP joint radiographic image.

the rightmost column in $i(x, y)$ as the end point of the segmentation curve. Then, select the maximum of the three adjacent pixels in the left column in $i(x, y)$ as the next point of the segmentation curve until arriving at the leftmost column. The segmentation curve $c(x)$ is defined as Eq. 3.7. The $\arg\max_y i(x, y)$ indicates the index of the maximum value on the y axis for a given x value in a given y range.

$$c(x) = \arg\max_y i(x, y) \begin{cases} y \in [0, N-1] & x = M-1 \\ y \in [c(x+1)-1, c(x+1)+1] & x < M-1 \end{cases} \tag{3.7}$$

The binary matrix of the proximal phalanx bone $s_0(x, y)$ and the metacarpal bone $s_1(x, y)$ can be expressed as Eq. 3.8, according to the segmentation curve $c(x)$.

$$\begin{aligned} s_0(x, y) &= \begin{cases} 1 & y < c(x) \\ 0 & \text{otherwise} \end{cases} \\ s_1(x, y) &= \begin{cases} 1 & y > c(x) \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{3.8}$$

An example of finger joint segmentation is shown in Fig. 3.1. And the segmentation result is shown in Fig. 3.4.

3.2 JSN progression quantification by PIPOC

The basic concept of JSN progression quantification by FIPOC or PIPOC can be described using the flowchart in Fig 3.5. Take a MCP joint as an

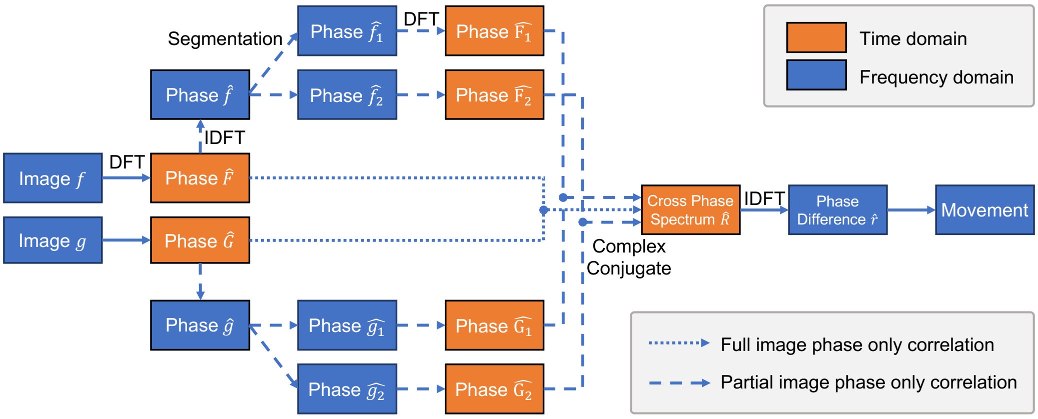


Figure 3.5: A flowchart describing the sequence of operations for implementing FIPOC and PIPOC algorithms.

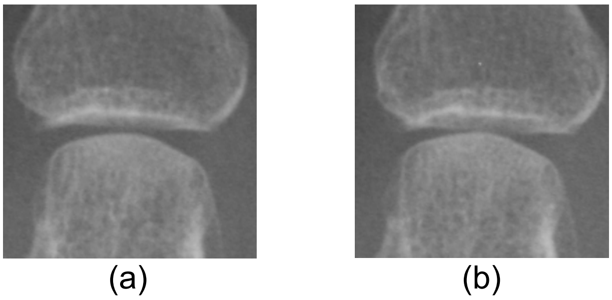


Figure 3.6: (a) The baseline radiographic image. (b) The follow-up radiographic image.

example. Consider two images, the baseline image $f(x, y)$ and its follow-up image $g(x, y)$, which are divided into k regions, as shown in Fig. 3.6.

Consider a region i, let α_i and β_i represent sub-pixel displacement from $f(x, y)$ to $g(x, y)$ in x and y directions respectively, and a binary matrix $s_i(x, y)$ that includes segmentation information. So, $g(x, y)$ can be represented as Eq. 3.9.

$$g(x, y) = \sum\nolimits_{i=0}^{k} f(x - \alpha_i, y - \beta_i) * s_i(x, y) \tag{3.9}$$

A 2D Hanning window function is applied to input images $f(x, y)$ and $g(x, y)$ to reduce the effect of discontinuity at image border [12]. The Hanning window $w(x, y)$ can be defined as:

$$w(x, y) = \frac{1 + cos(\frac{\pi x}{M})}{2} \frac{1 + cos(\frac{\pi y}{N})}{2} \tag{3.10}$$

Let $F(u, v)$ and $G(u, v)$ denote the 2D DFT of the two images. Considering the properties of DFT $\mathcal{F}$, $F(u, v)$ and $G(u, v)$ can be expressed as Eq. 3.11.

$$F(u, v) = \mathcal{F}(f(x, y)w(x, y)) \quad G(u, v) = \mathcal{F}(g(x, y)w(x, y)) \tag{3.11}$$

Next, extract the phase component of $F(u, v)$ and $G(u, v)$, the functions are divided by the amplitude, as Eq. 3.12. The phase spectrum is shown in Fig. 3.7.

$$\hat{F}(u, v) = \frac{F(u, v)}{|F(u, v)|} \quad \hat{G}(u, v) = \frac{G(u, v)}{|G(u, v)|} \tag{3.12}$$

FIPOC will calculate the phase difference spectrum $\hat{r}(u, v)$ between $\hat{F}(u, v)$ and $\hat{G}(u, v)$ immediately (the dotted line in Fig. 3.5). But when the displacement of each region is different, there will be several dirac delta functions in

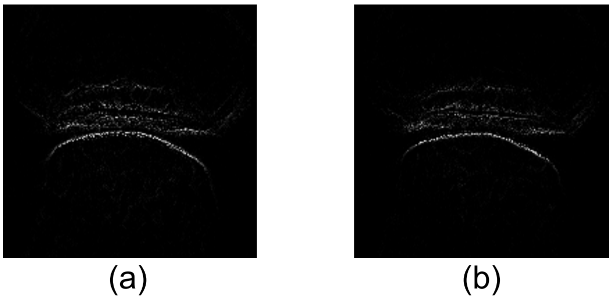


Figure 3.7: (a) The phase spectrum of the baseline radiographic image. (b) The phase spectrum of the follow-up radiographic image.

phase difference spectrum, as show in Eq. 3.13.

$$\hat{r}(u,v) = \sum\nolimits_{i=0}^{k} p_i \delta(\alpha_i, \beta_i) \tag{3.13}$$

Different from FIPOC, PIPOC segments the phase spectrum in spatial domain. Next, the phase spectrum $\hat{f}(x,y)$ of image $f(x,y)$ and the phase spectrum $\hat{g}(x,y)$ of image $g(x,y)$ in spatial domain are obtained by Inverse Discrete Fourier Transform (IDFT) $\mathcal{F}^{-1}$.

$$\hat{f}(x,y) = \mathcal{F}^{-1}(\hat{F}(u,v)) \quad \hat{g}(x,y) = \mathcal{F}^{-1}(\hat{G}(u,v)) \tag{3.14}$$

Segmenting region i by using segmentation matrix $s_i(x,y)$, as shown in Fig. 3.8 and Eq. 3.15

$$\hat{f}_i(x,y) = \hat{f}(x,y) * s_i(x,y) \quad \hat{g}_i(x,y) = \hat{g}(x,y) * s_i(x,y) \tag{3.15}$$

Subsequently combining DFT $\mathcal{F}$ and Eq. 3.15 to develop the phase spectrum of the region i in the frequency domain.

$$\hat{F}_i(u,v) = \mathcal{F}(\hat{f}_i(x,y)) \quad \hat{G}_i(u,v) = \mathcal{F}(\hat{g}_i(x,y)) \tag{3.16}$$

The normalized cross phase spectrum $\hat{R}_i(u,v)$ of region i between $F(u,v)$ and $G(u,v)$ can be obtained respectively as given in Eq. 3.17. Here, $\overline{G_i(u,v)}$ in Eq. 3.17 denotes the complex conjugate of $G_i(u,v)$.

$$\hat{R}_i(u,v) = \frac{\hat{F}_i(u,v)\overline{\hat{G}_i(u,v)}}{|\hat{F}_i(u,v)\overline{\hat{G}_i(u,v)}|} \tag{3.17}$$

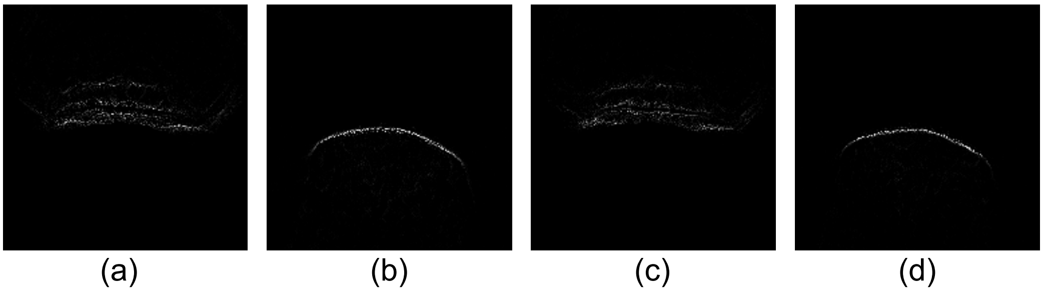


Figure 3.8: (a) The upper bone of the baseline radiographic image. (b) The lower bone of the baseline radiographic joint image. (c) The upper bone of its follow-up radiographic joint image. (d) The lower bone of its follow-up radiographic joint image.

Next, the phase difference spectrums $\hat{r_i}(x, y)$ of region i between the two images are obtained by IDFT $\mathcal{F}^{-1}$. The location of the dirac delta function δ represents the displacement between two images.

$$\begin{aligned} \hat{r_i}(x, y) &= \mathcal{F}^{-1}(\hat{R_i}(u, v)) \\ &= \delta(\alpha_i, \beta_i) \end{aligned} \tag{3.18}$$

In case of Fourier Transform (FT), the location of the peak of dirac delta function δ in the phase difference spectrum $\hat{r_i}(x, y)$ can be determined according to the maximum peak.

$$(\alpha_i{}', \beta_i{}') = \underset{(x,y)}{\arg\max}\, \hat{r_i}(x, y) \tag{3.19}$$

Consider the DFT, the least-square fitting method employed to estimate displacement (α_i, β_i) around the maximum peak $(\alpha_i{}', \beta_i{}')$. Since the δ function has a very sharp peak, limited number of data points 5×5 are used to fit δ function [12] in this work. Thus, the JSN_{fg} between image $f(x, y)$ and image $g(x, y)$ can be quantified according to the displacement difference between the proximal phalanx bone $s_0(x, y)$ and the metacarpal bone $s_1(x, y)$, as show below.

$$\text{JSN}_{fg} = \beta_0 - \beta_1 \tag{3.20}$$

accuracy. The loss of the two networks were optimized with the ground truth separately.

3.3 Joint angle correction by RIPOC

As reported in § 3.2, the PIPOC can simultaneously monitor the movement of multiple objects with sub-pixel accuracy. However, the experiments in § 5.2.2 shows that the inconsistent angle between the upper and lower bones of the joints can affect the accuracy of JSN progression quantification, and it may lead to mismatches. Although by using guide lines we can standardize hand posture when taking radiographic images, thus effectively reducing the number of inconsistent joint angles. Considering that RA can cause ankylosis in peripheral synovial joints and impair the movement and flexibility of digits, consistent joint angle may be difficult for RA patients with limited finger function. We introduced RIPOC [13] based joint angle correction in the JSN progression quantification to control mismatch cases.

The basic pipeline of the JSN progression quantification with joint angle correction can be described as shown in Fig. 3.9. Consider two joint radiographic images, a baseline joint radiography $f(x, y)$ and a follow-up joint radiography $g(x, y)$. The joint angle of $f(x, y)$ and $g(x, y)$ is different. These two joint radiographic images can be divided into upper and lower bones. The segmentation information is included in a set of binary matrix $s_i(x, y)$ ($i = 0$ indicates upper bone and $i = 1$ indicates lower bone). The θ_0 and θ_1 represent the angle difference of upper and lower bone respectively. The (α_0, β_0) and (α_1, β_1) represent sub-pixel displacement of upper and lower bones in (x, y) directions respectively. Then, the follow-up joint radiography $g(x, y)$ can be represented by the baseline joint radiography $f(x, y)$ as follow.

$$g(x,y) = \sum_{i=0}^{1} f(xcos\theta_i + ysin\theta_i + \alpha_i, -xsin\theta_i + ycos\theta_i + \beta_i) * s_i(x,y) \quad (3.21)$$

As reported in [14], segmentation in frequency domain can affect control the noise in phase difference spectrum when compared with segmentation in spatial domain. Take an example of quantifying the movement of the upper bone, the phase spectrum of upper bone $\hat{f}_0(x, y)$ and $\hat{g}_0(x, y)$ can be denoted as follow.

$$\begin{aligned} \hat{f}_0(x,y) &= \mathcal{F}^{-1}(\frac{\mathcal{F}(f(x,y)w(x,y))}{|\mathcal{F}(f(x,y)w(x,y))|}) * s_0(x,y) \\ \hat{g}_0(x,y) &= \mathcal{F}^{-1}(\frac{\mathcal{F}(g(x,y)w(x,y))}{|\mathcal{F}(g(x,y)w(x,y))|}) * s_0(x,y) \end{aligned} \quad (3.22)$$

Here, $\mathcal{F}$ denotes the 2D DFT, and $\mathcal{F}^{-1}$ denotes the 2D IDFT. $w(x, y)$ denotes the window function, we used the Hanning window in this work.

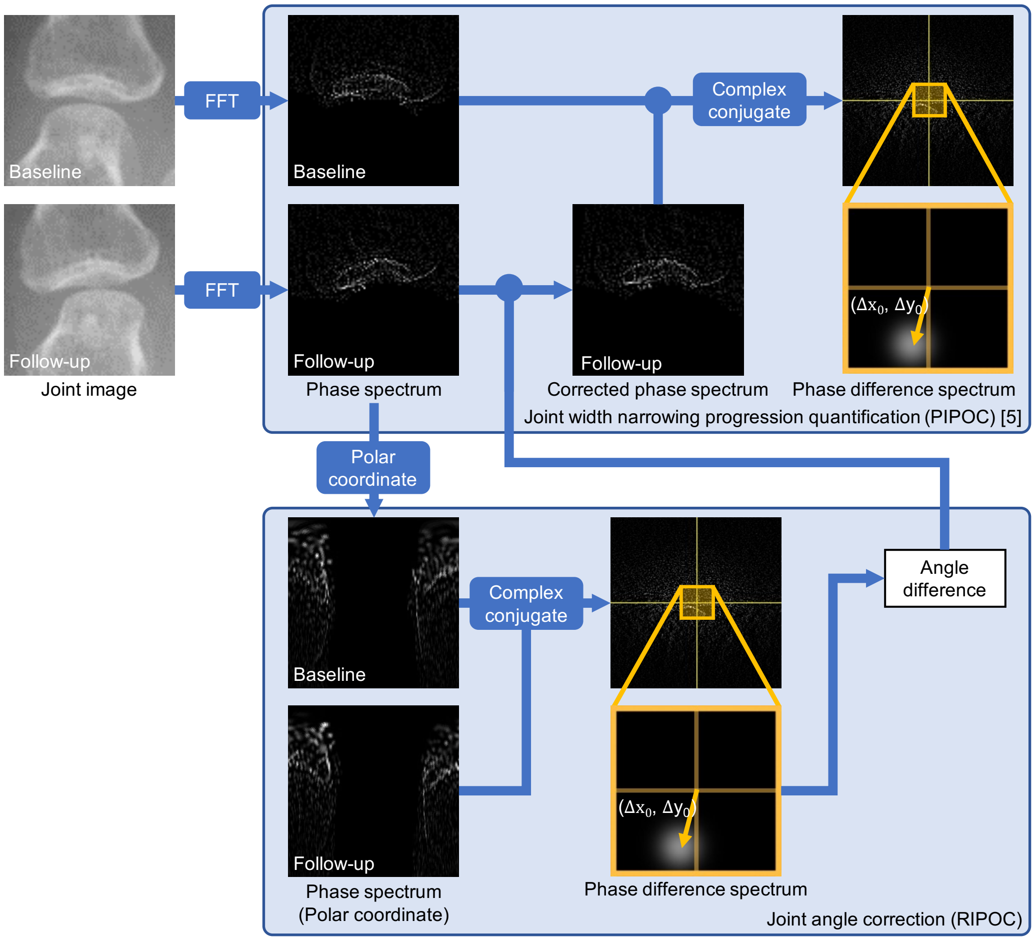


Figure 3.9: Overview of JSN progression quantification with joint angle correction.

Next, the bone angle of follow-up radiography is corrected to baseline radiography by using RIPOC [13], as shown in Fig. 3.10. Consider that the phase spectrum of upper bone of the follow-up radiography $\hat{g}_0(x,y)$ and the baseline radiography $\hat{f}_0(x,y)$ are related by:

$$\hat{g}_0(x,y) = \hat{f}_0(xcos\theta_0 + ysin\theta_0 + \alpha_0, -xsin\theta_0 + ycos\theta_0 + \beta_0) \tag{3.23}$$

According to the FT property the transform of $\hat{f}_0(x,y)$ and $\hat{g}_0(x,y)$ are related by:

$$\hat{G}_0(\xi,\eta) = e^{j2\pi(\xi\alpha_0+\eta\beta_0)} \times \hat{F}_0(\xi cos\theta_0 + \eta sin\theta_0, -\xi sin\theta_0 + \eta cos\theta_0) \tag{3.24}$$

Let $M_{\hat{F}_0}(\xi,\eta)$ and $M_{\hat{G}_0}(\xi,\eta)$ denote the magnitudes of $\hat{F}_0(\xi,\eta)$ and $\hat{G}_0(\xi,\eta)$

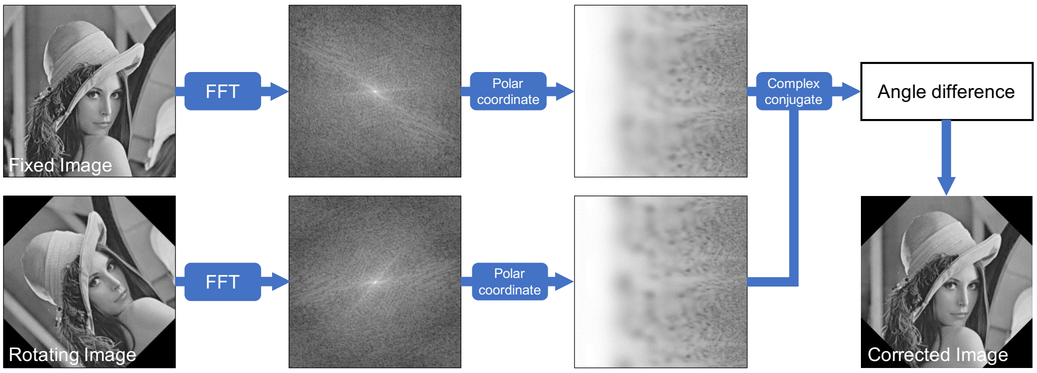


Figure 3.10: Principles of RIPOC. The rotation can be represented as translation displacement in polar coordinates. By detecting the displacement with POC, the rotation can be easily calculated and corrected.

respectively. Combining the Eq. 3.24, then, we can represent as follow.

$$M_{\hat{G}_0}(\xi,\eta) = M_{\hat{F}_0}(\xi cos\theta_0 + \eta sin\theta_0, -\xi sin\theta_0 + \eta cos\theta_0) \tag{3.25}$$

Rotation can be represented as a translation displacement in polar coordinates, as given in Eq. 3.26.

$$M_{\hat{G}_0}(\rho,\theta) = M_{\hat{F}_0}(\rho,\theta-\theta_0) \tag{3.26}$$

As shown in Fig. 3.10, combining the Eq. 3.26 and FIPOC, the angle difference θ_0 can be easily calculated. Then, the corrected phase spectrum of upper bone of follow-up radiography $\hat{g}_0'(x,y)$ can be represented as given in Eq. 3.27.

$$\hat{g}_0'(x,y) = \hat{f}_0(x+\alpha_0, y+\beta_0) \tag{3.27}$$

Then normalized cross phase spectrum $\hat{R}_i(u,v)$ of $\hat{g}_0'(x,y)$ and $\hat{f}_0(x,y)$ with FT can be defined as:

$$\hat{R}_0(u,v) = \frac{\mathcal{F}(\hat{f}_0(x,y))\overline{\mathcal{F}(\hat{g_0}'(x,y))}}{|\mathcal{F}(\hat{f}_0(x,y))\overline{\mathcal{F}(\hat{g_0}'(x,y))}|} = e^{j2\pi(\xi\alpha_0+\eta\beta_0)} \tag{3.28}$$

Therefore, the location of the peak of dirac delta function δ in the phase difference spectrum can be determined as

$$(\alpha_0', \beta_0') = \underset{(x,y)}{\arg\max}\, \mathcal{F}^{-1}(\hat{R}_0(u,v)) \tag{3.29}$$

The least-square fitting method is employed to estimate the displacement (α_0, β_0) around the maximum peak (α_0', β_0'). The same method can be used to measure the displacement (α_1, β_1) of lower bone. Therefore, the JSN_{fg} between image $f(x,y)$ and image $g(x,y)$ can be quantified as Eq. 3.30.

$$\mathrm{JSN}_{fg} = \beta_0 - \beta_1 \tag{3.30}$$

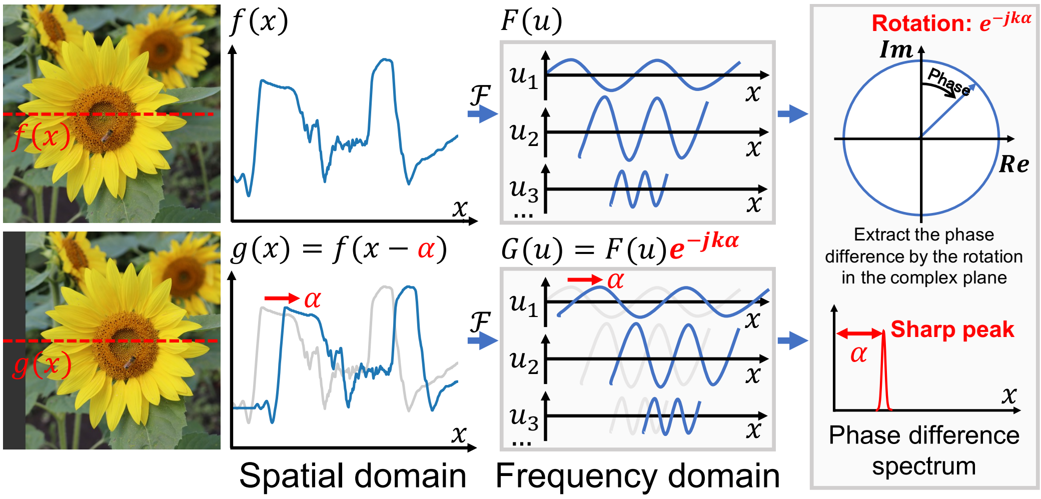


Figure 3.11: Principles of FIPOC. Consider two signals $f(x)$ and $g(x)$ with α displacement. Since each wave has the same phase difference, the displacement can be measured by the location of the dirac delta function in the phase difference spectrum.

3.4 FIPOC and PIPOC

Previously, we had proposed a JSN progression quantification method [15,16], which is based on FIPOC [12, 13, 17]. FIPOC is a well-known method for image registration, it can estimate the relative displacement between two images and it is based on the frequency domain analysis.

Figure 3.11 illustrates the principles of FIPOC using a 1-D signal analysis. Considering that there is a single displacement α in the spatial domain between the two signals $f(x)$ and $g(x)$, this displacement appears in the phase difference spectrum as a dirac delta function with coordinate α. FIPOC can measure the displacement by quantifying the coordinate of the dirac delta function. In ideal conditions (no noise, same image content), the dirac delta function in the phase difference spectrum can have exactly low phase dispersion. Thus, FIPOC can have an error range within 0.01 pixel when measured on a set of 200 pixels $\times$ 200 pixels images (the accuracy can improve with higher image resolution) [12].

Nevertheless, when the displacement between two signals is different in different regions, the phase dispersion will appear in the phase difference spectrum, as shown in Fig. 3.12. Take the baseline MCP joint image $f(x, y)$ and its follow-up image $g(x, y)$ as an example (like Fig. 3.13), let α_0 and α_1 represent the vertical displacement of the proximal phalanx bone and metacarpal bone respectively. When α_0 and α_1 is different, the displacement

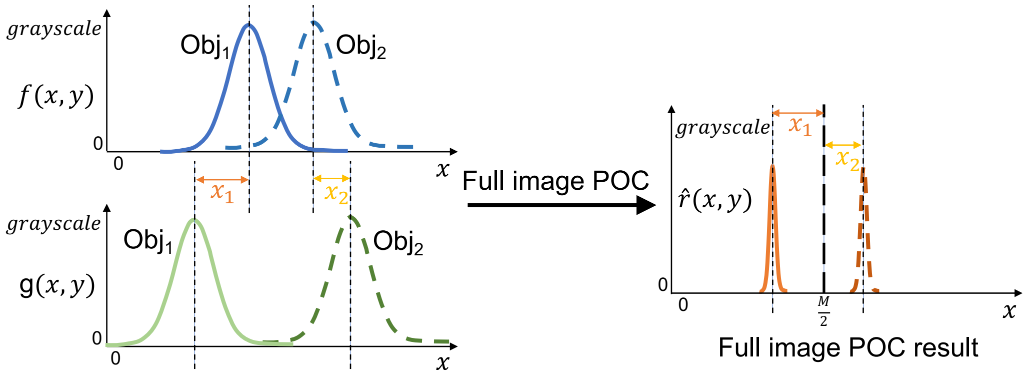


Figure 3.12: When the displacement between two signals is different in different regions, the phase dispersion will appear in the phase difference spectrum when using FIPOC.

of each region will show a dirac delta function in the phase difference spectrum, its center coordinate represents the displacement, and its intensity is positively related to the region. Thus, the JSN progression can be quantified by calculating the distance difference between these two dirac delta function. This basic idea is widely be used in image processing. In [18] Shimada *et al.* proposed a sparse disparity estimation algorithm based on FIPOC for depth prediction in binocular stereo vision. [18] estimate sparse disparity in different depths of field by separately calculating the positions of multiple dirac delta functions in phase difference spectrum. However, considering that the dirac delta functions in close proximity can affect each other and even overlap, as shown in Fig. 3.14. And could reduce the accuracy of their coordinate measurement. The precise position of each dirac delta function can be obtained if and only if the displacement differences between multiple regions are large enough (about 3 pixels [18]).

When the displacement difference is small and cannot be simply quantified by using the above method, then alternately in previous works they would segment images in spatial domain [15]. Compared to the FIPOC method to directly quantify the displacements of different regions in the phase difference spectrum, the proposed method combines FIPOC, spatial domain segmentation and image in-painting algorithm to quantify smaller displacement differences, additionally, it can have wider application prospects in issues where the displacement differences are small (like JSN progression quantization). Indeed, this approach can quantify the small displacements of multiple regions from independent images respectively. However, if we don't fill vacant space by using image in-painting algorithm, we cannot apply Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) for FIPOC

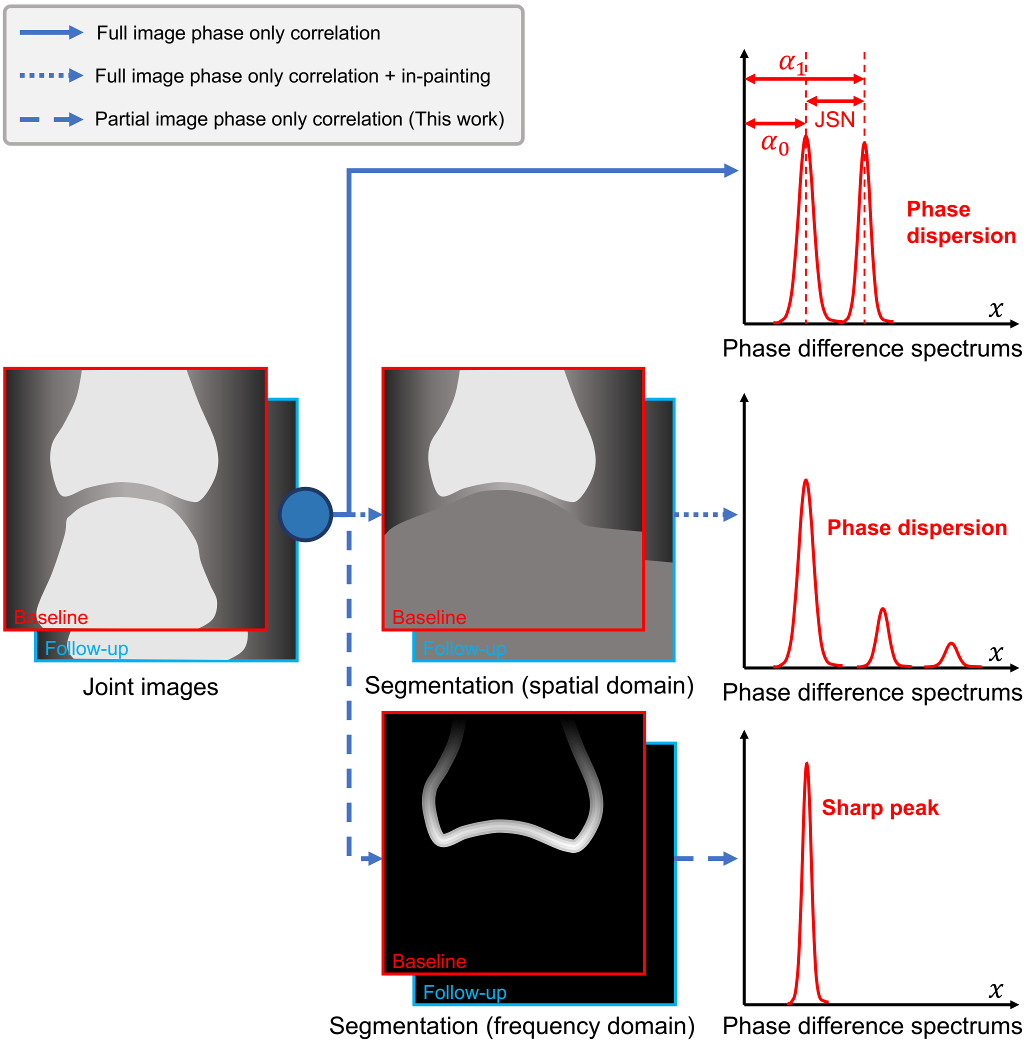

Full image phase only correlation
Full image phase only correlation + in-painting
Partial image phase only correlation (This work)
α_1
α_0
JSN
Phase dispersion
x
Phase difference spectrums
Baseline
Follow-up
Joint images
Segmentation (spatial domain)
Phase dispersion
Sharp peak
Segmentation (frequency domain)

Figure 3.13: Take two MCP joint image as an example, a schematic diagram of the comparison of the dirac delta function in the phase difference spectrum when using FIPOC (without segmentation), FIPOC (segment in spatial domain and combine with in-painting algorithm) or PIPOC (segment in frequency domain) when there are multiple regions with different displacement.

processing. The in-painting algorithm can generate some non-existent phase features, and thus lead to phase dispersion in phase difference spectrum. This phase dispersion can increase quantification error and even cause mismatches (the second row in Fig. 3.13). Therefore, we have found a method to exploit the effectiveness of FIPOC, no matter what shape the segmentation is in.

