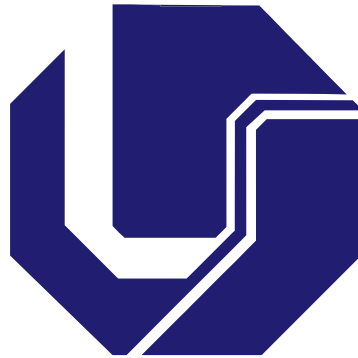


**UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE ENGENHARIA ELÉTRICA  
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**



**2D–3D Spatial Registration for Remote Inspection of  
Power Substations**

**Leandro Resende Mattioli**

**March**

**2021**

**LEANDRO RESENDE MATTIOLI**

**2D–3D Spatial Registration for Remote Inspection of Power  
Substations**

Tese apresentada ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Uberlândia como requisito parcial para a obtenção do título de Doutor em Ciências.

Área de concentração: Processamento da informação.

Orientador: Prof. Dr. Alexandre Cardoso.

Uberlândia/MG

Março/2021

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU  
com dados informados pelo(a) próprio(a) autor(a).

M444 2021	<p>Mattioli, Leandro Resende, 1987- 2D-3D Spatial Registration for Remote Inspection of Power Substations [recurso eletrônico] / Leandro Resende Mattioli. - 2021.</p> <p>Orientador: Alexandre Cardoso. Tese (Doutorado) - Universidade Federal de Uberlândia, Pós-graduação em Engenharia Elétrica. Modo de acesso: Internet. Disponível em: <a href="http://doi.org/10.14393/ufu.te.2021.198">http://doi.org/10.14393/ufu.te.2021.198</a> Inclui bibliografia. Inclui ilustrações.</p> <p>1. Engenharia elétrica. I. Cardoso, Alexandre, 1964-, (Orient.). II. Universidade Federal de Uberlândia. Pós- graduação em Engenharia Elétrica. III. Título.</p> <p>CDU: 621.3</p>
--------------	---

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:

Gizele Cristine Nunes do Couto - CRB6/2091



### ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Engenharia Elétrica				
Defesa de:	Tese de Doutorado, 285, PPGEELT				
Data:	Trinta de março de dois mil e vinte e um	Hora de início:	14:00	Hora de encerramento:	18:00
Matrícula do Discente:	11523EEL014				
Nome do Discente:	Leandro Resende Mattioli				
Título do Trabalho:	2D–3D Spatial Registration for Remote Inspection of Power Substations				
Área de concentração:	Processamento da informação				
Linha de pesquisa:	Computação gráfica				
Projeto de Pesquisa de vinculação:	Título: Adequação do Sistema de Realidade Virtual da Cemig para a Integração com Recursos de Inspeção por Imagens em Tempo Real e Treinamento Conjunto das Equipes de Campo e do COS Agência Financiadora: Aneel/CEMIG Início: 11/2018 Término 11/2021 No. do Projeto na agência: CEMIG GT 0618 Professor Coordenador: Alexandre Cardoso				

Reuniu-se por meio de videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Engenharia Elétrica, assim composta: Professores Doutores: Edgard Afonso Lamounier Junior- FEELT/UFU; Renato Aquino Lopes - FACOM/UFU, Márcio José da Cunha - FEELT/UFU; Romero Tori - USP; Luciano Pereira Soares - INSPER; Alexandre Cardoso - FEELT/UFU, orientador(a) do(a) candidato(a).

Iniciando os trabalhos o(a) presidente da mesa, Dr(a). Alexandre Cardoso, apresentou a Comissão Examinadora e o candidato(a), agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor(a) presidente concedeu a palavra, pela ordem sucessivamente, aos(às) examinadores(as), que passaram a arguir o(a) candidato(a). Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o(a) candidato(a):

Aprovado(a).

Esta defesa faz parte dos requisitos necessários à obtenção do título de Doutor.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.



Documento assinado eletronicamente por **Alexandre Cardoso, Professor(a) do Magistério Superior**, em 30/03/2021, às 16:22, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Luciano Pereira Soares, Usuário Externo**, em 30/03/2021, às 16:58, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Marcio José da Cunha, Professor(a) do Magistério Superior**, em 30/03/2021, às 19:35, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Edgard Afonso Lamounier Junior, Professor(a) do Magistério Superior**, em 30/03/2021, às 20:22, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Renato de Aquino Lopes, Professor(a) do Magistério Superior**, em 30/03/2021, às 20:26, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Romero Tori, Usuário Externo**, em 31/03/2021, às 12:32, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [https://www.sei.ufu.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **2648995** e o código CRC **797F3296**.

**LEANDRO RESENDE MATTIOLI**

**2D–3D Spatial Registration for Remote Inspection of Power  
Substations**

Tese apresentada ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Uberlândia como requisito parcial para a obtenção do título de Doutor em Ciências.

**Banca Examinadora:**

Prof. Dr. Alexandre Cardoso – Orientador (UFU)

Prof. Edgard A. Lamounier Jr., PhD. (UFU)

Prof. Dr. Luciano Pereira Soares (INSPER)

Prof. Dr. Márcio José da Cunha (UFU)

Prof. Dr. Renato Aquino Lopes (UFU)

Prof. Dr. Romero Tori (USP)

Uberlândia/MG

Março / 2021

# Acknowledgements

To my wife Aline, for all the patience and understanding during this cycle.

To my daughter Luana, for showing her beautiful smile during so many difficult moments.

To my parents Neila and Roberto, for being the best examples of kindness, honesty, and perseverance.

To my brother Fernando, whose wisdom is always a big source of inspiration and pride.

To all my friends and coworkers from the Virtual and Augmented Reality Group, for the great moments working together and talking during our precious coffee-breaks.

To professors Alexandre and Edgard, for all the help and motivation provided.

To professors Luciano, Márcio, Renato, and Romero for the valuable contributions.

Thank you all for being part of this journey!

# Resumo

A inspeção remota e o controle supervisório são requisitos críticos para fábricas modernas, vigilância de civis, sistemas de energia e outras áreas. Para reduzir o tempo da tomada de decisão, os operadores precisam de uma elevada consciência da situação em campo, o que implica em uma grande quantidade de dados a serem apresentados, mas com menor carga sensorial possível. Estudos recentes sugerem a adoção de técnicas de visão computacional para inspeção automática, e a Realidade Virtual (VR) como uma alternativa às interfaces tradicionais do SCADA. Entretanto, apesar de fornecer uma boa representação do estado da subestação, os ambientes virtuais carecem de algumas informações de campo, provenientes de câmeras e microfones. Como essas duas fontes de dados (VR e dispositivos de captura) não são integrados em uma única solução, perde-se a oportunidade de usar VR como uma ferramenta de inspeção remota conectada ao SCADA, durante a operação e rotinas de respostas a desastres. Este trabalho trata de um método para aumentar ambientes virtuais de subestações com imagens de campo, permitindo aos operadores a rápida visualização de uma representação virtual do entorno da área monitorada. O ambiente resultante é integrado com uma máquina de inferência estados por imagens, comparando continuamente os estados inferidos com aqueles reportados pela base SCADA. Na ocasião de uma discrepância, um alarme é gerado e possibilita que a câmera virtual seja imediatamente teletransportada para a região afetada, acelerando o processo de retomada do sistema. A solução se baseia em uma arquitetura cliente-servidor e permite múltiplas câmeras presentes em múltiplas subestações. Os resultados dizem respeito à qualidade do registro 2D-3D e à taxa de renderização para um cenário simples. As métricas quantitativas coletadas sugerem bons níveis de registro e estimativa de pose de câmera, além de uma taxa ótima de renderização para fins de inspeção de equipamentos em subestações.

