IIEEJ Transactions on Image Electronics and Visual Computing

Vol. 13, No. 2 2025

The Institute of Image Electronics Engineers of Japan

Editorial Committee of IIEEJ

Contact Information

The Institute of Image Electronics Engineers of Japan (IIEEJ)
3-35-4-101, Arakawa, Arakawa-ku, Tokyo 116-0002, Japan
Tel : +81-3-5615-2893　Fax : +81-3-5615-2894
E-mail : hensyu@iieej.org
http://www.iieej.org/ (in Japanese)
http://www.iieej.org/en/ (in English)
http://www.facebook.com/IIEEJ (in Japanese)
http://www.facebook.com/IIEEJ.E (in English)

IIEEJ Transactions on
Image Electronics and Visual Computing
Vol.13 No.2 December 2025

CONTENTS

Published two times a year by the Institute of Image Electronics Engineers of Japan（IIEEJ）
3-35-4-101，Arakawa，Arakawa-ku，Tokyo 116-0002，Japan
Tel：+81-5615-2893 Fax：+81-5615-2894 E-mail：hensyu@iieej.org http://www.iieej.org/

IIEEJ Invited Paper

JPEG 2000: Its History of 25 Years and the Future Prospect

Fumitaka ONO†, ††(*Honorary Member*), Osamu WATANABE‡

† Tokyo Polytechnic University, †† The University of Tokyo, ‡ Takushoku University

<**Summary**> This Invited Paper presents the 25 years history and the future prospect of JPEG 2000 standards. JPEG 2000 standards (ISO/IEC 15444 series) have been produced by ISO/IEC JTC 1/SC29/WG 1 and common texts are also registered as ITU-T Recommendations (ITU-T T.800 series). JPEG 2000 Part 1 was born in year 2000 and from then around 15 parts have been standardized to compose JPEG 2000 family of standards. Main components of JPEG 2000 are DWT (Discrete Wavelet Transform), EBCOT (Embedded Block Coding with Optimal Truncation) and MQ coder. In JPEG 2000, a lot of functions for image built-up, image exhibition, image processing are integrated in the coding system and various extensions needed for specific applications are provided in the family standards. The coding performance is highly increased from original JPEG standard and JPEG2000 has been adopted in various fields such as Digital Cinema, Document Archives around the world, and PDF specifications. The recent update is the introduction of High-Throughput JPEG 2000 (HTJ2K) and the throughput is improved by an order of magnitude at the expense of slightly reduced coding efficiency. It is expected that the birth of HTJ2K would strongly accelerate the penetration of JPEG 2000 in the near future.

Keywords: JPEG 2000, Standard, Still Image Coding, DWT, Throughput

1. Introduction

JPEG 2000 (ISO/IEC 11544 series | ITU-T T.800 series)[1), 2)] has been generated by ISO/IEC JTC 1/SC 29/WG 1, and its first standard, namely Part 1, was published on December 2000. JPEG 2000 is a grand standard composed of 15 parts at this moment and it has given a great impact on the world. In this Invited Paper, we overview the 25 years history of JPEG 2000 and prospect its future.

2. Background of the Birth of JPEG 2000

The first still image coding standard known as JPEG[3)] Part 1 (ISO/IEC 10918-1|ITU-T Rec.T.81) was published in 1992 (ITU-T) and 1994 (ISO/IEC). JPEG (Joint Photographic Experts Group) is the name of the team established to develop the standard under ISO/IEC and ITU-T. The JPEG committee then started the standardization of JPEG LS aiming the lossless coding, since the efficiency of lossless mode in JPEG standard was not so high. The Part 1 of JPEG LS[1)] (ISO/IEC 14495-1:1999 | ITU-T T.87:1998) was published with lossless and near-lossless coding capabilities.

The JPEG standard did not soon get wide acceptance in supposed application areas. But in a few years, when the framework of JPEG LS was almost fixed, rise of digital camera market, popularization of Internet, digitization trend in printing and publishing companies have made JPEG to play the central role in the era of multi-media communications including Internet.

In considering next standardization theme for JPEG team after the standardization of JPEG and JPEG LS, it was recognized that JPEG covers lossy coding and JPEG LS covers lossless coding. So if one sole method which will have lossy-to-lossless coding capability and better or equivalent coding efficiency to existing standards, it will be quite useful and to seek the method will worth standardization. Actually some coding methods, proposed as the candidates of JPEG LS, such as CREW[4)] from Ricoh Co. Ltd., had such possibilities and various desired functions not covered by existing standards were also raised to be included in the expected standard. For example, JBIG standard (ISO/IEC 11544 | ITU-T T.82) is a spin-off product from JPEG project to encode documents, and to unite the coding method of natural images and that of documents was also the theme to be solved for long time.

The committee has decided to start the discussion of new expected standard, and it was named “JPEG 2000” which shows the new era and it would also mean a kind of almighty standard in the future. JPEG 2000 may be expressed as J2K hereafter.

3. Procedure toward the Publication of J2K-1

The New Work Item Proposal of JPEG 2000 was issued in June 1996, and Call for Technical Contribution was issued in March 1997, and the submission due was set to November 1997. In the submission of the proposal, in addition to the algorithm, the decoded copy of six different bitrates (from 0.0625 to 2.0 bit per sample) of test image set was requested to be attached, and the coded data amount of lossless compression was also requested.

The number of proposed methods submitted to November 1997 meeting was 24 and the committee started to evaluate the proposed methods, where image quality evaluation was done by both subjective and objective ways. In the evaluation taken in November 1997 meeting at Sydney, DWT (Discrete Wavelet Transform) was adopted instead of DCT (Discrete Cosine Transform) used in JPEG, and WTCQ (Wavelet Trellis Coded Quantization) proposed by UA (University of Arizona) and SAIC (Science Applications

International Corp.) was selected as Base Method. Then Core Experiments were conducted in terms of desired features and complexity of algorithm. In the next meeting of March 1998, it was decided to create JPEG 2000 VM (Verification Model) which would lead to a reference implementation of JPEG 2000. In the succeeding meetings, various experiments and discussion were took place and the main components of JPEG 2000 were defined as DWT, EBCOT[1] (Embedded Block Coding with Optimized Truncation) for Part 1, TCQ[5] (Trellis Coded Quantization) for Part 2, and MQ Coder[1] for the arithmetic coding. By the way, Part 1 was supposed to be Royalty and fee free and technologies not declared as free were moved to other Parts.

About the ballot schedule, CD was issued in December 1999, FCD was issued in March 2000, and FDIS was issued in August 2000, and IS was published in December 2000. So, as the name shows, JPEG 2000 was published strictly within year 2000. By the way, July 2000 meeting was settled beforehand, but it was impossible to issue FDIS in July meeting. So, August meeting was additionally set up only in order to issue FDIS. The case of having two meetings about one month duration was quite exceptional.

4. Technologies of JPEG 2000

4.1 Target images and desired features[1), 2)]

JPEG 2000 was intended to create a new image coding standard satisfying following conditions.

The standard should provide low bit-rate operation with rate-distortion and subjective image quality performance superior to existing standards, without sacrificing performance at other points in the rate-distortion spectrum, incorporating at the same time many interesting features.

The target of images is as follows: For different types of still images (bi-level, gray-level, color, multi-component, etc.), with different characteristics (natural images, medical, scientific, remote sensing, text, rendered graphics, etc.) allowing different imaging models (client/server, real-time transmission, image library archival, limited buffer and bandwidth resources, etc.) preferably within a unified system.

The desired features are as follows:

- Superior low bit-rate performance: Network image transmission and remote sensing are some of the applications that need this feature.
- Lossless and lossy compression: The property of creating embedded bitstream to allow progressive lossy to lossless build-up is important. The highest quality is vital for medical images, or database transmission, and not necessary for display monitoring. In progressive transmission, reconstructing with increasing pixel accuracy or spatial resolution is essential depending on applications.
- Region-of-Interest Coding: This feature allows users to define certain ROI's in the image to be coded and transmitted with better quality and less distortion than the rest of the image.
- Random codestream access and processing: In ROI's coding, randomly accessed and/or decompressed will be important. Also, this ability would allow operations such as rotation, translation, filtering, feature extraction and scaling.
- Continuous-tone and bi-level compression: This feature will include compound documents with images and text, medical images with annotation overlays, and graphic and computer generated images with binary and near to binary regions, alpha and transparency planes, and facsimile.
- Tiling: The term 'tiling' refers to the partition of the original (source) image into rectangular non-overlapping blocks (tiles), which are compressed independently, as though they were entirely distinct images. All operations will be performed independently on the image tiles.

4.2 Main components[1), 2)]

Main components of JPEG 2000 will be DWT, EBCOT, and MQ coder. DWT is a multi-resolution transform and power of 2 decompositions are allowed in the form of dyadic decomposition. The reason of selecting DWT rather than DCT was its high performance at low bit-rate. The DWT can be irreversible or reversible by changing filters. The default type of irreversible transform in Part 1 is implemented by means of the Daubechies 9-tap/7-tap filter. The default type of reversible transformation in Part 1 is implemented by means of the 5-tap/3-tap filter.

EBCOT is the coding concept of JPEG 2000 Part 1. The detail will be given in various books[1] and papers[2] related to JPEG 2000. In EBCOT, image is divided into codeblock units and DWT coefficients of each codeblock is decomposed into bit plane data. Each bit is then classified into one of significance propagation pass, magnitude refinement pass, and cleanup pass. The binary decision is coded by arithmetic coding, and the number of contexts is merged into nine.

Concerning the entropy coder in strict sense, binary arithmetic coder named MQ coder was adopted in JPEG 2000. JPEG and JBIG adopted binary arithmetic coder named QM coder, which was invented jointly by IBM, Mitsubishi and ATT, and they were asking one time Royalty fee for using their patents. MQ coder, which is similar to QM coder but proposed by IBM and Mitsubishi is offered as Royalty and fee free. MQ coder is also adopted in JBIG2 (ISO/IEC 14492| ITU-T T.88).

4.3 Performance

JPEG 2000 fully supports resolution scalability, spatial random

access, multi-spectral and hyper-spectral content, and non-iterative precise rate control were realized. It retains high coding efficiency from lossless all the way down to bit rates on the order of less than 0.5 bits/pixel. Number of image components can be accommodated up to 16,384 components and $2^{32} \times 2^{32}$ image size, at any precision in the range of 1 to 38 bits/sample. JPEG 2000 codestreams also support random access index information that can facilitate rapid access to regions and/or resolutions of interest. Together, these features make JPEG 2000 attractive for a wide range of scientific, medical, professional and military applications, but also to consumer imaging applications as resolutions continue to expand and high dynamic range imaging gains popularity.

Some test results[6] show that JPEG 2000 file sizes are smaller than those of JPEG by 53% at 0.25bpp, 36% at 0.5 bpp, 18% at 0.75 bpp, and 11% at 1.0 bpp. The results mean that the improvement of coding performance from JPEG is larger at low bit rate, though distortion types of JPEG and JPEG 2000 at lower bit rate are fundamentally different.

4.4 Lazy mode

In EBCOT, bi-level information is encoded with arithmetic coding. So, it will cause time consuming issue to be solved in some applications. To cope this, some effort called as “selective arithmetic coding bypass” or “Lazy mode” has been done.

Arithmetic coding bypass has been introduced as an optional method to encode the fifth or lower bit plane data of significant propagation pass and magnitude refinement pass. It means to output the input data directly or output the raw data. Since the entropy of lower bit plane data becomes larger in principle, the advantage of the use of arithmetic coding will become smaller. By using Lazy mode, coding time of lossless compression will be reduced to about 80% and the compression efficiency loss was not recognized, in some experiments.

The idea of starting Lazy mode from earlier bit plane was proposed in October 2009 meeting and standardized as Part 2 AMD4::Block coder extension, in June 2012. It is reported[7] that by using this AMD, lossless compression time will be reduced to about 60%, with the compression efficiency loss of around 3%.

5. Application of JPEG 2000

It can be said that the main field of JPEG 2000 applications are newly explored area requiring high quality or specific features where JPEG standard has not been tried to use.

The most famous application of JPEG 2000 is Digital Cinema. DCI (Digital Cinema Initiative) was formed in March 2002, and selected JPEG 2000 for its compression standard in 2006. One of the main reasons to select JPEG 2000 is its hierarchical coding capability to produce 4K data by adding differential data to 2K data, since there are two image sizes of Digital Cinema theatre. It is also possible to edit and add the frames of the movie at any point since inter-frame correlation is not used in JPEG 2000. Also the fact that Royalty and fee is free for the usage of JPEG 2000 is quite important.

Table 1 Additional profiles and guidelines for digital cinema

AMD	Title
AMD2	Extended profiles for cinema and video production and archival applications
AMD3	Guidelines for digital cinema applications
AMD4	Profiles for broadcast applications
AMD5	Enhancements for digital cinema and archive profiles (additional frame rates)

Table 1 shows the list of Amendments[8] to Part 1, created to be used for Digital Cinema. Currently these amendments are incorporated into the body of Part 1 in newer edition.

JPEG 2000 has also become a de facto standard for media archival in many countries around the world. Some examples are Digital archive system in National Diet Library; Japan, National Archives of Japan. The Passport images and Drivers’ license images in Japan are also coded by JPEG 2000. JPEG 2000 is also adopted as image embedding method in PDF.

One commercial application that makes extensive use of JPX composition, JPIP interactive communication, and the scalability and accessibility features of JPEG 2000, is the WAVcam[9] wide area surveillance product[10] by Innovative Signal Analysis, Inc.

JPEG 2000 and JPIP are also deployed in astronomical content[11] such as the HiRISE (high resolution Mars imaging) and JHelioviewer (high resolution Sun images) projects, and new wide-field radio telescopes[12], such as the Square Kilometre Array (SKA), is creating unprecedented spatio-temporal views stretching back to the origin of the universe.