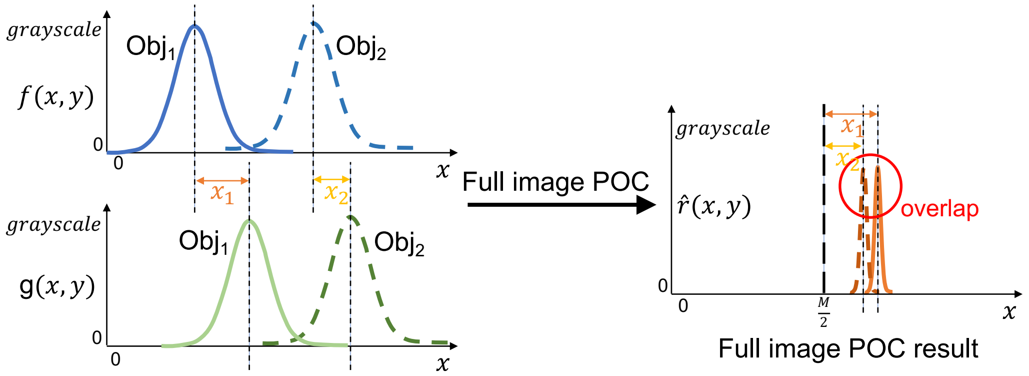


Figure 3.14: When the different of the displacements between two signals is small, the dirac delta functions in close proximity can affect each other and even overlap when using FIPOC.

PIPOC, the algorithm proposed in § 3.2, can determine the location of each dirac delta functions and avoids the impact of the various object segmentation including any form background on accuracy in the phase spectrum. In our JSN progression quantification experiments, PIPOC has a lower phase dispersion compared to the method combining FIPOC, spatial domain segmentation and in-painting algorithm. PIPOC can avoid the effects of in-painting algorithms, and obtain a sharper delta function in phase spectrum (the third row in Fig. 3.13). The proposed PIPOC can further improve the accuracy and robustness of the JSN progression quantification.

3.5 Measurement reliability quantification

In this section, we propose a metric to quantify the reliability of the JSN progression quantification with joint angle correction. This metric can be used to evaluate the probability of mismatch. This reliability metric is determined based on phase dispersion. According to the experiments in [14], changes in bone contours can affects the accuracy of JSN progression quantification, and result in phase dispersion in the phase difference spectrum. This phase dispersion appears as multiple low peaks in the phase difference spectrum instead of a single Dirac delta function. This phase dispersion can affect the Dirac delta function position quantization and cause measurement errors or even mismatches. Figure 3.15 shows two 3D graph of the phase difference spectrum for the clinical images, and the maximum peak value for each spectrum is normalized as 255. By comparing the phase difference spectrums in Figure 3.15, we can easily observe that the phase dispersion

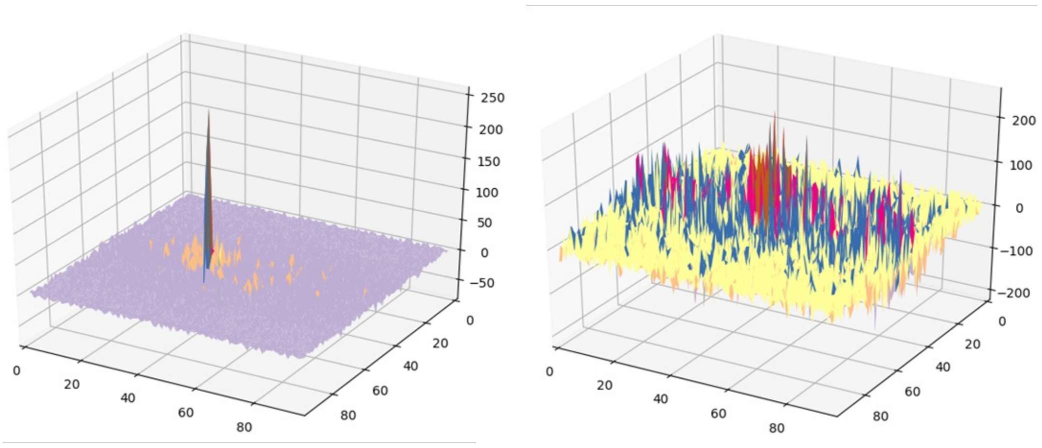


Figure 3.15: The difference between measurement mismatch and success on 3D graph of the phase difference spectrum. The left spectrum is a successful measurement, and the measurement on the right is mismatched. Successful measurement has lower phase dispersion, which manifests as sharper Dirac delta function peak, and higher signal-to-noise ratio (the ratio of the main Dirac delta function peak to other peaks).

of a mismatch case is clearly higher than that of the success case. In this study, the phase dispersion is used to represent a confidence index using SD. For the normalized image, to suppress the influence of the main Dirac delta function, the SD of the entire image is obtained by setting the 3 × 3 pixels around the peak as 0, and the sum of the upper and lower bones are used as the confidence index.

For the confidence index, the mean value for the measured clinical image dataset is 40.83, whereas the mean for the 10 pairs of images with confirmed mismatches are 79.42. Figure 3 shows an example of the joint images that are measured and their confidence levels re-spectively. The SD with no rotation is 22.12, while the SD with rotation is 50.82, indicating that the SD increases as the mismatch probability increases. We invite one radiologist and one radiological technologist in our experiment. According to the confidence index, we divide the JSN progression quantitative results into high index group (mismatch group) and low index group (success group). Both the radiologist and the radiological technologist find that the JSN progression quantitative results in the low index group are more consistent with the trends they assessed. In the literature, there are no means to discriminate the mismatch of a successful case in POC. But the proposed confidence index can help JSN progression quantification algorithm to be applied in clinical environment, and improve the accuracy of the measurement by giving

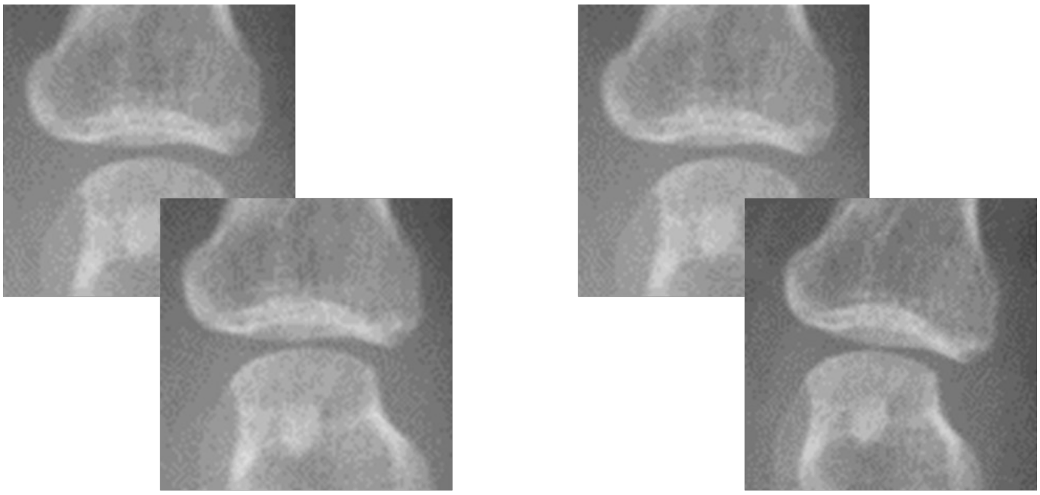


Figure 3.16: The confidence index of a successful case (left: 22.12) and a mismatch case (right:50.82).

a reference for the reliability. In addition, the best quantization result can be selected from multiple quantization results according to the confidence index, which will result in higher accuracy.

The proposed confidence index for JSN progression quantification base on PIPOC. The confidence index is defined by using the SD for the phase difference spectrum of POC. By focusing on each mismatch factor, we focus on the characteristic that the POC output is not a Dirac delta function but an output with many low peaks due to pattern matching failure, and use the SD, which is a variability index, as a measurement reliability index.

Bibliography

[1] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[2] Robert M Haralick, Stanley R Sternberg, and Xinhua Zhuang. Image analysis using mathematical morphology. *IEEE transactions on pattern analysis and machine intelligence*, (4):532–550, 1987.

[3] Jack Sklansky and Victor Gonzalez. Fast polygonal approximation of digitized curves. *Pattern Recognition*, 12(5):327–331, 1980.

[4] Derek York. Least squares fitting of a straight line with correlated errors. *Earth and planetary science letters*, 5:320–324, 1968.

[5] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001.

[6] Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. In *Proceedings. international conference on image processing*, volume 1, pages I–I. IEEE, 2002.

[7] Rainer Lienhart, Alexander Kuranov, and Vadim Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *joint pattern recognition symposium*, pages 297–304. Springer, 2003.

[8] Constantine P Papageorgiou, Michael Oren, and Tomaso Poggio. A general framework for object detection. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pages 555–562. IEEE, 1998.

[9] Thomas Huang, GJTGY Yang, and Greory Tang. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(1):13–18, 1979.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[11] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. *Advances in neural information processing systems*, 31, 2018.

[12] Kenji Takita, Takafumi Aoki, Yoshifumi Sasaki, Tatsuo Higuchi, and Koji Kobayashi. High-accuracy subpixel image registration based on phase-only correlation. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 86(8):1925–1934, 2003.

[13] B Srinivasa Reddy and Biswanath N Chatterji. An fft-based technique for translation, rotation, and scale-invariant image registration. *IEEE transactions on image processing*, 5(8):1266–1271, 1996.

[14] Yafei Ou, Prasoon Ambalathankandy, Ryunosuke Furuya, Seiya Kawada, Tianyu Zeng, Yujie An, Tamotsu Kamishima, Kenichi Tamura, and Masayuki Ikebe. A sub-pixel accurate quantification of joint space narrowing progression in rheumatoid arthritis. *arXiv preprint arXiv:2205.09315*, 2022.

[15] Yafei Ou, Prasoon Ambalathankandy, Takeshi Shimada, Tamotsu Kamishima, and Masayuki Ikebe. Automatic radiographic quantification of joint space narrowing progression in rheumatoid arthritis using poc. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 1183–1187. IEEE, 2019.

[16] Aimi Taguchi, Shun Shishido, Yafei Ou, Masayuki Ikebe, Tianyu Zeng, Wanxuan Fang, Koichi Murakami, Toshikazu Ueda, Nobutoshi Yasojima, Keitaro Sato, et al. Quantification of joint space width difference on radiography via phase-only correlation (poc) analysis: a phantom study comparing with various tomographical modalities using conventional margin-contouring. *Journal of Digital Imaging*, 34(1):96–104, 2021.

[17] Hassan Foroosh, Josiane B Zerubia, and Marc Berthod. Extension of phase correlation to subpixel registration. *IEEE transactions on image processing*, 11(3):188–200, 2002.

[18] Takeshi Shimada, Masayuki Ikebe, Prasoon Ambalathankandy, Masato Motomura, Tetsuya Asai, et al. Sparse disparity estimation using global phase only correlation for stereo matching acceleration. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1842–1846. IEEE, 2018.

Chapter 4

Materials

Imaging phantom based experiments were studied to evaluate our algorithm's performance. From our experiments we observed that the manually labeled JSW has pixel level mean error, and it is discussed in § 5.2.1. To evaluate the performance of our algorithm, we prepared phantom and clinical images. Phantom images with ground truth were used to measure accuracy in terms of absolute error.

4.1 Phantom study

Imaging phantoms are specially designed physical test objects for simulating anatomical structures of the body, which makes them easier and more convenient to use. The two-layer phantom can simulate the X-ray absorption coefficient (CT value) of bone cortex and cancellous bone. This is an advantage in X-ray imaging due to dose issues, phantom devices can be thus tested without using human volunteers.

4.1.1 Phantom design

A phalanx-shaped phantom was produced using vacuum-sintered bodies of a novel apatite called Titanium Medical Apatite (TMA) [1]. The chemical formula of TMA is $Ca_{10}(PO_4)_6 \cdot TiO_2 \cdot (OH)_2 \cdot nH_2O$. TMA powder was kneaded with distilled water, and solid cylinders of compacted TMA were formed by compression molding at $10MPa$. TMA was vacuum sintered using a resistance furnace at about $10^{-3}Pa$.

Using TMA to design imaging phantom has the following advantages: (1) The CT value of phantom in radiographs can be easily modified by changing the ratio of TMA and adhesive. (2) TMA bodies are easy to process and

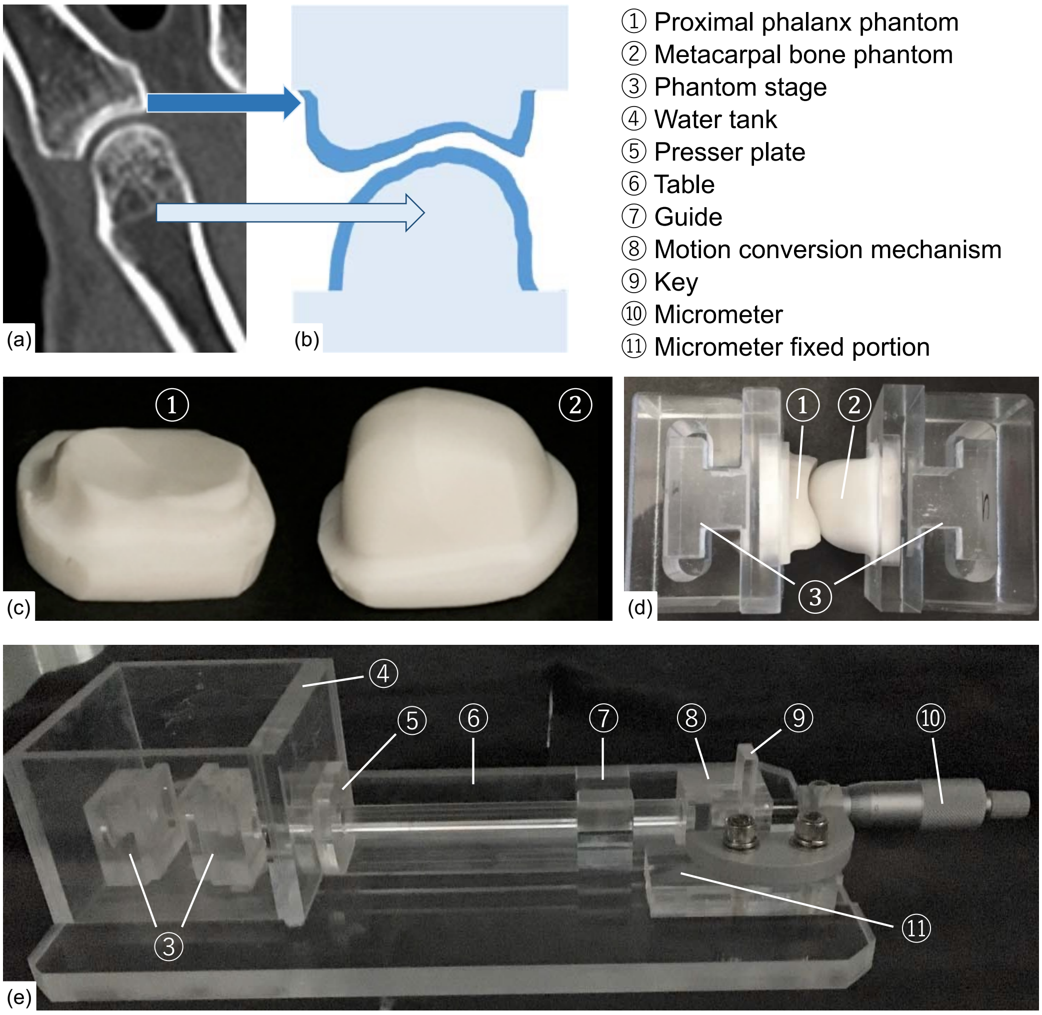

① Proximal phalanx phantom
② Metacarpal bone phantom
③ Phantom stage
④ Water tank
⑤ Presser plate
⑥ Table
⑦ Guide
⑧ Motion conversion mechanism
⑨ Key
⑩ Micrometer
⑪ Micrometer fixed portion
(a)
(b)
(c)
(d)
(e)
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪

Figure 4.1: A MCP joint-shaped two-layer phantom design and phantom imaging environment. (a) A MCP joint in radiographic clinical imaging. (b) A diagram of two-layer structure bone (dark blue: Bone cortex, light blue: Cancellous bone). (c) A set of MCP joint-shaped two-layer phantom. (d) The phantom joint connect with the attaching portion. (e) The phantom imaging environment.

model with a 3D-modeling machine or a lathe. (3) TMA vacuum-sintered bodies has a density of approximately $2300kg/m^3$ (corresponding to that of a compact bone or a tooth).

The phantom used in our experiment is a two-layer TMA vacuum-sintered body to simulate the CT value of bone cortex and cancellous bone. The diagram of the two-layered bone is shown in Fig. 4.1 (a) and (b). The phantom

Table 4.1: Phantom design preparation

	Bone cortex	Cancellous bone
TMA : adhesive	1:1.2	1:5
Particle Size (μm)	107 $\sim$ 250	107 $\sim$ 250
Temperature (K)	1370	1370

mimics MCP joint, proximal phalanx, and the metacarpal bone. The assembled phantoms are shown in Fig. 4.1 (c). The important properties of our phantom bones are given in Table 4.1 [1].

4.1.2 Imaging environment

The phantom joint was mounted on to the stage as shown in Fig. 4.1 (d). The phantom stage was connected to a micrometer, and thereby the JSW of phantom could be easily adjusted using the micrometer controls. JSW range is up to 13 mm, and has a minimum scale of 0.01 mm. There is substantial evidence that JSW has a close relationship with age and sex in healthy populations [2, 3]. In addition, RA is more frequent in females who are between 30 and 50 years of age, and their JSW is around 1.70 mm [2, 3]. In our work, JSW standard of phantom was set as 1.70 mm. Following two sets of phantom images with different specifications were provided. (i) JSW range: 1.20 mm - 2.20 mm, and increment step size: 0.10 mm (ii) JSW range: 1.65 mm - 1.75 mm, and increment step size: 0.01 mm.

Figure 4.2 shows some MCP joint phantom images that were used in our experiments. Clinical researches show that tomosynthesis is superior to radiography for the manual evaluation of JSN and the manual detection of bone erosion in patients with RA [4, 5]. However, considering that radiography is the most widely used imaging technique in the diagnosis of RA, we utilized both imaging techniques (radiography and tomosynthesis) in our phantom experiments.

The radiographic imaging device used in our phantom experiment is *FUJIFILM DR CALNEO Smart C47* from *Fujiflm* Corporation, Tokyo, Japan, with a 1.5 mm X-ray aluminum filter thickness. The tomosynthesis imaging device is *Side Station i3* from *Shimadzu* Corporation, Kyoto, Japan. The reconstruction function used in tomosynthesis is IR Hand F, and the effective slice thickness is 2 mm. Digital Imaging and Communications in Medicine (DICOM) standard was used in managing our datasets in phantom study. The spatial resolution used in our radiographic phantom study is 0.15 mm/pixel at 12 bit depth. And 0.26 mm/pixel at 12 bit depth in our

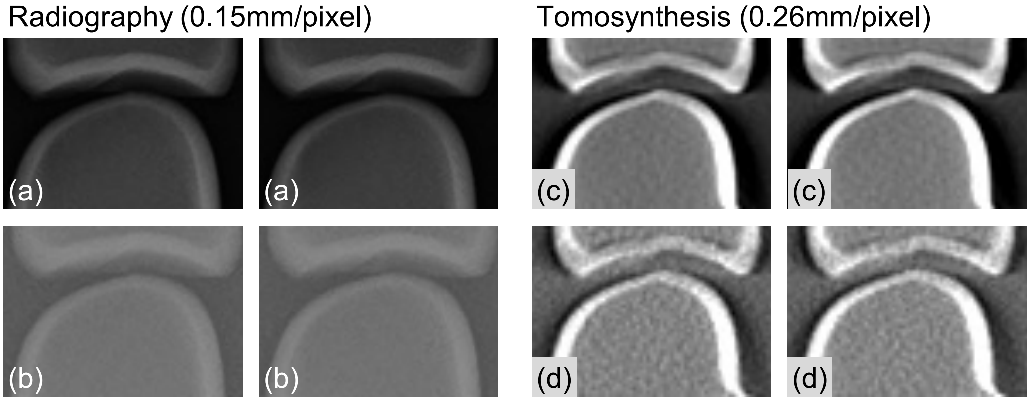


Figure 4.2: MCP joint phantom images in different imaging conditions (tank filled with air or water). Radigraphic imaging: (a) air (b) water. Tomosynthesis: (c) air (d) water. In each group we show two images with different JSW, left: 1.20mm, right : 1.30mm.

Table 4.2: Radiographic imaging configuration parameters

	Radiography		Tomosynthesis
	Phantom	Clinical	Phantom
Tube voltage (kV)	50	42	47
Tube current (mA)	100	100	250
Exposure time (mSec)	20	20	105
Source to image (cm)	100	100	N/A

tomosynthesis dataset. For detailed imaging parameter descriptions, please refer to Table 4.2.

In phantom-based experiments, filling air is easier to implement, for comparison with existing and possible future air-filled phantom studies, we imaged phantom in the air. Nevertheless, in clinical studies, the X-ray beam can be attenuated by the tissue, these attenuations are displayed as noise in the radiography. In related phantom studies, water is usually used to simulate the noise generated by the beam attenuation in the tissue [6, 7]. In our phantom study, the phalanx-shaped phantom was mounted on the stage as shown in Fig. 4.1 (d), and placed in a tank. We can image the phantom with low noise when the tank is filled with air, or filled with distilled water which has an X-ray absorbing properties similar to normal tissue. Our experimental phantom imaging setup is shown in Fig. 4.1 (e).

Table 4.3: Patient information in the clinical dataset

	Mean ± SD	Range
Age at enrollment (year)	55.83 ± 13.86	20.68 ∼ 88.00
Number of Photography*	4.30 ± 2.54	3 ∼ 17
Treatment Duration (year)	4.01 ± 3.43	0.88 ∼ 12.10

* Patients did two-handed or one-handed radiographic imaging.

4.2 Clinical dataset

4.2.1 Study population

For clinical assessment, we prepared dataset from *Sagawa Akira Rheumatology Clinic* (Sapporo, Japan), *Sapporo City General Hospital* (Sapporo, Japan) and *Hokkaido Medical Center for Rheumatic Diseases* (Sapporo, Japan). This dataset contains 1120 hand posteroanterior projection radiographs from patients diagnosed with early-stage RA. All images were used in the joint position detection experiments. Considering that several images were required to evaluate our work when calculating SD. Thus, images of patients who were radiographed at least three times were retained, which contains 549 hand posteroanterior projection radiographs of 77 RA patients out of which 88.0% are female. Detailed patients information are summarized in Table 4.3 (please note, the gender and age information of a small number of patients were not included upon patient request).

This study was conducted in accordance with the guidelines of the Declaration of Helsinki and approved by the Ethics Committee of the Faculty of Health Sciences, Hokkaido University (approval number: 19 - 46).

4.2.2 Imaging environment

The radiographic imaging device used in our clinical study is *DR-155HS2-5* from *Hitachi* Corporation, Tokyo, Japan, with 1.5 mm X-ray aluminum filter thickness. The centering point of the X-ray beam was the MCP joint of the middle finger. DICOM standard was used in managing our cilical dataset, and the image resolution is 2010 pixels × 1490 pixels, and a 0.175 mm × 0.175 mm pixel size at 12 bit depth. For detailed imaging parameter descriptions, please refer to Table 4.2.

Bibliography

[1] Kenichi Tamura, Tatsushi Fujita, and Yuriko Morisaki. Vacuum-sintered body of a novel apatite for artificial bone. *Open Engineering*, 3(4):700–706, 2013.

[2] Tore K Kvien, Till Uhlig, SIGRID ØDEGÅRD, and Marte S Heiberg. Epidemiological aspects of rheumatoid arthritis: the sex ratio. *Annals of the New York Academy of Sciences*, 1069(1):212–222, 2006.

[3] Alexander Pfeil, Joachim Böttcher, Bettina E Seidl, Jens-Peter Heyne, Alexander Petrovitch, Torsten Eidner, Hans-Joachim Mentzel, Gunter Wolf, Gert Hein, and Werner A Kaiser. Computer-aided joint space analysis of the metacarpal-phalangeal and proximal-interphalangeal finger joint: normative age-related and gender-specific data. *Skeletal radiology*, 36(9):853–864, 2007.

[4] Yohei Ono, Rina Kashihara, Nobutoshi Yasojima, Hideki Kasahara, Yuka Shimizu, Kenichi Tamura, Kaori Tsutsumi, Kenneth Sutherland, Takao Koike, and Tamotsu Kamishima. Tomosynthesis can facilitate accurate measurement of joint space width under the condition of the oblique incidence of x-rays in patients with rheumatoid arthritis. *The British Journal of Radiology*, 89(1062):20150967, 2016.

[5] Takatoshi Aoki, Masami Fujii, Yoshiko Yamashita, Hiroyuki Takahashi, Hodaka Oki, Yoshiko Hayashida, Kazuyoshi Saito, Yoshiya Tanaka, and Yukunori Korogi. Tomosynthesis of the wrist and hand in patients with rheumatoid arthritis: comparison with radiography and mri. *American Journal of Roentgenology*, 202(2):386–390, 2014.

[6] Rodney A Brooks and Giovanni Di Chiro. Statistical limitations in x-ray reconstructive tomography. *Medical physics*, 3(4):237–240, 1976.

[7] David A Chesler, Stephen J Riederer, and Norbert J Pelc. Noise due to photon counting statistics in computed x-ray tomography. *Journal of computer assisted tomography*, 1(1):64–74, 1977.

Chapter 5

Experiments and discussion

5.1 Joint position detection

5.1.1 Classification based joint position detection

The results of joint location detection are shown in Fig. 5.1, the false negative of classifier decreases gradually with the increase of the cascade layers, and false positive gradually increase after 22 levels. We selected the 22-layer classifier for joint position detection, which has a false negative ratio and a false positive ratio of 0.31% and 0.50% respectively. The performance on each joint is shown in the Table 5.1. From this table, we can observe that false positives occurred mainly in the carpometacarpal joint of the thumb,

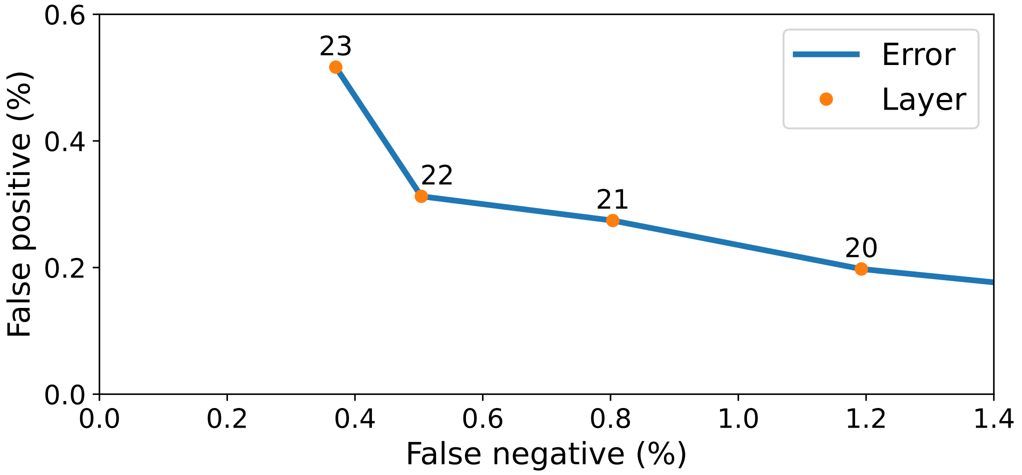


Figure 5.1: The variation in false positive and false negative ratios with the increase of the cascade layers.

Table 5.1: False negative and false positive counts and ratios of joint location detection

	False Negative				False positive	
	IP	DIP	PIP	MCP	CMC	Others
Thumb	24	N/A	N/A	5	63	2
Index	N/A	0	2	0	0	7
Middle	N/A	1	1	0	0	2
Ring	N/A	2	0	2	0	4
Small	N/A	11	1	0	0	1
Overall	**24(2.14%)**	**14(0.31%)**	**4(0.09%)**	**7(0.16%)**	**63**	**16**

* **CMC**: Carpometacarpal joint.

Table 5.2: The error of joint location detection method.

Finger	**Joint[1]**	**10 pixel**	**5 pixel**	**3 pixel**	**2 pixel**	**1 pixel**
Thumb	IP	99.76	89.86	67.15	49.52	26.57
	MCP	99.52	93.96	74.40	46.38	18.12
Index	DIP	100.00	100.00	98.07	85.27	42.27
	PIP	100.00	99.52	92.75	78.74	45.17
	MCP	100.00	96.62	81.88	59.18	27.78
Middel	DIP	100.00	99.28	96.62	84.06	40.82
	PIP	100.00	98.79	92.27	74.15	37.68
	MCP	99.52	98.07	86.47	64.49	31.16
Ring	DIP	100.00	100.00	98.55	89.13	50.72
	PIP	100.00	99.03	93.96	78.99	43.00
	MCP	100.00	99.28	86.47	65.70	25.12
Small	DIP	100.00	99.28	95.65	86.23	58.21
	PIP	100.00	98.31	91.30	73.91	38.16
	MCP	100.00	99.76	92.27	70.77	39.86
Overall		99.91	97.98	89.13	71.89	37.47

which is the joint that most closely resembles the target joints in a hand radiograph. And false negative appeared mainly in the thumb, especially the IP joint. In our opinion, the main reason for this situation is that the radiographic angle of the thumb is different from other fingers, resulting a difference in radiography. Differentiation of the joint position detection on the thumb may be effective in improving detection accuracy.

5.1.2 Key point detection based joint position detection

Table 5.2 shows the distribution of errors of the joint location detection method. As shown in this table, the proposed joint location detection method has an error of less than 3 pixels in 89.13% joints. The mean error of DIP joints and PIP joints are significantly lower than MCP joints. The mean error of joints on the thumb are significantly lower than other joints. Considering that we scaled different joints using different scales in the joint position correction network, the accuracy may be affected by the scaling, showing larger errors on MCP joints.

5.2 JSN progression quantification

5.2.1 Phantom study

Phantom images with ground truth were used in this experiment to calculate the absolute error of PIPOC , and to compare it with manual measurements. Our manual measurement experiments were performed by one radiologist and one radiological technologist after substantial training. An example and its measurement method is showed in Fig. 5.2.

In our study, the mean error E and RMSD is defined as:

$$E = \frac{2}{n(n-1)} \sum\nolimits_{f=2}^{n} (\sum\nolimits_{g=1}^{f-1} |\mathrm{JSN}_{fg} - T_{fg}|) \tag{5.1}$$

$$\mathrm{RMSD} = \sqrt{\frac{2}{n(n-1)} \sum\nolimits_{f=2}^{n} (\sum\nolimits_{g=1}^{f-1} (\mathrm{JSN}_{fg} - T_{fg})^2)} \tag{5.2}$$

Where n represents the number of phantom images in this set, JSN_{fg} is the measured JSN between image f and image g by using PIPOC or manual measurement. And T_{fg} represents the ground truth.

Figure 5.3, Fig. 5.4, Table 5.3 and Table 5.4 presents the measurement result of phantom study. The manual measurement result of the radiologist and the radiological technologist showed high similarity in terms of mean error and RMSD in multiple phantom data sets. The mean error of manual measurements are about 0.0555 mm (0.37 pixel) in low noise environment (air sets), and 0.1036 mm (0.69 pixel) in high noise environment (water sets) in radiographic phantom studies. For tomosynthesis datasets, the mean error is 0.1780 mm (0.68 pixel) in low noise environment, and 0.1969 mm (0.76 pixel) in high noise environment. This shows that visual measurement also

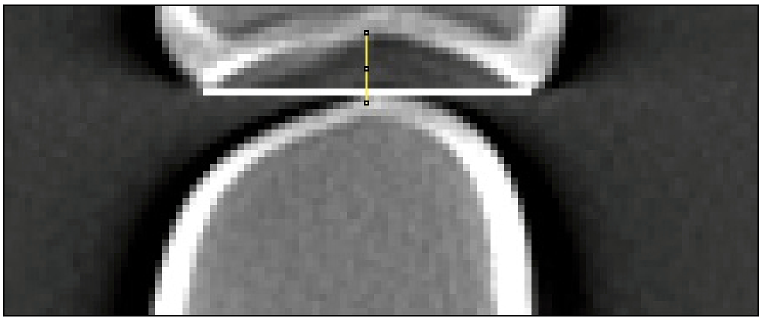


Figure 5.2: An example of manually measuring JSW (imaging device: tomosynthesis, environment: air, true JSW of phantom: 1.75 mm, manual measurement result: 10.25 pixel). Manual measurement was done once with care by one radiologist and one radiological technologist after substantial training. They did not know the ground truth of the out-of-order phantom images. They were asked to determine the center of the proximal phalanx bone phantom by drawing straight lines horizontally (the white line in figure) connecting both ends of the phantom base, then a straight line (the yellow line in figure) was drawn from the center vertically, and the JSW overlapping the straight line was measured.

can be greatly affected by the noise. On the other hand, this also indicates the manually annotated data have sub-pixel level mean error. Hence, the manually annotated ground truth may result in sub-pixel level deviation in algorithm evaluation of other works.