**Palavras-chave:** Virtualidade aumentada; Controle supervisório; Inspeção remota; Subestações de energia; Estimativa de pose.

# Abstract

Remote inspection and supervisory control are critical features for smart factories, civilian surveillance, power systems, among other domains. For reducing the time to make decisions, operators must have both a high situation awareness, implying a considerable amount of data to be presented, and minimal sensory load. Recent research suggests the adoption of computer vision techniques for automatic inspection, as well as virtual reality (VR) as an alternative to traditional SCADA interfaces. Nevertheless, although VR may provide a good representation of a substation's state, it lacks some real-time information, available from online field cameras and microphones. Since these two sources of information (VR and field information) are not integrated into one single solution, we miss the opportunity of using VR as a SCADA-aware remote inspection tool, during operation and disaster-response routines. This work discusses a method to augment virtual environments of power substations with field images, enabling operators to promptly see a virtual representation of the inspected area's surroundings. The resulting environment is integrated with an image-based state inference machine, continuously checking the inferred states against the ones reported by the SCADA database. Whenever a discrepancy is found, an alarm is triggered and the virtual camera can be immediately teleported to the affected region, speeding up system reestablishment. The solution is based on a client-server architecture and allows multiple cameras deployed in multiple substations. Our results concern the quality of the 2D-3D registration and the rendering framerate for a simple scenario. The collected quantitative metrics suggest good camera pose estimations and registrations, as well as an arguably optimal rendering framerate for substations' equipment inspection.

**Keywords:** Augmented virtuality; Supervisory control; Remote inspection; Power substations; Pose estimation.

# List of Figures

1.1	Wonderware InTouch custom panel. . . . .	1
1.2	RVCEMIG architecture. . . . .	2
1.3	Diffuse attention. . . . .	3
1.4	Several user interfaces for indoor robot teleoperation. . . . .	4
2.1	Reality-virtuality axes. . . . .	7
2.2	3D videoconferencing. . . . .	8
2.3	Image coordinate system (ICS). . . . .	10
2.4	World's Z-Up and Engine's Y-Up coordinate systems. . . . .	10
3.1	Bibliography entries time distribution. . . . .	16
3.2	Humanoid models mimicking humans. . . . .	16
3.3	VR/AV technology for marine systems. . . . .	17
3.4	Forbidden ladder climbing detection. . . . .	17
3.5	Checking deflection angle. . . . .	18
3.6	Foreground-background segmentation . . . . .	18
3.7	Drone-based remote inspection. . . . .	19
3.8	Image-to-image registration applied to substation monitoring. . . . .	19
3.9	Video monitoring image augmented with virtual thermometers. . . . .	20
3.10	Substation inspection robot. . . . .	20
3.11	2D–3D registration for construction machines teleoperation. . . . .	21
3.12	Embedding live actors and entities into a 360° photo, with and without chroma key. . . . .	22
3.13	Video surveillance image fusion. . . . .	22
3.14	Augmented virtuality game experience. . . . .	23
4.1	Augmented virtuality for remote inspection -- system architecture. . . . .	29

4.2	Registration server internal view. . . . .	30
4.3	Single PTZ-camera with time-multiplexing. . . . .	31
4.4	Nova Ponte cameras (thermal and RGB). . . . .	31
5.1	Converging lines and rectangular area $v_2$ . . . . .	35
5.2	Elements for calculating $h_3$ . . . . .	36
5.3	Determining the pose of the rectangular region $v_2$ . . . . .	37
5.4	Photo used for the <i>avmath</i> experiment. . . . .	39
5.5	Perspective projection tab of the <i>avmath</i> GUI application. . . . .	39
5.6	Inverse perspective projection tab of the <i>avmath</i> GUI application. . . . .	40
5.7	GUI application to extract keypoints. . . . .	41
5.8	Nova Ponte's substation power disconnecter keypoints convention. . . . .	41
5.9	Virtual environment and 2D–3D registration for the <i>avloopback</i> prototype. . .	42
5.10	Nova Ponte power substation. . . . .	43
5.11	Field photo used in the <i>avcamera</i> prototype. . . . .	43
5.12	2D–3D registration in a transparent rectangular region for the <i>avcamera</i> prototype. .	44
5.13	Back end simplified database model. . . . .	45
5.14	Image space keypoints and projections from world space keypoints. . . . .	47
5.15	Equipment state simulator. . . . .	48
5.16	Assets decorated with interactive 3D objects. . . . .	49
5.17	2D–3D spatial registration configuration dialog. . . . .	50
5.18	Alarm dialog. . . . .	51
5.19	Alarm queue icon. . . . .	52
6.1	Transparent rectangular region for better inspecting image and world keypoints. .	56
6.2	Keypoints and projections for the <i>avmath</i> prototype. . . . .	60
6.3	Keypoints and projections for the <i>avloopback</i> prototype (scenario 1). . . . .	62
6.4	Keypoints in the rendered AV <i>image</i> for the <i>avloopback</i> prototype (scenario 1). .	64
6.5	Focal length and rectangular region size adjustments. . . . .	65
6.6	Keypoints and projections for the <i>avloopback</i> prototype (scenario 2). . . . .	66
6.7	Keypoints in the rendered AV <i>image</i> for the <i>avloopback</i> prototype (scenario 1). .	68
6.8	Keypoints and projections for the <i>avcamera</i> prototype. . . . .	70
6.9	Keypoints in the rendered AV <i>image</i> for the <i>avcamera</i> prototype . . . . .	71
6.10	Thermal and RGB image dataset sample. . . . .	72
6.11	Focal length scale factor impact on $\overline{d_{PNP}}$ metric. . . . .	73

6.12 Ternary-search iterations . . . . .	74
6.13 Keypoints discrepancy. . . . .	75
6.14 2D–3D registration for experiment T1. . . . .	76
6.15 Power disconnecter registrations for other experiments. . . . .	77
6.16 Multiple calibration requests. . . . .	79
6.17 Multiple registration requests. . . . .	80
6.18 Detailed view of the registration requests. . . . .	80
6.19 Video streaming – no delay between frames. . . . .	81
6.20 Video streaming – small delay between frames. . . . .	82

# List of Tables

3.1	Search terms. . . . .	15
3.2	Unfiltered results. . . . .	15
3.3	Features table. . . . .	24
5.1	Development tools and languages. . . . .	38
5.2	Representative icons for spatial registration. . . . .	50
6.1	Prototype-metrics table. . . . .	58
6.2	<i>avmath</i> experiment parameters. . . . .	58
6.3	<i>avmath</i> prototype keypoints. . . . .	59
6.4	<i>avmath</i> PNP errors. . . . .	59
6.5	<i>avloopback</i> experiment parameters. . . . .	60
6.6	<i>avloopback</i> prototype keypoints. . . . .	61
6.7	<i>avloopback</i> scenario 1 – PNP errors. . . . .	62
6.8	<i>avloopback</i> scenario 1 – errors in the rendered image. . . . .	64
6.9	<i>avloopback</i> scenario 2 – PNP errors. . . . .	66
6.10	<i>avloopback</i> scenario 2 – errors in the rendered image. . . . .	67
6.11	<i>avcamera</i> experiment parameters. . . . .	68
6.12	<i>avcamera</i> prototype keypoints. . . . .	69
6.13	<i>avcamera</i> PNP errors. . . . .	70
6.14	<i>avcamera</i> – errors in the rendered image. . . . .	71
6.15	Experiments codes. . . . .	73
6.16	<i>Perspective-n-Point</i> and calibration results. . . . .	74
6.17	<i>avcemig</i> – final registration results. . . . .	75
6.18	Hardware used in the performance experiments. . . . .	78
6.19	<i>avcemig</i> – registration performance. . . . .	78