6. HTJ2K

To increase JPEG 2000 throughput and reduce the computational complexity of the current EBCOT, the standardization of another Part has begun. In March 2017, the new work of HTJ2K[13] began, and a CfP (Call for Proposals) was issued at the July 2017 meeting. The scope includes: (a) The improvement of throughput will not cause substantial degradation of coding efficiency, (b) Transcoding with EBCOT in Part 1 is easily possible, (c) The algorithm can be realized with low-power assumption, (d) The mixed use of the new Part and other Parts in the JPEG 2000 family of standards shall be possible.

Finally, HTJ2K[13), 14] was published in 2019, and the update of the codestream syntax in Part 1, the generation of conformance

testing of HTJ2K was also completed along the way.

This High-Throughput (HT) block coding algorithm can be used in place of EBCOT in Part 1. The HT block coding algorithm increases decoding and encoding throughput and allows mathematically lossless transcoding to and from EBCOT in Part 1. This is achieved at the expense of some loss in coding efficiency and substantial elimination of quality scalability. The HT block coding algorithm, which may be called FBCOT (F for Fast), adopts a coding pass structure like that of EBCOT in Part 1. No more than three coding passes are required for any given code-block in the final codestream, and arithmetic coding is replaced with a combination of variable length coding tools, adaptive run-length coding, known as Block type MELCODE[15], and simple bit-packing. The algorithm involves three passes: a significance propagation pass (HT SigProp coding pass), a magnitude refinement pass (HT MagRef coding pass), and a cleanup pass (HT Cleanup coding pass).

The HT MagRef coding pass is identical to that of EBCOT in Part 1, operating in the bypass mode, except that code bits are packed into bytes in little-endian order. That is, the first code bit in a byte appears in its LSB, as opposed to its MSB.

The HT SigProp coding pass is also very similar to that of EBCOT in Part 1, operating in the BYPASS mode, with the following two differences:

- Code bits are again packed into bytes of the raw bitstream with a little-endian bit order, instead of big-endian bit packing order; and
- The significance bits associated with a set of four stripe columns are emitted first, followed by the associated sign bits, before advancing to the next set of stripe columns, instead of inserting any required sign bit immediately after the same sample's magnitude bit.

The HT Cleanup coding pass, however, is significantly different from that of EBCOT in Part 1, and most of this new Part is devoted to its description.

Aside from the block coding algorithm itself and the parsing of packet headers, the HT block coding algorithm preserves the syntax and semantics of other Parts of the codestream specified in Part 1. This allows mathematically lossless transcoding to/from legacy JPEG 2000.

HTJ2K speeds up JPEG 2000 by an order of magnitude at the expense of a slight reduction in coding efficiency. HTJ2K retains JPEG 2000's advanced features, with reduced quality scalability, while being faster and much more efficient than traditional JPEG (ISO/IEC 10918-1). This is achieved by replacing EBCOT of Part 1 with FBCOT for today's vectorized computing architectures.

Optimized throughputs of HTJ2K software and GPU implementations are comparable to, if not higher than, those achievable with the original, and much less functional, JPEG-1 algorithm. For example, 4K 4:4:4 36-bit/channel videos can be encoded to 2 bits/pixel at more than 90 fps and decoded at more than 140 fps on a 3.4GHz 4-core Skylake desktop processor, while a mid-range GTX 1080 GPU can decode the identical content at around 500 fps. Moreover, ultra-low-latency and high-throughput encoding can be achieved with an FPGA-based hardware implementation. The expected latency is a few dozen lines. This ultra-low latency enables the hardware encoder to achieve "sub-frame latency," an important feature for streaming applications.

HTJ2K preserves all features of JPEG 2000 Part 1, except for quality scalability. It is compatible with the extensions defined by JPEG 2000 Part 2, the interactive communication protocols defined by JPEG 2000 Part 9, and other parts of the JPEG 2000 family. It fully supports resolution scalability, efficient spatial random access, multi-spectral and hyper-spectral content, and high-throughput non-iterative precise rate control. It retains high coding efficiency from lossless compression down to bit rates of 0.5 bits/pixel. Unlike J2K-1, the HT coder is not fully embedded, and hence, quality scalability is largely sacrificed.

HTJ2K[16] codestreams can, however, preserve quality layer boundaries originally in J2K-1 codestreams by enabling truly reversible transcoding between HTJ2K and J2K-1. HTJ2K addresses the needs of professional video capture, editing, streaming, and contribution markets, including high-throughput lossless coding for high-precision and half-float content. HTJ2K simultaneously addresses a broad range of applications from point-to-point streaming to extremely efficient image capture, preview, and browsing. At the same time, high energy efficiency makes it an excellent candidate for mobile and satellite imaging applications. Finally, the accompanying JPH file format updates JPEG 2000 to support modern colour spaces, enabling HDR content with almost unlimited precision and allowing the representation of raw colour sensor data with custom colour filter array patterns. Finally, like JPEG 2000 Part 1, the HTJ2K standard is intended to be royalty-free.

7. Current Structure

As stated above, Part 1 of JPEG 2000 was issued on December 2000. The subsequent parts were also developed in parallel. For example, on December 2000, Part 2 was at FCD stage, Part 3 and Part 4 were at CD stage, Part 5 was at FCD stage. Part 6 was at WD level. Part 7 was at PDTR stage, but later it was cancelled and its content was took into Part 4. The subsequent Parts were developed later and current structure is shown in **Table 2**.

Part 1 defines the syntax of a JPEG 2000 codestream and baseline which means the minimum specification to be installed to

declare JPEG 2000 compatible. Part 1 also defines a basic file format called JP2, which allows metadata such as color space information to be provided with a JPEG 2000 codestream.

Part 2 defines codestream and file format extensions including: multi-component transformations; more flexible wavelet transform kernels and decomposition structures; alternate quantization schemes; and non-linear point transforms. The Part 2 JPX file format extends the Part 1 JP2 file format to allow: more comprehensive color space descriptions and HDR sample representations; multiple codestreams; composition, cropping, geometric transforms; rich animations; descriptive metadata; and a rich metadata set for photographic imagery.

Part 3 defines a file format for motion sequences of JPEG 2000 images, where each image is coded as an independent JPEG 2000 codestream. In original JPEG, committee missed to define official “Motion JPEG Format” and many proprietary formats were born in the world. So, this Part was born based on the lessons learned in JPEG standard.

Part 4 specifies test procedures for both encoding and decoding processes defined in Part 1, including the definition of a set of decoder compliance classes. The Part 4 test files include both bare codestreams and JP2 files.

Part 5 consists of two source code packages that implement Part 1. One is written in C and the other is in Java. They are both freely available under open-source licenses.

Part 6 defines the JPM file format for multi-page document imaging, which uses the Mixed Raster Content (MRC) model of ISO/IEC 16485. JPM is an extension of the JP2 file format defined in Part 1. Although it is a member of the JPEG 2000 family, it supports the use of many other coding or compression technologies, including JBIG2 and JPEG.

Part 8 (JPSEC) standardizes tools to ensure the security of transaction, protection of contents (IPR), and protection of technologies (IP), and to allow applications to generate, consume, and exchange JPEG 2000 secured bitstreams.

Part 9 (JPIP) defines tools for supporting incremental and selective access to imagery and metadata in a networked environment.

Part 10 (JP3D) is the volumetric extension of JPEG 2000 Part 1. It explicitly defines the notion of an extra spatial dimension (the Z-dimension), extending key JPEG 2000 concepts such as tiles, precincts and code-blocks to all three dimensions, so as to provide region-of-interest accessibility properties in 3D. Part 10 also adds support for wavelet decomposition structures that extend hierarchically in all three dimensions.

Part 11 (JPWL) defines tools and methods to achieve the efficient transmission of JPEG 2000 imagery over an error-prone wireless network. More specifically, Part 11 extends the elements in the core coding system described in Part 1 with mechanisms for error protection and correction. These extensions are backward compatible: decoders which implement Part 1 are able to skip the extensions defined in Part 11. By the way, Part 12 was “ISO base media file format” and as common to ISO/IEC 14496-12, it was withdrawn later.

Table 2 Current structure of JPEG2000

Part	Title	Year
1	Core coding system	2024(Ed.5)
2	Extensions	2023(Ed.3)
3	Motion JPEG 2000 (MJ2 or MJP2)	2007 (Ed.2)
4	Conformance	2024 (Ed.4)
5	Reference software (Free)	2021 (Ed.3)
6	Compound image file format	2013 (Ed.2)
8	JPEG 2000 Secured (JPSEC)	2023 (Ed.2)
9	Interactivity tools, APIs and protocols	2023 (Ed.2)
10	Extensions for three-dimensional data	2011 (Ed.2)
11	Wireless (JPWL)	2007 (Ed.1)
13	An Entry-level JPEG 2000 Encoder	2008 (Ed.1)
14	XML representation and reference	2013 (Ed.1)
15	High-Throughput JPEG 2000	2019 (Ed.1)
16	Enhanced encapsulation of JPEG 2000 images into ISO/IEC 14496-12	2025(Ed.3)
17	Extensions for coding of discontinuous media	2023 (Ed.1)

Part 13 defines an entry-level encoder implementation of Part 1. In image coding standard, usually definition of only decoding is needed since there is a freedom in encoding procedure. The reason why Part 13 was defined is to provide the example of parameters and suggestion for the implementation considered to be free from the fear of patents. This Part was proposed by Japan.

Part 14 specifies an XML representation of the JPEG 2000 file format and marker segments, along with methods to for accessing the internal data of a JPEG 2000 image.

As stated above, Part 15 speeds-up JPEG 2000 by an order of magnitude at the expense of slightly degraded coding efficiency. The resulting HTJ2K system retains JPEG 2000's advanced features, with reduced quality scalability, while being faster and much more efficient than traditional JPEG-1.

Part 16 specifies the encapsulation of codestreams specified in the JPEG 2000 family of Standards into file formats derived from ISO/IEC 14496-12, including the file format specified in ISO/IEC 23008-12, commonly referred to as HEIF (High Efficiency Image File Format). A revision is underway to support more flexible wrapping of all JPEG 2000 codestreams, including HTJ2K.

Part 17 specifies two different types of breakpoint components, designated as “QuadBPT” and “TriBPT” components, with

associated decoding and synthesis tools. Associated with the type of breakpoint component is a corresponding breakpoint-dependent wavelet transform, with its synthesis tools. The reconstruction procedures described in this Part produce individual sample values.

8. Conclusion

JPEG2000 was definitely produced as a state-of-the-art standard expected to spread in 21st century. As stated above, JPEG 2000 project started with very high concern since JPEG standard has already gathered wide acceptance in the digitalized society, and there were some anticipation that JPEG 2000 may spread more than original JPEG and may bring huge sales or royalty income, if their charged patents will be adopted in the standard. The number of JPEG committee members was less than 30 before the start of JPEG 2000, but it has reached to 125 in July 1999 meeting, and this is still highest record of attendants of SC29/WG 1 meeting in person. The meeting style was also forced to change after starting JPEG 2000. Plenary of WG 1 meeting and HOD (Head of Delegation) meeting were newly introduced into SC29/WG 1, and such operation continued for long years.

While JPEG 2000 Part 1 is now 25 years old, there are still rooms of utilizing its potential capability of JPEG2000 standards. The trend of increasing high resolution media and mobile multimedia communications will require the JPEG 2000's parallel processing architectures, scalability, accessibility and interactive communication capabilities. Recent standardization of HTJ2K and additional Parts to the standards have enhanced these opportunities, and the emergence of new image media types, may work to still extend the technologies of JPEG 2000 families to fit them.

References

1) D. Taubman, M. Marcellin, JPEG 2000 Image Compression Fundamentals, Standards and Practice, Kluwer Academic Publishers (2002).
2) C. Christopoulos, A. Skodras, T. Ebrahimi: "The JPEG 2000 Still Image Coding System: An Overview", IEEE Trans. on Consumer Electronics, Vol. 46, No. 4, pp. 1103–1127, November 2000 (2000).
3) W.B. Pennebaker, J. L. Mitchell, JPEG: Still Image Data Compression Standard, Springer-Verlag US (1993).
4) A. Zandi, J. D. Allen, E. L. Schwartz, M. Boliek : "CREW: Compression with Reversible Embedded Wavelets", Proc. of DCC '95 (1995).
5) T. T. Chinen, T. J. Flohr, M. W. Marcellin: "TCQ in JPEG 2000", Proc. of SPIE 4115, Applications of Digital Image Processing XXIII, (28 Dec. 2000); https://doi.org/10.1117/12.411575
6) T. Chinen, A. Chien: "WG1N1583: Visual Evaluation of JPEG 2000; Color Image Compression Performance" (2000).
7) T. Ishikawa, T. Fukuhara: "Current Status of SC29/WG 1", Journal of IIITE, Vol.67, No.7, pp.562–565 (2013).
8) T. Ishikawa: "Digital Cinema and JPEG 2000", https://www.ams.giti.waseda.ac.jp/data/pdf-files/2010VMA_ishikawa.pdf
9) D. Taubman: "JPEG 2000: Current Capabilities and Future Extensions", Journal of IIEEJ, Vol. 42, No. 3, pp. 340–344 (2013).
10) http://www.wavcam.com
11) M. W. Powell et al.: "A scalable image processing framework for gigapixel Mars and other celestial body images" Aerospace Conference, March 2010.
12) D. Muller et al: "JHelioviewer: visualizing large sets of solar images using JPEG 2000", Computing in Science and Engineering, 11(5), pp. 38–47 (2009).
13) WG1 N87018 "High Throughput JPEG 2000 (HTJ2K) and the JPH file format: a primer"; https:// ds.jpeg.org/whitepapers/jpeg-htj2k-whitepaper.pdf
14) O. Watanabe, D. Taubman: "A Matlab Implementation of the Emerging HTJ2K Standard", 2019 IEEE 8th Global Conference on Consumer Electronics (GCCE), Osaka, Japan, 2019, pp. 491–495, doi: 10.1109/GCCE46687.2019.9015602.
15) F. Ono, S. Kino, M. Yoshida, T. Kimura: "Bi-level Image Coding with MELCODE – Comparison of Block Type Code and Arithmetic Type Code –" 7.6.1–7.6.6, Globecom '89 (Nov. 1989).
16) P. -A. Lemieux, D. Taubman: "RTP Payload Format for JPEG 2000 Streaming with Sub-Codestream Latency", IETF, https://www.rfc-editor.org/rfc/rfc9828.pdf, Aug. 2025.