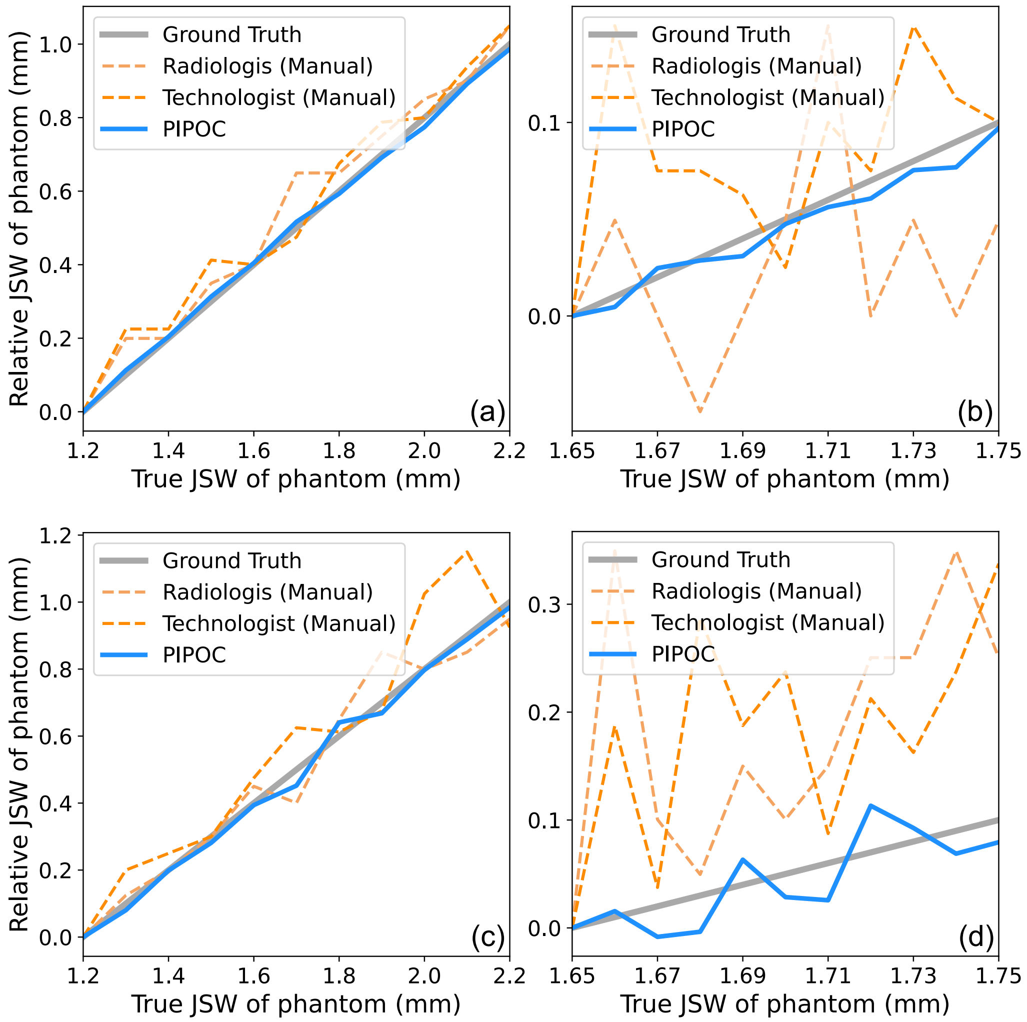


Figure 5.3: The measurement result of PIPOC and manual in phantom study when using radiographic images. Blue lines are the relative JSW of each image to the first image obtained by PIPOC . Orange dot lines are the difference of manually measured JSW between every image and the first image. The phantom of sub-figure (a), (b) is placed in air. The phantom of sub-figure (c), (d) is placed in distilled water. The true JSW of phantom is from 1.20mm to 2.20mm at increments of 0.10mm in sub-figure (a), (c). And it is from 1.65mm to 1.75mm at increments of 0.01mm in sub-figure (b), (d).

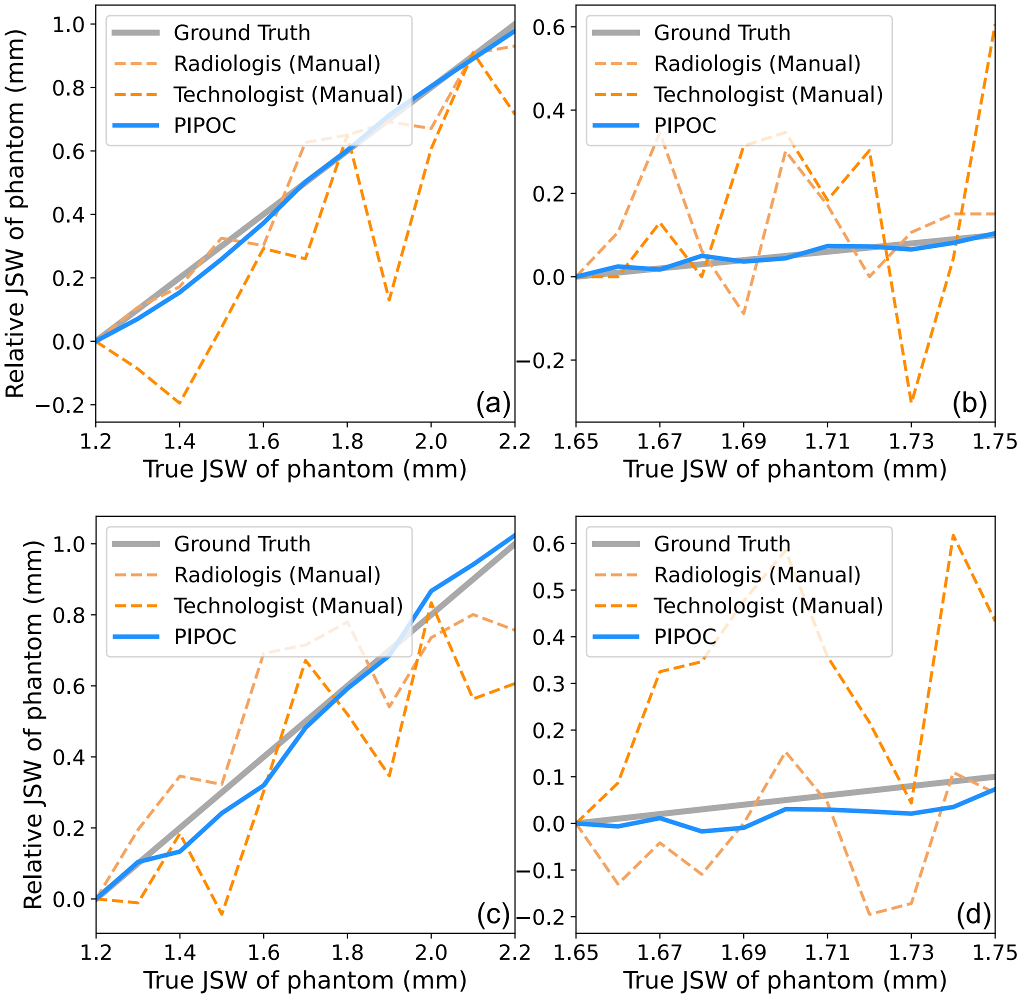


Figure 5.4: The measurement result of PIPOC and manual in phantom study when using tomosynthesis images. Blue lines are the relative JSW of each image to the first image obtained by PIPOC. Orange dot lines are the difference of manually measured JSW between every image and the first image. The phantom of sub-figure (a), (b) is placed in air. The phantom of sub-figure (c), (d) is placed in distilled water. The true JSW of phantom is from 1.20mm to 2.20mm at increments of 0.10mm in sub-figure (a), (c). And it is from 1.65mm to 1.75mm at increments of 0.01mm in sub-figure (b), (d).

Table 5.3: The mean error in millimeter for PIPOC and manual measurement when using phantom images

	Air			Water		
Radiography	Fig.5.3(a)	Fig.5.3(b)	Average	Fig.5.3(c)	Fig.5.3(d)	Average
Radiologist (Manual)	0.0509	0.0620	0.0565	0.0727	0.1196	0.0961
Technologist (Manual)*	0.0595	0.0497	0.0546	0.1186	0.1034	0.1110
Mean of Manual	0.0552	0.0559	0.0555	0.0957	0.1115	0.1036
PIPOC (Ours)	**0.0193**	**0.0066**	**0.0130**	**0.0251**	**0.0200**	**0.0226**
FIPOC [1]	0.0400	-	0.0400	-	-	-
Tomosynthesis	Fig.5.4(a)	Fig.5.4(b)	Average	Fig.5.4(c)	Fig.5.4(d)	Average
Radiologist (Manual)	0.0815	0.1477	0.1146	0.2009	0.1316	0.1662
Technologist (Manual)*	0.2155	0.2671	0.2413	0.2210	0.2342	0.2276
Mean of Manual	0.1485	0.2074	0.1780	0.2110	0.1829	0.1969
PIPOC (Ours)	**0.0180**	**0.0124**	**0.0152**	**0.0329**	**0.0243**	**0.0286**

* Measured manually by a radiological technologist.

Table 5.4: The RMSD in millimeter for PIPOC and manual measurement when using phantom images

	Air			Water		
Radiography	Fig.5.3(a)	Fig.5.3(b)	Average	Fig.5.3(c)	Fig.5.3(d)	Average
Radiologist (Manual)	0.0665	0.0758	0.0711	0.0923	0.1450	0.1186
Technologist (Manual)*	0.0709	0.0632	0.0671	0.1440	0.1237	0.1339
Mean of Manual	0.0687	0.0695	0.0691	0.1182	0.1343	0.1263
PIPOC (Ours)	**0.0220**	**0.0081**	**0.0150**	**0.0303**	**0.0245**	**0.0274**
Tomosynthesis	Fig.5.4(a)	Fig.5.4(b)	Average	Fig.5.4(c)	Fig.5.4(d)	Average
Radiologist (Manual)	0.1000	0.1835	0.1418	0.2365	0.1603	0.1984
Technologist (Manual)*	0.2616	0.3307	0.2962	0.2678	0.2800	0.2739
Mean of Manual	0.1808	0.2571	0.2190	0.2521	0.2201	0.2361
PIPOC (Ours)	**0.0220**	**0.0154**	**0.0187**	**0.0407**	**0.0292**	**0.0349**

* Measured manually by a radiological technologist.

In paper [1], only one phantom dataset (environment: air, JSW range: 1.20 mm - 2.20 mm, increment step size: 0.10 mm) is used in experiment. The mean error of FIPOC is slightly lower than manual measurement. When compared to FIPOC , PIPOC can further improve the accuracy and robustness in JSN progression quantification, by eliminating the impact of image in-painting algorithm. As show in Table 5.3 and Table 5.4, our work only has a 11.9% to 35.0% mean error, and a 11.7% to 32.0% RMSD when compared to manual measurement in radiographic phantom study. For the tomosynthesis datasets, our work only has 6.0% to 15.6% mean error, and a 6.0% to 16.1% RMSD in comparison to manual measurement. This illustrates the improved performance of JSN progression quantification when using phantom datasets. Considering the spatial resolution of radiography (0.15 mm/pixel) and tomosynthesis (0.26mm/pixel), we can notice that PIPOC performs better on tomosynthesis images when compared in pixels. We regard this difference is due to the sharper edge information of the tomosynthesis. Thus, we recommend using PIPOC in tomosynthesis preferentially with same spatial resolution.

In related works, the ground truth of joint space is usually measured by the radiologist or the rheumatologist manually. But as discussed above, manual measurement also have sub-pixel level mean error. Thus, manually measured ground truth may result in sub-pixel deviation in algorithm evaluation. This deviation is negligible when evaluating the algorithm on a pixel scale. But it can be inaccurate on a sub-pixel scale. To the best of our knowledge there are no published algorithms/methods which can compute ground truth RA joint space with sub-pixel accuracy. We propose to use the SD σ of multiple measurements to demonstrate the reliability of PIPOC without ground truth. The definition of SD can be described as follows.

In case of three images f, g and k, the JSN_{fg-k} between image f and image g can be indirectly calculated by introducing intermediate image k, as given in Eq. 5.3.

$$\mathrm{JSN}_{fg-k} = \mathrm{JSN}_{fk} + \mathrm{JSN}_{kg} \tag{5.3}$$

Considering a set of images, the $\overline{\mathrm{JSN}_{fg}}$ can be obtained by taking the average of multiple measurements.

$$\overline{\mathrm{JSN}_{fg}} = \frac{1}{n}\sum\nolimits_{k=1}^{n} \mathrm{JSN}_{fg-k} \tag{5.4}$$

So, the SD σ of JSN_{fg} is defined as Eq. 5.5.

$$\sigma_{fg} = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(\mathrm{JSN}_{fg-k} - \overline{\mathrm{JSN}_{fg}})^2} \tag{5.5}$$

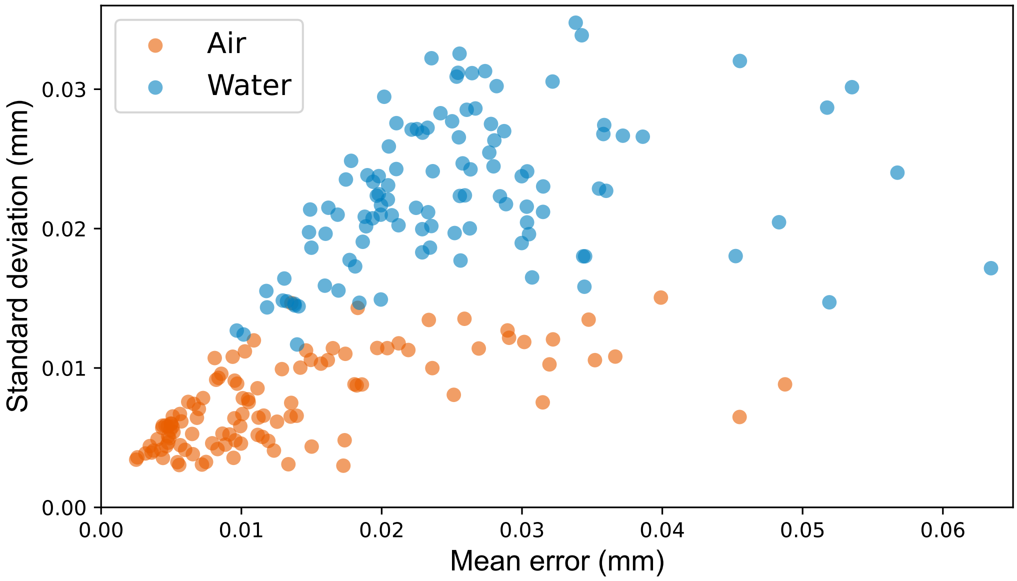


Figure 5.5: The relationship between the mean error E and the SD σ of all sets of JSN in radiographic phantom data.

The SD σ_{fg} represents a dispersion of a set of JSN$_{fg-k}$ ($k \in [1, n]$). According to our experiments when using phantom datasets, the SD σ and the mean error E has a high positive correlation, as show in Fig. 5.5. The Pearson correlation coefficient between σ and E is 0.641 (count: 220, p-value: $<$ .001). For the above reason, and the most important advantage that the SD σ not relying on the ground truth, we used it to measure the performance of our work in clinical databases. In addition, we also found that noise in radiography due to beam attenuation in tissue can greatly affects the accuracy of measurements especially in terms of SD.

5.2.2 Clinical data

549 hand PA radiographs have been analyzed in this subsection. Compared to phantom data, clinical data lay out additional challenges. The major challenge in this work is the uncertainty of hand posture, different hand postures can present differentiated bone contours.

According to our experiments, changes in bone contours can affects the accuracy of JSN progression quantification. Here, we showcase (see Fig. 5.6) majority of mismatch bone contour cases. The most frequent reason is the inconsistent angle between the upper and lower bones of joint, as show in Fig. 5.6 (a). This mainly occurs on IP and MCP joints. PIPOC has high

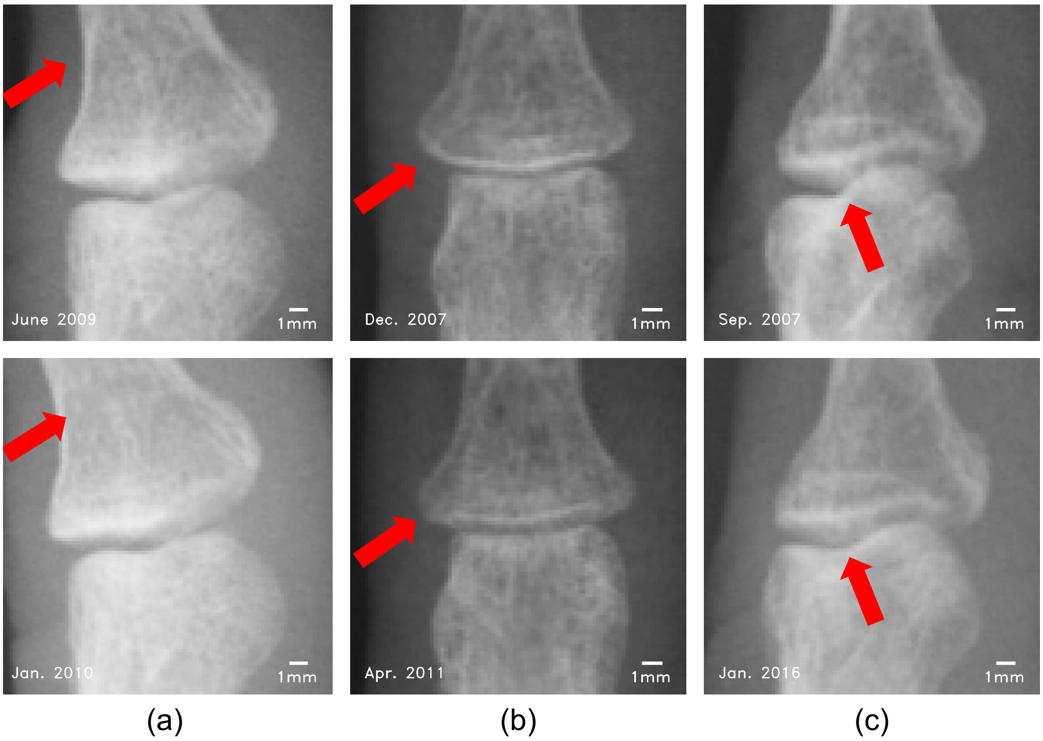


Figure 5.6: Joints with mismatched registration. (a) Inconsistent joint angle. (b) Bended finger. (c) Inconsistent projection angle.

Table 5.5: The mean SD in millimeter and the mismatching ratios for PIPOC

	Clinical Data				Phantom Data	
	IP	DIP	PIP	MCP	Air	Water
Thumb	0.093 (7.2%)	N/A	N/A	0.078 (4.5%)	-	-
Index	N/A	0.047 (4.0%)	0.065 (5.2%)	0.051 (1.6%)	-	-
Middle	N/A	0.055 (5.8%)	0.061 (3.4%)	0.057 (4.3%)	-	-
Ring	N/A	0.029 (4.1%)	0.033 (1.8%)	0.023 (1.8%)	-	-
Small	N/A	0.044 (5.9%)	0.053 (3.7%)	0.038 (2.0%)	-	-
Overall	**0.093** **(7.2%)**	**0.044** **(5.0%)**	**0.053** **(3.5%)**	**0.050** **(2.8%)**	**0.007**	**0.025**

accuracy for translation detection, but weak resistance to rotation. Another important reason of mismatched registration is the bending of the fingers, which appears on DIP and PIP joints, for an example see Fig. 5.6 (b). Finger bending can result in the changes of the far margin appearance of upper bone. Besides, inconsistent projection angle also can be the reason, see Fig. 5.6 (c). Most of the time it happens only on the IP joint, which is caused by inconsistent joint position or thumb roll. The individuated finger movements differ greatly as studied in [2]. Movements of the thumb, index finger, and little finger typically were more highly individuated than were movements of the middle or ring fingers. The angular motion tended to be greatest at the PIP joint of each digit [2]. It is worth noting that, the flexibility of joint and SD express high positive correlation (refer Table 5.5).

In summary, the hand posture should be consistent and avoid bending of the fingers, especially the thumb when using our work for JSN progression quantification. Thus, we strongly recommend that using guide lines lines to standardize hand posture in taking radiography, this simple step can greatly improve the accuracy of PIPOC .

Table 5.6: Comparison with related works. Mean error and SD in millimeter. Numbers in braces indicate the corresponding percentage of the ground truth value for the respective joint.

Mean Error		Dataset (radiographs)	Resolution (mm/pixel)	DIP	PIP	MCP	Overall
Neural Network [3]	'00	54	0.1	0.118	0.071	0.091	0.093
Active Shape Models [4]	TMI'08	160 MCP*	0.0846	-	-	0.283(16.1%)	0.283(16.1%)
Edge Detection [5]	TBME'15	104	0.1	(5.8%)	(7.2%)	(7.1%)	(6.8%)
FIPOC [1]	ISBI'19	Phantom	0.15	-	-	0.040	0.040
Manual Measurement	-	Phantom	0.15	-	-	0.056	0.056
PIPOC (Ours)	**-**	**Phantom**	**0.15**	**-**	**-**	**0.013**	**0.013**
SD							
Active Shape Models [4]	TMI'08	160 MCP*	0.0846	-	-	0.080(4.5%)	0.080(4.5%)
Edge Detection [5]	TBME'15	104	0.1	(4.8%)	(5.3%)	(4.4%)	(4.8%)
PIPOC (Ours)	**-**	**549**	**0.175**	**0.044**	**0.053**	**0.050**	**0.049**
PIPOC (Ours)	**-**	**Phantom**	**0.15**	**-**	**-**	**0.007**	**0.007**

* The dataset in [4] contains 160 MCP joint radiographs. Considering that each hand radiograph contains 5 MCP joints, this dataset can be equivalent to 32 hand radiographs.

5.2.3 Comparison with related works

Table 5.6 compares our work with previous JSW/JSN quantification works. In paper [3], they only used RMSD instead of mean error to evaluate the accuracy of their work, so we standardized the error metric accordingly. Considering that the error should conform to a Gaussian distribution, the mean error and RMSD can be transformed by Eq. 5.6.

$$\begin{aligned} E &= \int_{-\infty}^{+\infty} |x| \cdot \frac{1}{\sqrt{2\pi} \cdot \mathrm{RMSD}} e^{-\frac{x^2}{2 \cdot \mathrm{RMSD}^2}} dx \\ &= \sqrt{\frac{2}{\pi}} \cdot \mathrm{RMSD} \end{aligned} \tag{5.6}$$

In paper [5], authors only give the corresponding percentage of the error to the ground truth. Considering the JSW of MCP is around 1.70 mm [6, 7], the mean error of MCP joint in millimeter is around 0.121. It is noteworthy that, papers [3–5] used manual measurement results as ground truth. As discussed above and in Table 5.6, manual measurement has an error about 0.056 mm (low noise) / 0.104 mm (high noise) when using phantom data (spatial resolution: 0.15 mm/pixel). Although this value can decrease with higher spatial resolution, it is undeniable that in these works which employ manual measurement as the ground truth, the mean error may have a deviation.

The calculation procedure of SD in paper [4] is different from ours. They measured JSW of each joint 10 times with varying clipping of the entire radiograph. The SD quantified the uncertainty of measuring a radiograph. In our work, an intermediate radiograph is introduced in SD calculation. The JSN progression between the two radiographs and the intermediate image is calculated respectively, thus, the SD can be obtained by changing the intermediate image. When using the SD calculation method given in paper [4], we measured a lower SD (DIP joint: 0.0099 mm, PIP joint: 0.0095 mm, MCP joint: 0.0061 mm. These SDs do not include mismatched data, the mismatching ratios are shown in Table 5.5).

Compared to the method which combines FIPOC , spatial domain segmentation and image in-painting algorithm [1], this work has a lower mean error in phantom study, and faster processing speed (on our clinical dataset, PIPOC (Ours): 0.0121 sec/time, FIPOC [1]: 0.0358 sec/time). These improvements are due to the removal of the in-painting algorithm.

We can observe from Table 5.6 that even though the spatial resolution of our work is poorer than those in the related works, our mean error and the SD are significantly lower.

5.3 Joint angle correction in JSN progression quantification

In most of the cases, phase images of the same joint in different radiographic images are similar, the PIPOC method can measure JSN between two joint radiographic images with high accuracy and reliability. However, sometimes there are uncontrollable conditions during radiography, like inconsistent joint angle, finger bending, and inconsistent projection angle. For the above reasons it can lead to chaos among the phase difference spectrum of PIPOC and result in mismatching [8]. In our experiments, the JSN progression between every two images is measured to reduce the influence of mismatching. Consider n joint radiographic images from RA patient, the $\overline{\mathrm{JSN}_{fg}}$ between image f and image g can be indirectly obtained by taking the average of multiple measurements.

$$\overline{\mathrm{JSN}_{fg}} = \frac{1}{n}\sum\nolimits_{k=1}^{n}(\mathrm{JSN}_{fk} + \mathrm{JSN}_{kg}) \tag{5.7}$$

In Eq. 5.7, the JSN_{fk} is the JSN between image f and image k by using JSN progression quantification with/without joint angle correction (in § 3.3).

We take a set of MCP joint radiographic images as an example. In this set, the 3rd, 8th, 10th, and 12th joint radiographic images have significant joint rotation, as shown in the lower line of Fig. 5.7. When we quantify the JSN progression without joint angle correction, the joint rotation can greatly affecte the accuracy and result in mismatching (the orange line of line chart in Fig. 5.7). As shown by the blue line in the line chart, the mismatch caused by inconsistent joint angle can be controlled by introducing RIPOC based joint angle correction. A radiologist and a rheumatologist were invited to evaluate quantification results with or without joint angle correction. In our data with inconsistent joint angle, compared to JSN quantification results without joint angle correction, the results with joint angle correction method are closer to their qualitative judgment of the narrowing trend of joint space. In our experiments, this method can enhance the robustness of the JSN progression quantification based on PIPOC, especially for inconsistent joint angles. Joint angle correction can provide new information in the form of the joint rotation in radiographic images for diagnosis and prognosis. Since rotation is caused by subluxation due to worsening JSN, the rotation parameter can be used as one of the RA diagnostic indicators.

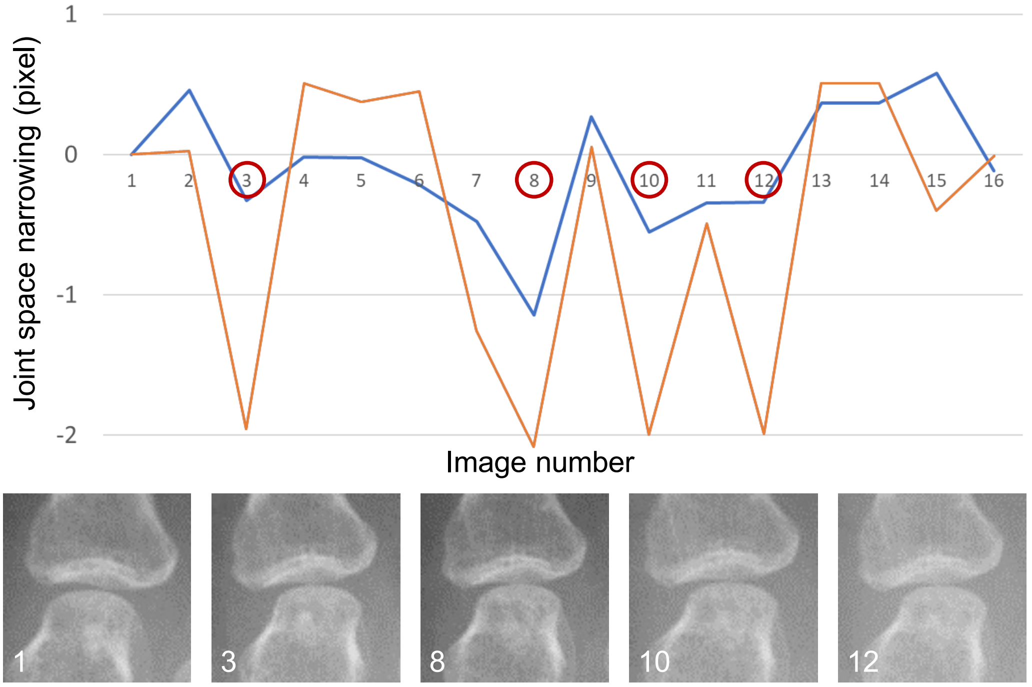


Figure 5.7: The result of JSN progression quantification. The upper line chart shows the difference between the result of JSN progression quantification with joint angle correction (blue line) and without joint angle correction (orange line). The following shows some MCP joint radiographic images and their numbers used to generate the upper line chart. These radiographic images are from a same RA patient, but these MCP joint radiographic images have different joint angles. As show in the line chart, inconsistent joint angles can cause PIPOC to mismatch when quantifying JSN without joint angle correction (orange line). The proposed joint angle correction method can effectively manage these outliers.

Bibliography

[1] Yafei Ou, Prasoon Ambalathankandy, Takeshi Shimada, Tamotsu Kamishima, and Masayuki Ikebe. Automatic radiographic quantification of joint space narrowing progression in rheumatoid arthritis using poc. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 1183–1187. IEEE, 2019.

[2] Charlotte Häger-Ross and Marc H Schieber. Quantifying the independence of human finger movements: comparisons of digits, hands, and movement frequencies. *Journal of Neuroscience*, 20(22):8542–8550, 2000.

[3] J Duryea, Y Jiang, M Zakharevich, and HK Genant. Neural network based algorithm to quantify joint space width in joints of the hand for arthritis assessment. *Medical physics*, 27(5):1185–1194, 2000.

[4] Georg Langs, Philipp Peloschek, Horst Bischof, and Franz Kainberger. Automatic quantification of joint space narrowing and erosions in rheumatoid arthritis. *IEEE transactions on medical imaging*, 28(1):151–164, 2008.

[5] Yinghe Huo, Koen L Vincken, Désirée van der Heijde, Maria JH De Hair, Floris P Lafeber, and Max A Viergever. Automatic quantification of radiographic finger joint space width of patients with early rheumatoid arthritis. *IEEE Transactions on Biomedical Engineering*, 63(10):2177–2186, 2015.

[6] Tore K Kvien, Till Uhlig, SIGRID ØDEGÅRD, and Marte S Heiberg. Epidemiological aspects of rheumatoid arthritis: the sex ratio. *Annals of the New York Academy of Sciences*, 1069(1):212–222, 2006.

[7] Alexander Pfeil, Joachim Böttcher, Bettina E Seidl, Jens-Peter Heyne, Alexander Petrovitch, Torsten Eidner, Hans-Joachim Mentzel, Gunter Wolf, Gert Hein, and Werner A Kaiser. Computer-aided joint space analysis of the metacarpal-phalangeal and proximal-interphalangeal finger joint: normative age-related and gender-specific data. *Skeletal radiology*, 36(9):853–864, 2007.

[8] Yafei Ou, Prasoon Ambalathankandy, Ryunosuke Furuya, Seiya Kawada, Tianyu Zeng, Yujie An, Tamotsu Kamishima, Kenichi Tamura, and Masayuki Ikebe. A sub-pixel accurate quantification of joint space narrowing progression in rheumatoid arthritis. *arXiv preprint arXiv:2205.09315*, 2022.

Chapter 6

Conclusion and future works

This thesis aims for computer-aided monitoring and diagnosis of RA. We proposed a JSN progression quantization framework based on image registration. Our extensive experiments demonstrate that our proposed framework promises high precision monitoring when compared to two mainstream related works; the scoring framework based on ML classification, and the JSW quantization framework based on edge detection. Our proposed framework can fill the gap in monitoring the JSN progression with high sensitivity, and offers a broad application prospect.