6.20	Mean and standard deviation values for multiple calibration requests. . . . .	79
6.21	Mean and standard deviation values for the experiment with multiple registra- tion requests. . . . .	81
6.22	Results for the first prototypes. . . . .	83
6.23	Performance metrics values. . . . .	83

# Acronyms

API	Application Programming Interface
AR	Augmented Reality
AV	Augmented Virtuality
CAD	Computer-Aided design
CV	Computer Vision
EPnP	Efficient Perspective-n-Point Camera Pose Estimation
FoV	Field of View
FPS	Frames per Second (rendering metric)
GUI	Graphical User Interface
IIoT	Industrial Internet of Things
LAE	Live actors and entities
LED	Light-emitting Diode
LIDAR	Light Detection and Ranging
MAR	Mixed and Augmented Reality
MSE	Mean Squared Error
POSIT	Pose from Orthography and Scaling with Iterations
PTZ	Pan-Tilt-Zoom
RGB	Red, Green and Blue (color model)
SCADA	Supervisory Control and Data Acquisition
VR	Virtual Reality

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Objectives . . . . .	5
1.2	Fundings . . . . .	6
1.3	Thesis Structure . . . . .	6
<b>2</b>	<b>Fundamentals</b>	<b>7</b>
2.1	Augmented Virtuality . . . . .	7
2.2	Perspective Projection and Camera Pose Estimation . . . . .	9
2.2.1	Coordinate Systems . . . . .	9
2.2.2	Perspective Projection . . . . .	10
2.2.3	Camera Pose Estimation . . . . .	11
2.3	Power Substation Remote Inspection . . . . .	12
<b>3</b>	<b>Related Work</b>	<b>14</b>
3.1	Review Method . . . . .	14
3.2	Monitoring and Inspection . . . . .	16
3.3	2D–3D Spatial Registration . . . . .	21
3.4	Comparative Analysis . . . . .	23
3.5	Conclusions . . . . .	24
<b>4</b>	<b>Problem-Solving Methodology and System Architecture</b>	<b>26</b>
4.1	Mathematical Model of the 2D–3D Registration . . . . .	26
4.2	PNP-based Perspective Projection and Inverse Perspective Transform . . . . .	27
4.3	Loopback Registration Experiment . . . . .	27
4.4	Field Photo Registration – Augmented Virtuality Prototype . . . . .	28
4.5	Augmented Virtuality for Remote Inspection . . . . .	29

<b>5</b>	<b>Development</b>	<b>33</b>
5.1	2D–3D Registration Mathematical Model . . . . .	33
5.1.1	Image overlay . . . . .	34
5.2	Software Tools and Languages . . . . .	37
5.3	Perspective and Inverse Perspective Transforms Workbench (avmath) . . . . .	38
5.4	Loopback Experiment (avloopback) . . . . .	40
5.5	Augmented Virtuality Prototype (avcamera) . . . . .	43
5.6	Augmented Virtuality for Remote Inspection (avcemig) . . . . .	44
5.6.1	Database model . . . . .	44
5.6.2	Focal length scale auto-set method . . . . .	46
5.6.3	SCADA integration . . . . .	48
5.6.4	User interface . . . . .	49
5.6.5	API specification . . . . .	52
5.7	Conclusions . . . . .	53
<b>6</b>	<b>Results</b>	<b>54</b>
6.1	Metrics for Quality Evaluation . . . . .	54
6.1.1	2D–3D Registration Metrics . . . . .	54
6.1.2	Performance Impact . . . . .	57
6.1.3	Prototype-Metrics Table . . . . .	57
6.2	Perspective-n-Point Prototype (avmath) . . . . .	58
6.3	Loopback Prototype (avloopback) . . . . .	60
6.3.1	Scenario 1: No adjustments . . . . .	61
6.3.2	Scenario 2: Adjustments for focal length and rectangular region size . . . . .	65
6.4	Field Camera Prototype (avcamera) . . . . .	68
6.4.1	Mean Euclidean distance and relative errors . . . . .	70
6.4.2	Distance between keypoints in the rendered AV image . . . . .	71
6.5	Final Prototype (avcemig) . . . . .	72
6.5.1	Experiments details . . . . .	73
6.5.2	Mean Euclidean distance and relative errors . . . . .	73
6.5.3	Distance between keypoints in the rendered AV image . . . . .	75
6.5.4	System performance . . . . .	77
6.6	Results Overview . . . . .	82

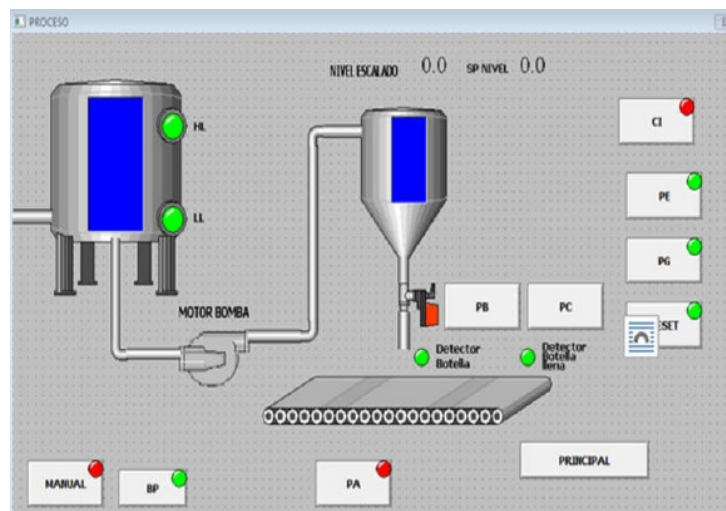
<b>7</b>	<b>Conclusion</b>	<b>84</b>
7.1	Ongoing and future work . . . . .	85
7.1.1	Automatic image keypoints extraction . . . . .	85
7.1.2	Sepia effect . . . . .	85
7.1.3	Registration-oriented modeling . . . . .	86
7.1.4	Foreground-background segmentation . . . . .	86
7.2	Final considerations . . . . .	86
<b>A</b>	<b>Web API Specification</b>	<b>88</b>
A.1	Requests for cameras and images . . . . .	88
A.2	Request for video streaming . . . . .	89
A.3	Requests for camera calibration . . . . .	90
<b>B</b>	<b>Publications</b>	<b>92</b>

## Introduction

In an industrial context, remote monitoring and operation can be realized using a supervisory control and data acquisition (SCADA) system connected to some industrial network. The former acts as a database, serving to its clients the equipment state and measurements, as well as receiving commands to alter such states and to define new setpoints. From another perspective, SCADA systems allow the design and customization of control panels, with multiple kinds of visualization techniques, such as real-time graphs, gauges, virtual LEDs, alarm lights, etc). This allowed the replacement of physical panels with computer screens, where it was viable – commonly the case for factory control rooms.