Fumitaka ONO (*Honorary Member*)

He has received his BE, ME and Ph.D. from The University of Tokyo respectively. He has been with Mitsubishi Electric Corp., and then with Tokyo Polytechnic University. His interested area covers image coding, entropy coding, and image processing. He is IEEE Life Fellow, IEICE Fellow and IIEEJ Fellow. He has been engaged in the international standardization work since 1985, and had been ISO/IEC JTC 1/SC29/WG1 JBIG Rapporteur. He has received Award from Ministry of Education and Science, Award from Ministry of Industry and Trade, SCAT Chairman's Grand Award, Contribution Award from IPSJ/ITSCJ, and so on. He is currently Professor Emeritus of TPU, and Visiting Researcher of The University of Tokyo.

Osamu WATANABE

He received the B.S., M.S., and Ph.D. degrees from Tokyo Metropolitan University, in 1999, 2001, and 2004, respectively. In 2004, he joined the Faculty of Takushoku University as a Research Associate of electronics and computer systems, where he has been a Professor with the Department of Electronics and Computer Systems, Since 2021. His research interests include image coding and image processing. He is a member of IEICE and ITE. He received the IPSJ/ITSCJ Standardization Contribution Award, in 2021. He is also an Expert of ISO/IEC/JTC1/SG29/WG1 Committee "Joint Photographic Experts Group (JPEG)." Since October 2020, he has been the JPEG Image Coding and Quality Group Co-Chair. He is an Associate Editor of ITE Journal.

IIEEJ Paper

Parallel-Line Detection and Attitude Estimation from Omnidirectional Images

Hideki KOMAGATA† (*Member*)

†Tokyo University of Information Sciences

<Summary> Omnidirectional (360°) cameras are being increasingly used for recording and presenting a wide range of scenes such as street views, dashboard cameras, real estate literature introductions, and drone imagery. As most of these scenes are captured using fisheye lenses with large distortion, omnidirectional cameras are not commonly used for three-dimensional shape measurement or image processing. Although various conventional methods have addressed these limitations by applying perspective projection transformation to 360° images, this transformation increases the processing time. A fundamental challenge in image recognition is the detection of straight and parallel lines. Accordingly, I propose a method to detect these lines directly from 360° images without using perspective projection transformation and then estimate the axial direction and camera attitude of the detected parallel lines. Through experiments, I compared the proposed method with conventional line detection using perspective projection and demonstrated a short processing time of the proposed method and accurate detection of parallel lines. The processing times of the proposed method were approximately 10 and 30 ms per frame for cameras with 2K and 4K resolutions, respectively. In addition, the detection accuracy of parallel lines was 87% when excluding environment factors. The attitude estimation errors were approximately 11° and 5° for 2K and 4K resolutions, respectively.

Keywords: omnidirectional camera, 360° camera, camera calibration, parallel-line detection, attitude estimation

1. Introduction

Omnidirectional (360°) and fisheye cameras are widely used for purposes such as dashboard cameras, immersive video capture, and drone photography[1),2)]. In addition, because these cameras can capture a wide range of scenes, they can be applicable to the introduction of real estate properties, reconstruction of buildings and natural structures in three-dimensional (3D) space[3)-5)], and visual aids for robot navigation systems[6),7)].

A fisheye camera is constructed with a specially processed lens that can capture images covering approximately 180° on a hemisphere. However, it cannot capture 360° images using only lens processing. Thus, Nayar developed an omnidirectional camera by attaching two fisheye lenses back-to-back[8)]. Alternatively, an omnidirectional view can be obtained using a general perspective projection camera and hyperbolic mirror[9)-11)]. This approach based on a hyperbolic mirror relies on a single image, and thus it does not require stitching the boundaries of two or more images. However, its mechanism is complicated, and its structure impedes capturing the area along the optical axis. Currently, dual-fisheye cameras are commercially available and becoming widely used[1)]. Thus, this study was focused on dual-fisheye cameras.

While omnidirectional and fisheye cameras allow to capture a wide range of scenes at once, they produce large imaging distortions. Many computer vision algorithms assume no distortion, hindering their application to omnidirectional or fisheye cameras. This is a major obstacle when measurements for tasks such as 3D reconstruction of the interior of buildings or robot navigation. Therefore, most existing studies have adopted a method of calibrating the internal camera parameters in advance and performing perspective projection image transformation[12)]. However, this transformation requires coordinate calculations for all pixels, being computationally expensive. In addition, it stretches the periphery and reduces the resolution, which may affect subsequent processing. In particular, the computation time can be problematic when implementing the transformation in small devices such as autonomous mobile robots or drones.

A basic challenging task in feature detection for computer vision is the detection of straight and parallel lines. For perspective projection images, the Hough transform[13)] is a common step, but its application is difficult for omnidirectional images because straight lines are projected as curved lines.

I propose a method to directly detect straight and parallel lines in an omnidirectional image without requiring the perspective projection transformation. The proposed method does not require coordinate transformation for every pixel,

(a) Dual-fisheye image (b) Equirectangular image (c) Cube mapping image

Fig. 1 Omnidirectional (360°) images

thereby reducing the calculation time. In addition, most straight lines in a real environment, especially inside a building, are either horizontal or vertical. Accordingly, I estimate the camera attitude from a group of parallel lines captured by photographing the inside of a building, and verify the accuracy of attitude estimation. The attitude is tentatively estimated in the first frame. From the second frame, it is updated based on the attitude estimated in the previous frame. Attitude estimation is important in robot navigation, drone control, and other applications.

The contributions of this study are summarized as follows:

i. I propose a method to directly detect parallel lines on an omnidirectional image without requiring a perspective projection transformation.
ii. I propose an additional method to tentatively estimate the camera attitude from a set of parallel lines captured in the first frame.
iii. The attitude of the current frame is updated considering the set of parallel lines appearing in the second and subsequent frames, as well as the attitude estimated in the previous frame.

Furthermore, the assumptions and limitations of the proposed methods are described as follows:

i. All frames in an omnidirectional image must include a Manhattan pattern, containing three orthogonal dominant directions that are often satisfied by man-made structures[14]. Specifically, parallel-lines must appear along two or more orthogonal axes (***X***, ***Y***, and ***Z*** axes) in the captured image.
ii. The number of lines that are not parallel to any axis must be less than that of lines parallel to the ***X***, ***Y***, or ***Z*** axes.
iii. The displacement between two adjacent frames must be small.

2. Related Work

Figure 1(a) shows an example of a dual-fisheye image captured using the RICOH THETA V omnidirectional camera, which has two fisheye lenses mounted back-to-back. The camera is manufactured by RICOH Corporation. Although the dual-fisheye image is the original image, it does not provide intuitive information. Therefore, panoramic conversion using equirectangular projection is often used, as shown in Fig.1 (b). Some omnidirectional cameras, including RICOH THETA V, can provide images in either the dual-fisheye or equirectangular format.

In computer vision, a calibration chart with a known shape can be placed within the scene before acquisition, and the camera position and attitude can be measured based on the points on the chart[15]. However, this method depends on the chart to be placed in advance. Conversely, visual simultaneous localization and mapping (SLAM)[16], which allows to calculate the camera self-position and pose simultaneously based on 3D reconstruction from multiple captured image frames, has become widely used in recent years. Visual SLAM is performed by detecting corners, called key points, and matching them across frames. However, visual SLAM cannot be directly applied to omnidirectional images including dual-fisheye and equirectangular images because the corners in real space are distorted. Therefore, Caruso et al.[17] converted hemispherical images captured by a fisheye camera into five perspective projection images and applied visual SLAM. Similarly, Kato et al.[18] and Wang et al.[19] cube-mapped an omnidirectional image onto six perspective projection images, as shown in Fig.1(c), to then apply visual SLAM. However, the abovementioned methods require coordinate calculations of perspective projection for all pixels, increasing the computational cost.

In addition to visual SLAM, detecting straight and parallel lines is an important task in computer vision. For example, Kawanishi et al.[20] performed 3D reconstruction using parallel lines acquired from a hyperbolic camera in a textureless environment. Pumarola et al.[21], Fu et al.[22], and Lee et al.[23] improved the accuracy by adding parallel lines to estimation using conventional point-cloud-based SLAM. Furthermore, line detection is widely used for real-time lane detection in driving assistance systems[24), 25]. Hence, straight and parallel-line features are expected to be useful for a variety of purposes.

For general perspective projection images, the Hough transform[13] or probabilistic Hough transform[26] is often used to detect lines. However, this method cannot be directly applied to omnidirectional images, because straight lines are projected as curves, hindering voting under the Hough transform. Okuda et al.[27] performed a perspective projection transformation on part of an omnidirectional image, detected lines using Canny edge detection[28] and the probabilistic Hough transform[26], and classified the detected lines into three types of parallel lines depending on whether they were within ±10° of the horizontal, ±10° of the vertical, or in another orientation in an image. In addition, Zhang et al.[29] detected vanishing points by fitting straight lines in the real space to conic sections in the omnidirectional image. Then, they performed a perspective projection transformation of the detected points on the conic section based on vanishing points and detected parallel lines by voting randomly sampled points. Their method restricts a perspective projection transformation to points on conic sections, which reduces computational cost. However it uses voting to detect lines, which can take a little longer to process.

3. Proposed Method

To avoid the high computational cost of the conventional method described in Chapter 2, I propose a method to directly detect straight and parallel lines in an omnidirectional image without requiring a perspective projection transformation or voting. In addition, I estimate the camera attitude from the obtained parallel lines. **Figure 2** shows a flowchart of the proposed method, which is detailed in Sections 3.1–3.7.

3.1 Initial settings for internal camera parameters

Each fisheye lens used in omnidirectional cameras is generally designed to establish an equidistant projection model in which image height $\boldsymbol{r}$ projected onto the camera imaging surface and angle θ from the optical axis have the following proportional relation:

$$r = f\theta, \tag{1}$$

where $\boldsymbol{f}$ is the focal length of the lens. Using Eq. (1), the camera model that expresses the relation between any point coordinates $\boldsymbol{S} = [S_u, S_v]$ on an image and unit direction vector $\boldsymbol{V}$ of that point in the camera coordinate system is given by[30]

$$\boldsymbol{V} = \hat{\boldsymbol{V}}/|\hat{\boldsymbol{V}}|, \tag{2}$$

$$\hat{\boldsymbol{V}} = \begin{bmatrix} \hat{u} \\ \hat{v} \\ \hat{r}/\tan\hat{r} \end{bmatrix}, \tag{3}$$

$$\hat{r} = \sqrt{\hat{u}^2 + \hat{v}^2}, \tag{4}$$

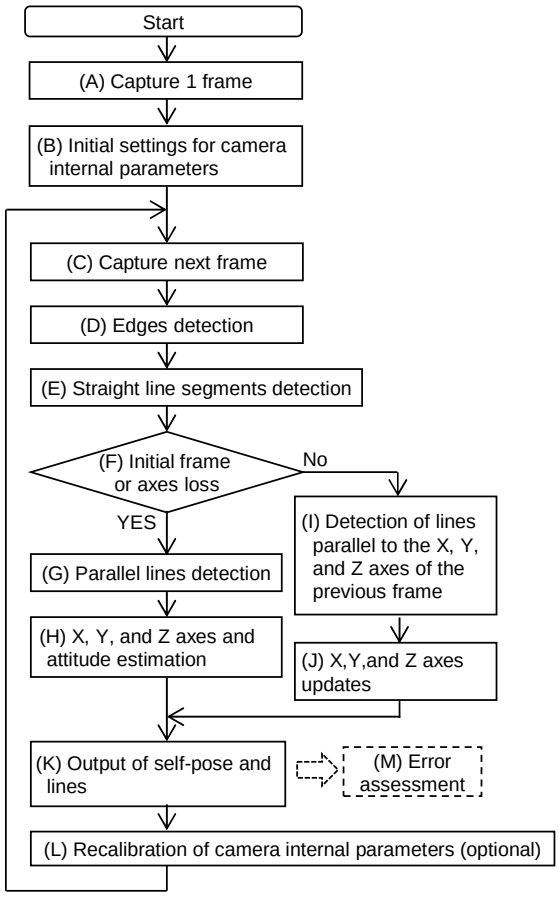


Fig. 2 Flowchart of proposed method

$$\begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix} = \begin{bmatrix} (S_u - C_u)/f_u \\ -(S_v - C_v)/f_v \end{bmatrix}, \tag{5}$$

where $\boldsymbol{C} = [C_u, C_v]$ represents the optical center coordinates on the image, and f_u and f_v are conversion coefficients between pixels and distance. By convention, the origin of the image coordinate system is set to the upper-left corner, the $\boldsymbol{u}$ axis in the image coordinate system is parallel to the $\boldsymbol{X}$ axis of the camera coordinate system, and the $\boldsymbol{v}$ axis of the image coordinate system is symmetrical to the $\boldsymbol{Y}$ axis of the camera coordinate system. Moreover, the optical axis is set as the $\boldsymbol{Z}$ axis of the camera coordinate system. The detailed derivation of these equations is provided in **Appendix A**.

Variables f_u, f_v, C_u, and C_v are the internal camera parameters. For circular fisheye lenses (Fig. 1(a)), the optical-axis center coordinate, $[C_u, C_v]$, is approximately the central coordinate of each circle. Assuming that each fisheye lens has a 180° angle of view, the following equation holds:

$$f_u \fallingdotseq f_v \fallingdotseq l/\pi, \tag{6}$$

where $\boldsymbol{l}$ is the diameter (in pixels) of each outer circle on the image.