We have improved the FIPOC algorithm by adding a phase spectrum segmentation step, so that it can measure the displacements of multiple regions at the same time. We named the improved FIPOC algorithm as PIPOC. As an image registration algorithm, PIPOC is used in JSN progression quantification. Compared to FIPOC, PIPOC can effectively avoid the impact of the segmentation and in-painting process and reduce the phase dispersion in the phase difference spectrum. In our phantom study, PIPOC shows a much lower mean error than FIPOC.

We detect the joint positions by using two different methods; a joint classifier which is trained by haar-like feature based AdaBoost, and a key point detection network that consists of three consecutive CNNs. These networks sequentially implement the following functions: normalizing the input hand radiography, detecting rough joint positions, and correcting joint positions. In our experiments, the proposed joint location detection method has an error of less than 3 pixels in 89.13% joints. We also found that the mean error of distal interphalangeal joints and proximal interphalangeal joints are significantly lower than metacarpophalangeal joints. The mean error of joints on thumb are significantly lower than other joints.

We introduced a RIPOC based joint angle correction in the JSN progression quantification. The proposed method can manage the mismatch of

POC due to joint rotation. In this method, the rotation can be represented as translation displacement in polar coordinates, and the rotation is compensated by using displacement quantification by POC. The measurement is robust against rotation, which has been a major weakness of conventional POC algorithms.

In our phantom studies, and experiments using clinical dataset our algorithm can measure the displacements of upper and lower bones with subpixel accuracy. The measured mean error of our algorithm is in range of 11.9% - 35.0% in comparison to manual measurements using radiographic phantom datasets, in range of 6.0% - 16.1% when using tomosynthesis phantom datasets, and with a SD of 0.0519 mm when using radiographic clinical dataset. Our work greatly improves the accuracy and sensitivity of JSN progression quantification, which might help radiologists/rheumatologists to make more timely judgments on diagnosis and prognosis in RA patients.

Currently, ML is applied to difficult tasks in medical image processing [1–4]. We anticipate future studies in this direction. Our experiments in this study have shown the superiority of image registration based JSN progression quantification framework in RA monitoring compared to the current most popular classification based SvdH scoring framework, and the margin detection based JSW quantification framework. Those ML-based image registration algorithms can be used for JSN quantization in RA. To address the posture (finger movement) related constraints and inconsistent joint angle which is likely to result in mismatched registration. Given the advantages of ML, it may be possible to achieve higher robustness (lower mismatching ratios) at the expense of a small amount of accuracy [2]. Furthermore, we can quantify JSN by ML using the image features extracted by our work, this can improve the overall performance of the algorithm.

Bibliography

[1] Norah Asiri, Muhammad Hussain, Fadwa Al Adel, and Nazih Alzaidi. Deep learning based computer-aided diagnosis systems for diabetic retinopathy: A survey. *Artificial intelligence in medicine*, 99:101701, 2019.

[2] Grant Haskins, Uwe Kruger, and Pingkun Yan. Deep learning in medical image registration: a survey. *Machine Vision and Applications*, 31(1):1–18, 2020.

[3] Bradley J Erickson, Panagiotis Korfiatis, Zeynettin Akkus, and Timothy L Kline. Machine learning for medical imaging. *Radiographics*, 37(2):505–515, 2017.

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

Chapter A

Appendix: Real-time TM: a survey and cross-implementation HW benchmark

A.1 Real-time TM in medical image processing

Radiography is an imaging technique to view the internal form of an object. Medical radiography has been widely used as a diagnostic tool in modern medicine. Considering that patients' conditions are often monitored over decades of time scales, changes in the model or the parameters of the imaging equipment often lead to inconsistencies in the histogram distribution of the radiographic images and finally reduce the accuracy of the algorithm. On the other hand, radiography images typically have a wide DR, the DR of an image is the difference between the darkest and lightest tones in that image [1]. But Standard Dynamic Range (SDR) display devices are limited to 8 bits per color channel. This leads to the brightest and darkest regions being over-exposed and under-exposed and shows no contrast with the associated loss of image detail. TM has been widely introduced as an important step in image pre-processing in order to standardize images under different imaging equipment and to display HDR radiographic images on SDR monitors. As a future work, we intend to integrate our work for accelerated HW implementation that can also support HDR X-ray imaging techniques (a part of DICOM standard [2]). In this work, we explore the literature for all known HW TMOs, which have been implemented on an FPGA / GPU / ASIC plat-

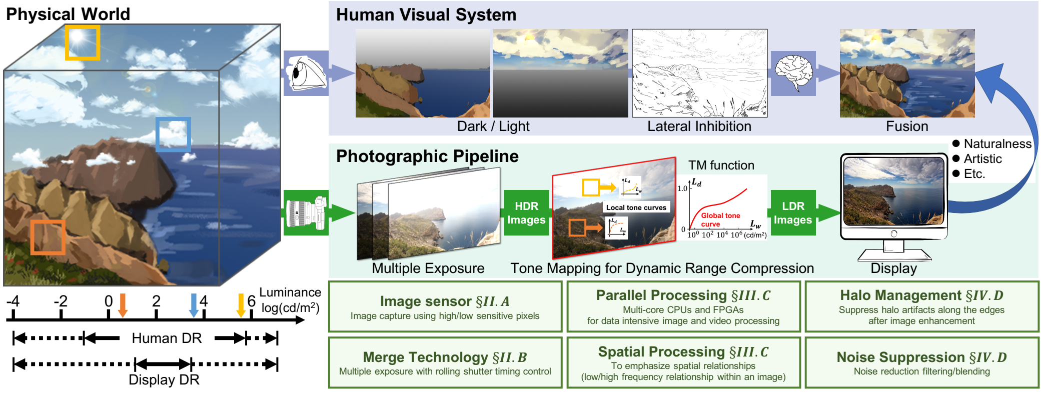

Physical World
Human Visual System
Dark / Light
Lateral Inhibition
Fusion
Photographic Pipeline
HDR Images
Local tone curves
TM function
1.0
L_d
Global tone curve
0
L_w
10^0 10^2 10^4 10^6 (cd/m²)
LDR Images
● Naturalness
● Artistic
● Etc.
Multiple Exposure
Tone Mapping for Dynamic Range Compression
Display
-4 -2 0 2 4 6
Luminance log(cd/m²)
Human DR
Display DR
Image sensor §II. A
Image capture using high/low sensitive pixels
Parallel Processing §III. C
Multi-core CPUs and FPGAs for data intensive image and video processing
Halo Management §IV. D
Suppress halo artifacts along the edges after image enhancement
Merge Technology §II. B
Multiple exposure with rolling shutter timing control
Spatial Processing §III. C
To emphasize spatial relationships (low/high frequency relationship within an image)
Noise Suppression §IV. D
Noise reduction filtering/blending

Figure A.1: This figure illustrates HVS and camera processing of HDR scenes. HVS through its non-linear and continuous adaptation can adjust across a very broad luminance range. Whereas a camera requires multiple exposure images (or HDR image sensor) to capture the wide ambient luminance levels that exists in the scenery. The HDR images produced from these multi-exposure images have higher bit width and a TMO is required to faithfully display it on a common display device. Global TM functions are good for capturing overall preview of the input image. Local TM function by considering pixel neighborhood information for each input pixel, can emphasize more local details. Additional filters are used to improve the subjective quality of TM images.

form. By carefully assessing the image quality of various TMOs we intend to study the suitability of these operators for TM medical images. In literature, various objective metrics have been used, there by making it difficult to a direct comparison between them. We have studied and indexed all of them in this work and demonstrate the link between HW cost and image quality.

The rising demand for high quality display has ensued active research in HDR imaging, which has the potential to replace the SDR imaging. This is due to HDR's features like accurate reproducibility of a scene with its entire spectrum of visible lighting and color depth. But this capability comes with expensive capture, display, storage and distribution resource requirements. Also, display of HDR images/video content on an ordinary display device with limited DR requires some form of adaptation. Many adaptation algorithms, widely known as TMOs, have been studied and proposed in the last few decades. In this paper, we present a comprehensive survey of 60 TM algorithms that have been implemented on HW for acceleration and real-time performance. In this state-of-the-art survey, we will discuss those TM algorithms which have been implemented on GPU, FPGA, and ASIC in terms of their HW specifications and performance. Output image quality is an important metric for TM algorithms. From our literature survey we found that, various objective quality metrics have been used to demonstrate the quality of those algorithms HW implementation. We have compiled those metrics used in this survey, and analyzed the relationship between HW cost, image quality and computational efficiency. Currently, ML-based algorithms have become an important tool to solve many image processing tasks, and this paper concludes with a discussion on the future research directions to realize ML-based TMOs on HW.

Superior display quality is a dominant feature that has been driving the consumer electronics industry. Unlike the past, the growing demand for definitive viewing experience is not only limited to entertainment, gaming, and media industry but has been increasingly sought for applications in security and surveillance, automotive, medical, Augmented Reality (AR)-Virtual Reality (VR), drones and robotics imaging systems. HDR imaging has come to become a compelling aspect of the new 4K/8K Ultra-high-definition format [3]. From Fig. A.1, we can notice that our eyes can see objects both in a dark night and in a sunny day, although the luminance level of a scene in sunlight is about $10^6 cd/m^2$ and one with starlight is about $10^{-3} cd/m^2$ [4]. This means that our HVS is capable of adapting to wide lighting variations within range of nearly 10 orders of magnitude. From [1], it is learnt that HVS can easily adapt up to 5 orders of magnitude within the same scene. This technique accounts for more realistically contrasted visuals by bringing out colors and detail in low-light areas so that visuals in shadow are not compressed, while not saturating the highlights. HDR imaging aims to increase the DR recorded in a digital image from a given scene. DR of an image is defined as the ratio between the darkest and the brightest points captured from a scene [1]. It can be expressed in orders of magnitude (powers of ten), in stops (powers of two) or dB. In other words, HDR can make the dark visuals deeper and the lights brighter, with more color shades with optimized contrast ratio of the display. However, this increase in amount of detail and extended color space comes at the price of higher data-width, thereby requiring more HW/Software (SW) resources to create, distribute and display HDR content. Additionally, most of the modern standard display systems have problems when dealing with visuals that may have details simultaneously both in sun, and in shadows.

We have to adapt the HDR image so that we can match the DR of HDR scene with the DR of the standard display devices. This process is widely known as TM. Depending upon the DR of the captured image TM function can expand or compress it in order to enhance the display quality [5]. The purpose of applying TM on an image can be different and depends on the target application. In some cases it may be to augment the artistic quality of the image [6], while for other applications it might be to emphasize as many details as possible, or to maximize the image contrast [7]. However, the ultimate goal of TM is to match the perception of tone mapped image with the real world perception [1]. A TMO f can be defined as a transformation function $f(I)$:

$$f(I) : L_w(x, y) \rightarrow L_d(x, y) \tag{A.1}$$

Here, TM function f maps real world luminance $L_w(x, y)$ to display luminance $L_d(x, y)$ [4], and I is an image with dimension $x \times y$. TM has been an active area of research for the last two decades, resulting in the design and development of many hundreds of different TM algorithms which can be broadly grouped in to global, local, frequency and segmentation operators [4] [8].

Strong demand for real-time embedded vision-based applications is on the rise in various domains like advanced automotive systems, medical imaging, robotics and Unmanned Aerial Vehicles. The main building blocks for such vision-based systems are the image sensors, image processing algorithms and display monitors. For real-time applications with hard time constraint a HW acceleration is indispensable [9]. Global TM which can be mathematically expressed as follows:

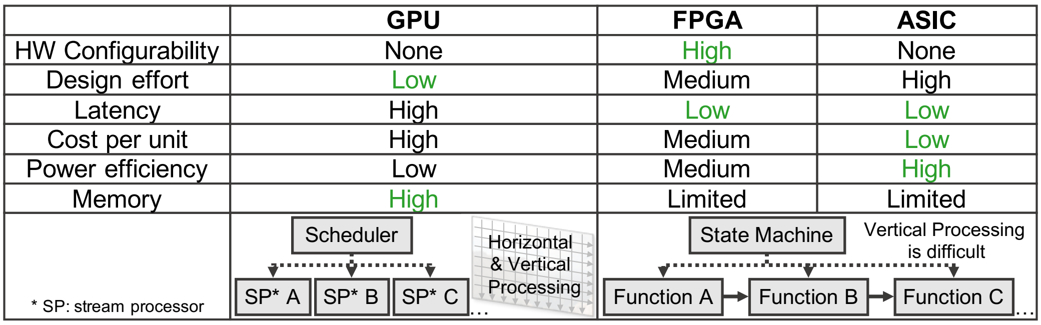
$$L_d(x, y) = TM(L_w(x, y)) \tag{A.2}$$

They are easy to implement on HW as same function maps all pixels in the image. Whereas, local TM maps using the following equation:

$$L_d(x, y) = L_w(x, y) \cdot H(x, y) \tag{A.3}$$

$$H(x, y) = L_w(x, y) \cdot KF(x, y) \tag{A.4}$$

Here, H is a scaling function and KF is the kernel filter. The implementation of local TM is complex as it demands more memory and computation when compared to global TM. Additionally, it is prone to artifacts (halo) and additional filters are required to suppress them (refer §A.4.4). Furthermore, the TM methods involving image pyramids or layer separation are very memory intensive to be realized on HW. Usually, these algorithms require multi-pass filtering which demands storing intermediate results and managing several layers. For example a naive implementation of Guided filter [10] requires 6× box filtering, storage of several image layers, and computation of weights (related to each layer) × detail layer. Therefore, implementing such TM algorithms on resource constrained devices is an arduous task. TM function have been implemented on different platforms (see Table A.2, Table



	GPU	FPGA	ASIC
HW Configurability	None	High	None
Design effort	Low	Medium	High
Latency	High	Low	Low
Cost per unit	High	Medium	Low
Power efficiency	Low	Medium	High
Memory	High	Limited	Limited

Scheduler
SP* A
SP* B
SP* C
...
Horizontal & Vertical Processing
State Machine
Vertical Processing is difficult
Function A
Function B
Function C
...
* SP: stream processor

Figure A.2: Feature comparison of GPU, FPGA, and ASIC. In this paper we survey TMOs using GPU, FPGA, and ASIC accelerated real-time systems.

A.3, and Table A.4), and the choice is determined by the target application and specification. Adams studied these technologies for real-time Signal Processing (SP) application and draws a comparison between them in [11].

Graphics accelerators were initially designed for professional graphics workstation and gaming. These devices consists of many optimized processing cores that can perform high speed matrix multiplications in parallel. The inherent parallelism in image processing algorithms make them suitable for GPU implementation. Modern memory rich GPU micro-architectures with faster clocks, powerful scheduler, higher memory bandwidths, and floating point processing capabilities make them compelling for any image processing application. Additionally, GPU programming model can support vertical and horizontal parallelism as shown in Fig. A.2 using the multiple thread block [12]. However, one of the significant drawback of GPU-based system is the slow transfer of image data between host Personal Computer (PC) and GPU, for example, 10 ms is required for a 640 × 480 image transfer [13]. Furthermore, managing various memory components like shared memory, global, local and constant memory is a non-trivial task. The ability to build optimized custom HW makes FPGA and ASIC better than GPU. FPGA in spite of their low clock frequency can achieve high performance by design optimizations. Ideally, designers strive to exploit the parallelism by utilizing the on-chip memory banks, reduce the number of operations and memory accesses. The most attractive feature of FPGA are their reprogrammability, which allows the designer to optimize the module by design iteration till a satisfactory design is achieved. However, this feature comes at the expense of slow speed when compared to ASIC. The freedom of designing at the lowest level of transistors make them deliver the most optimal solutions. However, their high entry cost, long backend design cycle make them suitable only for large volume designs. We have listed the features of these three platforms in

Fig. A.2.

Additionally, embedded applications are energy and resource constrained there by simply porting SW algorithms on a HW platform may result in poor performance or even system failure. HW implementation of image processing algorithms have to be optimized for HW porting [14], and this redesign effort, which can exploit the HW platform for optimal performance has produced many novel HW TM algorithms and architectures. Previously, in literature, there are many TM survey papers like [15–21], and these surveys only covered the SW algorithms.

In this survey, we study HW TM algorithms and to the best of our knowledge there has been no such earlier survey. Following are our main contributions:

1. A comprehensive introduction to HDR imaging (§ A.2), HW TMOs and their functional architectures (§ A.3).

2. Detailed survey of TMOs that have been implemented on an ASIC, FPGA and GPU platform (§ A.3.3).

3. Comparison of TMOs based on their HW specification and performance (§ A.4.1). Enumerate various image quality metrics used to specify HW TMOs (§ A.4.3). Demonstrate the relationship and trade-off between HW cost, image quality and computational efficiency (§ A.4.4).

4. Discussion on a future perspective for implementing ML-based TMOs on HW (§ A.5).

A.2 HDR imaging

HDR images can be captured from real world scenes, rendered on computers by various computer graphics tools. In this chapter, we will focus mainly on methods of obtaining HDR images by using conventional cameras and special HDR sensors, which are useful for building real-time systems. For the computer graphics methods, there are well known books describing those methods [22, 23]. The gaming industry has been employing HDR rendering for very long time, they were used for rendering special visual effects like dazzling, slow dark-adaptation there by enhancing the immersive impression [24]. Today HDR imaging is used in many applications to enhance functionality of cinematography and photography [25], biomedical imaging (see DICOM standard [2]) [26], remote sensing [27] and many more computer vision applications [28].

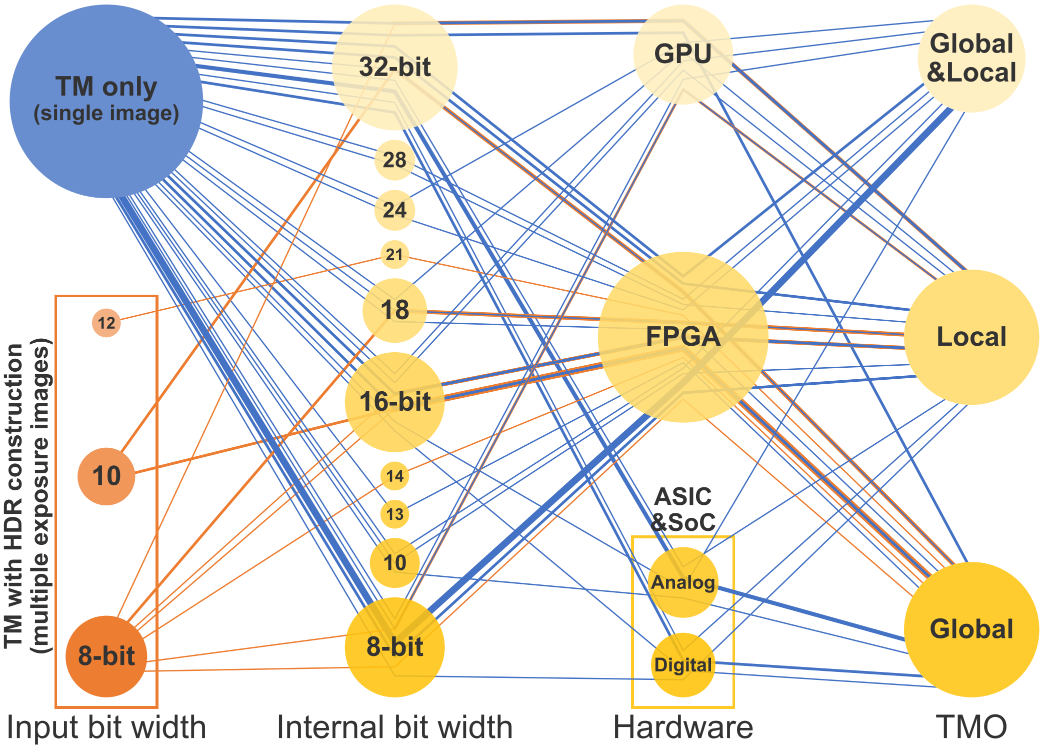


Figure A.3: This diagram presents an overview of HW TMOs. We can observe that majority of the work use single image HDR. Also, from this figure, we can observe that 32/16/8-bit widths are frequently used for internal computations, and FPGAs are the preferred platform for TMO acceleration.

Figure A.3 shows the frequency of various bit width used in all HW TM papers. Moreover, the plot describes how different researches implemented different TMO for HDR images produced with distinct bit width on different HW. The choice of bit width will directly impact on the image quality, HW cost and power consumption [29]. Most researchers choose 16-bit or 32-bit luminance images for TM, and only a few works report other bit width. Many works proposed TMO on low-power embedded platforms, which are often implemented using HDR images with fixed-point arithmetic. Compared to floating-point arithmetic, fixed-point arithmetic has some advantages on embedded platforms such as low-power consumption, the small circuit size and high-speed computing [30–32]. On the other hand, floating-point arithmetic can save a huge range of luminance with small bit width.

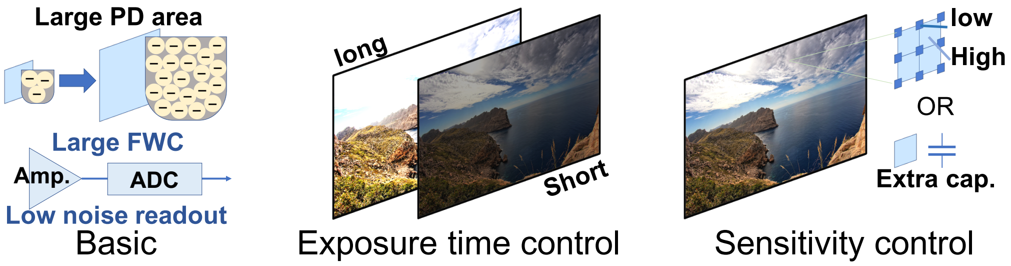


Figure A.4: HDR image sensing methodologies (image courtesy [33]).

A.2.1 HDR image sensor

As shown in Fig. A.4 there are broadly three different architectures that are used to design HDR image sensors. Noise reduction and full well capacity i.e., limitation of electrons a pixel can store, see Fig. A.4(a) are important parameters in Peak Signal to Noise Ratio (PSNR) and it determines the dynamic range. Variable gain amplifiers and dual ISO [34, 35] are one of the effective methods in reducing noise by amplifying the signal in low-light conditions. Multi-sampling in the readout circuit [36] is also an effective method for noise suppression. The other Obtaining a large full well capacity by designing a large pixel area in a single sensor leads to an increase in the dynamic range. However, large size pixels are very expensive and are mainly used in sophisticated camera sensors for cinematic cameras like ARRI®, RED® [37, 38]. Therefore, various efforts have been made to improve higher dynamic range (DR).

Generally, the Complementary Metal–Oxide–Semiconductor (CMOS) sensor use rolling shutter mechanism, and it is easy to control pixel resets for short/long exposures by timing control between lines as shown in see Fig. A.4(b). However, this can cause image shifts when composing images for video. Recently, HDR technique with a constant exposure time are being reconsidered, for example, by preparing two pixels with high/low sensitivities as shown in see Fig. A.4(c) [39–42] or by storing saturated electrons at high illumination in overflow capacitors [43]. These technologies have been developed in the past; however, they are currently being implemented for automotive applications. Since, there is no need for short exposures, LED flicker in the signal (high-speed blinking) can be removed by averaging within its constant exposure time [44]. Logarithmic compression pixels are also being developed, but they are not robust to noise because, their analog signal output range is not expanded [45, 46]. As a result, its application is limited to special low-power/cost post processing sensors such as edge and motion extraction [47]. Image synthesis using multiple-sensors could be considered

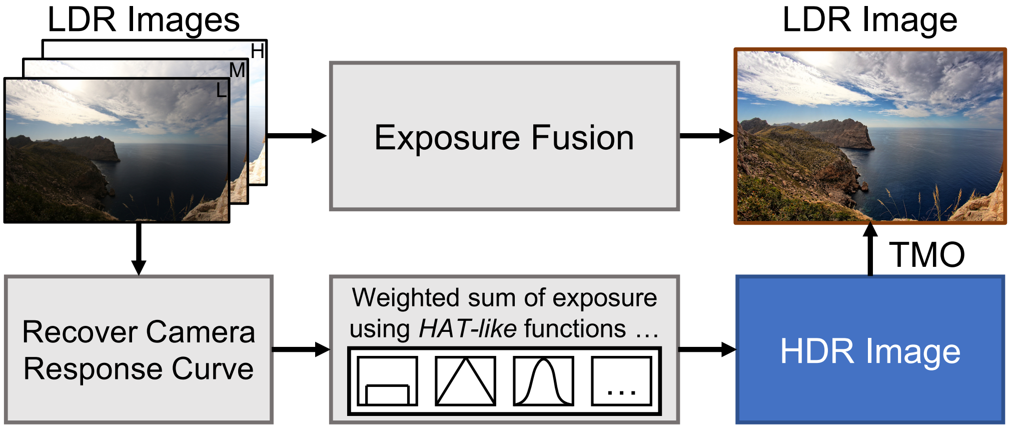

LDR Images
H
M
L
Exposure Fusion
LDR Image
TMO
Recover Camera Response Curve
Weighted sum of exposure using *HAT-like* functions …
…
HDR Image

Figure A.5: Using a camera response curve the full DR of the scene is captured from a set of LDR images with different exposure times. Algorithms like Bachoo et al's [49] and Popadic et al's [50] can directly generate HDR-like images from bracketed images.

for HDR, but it is actually intended to obtain depth from parallax and angle of view information as it has two or more optical systems [48].

A.2.2 HDR merge

HDR images can also be composed by combining multiple LDR images with different exposure time into a single HDR. Here, the exposure time which is also known as the shutter speed is the duration of time when the digital sensor inside the camera is permitted to capture light. The amount of light that reaches the film or image sensor is directly proportional to the exposure time. Therefore, a long exposure time image will have an overall greater luminance. It will detect smaller amount of light sources even in darker areas. But the picture might saturate in bright parts of the scene due to a too much of light for the sensor. On the other hand, a short exposure image will record bright parts of the scene but would not been able to register darker light sources. Exposure time values are often referred to as "stops". A stop implies doubling the exposure time (relative to a reference time). +1 stop is doubling, +2 stops is times 4, and -1 is halving the exposure time [51, 52]. There are different techniques to compose an HDR image, one by combining multiple images of varying exposures [53–55] and another by using multi-sensor camera [56].

Table A.1: TM algorithms suitable for HW porting

TM Function	TM Algorithm	GPU*†	FPGA*	Analog*†	Digital*	Definition	Parameters
Exp. based	Reinhard [57]	[58], [59] [60], [60]	[61], [62], [63] [64]	[65], [66] [67], [68]	[62], [69] [70]	Global:$L_d(x,y) = \frac{L(x,y)}{1+L(x,y)}$ Local: $L_d(x,y) = \frac{L(x,y)}{1+V(x,y,s_m)}$	τ: Image key§ b: Bias parameter $L_a(x,y)$: Local adaptation luminance $L_d(x,y)$: Display luminance $L_w(x,y)$: World luminance L_{max}: Maximum world luminance L_{min}: Minimum world luminance $L(x,y)$: Scaled pixel luminance MAX: Maximum display luminance TM: TM function $V(x,y,s_m)$: Local area luminance
	Others	[71]	[72], [73], [74] [75], [76], [77] [78], [79], [80]	[81], [82] [83]	[76]		
Logarithm	Duan [84]	[85]	[86], [87], [88] [89]			$L_d(x,y) = C * MAX$ $C = \frac{ln(L_w(x,y)+\tau)-ln(L_{min}+\tau)}{ln(L_{max}+\tau)-ln(L_{min}+\tau)}$	
	Drago [90]	[60]	[91], [92]	[65], [82]		$L_d(x,y) = \frac{log_t(L_w(x,y)+1)}{log_{10}(L_{max}+1)}$ $t(b) = 2 + 8(\frac{L_w(x,y)}{L_{max}})^{\frac{lnb}{ln0.5}}$	
	Ashikhmin [93]	[94], [60] [60]				$L_d(x,y) = \frac{L_w(x,y)TM(L_a(x,y)}{L_a(x,y))}$ $TM(L_a(x,y)) = \frac{MAX*(L_a(x,y)-L_{min})}{L_{max}-L_{min}}$	
	Others	[60]	[61], [95], [96]	[65]			
Hist. based	HE-like method‡	[97], [98] [6], [99] [100], [13]	[13], [101], [102] [103], [104], [96]	[105]	[106]	Construct uniformly distributed image histogram	
Edge preserving	Durand [107]	[60]	[108], [109]			Multi-scale decomposition and compression	
	Others	[6], [98] [13]	[110], [79], [13] [101], [111], [112]				
Gradient domain	Fattal [113]		[114], [115]		[116]	Gradient domain compression	

*Different font colors represent different TMOs (🟦Global 🟩Local 🟧Global and Local). †Papers [60, 65] and [82] implemented multiple TM algorithms on HW respectively. ‡Histogram equalization-like algorithms inspired from [117]. §Image key value τ refers to brightness of an image. The low key image will be dark, and high key image will be bright (p. 126 [4]).

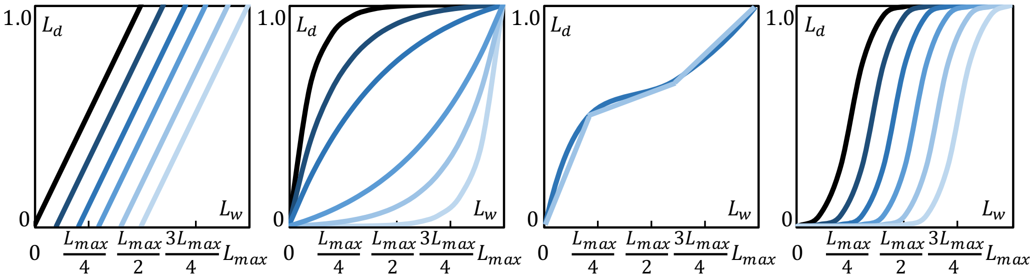


Figure A.6: TM functions (a) Simple linear mean value mapping (b) Linear gamma function (c) Piecewise linear function (d) Sigmoid function for histogram adjustments.

For an LDR image, its DR is bounded by the range of sensor, i.e. an 8-bit camera can only capture ratios up to 255:1. However, we can achieve a greater DR using the same camera by combining multiple images which have been captured with different exposure time. Each of these LDR images will cover a different range of the luminance in the scene. This allows to have a greater resolution on the luminance captured . Images with a short exposure time will be adapted for capturing very bright parts of the scene but will fail to capture darker parts. Long exposure time images being the opposite, they will saturate in bright parts of the image. All intermediate images captured with various exposure time will help cover the whole range of luminance. This strategy is demonstrated in Fig. A.5. The final composite image will have a greater DR than it is achievable with a single shot by the camera.

A.3 TM algorithms and their HW implementations

In the previous section we discussed how to obtain/produce HDR content, and in Fig. A.1 we presented the overall TM pipeline. Generally, the HDR content requires to be stored in a medium with a greater amount of bits/pixel than that of a single LDR image. Although, HDR display systems do exist [118], and TVs with an extended DR are currently available in the commercial market, but they are not as widespread due to their limitations in terms of its cost, DR and color gamut. The process of TM consists of reducing the DR of the HDR image into an image that can be easily displayed on wide range of display devices which have limited DR and color gamut. Usually the TM is performed on the luminance image because of the lower computational and memory requirements, when compared to TM on the R, G, B channels.

A.3.1 TM algorithms

The development of HDR TMOs has been an active field of research in HDR retargeting for the last three decades. Over the years many solutions have been proposed as is evident from the many publications [1,7,25,119,120]. The main issue with TM is the very nature of HVS which is highly adaptive and complex. Simple TMOs are based on operations such as, scaling, clipping or a gamma correction, and Fig. A.6 presents a brief review of these methods. As stated earlier, global methods, use same function over the whole image. It is very likely that such methods may fail to reproduce best visual quality if there are quick luminance changes in the image. Hence, sophisticated local methods are required to generate images with more details. These methods use the surrounding pixels to determine the appropriate intensity. When applying global only TM, may result in suppressing details in dark and bright areas and a local operator is required to restore those details. Please note the notations we will use to describe various TMOs are given in table A.1.