These customized panels, however, are not limited to schematic drawings. Commercial solutions like Wonderware InTouch (AVEVA, 2020) can make use of computer graphics animations to provide a better user experience, delivering views that mimic parts of the real factory geometry.

Figure 1.1: Wonderware InTouch custom panel.

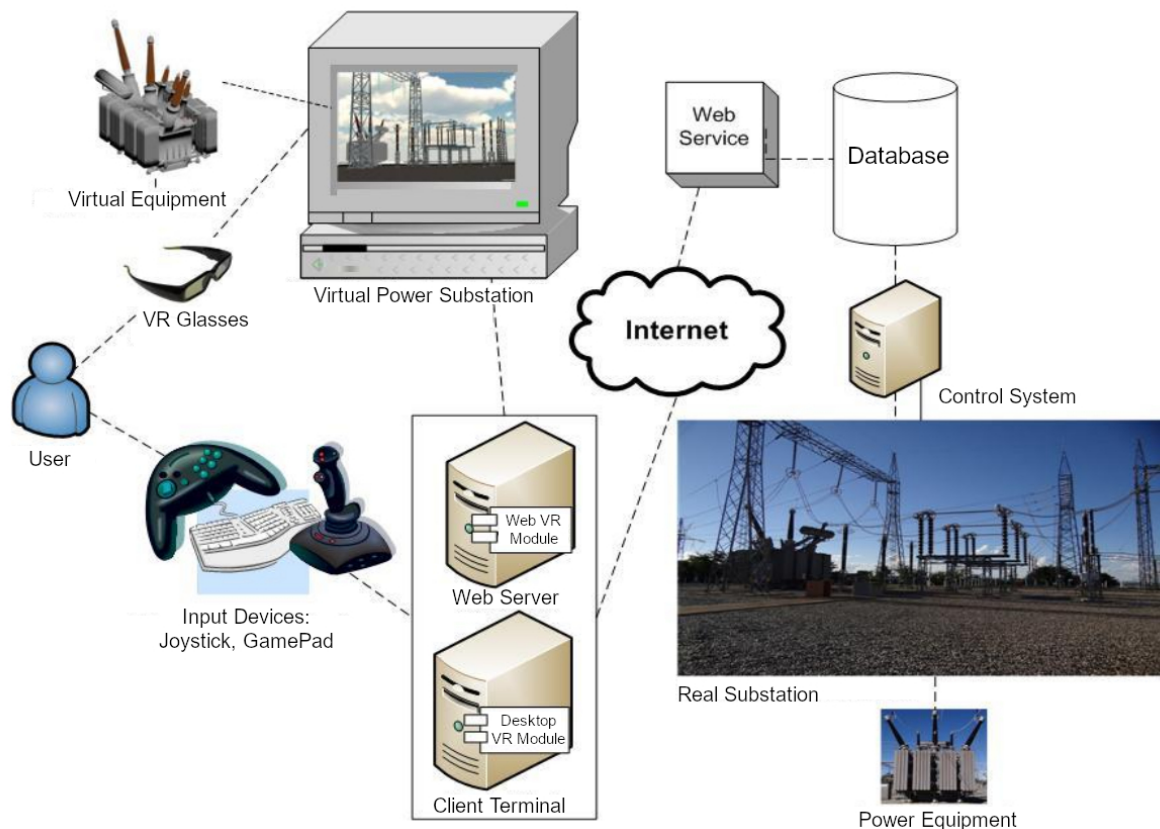


Source: (GUTIÉRREZ-MARROQUÍN; JIMÉNEZ; FERNÁNDEZ-FERNÁNDEZ, 2017).

In particular, electrical power substations have benefited from the Virtual Reality (VR) technology, exploring the potential of this advanced user interface technology as a complement for

the usual single-line diagrams. Cardoso et al. (2016) have provided a system for integrating real-time SCADA data with virtual environments of power substations, through a Web service interface, as shown in Figure 1.2.

Figure 1.2: RVCEMIG architecture.



Source: Adapted from (CARDOSO et al., 2016)

Since power systems are critical, one may not always rely solely on the state reported by the SCADA conventional sensors. Some failures need a quick visual inspection for better diagnostics. To safely allow the site to be unmanned, remote visual inspection, called Remote Inspection (RI) henceforth, becomes a valuable technique. In this manner, one reasonable option to be considered is the integration of RI data with a VR-based SCADA user interface. Referring to such integration, it is possible, for instance, to check the network's reported state for a disconnect switch against the one inferred from the last image (PAL et al., 2018; HONGKAI et al., 2017), acquired by an RI system.

Remote inspection systems can “proactively identify problems, get advantage in the business process, and provide a great convenience for the subsequent artificial decision” (LUO; DAI; QI, 2013). Ideally, operation center staff, as well as users from other departments, are capable of observing the electrical equipment and the environmental and contextual conditions in all remote substations (LUO; TU, 2005).

However, traditionally, such inspection systems demand a high level of diffuse attention from

the user, who needs to visualize and analyze images in multiple screens or windows (Figure 1.3). Even though each information channel already presents high relevance information, the operator can be “easily overwhelmed with the task of integrating these varied forms of data into a complete global view and understanding of a scene” (SEBE et al., 2003).

Figure 1.3: Diffuse attention.



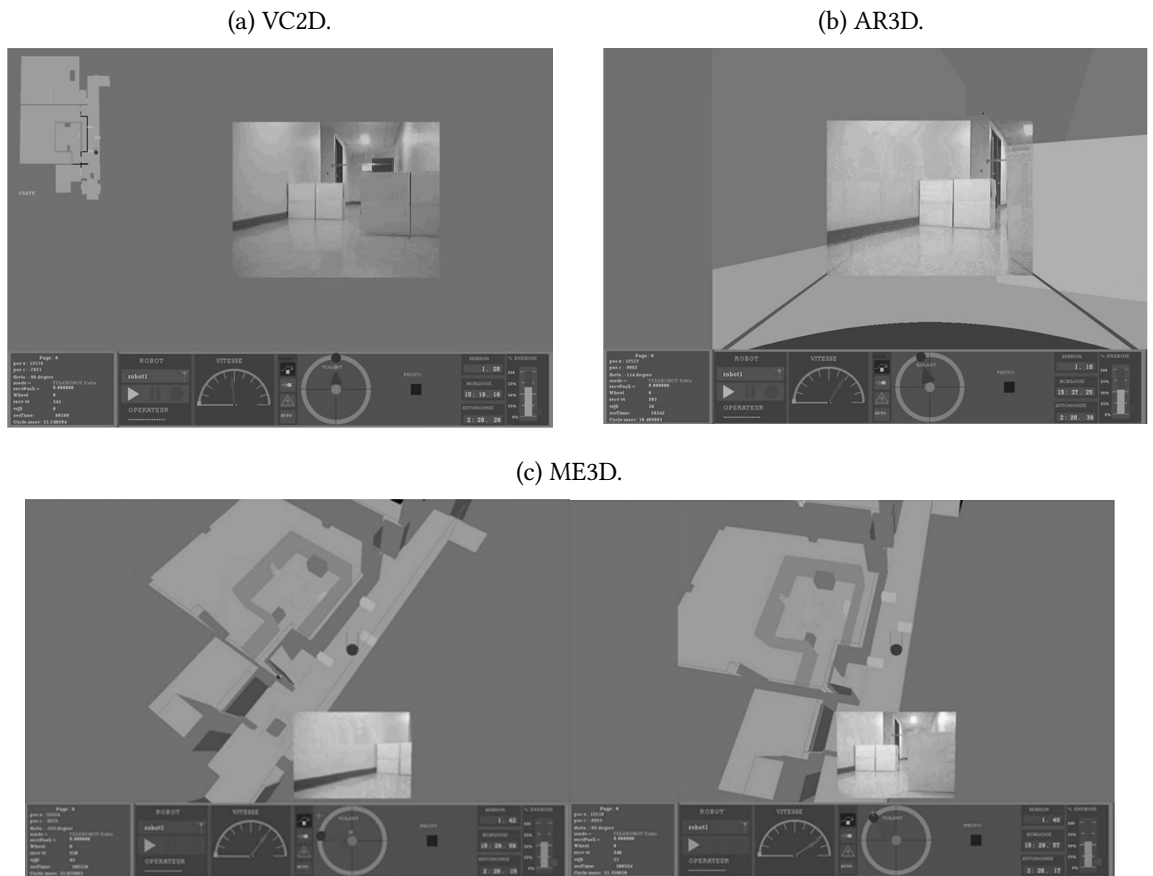
Source: (SEBE et al., 2003).