In this study, as illustrated in Fig.1 (a), one frame is captured with the omnidirectional camera, and procedure (B)

described in Fig.2 is separately performed as follows for the left and right fisheye images:

i) To detect only the outer circle, the captured image is binarized using a low threshold. In the experiments reported in Chapter 4, the binarization threshold is 30 in an intensity range of 0–255.

ii) Only the outermost contour is detected using a contour detection method[31].

iii) The detected contour is fitted to an ellipse[32]. The center coordinate of the ellipse is $\boldsymbol{C}$, and its major axis is $\boldsymbol{l}$.

v) Parameters f_u and f_v are calculated using Eq. (6).

The results are considered as the initial values of the internal camera parameters.

3.2 Edge detection

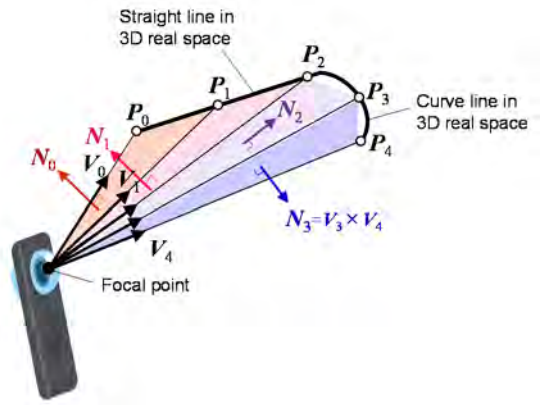
In this Section, I explain procedures (C) and (D) from Fig.2. First, take a frame of a scene that contains parallel lines. Next, the captured 360° color image is converted into a grayscale image, which is used for edge detection. The widely used Y value (256 brightness levels) of the Y-Cb-Cr color space is used for grayscale conversion, and Canny edge detection[28] is adopted. Canny edge detection requires adjustment of the detection sensitivity using hysteresis thresholds T_1 and T_2 and smoothing parameter σ. In the experiments of this study, these parameters were empirically set to $T_1 = 50$, $T_2 = 150$, and $\sigma = 3$. Nevertheless, they are influenced by the scene environment and type of camera.

The edges are then labeled using contour detection[31] and divided into line segments. In this case, outer circles, as those shown in Fig.1(a), are also detected as edges. Therefore, a predefined mask image is used to filter edge labels located near the outer circles.

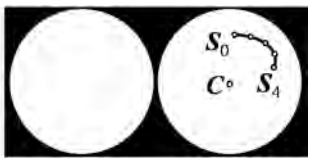
3.3 Straight line detection

In this Section, I explain procedure (E) from Fig.2. The edges detected in Section 3.2 are not necessarily straight lines segments in 3D space but may be curved instead. Only straight lines should be detected from the extracted edges. For distortion-free perspective images, the Hough transform[13] is often used to detect straight lines. However, for an omnidirectional image, straight lines are projection-distorted, as shown in Fig.1(a), and the Hough transform cannot be applied directly. Therefore, I propose a method to detect straight lines in 3D space.

Figure 3(a) illustrates a model comprising a straight line (between $\boldsymbol{P}_0$ and $\boldsymbol{P}_2$) and curved line (between $\boldsymbol{P}_2$ and $\boldsymbol{P}_4$) in 3D space, and Fig.3(b) shows their projected images,



(a) Lines and curves in 3D space



(b) Projection of image in panel (a)

Fig. 3 Straight line detection

$\boldsymbol{S}_i$ $(i \in \{0,1,2,3,4\})$, in the omnidirectional image. In the camera coordinate system, the unit direction vector from the camera focal point to $\boldsymbol{P}_i$ $(i \in \{0,1,2,3,4\})$ is denoted as $\boldsymbol{V_i}$, and the unit normal vector of the plane consisting of the focal point, $\boldsymbol{P}_i$, and point $\boldsymbol{P}_{i+1}$ is denoted as $\boldsymbol{N}_i$. The relation between $\boldsymbol{V_i}$ and $\boldsymbol{S}_i$ is given by Eqs. (2)–(5). Vector $\boldsymbol{V_i}$ can be calculated using the coordinate values from the edge detected in Section 3.2 and internal camera parameters. Vector $\boldsymbol{N}_i$ can be calculated as follows:

$$\boldsymbol{N}_i = \frac{\boldsymbol{V}_i \times \boldsymbol{V}_{i+1}}{|\boldsymbol{V}_i \times \boldsymbol{V}_{i+1}|}, \tag{7}$$

where $\times$ denotes the cross product.

Vector $\boldsymbol{V}_i$ has an error because $\boldsymbol{S}_i$ depends on the camera resolution and lenses do not necessarily comply with Eqs. (2)–(5). If $\boldsymbol{V}_i$ and $\boldsymbol{V}_{i+1}$ are close, this has a large effect on the accuracy of $\boldsymbol{N}_i$. Therefore, instead of calculating $\boldsymbol{S}_i$ for all pixels on the edge, I select $\boldsymbol{S}_i$ at regular intervals of T_3 pixels and calculate $\boldsymbol{V_i}$ and $\boldsymbol{N}_i$.

If there is a straight line in the 3D real space, $\boldsymbol{N}_i$ follows the same direction. In Fig.3 (a), as line $\boldsymbol{P}_0$–$\boldsymbol{P}_2$ is straight, $\boldsymbol{N}_0$ and $\boldsymbol{N}_1$ are equal. However, in actual calculations, the error in $\boldsymbol{N}_i$ leads to consider the angle between adjacent planes as error angle α_i calculated as follows:

$$\alpha_i = \cos^{-1}(\boldsymbol{N}_i \cdot \boldsymbol{N}_{i+1}), \tag{8}$$

where $\cdot$ denotes the inner product.

If an edge is a straight line in the 3D real space, α_i is close to 0, whereas if it is a curved line, α_i is large. In

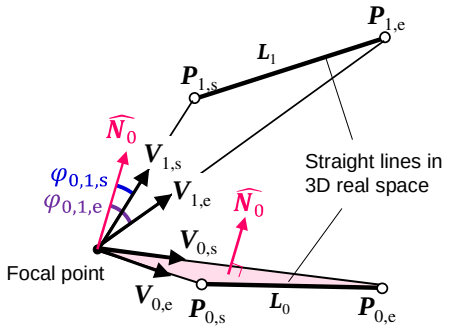


Fig. 4 Parallel line detection

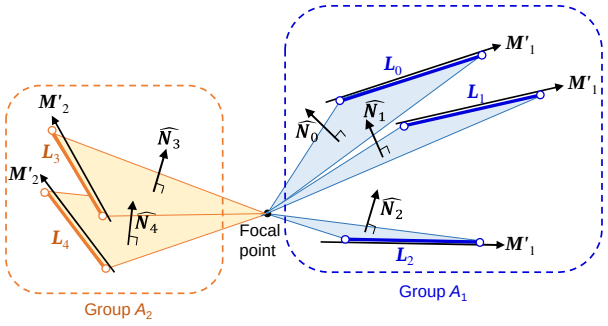


Fig. 5 Estimation of the orientation of parallel lines

addition, short straight lines may be noise, such as wall patterns. Therefore, edges where α_i is smaller than threshold T_4 are detected, and only edges longer than threshold T_5 are selected as straight lines. However, even if it is a 3D real curve, if it appears as a straight line from the field of view, α_i will be close to 0. Please note that this algorithm cannot exclude such curves. As multiple line segments may satisfy this condition, I assign index $\boldsymbol{j}$ to each line segment, and $\boldsymbol{L}_j$ represents the line segments in the 3D real space that correspond to a selected edge. In addition, the total number of elements in $\boldsymbol{L}_j$ is $\boldsymbol{m}$, and thus $j \in \{0, \dots, m-1\}$.

3.4 Parallel-line detection

In this Section, I explain procedure (G) from Fig.2. **Figure 4** shows an example of line segments $\boldsymbol{L}_j$ $(j \in \{0,1\})$ in the 3D real space. The initial and final points of $\boldsymbol{L}_j$ are denoted as $\boldsymbol{P}_{j,s}$ and $\boldsymbol{P}_{j,e}$, respectively, and the unit vectors pointing from the focal point to these points are denoted as $\boldsymbol{V}_{j,s}$ and $\boldsymbol{V}_{j,e}$, respectively. In addition, the unit normal vector of the plane formed by the focal point and $\boldsymbol{L}_j$ is denoted as $\widehat{\boldsymbol{N}_j}$ (Fig.4 only shows $\widehat{\boldsymbol{N}_0}$), while the angle between $\widehat{\boldsymbol{N}_j}$ and $\boldsymbol{V}_{a,s}$ is denoted as $\varphi_{j,a,s}$, and that between $\widehat{\boldsymbol{N}_j}$ and $\boldsymbol{V}_{a,e}$ is denoted as $\varphi_{j,a,e}$, where a is the index of the line segment to be compared with $\boldsymbol{L}_j$, where $a \in \{j+1, \dots, m-1\}$. In Fig.4, only the cases for $j=0$ and $a=1$ are shown. $\widehat{\boldsymbol{N}_j}$, $\varphi_{j,a,s}$, and $\varphi_{j,a,e}$ can be calculated as follows:

$$\widehat{\boldsymbol{N}_j} = \frac{\boldsymbol{V}_{j,s} \times \boldsymbol{V}_{j,e}}{|\boldsymbol{V}_{j,s} \times \boldsymbol{V}_{j,e}|}, \tag{9}$$

$$\varphi_{j,a,s} = \cos^{-1}\left(\widehat{\boldsymbol{N}_j} \cdot \boldsymbol{V}_{a,s}\right), \tag{10}$$

$$\varphi_{j,a,e} = \cos^{-1}\left(\widehat{\boldsymbol{N}_j} \cdot \boldsymbol{V}_{a,e}\right). \tag{11}$$

If $\boldsymbol{L}_j$ and $\boldsymbol{L}_a$ are parallel, angles $\varphi_{j,a,s}$ and $\varphi_{j,a,e}$ are equal. Even if $\boldsymbol{L}_j$ and $\boldsymbol{L}_a$ are perpendicular, $\varphi_{j,a,s}$ and $\varphi_{j,a,e}$ are 0 and thus equal. Therefore, I define the difference between these angles as

$$\varphi_{j,a} = \left|\varphi_{j,a,s} - \varphi_{j,a,e}\right| \tag{12}$$

and detect parallel lines that satisfy the following conditions:

$$\varphi_{j,a,s} > T_6 \cap \varphi_{j,a,e} > T_6 \cap \varphi_{j,a} < T_7, \tag{13}$$

where T_6 and T_7 are the corresponding angle thresholds. The line segments detected by the above method are divided into several groups. Let A_k be the group of parallel lines with the $\boldsymbol{k}$-th largest number of lines, where, $k \in \{0, \dots, \eta\}$ and η is the number of groups. Next, the orientation of the parallel lines within each group is estimated. **Figure 5** illustrates two parallel-line groups, $A_1 = \{L_0, L_1, L_2\}$ and $A_2 = \{L_3, L_4\}$. Let the directional vectors of A_1 and A_2 in the camera coordinate system of the real space be $\boldsymbol{M'}_1$ and $\boldsymbol{M'}_2$, respectively. In addition, $\widehat{\boldsymbol{N}_0} - \widehat{\boldsymbol{N}_4}$ are calculated for line segments $L_0 - L_4$ using Eq. (9). If the parallel lines are accurately detected and calculated, the following relations hold:

$$\boldsymbol{M'}_1 \parallel \widehat{\boldsymbol{N}_0} \times \widehat{\boldsymbol{N}_1}, \tag{14}$$

$$\boldsymbol{M'}_1 \parallel \widehat{\boldsymbol{N}_1} \times \widehat{\boldsymbol{N}_2}, \tag{15}$$

$$\boldsymbol{M'}_2 \parallel \widehat{\boldsymbol{N}_3} \times \widehat{\boldsymbol{N}_4}. \tag{16}$$

However, as measurements inherently contain errors, averaging is applied to mitigate their effects. Specifically, as shown in Eq. (17), unit vectors of cross products are calculated for all line segments L_j in each group A_k and averaged to estimate direction vectors $\widehat{\boldsymbol{M}_k}$ of the parallel line.

$$\widehat{\boldsymbol{M}_k} = \sum_{j=0}^{m-1} \left\{ \frac{\widehat{\boldsymbol{N}_j} \times \widehat{\boldsymbol{N}_{j+1}}}{|\widehat{\boldsymbol{N}_j} \times \widehat{\boldsymbol{N}_{j+1}}|} \;\middle|\; L_j \in A_k \right\} \tag{17}$$

Unit vectors $\boldsymbol{M}_k$ of $\widehat{\boldsymbol{M}_k}$ are given by:

$$\boldsymbol{M}_k = \widehat{\boldsymbol{M}_k} / \left|\widehat{\boldsymbol{M}_k}\right|. \tag{18}$$

3.5 Estimation of *X*, *Y*, and *Z* axes and attitude

In this Section, I explain procedure (H) from Fig.2. As illustrated in **Fig.6**, the edges of many architectural environment images often establish three orthogonal axes denoted as $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$. Therefore, I estimate the correspondence between the camera and 3D real space by assigning the parallel lines detected in Section 3.4 to the $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$, or another axis. As there is no definition for axes in real space to correspond to the $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$ axes, I assign the

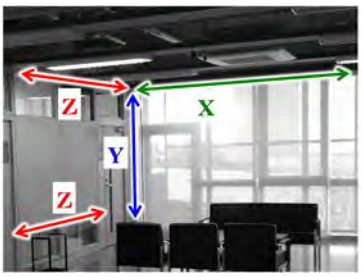


Fig. 6 ***X***, ***Y***, and ***Z*** axes in real environment

parallel lines that are close to these axes in the camera coordinate system from the first captured frame as the corresponding axes.