The simple linear mapping shown in Fig. A.6(a) is easy to implement as it maps the mean value of the image to a key value in the interval $[0, 1]$. Another linear mapping technique illustrated in Fig. A.6(b) utilizes the gamma function used for displays. It applies a linear mapping between 0 and the maximum luminance and raise the result to an exponent determined from the image key value as shown in Eq. A.5. These functions can be easily approximated using piece wise linear segments as shown in Fig. A.6(c).

$$L_d = L_w^{\frac{log\tau}{log(L_{avg}/L_{max})}} / L_{max} \tag{A.5}$$

Response of HVS is perceived to be logarithmic, therefore designing logarithmic TM curves are highly desirable. One such function is

$$L_d = \frac{log(\tau L_w + 1)}{log(\tau L_{max} + 1)} \tag{A.6}$$

which can expand low intensity signals while compressing high intensity ones. Many researchers have advanced and developed this method like Drago *et al.* [90], and furthermore, few HW implementations have also been realized as listed in table A.1.

Figure A.7 shows a general pipeline which is useful for implementing a global TM function. As a first step the TM pipeline obtains the luminance image and from that it calculates global statistics (like $L_{max}, L_{min}, L_{avg}$). In some algorithms these statistics are also calculated from previous frame based on a assumption that there is very little change between successive

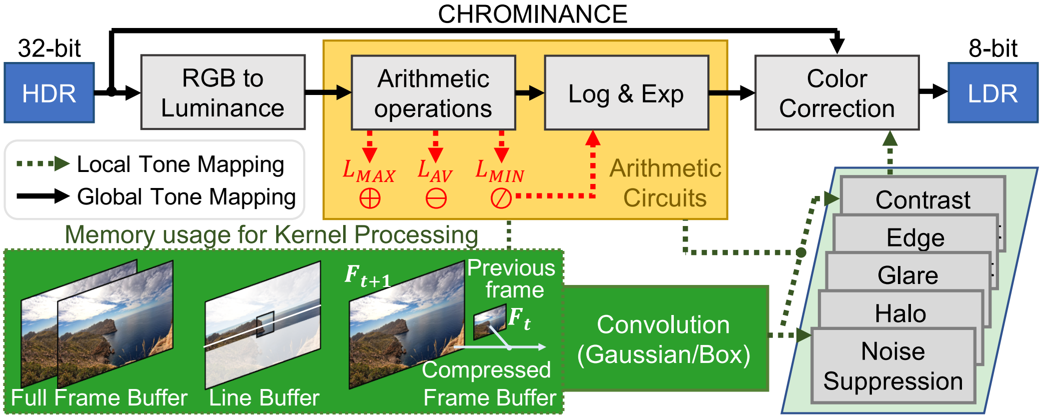


Figure A.7: General block diagram for HW global and local TM systems. For local calculations in local TM kernel processing a full frame buffer/line buffer or compressed frame buffer is required.

frames when imaging at 30/60 frames per second [86, 87, 91]. In the pipeline next step is to realize a logarithmic or exponential like function to compute the tone mapped image. The HW implementation of these functions are a challenge and a common approach is to approximate such functions, by maintaining an acceptable level of error in the realized implementation. As these functions are regularly used in many HW TMOs, a detailed discussion is presented in § A.3.4. The final step in the pipeline after TM is to restore the color for displaying the output image.

Previously discussed global TM functions use a single curve to compress the HDR image, and this may be acceptable for images with medium DR. For scenes with rapidly changing brightness such a simple compression curve is inadequate. Local TM functions improve the quality of output images by preserving local contrast along with global contrast. Local TM algorithms are computationally more expensive and time consuming compared to global TM algorithms [69]. We illustrate its operation using the block diagram shown in Fig. A.7. As was in the case of global TM, we initially obtain the luminance values for the input image. The high computation cost for local TMO is due to the local information calculation for which a full frame or a few lines of the input image has to be buffered as shown in the Fig. A.7. Some algorithms have also implemented compressed frame buffer (down-sampled images) to reduce the memory cost [104, 110]. To meet the real-time constraints, as a common approach previous frame is used to compute the local information for current frame. The color correction step is described in detail in § A.3.2.

One of the major drawbacks of local TM algorithms are the creation of halo artifact among the high contrast edges and the graying out of the low-contrast areas [121, 122]. Durand and Dorsey in the year 2002 demonstrated an interesting application of the bilateral edge-preserving filter as a TMO to compress the DR in HDR scenes [107]. They were inspired from an earlier work of Tumblin and Turk who proposed a low-curvature image simplifier based on anisotropic diffusion to decompose an image into base (low frequency) and detail (high frequency) layer [123]. Following this strategy many edge-preserving filter based TMOs have been proposed like, trilateral [124], weighted least square [125], local laplacian [126], etc. These edge-preserving filter based TMOs are effective in suppressing halo artifacts, however, obtaining a satisfactory result is a non-trivial task as one has to determine the gain for each layer and weights of the filter. Furthermore, additional filters to suppress halo and noise artifacts have been proposed. But, such filtering will require that the input image (of size $M \times N$) be convolved with a filter (of size $k \times r$). Benedetti *et al.* demonstrated a simple HW sliding window convolution block, which can output one pixel every clock [127]. The latency associated with this sliding window method is calculated as:

$$T = BufferDepth \times \left\lfloor \frac{KernelSize}{2} \right\rfloor + \left\lceil \frac{KernelSize}{2} \right\rceil . \qquad \text{(A.7)}$$

A.3.2 Color conversion

TM algorithms are methods for mapping a real-world luminance range to the luminance range of the display devices, and this transformation can cause changes in color appearance of the tone mapped images. In literature, there are some well developed methods which propose color models for such TM methods which have been validated by psychophysical experiments [128–130]. Kim *et al.* proposed a color correction formula after conducting experiments that cover most of the DR covered by human visual system [130]. Meylan *et al.* proposed a retinal local adaption TM algorithm which is applied on the color filter array (CFA) images [131]. They obtain the luminance information with full resolution from the Fourier spectrum of the Bayer image [132]. However, this can introduce spatial artifacts in demosaiced images, therefore, some additional preprocessing techniques are required for generating good output images [132].

Luminance channel is preferably used for HW TM adjustment (lower computational and memory cost), and as shown in Fig. A.7 is the first operation in the TM pipeline. Color conversion is not unique, and there are many well defined methods to obtain the luminance values from the color image. An easy method to obtain luminance is to compute it as a linear combination of the red, green, and blue component according to the RGB-to-XYZ conversion scheme. Here,

$$Y = 0.2126R + 0.7152G + 0.0722B \tag{A.8}$$

Y is the luminance for a given RGB image. Another effortless procedure is to use CIELab or YUV color spaces, and one could directly obtain luminance channel as the grayscale version of the color image as they consider the luminance and color channel to be independent. In TM pipeline, for a given HDR image luminance value is calculated and the chrominance values are buffered/stored as they are required later for restoring the color post TM. Different studies have used different luminance methods, and we have listed them in table A.2, table A.3, and table A.4. Some studies have used monochrome images [61, 68, 70, 86, 102], this approach can have certain advantages in terms of reduced memory, and fewer calculations.

After TM, a common approach to restore the color is based on Schlick's color ratios [133]:

$$C_{out} = (\frac{C_{in}}{L_{in}})^{\gamma} L_{out} \tag{A.9}$$

In Eq. A.9, C_{in} represent the original RGB image, L_{in} is the luminance value obtained from the color image (like Y as shown above). If L_{out} is the tone mapped luminance value then, we can compute three output chrominance values as in Eq. A.9, where γ is a color saturation factor for displaying color images, and its value is usually set between 0.4 and 0.6 [113].

A.3.3 HW TM algorithms

As described in the introduction, it is a challenging task to accelerate complex TM algorithms. A direct HW porting is very likely to result in an inefficient implementation. Usually, HW porting requires significant re-design effort to realize an efficient HW-based acceleration for the following reasons. Ideally, the main objective of any design HW specification is to keep cost, speed and power to the minimum. Additionally, the design methodology adopted for any HW development also depends on the application and time to market. In § A.1 we introduced the HW platforms like ASIC, FPGA and GPUs for TM implementations. Each has its own advantages and disadvantages, which we presented briefly in Fig. A.2, from which we could notice that the choice of platform depends on various factors like: flexibility, design time and cost. A full custom ASIC design development will be very expensive due to increased manufacturing and design time, and increased non-recurring engineering costs. Even though the ASIC design solution can be very efficient in terms of area and performance, it is only viable for proven designs that can be mass produced. GPUs and FPGA platforms have been preferred for many image processing applications and we will discuss more about them in this section. Recently, FPGA vendors are also supporting openCL development processes, and openCL-based implementations are suitable for general-purpose computing on GPU, and it has flavor similar to the proprietary CUDA language from NVIDIA.

Table A.2, table A.3, and table A.2 list all the surveyed TMO HW implementations, and their underlying SW algorithms which were introduced in table A.1. In this section we will discuss HW TM algorithms in detail. At the outset, one of the important feature of these algorithms is that they can be realized on HW using non-complex circuitry with straightforward pipelines like the one shown in Fig. A.9. Over two decades of research has resulted in many HW accelerated TM algorithms, which we have grouped according to the implementation HW. Tables A.5, A.6, and A.7 provides more detailed performance metrics of the algorithms introduced in table A.2, table A.3, and table A.4. We have analyzed the underlying trends in these researches and have comprehensively illustrated them in Fig. A.8.

Initially, FPGAs were adopted as HW platform for accelerating TM algorithms as we can see from Fig. A.8(a) and table A.5 which lists all the FPGA implementations. They have been handy in realizing histogram-based algorithms which could TM images with 4K resolutions. Popularity of the reprogrammable FPGAs for realizing TM algorithm has been growing and researchers have implemented different types of TM functions best suited for their target applications. Low-cost, low power and reprogrammability has been the USP of FPGA devices, and table A.6 summarizes the performance metrics of all the FPGA TMO implementations for comparison. The continuous adoption of newer technology by FPGA vendors has resulted in their low power consumption, and recent TM HW implementations have utilized 28 nm node size FPGAs as shown in Fig. A.8(b). Recently, along with FPGAs, ASIC, Application-Specific Instruction set Processor (ASIP), System on Chip (SoC), and Vision System on Chip (VSoC) have been used by researchers to develop full imaging pipeline that included the camera and display drivers as shown in table A.7 and Fig. A.8(f).

Table A.2: Survey of TM GPU implementations

				TM				Image			
	Previous Work	Year	HW Platform	TM Operator	TM Algorithm	Kernel Size	Previous Frame	Channel	Luminance	Bit Width	HDR Merge
Goodnight [58]		2005	GPU	Global&Local	Reinhard [57]	49×49^8	-	Color	-	24	✘
Krawczyk [59]		2005	GPU	Local	Reinhard [57]	61×61^8	-	Color	XYZ	8 to 32	[134]
Roch [94]		2007	GPU	Local	Ashikhmin [93]	5×5^2	-	Color	-	32	✘
Zhao [60]		2008	GPU	-	Multiple‡	-	-	Color	-	32	✘
Bachoo [49]		2009	GPU	Local	-	128×128	-	Monochrome	-	8 to 8	[135]
Tian [85]		2012	GPU	Local	Duan [84]	64×64	-	Color	-	18	✘
Akil [97]		2012	GPU	Global	[136]	N/A	-	Color	XYZ/YUV	32	✘
Ureña [13]		2012	FPGA/GPU	Global&Local	-	7×7	✔	Color	HSV	8	✘
Ureña [98]	[13]	2013	GPU	Global&Local	-	11×11	-	Color	HSV	16	✘
Eilertsen [6]		2015	GPU	Local	-	230×230	-	Color	-	16	✘
Khan [99]		2017	GPU	Global	[117]	N/A	-	Color	0.265R+0.670G+0.064B	32	✘
Liao [100]		2017	GPU	Local	-	32×32	-	Monochrome	-	32	✘
Tsai [71]	[77]	2019	GPU	Local	[137]	7×7	-	Color	HSV	8	✘

*Local luminance calculation kernel size. $(i \times i^k)$:k scales Gaussian pyramid with a maximum size of $i \times i$.

†$(k$ to $i)$:k-bit images into i-bit image by HDR merge. $(k.i)$:k bits integer and i bits fraction. (k^i):k bits mantissa and i bits exponent.

‡Paper [60] compared multiple TM algorithms ([57, 90, 93, 107, 138–140] and logarithmic TM function) on GPU.

Table A.3: Survey of TM FPGA implementations

				TM				**Image**			
	Previous Work	Year	HW Platform	TM Operator	TM Algorithm	Kernel Size	Previous Frame	Channel	Luminance	Bit Width	HDR Merge
Balercia [72]		2006	FPGA	Global&Local	[141–143]	-	-	Color	YCbCr	8	✘
Hassan [61]		2007	FPGA	Global&Local	Reinhard [57]	$64{\times}64^9$	✔	Monochrome	0.27R+0.67G+0.06B	28	✘
Marsi [110]		2007	FPGA	Local	[141]	5×5	✔	Color	HSV	10	✘
Iakovidou [73]		2008	FPGA	Local	[144]	$61{\times}61^3$	✔	Color	YCbCr	8	✘
Vakili [62]		2011	FPGA&ASIP	Global	Reinhard [57]	N/A	-	Color	0.265R+0.670G+0.064B	10.22	✘
Lapray [87]	[86]	2012	FPGA	Global	Duan [84]	N/A	-	Monochrome	-	10 to 16	[55]
Kiser [63]		2012	FPGA	Global	Reinhard [57]	N/A	-	Color	0.27R+0.67G+0.06B	18	✘
Ureña [13]		2012	FPGA/GPU	Global&Local	-	7×7	✔	Color	HSV	8	✘
Mann [74]		2012	FPGA	Global	-	N/A	-	Color	-	8 to 14	[145, 146]
Lapray [64]	[87]	2013	FPGA	Global	Reinhard [57]	N/A	-	Color	-	10 to 32	[55]
Ofili [75]		2013	FPGA	Global&Local	[147]	3×3	✔	Color	-	20.12	✘
Vytla [114]	[61]	2013	FPGA	Local	[113]	$32{\times}32^4$	-	Color	0.299R+0.587G+0.114B	32	✘
Cañada [101]		2013	FPGA	Global&Local	[148]	7×7	-	Color	HSI	8	✘
Popovic [91]		2014	FPGA	Global	Drago [90]	N/A	✔	Color	YUV	16	✘
Shiau [76]		2014	FPGA/SoC	Global&Local	-	3×3	-	Color	HSV	8	✘
Li [77]		2015	FPGA	Local	[149]	$32{\times}32^3$	-	Color	HSV	8	✘
Lapray [88]	[64]	2016	FPGA	Global	Duan [84]	N/A	-	Monochrome	-	10 to 32	[55]
Ambalathankandy [78]		2016	FPGA	Global&Local	[150, 151]	5×5	✔	Color	0.299R+0.587G+0.114B	20.12	✘
Shahnovich [152]		2016	FPGA	Global&Local	-	5×5	✔	Color	-	16^8	✘
Liu [115]	[114]	2016	FPGA	Local	[113]	3×3	-	Monochrome	-	28	✘
Popovic [92]	[91]	2016	FPGA	Global	Drago [90]	N/A	✔	Color	YUV	8 to 16	[153]
Li [102]		2016	FPGA	Global	-	N/A	-	Monochrome	-	16	✘
Lee [89]		2016	FPGA	Global	Duan [84]	N/A	✔	Color	-	12 to 21	[55]
Nosko [108]		2017	FPGA	Local	Durand [107]	9×9	✔	Color	0.299R+0.587G+0.114B	8 to 16	[55]
Zemvc'ik [79]		2017	FPGA	Local	-	11×11	-	Color	YCbCr	8 to 18	[120]
Popadi'c [50]		2017	FPGA	Global	-	N/A	-	Color	-	8 to 8	✔
Nosko [109]	[108]	2018	FPGA	Local	Durand [107]	9×9	✔	Color	0.299R+0.587G+0.114B	10.8	[55, 154]
Yang [95]		2018	FPGA	Local	-	32×32	-	Color	-	20.12	No
Yang [103]	[152]	2019	FPGA	Global	-	N/A	-	Color	-	10^3	✘
Ambalathankandy [104]	[78]	2019	FPGA	Global&Local	[155–157]	31×31	✔	Color	$\sqrt{(R^2+G^2+B^2)/3}$	10	✘
Park [80]		2019	FPGA	Local	[158]	29×29	✔	Color	HSV	5.13	✘
Xiang [111]	[112]	2020	FPGA	Local	[159]	$15{\times}15^2$	✔	Color	-	16	✘
Palacios [96]		2020	FPGA	Global&Local	[160]	$25{\times}25^3$	✔	Color	(R+G+B)/3	8	✘

Table A.4: Survey of TM HW implementations

			TM				Image			
Previous Work	Year	HW Platform	TM Operator	TM Algorithm	Kernel Size	Previous Frame	Channel	Luminance	Bit Width	HDR Merge
Wang [69]	2007	ASIC	Global	[57]	N/A	✔	Color	0.2654R +0.6704G +0.0642B	16.16	✘
Chiu [116] [69]	2010	SoC	Local	[113]	8×8	✔	Color	0.2654R +0.6704G +0.0642B	16.16	✘
Punchihewa [106]	2011	SoC	Global	-	N/A	-	Monochrome	-	16	[161–163]
Vakili [62]	2011	FPGA&ASIP	Global	Reinhard [57]	N/A	-	Color	0.265R+0.670G+0.064B	10.22	✘
Sicard [81]	2013	Analog	Local	[131]	5×5	-	Monochrome	-	32	✘
Vargas [70]	2014	VSoC	Global	Reinhard [1]	N/A	✔	Monochrome	-	32	[134]
Gouveia [65]	2014	Analog	-	[90], [57]	-	-	Monochrome	-	16	✘
Mughal [82]	2014	Analog	-	[4, 90, 133]	-	-	Monochrome	-	10	✘
Shiau [76]	2014	FPGA/SoC	Global&Local	-	3×3	-	Color	HSV	8	✘
Mughal [66] [82]	2015	Analog	Global	Reinhard [57]	N/A	-	Monochrome	-	10	✘
Fern'andez [105]	2015	Analog	Global	-	N/A	-	Monochrome	-	32	✘
Shi [68] [67]	2016	Analog	Global	Reinhard [57]	N/A	-	Monochrome	-	32	✘
Chen [83]	2016	Analog	Global&Local	[147]	3×3	-	Color	YUV	16	✘
Guicquero [164]	2016	Analog	Global	-	N/A	-	Monochrome	-	32	[165]

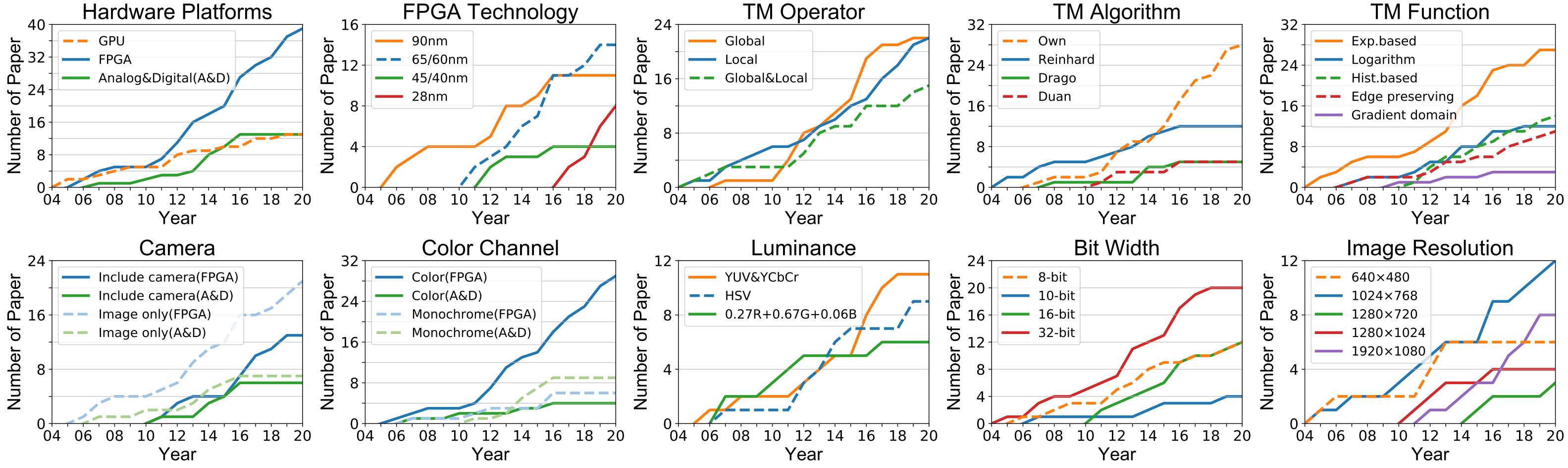


Figure A.8: From the cumulative number of papers for different years we could observe the following research trends (from our table A.2, table A.3, and table A.4) (a) HW platforms for accelerating TM algorithms, (b) FPGA technology node size, (c) Global or local TMOs, (d) Popular TM algorithms, (e) Camera in pipeline, (f) Color or monochrome processing, (g) Luminance channel, (h) HDR input bit width and (i) TM output resolution.

Table A.5: GPU implementations

	FPGA		**Performance**		
	Model	Technology (nm)	Frame Size (pixel)	Speed (FPS)	Throughput (Mpix/s)
[58]	Radeon 9800 Pro	150	512×512	30	7.9
[59]	GeForce 6800GT	130	1024×768	10	7.9
[94]	GeForce Go 6800	130	2048×2048	7	29.4
[60]	GeForce 8800 GTS	90	-	-	-
[49]	Nvidia Quadro NVS 140M	80	1600×1200	20	38.4
[13]	GeForce 7900 GTX	90	640×480	30	9.2
[97]	GeForce 8800 GTX	90	1002×666	37	24.7
[85]	GeForce GT 550M×2	40	1024×768	2.8	2.2
[98]	NVIDIA ION2	40	640×480	27	8.3
[6]	GeForce GTX 980	28	1980×1080	46.5	99.4
[99]	GeForce Titan Black	28	2048×1536	24	75.5
[100]	GeForce GT 750M	28	1280×960	3.6	4.5
[71]	GeForce GTX 650 Ti	28	4096×4096	7.5	125.8

Table A.6: Normalized cost and performance measure of TM FPGA implementations

		FPGA					**FPGA Cost**								**Performance**			
	Camera	Model	Technology (nm)	Clock (MHz)	Latency (clock)	Power (mW)	Memory (bit)	Block RAM	Logic Element	LUT	DSP	Register	Flip Flop	Others*	Frame Size (pixel)	Speed (FPS)	Throughput (Mpix/s)	pix/clock
[72]	✘	Virtex II Pro	90	80.736	-	-	-	-	-	307	-	-	137	Slice=223	640×480	30	9.2	0.11
	✘	Spartan 3	90	56.182	-	-	-	-	-	306	-	-	137	Slice=231	532×288	24	3.7	0.07
[61]	✘	Stratix II	90	77.15	64	-	3,153,408		34,806	-	54	-	-	-	1024×768	60	47.2	0.61
[110]	✘	Virtex II	150	1.3	-	-	-	20	17,280	-	-	-	1,362	Multiplier=16	125×86	24	0.3	0.20
[73]	✘	Stratix II GX	90	66.66	2×Frame+300	-	2,609,151	-	-	49,763	-	43,793	-	-	2.5Mpixel	25	50	0.75
[62]	✘	Virtex 5	60	85	3,293,244(total)	-	3,932,160	-	-	5,806	18	-	-	-	256×192	25.2	1.24	0.01
[87]	EV76C560	Virtex 5	60	94.733	65×Row	-	-	40	-	14,168	4	8,132	-	-	1280×1024	30	39.3	0.41
[63]	✘	Spartan 6	45	75	12288@163μs	-	-	3	-	874	17	-	-	Slice=1,000	1920×1080	30	62.2	0.83
[13]	✘	Spartan 3	90	40.25	-	900	-	39	-	30,086	26	-	-	Slice=16,545	640×480	60	18.4	0.46
[74]	EyeTap	Spartan 6	45	78.125	-	-	-	100	-	-	-	-	-	-	-	30	-	-
[64]	EV76C560	Virtex 6	40	125	136@1.2μs	6	-	17	-	16,880	-	20,192	-	-	1280×1024	60	78.6	0.63
[75]	✘	Stratix II	90	114.9	Row+12+7	-	68,046	-	-	8,546	60	10,442	-	-	1024×768	126	99	0.86
	✘	Cyclone III	65	116.5	Row+12+7	250	86,046	-	-	12,154	36	10,518	-	-	1024×768	126	99	0.86
[114]	✘	Stratix II	90	114.18	-	-	307,200	-	9,019	-	88	6,586	-	-	1Mpixel	100	100	0.88
[101]	✘	Spartan 3	90	40	-	897	720,000	-	-	29,007	26	11594	-	Slice=16,554	640×480	60	18.4	0.46
[91]	✘	Virtex 5	65	214.27	-	-	-	8	-	4,536	30	5,036	-	-	1920×1080	60	124.4	0.58
[76]	✘	Cyclone IV	60	55.55	2×Row+18	-	-	-	1,784	-	-	-	510	Linebuff=4	2560×2048	-	-	-
[77]	✘	Cyclone II	90	170.24	-	-	98,594	-	838	-	10	680	-	CF=558	1920×1080	80	165.9	0.97
	✘	Stratix III	65	288.77	-	-	93,982	-	80	-	10	1,153	-	CF=418	1920×1080	140	290.3	1.01
[88]	EV76C560	Virtex 5	65	114	136@1.2μs	-	-	30	-	6,730	-	-	6,378	-	1280×1024	60	78.6	0.69
[78]	✘	Cyclone III	65	100	83	674.25	87,176	-	93,989	-	28	26,004		CF=67,985	1024×768	126	99	0.99
[152]	✘	Cyclone III	65	100	2088	149.5	31,000	-	4,20	-	-	3,031	-	-	1024×768	126	99	0.99
[115]	✘	Stratix II	90	114	-	-	614,440	-	9,019	-	-	-	-	-	1280×720	123	113.3	0.99
[92]	–[†]	Virtex 5×8	65	125	-	31,720	-	3	-	4,764	54	2,489	-	-	1024×2048	25	52.4	0.42
[89]	✘	Virtex 6	40	283	-	-	-	5	-	4,070	-	-	2,759	Slice=1,194	4096×2160	30	265.4	0.94
[108]	Python2000	Zynq 7000	28	200	-	-	2,160,000	-	29,850	-	-	-	-	-	1920×1080	96	199	1.00
[79]	Flare2KSDI	Zynq 7000	28	-	-	12,000	-	47	-	24,700	-	30,405	-	-	1920×1080	30	62	-
[50]	✔	-	-	400	-	-	-	-	-	-	-	-	-	-	2Mpixel	33	66	0.17
[109]	Python2000	Zynq 7000	28	200	-	8,000[‡]	-	34	-	14,706	38	20,316	-	-	1920×1080	96	199	1.00
[95]	✘	Cyclone III	65	100	-	-	77,408	-	13,216	-	-	-	-	-	1024×768	126	99	0.99
[103]	✔	Cyclone III	65	100	-	-	107,408	-	15,471	-	-	-	-	-	1024×768	126	99	0.99
[104]	CX590	Kintex-7	28	162	@0.81μs	453	9,738,000		-	9,799	21	15,345	-	-	1920×1080	60	124	0.77
	CX590	Virtex 5	65	148	@0.88μs	804	7,488,000	-	-	10,903	22	12,794	-	-	1920×1080	48	100	0.84
[80]	✘	Zynq 7000	28	148.5	@0.241ms	-	2,052,000	-	-	14,888	-	21,627	-	-	1920×1080	60	124	0.84
[111]	✘	Virtex 7	28	150	1024	819	-	15	-	42,611	675	22,693	-	-	1024×768	189	148.6	0.99
[96]	✘	Artix 7	28	116.81	<10ms	1,070	-	51	-	12,074	92	-	18,589	LUTRAM=110	1280×720	94	86.6	0.74

*CF: Combinatorial Function [†]Smartphone camera (×16) [‡]Includes a camera.

Table A.7: HW cost and performance measure of analog and digital TM implementations

	Image Sensor		**HW**						**HW Cost**		**Performance**			
	CMOS Sensor	Sensor DR (dB)	Monolithic	HW Platform	Foundry	Technology (μm)	Clock (MHz)	Power (mW)	Area (mm^2)	Gate Counts	Frame Size (pixel)	Speed (FPS)	Throughput (Mpix/s)	pix/ clock
[69]	✘	N/A	✘	ASIC	TSMC	0.18	30	-	4.18	-	720×480	30	10.4	0.35
[116]	✘	N/A	✘	SoC	TSMC	0.13	100	177.1478	2.85×2.85(Core)	769,620	1024×768	60	47.2	0.47
[106]	✔	-	✘	SoC	-	-	-	-	1.98×3.88(Chip)	-	64×64	-	-	-
[81]	✘	N/A	✘	Analog	AMS	0.35	-	-	-	-	256×256	-	-	-
[76]	✘	N/A	✘	SoC	TSMC	0.13	200	-	1.835×1.835(Core)	15,742	2560×2048	37	194	0.97
[70]	✔	151	✔	VSoC	AMS	0.35	-	111.2	7.33×6.78(Core)	-	180×148	30	0.8	-
[65]	✔	120	in-pixel	Analog	Lfoundry	0.15	-	-	-	-	-	-	-	-
[66]	✘	N/A	in-pixel	Analog	AMS	0.35	-	-	-	-	-	-	-	-
[105]	✔	102	✘	Analog	-	-	-	-	-	-	-	-	-	-
[68]	✔	-	in-pixel	Analog	-	0.15	52	54.72	-	-	1024×768	66	51.9	1
[83]	✘	N/A	✔	Analog	TSMC	0.35	-	41	0.039	-	-	-	-	-
[164]	✔	-	✔	Analog	-	-	-	-	-	-	512×512	-	-	-

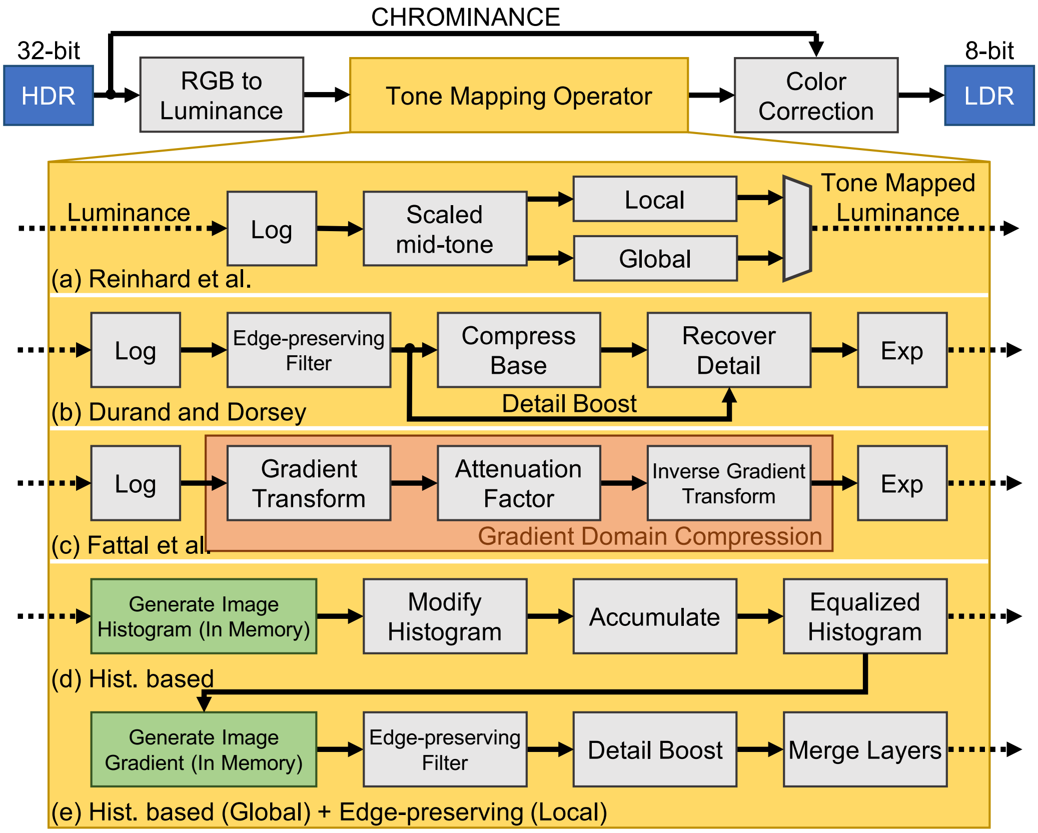


Figure A.9: General HW TM pipeline using luminance channel: (a) for Reinhard like algorithm [57] (b) Multiscale decomposition like Durand's algorithm [24] (c) Fattal's gradient domain compression [113] (d) Global and (e) Local histogram equalization.