On a first look, this fact might seem irrelevant for the case of tele-assisted substations, since the installation of a high amount of cameras, covering the full equipment area and grabbing images in real-time, is currently impractical, due to network speed constraints for data transmission and geometrical constraints. In this sense, the most usual scenario has, traditionally, a considerably smaller number of cameras (BASTOS; MACHADO, 2010). But the very nature of substations implies in many repeated structures and similar power devices, which may confuse the operator during remote inspection. In fact, this similarity may mislead the operator about the actual location of the problem.

Therefore, bringing more contextual information to the data captured from the field might help the decision-making processes. The region surrounding the equipment, outside the captured image bounds, for example, cannot be easily analyzed only by employing a single-line diagram, since it represents the electrical and logical layout but not distances and other data from the physical layout. Again, this missing information in the field images requires the operator to memorize and remember the context of each camera. If a disaster such as an explosion occurs, the operator might need to take a look into several CAD drawings to find out, among the floor plans and side views, which assets are effectively affected by the event, due to physical proximity. For many substations and many field cameras deployed, which might be the case once network limitations are not a problem anymore, the challenge is even more difficult.

Considering the need for adding more context to RI systems, we might enumerate several user-centered approaches, taking into account technical and cognitive aspects. Along these lines, Labonte, Boissy, and Michaud (2010) have evaluated several ways of integrating real-time photos with a virtual environment counterpart, for robot teleoperation purposes (Figure 1.4). According to their results, the ME3D mixed-perspective exocentric display and the AR3D video-centric display were both capable of improving “efficiency and cognitive load (and, therefore, security) for novice operators, compared to VC2D” Labonte, Boissy, and Michaud (2010).

Figure 1.4: Several user interfaces for indoor robot teleoperation.



Adapted from (LABONTE; BOISSY; MICHAUD, 2010).

The problem of inserting an image so that their objects match their counterparts in the virtual environment is known as 2D–3D spatial image registration (POSTOLKA et al., 2020), and it is one of the key techniques for augmented virtuality (AV) systems (ISO, 2019). The AR3D user interface proposed by Labonte, Boissy, and Michaud (2010) does real-time 2D–3D spatial registration, which is one of the main topics discussed in our work. The technique significantly improves operator situation-awareness, especially when the camera views are unintuitive or limited (VAGVOLGYI et al., 2017).

Considering the case of power substations, a field image, as captured by a camera, can be surrounded by a tridimensional model of the nearby “as-built” structure, providing more

contextual information and extending its scope. Telemetry information acquired through sensors of the SCADA system can also be displayed along with the virtual environment. To help with remote inspection, these two sources of information – the field images and the SCADA records – may be compared against each other to detect inconsistencies.

This work proposes and evaluates a novel way of integrating these technologies, allowing multiple VR applications to query the last known color or thermal images for a given set of regions of interest, providing 2D–3D registration for these images and triggering alarms or teleporting the VR camera whenever there's a difference between the state reported by SCADA and the one inferred from the image. This allows for presumably quicker system reestablishment routines and failure diagnostics.

Besides, this work is a design science research (DSR) (CATER-STEEL; TOLEMAN; RAJAEIAN, 2019), motivated by requirements from a company organization. In this manner, we have developed artifacts and evaluated them with a case study and experimental methods, contributing to a knowledge base.

## 1.1 Objectives

The main objective of this research is to propose an architecture for making efficient spatial registration available to remote inspection systems and to demonstrate that this approach is not only feasible but also viable. In this sense, the following specific objectives are enumerated:

1. to identify the barriers related to real-time 2D–3D registration for remote inspection uses, considering a scenario with multiple image sources and multiple virtual environments to be augmented;
2. to describe the registration process in terms of mathematical manipulations and pose estimation algorithms;
3. to evaluate the proposed solution through a proof of concept, using a virtual environment and field images from a real power substation;
4. to assess the quality of the registration according to some quantitative metrics, considering not only the virtual camera's pose matching exactly the image capture conditions but also some poses with small variations (different points of view).

The research and development steps associated with these goals are presented in a separate chapter of this text.

## 1.2 Fundings

This research was funded by the Minas Gerais Energy Company (CEMIG) and the Brazilian Electricity Regulatory Agency (ANEEL), through project GT-0618. This study was also financed, in part, by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Brasil (CAPES), Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

## 1.3 Thesis Structure

This text is structured as follows. The first chapter presents the motivation and objectives of our research work. Chapter 2 gives a short introduction to some related technical topics, namely Augmented Reality, Perspective Projection, and Power Substation Inspection, providing some background for the comprehension of further chapters. Related work and some comparative analysis are then presented in Chapter 3. The proposed method and the system architecture are detailed in Chapter 4. In Chapter 5 the development details for each prototype are presented. Qualitative and quantitative metrics and results are discussed in Chapter 6. Finally, the main conclusions and future work are given in Chapter 7.

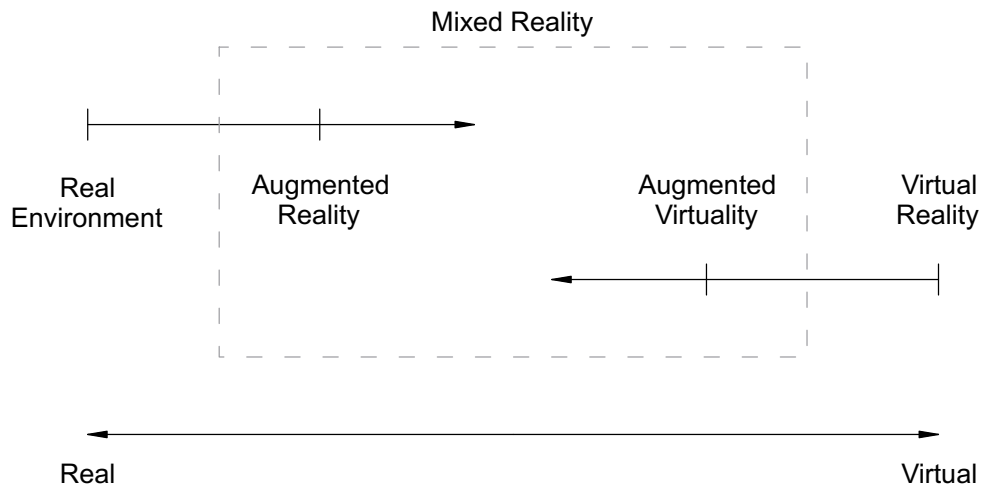
# Fundamentals

This chapter presents the key concepts used in this work, as well as the technical background for perspective projection and camera pose estimation.