First, I estimate whether group A_1 of parallel lines with the greatest number of lines corresponds to the ***X***, ***Y***, or ***Z*** axis. Let the axis to be estimated be β and the unit direction vectors of the ***X***, ***Y***, and ***Z*** axes in the camera coordinate system be $\boldsymbol{I}_X = [1,0,0]$, $\boldsymbol{I}_Y = [0,1,0]$, and $\boldsymbol{I}_Z = [0,0,1]$, respectively. By calculating the inner product of $\boldsymbol{M}_1$ obtained by Eq. (18) and $\boldsymbol{I}_U$ $(U \in \{X,Y,Z\})$, the axis (***X***, ***Y***, or ***Z***) closest to A_1 can be estimated. Axis β corresponding to A_1 can be estimated as follows:

$$\beta = \underset{U\in\{X,Y,Z\}}{\arg\max}\, \boldsymbol{M}_1 \cdot \boldsymbol{I}_U. \tag{19}$$

Next, the second axis is estimated. Let γ indicate the group of parallel lines that are the most perpendicular to β and δ be the axis indicated by A_γ. Orthogonality can be calculated using an inner product. Thus, γ is given by

$$\gamma = \underset{i\in\{2..m\}}{\arg\min}\, \boldsymbol{M}_1 \cdot \boldsymbol{M}_i. \tag{20}$$

Then, like in Eq. (19), the second axis δ can be calculated as follows:

$$\delta = \underset{U\in\{X,Y,Z\}}{\arg\max}\, \boldsymbol{M}_\gamma \cdot \boldsymbol{I}_U. \tag{21}$$

Let the remaining axis be denoted as ε. If the unit direction vectors of the β, δ, and ε axes in the camera coordinate system are denoted as $\boldsymbol{W}_\beta$, $\boldsymbol{W}_\delta$, and $\boldsymbol{W}_\varepsilon$, respectively, they can be expressed as

$$\boldsymbol{W}_\beta = \boldsymbol{M}_1, \tag{22}$$

$$\boldsymbol{W}_\delta = \boldsymbol{M}_\gamma, \tag{23}$$

$$\boldsymbol{W}_\varepsilon = \frac{\boldsymbol{M}_1 \times \boldsymbol{M}_\gamma}{|\boldsymbol{M}_1 \times \boldsymbol{M}_\gamma|}. \tag{24}$$

As β, δ, or ε correspond to ***X***, ***Y***, or ***Z***, they can be collectively expressed as $\boldsymbol{W} = [\boldsymbol{W}_X, \boldsymbol{W}_Y, \boldsymbol{W}_Z]$ or $\boldsymbol{W}_U$ $(U \in \{X,Y,Z\})$.

3.6 Update of *X*, *Y*, and *Z* axes

In this Section, I explain procedures (F)–(K) from Fig.2. From the second frame onward, new axes $\widehat{\boldsymbol{W}_U}$ $(U \in \{X,Y,Z\})$ and their unit vectors $\widehat{\boldsymbol{W}_U}$ are estimated by using the previously estimated frame axes $\boldsymbol{W}$. This estimation assumes no considerable movement between the current and previous frames.

Angle threshold T_8 is defined to determine whether line segment L_i detected in Section 3.3 is close to axis $\boldsymbol{W}_U$ from the previous frame. If the set of parallel lines judged to be parallel to $\boldsymbol{W}_U$ is B_U, it can be calculated as follows:

$$\mathrm{B_U} = \left\{ L_i \;\middle|\; T_8 > \frac{\pi}{2} - \cos^{-1}(\boldsymbol{W}_U \cdot \boldsymbol{N}_i) \right\}. \tag{25}$$

Then, as in Eqs. (17) and (18), axes $\boldsymbol{W}_U$ are updated as follows:

$$\widehat{\boldsymbol{W}_U} = \sum_i \left\{ \frac{\boldsymbol{N}_i \times \boldsymbol{N}_{i+1}}{|\boldsymbol{N}_i \times \boldsymbol{N}_{i+1}|} \;\middle|\; L_i \in B_U \right\}, \tag{26}$$

$$\widehat{\boldsymbol{W}_U} = \widehat{\boldsymbol{W}_U} / |\widehat{\boldsymbol{W}_U}|, \tag{27}$$

where $\boldsymbol{W}_U$ or $\widehat{\boldsymbol{W}_U}$ is the vector that represents the camera attitude. However, conversion into angle notation is often convenient in practice. For example, converting $\widehat{\boldsymbol{W}_U}$ into Euler angles E_U $(U \in \{X,Y,Z\})$ following the ***X Y Z*** convention is given by[33]

$$E_X = \tan^{-1} \frac{-\widehat{W_{Z,y}}}{\widehat{W_{Z,z}}}, \tag{28}$$

$$E_Y = \tan^{-1} \left(\frac{\widehat{W_{Z,x}}}{\widehat{W_{Z,z}}} \cos E_X \right), \tag{29}$$

$$E_Z = \tan^{-1} \frac{-\widehat{W_{Y,x}}}{\widehat{W_{X,x}}}. \tag{30}$$

In the experiments reported in Chapter 4, error evaluation is performed using Euler angles E_U.

3.7 Calibration of internal camera parameters

This Section presents procedure (L) from Fig.2, which is optional. Many omnidirectional cameras conform to the equidistant projection model established by Eqs. (1)–(5) (Section 3.1), but this model is not necessarily accurate. Komagata et al.[30] used distortion parameters g_1, g_2, g_3, g_4, k_1, k_2, k_3, and k_4, expressed the deviation from the projection method of the fisheye lens in detail using Eqs. (2), (4), (5), (31), (32), and (33), and proposed a method to calibrate these parameters using horizontal and vertical parallel lines.

$$\hat{\boldsymbol{V}} = \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{r}/\tan\tilde{r} \end{bmatrix} \tag{31}$$

$$\tilde{r} = \sqrt{\tilde{u}^2 + \tilde{v}^2} \tag{32}$$

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} = \begin{bmatrix} g_1 + g_3 & g_2 \\ g_4 & g_3 \\ g_1 & g_2 + g_4 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \hat{u}^2 \\ \hat{u}\hat{v} \\ \hat{v}^2 \end{bmatrix} + \left(1 + \sum_{i=1}^{4} k_i \hat{r}^{2i}\right) \begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix} \tag{33}$$

I use this procedure for recalibration. However, as three axes are extracted in Section 3.6, two axes are arbitrarily selected. In addition, false detection of parallel lines reduces the calibration accuracy. Thus, in the experiments reported in Chapter 4, lines likely to be erroneous are manually removed from B_U, and only the distortion parameters are calibrated.

4. Experimental Results and Discussion

4.1 Experimental methods

I used the RICOH THETA SC and V cameras with resolutions of 1920 × 1080 and 3840 × 1920 pixels, respectively. They are dual-fisheye omnidirectional cameras and can record both still images and videos. A stereo plate (SLIK Plate II 201152) was attached to a handheld tripod, and both cameras were fixed on top of the plate in parallel. I moved with the tripod and captured five videos while walking down the hallways and stairs of a university campus. The hallways showed Manhattan patterns, but the stairs did not show these patterns. I deliberately included hallways and stairs in the videos for evaluation. The videos were shot at 29.97 fps for 29, 24, 13, 14, and 27 s, obtaining recordings of 840, 630, 420, 510, and 810 frames, respectively.

For each video captured by each camera, I extracted parallel lines and estimated their axes in each frame, following the steps in the flowchart shown in Fig.2. The corresponding thresholds were determined through trial and error in a preliminary study as follows: $T_3 = 10$ pixels, $T_4 = 5°$, $T_5 = 50$ pixels, $T_6 = T_7 = 5°$, and $T_8 = 10°$. Setting T_4–T_8 to large values and T_3 to a small value increased the detection sensitivity. Although these values should be adjusted for each camera, for experimental comparison, the same values were used for both the RICOH THETA SC and V cameras.

For the experiment, each omnidirectional camera captured two fisheye images, as shown in Fig.1(a). Euler angles E_U ($U \in \{X, Y, Z\}$) in Eqs. (28)–(30) were separately calculated for the left and right images. Let the Euler angles for the $\boldsymbol{i}$-th frame ($0 \leq \boldsymbol{i} \leq$ n) of the left image be $E_{U,L,i}$, and those of the right image be $E_{U,R,i}$. If the method detailed in Chapter 3 is accurate, $E_{U,L,i}$ and $E_{U,R,i}$ should displace simultaneously. Therefore, I define the error for the $\boldsymbol{i}$-th frame as follows:

$$D_i = \frac{1}{3}\sum_U \left| \left| E_{U,L,i} - E_{U,L,i-1} \right| - \left| E_{U,R,i} - E_{U,R,i-1} \right| \right|, \quad (34)$$

and the average error over $\boldsymbol{n}$ frames as follows:

$$e = \frac{1}{n}\sum_{i=1}^{n} D_i. \quad (35)$$

I calculated errors D_i and e for the five videos and measured the processing time per frame on a general-purpose personal computer to consider whether the proposed method was practical. To measure the processing time, I used a Windows 11 computer equipped with an Intel Core i7-11700 CPU and Micron 2210 SSD. I also created a C++ language code in Visual Studio 2022 and OpenCV 4.7.0.

Furthermore, I simulated the conventional method proposed by Okuda et al.[27] (Conv. A) and a model that approximates Conv. A to the method of Zhang et al.[29] (Conv. B) using the following procedure and conditions to quantitatively evaluate the processing time and parallel-line detection accuracy:

i) In Conv. A, perspective projection images are separately created for the left and right dual-fisheye images, as shown in Fig.1(a). The sizes of the perspective projection images during conversion are the same as the sizes of the original images (i.e., 1920 × 1080 and 3840 × 1920 pixels for RICOH THETA SC and V, respectively). In Conv. B, this process is skipped.

ii) The perspective projection images of Conv. A or the dual-fisheye images of Conv. B are represented in grayscale using the Y value of the Y-Cb-Cr color space.

iii) The edges in the grayscale images are detected using the Canny method with the same parameters as those for the proposed method that is, $T_1 = 50$, $T_2 = 150$, and $\sigma = 3$.

iv) In Conv. A, the probabilistic Hough transform is applied to the obtained edges. In Conv. B, only the obtained edges are transformed into perspective projection coordinates, and then the probabilistic Hough transform is applied. The thresholds are empirically set as follows: distance resolution of the voting space $\rho = 1$ pixel, angular resolution of the voting space $\theta = 1°$, and voting threshold $T_9 = 100$. In addition, detected line segments should be at least 50 pixels long. If the distance between the lines is within 5 pixels, the segments are considered to be connected.

v) Among the line segments detected in step iv), those within ±10° with respect to the horizontal are drawn in red on the perspective projection image obtained in step i) as horizontal parallel lines. Similarly, segments within ±10° with respect to the vertical are drawn in blue as vertical parallel lines. Segments that do not fit into any of these categories are drawn in green.

vi) A visual check is performed every 100 frames to verify the correctness of step v), and the number of lines that are successfully classified is determined. In addition, lines that failed the consecutive verification are classified as failures due to environment (e.g., detecting handrails on stairs) or other factors.

It should be noted that although Conv. B is close to the method of Zhang et al.[29], the algorithm is not identical. Similarly, using the proposed method, I visually check the line segments every 100 frames to confirm whether they are correctly classified and divide the results into three patterns: success, failure due to environment factors, and failure due to other factors. Then, I determine the number of line segments per pattern.

In step i), the fisheye image coordinates corresponding to the coordinates of the perspective projection image are calculated by applying the following equations for every pixel.

$$\begin{bmatrix} S_u \\ S_v \end{bmatrix} = \begin{bmatrix} C_u + f_u \tilde{u}\tilde{\theta}/\tilde{r} \\ C_v - f_v \tilde{v}\tilde{\theta}/\tilde{r} \end{bmatrix}, \tag{36}$$

$$\tilde{\theta} = \tan^{-1}\tilde{r}, \tag{37}$$

$$\tilde{r} = \sqrt{\tilde{u}^2 + \tilde{v}^2}, \tag{38}$$

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} = \begin{bmatrix} (Q_u - C_u')/f_u' \\ -(Q_v - C_v')/f_v' \end{bmatrix}, \tag{39}$$

where $\boldsymbol{Q} = [Q_u, Q_v]$ represents the coordinates on the perspective projection image, $[S_u, S_v]$ represents the fisheye image coordinates corresponding to $\boldsymbol{Q}$, $\tilde{r}$ is the distance from the optical center when the focal length is assumed to be 1 in the perspective projection model, and $\tilde{\theta}$ is the angle with respect to the optical axis. In addition, f_u, f_v, C_u, and C_v are the internal camera parameters for the fisheye image, and the same values of the proposed method described in Section 3.1 are used. Variables f_u', f_v', C_u', and C_v' are the internal camera parameters for the perspective projection image that can be set arbitrarily. Thus, C_u' and C_v' are set to half the image size of the perspective projection image, while f_u' and f_v' are both set to 333 to capture a wide angle. Equations. (36)–(39) correspond to the inverse transformations of Eqs. (2)–(5). In addition, as the omnidirectional images captured with the RICOH THETA SC and V cameras are input in formats tilted by 90° as shown in Fig.1(a), additional 90° rotations are added.

4.2 Experimental results for initial estimation of internal camera parameters

In this Section, I present the experimental results of procedures (B), initial settings for internal camera parameters, and (L), recalibration of internal camera parameters (optional), from Fig.2.

Table 1 Initial values of internal camera parameters

	THETA SC		THETA V	
	Left image	Right image	Left image	Right image
C_u	476	1,442	925	2,910
C_v	480	480	950	950
f_u	300	300	615	615
f_v	300	300	615	615

Table 2 Calibration results of distortion parameters

	THETA SC		THETA V	
	Left image	Right image	Left image	Right image
g_1	4.92×10^{-2}	-5.95×10^{-3}	-1.87×10^{-2}	4.83×10^{-3}
g_2	3.47×10^{-3}	1.95×10^{-2}	-3.84×10^{-2}	2.61×10^{-2}
g_3	-4.08×10^{-2}	1.50×10^{-2}	-9.05×10^{-3}	2.12×10^{-2}
g_4	-1.02×10^{-1}	-1.37×10^{-2}	4.27×10^{-2}	-2.00×10^{-2}
k_1	3.66×10^{-2}	-4.83×10^{-2}	-1.51×10^{-2}	7.84×10^{-3}
k_2	-2.10×10^{-2}	2.31×10^{-3}	6.91×10^{-3}	3.81×10^{-3}
k_3	1.76×10^{-3}	2.67×10^{-4}	2.51×10^{-3}	1.57×10^{-3}
k_4	1.81×10^{-3}	-1.06×10^{-5}	9.20×10^{-4}	3.90×10^{-4}

As described in Section 4.1, video images were captured using both the THETA SC and V cameras, and the first frame was used to calculate the initial values of the internal camera parameters separately for the left and right fisheye images using the method described in Section 3.1. The calculation results are listed in **Table 1**. Next, 20 frames were randomly selected from the video, and the distortion parameters were calibrated as described in Section 3.7. The calibration results are listed in **Table 2**.