Photographic tone reproduction for digital images

Reinhard *et al.*'s global operator performs a global scaling of the DR with a local processing is achieved by performing a dodging and burning like processes, which is a method previously used by photographers before the digital era [57]. The photographers used the dodge and burn technique to increase or decrease the light of certain parts of the scene. This method can compress the DR effectively, but the circular surround limits its widespread application due to generation of artifacts like halos in the tone mapped images. The effectiveness of HW Reinhard's TM method is reflected in number of adoptions it had on ASIC (both analog & digital implementation), FPGA and FPGA platforms as seen in the table A.1 and in the line chart Fig. A.8. An attractive feature of this algorithm is the its realization using simple pipeline with low complexity, and it is shown in Fig. A.9.

Goodnight *et al.* proposed using an implementation of Reinhard's TMO using a FPGA, and also discussed an application of TM for rendering [58]. They cleanly map the TM algorithm to the pixel processor, which allows an interactive application to achieve higher levels of realism. Additionally, they describe how the graphics HW limits the effective compression of DR and discuss modifications to the algorithm that could alleviate these problems. Krawczyk *et al.*'s work realized the local TM by constructing a Gaussian pyramid and their FPGA implementation as a stand-alone HDR processing module and achieve the real-time performance [59].

Hassan and Carletta in 2007 implemented a TMO on FPGA by combining Reinhard's and Fattal's [113] (refer sec. A.3.3) local operators using approximation of the Gaussian pyramids [61]. Kiser *et al.* mainly proposes two improvements of real-time video TM system [63]. A pre-clamping operator which the light compensation algorithm based on reference white and reference black is used to adjust the brightness of underexposure and overexposed area. They present that the pre-clamped image effectively uses more of the available output DR than no-pre-clamped image. Another improvement in this work is that the TM parameter curve over time is smoothed for video flicker removal. Kiser *et al.* implemented the real-time 1080p tone mapped video system on Xilinx Spartan-6 LX150T with 12288 clock latency.

Chiu *et al.* [116] developed a TM processor based on an ARM core with an ASIC. Their processor includes a modified global photographic TM and a block-based gradient domain compression, based on algorithms proposed by Reinhard *et al.* [57] and Fattal *et al.* [113](refer Section A.3.3), respectively. The processor can run at a 100 MHz clock rate and can compress 1024×768 HDR images at 60 Frames Per Second (FPS). However, this approach does not offer the flexibility like others [166] [167] [62] since some critical modules are implemented on an ASIC core that occupies $8.1mm^2$ of physical area in $0.13\mu m$ TSMC technology. A proof-of-concept HDR CMOS image sensor with Reinhard's global TMO was developed by Vargas-Sierra *et al.* [70] and prototyped the complete VSoC with a core area of $7.33mm \times 6.78mm$ was fabricated on $0.35\mu m$ opto-flavored technology. This system achieved video rates for QCIF resolution images with 25-bit and TM them to 7-bit for display. But, an off-chip processing is required to compute image histogram before the final display [168]. Using a new programmable pixel based on variable integration time Gouevia *et al.* [65] simulate the monotonic non-linear response of three TM functions [1, 57, 90](refer Section A.3.3) for test images given in [18, 169].

Fast bilateral filtering for the display of HDR images

Durand and Dorsey in 2002 proposed a frequency domain based technique to TM HDR images [107]. This approach is similar to an earlier frequency domain filtering [170] in which low frequencies are attenuated more than higher frequencies. Inspired by Tumblin and Turk work on low-curvature image simplifier [123] preserve local contrast in tone mapped HDR images by decomposing it into two layers (base and detail). Durand and Dorsey generate a filtered output image by combining a compressed base layer with its detail layer. The high computational complexity of this filter $O(nk^2)$, where n is the number of pixels in the image and k is the filter radius, makes it an expensive choice for real-time applications. But then this algorithm has also been implemented on HW. Marsi *et al.* [110], used a low-pass filter to split the input image in to a low frequency base layer and high frequency detail layer. They targeted an automotive driving assistance application using a Xilinx Virtex-II FPGA, and this system included temporal smoothing to prevent flickering and color shifting. They used a $\frac{1}{4}$ down-sampled previous frame, for temporal smoothing in order to reduce memory usage and the image was stored on FPGA. Their operator achieves 24 FPS for 125×86 resolution. Nosko *et al.* [108, 109] described a fast implementation of HDR merge processing by multiple exposure and a local TMO involving bilateral filtering [107]. This work also proposes an application of de-ghosting method, which is dedicated for FPGA implementation. Compared with the use of Gaussian filter to detect local luminance, bilateral filter can preserve sharp edges, but also require more HW cost.

Gradient domain HDR compression

Fattal *et al.* proposed a novel frequency-based method to achieve tone compression by manipulating gradients of the image [113]. According to Fattal *et al.*, in the HDR images changes in luminance are proportional to the changes in the gradients. Hence, tone compression can be achieved by attenuating the magnitudes of the large gradients while maintaining their directions, and keeping the small gradients which corresponds to the fine details. However, for producing the output LDR image requires the computationally expensive Poisson-solver, which makes it unappealing for real-time applications. However, Vytla *et al.* [114] developed a HW implementation of gradient domain HDR TM using the Poisson equation solver inspired by Fattal's operator. As expected, this implementation is computationally very expensive, as it seeks to solve Poisson equation. The authors of [171] developed a local Fattal's operator to solve the Poisson equation locally and repeatedly, thus making it parallel and hence executable in real-time. The modified Poisson solver uses only local information from pixel and its 3×3 window neighbors, for computing a tone mapped pixel independent of Fattal's operator on other pixel locations within the window. An Altera Stratix II FPGA was used to implement this algorithm, and it outputs grayscale tone mapped images.

A TM algorithm for high contrast images

This local TM method is similar to the Reinhard's photographic TMO [57]. The underlying mechanism of this TMO is like the initial stages of our human visual system. Where two different range compression functions are used. For preserving the local contrast the intensity range is compressed, which in principle leads to averaging over the large neighborhood without keeping the fine details. The overall visual contrast is generated by adding the fine details at the later stage [93]. Roch *et al.* [94] proposed a local TM algorithm implementation [93] on graphics cards. They also present a modification of the luminance local adaptation computation, that maintains the same quality appearance of the original TMO.

Tone-mapping HDR images by novel histogram adjustment

Duan *et al.* proposed an iterative approach to adjust the histogram of photoquantity in logarithm scale [84]. Their method takes the weighted sum of the contrast modification between the direct linear scaling and the naive equalization. However, instead of performing the weighted sum of the two directly, their method performs recursive binary cuts to find the pivot point that defines the sub-interval for the next iteration. The cut is found based on the weighted sum of the supposed equalized point and the linearly scaled point. Further, their approach can be applied to local patches of the image independently. They demonstrated that locally adjusted contrast is able to preserve the details better than a global image contrast adjustment. However, this creates edges between locally tone mapped patches. Their solution is to apply further smoothing across the patches to eliminate the edges, this algorithm was adopted on HW in the following works.

Tian *et al.* [85] proposed a real-time HW local implementation based on global TM [84]. They proposed an algorithm of segmenting the image into 64×64 independent rectangular blocks to sense local luminance. A boundary and halo artifact elimination algorithm and a noise suppression algorithm are included in this work to improve the image quality of tone mapped image. Lapray *et al.* [64, 86–88] in a series of publications presented several full imaging systems using a Virtex-5 and Virtex-6 FPGA-based processing cores. Their HDR imaging pipeline uses a HDR monochrome image sensor to provide a 10-bit data output and making use of Debevec and Malik's [55] fusion method to produces HDR video from multiple images. Using a special memory management unit [64], they can generate HDR images at the same frame rate as their camera output, which requires current frame and two previously captured frames. For HDR TM they used global TM algorithms of Duan *et al.* [84] and Reinhard *et al.* [57].

Adaptive logarithmic mapping for displaying high contrast scenes

Drago *et al.* proposed a fast global TMO which is based on luminance logarithmic compression [90]. This method operates based on scene content, and uses a predetermined logarithmic basis to preserve contrast and details. For example, if an image is mapped to low base logarithm it will appear more brighter than one which is mapped to higher base logarithm. In other words, an image mapped to higher base logarithm would appear darker therefore, revealing details from brighter areas. The complexity of this algorithm is evident from the TM function (ref table A.1), whose implementation requires additions, subtractions, multiplications, and precise calculation of exponential. Popovic *et al.* [91] used global TM similar to Drago's operator and used a logarithmic mapping function, to calculate displayed luminance from the ratio of world luminance and its maximum. Logarithm calculations are known to be computationally expensive, so Popovic *et al.*, used Taylor and Chebyshev polynomials to approximate logarithms (refer sec. A.3.4 and [172]). Further, they designed a camera ring consisting of 18 independent cameras with different exposures to create panoramic HDR video [92].

Finally, Zhao *et al.* [60] presented FPGA implementations of two state-of-the-art TMOs with real-time performance. Additionally, the reported six other FPGA-based TMOs [57, 90, 93, 107, 138–140], and conducted an experimental evaluation to explore which TMO is faster for HW implementation.

Novel TM algorithms

From Fig. A.8(d) we know that recently, many researchers prefer to design and implement their own novel TM algorithms which best suits their design objectives or target applications. In table A.1 we have grouped these algorithms based on the functions, and we will describe these implementation next. In order to realize these TMO functions different mapping methods like basic exponential, logarithmic, histogram-based or sigmoid functions are employed. § A.3.4 briefly describes various methods and cites references which can be followed to realize them on HW.

Exponential mapping applies an exponential function to the HDR pixel, to remap its value to interval $[0, 1]$. A simple operator can be expressed as

$$L_d(x, y) = 1 - exp(-\frac{L_{w(x,y)}}{L_{mean}}) \tag{A.10}$$

Digit-by-digit algorithm can be used to easily implement the exponential function on HW [173]. Logarithmic compression can be used to remap medium DR pixels to 8-bit display devices [1]. A simple log operator can be defined as

$$L_d(x, y) = \frac{log_{10}(1 + L_W(x, y))}{log_{10}(1 + L_W(max))} \tag{A.11}$$

. Simple lookup methods can approximate log functions with good accuracy [174].

Histogram Equalization (HE) based TMOs map HDR pixels to LDR display devices while preserving the most important details in the HDR images. This non-linear method modifies the HDR pixel distribution by redistributing it to produce a uniform distribution over a new range of the LDR display range. The global HE operates by redistributing the HDR pixels into LDR of N levels such that all LDR levels have equal quantity of pixels. This operation using a single transformation function cannot enhance features of the local region. There are some recent global HE methods which offer more sophisticated histogram manipulations like recursively separated and weighted HE [175], but they do exhibit the limitations of single transformation function. Therefore, time consuming local histogram equalization based methods, which make use of well-defined transformation function for each local region is required to enhance darker regions in an image [176]. Using block diagrams presented in Fig. A.9(d) and (e) we illustrate the pipelines required to implement global and local HE-based TM.

A.3.4 HW architectures for transcendental functions

Transcendental functions like the exponential function, the logarithm, and the trigonometric functions are frequently finding applications in digital signal processing, communication systems, robotics, computer graphics etc. In this section we report the HW implementations which are used in realizing TM functions, these are carefully designed to be accurate and also be sufficiently fast in order to operate in real-time. Performance of these HW architectures are also important and are characterized by its error characteristics which can be measured using one of the following metrics: (i) maximum absolute error, (ii) mean error, (iii) median, (iv) SD, (v) root mean square error and (vi) error probability distribution. In later sections A.4.2 and A.4.3 we discuss how HW implementations affect the tone mapped images.

For realizing these HW transcendental functions there are many well known approaches some of them are:

Lookup table

Lookup Table (LUT)-based function implementation is a simple approach in which pre-computed values for every argument x of the function $y = f(x)$ is stored in a memory array. LUT-based methods were popular initially, because in the multiplier-less FPGAs functions can be realized quickly using simple lookup and addition operations. However, the size of the table is directly dependent on the data-width (x, y) and can become prohibitively large when seeking better precision [177]. For example, if x, y are numbers of data-width W-bit, a memory of size $2^W \times W$ is required.

Taylor series expansion

The Taylor-series approximation is used to approximate $f(x)$ as an infinite sum of terms that are expressed in terms of the function's derivatives at a single point. Using Taylor-series, we can approximate complex functions from a series of low level functions which can be mapped on HW. Generally, we can represent a real function $P(x)$ at any arbitrary point $x = a$, using Taylor-series as given in Eq. A.12

$$P_n(x) = \sum_{i=1}^{n} \frac{f^{(i)}(a)}{i!}(x-a)^i \tag{A.12}$$

Nilsson *et al.*, in [178] demonstrate how Taylor series expansion of most transcendental functions such as exponential, logarithmic and trigonometric functions can be easily implemented on HW. For example, the exponential function e^x can be expressed using Taylor-series as in Eq. A.13 which can be easily realized using 7 multipliers and 6 adders [178].

$$e^x = e^a \left[\sum_{n=0}^{\infty} \frac{(x-a)^n}{n!} \right] \tag{A.13}$$

Digit-by-digit algorithm

Convergence of the Taylor-series given in Eq. A.13 becomes computationally expensive for x approaching 1 [177] and LUT-based method are bound by

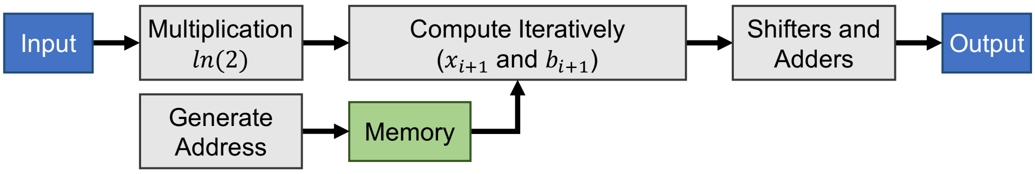


Figure A.10: Block diagram for realizing digit-by-digit algorithm for exponential function [173]. Logarithm and other transcendental functions can be calculated similarly.

memory constraints if targeting small low-power embedded platforms. There exists more HW friendly methods, for example given an exponential function:

$$y = m \times 2^{n \times x} \tag{A.14}$$

Equation A.14 can be rewritten in terms of the integer (I) and fraction (F)parts of the exponent $n \times x$ as,

$$y = m \times 2^{(I+F)} \tag{A.15}$$

The integer part of Eq. A.15 can be easily realized using shift registers only. The fractional part (2^F) can be expressed as $e^{F \times ln2}$. Equation A.14 can be rewritten as:

$$y = m \times 2^I \times e^{F \times ln2} \tag{A.16}$$

We can draw inspiration from the Kantabutra's digit-by-digit method [173] to realize the HW implementation of the fractional part in Eq. A.16, and a plausible architecture is shown in Fig. A.10. By taking $x \times log_2 e = I + F$ we can extract the integer and fraction part of the exponential function in Eq. A.14. We can easily implement $log_2 e$ as the sum of powers of two there by requiring shift and add logic only [179]. For the implementation of the exponential operator we can follow the footsteps of earlier works (p. 226 [177]) [173] [116], where they assume that the argument x is limited to the range $[0, ln2)$. The function $y_i = e^x$ is approximated by using the following recurring equations:

$$y_{i+1} = \begin{cases} y_i \times b_i & a = 1 \\ y_i & otherwise \end{cases} \tag{A.17}$$

$$x_{i+1} = \begin{cases} x_i - ln(b_i) & a = 1 \\ x_i & otherwise \end{cases} \tag{A.18}$$

where, the data pair x_i, y_i are set to initial values x and 1, satisfy Eq. A.19.

$$y_i = e^{-x_i} e^x \tag{A.19}$$

Here, the value of x_i is updated as shown below [173]:

$$x_{i+1} = x_i - ln(b_i) \tag{A.20}$$

where, $b_i = 1 + a_i 2^{-i}$ and $a_i \in 0, 1$. The value of a is set to 1 if $x_i \geq ln(1+2^{-i})$ and 0 otherwise. Depending on the data-width W_{frac} of the fraction part of the TM function we can pre-calculate and store W_{frac} values of $ln(1+2^{-i})$ with $0 \leq i < W_{frac}$ in a memory table. The block diagram of this iterative computation technique is shown in Fig .A.10. This approach can be used to compute logarithm and other transcendental functions.

A.4 Normalized HW cost, relative comparison, and discussion

Benchmarking different HW TMO implementations for performance is a complex task, and a poor benchmarking method may result in incorrect outcomes. In this paper, we have followed an earlier work of Park *et al.* [80], in developing our cross implementation benchmarking methodology to compare HW TMOs which have been realized on different FPGAs.

A.4.1 Hardware

As stated earlier, one of the main objective of our survey is to determine the quality of various HW TMOs. In Fig. A.11 we present ASIC and FPGA implementations grouped on a map. In this map a blue dot represents an FPGA implementation, and green one is ASIC. The positions on this map highlights these implementations according to their frequency and throughput. We computed throughput as given in Eq. A.21

$$Thoughput = Height_{Image} \times Width_{Image} \times FPS \tag{A.21}$$

Frequency and throughput are very important parameters, as it allows us to predict the algorithm's effectiveness for processing HDR images of wide-ranging resolutions. In other words, more throughput per cycle implies that the algorithm is faster, and is well optimized. The orange circle in Fig. A.11 represents the year of publication. A paper which has full circle (in orange color) means chronologically newer paper, and arguably this paper would have improved design architecture than those in previous papers, thereby, realizing some design optimization. The black circle gives information of TM. The width of black circle corresponds to the size of kernel that is used to calculate the local luminance. Works based on global TMO are without

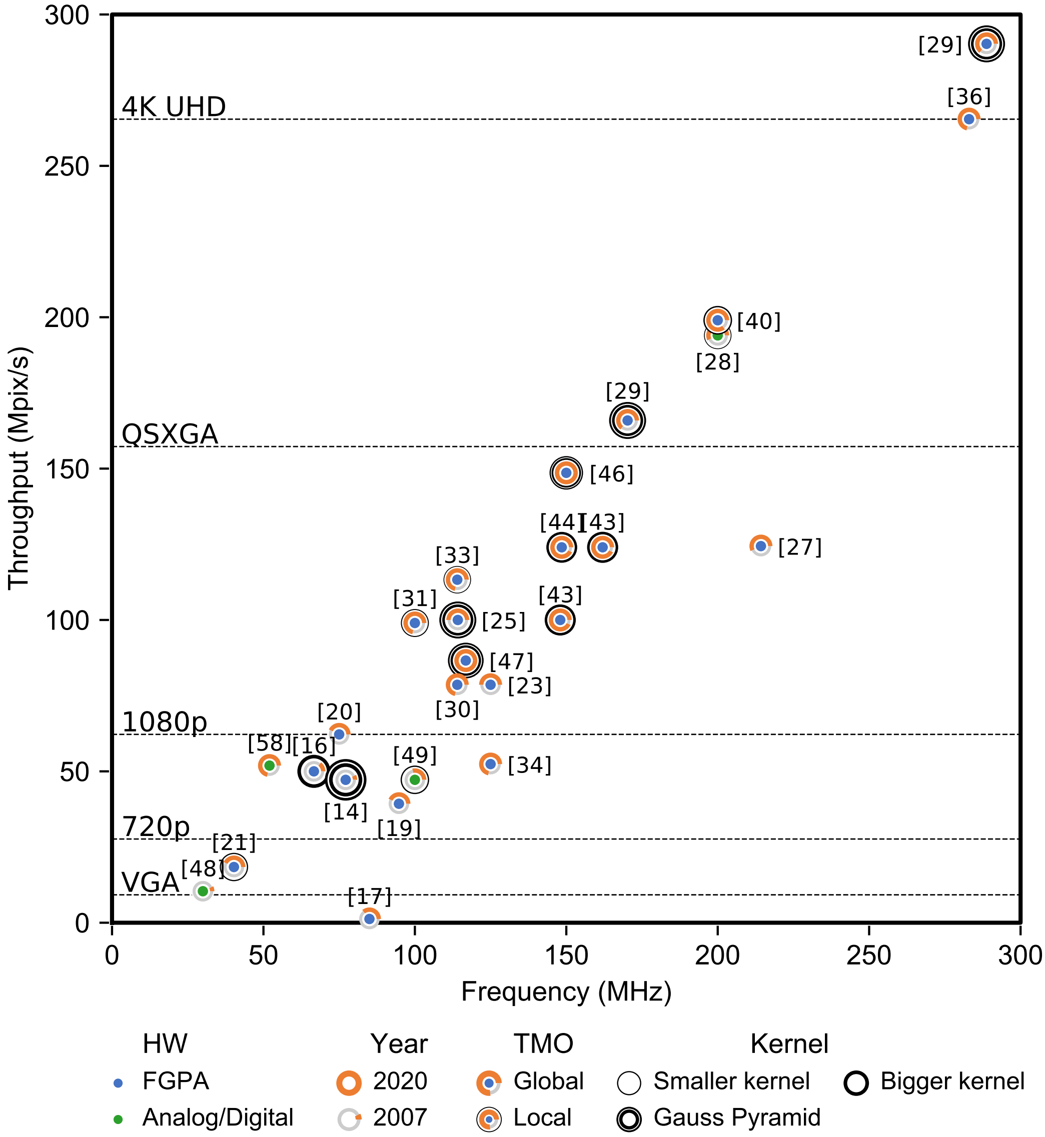


Figure A.11: TMOs performance measurement: Throughput versus operating frequency is an important measure for real-time performance.

black circle because they do not use any local statistics. With respect to local operators, some of those works have calculated local luminance by using Gauss Pyramid, and such implementations are represented as double black circles and their performances are also included in Fig. A.11. Figure A.12 illustrates the HW cost of FPGA TM implementations. Like in Fig. A.11 the location of each point is grouped with respect to the system throughput and frequency, so that same position corresponds to same paper in both figures

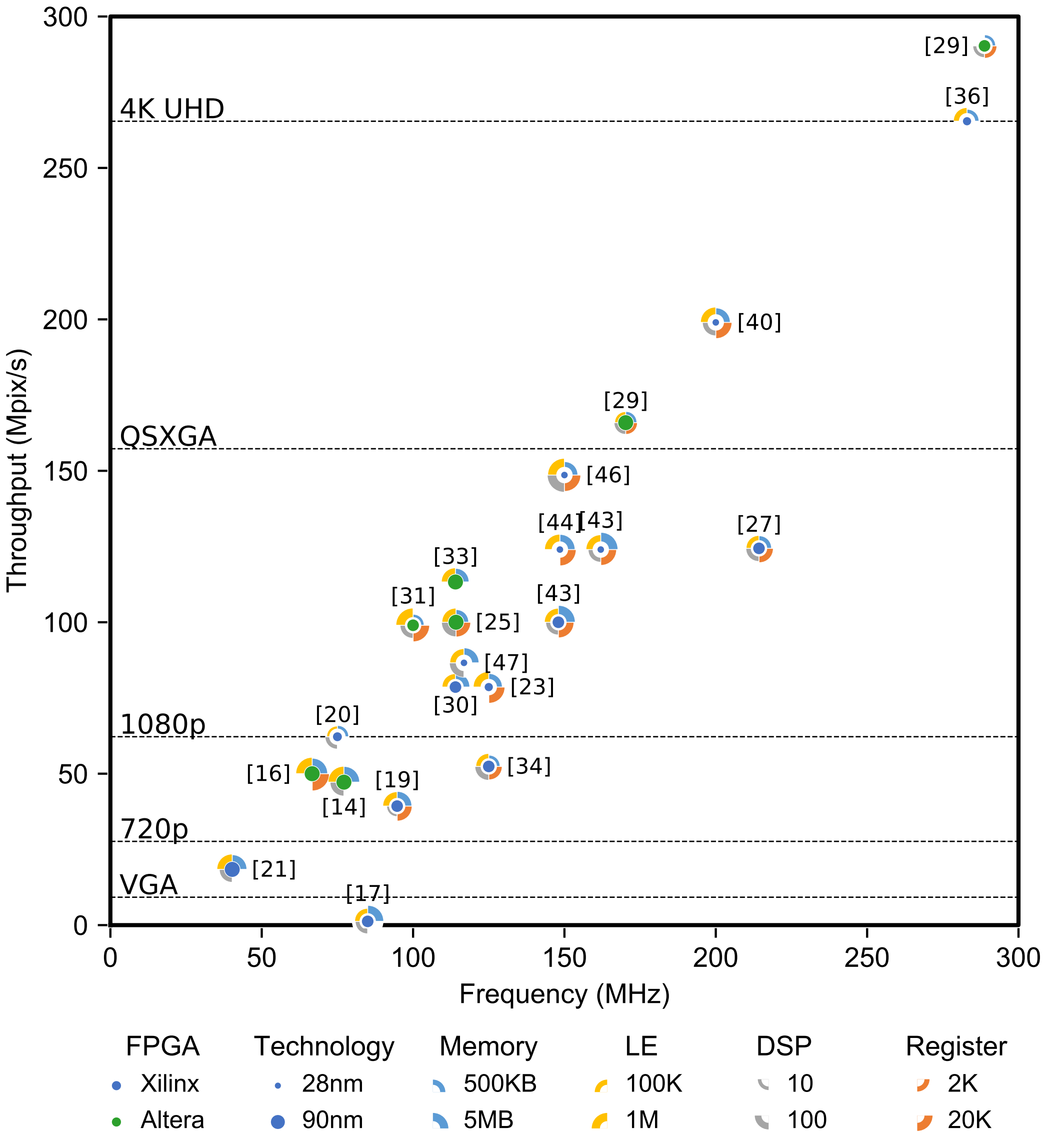


Figure A.12: Throughput versus relative HW cost: TMO computational complexity is evaluated in terms of memory, DSP, logic elements and registers. Global TMOs are usually light-weight in comparison to local TMOs.

of merit. The color of center circle of each point informs about the FPGA manufacturer (ALTERA/XILINX) that were used in research. And the size corresponds to the FPGA's manufactured technology. Smaller size represent that they use newer 22nm technology.

The comparison of different HW algorithm implementations are hard for the following two reasons. First, use of different target platforms makes a fair comparison difficult. Second, in manuscripts authors do not specify their HW implementations thoroughly enough to ensure easy comparison with all other works. In this survey, to compare TMOs implemented using different FPGA families, we follow the HW normalization strategy proposed by Park *et al.* [80]. For algorithms implemented on Xilinx Virtex-4, each LUT and register consumed can be substituted with 16-bit memory. For systems that are implemented on a Virtex-7/Zynq7000, and those seven series FPGA's LUT and registers are equalized with 32-bit memory. For Altera Cyclone III based systems, according to an article analyzing the difference between the two FPGA fabrics [180], one unit of the logic element used in Cyclone III is 1.3 times larger than one unit of LUT used in Virtex-4, so the resource utilization is converted into the estimated amount of LUTs of Virtex-4. This method of normalizing HW resources between different FPGA fabric usages for comparison has also been adopted by Choi *et al.* [181]. The size of the circle sector area is proportional to the HW cost. Different colors represent different type HW. By studying Fig. A.11 we can see that recent works [76, 104, 109] have high throughput.With streaming applications like TM which continuously process data, throughput is the most interesting design aspect as it will define the performance of the TM application. More throughput means that more data can be processed in the same instant of time. To improve the performance of slower algorithms, one of the most effective ways is by adding extra pipeline stages. FPGA designs have a synchronous nature consisting of delay elements and logic, which means that it highly benefits from extra pipeline stages.

Another interesting design aspect is the memory cost, Fig. A.13 shows the relationship between TMO HW implementation memory and output frame size. Frame size is computed as shown in Eq. A.22. From the Fig. A.13 we can observe that, as expected global TMOs require less memory than local TMOs. In the Fig. A.13, we have grouped the algorithms based on the performance as which of these implementations are better designed to reduce memory cost. Recent local and global TMO works [79, 80, 109] report low memory usage by highly optimizing their designs. For example, Park *et al.*, designed a frame-less TMO system, and only used a small line buffer [80]. They further optimized their design by building an approximate convolution block of a 29×29 Gaussian filter. A conventional 2-D filtering scheme would have required 29×29 convolutional operations between a pixel and its coefficient. They implemented it by using two 1-D separable filters operating vertically and horizontally, thereby reducing the number of operations to 29×1 plus few additional adders.

$$FrameSize = Height_{Image} \times Width_{Image} \times BitWidth \tag{A.22}$$

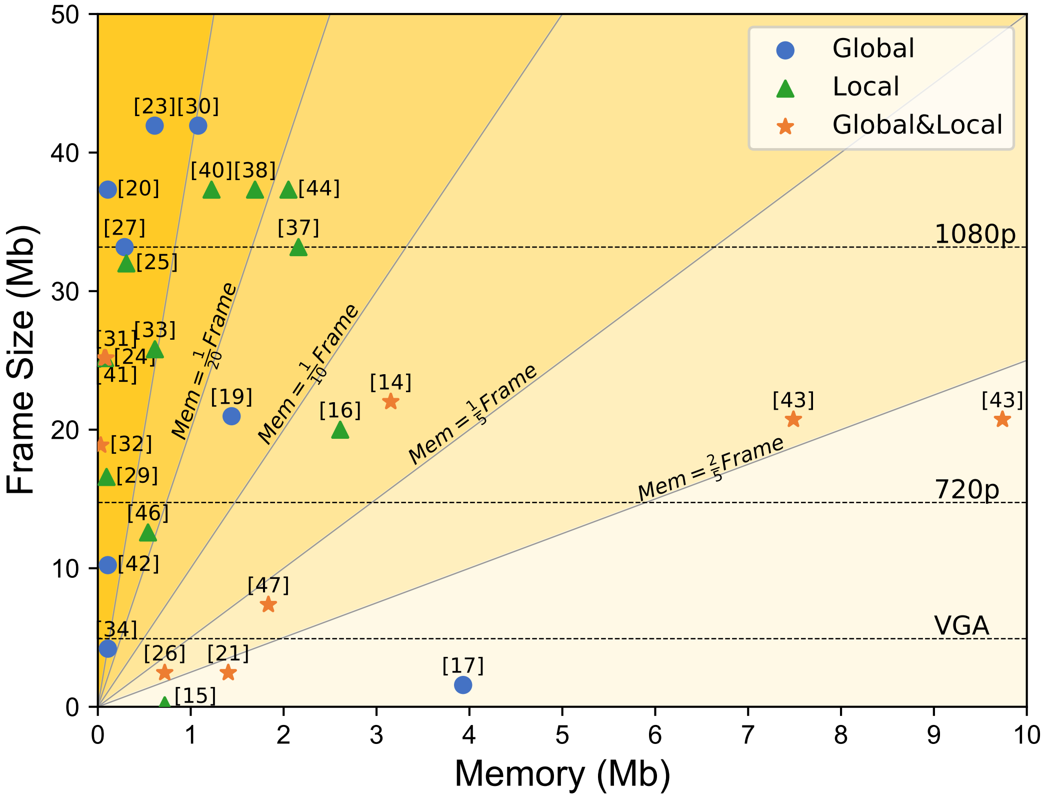


Figure A.13: Plot illustrates memory cost of TM algorithms versus output images. Typically global TMOs have lower memory requirements than local TMOs. (The bit width of reference lines in this figure is set to 16-bit.)

Table A.8: Image quality metrics used to measure HW TM accuracy

Image Quality Metrics		**FPGA**	**FPGA**	**Analog**	**Digital**	**Ideal Value**	**Definition**
PSNR*	Peak Signal to Noise Ratio	[58], [94]	[72], [61], [62], [13], [75], [75] [114], [101], [91], [77], [88], [88] [78], [78], [152], [92], [108] [109], [104], [112], [111], [96]	[83], [164]	[62], [69] [116], [70]	higher	PSNR=20$log_{10}(\frac{MAX^{\dagger}}{RMSE})$
SSIM	Structural Similarity Index		[75], [91], [77], [88], [78] [152], [109], [104], [112], [111]	[83]		1	[182]
TMQI	Tone Mapped Image Quality Index	[99]	[95], [103], [104]	[68], [67]		1	[183]

*This group include PSNR-like metrics (■PSNR 🟩RMS% 🟧RMSE). †MAX is the maximum possible pixel value of the image.