## 2.1 Augmented Virtuality

Augmented Virtuality (AV) systems insert content taken from the physical reality into a predominantly virtual environment (MILGRAM; COLQUHOUN, 2001). One classical approach to visualize relations between AV and other similar technologies is through the Reality-Virtuality Continuum (MILGRAM; COLQUHOUN, 2001, p. 9). Figure 2.1 shows a modified version of the concept, in which we keep two separated axes to indicate that it is not possible to reach one end from its opposite. Indeed, it does not make sense to obtain a purely virtual environment by adding virtual content to the physical world and vice versa.

Figure 2.1: Reality-virtuality axes.



Source: Adapted from (MILGRAM; COLQUHOUN, 2001, p. 9).

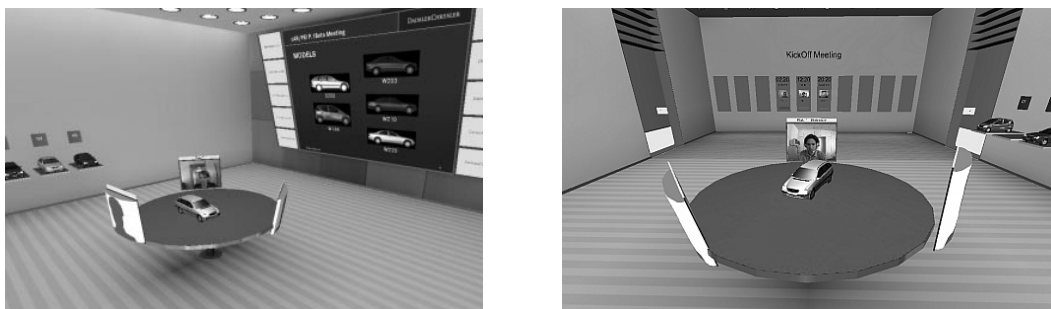
According to a standard published by the International Organization for Standardization (ISO), an augmented virtuality system is a “type of mixed reality system in which physical world data are embedded and/or registered with the representation of virtual world data” (ISO, 2019).

An interesting option to see the difference between Augmented Reality (AR) and Augmented Virtuality (AV) is to check in which domain navigation occurs. Generally, one can assume that in AR systems the navigation is done in the physical world, whereas in AV systems the navigation is done in the virtual environment, employing one or more input devices, such as joysticks and body trackers.

Although not as well established and diffused as the VR and AR technologies, many applications reveal the potential for the use of Augmented Virtuality. It can be used to improve the users’ experience while being in immersive VR systems, allowing them to see their own body or others’ (NAHON; SUBILEAU; CAPEL, 2015). In fact, AV techniques can be used to replace or improve avatars in virtual environments, providing to them some characteristics from reality. Whether this will be limited to changing the model material colors to match real people (HU; WU; ZHOU, 2015) or to insert 3D reconstructed textured objects is just a matter of the application’s needs.

Although not limited to the insertion of sensorial information, according to the ISO definition, in this work we’re mainly interested in the insertion of physical world images into the virtual worlds. When matching positions from both spaces, this process is called 2D–3D registration. However, there are also some kinds of applications in which the rectangular region containing the image is either fixed at a given position and orientation or manually adjusted. One classical example is 3D Videoconferencing (REGENBRECHT et al., 2003), in which each user can view other users’ images in a virtual meeting room (Figure 2.2). Another example concerns remote built environments tele-inspection (WANG; CHEN, 2008).

Figure 2.2: 3D videoconferencing.



Source: Adapted from (REGENBRECHT et al., 2003).

Some recent uses of Augmented Virtuality with 2D–3D registration in teleoperation, surveillance, entertainment, and other applications will be presented in Chapter 3.

## 2.2 Perspective Projection and Camera Pose Estimation

The common approach for performing 2D–3D registration is to estimate the camera pose from a set of correspondences between the image and the 3D model, considering the inverse perspective projection for that camera. This section briefly presents the mathematical model of both direct and inverse perspective projections and some popular camera pose estimation algorithms.

### 2.2.1 Coordinate Systems

The perspective transformation and the spatial registration provide mappings between the image space and world space. The former relates to pixels' coordinates, whereas the latter is defined based on a global Cartesian coordinate system (VINCE, 2006, p. 77). Unfortunately, there is no universally accepted standard for the 3D axes definition among software packages. Some developers and authors use right-handed coordinates (AUTODESK, n.d.) while others use left-handed coordinates (UNITY TECHNOLOGIES, 2020). Furthermore, there is neither a consensus whether the vertical direction should be the  $z$  or the  $y$ -axis. This fact, however, can be explained by historical reasons: computer graphics, which started with 2D graphics, already used the  $y$ -axis for the screen vertical direction, so  $z$  has become depth in such systems. This section presents the coordinate systems used in this study, all defined in the  $\mathbb{R}^3$  space for convenience.

**Image homogeneous coordinate system (ICS)** In this coordinate system, a point  $Q = (q_x, q_y, \lambda)$ , with  $\lambda \neq 0$ , refers to a pixel in the image, located at  $(q_x/\lambda, q_y/\lambda)$ . The vector-valued function  $\eta(\mathbf{g})$  is used in this text to denote homogeneous coordinates normalization (2.1):

$$\eta(\mathbf{g}) = \frac{1}{\lambda} \cdot \mathbf{g} = \begin{bmatrix} g_x & g_y & 1 \end{bmatrix}^T. \quad (2.1)$$

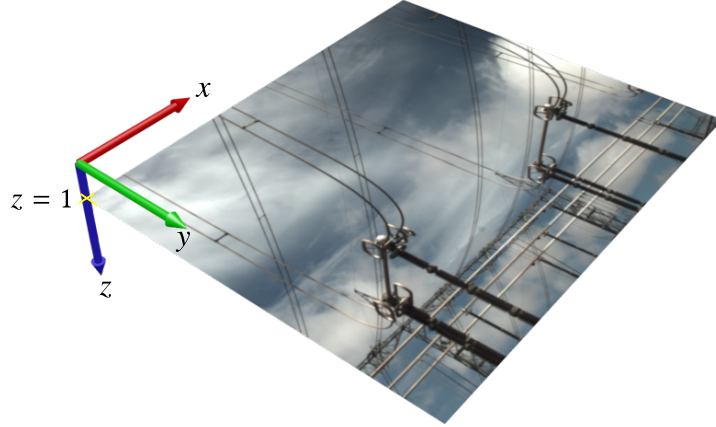
The  $z$ -axis points into the image, making the coordinates right-handed. Figure 2.3 shows the axes, associated with a sample image, representing normalized coordinates, i.e., with  $\lambda = 1$ .