Section 4.3 presents the comparison of the computational speed between the method including the distortion parameters and conventional method. Section 4.4 presents accuracy evaluation of the methods with and without distortion parameters.

4.3 Experimental results and discussion for calculation time using conventional method

As mentioned in Section 4.1, five videos were captured using the RICOH THETA SC and V omnidirectional cameras, and the calculations detailed in Chapter 2 (processing according to the flowchart shown in Fig.2) were performed. The conventional method described in Section 4.1 was also applied to the same videos. The calculation times (in milliseconds) required for these processes per frame are listed in **Table 3**. The columns of (a) and (d) in Table 3 show the computation times of the proposed method, and columns of (b) and (c) show the computation times of the conventional

Table 3 Calculation time in milliseconds per frame

Model and video number	Line detection time comparison			(d) Processing after (a).
	(a) Proposed.	(b) Conv. A	(c) Conv. B	
SC 1	9.778	151.960	42.394	0.083
SC 2	11.530	173.709	73.670	0.078
SC 3	11.064	171.435	73.464	0.110
SC 4	9.624	153.423	41.261	0.090
SC 5	10.718	167.268	59.753	0.101
V 1	30.351	564.851	135.220	0.101
V 2	33.429	613.213	204.874	0.109
V 3	30.147	595.923	188.312	0.156
V 4	26.126	546.961	132.574	0.068
V 5	32.559	598.625	171.627	0.109
SC Ave.	10.543	163.559	58.108	0.092
V Ave.	30.522	583.915	166.521	0.109

methods. In addition, columns (a), (b), and (c) of Tables 3 list the times required to detect straight lines, and column(d) lists the time required to detect parallel lines and estimate axes after line detection by the proposed method listed in column (a). None of the values include the time required for image input/output or drawing. The last two rows of Table 3 show the average values separately calculated for the SC and V cameras.

In Table 3, the average for the SC camera of column (a) is 1/15.5 of column (b) and 1/5.5 of column(c). The average for the V camera of column (a) is 1/19.1 of column (b) and 1/5.5 of column (c). Therefore, the proposed method can detect straight lines much faster than the conventional method.

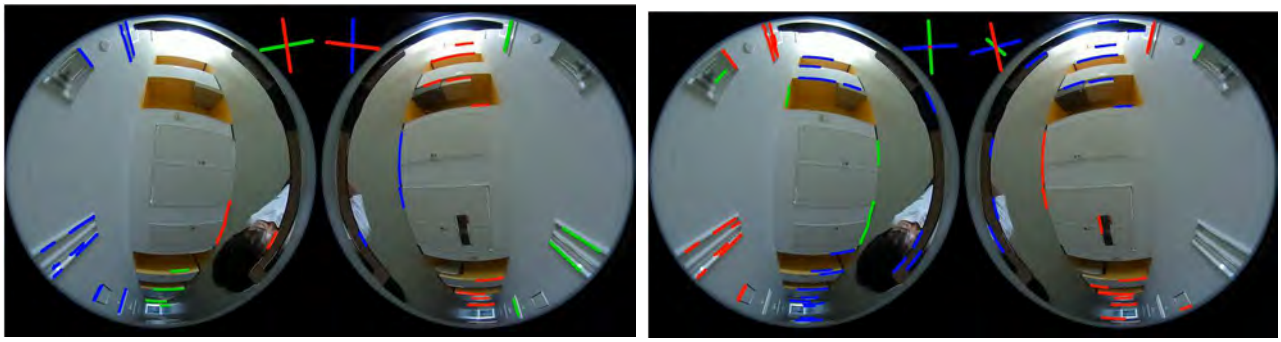
I analyze the factors that lead to the abovementioned results. The proposed method directly judges whether each line is a straight line by applying the coordinates of the detected edges to the relevant equations. Conversely, the Hough transform or its probabilistic version used in the conventional method requires voting and thresholding, which increase the calculation time. In addition, the perspective projection transformation used in Conv. A requires calculating the corresponding coordinates for every pixel in the transformed image. In other words, under the experimental conditions for the SC camera described in Section 4.1, approximately 4 million calculations are required for 1920 × 1080 pixels × 2 images (left and right) per frame. Under the experimental conditions for the V camera, approximately 15 million calculations are required for 3840 × 1920 pixels × 2 images per frame. Although the calculation time can be reduced to some extent by setting a small image size after conversion, it is still expected to take a long time in this case. The calculation time for the proposed method listed in column (a) of Table 3 is approximately 10 ms for every 2-megapixel SC image and 30 ms for every 7-megapixel V image. Therefore, the processing time required is roughly proportional to the number of pixels. In addition, as the SC and V cameras can process 100 and 30 frames per second, respectively, real-time processing is possible when using the CPU of a commercial personal computer.

The calculation time for processing after line detection of column (a) of Table 3, that is, the processing time for procedures (F)–(K) (Fig.2), is listed in column (d) of Table 3. This process takes approximately 0.1 ms, regardless of the difference between the images from the SC and V cameras. As the value is smaller than that listed in column (a) of Table 3, the processing load is negligible in most cases.

4.4 Experimental results and discussion for comparing line detection with conventional method

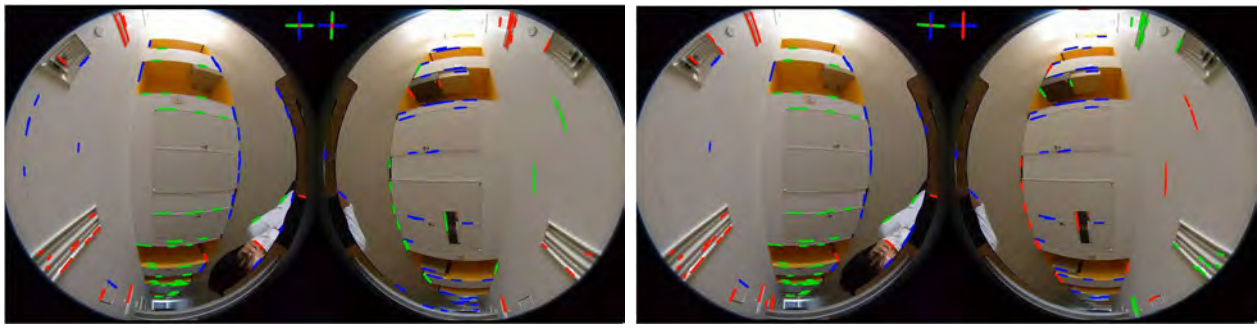
Figure 7 shows parallel-line detection and attitude estimation results obtained from the proposed method. Figures **7**(a) and (b) show results from the THETA SC camera, and Figs.**7**(c) and (d) show results from the THETA V camera. In addition, Figs.**7**(a) and (c) show results without the recalibration of the internal camera parameters described in Section 3.7 (procedure (L) from Fig.2), and Figs.**7**(b) and (d) show results with recalibration. The models in Figs. **7**(a)–(d) are called SC-c, SC-rc, V-c, and V-rc, respectively. The green, red, and blue curves in Fig.7 are determined to be parallel to the X, Y, and Z axes of the world coordinate system, respectively, and the axes on the top center of the images are the camera attitudes separately estimated for left and right images. The attitudes are also color-coded in green, red, and blue for the X, Y, and Z axes, respectively. Attitude estimation is further discussed in Section 4.5.

Figure 8 shows an example of parallel-line detection using the conventional method proposed by Okuda et al.[27] (Conv. A) involving steps i)–v) presented in Section 4.1 for the RICOH THETA SC camera. The red and blue curves indicate detection of lines parallel to the horizontal and vertical axes, respectively. The green curves represent parallel lines that are neither horizontal nor vertical. For a Manhattan image like that shown in Fig.6, green represents the Z axis. In the left peripheral area of the right-side image in Fig.8, Moiré patterns are detected using Conv. A. However, the accuracy



(a) Results without recalibration of SC images (SC-c)

(b) Results with recalibration of SC images (SC-rc)



(c) Results without recalibration of V images (V-c)

(d) Results with recalibration of V images (V-rc)

Fig. 7 Results of parallel line detection and attitude estimation using proposed method

Fig. 8 Results of parallel-line detection using conventional method

difference between Conv. A and proposed methods cannot be determined by visual inspection alone.

As described in step vi) of Section 4.1, I quantitatively compared the parallel-line detection accuracy of the proposed and conventional methods (Conv. A and Conv. B). **Table 4** represents the results for SC and V images using the proposed method, Conv. A, and Conv. B. Row (i) (Success) in Table 4 lists the number of successfully detected lines, including identification of the ***X***, ***Y***, or ***Z*** axis, while rows (ii) and (iii) list the numbers of lines that failed to be detected. Specifically, row (iii) corresponds to failures owing to environment factors such as the slope of stairs, reflection of the cameras, and Moiré patterns during perspective projection transformation. Row (ii) lists the number of false positives for factors other than those listed in row (iii). Row (iv) lists the percentage of success (row (i)) in the total number of detections, i.e., $(\mathrm{i})/\big((\mathrm{i})+(\mathrm{ii})+(\mathrm{iii})\big)$. Row (v) lists the percentage of success when environmental factors are excluded i.e., $(\mathrm{i})/\big((\mathrm{i})+(\mathrm{ii})\big)$. In addition, row (vi) shows the model averages for SC and V images from row (v) for the proposed and conventional methods.

From the results listed in rows (iv) and (v), no large difference is observed between columns (a) and (b), between columns (c) and (d), and between columns (e) and (f). Therefore, the difference in model (resolution) does not affect the parallel-line detection accuracy. However, the accuracy in row (vi) of Conv. A and Conv. B is 70.5% and 76.7%, respectively, while that of the proposed method is 87.3%. Hence, the proposed method improves the parallel-line detection accuracy by 10.6% compared with the conventional

Table 4 Results of comparative experiment on parallel-line detection accuracy

	Proposed		Conv. A		Conv. B	
	(a) SC-c	(b) V-c	(c) SC-c	(d) V-c	(e) SC-c	(f) V-c
(i) Success	809	1402	699	1347	1123	1388
(ii) Fail. (non-env.)	125	192	290	569	316	456
(iii) Fail. (env.)	200	375	282	506	372	491
(iv) Success rate 1 $\left(=\text{(i)}/\left(\text{(i)}+\text{(ii)}+\text{(iii)}\right)\right)$	71.3 %	71.2 %	55.0 %	55.6 %	62.0 %	59.4 %
(v) Success rate 2 $\left(=\text{(i)}/\left(\text{(i)}+\text{(ii)}\right)\right)$	86.6 %	88.0 %	70.7 %	70.3 %	78.0 %	75.3 %
(vi) Ave. of (v)	87.3 %		70.5 %		76.7 %	

method, thereby achieving a high accuracy. The proposed method utilizes the state of the previous frame to detect horizontal and vertical parallel lines, which may have affected the improvement in accuracy.

4.5 Experimental results and discussion for attitude estimation

In this Section, I analyze the attitude estimation accuracy of the proposed method. I measured average error $\boldsymbol{e}$ (Eq. (35)) of the attitude estimation described in Section 4.1 for the four models, namely, SC-c, SC-rc, V-c and V-rc, using the proposed method. **Table 5** lists the results of the measurements for the five videos and their averages and standard deviations (in degrees). Comparing the average values of each model in Table 5, columns (c) and (d) show approximately half the error of columns (a) and (b). The number of pixels in the RICOH THETA V camera is twice as many horizontally and approximately twice as many vertically as those in the RICOH THETA SC camera, possibly contributing to reduce errors. The average in column (b) is 6% lower than that in column (a), and the average in column (d) is 16% lower than that in column (c). Therefore, recalibration has some effect. The most accurate condition V-rc in column (d) shows an average attitude estimation error of less than 5°.

Next, the change in error per over a time series is discussed. **Figure 9** shows the results of calculating intraframe error D_i (Eq. (34)) for models SC-c, SC-rc, V-c, and V-rc and frames from 1 to 200 of video 1. The horizontal axis in Fig.9 represents the frame number, and the vertical axis represents error $\boldsymbol{D_i}$ (in degrees). The errors for all the models in Fig.9 vary, showing large fluctuations. Although model V-rc (panel (d)) has a temporary error close to 30°, the attitude estimation error remains below 3° in most frames.

To analyze the cause of this unusual error, I obtained Euler angles $\boldsymbol{E_X}$, $\boldsymbol{E_Y}$, and $\boldsymbol{E_Z}$ (Eqs. (28)–(30)) of model V-rc from frame 1 to 200 of video 1. The results are shown in **Fig.10**. As the Euler angles are calculated independently for the left and right fisheye images, two broken lines are obtained in each graph in Fig.10. In Fig.10(a), the results for the left and right images change roughly symmetrically, while in Fig.10(b), they have a similar change trend. Although almost no change is observed in Fig.10(c), around frame 145, the left image temporarily changes drastically. To investigate the underlying cause of this change, I obtained the parallel line detection and attitude estimation results for frame 145 using model V-rc. The results are shown in **Fig.11**. The left image in Fig.11 shows that the slope of the stairs is detected as a parallel line. This is a straight line that does not fit on any of the X, Y, or Z axes in Fig.6, In other words, this frame does not show a Manhattan pattern, leading to false detection caused by an attitude estimation error. Other frames also contained parallel lines that could not be detected, and these two causes were assumed to be the major sources of error.