A.4.2 Data conversion for optimal HW specification

Conventional image processing algorithm's HW implementation design flow begins with a SW model designed in C/C++/MATLAB/Python and simulated using DSP SW tools like MATLAB/Simulink (Mathworks Inc) or others like Scilab etc. These SW implementations using floating-point arithmetic achieve high degree of accuracy in capturing the mathematical model of the algorithm at behavioral level, often without considering the underlying HW implementation details. HW circuits are designed meticulously because, cost of modern electronic devices is usually measured in terms of silicon area (chip footprint), power consumption and algorithm/application execution time. Engineer's strive hard to keep these three factors to a minimum while attaining all the system objectives. Balancing these goals are extremely challenging in nature, and usually a delicate trade-off between system performance and cost has to be planned in advance. Therefore, direct porting of a SW TM algorithm to any HW platform will be inefficient and even may lead to system failures.

The choice of arithmetic operators used to implement the algorithms, have a decisive impact on the cost-performance trade-off. On a HW platform fixed-point arithmetic operators are ideal choice as they require low area (footprint), low power, and have low latency [184]. But, the floating-point to fixed-point conversion process is an optimization problem which derives the data word-length [185]. This adaptation is time consuming, and has to be carefully crafted as this leads to a trade-off between cost and performance. It is worth noting that there are automatic floating-point to fixed-point conversion tools which can significantly speed-up this process [186]. Recently, some High Level Synthesis tools (HLS) like Intel HLS, Cadence Stratus etc. have emerged, which can directly generate register transfer level (RTL) implementations from a C/C++ fixed-point specification of the algorithms [187–190]. Use of these tools can speed-up the re-design effort, but the designs may not be fully optimized.

A.4.3 Image quality metrics used to measure HW TMO accuracy

Image Quality Assessment (IQA) plays vital role at many levels of the design cycle, and an early assessment is inevitable to prove the usefulness of the algorithm. Subjective user study is the most reliable means to measure image quality. However, it is not always feasible for practical reasons. During the HW development stages, objective image quality metrics are used to evaluate the system performance and analyze if there is more room for

HW optimization while maintaining acceptable image quality. IQA studies have been actively carried out and there are several quality metrics in literature [191–193]. These IQA methods try to accurately predict the subjective preferences of a common human user by surveying the perceived quality of visual data presented to the user. For algorithms implemented on HW, IQA is measured in terms of image distortion which is caused by the approximations due to floating-point to fixed-point translation. PSNR is a preferred metric to measure the pixel value distortion between SW and HW tone mapped images. Various other PSNR-like metrics also have been used and they are all listed in table A.8.

PSNR value approaches infinity as the Root-mean-square Error (RMSE) approaches zero, this shows that a higher PSNR value provides a higher image quality. Nonetheless, PSNR perform badly in discriminating structural content in images since various types of degradation applied to the same image can yield similar RMSE value [194, 195]. Therefore, some more elaborate methods attempt to incorporate structural information in IQA. Wang *et al.* [182] proposed Structural Similarity Index Measure (SSIM) based on human visual perception for measuring the similarity between two images, and Yeganeh *et al.* [183] proposed an objective quality assessment algorithm for TM and named as Tone Mapped Image Quality Index (TMQI) which is based on SSIM and naturalness. In TM HW implementation, SSIM is used for pixel-to-pixel measurement. They estimate the similarity between tone mapped images by SW operator and HW implementation. In HW implementation, losses can occur due to approximate calculations, and PSNR or SSIM metrics are used to measure these losses. A higher PSNR or a higher SSIM is desirable which implies smaller loss in HW implementation.

A.4.4 HW specification versus image quality

Every image processing application seeks to achieve good output image quality, and TMOs are no different. Local TMOs are known to produce better images than global TMOs as they can reproduce both global and local contrast [4]. However, local TMOs are computationally more expensive than global TMOs and also may generate artifacts like halos around edges and amplify noise [107]. Therefore, additional functions are required along with local TMOs to reduce such artifacts.

Noise suppression and Halo management in tone mapped images

As stated earlier, local contrast enhancement algorithms risk boosting noise. Many algorithm designers proposed various techniques to suppress noise.

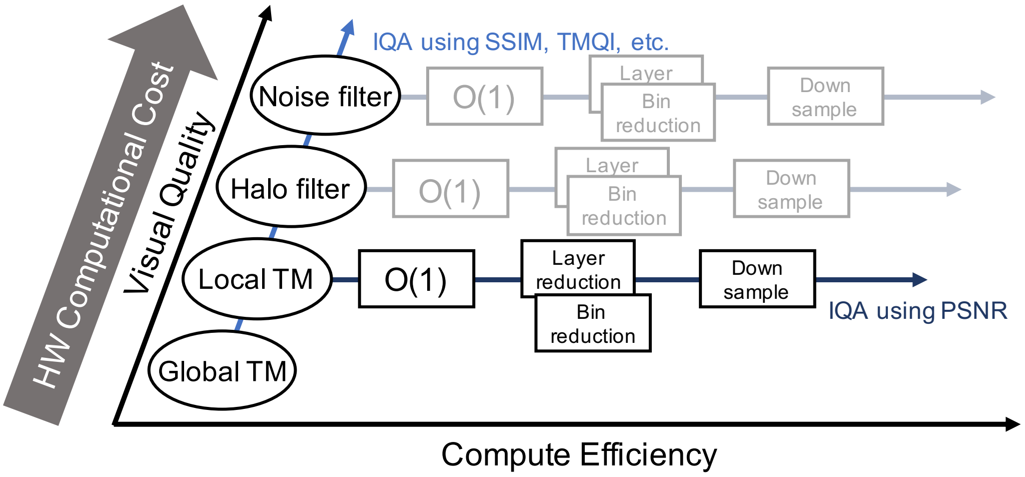


Figure A.14: Plot illustrating relationship between image quality and HW cost. Computational efficiency can be improved by implementing HW friendly techniques.

Eilertsen *et al.* presented a novel noise control with display adaptivity to produce high contrast and detailed video given the display limitations [6]. Li *et al.* presented a new logarithmic CMOS sensor and a histogram-based tonemap operator which is derived from cumulative distribution function with an objective to suppress noise in the tone mapped image [102]. Ambalathankandy *et al.* in [78] implemented a halo reducing filter based on a Gaussian-like filter [151]. In their TMO, halos were created around small bright features due to strong attenuation of neighboring pixels due to convolution operation with low-pass filtering. The HW scheme for reducing such halos resulted in a very expensive implementation.

Nosko *et al.* [109] proposed a ghost removal algorithm which significantly improves perceived quality of HDR image. This method is based on an earlier work of Grosch [154], and requires only simple arithmetic operations and thus it is suitable for implementation on FPGA. The ghost detection step is implemented before the HDR merging step. By constructing the ghost-map, marked pixel positions are treated differently from unmarked ones during the HDR merging. Recently, Ambalathankandy *et al.* [104] designed and implemented an Local Histogram Equalization (LHE)-based TM that requires only one box filtering with a wide kernel. Their TM algorithm uses two curves, one corresponding to the edge region and the other for gradation. As noises in gradation part are much more noticeable than in the edge and texture part, they use an alpha blending function to suppress noise in gradation region, and they include a halo control mechanism to manage

Table A.9: Filter characteristics

Filter	**Algorithm**	**Computational Complexity**	**Memory**	**HW Friendly**
Box	McDonnell Box Filtering [196]	Constant Time	Low	Yes
	Integral Images [197]	Constant Time	High	No
Gaussian	Classic 2D (FFT-IFFT) [198]	$O(NlogN)$	High	Special HW
	Multiple Box* [199]	Constant Time	Low	Yes
	Spline Approximation [200]	Constant Time	High	No

* *Cf.* Central limit theorem.

light/dark halos individually using a simple weighting function on the bin-reduced histogram.

Computational efficiency and image quality

In § A.4.4 we reported some earlier works which implemented additional filters for improving the image quality. However, these local operations must be repeated over large amounts of data. This results in substantial computational effort as shown in Fig. A.14, and the visual quality does improve but at the expense of additional HW. At the same time, depending on the bit width, image and kernel size will also result in increased HW cost. As shown in Fig. A.14, computational efficiency can be improved by designing $O(1)$ algorithms which are not memory intensive and HW-friendly. Some of these approaches and their characteristics are listed in table A.9. McDonnell box filter [196] can be implemented by line buffers without frame buffers, therefore it is ideal for HW design. Other approaches like layer and/or bin of a histogram are directly related to the trade-off between calculation cost and image quality. Also, in spatial approximation, down-sampling or tiling with interpolation is applied for improvement of HW efficiency. When applying filter stacking, objective metrics like PSNR is used to ensure that HW-efficient approaches maintain acceptable accuracy. Metrics like SSIM and TMQI are suitable for assessing image quality when incorporating new filters.

A.4.5 Video TM artifacts

In the previous sections, we presented different HW TMOs designed to tone map HDR images. Recently, high quality HDR video content generation is becoming easily available [6, 56], therefore, the temporal aspect of TMO cannot be neglected. Direct application of TMOs to every frame of HDR video can lead to generation of various temporal artifacts [201]. A common problem encountered in video capture under artificial lighting is the flickering artifact which is caused by the combination of frame rate and shutter speed. Flickering artifact appear as distracting flashes which is undesirable and must be removed. When the flicker frequency is higher than the frame rate the artifact can be easily managed from the sensor side by averaging pixels within the constant exposure time [44] (also see Sec. A.2.1). If the flicker frequency is lower than the sensor frame rate then, some post-processing techniques will be required to remove the video flicker [202]. Furthermore, Kiser *et al.* have proposed a leaky integrator-based flicker removal method for HW TMO [63].

A.4.6 Design bottlenecks

Real-time image processing applications are bound by timing constraints, i.e., for every pixel clock the processing pipeline should consume one pixel from the camera side and deliver one to the display side. Any missed pixel on either side would lead to loss of information or cause blanking display, this is known as the throughput constraint. When porting SW algorithms to HW an inherent design problem is that the SW code is developed on a general purpose Central Processing Unit (CPU). Therefore, the algorithm is highly sequential, and it is useful to exploit the fast CPU. However, this is not the case on HW platform. For example, FPGAs are clocked at much lower frequencies and designers should exploit this parallelism to implement real-time systems. Another type of constraint that has to be met for real-time TM system, is the pipeline latency. Here, latency implies how many clock cycles are required to process one input pixel to processed/tone-mapped pixel. Previously, in § A.3.3 we discussed various HW implemented TM algorithms, and their underlying TM functions could be easily realized using the methods discussed in § A.3.4.

Memory bottleneck is crucial for implementing image processing algorithms on HW. FPGAs which have highly parallel logic blocks and fast re-configurable interconnects to speed up window (kernel) operations offers a solution to this problem. However, the interface speed between the TM accelerator and the rest of the FPGA system can become the bottleneck. The cost of moving data between off-chip memory and the accelerator can be detrimental and outweigh the benefits of implementing the FPGA system. Therefore, a well thought out operation sequence that obeys raster order should be chosen, because other computation order would usually require the whole frame to be buffered. Like caching can reduce the memory bottlenecks on CPUs, streaming FIFO interfaces can reduce the amount of pixel accesses on FPGA HW. Also, FPGAs are provided with BRAMs which can be read and written at the same time at every clock cycle, allowing one stream of values to be stored and one stream to be extracted in parallel.

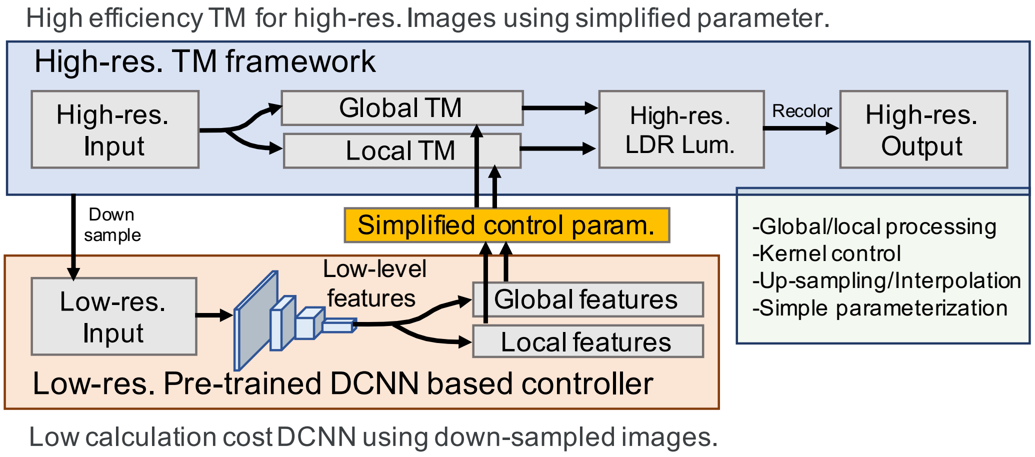


Figure A.15: Block diagram for a plausible ML-based TM implementation on HW.

A.5 Future perspective

Currently ML-based methods have become a very important tool to solve many computer vision tasks like image classification, face detection, and video analysis. As a future perspective we would like to leverage its potential by accelerating ML-based TMOs using HW platforms. In this section we will explore the challenges and opportunities that we will encounter for such systems. Usually image processing tasks would require multiple convolution with fully connected layers, which are exorbitantly computationally intensive (for example, the operations in CNNs are over billion operations [203]). Realizing such systems on resource constrained embedded systems would require very novel architectures and algorithms. FPGAs have been preferred platform for realizing CNN HW accelerators for their following well-known features: reprogrammability, low-power design features, and quick design time [204, 205].

Using Fig. A.15 we demonstrate how Gharbi *et al.*'s HDR-Net like architectures are ideal baselines for realizing TMOs with Deep Neural Network (DNN) on HW [206]. We find such designs are more HW friendly because of the following features. First, the bilateral grid inspired architecture represents local tone-control as simple parameterized luminance grid in the space. Thus, the HDR-Net like high throughput design can approximate a TM system by using a high-resolution guidance map which slices into the grid to produce a unique, interpolated, affine transform to be applied to each input pixel. Second, a lightweight DNN is vital for realizing HW TM system. Thus, low-resolution DNN with down-sampling and optimum interpolation are important. Also, simple data transfer between DNN and TM system is required for reducing HW load. In this scenario, output format of such DNN architectures becomes simple. Finally, good high resolution off-line data set and training method through whole architecture is key for realizing this system.

In this survey we report a comprehensive list of about 60 TM algorithms that have been implemented on HW platforms like ASIC, FPGA and FPGAs to accelerate the data intensive algorithms for real-time performance. Design and implementation of such algorithms are usually complicated, as HW porting of their SW equivalent may need to be redesigned for efficient HW implementations. This effort leads to various design challenges that are encountered during the HW development. Usually the SW algorithms are realized with floating-point data type and fixed-point conversion of the algorithms lead to loss of accuracy (image quality). Also, authors prefer to design their own optimized TM functions over others in order to avoid system redesign (line chart A.8(e)). In our literature survey we found that, various objective quality metrics have been used to demonstrate this distortion. For easy reference we have summarized all these objective metrics used in this survey. Finally, in this paper we also demonstrate the link between HW cost and image quality thereby, illustrating the underlying trade-off. From Fig. A.8(a) we can observe that from early days FPGAs have been preferred platform for realizing real-time TM applications. Features like rapid-prototyping and good support with developmental tools have made FPGAs a popular HW platform. Another attractive feature of FPGA proven design is that it can be ported to structured ASICs which are available from many vendors, there by giving developers a faster route to market their products [207].

Bibliography

[1] Erik Reinhard, Wolfgang Heidrich, Paul Debevec, Sumanta Pattanaik, Greg Ward, and Karol Myszkowski. *High dynamic range imaging: acquisition, display, and image-based lighting.* Morgan Kaufmann, 2010.

[2] Mario Mustra, Kresimir Delac, and Mislav Grgic. Overview of the dicom standard. In *2008 50th International Symposium ELMAR*, volume 1, pages 39–44. IEEE, 2008.

[3] The best 4k monitors for 2020. `https://www.pcmag.com/roundup/370480/the-best-4k-monitors`. Accessed on December 24, 2019.

[4] Francesco Banterle, Alessandro Artusi, Kurt Debattista, and Alan Chalmers. *Advanced high dynamic range imaging.* AK Peters/CRC Press, 2017.

[5] Laurence Meylan. Tone mapping for high dynamic range images. Technical report, EPFL, 2006.

[6] Gabriel Eilertsen, Rafał K Mantiuk, and Jonas Unger. Real-time noise-aware tone mapping. *ACM Transactions on Graphics (TOG)*, 34(6):1–15, 2015.

[7] Bernd Hoefflinger. *High-dynamic-range (HDR) vision.* Springer, New York, 2007.

[8] Dani Lischinski, Zeev Farbman, Matt Uyttendaele, and Richard Szeliski. Interactive local adjustment of tonal values. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 646–653. ACM, 2006.

[9] Tobias Kalb, Lester Kalms, Diana Göhringer, Carlota Pons, Fabien Marty, Ananya Muddukrishna, Magnus Jahre, Per Gunnar Kjeldsberg, Boitumelo Ruf, Tobias Schuchert, et al. Tulipp: Towards ubiquitous low-power image processing platforms. In *2016 International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation (SAMOS)*, pages 306–311. IEEE, 2016.

[10] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence*, 35(6):1397–1409, 2012.

[11] Leon Adams and Strategic Marketing. Choosing the right architecture for real-time signal processing designs. *Texas Instruments*, (SPRA879).

[12] Eric Young and Frank Jargstorff. Image processing and video algorithms with cuda. nvision 2008 presentation. Technical report, 2008.

[13] Raquel Ureña, Juán Manuel Gómez-López, Christian Morillas, Francisco Pelayo, et al. Real-time tone mapping on gpu and fpga. *EURASIP Journal on Image and Video Processing*, 2012(1):1, 2012.

[14] Donald G Bailey. *Design for embedded image processing on FPGAs.* John Wiley & Sons, 2011.

[15] Kresimir Matkovic, Laszlo Neumann, and Werner Purgathofer. A survey of tone mapping techniques. *esc*, 1:1, 1997.

[16] Kate Devlin. A review of tone reproduction techniques. *Computer Science, University of Bristol, Tech. Rep. CSTR-02-005*, 2002.

[17] Frédéric Drago, William L Martens, Karol Myszkowski, and Hans-Peter Seidel. Perceptual evaluation of tone mapping operators with regard to similarity and preference. *MAX PLANCK INSTITUT FUR INFORMATIK-REPORT-MPI I*, (2), 2002.

[18] Martin Čadík, Michael Wimmer, Laszlo Neumann, and Alessandro Artusi. Evaluation of hdr tone mapping methods using essential perceptual attributes. *Computers & Graphics*, 32(3):330–349, 2008.

[19] Yunbo Rao and Leiting Chen. A survey of video enhancement techniques. *Journal of Information Hiding and Multimedia Signal Processing*, 3(1):71–99, 2012.

[20] Gabriel Eilertsen, Jonas Unger, Robert Wanat, and Rafał Mantiuk. Survey and evaluation of tone mapping operators for hdr video. In *ACM SIGGRAPH 2013 Talks*, page 11. ACM, 2013.

[21] Gabriel Eilertsen, Rafał K Mantiuk, and Jonas Unger. A comparative review of tone-mapping algorithms for high dynamic range video. In *Computer Graphics Forum*, volume 36, pages 565–592. Wiley Online Library, 2017.

[22] Greg Ward and Rob Shakespeare. *Rendering with Radiance: the art and science of lighting visualization(2nd ed.).* CUMINCAD, San Francisco, 1998.

[23] Francois X Sillion and Claude Peuch. *Radiosity & global illumination.* CUMINCAD, San Francisco, 1994.

[24] Frédo Durand and Julie Dorsey. Interactive tone mapping. In *Rendering Techniques 2000*, pages 219–230. Springer, New York, 2000.

[25] Frédéric Dufaux, Patrick Le Callet, Rafal Mantiuk, and Marta Mrak. *High dynamic range video: from acquisition, to display and applications.* Academic Press, 2016.

[26] Julia H Jungmann, Luke MacAleese, Jan Visser, Marc JJ Vrakking, and Ron MA Heeren. High dynamic range bio-molecular ion microscopy with the timepix detector. *Analytical chemistry*, 83(20):7888–7894, 2011.

[27] Gyanesh Chander, Brian L Markham, and Dennis L Helder. Summary of current radiometric calibration coefficients for landsat mss, tm, etm+, and eo-1 ali sensors. *Remote sensing of environment*, 113(5):893–903, 2009.

[28] Richard Szeliski. *Computer vision: algorithms and applications.* Springer-Verlag, London, 2010.

[29] D-U Lee, Altaf Abdul Gaffar, Ray CC Cheung, Oskar Mencer, Wayne Luk, and George A Constantinides. Accuracy-guaranteed bit-width optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(10):1990–2000, 2006.

[30] Timo Viitanen, Pekka Jääskeläinen, Otto Esko, and Jarmo Takala. Simplified floating-point division and square root. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2707–2711. IEEE, 2013.

[31] Toshiyuki Dobashi, Atsushi Tashiro, Masahiro Iwahashi, and Hitoshi Kiya. A fixed-point implementation of tone mapping operation for hdr images expressed in floating-point format. *APSIPA Transactions on Signal and Information Processing*, 3(e11), 2014.

[32] Christoph H Lampert and Oliver Wirjadi. Anisotropic gaussian filtering using fixed point arithmetic. In *2006 International Conference on Image Processing*, pages 1565–1568. IEEE, 2006.

[33] Sample hdr photo processed with easyhdr. `https://www.easyhdr.com/examples/cap-de-formentor/`. Accessed on February 1st 2020.

[34] Muhammad Siddiqui, Muhammad Atif, and Klaus Zimmermann. Method and unit for generating high dynamic range image and video frame, July 24 2012. US Patent 8,228,392.

[35] Sony $fx - 9$ full-frame 6k sensor camera. `https://pro.sony/ue_US/products/handheld-camcorders/pxw-fx9`. Accessed on October 24, 2020.

[36] Toshio Yasue, Kohei Tomioka, Ryohei Funatsu, Tomohiro Nakamura, Takahiro Yamasaki, Hiroshi Shimamoto, Tomohiko Kosugi, Sung-Wook Jun, Takashi Watanabe, Masanori Nagase, et al. A 33mpixel cmos imager with multi-functional 3-stage pipeline adc for 480fps high-speed mode and 120fps low-noise mode. In *2018 IEEE International Solid-State Circuits Conference-(ISSCC)*, pages 90–92. IEEE, 2018.

[37] Cinematic multicam in hdr. `https://www.arri.com/en/camera-systems/cameras/amira#accordion-180920`. Accessed on October 24, 2020.

[38] High dynamic range video with hdrx. `https://www.red.com/dsmc2`. Accessed on October 24, 2020.

[39] Shigetoshi Sugawa, Nana Akahane, Satoru Adachi, Kazuya Mori, Toshiyuki Ishiuchi, and Koichi Mizobuchi. A 100 db dynamic range cmos image sensor using a lateral overflow integration capacitor. In *ISSCC. 2005 IEEE International Digest of Technical Papers. Solid-State Circuits Conference, 2005.*, pages 352–603. IEEE, 2005.

[40] Nana Akahane, Shigetoshi Sugawa, Satoru Adachi, Kazuya Mori, Toshiyuki Ishiuchi, and Koichi Mizobuchi. A sensitivity and linearity improvement of a 100-db dynamic range cmos image sensor using a lateral overflow integration capacitor. *IEEE Journal of Solid-State Circuits*, 41(4):851–858, 2006.

[41] Noriko Ide, Woonghee Lee, Nana Akahane, and Shigetoshi Sugawa. A wide dr and linear response cmos image sensor with three photocurrent integrations in photodiodes, lateral overflow capacitors, and column capacitors. *IEEE journal of solid-state circuits*, 43(7):1577–1587, 2008.

[42] S Iida, Y Sakano, T Asatsuma, M Takami, I Yoshiba, N Ohba, H Mizuno, T Oka, K Yamaguchi, A Suzuki, et al. A 0.68 e-rms random-noise 121db dynamic-range sub-pixel architecture cmos image sensor with led flicker mitigation. In *2018 IEEE International Electron Devices Meeting (IEDM)*, pages 10–2. IEEE, 2018.

[43] Yasuyuki Fujihara, Satoshi Nasuno, Shunichi Wakashima, Yusuke Aoyagi, Rihito Kuroda, and Shigetoshi Sugawa. 190–1100 nm waveband

multispectral imaging system using high light resistance wide dynamic range cmos image sensor. In *2016 IEEE SENSORS*, pages 1–3. IEEE, 2016.

[44] Sony imx490 1/1.55-type 5.44megapixels cmos image sensor. `https://www.sony-semicon.co.jp/products/common/pdf/IMX490.pdf`. Accessed on November 11, 2020.

[45] Spyros Kavadias, Bart Dierickx, Danny Scheffer, Andre Alaerts, Dirk Uwaerts, and Jan Bogaerts. A logarithmic response cmos image sensor with on-chip calibration. *IEEE Journal of Solid-state circuits*, 35(8):1146–1152, 2000.

[46] Markus Loose, Karlheinz Meier, and Johannes Schemmel. A self-calibrating single-chip cmos camera with logarithmic response. *IEEE Journal of Solid-state circuits*, 36(4):586–596, 2001.

[47] Christopher Young, Alex Omid-Zohoor, Pedram Lajevardi, and Boris Murmann. A data-compressive 1.5/2.75-bit log-gradient qvga image sensor with multi-scale readout for always-on object detection. *IEEE Journal of Solid-State Circuits*, 54(11):2932–2946, 2019.

[48] Multi-camera smartphones: Benefits and challenges. `https://www.dxomark.com/multi-camera-smartphones-benefits-and-challenges/`. Accessed on October 24, 2020.

[49] Asheer Kasar Bachoo. Real-time exposure fusion on a mobile computer. In *20th Annual Symposium of the Pattern Recognition Association of South Africa*, pages 111–115, New York, 2009. ACM.

[50] Ilija Popadić, Branislav M Todorović, and Irini Reljin. Method for hdr-like imaging using industrial digital cameras. *Multimedia Tools and Applications*, 76(10):12801–12817, 2017.

[51] Shutter speed. `https://en.wikipedia.org/wiki/Shutter_speed`. 5 December 2019.

[52] Jacob Telleen, Anne Sullivan, Jerry Yee, Oliver Wang, Prabath Gunawardane, Ian Collins, and James Davis. Synthetic shutter speed imaging. In *Computer Graphics Forum*, volume 26, pages 591–598. Wiley Online Library, 2007.

[53] S. Mann and R. W. Picard. On being ‘undigital‘ with digital cameras: Extending dynamic range by combining differently exposed pictures. In *PROCEEDINGS OF IS&T*, pages 442–448, 1995.

[54] Tomoo Mitsunaga and Shree K Nayar. Radiometric self calibration. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 1, pages 374–380. IEEE, 1999.

[55] Paul E Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH 2008 classes*, page 31. ACM, 2008.

[56] Michael D Tocci, Chris Kiser, Nora Tocci, and Pradeep Sen. A versatile hdr video production system. In *ACM Transactions on Graphics (TOG)*, volume 30, page 41. ACM, 2011.

[57] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. Photographic tone reproduction for digital images. In *ACM transactions on graphics (TOG)*, volume 21, pages 267–276. ACM, 2002.

[58] Nolan Goodnight, Rui Wang, Cliff Woolley, and Greg Humphreys. Interactive time-dependent tone mapping using programmable graphics hardware. In *ACM SIGGRAPH 2005 Courses*, page 180. ACM, 2005.

[59] Grzegorz Krawczyk, Karol Myszkowski, and Hans-Peter Seidel. Perceptual effects in real-time tone mapping. In *Proceedings of the 21st spring conference on Computer graphics*, pages 195–202. ACM, 2005.

[60] Hanli Zhao, Xiaogang Jin, and Jianbing Shen. Real-time tone mapping for high-resolution hdr images. In *2008 International Conference on Cyberworlds*, pages 256–262. IEEE, 2008.

[61] Firas Hassan and Joan E Carletta. An fpga-based architecture for a local tone-mapping operator. *Journal of Real-Time Image Processing*, 2(4):293–308, 2007.

[62] Shervin Vakili, Diana C Gil, JM Pierre Langlois, Yvon Savaria, and Guy Bois. Customized embedded processor design for global photographic tone mapping. In *2011 18th IEEE International Conference on Electronics, Circuits, and Systems*, pages 382–385. IEEE, 2011.

[63] Chris Kiser, Erik Reinhard, Mike Tocci, and Nora Tocci. Real time automated tone mapping system for hdr video. In *IEEE International*

Conference on Image Processing, volume 134, pages 2749–2752. IEEE Orlando, FL, 2012.

[64] Pierre-Jean Lapray, Barthélémy Heyrman, Matthieu Rossé, and Dominique Ginhac. A 1.3 megapixel fpga-based smart camera for high dynamic range real time video. In *2013 Seventh International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–6. IEEE, 2013.

[65] Luiz Carlos Gouveia, Waqas Mughal, and Bhaskar Choubey. A reconfigurable cmos pixel for applying tone mapping on high dynamic range images. In *2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, pages 1098–1101. IEEE, 2014.

[66] Waqas Mughal and Bhaskar Choubey. Fixed pattern noise correction for wide dynamic range cmos image sensor with reinhard tone mapping operator. In *2015 Nordic Circuits and Systems Conference (NORCAS): NORCHIP & International Symposium on System-on-Chip (SoC)*, pages 1–4. IEEE, 2015.

[67] Lan Shi, Martin Berger, Bastian Bier, Christopher Soell, Juergen Roeber, Rebecca Fahrig, Bjoern Eskofier, Andreas Maier, and Jennifer Maier. Analog non-linear transformation-based tone mapping for image enhancement in c-arm ct. In *2016 IEEE Nuclear Science Symposium, Medical Imaging Conference and Room-Temperature Semiconductor Detector Workshop (NSS/MIC/RTSD)*, pages 1–3. IEEE, 2016.

[68] Lan Shi, David Hadlich, Christopher Soell, Thomas Ussmueller, and Robert Weigel. A tone mapping algorithm suited for analog-signal real-time image processing. In *2016 12th Conference on Ph. D. Research in Microelectronics and Electronics (PRIME)*, pages 1–4. IEEE, 2016.

[69] Tsun-Hsien Wang, Wei-Su Wong, Fang-Chu Chen, and Ching-Te Chiu. Design and implementation of a real-time global tone mapping processor for high dynamic range video. In *2007 IEEE International Conference on Image Processing*, volume 6, pages VI–209. IEEE, 2007.

[70] Sonia Vargas-Sierra, Gustavo Liñán-Cembrano, and Ángel Rodríguez-Vázquez. A 151 db high dynamic range cmos image sensor chip architecture with tone mapping compression embedded in-pixel. *IEEE Sensors Journal*, 15(1):180–195, 2014.

[71] Chi-Yi Tsai and Chih-Hung Huang. Real-time implementation of an adaptive simultaneous dynamic range compression and local contrast

enhancement algorithm on a gpu. *Journal of Real-Time Image Processing*, 16(2):321–337, 2019.

[72] Tommaso Balercia, Andrea Zitti, Henry Francesconi, Simone Orcioni, and Massimo Conti. Fpga implementations of a simplified retinex image processing algorithm. In *2006 13th IEEE International Conference on Electronics, Circuits and Systems*, pages 176–179. IEEE, 2006.

[73] Chryssanthi Iakovidou, Vassilios Vonikakis, and Ioannis Andreadis. Fpga implementation of a real-time biologically inspired image enhancement algorithm. *Journal of Real-Time Image Processing*, 3(4):269–287, 2008.

[74] Steve Mann, Raymond Chun Hing Lo, Kalin Ovtcharov, Shixiang Gu, David Dai, Calvin Ngan, and Tao Ai. Realtime hdr (high dynamic range) video for eyetap wearable computers, fpga-based seeing aids, and glasseyes (eyetaps). In *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–6. IEEE, 2012.

[75] Chika Ofili, Stanislav Glozman, and Orly Yadid-Pecht. Hardware implementation of an automatic rendering tone mapping algorithm for a wide dynamic range display. *Journal of Low Power Electronics and Applications*, 3(4):337–367, 2013.