**World coordinate system (WCS)** The world positions are described with a standard right-handed (counter-clockwise) coordinate system, with the  $z$ -axis in the vertical direction.

**Game engine's coordinate system (GCS)** The software package used for composing the substation scene, namely Unity 3D (UNITY TECHNOLOGIES, 2020), has a left-handed (clockwise) coordinate system with the  $y$ -axis in the vertical direction. Therefore, all results expressed in WCS must still be transformed to the engine's coordinate system.

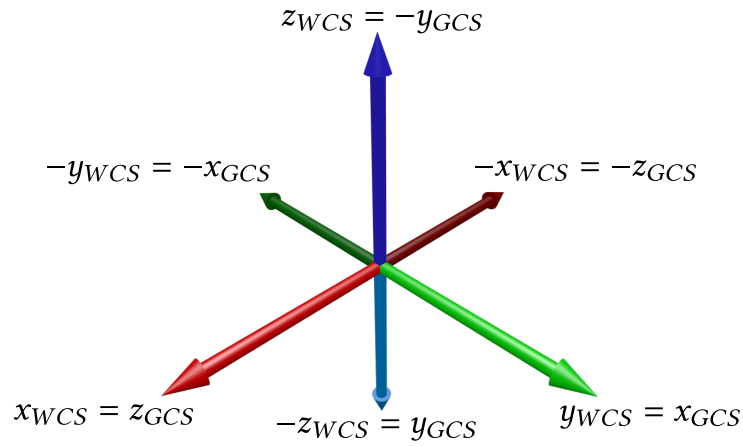
Figure 2.4 shows the relation between WCS and GCS axes.

Figure 2.3: Image coordinate system (ICS).



Source: The author.

Figure 2.4: World's Z-Up and Engine's Y-Up coordinate systems.



Source: The author.

### 2.2.2 Perspective Projection

The perspective projection is a particular kind of linear transformation, capable of mapping points from world space to their correspondents in image space. Let  $P = (p_x, p_y, p_z)$  be a point defined in world space and  $Q = (q_x, q_y, \lambda)$  be the homogeneous coordinates of the pixel that is the result of the perspective projection of  $P$  in the image plane.

Considering the finite projective camera model (HARTLEY, 2004, p. 154–157), this transformation can be stated as:

$$\begin{bmatrix} q_x \\ q_y \\ \lambda \end{bmatrix} = C [R \mid t] \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}, \quad (2.2)$$

where

$C$  is the  $3 \times 3$  matrix of the camera intrinsic parameters, explained below and

$[R|t]$  is the  $3 \times 4$  joint rotation-translation matrix divided up into the  $3 \times 3$  rotation matrix  $R$  plus the translation vector  $t$ .

The camera matrix  $C$  is given by:

$$C = \begin{bmatrix} f_x & \tau & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.3)$$

where

$(f_x, f_y)$  are the focal lengths,

$\tau$  is the skew coefficient between the x and the y axis and

$(c_x, c_y)$  is the optical center (principal point).

Some simplifications might be applied to special cases (HARTLEY, 2004, p. 154–157). If pixels are squares, we can consider  $f_x = f_y$ . If there is no skew effect, then  $\tau = 0$ . Besides, if the origin of the image coordinate system is located precisely at the image center, then  $c_x = c_y = 0$ .

### 2.2.3 Camera Pose Estimation

Let  $\mathbb{I}$  be the set of image pixels' homogeneous coordinates and  $\mathbb{W}$  be the set of world space points. We are interested in a set of  $n$  points in the image,  $\{Q_i \in \mathbb{I} | 1 \leq i \leq n\}$ , and  $n$  points in world space,  $\{P_i \in \mathbb{W} | 1 \leq i \leq n\}$ , such that for each  $i$  there is a unique correspondence ( $P_i \mapsto Q_i$ ). Stated another way, suppose we have both image homogeneous coordinates (ICS) and their related world space coordinates (WCS) for some keypoints.

The problem of estimating the joint rotation-translation matrix,  $[R|t]$ , from the keypoints and the camera's intrinsic parameters  $C$  is called Perspective-n-Point (FISCHLER; BOLLES, 1981). This is particularly useful for establishing mappings ( $P_i \mapsto Q_i$ ),  $\forall P_i \in \mathbb{W}$  and  $\forall Q_i \in \mathbb{I}$ , that is, not only for the keypoints, but also for all other resulting image pixels coordinates.

Equation (2.2) can be reorganized splitting the joint rotation-translation matrix,  $[R|t]$ , and adjusting the matrices dimensions:

$$\begin{bmatrix} q_x \\ q_y \\ \lambda \end{bmatrix} = C \cdot R \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} + t. \quad (2.4)$$

Once the pose is estimated, equation (2.4) can be used to evaluate the homogeneous coordinates of the image pixel, given the coordinates of the point in world space.

The pose estimation is especially interesting for augmented and mixed reality applications since it allows the computation of a virtual object's pose in the image coordinate system.

However, even if the intrinsic parameters are unknown, they can still assume values, due to some further simplifications, as explained in Section 2.2.2, treating the camera as if it were almost ideal. In such a scenario, distortions are ignored. The principal point is defined in the image center and the focal length elements on the camera matrix assume the same value, proportional to one of the image dimensions.

César et al. (2011) compared several Perspective-n-Point (PNP) algorithms, revealing the techniques known as Efficient Perspective-n-Point Camera Pose Estimation (EPnP) (LEPETIT; MORENO-NOGUER; FUA, 2009) and Pose from Orthography and Scaling with Iterations (POSIT) (DEMENTHON; DAVIS, 1995) as the most robust. Both methods require the coordinates of four or more non-coplanar keypoints in the virtual world space and their corresponding coordinates in the image. The EPnP brings a non-iterative solution, of complexity  $O(n)$ , from the evaluation of a weighted sum of eigenvectors of a  $12 \times 12$  matrix and the solution of a constant number of quadratic equations to adjust the weights. In contrast, the POSIT technique first estimates the object's pose by solving a linear system. After this first estimation, the algorithm enters a loop where the parameters from a previous iteration are used to re-calculate the keypoints projections, which will be used instead of the original ones to repeat the pose estimation, resulting in a, presumably, more accurate result. Recently, a new method for obtaining these camera parameters without having points' correspondences has been proposed (BROWN; WINDRIDGE; GUILLEMAUT, 2019). Yet, the original Random Sample Consensus (RANSAC) algorithm, proposed by Fischler and Bolles (1981), has been recently adapted for processing aerial video (ZHENG et al., 2021). The system described in this text uses the iterative PNP solver offered by the OpenCV library (BRADSKI, 2000; KAEHLER; BRADSKI, 2016).

## 2.3 Power Substation Remote Inspection

The electrical power grid needs constant operation from the energy companies so that the following goals are achieved:

- to continuously supply power, according to the system load, keeping power generation balanced with power demand;
- to provide high-quality energy, coping with standards for allowed voltage and frequency variations and
- to fulfill these requirements with low costs.

An electrical grid power system is undoubtedly a very complex system, due to the number of variables requiring measurement, interacting with each other, and influencing dispatch decisions. Hence, the idea of having a centralized site for managing the widespread assets is much compelling. Although SCADA systems are often associated with factories, its concept suits the needs of the power grid field. According to Bailey and Wright (2003):

“SCADA refers to the combination of telemetry and data acquisition. SCADA encompasses the collection of the information, transferring it back to the central site, carrying out any necessary analysis and control, and then displaying that information on a number of operator screens or displays. The required control actions are then conveyed back to the process”.