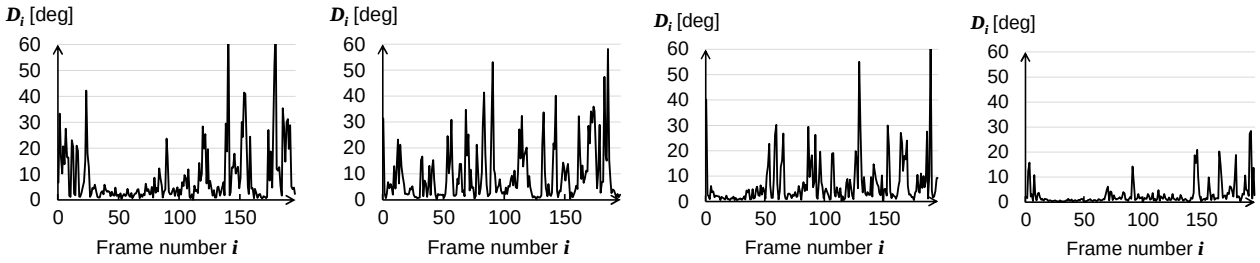
Table 5 Comparison of average error $\boldsymbol{e}$ (in degrees) of attitude estimation using proposed method

Video number	(a) SC-c	(b) SC-rc	(c) V-c	(d) V-rc
1	13.024	11.090	6.777	5.636
2	7.917	9.395	5.034	4.131
3	10.557	10.984	4.026	2.825
4	14.747	13.886	4.990	4.997
5	13.317	10.681	6.969	5.716
Ave.	11.913	11.207	5.559	4.661
SD.	2.410	1.470	1.133	1.080

Attitude estimation is difficult when the image does not show Manhattan patterns. Conversely, the proposed method can estimate the attitude with an average accuracy of less than 5° using only the camera image when Manhattan patterns appear.

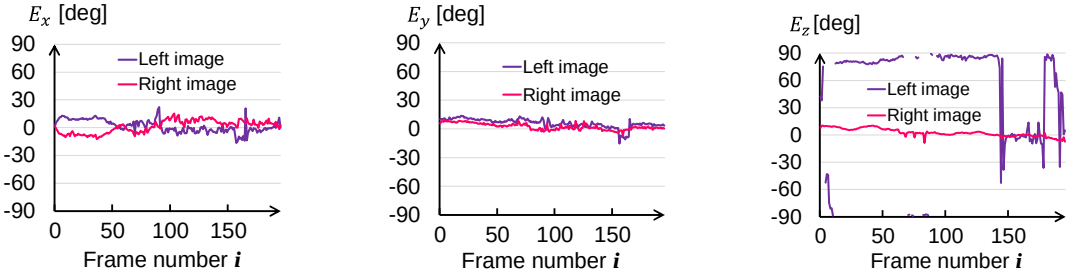
5. Conclusion

I propose a method to detect straight and parallel lines



(a) SC-c (b) SC-rc (c) V-c (d) V-rc

Fig. 9 Euler error from frames 1–200 of video 1



(a) Euler angle E_X (b) Euler angle E_Y (c) Euler angle E_Z

Fig. 10 Euler angles from frames 1–200 of video 1

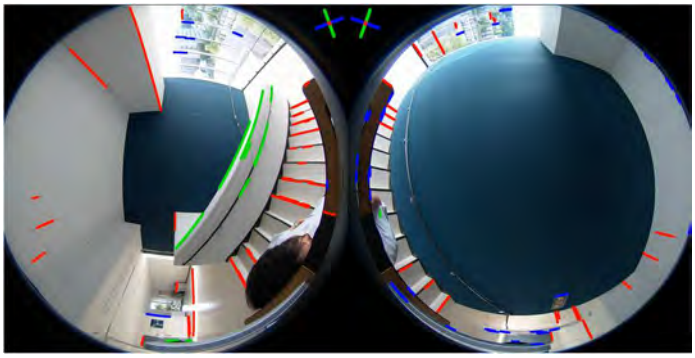
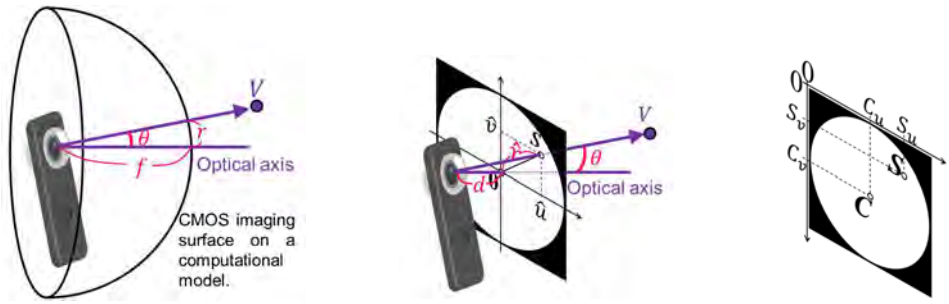


Fig. 11 Results of parallel line detection and attitude estimation using proposed V-rc model for frame 145 of video 1

directly from 360° images and estimate the attitude. Straight lines are detected from extracted edges that are fit to a fisheye projection model. Parallel-lines are detected by separating them into the ***X***-, ***Y***-, and ***Z***-axis components, assuming that the image contains Manhattan patterns. The attitude is estimated in the first frame and subsequently updated from the second frame based on the attitude of the previous frame. The proposed method can detect parallel lines with higher accuracy than conventional methods in 360° images. When environmental factors are eliminated and Manhattan patterns appear in the images, the detection accuracy of parallel lines by the proposed method is 87.3%, an improvement of 10.6% compared with the 76.7% accuracy of the conventional method. Moreover, the proposed method can detect lines much faster than conventional methods. In particular, even when using a 4K camera with resolution of 3840 × 1920 pixels, calculations can be completed in approximately 30 ms per frame using the CPU of commercial and inexpensive personal computer. In other words, even in embedded systems where it is difficult to use a graphics processor, the proposed method may enable real-time detection of parallel lines, and it may be possible to use the method in future implementations for 3D reconstruction and navigation control of robots.

Using a 3840 × 1920 pixel omnidirectional camera, attitude estimation reaches an accuracy of approximately 5°. However, the attitude estimation error may temporarily



(a) Equidistant projection model (b) Virtual surface where distance ***d*** varies (c) Digital coordinates in pixels on a 360° image

Fig. 12 Relations between real space direction and digital coordinates on a 360° image in equidistant projection camera model

increase due to non-detection or erroneous detection of parallel lines caused by compositions other than Manhattan patterns. If this factor can be eliminated algorithmically or by other means, it may be possible to use the technology as a replacement for gyro sensors used in drones and robot navigation systems. Visual SLAM is being actively researched for robot navigation as a calculation method based on point clouds. However, detecting feature points is difficult in some environments and the visual SLAM accuracy may be improved by combining it with the proposed line detection method.

References

1) K. Karkhanis: ``Complete Review of 360 Degree Cameras: A Comprehensive White Paper (July 5, 2023)'', Available at SSRN: https://ssrn.com/abstract=4501119 (2023).
2) A. Murtiyoso, P. Grussenmeyer: ``Documentation of Heritage Buildings using Close-range UAV Images: Dense Matching Issues, Comparison and Case Studies'', The Photogrammetric Record, Vol.32, No.159, pp.206–229 (2017).
3) J. Chen, Z. Wan, M. Narayana, Y. Li, W. Hutchcroft, S. Velipasalar, S. B. Kang: ``BGDNet: Background-guided Indoor Panorama Depth Estimation'', Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.1272–1281 (2024).
4) C. Chioni, A. Murtiyoso, S. Favargiotti, G. A. Massari: ``Low-Cost Workflow for 3D Urban Forest Virtual Reconstruction'', The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol.XLVIII-1/W2-2023, pp.1723–1728 (2023).
5) C. Sun, C. Hsiao, N. Wang, M. Sun, H. Chen: ``Indoor Panorama Planar 3D Reconstruction via Divide and Conquer'', Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.11338–11347 (2021).
6) F. Morbidi, D. Louise, C. S. Teodorescu, B. Fraud, E. Leblong, T. Carlson, et al. : ``Assistive Robotic Technologies for Next-Generation Smart Wheelchairs: Codesign and Modularity to Improve Users' Quality of Life'', IEEE Robotics and Automation magazine, Vol.30, No.1, pp.24–35 (2022).
7) Y. Kita, R. Takase, T. Komuro, N. Kato, N. Kita: ``Detection and Localization of Pallets on Shelves using a Wide-angle Camera'', International Conference on Advanced Robotics, pp.785–792 (2019).
8) S. K. Nayar: ``Catadioptric Omnidirectional Camera'', Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp.482–488 (1997).
9) Y. Yagi, Y. Nishizawa, M. Yachida: ``Map-based Navigation for a Mobile Robot with Omnidirectional Images Sensor COPIS'', IEEE Trans. on Robotics and Automation, Vol.11, No.5, pp.634–648 (1995).
10) T. Svoboda, T. Pajdla, V. Hlavac: ``Central Panoramic Cameras Geometry and Design'', Research report K335/97/147, Czech Technical University (1997).
11) Y. Yagi: ``Omnidirectional Sensing and Its Applications'', IEICE Trans. on Information and Systems, Vol.E82-D, No.3, pp.568–579 (1999).
12) S. Jiang, K. You, Y. Li, D. Weng, W. Chen: ``3D Reconstruction of Spherical Images: a Review of Techniques Applications and Prospects'', Geospatial Information Science, arXiv:2302.04495 (2024).
13) D. H. Ballard: ``Generalizing the Hough Transform to Detect Arbitrary Shape'', Pattern Recognition, Vol.13, No.2, pp.111–122 (1981).
14) J. Deutscher, M. Isard, J. MacCormick: ``Automatic camera calibration from a single manhattan image'', Proc. of the 7th European Conference on Computer Vision (ECCV), pp.175–188 (2002).
15) Z. Zhang: ``A Flexible New Technique for Camera Calibration'', IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.22, No.11, pp.1330–1334 (2000).
16) T. Taketomi, H. Uchiyama, S. Ikeda: ``Visual SLAM Algorithms: a Survey from 2010 to 2016'', IPSJ Trans. on Computer Vision and Applications, Vol.9, No.16 (2017).
17) D. Caruso, J. Engel, D. Cremers: ``Large-scale Direct SLAM for Omnidirectional Cameras'', 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 141–148 (2015).
18) Y. Kato, J. Hara, H. Watanabe: ``Creation of 3D Environmental Map using Omnidirectional Camera Images'', 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE), pp.450–453 (2021).

19) Y. Wang, S. Cai, S. Li, Y. Liu, Y. Guo, T. Li, M. Cheng: ``Cubemapslam: A Piecewise-Pinhole Monocular Fisheye Slam System'', Asian Conference on Computer Vision (ACCV), pp.34–49 (2018).
20) R. Kawanishi, A. Yamashita, T. Kaneko, H. Asama: ``Parallel Line-based Structure from Motion by using Omnidirectional Camera in Textureless Scene'', Advences Robotics, Vol.27. No.1, pp.19–32 (2013).
21) A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, F. Moreno-Noguew: ``PL-SLAM: Real-Time Monocular Visual SLAM with Points and Lines'', IEEE Conference on Robotics and Automation (ICRA), vol.2017, pp.4503–4508 (2017).
22) Q. Fu, J. Wang, H. Yu, I. Ali, F. Guo, Y. He, H. Zhang: ``PL-VINS: Real-Time Monocular Visual-Inertial SLAM with Point and Line Features'', arXiv:2009.07462 (2020).
23) J. Lee, S. Park: ``PLF-VINS: Real-Time Monocular Visual-Inertial SLAM With Point-Line Fusion and Parallel-Line Fusion'', IEEE Robotics and Automation Letters, Vol.6, No.4, pp.7033–7040 (2021).
24) A. B. Hillel, R. Lerner, D. Levi, G. Raz: ``Recent Progress in Road and Lane Detection: a Survey'', Machine Vision and Applications, Vol.25, No.3 (2014).
25) J. Tang, S. Li, P. Liu: ``A Review of Lane Detection Methods Based on Deep Learning'', Pattern Recognition, Vol.111 (2021).
26) J. Matas., C. Galambos, J. Kittler: ``Progressive Probabilistic Hough Transform'', Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (1999).
27) Y. Okuda, Y. Kita, I. Matsuda, S. Itoh: ``Estimation of Wall Planes in an Indoor Scene Captured by a 360-degree Camera'', Forum on Information Technology 2021 (FIT 2021, Japanese), Vol.20, No.3, pp.109–112 (2021).
28) J. Canny: ``A Computational Approach To Edge Detection'', IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.8, pp.679–698 (1986).
29) M. Zhang, X. Hu, J. Yao, L. Zhao, J. Li, J. Gong: ``Line-Based Geometric Consensus Rectification and Calibration From Single Distorted Manhattan Image'', IEEE Access, vol.7, pp.156400–156412 (2019).
30) H. Komagata, I. Ishii, A. Takahashi, D. Wakatsuki, H. Imai: ``A Geometric Calibration Method of Internal Camera Parameter for Fish-Eye Lenses'', IEICE Trans. on Infomation & System (Japanese Edition), Vol.J89-D, No.1, pp.64–73 (2006).
31) S. Suzuki, K. Abe: ``Topological Structural Analysis of Digitized Binary Images by Border Following'', Computer Vision, Graphics, and Image Processing, Vol.30, No.1, pp.32–46 (1985).
32) A. W. Fitzgibbon, R. B. Fisher: ``A Buyer's Guide to Conic Fitting'', Proc. of the 6th British conference on Machine vision, Vol.2, pp.513–522 (1995).
33) Euler Angles Intuition –Rotation Matrix and Angles–, Table of Euler Angles from Rotation Matrix, https://programming-surgeon.com/en/euler-angles-en/#matrix-to-angles (2024).