[76] Yeu-Horng Shiau, Pei-Yin Chen, Hung-Yu Yang, and Shang-Yuan Li. A low-cost hardware architecture for illumination adjustment in real-time applications. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):934–946, 2014.

[77] Shih-An Li and Chi-Yi Tsai. Low-cost and high-speed hardware implementation of contrast-preserving image dynamic range compression for full-hd video enhancement. *IET Image Processing*, 9(8):605–614, 2015.

[78] Prasoon Ambalathankandy, Alain Horé, and Orly Yadid-Pecht. An fpga implementation of a tone mapping algorithm with a halo-reducing filter. *Journal of Real-Time Image Processing*, 16(4):1317–1333, 2019.

[79] P Zemčík, P Musil, and M Musil. Real-time hdr video processing and compression using an fpga. In *High Dynamic Range Video*, pages 145–154. Elsevier, 2017.

[80] Jin Woo Park, Hyokeun Lee, Boyeal Kim, Dong-Goo Kang, Seung Oh Jin, Hyun Kim, and Hyuk-Jae Lee. A low-cost and high-throughput fpga implementation of the retinex algorithm for real-time video enhancement. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2019.

[81] Gilles Sicard, Hassan Abbas, Hawraa Amhaz, Hakim Zimouche, Robin Rolland, and David Alleysson. A cmos hdr imager with an analog local adaptation. In *Int. Image Sensor Workshop (IISW'13)*, pages 1–4, 2013.

[82] Waqas Mughal, Luiz Carlos Gouveia, and Bhaskar Choubey. On threshold comparing biomorphic image sensors. In *BICA*, pages 140–145, 2014.

[83] Peng Chen, Kartikeya Murari, and Orly Yadid-Pecht. Analog current mode implementation of global and local tone mapping algorithm for wdr image display. *Electronic Imaging*, 2016(12):1–5, 2016.

[84] Jiang Duan, Marco Bressan, Chris Dance, and Guoping Qiu. Tone-mapping high dynamic range images by novel histogram adjustment. *Pattern Recognition*, 43(5):1847–1862, 2010.

[85] Qiyuan Tian, Jiang Duan, and Guoping Qiu. Gpu-accelerated local tone-mapping for high dynamic range images. In *2012 19th IEEE International Conference on Image Processing*, pages 377–380. IEEE, 2012.

[86] Pierre-Jean Lapray, Barthélémy Heyrman, Matthieu Rossé, and Dominique Ginhac. Smart camera design for realtime high dynamic range imaging. In *2011 Fifth ACM/IEEE International Conference on Distributed Smart Cameras*, pages 1–7. IEEE, 2011.

[87] Pierre-Jean Lapray, Barthélémy Heyrman, Matthieu Rossé, and Dominique Ginhac. Hdr-artist: High dynamic range advanced real-time imaging system. In *2012 IEEE International Symposium on Circuits and Systems*, pages 1428–1431. IEEE, 2012.

[88] Pierre-Jean Lapray, Barthélémy Heyrman, and Dominique Ginhac. Hdr-artist: an adaptive real-time smart camera for high dynamic range imaging. *Journal of Real-Time Image Processing*, 12(4):747–762, 2016.

[89] Sang-Seol Lee, Eunchong Lee, Youngbae Hwang, and Sung-Joon Jang. Hardware implementation of fast high dynamic range processor for

real-time 4k uhd video. In *2016 International SoC Design Conference (ISOCC)*, pages 309–310. IEEE, 2016.

[90] Frédéric Drago, Karol Myszkowski, Thomas Annen, and Norishige Chiba. Adaptive logarithmic mapping for displaying high contrast scenes. In *Computer graphics forum*, volume 22, pages 419–426. Wiley Online Library, 2003.

[91] Vladan Popovic, Elieva Pignat, and Yusuf Leblebici. Performance optimization and fpga implementation of real-time tone mapping. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 61(10):803–807, 2014.

[92] Vladan Popovic, Kerem Seyid, Eliéva Pignat, Ömer Çogal, and Yusuf Leblebici. Multi-camera platform for panoramic real-time hdr video construction and rendering. *Journal of Real-Time Image Processing*, 12(4):697–708, 2016.

[93] Michael Ashikhmin. A tone mapping algorithm for high contrast images. In *Proceedings of the 13th Eurographics workshop on Rendering*, pages 145–156. Eurographics Association, 2002.

[94] Benjamin Roch, Alessandro Artusi, Despina Michael, Yiorgos Chrysanthou, and Alan Chalmers. Interactive local tone mapping operator with the support of graphics hardware. In *Proceedings of the 23rd Spring Conference on Computer Graphics*, pages 213–218. ACM, 2007.

[95] J Yang, A Hore, and O Yadid-Pecht. Local tone mapping algorithm and hardware implementation. *Electronics Letters*, 54(9):560–562, 2018.

[96] Jorge Andrés Palacios, Vincenzo Caro, Miguel Durán, and Miguel Figueroa. A hardware architecture for multiscale retinex with chromacity preservation on an fpga. In *2020 23rd Euromicro Conference on Digital System Design (DSD)*, pages 73–80. IEEE, 2020.

[97] Mohamed Akil, Thierry Grandpierre, and Laurent Perroton. Real-time dynamic tone-mapping operator on gpu. *Journal of Real-Time Image Processing*, 7(3):165–172, 2012.

[98] Raquel Ureña, Christian Morillas, and Francisco J Pelayo. Real-time bio-inspired contrast enhancement on gpu. *Neurocomputing*, 121:40–52, 2013.

[99] Ishtiaq Rasool Khan, Susanto Rahardja, Muhammad Murtaza Khan, Muhammad Mobeen Movania, and Fidaa Abed. A tone-mapping technique based on histogram using a sensitivity model of the human visual system. *IEEE Transactions on Industrial Electronics*, 65(4):3469–3479, 2017.

[100] Kuo-Feng Liao and Yarsun Hsu. Accelerate local tone mapping for high dynamic range images using opencl with gpu. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*, pages 141–147. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2017.

[101] P Martínez Cañada, C Morillas, Raquel Ureña, JM Gómez López, and Francisco J Pelayo. Embedded system for contrast enhancement in low-vision. *Journal of Systems Architecture*, 59(1):30–38, 2013.

[102] Jing Li, Orit Skorka, Kamal Ranaweera, and Dileepan Joseph. Novel real-time tone-mapping operator for noisy logarithmic cmos image sensors. *Electronic Imaging*, 2016(12):1–13, 2016.

[103] Jie Yang, Ulian Shahnivich, and Orly Yadid-Pecht. Mantissa-exponent based tone mapping for wide dynamic range image sensors. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2019.

[104] Prasoon Ambalathankandy, Masayuki Ikebe, Takayuki Yoshida, Takeshi Shimada, Shinya Takamaeda, Masato Motomura, and Tetsuya Asai. An adaptive global and local tone mapping algorithm implemented on fpga. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[105] Jorge Fernández-Berni, Ricardo Carmona-Galán, and Ángel Rodríguez-Vázquez. Single-exposure hdr technique based on tunable balance between local and global adaptation. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 63(5):488–492, 2015.

[106] Amal Punchihewa, Takayuki Hamamoto, and Takahiro Kojima. From a review of hdr sensing and tone compression to a novel imaging approach. In *2011 Fifth International Conference on Sensing Technology*, pages 40–46. IEEE, 2011.

[107] Frédo Durand and Julie Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. In *ACM transactions on graphics (TOG)*, volume 21, pages 257–266. ACM, 2002.

[108] Svetozar Nosko, Martin Musil, Petr Musil, and Pavel Zemcik. True hdr camera with bilateral filter based tone mapping. In *Proceedings of the 33rd Spring Conference on Computer Graphics*, page 15. ACM New York, NY, USA, 2017.

[109] Svetozar Nosko, Martin Musil, Pavel Zemcik, and Roman Juranek. Color hdr video processing architecture for smart camera. *Journal of Real-Time Image Processing*, pages 1–12, 2018.

[110] Stefano Marsi, Gaetano Impoco, Anna Ukovich, Sergio Carrato, and Giovanni Ramponi. Video enhancement and dynamic range control of hdr sequences for automotive applications. *EURASIP Journal on Advances in Signal Processing*, 2007(1):080971, 2007.

[111] Xiaoqiang Xiang, Lili Liu, Luying Que, Conghan Jia, Bo Yan, Yongjie Li, Jinhong Guo, and Jun Zhou. A biological retina inspired tone mapping processor for high-speed and energy-efficient image enhancement. *Sensors*, 20(19):5600, 2020.

[112] Lili Liu, Xiaoqiang Xiang, Yuxiang Xie, Yongjie Li, Bo Yan, and Jun Zhou. A high throughput and energy-efficient retina-inspired tone mapping processor. In *2019 IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pages 321–321. IEEE, 2019.

[113] Raanan Fattal, Dani Lischinski, and Michael Werman. Gradient domain high dynamic range compression. In *ACM transactions on graphics (TOG)*, volume 21, pages 249–256. ACM, 2002.

[114] Lavanya Vytla, Firas Hassan, and Joan E Carletta. A real-time implementation of gradient domain high dynamic range compression using a local poisson solver. *Journal of Real-Time Image Processing*, 8(2):153–167, 2013.

[115] Jian Liu, Firas Hassan, and Joan E Carletta. A study of hardware-friendly methods for gradient domain tone mapping of high dynamic range images. *Journal of Real-Time Image Processing*, 12(1):165–181, 2016.

[116] Ching-Te Chiu, Tsun-Hsien Wang, Wei-Ming Ke, Chen-Yu Chuang, Jhih-Siao Huang, Wei-Su Wong, Ren-Song Tsay, and Cyuan-Jhe Wu. Real-time tone-mapping processor with integrated photographic and gradient compression using 0.13 μm technology on an arm soc platform. *Journal of Signal Processing Systems*, 64(1):93–107, 2011.

[117] Gregory Ward Larson, Holly Rushmeier, and Christine Piatko. A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Transactions on Visualization and Computer Graphics*, 3(4):291–306, 1997.

[118] Helge Seetzen, Wolfgang Heidrich, Wolfgang Stuerzlinger, Greg Ward, Lorne Whitehead, Matthew Trentacoste, Abhijeet Ghosh, and Andrejs Vorozcovs. High dynamic range display systems. *ACM transactions on graphics (TOG)*, 23(3):760–768, 2004.

[119] Kate Devlin1 Alan Chalmers1 Alexander Wilkie and Werner Purgathofer. Tone reproduction and physically based spectral rendering. Eurographics, 2002.

[120] Karol Myszkowski, Rafal Mantiuk, and Grzegorz Krawczyk. High dynamic range video. *Synthesis Lectures on Computer Graphics and Animation*, 1(1):1–158, 2008.

[121] Marius Herscovitz and Orly Yadid-Pecht. A modified multi scale retinex algorithm with an improved global impressionof brightness for wide dynamic range pictures. *Machine Vision and Applications*, 15(4):220–228, 2004.

[122] Laurence Meylan and Sabine Susstrunk. High dynamic range image rendering with a retinex-based adaptive filter. *IEEE Transactions on image processing*, 15(9):2820–2830, 2006.

[123] Jack Tumblin and Greg Turk. Lcis: A boundary hierarchy for detail-preserving contrast reduction. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 83–90, 1999.

[124] Prasun Choudhury and Jack Tumblin. The trilateral filter for high contrast images and meshes. In *ACM SIGGRAPH 2005 Courses*, pages 5–es. 2005.

[125] Zeev Farbman, Raanan Fattal, Dani Lischinski, and Richard Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Transactions on Graphics (TOG)*, 27(3):1–10, 2008.

[126] Sylvain Paris, Samuel W Hasinoff, and Jan Kautz. Local laplacian filters: Edge-aware image processing with a laplacian pyramid. *ACM Trans. Graph.*, 30(4):68, 2011.

[127] Arrigo Benedetti, Andrea Prati, and Nello Scarabottolo. Image convolution on fpgas: the implementation of a multi-fpga fifo structure. In *Proceedings. 24th EUROMICRO Conference (Cat. No. 98EX204)*, volume 1, pages 123–130. IEEE, 1998.

[128] Ahmet Oguz Akyuz and Erik Reinhard. Color appearance in high-dynamic-range imaging. *Journal of Electronic Imaging*, 15(3):033001, 2006.

[129] Jiangtao Kuang, Garrett M Johnson, and Mark D Fairchild. icam06: A refined image appearance model for hdr image rendering. *Journal of Visual Communication and Image Representation*, 18(5):406–414, 2007.

[130] Min H Kim, Tim Weyrich, and Jan Kautz. Modeling human color perception under extended luminance levels. In *ACM SIGGRAPH 2009 papers*, pages 1–9. 2009.

[131] Laurence Meylan, David Alleysson, and Sabine Süsstrunk. Model of retinal local adaptation for the tone mapping of color filter array images. *JOSA A*, 24(9):2807–2816, 2007.

[132] David Alleysson, Sabine Susstrunk, and Jeanny Hérault. Linear demosaicing inspired by the human visual system. *IEEE Transactions on Image Processing*, 14(4):439–449, 2005.

[133] Christophe Schlick. Quantization techniques for visualization of high dynamic range pictures. In *Photorealistic rendering techniques*, pages 7–20. Springer, Berlin, 1995.

[134] Rafal Mantiuk, Grzegorz Krawczyk, Karol Myszkowski, and Hans-Peter Seidel. Perception-motivated high dynamic range video encoding. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 733–741. ACM, 2004.

[135] A Ardeshir Goshtasby. Fusion of multi-exposure images. *Image and Vision Computing*, 23(6):611–618, 2005.

[136] Piti Irawan, James A Ferwerda, and Stephen R Marschner. Perceptually based tone mapping of high dynamic range image streams. In *Proceedings of the 16th Eurographics conference on Rendering Techniques*, pages 231–242, 2005.

[137] Chi-Yi Tsai and Chien-Hsing Chou. A novel simultaneous dynamic range compression and local contrast enhancement algorithm for digital video cameras. *EURASIP Journal on Image and Video Processing*, 2011(1):6, 2011.

[138] Sumanta N Pattanaik, Jack Tumblin, Hector Yee, and Donald P Greenberg. Time-dependent visual adaptation for fast realistic image display. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 47–54. ACM Press/Addison-Wesley Publishing Co., 2000.

[139] Sumanta Pattanaik and Hector Yee. Adaptive gain control for high dynamic range image display. In *SCCG*, volume 2, pages 83–87. Citeseer, 2002.

[140] Mark Colbert, Erik Reinhard, and Charles E Hughes. Painting in high dynamic range. *Journal of Visual Communication and Image Representation*, 18(5):387–396, 2007.

[141] Stefano Marsi, Giovanni Ramponi, and Sergio Carrato. Image contrast enhancement using a recursive rational filter. In *2004 IEEE International Workshop on Imaging Systems and Techniques (IST)(IEEE Cat. No. 04EX896)*, pages 29–34. IEEE, 2004.

[142] Edwin H Land. Recent advances in retinex theory. *Central and peripheral mechanisms of colour vision*, 26(1):5–17, 1985.

[143] Giuliano Orsini, Giovanni Ramponi, Paola Carrai, and Riccardo Di Federico. A modified retinex for image contrast enhancement and dynamics control. In *Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429)*, volume 3, pages III–393. IEEE, 2003.

[144] Vassilios Vonikakis, Ioannis Andreadis, and Antonios Gasteratos. Fast centre–surround contrast modification. *IET Image processing*, 2(1):19–34, 2008.

[145] Mark A Robertson, Sean Borman, and Robert L Stevenson. Estimation-theoretic approach to dynamic range enhancement using multiple exposures. *Journal of Electronic Imaging*, 12(2):219–229, 2003.

[146] Mir Adnan Ali and Steve Mann. Comparametric image compositing: Computationally efficient high dynamic range imaging. In *2012 IEEE*

International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 913–916. IEEE, 2012.

[147] CA Ofili, Stanislav Glozman, and Orly Yadid-Pecht. An in-depth analysis and image quality assessment of exponent-based tone mapping algorithm. *International Journal Information Models and Analysis*, 1(3):236–250, 2012.

[148] Stephen M Pizer, E Philip Amburn, John D Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B Zimmerman, and Karel Zuiderveld. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, 39(3):355–368, 1987.

[149] Chi-Yi Tsai. A fast dynamic range compression with local contrast preservation algorithm and its application to real-time video enhancement. *IEEE transactions on multimedia*, 14(4):1140–1152, 2012.

[150] Alain Horé and Orly Yadid-Pecht. A statistical derivation of an automatic tone mapping algorithm for wide dynamic range display. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2475–2479. IEEE, 2014.

[151] Alain Horé and Orly Yadid-Pecht. A new filter for reducing halo artifacts in tone mapped images. In *2014 22nd International Conference on Pattern Recognition*, pages 889–894. IEEE, 2014.

[152] Ulian Shahnovich, Alain Hore, and Orly Yadid-Pecht. Hardware implementation of a real-time tone mapping algorithm based on a mantissa-exponent representation. In *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2210–2213. IEEE, 2016.

[153] Tom Mertens, Jan Kautz, and Frank Van Reeth. Exposure fusion. In *15th Pacific Conference on Computer Graphics and Applications (PG'07)*, pages 382–390. IEEE, 2007.

[154] Thorsten Grosch et al. Fast and robust high dynamic range image generation with camera and object movement. In *Vision, Modeling and Visualization*, 2006.

[155] Sohsuke Shimoyama, Masaki Igarashi, Masayuki Ikebe, and Junich Motohisa. Local adaptive tone mapping with composite multiple gamma functions. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 3153–3156. IEEE, 2009.

[156] Masaki Igarashi, Akira Mizuno, and Masayuki Ikebe. Accuracy improvement of histogram-based image filtering. In *2013 IEEE International Conference on Image Processing*, pages 1217–1221. IEEE, 2013.

[157] Yuta Kimura and Masayuki Ikebe. Halo control for lhe based local adaptive tone mapping. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 3911–3915. IEEE, 2015.

[158] Yonghun Shin, Soowoong Jeong, and Sangkeun Lee. Efficient naturalness restoration for non-uniform illumination images. *IET Image Processing*, 9(8):662–671, 2015.

[159] Xian-Shi Zhang and Yong-Jie Li. A retina inspired model for high dynamic range image rendering. In *International Conference on Brain Inspired Cognitive Systems*, pages 68–79, Switzerland, 2016. Springer.

[160] Ana Belén Petro, Catalina Sbert, and Jean-Michel Morel. Multiscale retinex. *Image Processing On Line*, pages 71–88, Apr. 2014.

[161] Shree K Nayar and Tomoo Mitsunaga. High dynamic range imaging: Spatially varying pixel exposures. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 472–479. IEEE, 2000.

[162] Jinwei Gu, Yasunobu Hitomi, Tomoo Mitsunaga, and Shree Nayar. Coded rolling shutter photography: Flexible space-time sampling. In *2010 IEEE International Conference on Computational Photography (ICCP)*, pages 1–8. IEEE, 2010.

[163] Masaaki Sasaki, Mitsuhito Mase, Shoji Kawahito, and Yoshiaki Tadokoro. A wide-dynamic-range cmos image sensor based on multiple short exposure-time readout with multiple-resolution column-parallel adc. *IEEE Sensors journal*, 7(1):151–158, 2007.

[164] William Guicquero, Antoine Dupret, and Pierre Vandergheynst. An algorithm architecture co-design for cmos compressive high dynamic range imaging. *IEEE Transactions on Computational Imaging*, 2(3):190–203, 2016.

[165] Arthur Spivak, Alexander Belenky, Alexander Fish, and Orly Yadid-Pecht. Wide-dynamic-range cmos image sensors—comparative performance analysis. *IEEE transactions on electron devices*, 56(11):2446–2461, 2009.

[166] Sergio Saponara, Luca Fanucci, Stefano Marsi, Giovanni Ramponi, David Kammler, and Ernst Martin Witte. Application-specific instruction-set processor for retinex-like image and video processing. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 54(7):596–600, 2007.

[167] Sergio Saponara, Luca Fanucci, Stefano Marsi, and Giovanni Ramponi. Algorithmic and architectural design for real-time and power-efficient retinex image/video processing. *Journal of real-time image processing*, 1(4):267–283, 2007.

[168] Juan Antonio Leñero-Bardallo, Ricardo Carmona-Galán, and Ángel Rodríguez-Vázquez. A wide linear dynamic range image sensor based on asynchronous self-reset and tagging of saturation events. *IEEE Journal of Solid-State Circuits*, 52(6):1605–1617, 2017.

[169] Mikko Kuhna, Mikko Nuutinen, and Pirkko Oittinen. Method for evaluating tone mapping operators for natural high dynamic range images. In *Digital Photography VII*, volume 7876, page 78760O. International Society for Optics and Photonics, 2011.

[170] A van Oppenheim, Ronald Schafer, and Thomas Stockham. Nonlinear filtering of multiplied and convolved signals. *IEEE transactions on audio and electroacoustics*, 16(3):437–466, 1968.

[171] Firas Hassan, Lavanya Vytla, and Joan E Carletta. Exploiting redundancy to solve the poisson equation using local information. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 2689–2692. IEEE, 2009.

[172] Uwe Meyer-Baese and U Meyer-Baese. *Digital signal processing with field programmable gate arrays*, volume 65. Springer, New York, 2007.

[173] Vitit Kantabutra. On hardware for computing exponential and trigonometric functions. *IEEE Transactions on Computers*, 45(3):328–339, 1996.

[174] Ping Tak Peter Tang. Table-lookup algorithms for elementary functions and their error analysis. In *Proceedings 10th IEEE Symposium on Computer Arithmetic*, pages 232–236.

[175] Mary Kim and Min Gyo Chung. Recursively separated and weighted histogram equalization for brightness preservation and contrast enhancement. *IEEE Transactions on Consumer Electronics*, 54(3):1389–1397, 2008.

[176] Alberto Boschetti, Nicola Adami, Riccardo Leonardi, and Masahiro Okuda. High dynamic range image tone mapping based on local histogram equalization. In *2010 IEEE International Conference on Multimedia and Expo*, pages 1130–1135. IEEE, 2010.

[177] Israel Koren. *Computer arithmetic algorithms.* AK Peters/CRC Press, 2001.

[178] Peter Nilsson, Ateeq Ur Rahman Shaik, Rakesh Gangarajaiah, and Erik Hertz. Hardware implementation of the exponential function using taylor series. In *2014 NORCHIP*, pages 1–4. IEEE, 2014.

[179] Alain Horé and Orly Yadid-Pecht. On the design of optimal 2d filters for efficient hardware implementations of image processing algorithms by using power-of-two terms. *Journal of Real-Time Image Processing*, 16(2):429–457, 2019.

[180] Fpga logic celss comparison. `http://ee.sharif.edu/~asic/Docs/fpga-logic-cells_V4_V5.pdf`. Apr. 22, 2019.

[181] Jiwoong Choi, Boyeal Kim, Hyun Kim, and Hyuk-Jae Lee. A high-throughput hardware accelerator for lossless compression of a ddr4 command trace. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 27(1):92–102, 2018.

[182] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[183] Hojatollah Yeganeh and Zhou Wang. Objective quality assessment of tone-mapped images. *IEEE Transactions on Image Processing*, 22(2):657–667, 2012.

[184] Changchun Shi. *Floating-point to fixed-point conversion.* University of California, Berkeley, 2004.

[185] Changchun Shi and Robert W Brodersen. An automated floating-point to fixed-point conversion methodology. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 2, pages II–529. IEEE, 2003.

[186] Mathworks:floating-point to fixed-point conversion. `https://www.mathworks.com/help/dsp/floating-point-to-fixed-point-conversion.html`. Accessed on February 24, 2020.

[187] Intel quartus prime design software hls compiler. `https://www.intel.com/content/www/us/en/software/programmable/quartus-prime/hls-compiler.html`. Accessed on February 24, 2020.

[188] Vivado design suite high-level synthesis. `https://www.xilinx.com/support/documentation/sw_manuals/xilinx2019_1/ug902-vivado-high-level-synthesis.pdf`. Accessed on February 24, 2020.

[189] Mentor catapult high-level synthesis. `https://www.mentor.com/hls-lp/catapult-high-level-synthesis/c-systemc-hls`. Accessed on February 24, 2020.

[190] Cadence stratus high-level synthesis. `https://www.cadence.com/en_US/home/tools/digital-design-and-signoff/synthesis/stratus-high-level-synthesis.html`. Accessed on February 24, 2020.

[191] Weisi Lin and C-C Jay Kuo. Perceptual visual quality metrics: A survey. *Journal of visual communication and image representation*, 22(4):297–312, 2011.

[192] Anmin Liu, Weisi Lin, and Manish Narwaria. Image quality assessment based on gradient similarity. *IEEE Transactions on Image Processing*, 21(4):1500–1512, 2011.

[193] Rafał K Mantiuk, Anna Tomaszewska, and Radosław Mantiuk. Comparison of four subjective methods for image quality assessment. In *Computer graphics forum*, volume 31, pages 2478–2491. Wiley Online Library, 2012.

[194] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369. IEEE, 2010.

[195] Zhou Wang and Alan C Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine*, 26(1):98–117, 2009.

[196] MJ McDonnell. Box-filtering techniques. *Computer Graphics and Image Processing*, 17(1):65–70, 1981.

[197] Fatih Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *2005 IEEE Computer Society Conference on*

Computer Vision and Pattern Recognition (CVPR'05), volume 1, pages 829–836. IEEE, 2005.

[198] Ronald Newbold Bracewell and Ronald N Bracewell. *The Fourier transform and its applications*, volume 31999. McGraw-Hill New York, 1986.

[199] William M Wells. Efficient synthesis of gaussian filters by cascaded uniform filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):234–239, 1986.

[200] Yu-Ping Wang and Seng Luan Lee. Scale-space derived from b-splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1040–1055, 1998.

[201] Ronan Boitard, Rémi Cozot, Dominique Thoreau, and Kadi Bouatouch. Survey of temporal brightness artifacts in video tone mapping. In *HDRi2014-Second international conference and SME workshop on HDR imaging*, volume 9, pages 1–6, 2014.

[202] Ali Kanj. *Flicker Removal and Color Correction for High Speed Videos*. PhD thesis, Universite Paris-Est Marne-la-Vallee, Marne-la-Vallee, 2017. 2017PESC1115.

[203] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[204] Clément Farabet, Berin Martini, Polina Akselrod, Selçuk Talay, Yann LeCun, and Eugenio Culurciello. Hardware accelerated convolutional neural networks for synthetic vision systems. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 257–260. IEEE, 2010.

[205] Chen Zhang, Peng Li, Guangyu Sun, Yijin Guan, Bingjun Xiao, and Jason Cong. Optimizing fpga-based accelerator design for deep convolutional neural networks. In *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 161–170, 2015.

[206] Michaël Gharbi, Jiawen Chen, Jonathan T Barron, Samuel W Hasinoff, and Frédo Durand. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (TOG)*, 36(4):1–12, 2017.

[207] Intel easic devices. https://www.intel.com/content/www/us/en/products/programmable/asic/easic-devices.html. Accessed on January 2, 2020.

List of publications

Journal

1. **Y. Ou**, P. Ambalathankandy, R. Furuya, S. Kawada, T. Zeng, Y. An, T. Kamishima, K. Tamura, and M. Ikebe, ”A Sub-pixel Accurate Quantification of Joint Space Narrowing Progression in Rheumatoid Arthritis,” in *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 1, pp. 53-64, Jan. 2023, DOI: 10.1109/JBHI.2022.3217685. (Impact Factor: 7.021)

2. **Y. Ou**, P. Ambalathankandy, S. Takamaeda, M. Motomura, T. Asai and M. Ikebe, ”Real-Time Tone Mapping: A Survey and Cross - Implementation Hardware Benchmark,” in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 5, pp. 2666-2686, May 2022, DOI: 10.1109/TCSVT.2021.3060143. (Impact Factor: 5.859)

3. **Y. Ou**, P. Ambalathankandy, R. Furuya, T. Kamishima, and M. Ikebe, ”Joint Angle Correction in Rheumatoid Arthritis and Its Reliability Analysis Based on Phase Dispersion Quantification” in *Open Journal of Medical Imaging*, vol. 12, no. 4, pp, 171-179, 2022, DOI: 10.4236/ojmi.2022.124018.

4. T. Okino, **Y. Ou**, M. Ikebe, K. Tamura, K. Sutherland, J. Fukae, K. Tanimura, T. Kamishima, ”Fully automatic software for detecting radiographic joint space narrowing progression in rheumatoid arthritis: phantom study and comparison with visual assessment” in *Japanese Journal of Radiology*, 2022, DOI: 10.1007/s11604-022-01373-z. (in press)

5. T. Okino, **Y. Ou**, M. Ikebe, A. Furusaki, A. Sagawa, M. Kato, T. Atsumi, and T. Kamishima, ”Automatic Subpixel Measurement of Radiographic Finger Joint Space Narrwing IN Rheumatoid Arthritis Patients under Tocilizumab Treatment,” in *Annals of the Rheumatic Dis-*

eases, vol. 81, pp. 1770-1771, 2022, DOI: 10.1136/annrheumdis-2022-eular.276.

6. P. Ambalathankandy, **Y. Ou**, and M. Ikebe, ”Warm-cool color-based high-speed decolorization: an empirical approach for tone mapping applications” in *Journal of electronic imaging*, vol. 30, no. 4, pp. 043026, August 2021, DOI: 10.1117/1.JEI.30.4.043026.

7. M. Ikebe, P. Ambalathankandy, and **Y. Ou**, ”HDR Tone mapping: System Implementations and Benchmarking” in *ITE Transactions on Media Technology and Applications*, vol. 10, no. 2, pp. 27-51, 2022, DOI: 10.3169/mta.10.27.

8. A. Taguchi, S. Shishido, **Y. Ou**, M. Ikebe, T. Zeng, W. Fang, K. Murakami, T. Ueda, N. Yasojima, K. Sato, K. Tamura, K. Sutherland, N. Oki, K. Chiba, K. Minowa, M. Uetani and T. Kamishima, ”Quantification of Joint Space Width Difference on Radiography Via Phase-Only Correlation (POC) Analysis: a Phantom Study Comparing with Various Tomographical Modalities Using Conventional Margin-Contouring,” in *Journal of Digital Imaging*, vol. 34, no. 1, pp. 96–104, 2021, DOI: 10.1007/s10278-020-00406-1.

9. C. Hou, Y. Ishi, H. Motegi, M. Okamoto, **Y. Ou**, J. Chen, and S. Yamaguchi, ”Overexpression of CD44 is associated with a poor prognosis in grade II/III gliomas,” in *Journal of Neuro-Oncology*, vol. 145, no. 2, pp. 201–210, September 2019, DOI: 10.1007/s11060-019-03288-8.

Conference

1. **Y. Ou**, P. Ambalathankandy, R. Furuya, S. Kawada, T, Kamishima, and M. Ikebe, ”Joint space narrowing progression quantification with joint angle correction in rheumatoid arthritis” in *The 14th Biomedical Engineering International Conference (BMEiCON2022)*, 10 November 2022, IEEE, Songkhla, Thailand.

2. **Y. Ou**, P. Ambalathankandy, T. Shimada T, T. Kamishima, and M. Ikebe, ”Automatic Radiographic Quantification of Joint Space Narrowing Progression in Rheumatoid Arthritis Using POC,” In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, 8 April 2019, pp. 1183-1187, IEEE, Venice, Italy.

3. P. Ambalathankandy, **Y. Ou**, S. Kaneko and M. Ikebe, ”Perceptual evaluation of decolorization algorithms to study subjectively appealing color contrast information,” in *European Conference on Visual Perception (ECVP 2022)*, 28 August 2022, Nijmegen, Netherlands.

4. P. Ambalathankandy, **Y. Ou**, J, Kochiyil, S. Takamaeda, M. Motomura, T. Asai, and M. Ikebe, ”Radiography Contrast Enhancement: Smoothed LHE Filter a Practical Solution for Digital X-Rays with Mach Band,” In *2019 Digital Image Computing: Techniques and Applications (DICTA)*, 2 December 2019, pp. 1-8, IEEE, Perth, Australia.

5. Y. Huang, G. Mao, M. Liu, and **Y. Ou**, ”Multi-focus Image Fusion Combined with CNN and Algebraic Multi-grid Method,” In *International Conference of Pioneering Computer Scientists, Engineers and Educators*, 20 September 2019, pp. 120-129, Springer, Singapore.