Appendix A. Equidistant projection camera model

In the equidistant projection model described by Eq. (1), the angle and distance are proportional. When this relationship is modeled, the imaging surface of a complementary metal-oxide-semiconductor (CMOS) camera is expressed as a hemisphere, as illustrated in **Fig.12** (a). However, because calculations are difficult on a hemisphere, I assume a virtual plane, as shown in **Fig.12** (b). The coordinates of the object on this virtual plane are $[\hat{u}, \hat{v}]$, while the distance between the object and coordinate origin is $\hat{r}$, and the positions where the following equation holds define the virtual plane:

$$\hat{r} = \frac{r}{f} = \theta. \tag{A1}$$

The distance between the virtual surface and camera focal point is denoted as $\boldsymbol{d}$. Due to the nature of the equidistant projection model, $\boldsymbol{d}$ is a function of $\hat{r}$. From Fig.12 (b), the following equation holds:

$$\frac{\hat{r}}{d} = \tan\theta. \tag{A2}$$

By substituting Eq. (A1) into Eq. (A2), the following equation is obtained:

$$d = \frac{\hat{r}}{\tan\hat{r}}. \tag{A3}$$

From Eq. (A3), Eqs. (2)–(4) hold. Next, the spatial coordinates in Fig.12 (b) are converted into digital coordinates, as illustrated in Fig.12 (c). Usually, the vertical axis of a digital image is opposite to that of the spatial coordinates, and the upper-left corner is set as the origin. Therefore, if the magnification coefficients are $[f_u, f_v]$, then Eq. (5) holds.

(Received Dec. 12, 2024)
(Revised Nov. 19, 2025)

Hideki KOMAGATA (***Member***)

He received his B.E., M.E., and Ph.D. degrees in information engineering from Niigata University, Niigata, Japan, in 2003, 2005 and 2010, respectively. He worked at the School of Biomedical Engineering at Saitama Medical University in Japan from 2006 to 2024. He is currently an Associate Professor at Department of Informatics at Tokyo University of Information Sciences in Japan. His research interests include computer vision and medical imaging. He is a member of IEICE (the Institute of Electronics Information and Communication Engineers) and IIEEJ (the Institute of Image Electronics Engineers of Japan).

Call for Late Breaking Papers

The 9th IIEEJ International Conference on Image Electronics and Visual Computing 2026 (IEVC2026)

Hiroshima Univ., Hiroshima, Japan / March 16-19, 2026

https://www.iieej.org/en/ievc2026/

The International Conference on Image Electronics and Visual Computing 2026 (IEVC2026) will be held in Hiroshima City, Japan, on March 16-19, 2026, as the 9th international academic event of the Institute of Image Electronics Engineers of Japan (IIEEJ). The conference aims to bring together researchers, engineers, developers, and students from various fields in both academia and industry for discussing the latest researches, standards, developments, implementations and application systems in all areas of image electronics and visual computing.

Venue:

Hiroshima University Kasumi Campus, Hiroshima City, Japan

Paper Submission:

The official language is English, and authors should submit their papers as PDF through the online submission system at the following IEVC2026 official website: https://www.iieej.org/en/ievc2026/
The paper submission guide and IEVC formats (TeX format / MS Word format) are also provided at this site. (Submissions to the General Papers category have already closed.)

Late Breaking Papers:

Papers submitted for this category will be accepted for the conference with simple checking. The paper length is 1 page, and authors can select one from the following two types: 1) Technical papers or 2) Art/Demo papers. The Late Breaking Papers will NOT be published in the IEEE Xplore Digital Library.

Important Dates

- Paper Submission (1 page):	Jan. 9, Friday, 2026
- Notification of Acceptance:	Jan. 23, Friday, 2026
- Camera-Ready Submission (1 page):	Jan. 30, Friday, 2026

Contact:

IEVC2026 Secretariat
The Institute of Image Electronics Engineers of Japan (IIEEJ)
E-mail: ievc2026-contact@iieej.org

Call for Papers
Special Issue on
Image-Related Technology for Social Contribution

IIEEJ Editorial Committee

In recent years, global environmental issues, declining birthrates and increasing aging populations, disaster prevention and mitigation, and the correction of regional disparities have become increasingly serious, diverse, and complex. Amid these challenges, advances in imaging and video technologies, sensing technologies, AI, robotics, drone utilization, and data analytics have led to expanded applications of imaging and electronic technologies. These technologies are now being utilized not only to solve problems and improve efficiency in various fields—such as disaster prevention and mitigation, medical and welfare support, education and cultural promotion, environmental conservation, and regional revitalization—but also to create new forms of social value. At the same time, when implementing such technologies in modern society, it is necessary to consider aspects such as fairness, ethics, and privacy. Efforts are therefore required to rebuild the relationship between technology and society in a desirable and sustainable manner.

In this special issue, we invite various categories (general paper, short paper, system development paper, data paper, practice-oriented paper) of papers under the theme of “Social Contribution.” We would be grateful if you could submit your papers on this special issue.

1. Topics covered include but not limited to

   Regional Revitalization, Regional Problem Solving, Disaster Prevention, Disaster Recovery, Education, Medicine, Healthcare, Home Life, Welfare, Safety and Security Technologies, Crisis Management, Wearable Devices, VR/AR, Smartphone Applications, Universal Design, Affective design, Usability, UI/UX, Ergonomics, Accessibility, AI Utilization, Automation, Robotics, Drone Utilization, Data Analysis, Digital Archiving

2. Treatment of papers

   Submission paper style format and double-blind peer review process are the same as the regular paper. If the number of accepted papers is less than the minimum number for the special issue, the acceptance paper will be published as the regular contributed paper. We ask for your understanding and cooperation.

3. Publication of Special Issue:

   IEEJ Transactions on Image Electronics and Visual Computing Vol.15, No.1 (June 2027)

4. Submission Deadline:

   Friday, October 30, 2026

5. Contact details for Inquires:

   IIEEJ Office E-mail: hensyu@iieej.org

6. Online Submission

   URL: http://www.editorialmanager.com/iieej/

Revised: January 6, 2017

Revised: July 6, 2018

Revised: Dec. 10, 2024

Guidance for Paper Submission

1. Submission of Papers

(1) Preparation before submission

- The authors should download "Guidance for Paper Submission" and "Style Format" from the "Academic Journals", "English Journals" section of the Society website and prepare the paper for submission.
- Two versions of "Style Format" are available, TeX and MS Word. To reduce publishing costs and effort, use of TeX version is recommended.
- There are four categories of manuscripts as follows:
  - Ordinary paper: It should be a scholarly thesis on a unique study, development or investigation concerning image electronics engineering. This is an ordinary paper to propose new ideas and will be evaluated for novelty, utility, reliability and comprehensibility. As a general rule, the authors are requested to summarize a paper within eight pages.
  - Short paper: It is not yet a completed full paper, but instead a quick report of the partial result obtained at the preliminary stage as well as the knowledge obtained from the said result. As a general rule, the authors are requested to summarize a paper within four pages.
  - System development paper: It is a paper that is a combination of existing technology or it has its own novelty in addition to the novelty and utility of an ordinary paper, and the development results are superior to conventional methods or can be applied to other systems and demonstrates new knowledge. As a general rule, the authors are requested to summarize a paper within eight pages.
  - Data Paper: A summary of data obtained in the process of a survey, product development, test, application, and so on, which are the beneficial information for readers even though its novelty is not high. As a general rule, the authors are requested to summarize a paper within eight pages.
  - Survey Paper: A summary of existing Research and Developments, organized under some viewpoint, compared for the sake of positioning purpose, observed as the changes in generations. Comprehensive references, overall perspective, objective evaluation, are needed without advertising specific organizations. It is also appreciated that the status and problems of the field, and the effect of them to the researchers and concerned people are undersood by the author, and the resulant paper encourages the new entry into the field, accelerates further development of related technologies, and prompts the development in even other fields or brand new researches. As a general rule, the authors are requested to summarize a paper within eight pages.

- To submit the manuscript for ordinary paper, short paper, system development paper, or data paper, at least one of the authors must be a member or a student member of the society.
- We prohibit the duplicate submission of a paper. If a full paper, short paper, system development paper, or data paper with the same content has been published or submitted to other open publishing forums by the same author, or at least one of the co-authors, it shall not be accepted as a rule. Open publishing forum implies internal or external books, magazines, bulletins and newsletters from government offices, schools, company organizations, etc. This regulation does not apply to a preliminary draft to be used at an

annual meeting, seminar, symposium, conference, and lecture meeting of our society or other societies (including overseas societies). A paper that was once approved as a short paper and being submitted again as the full paper after completion is not regarded as a duplicate submission.

(2) Submission stage of a paper

- Delete all author information at the time of submission. However, deletion of reference information is the author's discretion.
- At first, please register your name on the paper submission page of the following URL, and then log in again and fill in the necessary information. Use the "Style Format" to upload your manuscript. An applicant should use PDF format (converted from dvi of TeX or MS Word format) for the manuscript. As a rule, charts (figures and tables) shall be inserted into the manuscript to use the "Style Format". (a different type of data file, such as audio and video, can be uploaded at the same time for reference.)

  http://www.editorialmanager.com/iieej/
- If you have any questions regarding the submission, please consult the editor at our office.

Contact:
Person in charge of editing
The Institute of Image Electronics Engineers of Japan
3-35-4-101, Arakawa, Arakawa-Ku, Tokyo 116-0002, Japan
E-mail: hensyu@iieej.org
Tel: +81-3-5615-2893, Fax: +81-3-5615-2894

2. Review of Papers and Procedures

(1) Review of a paper

- A manuscript is reviewed by professional reviewers of the relevant field. The reviewer will deem the paper "acceptance", "conditionally acceptance" or "returned". The applicant is notified of the result of the review by E-mail.
- Evaluation method

  Ordinary papers are usually evaluated on the following criteria:
  - ✓ Novelty: The contents of the paper are novel.
  - ✓ Utility: The contents are useful for academic and industrial development.
  - ✓ Reliability: The contents are considered trustworthy by the reviewer.
  - ✓ Comprehensibility: The contents of the paper are clearly described and understood by the reviewer without misunderstanding.

A short paper can be evaluated by having a quickness on the research content and evaluated to have new knowledge with results even if that is partial or for specific use, apart from the novelty and utility of an ordinary paper.

A system development paper is evaluated based on the following criteria, apart from the novelty and utility of an ordinary paper.

- ✓ Novelty of system development: Even when integrated with existing technologies, the novelty of the combination, novelty of the system, novelty of knowledge obtained from the developed system, etc. are recognized as the novelty of the system.
- ✓ Utility of system development: It is comprehensively or partially superior compared to similar systems. Demonstrates a pioneering new application concept as a system. The combination has appropriate optimality for practical use. Demonstrates performance

limitations and examples of performance of the system when put to practical use.

A data paper is considered novel if new deliverables of test, application and manufacturing, the introduction of new technology and proposals in the worksite have any priority, even though they are not necessarily original, apart from the novelty and utility of an ordinary paper. Also, if the new deliverables are superior compared to the existing technology and are useful for academic and industrial development, they should be evaluated.

A survey paper is evaluated by comprehensiveness, overviewing point, and objectiveness apart from the novelty of an ordinary paper. Reliability, comprehensibility, completeness of reference papers are common to those in an ordinary paper. Utility is evaluated how the paper will enlighten the readers in the target fields.

(2) Procedure after a review

- In case of acceptance, the author prepares a final manuscript (as mentioned in 3.).
- In the case of acceptance with comments by the reviewer, the author may revise the paper in consideration of the reviewer's opinion and proceed to prepare the final manuscript (as mentioned in 3.).
- In case of conditional acceptance, the author shall modify a paper based on the reviewer's requirements by a specified date (within 60 days), and submit the modified paper for approval. The corrected parts must be colored or underlined. A reply letter must be attached that carefully explains the corrections, assertions and future issues, etc., for all of the acceptance conditions.
- In case a paper is returned, the author cannot proceed to the next step. Please look at the reasons the reviewer lists for the return. We expect an applicant to try again after reviewing the content of the paper.

(3) Review request for a revised manuscript

- If you want to submit your paper after conditional acceptance, please submit the reply letter to the comments of the reviewers, and the revised manuscript with revision history to the submission site. Please note the designated date for submission. Revised manuscripts delayed more than the designated date be treated as new applications.
- In principle, a revised manuscript willl be reviewed by the same reviewer. It is judged either acceptance or returned.
- After the judgment, please follow the same procedure as (2).

3. Submission of final manuscript for publication

(1) Submission of a final manuscript

- An author, who has received the notice of "Acceptance", will receive an email regarding the creation of the final manuscript. The author shall prepare a complete set of the final manuscript (electronic data) following the instructions given and send it to the office by the designated date.
- The final manuscript shall contain a source file (TeX edition or MS Word version) and a PDF file, eps files for all drawings (including bmp, jpg, png), an eps file for author's photograph (eps or jpg file of more than 300 dpi with length and breadth ratio 3:2, upper part of the body) for authors' introduction. Please submit these in a compressed format, such as a zip file.
- In the final manuscript, write the name of the authors, name of an organizations, introduction of authors, and if necessary, an appreciation acknowledgment. (cancel macros in the Style file)

- An author whose paper is accepted shall pay a page charge before publishing. It is the author's decision to purchase offprints. (ref. page charge and offprint price information)

(2) Galley print proof

- The author is requested to check the galley (hard copy) a couple of weeks before the paper is published in the journal. Please check the galley by the designated date (within one week). After making any corrections, scan the data and prepare a PDF file, and send it to our office by email. At that time, fill in the Offprint Purchase Slip and Copyright Form and return the scanned data to our office in PDF file form.
- In principle, the copyrights of all articles published in our journal, including electronic form, belong to our society.
- You can download the Offprint Purchase Slip and the Copyright Form from the journal on our homepage. (ref. Attachment 2: Offprint Purchase Slip, Attachment 3: Copyright Form)

(3) Publication

- After final proofreading, a paper is published in the Academic journal or English transaction (both in electronic format) and will also be posted on our homepage.

IIEEJ Trans. on Image Electronics and Visual Computing

Vol. 13, No. 2 2025

Editor in Chief: Osamu Uchida
The Institute of Image Electronics Engineers of Japan
3-35-4-101, Arakawa, Arakawa-ku, Tokyo 116-0002, Japan

Print: ISSN 2188-1898
Online: ISSN 2188-1901
CD-ROM: ISSN 2188-191x