However, telemetry is error-prone, and sometimes the measured values cannot reveal the cause of the issue, especially because not all environmental variables are known to SCADA. In a factory, this would mean having a worker or a group analyze the situation *in loco*, to understand and solve the problem as soon as possible. For tele-assisted power substations, the costs are significantly higher, since the distances are bigger and, depending on the case, an entire city may become off-line until the problem is solved.

Real-time substation remote inspection has other unique characteristics, in comparison to other remote surveillance systems. In particular:

- thermal images can be used stand-alone or along with RGB images;
- power disconnectors states can be inferred by image processing routines and compared against data reported by the SCADA system, detecting inconsistencies and triggering alarms.

Considering the integration with VR, through 2D–3D registration, we can also mention the following features:

- once positioned in the virtual environment, the photos, together with the geometrical model, can act as input to specialist systems aimed at aiding decision-taking;
- specific scenarios, such as fire, electric arc, and smoke can be detected and, thanks to the virtual environment, be geolocated rapidly.

## Related Work

This chapter presents some Augmented Virtuality applications as well as some recent research on substation automatic inspection through computer vision.

### 3.1 Review Method

To search for publications related to the proposed system, the following libraries were selected:

- IEEE XPlore Digital Library
- IEEE Computer Society Digital Library
- ACM Digital Library
- Springer Link

First, exploratory research was made, using terms such as monitoring, augmented virtuality, image superposition, pose estimation, and unattended substation.

Then, more directed research was made on the same papers' bases. The results from the exploratory review were used to give some background on common techniques, which were used to refine the search terms.

The main search queries used for retrieving related work are listed in Table 3.1.

Since an augmented virtuality system is a type of mixed reality system, according to recent standards (ISO, 2019), the term “mixed reality” was considered in all queries containing the “augmented virtuality” term.

Also, each conceptual query was adapted for the syntax and limitations of each search engine. For too broad queries, the results were filtered then using the publication dates, keeping only entries published since 2016. Older papers and books were eventually used for better comprehension of traditional algorithms or for providing formal definitions of some concepts.

Table 3.1: Search terms.

<b>Alias</b>	<b>Query</b>
avmr-inspection	("augmented virtuality" OR "mixed reality") AND (inspection OR monitoring OR surveillance OR "video-surveillance" OR "video-monitoring")
avmr-scada	("augmented virtuality" OR "mixed reality") AND (scada OR supervisory)
avmr-substation	("augmented virtuality" OR "mixed reality") AND substation
registration-inspection	("spatial registration" OR "image registration") AND (inspection OR monitoring OR surveillance OR "video-surveillance" OR "video-monitoring")
registration-substation	("spatial registration" OR "image registration") AND substation

*Source:* The author

The number of entries for each database is presented in Table 3.2.

Table 3.2: Unfiltered results.

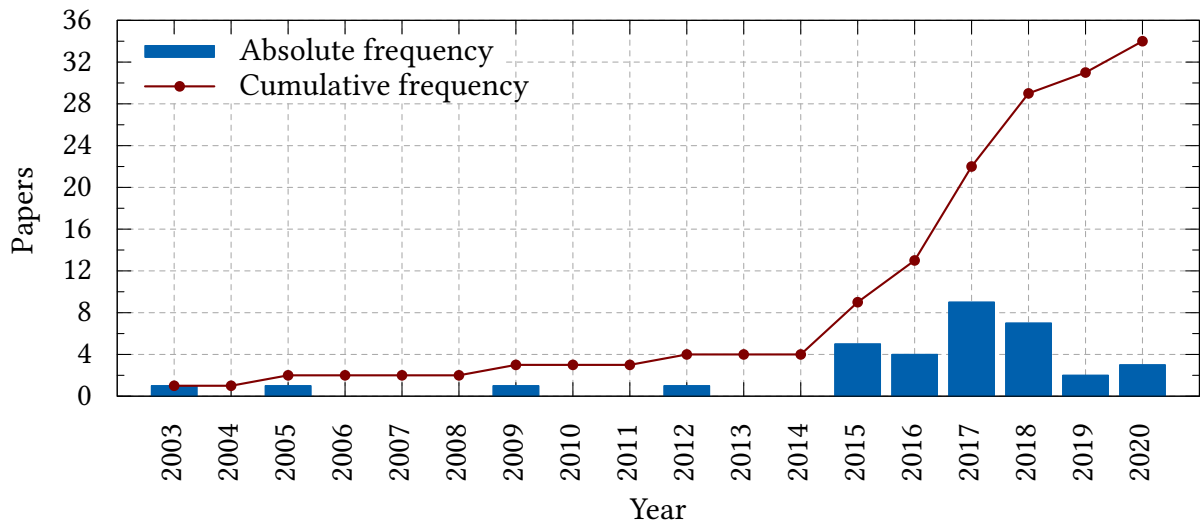
<b>Alias</b>	<b>IEEE XPlore</b>	<b>SpringerLink</b>	<b>IEEE CS</b>	<b>ACM</b>
avmr-inspection	215	339	1285	702
avmr-scada	7	135	423	202
avmr-substation	0	19	449	2
registration-inspection	332	2529	1098	339
registration-substation	6	16	260	0

*Source:* The author

It should be noted that, since the term "mixed reality" was included in all queries related to "augmented virtuality", many papers discussing AR were retrieved. In fact, no paper effectively describing the application of AV for power substations was found during our review.

The entries were then filtered by their titles, and then by their abstracts. The remaining set was used to extract related work, along with some of these articles' references. The citations of this chapter have the time distribution shown in Figure 3.1, except for software-related citations.

Figure 3.1: Bibliography entries time distribution.

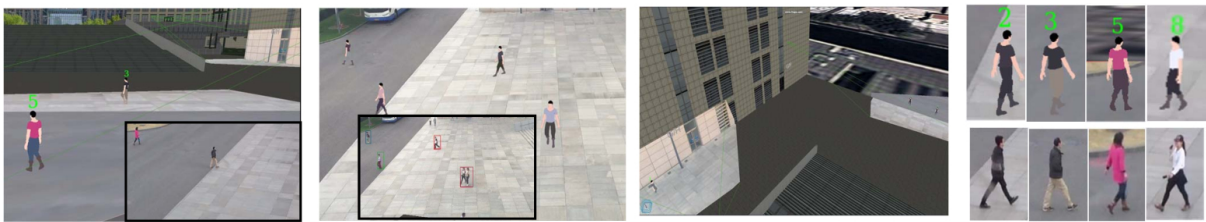


Source: The author.

## 3.2 Video monitoring and Real-Time Inspection

Augmented virtual environments have already been used in urban video monitoring applications (SEBE et al., 2003). Hu, Wu, and Zhou (2015) propose the control of humanoid virtual models' positions according to real humans' positions captured by video cameras, for outdoor environments (Figure 3.2). For representative purposes, sending only positions and orientations instead of the full images is an interesting strategy since it demands much fewer network resources, comparing to real-time video streaming.

Figure 3.2: Humanoid models mimicking humans.



Source: (HU; WU; ZHOU, 2015).

When multiple cameras are installed in far remote locations, a guided tour in the monitored environment is peculiarly interesting to aid operation. Scene-graphs might be put in place to this situation (XIE et al., 2016).

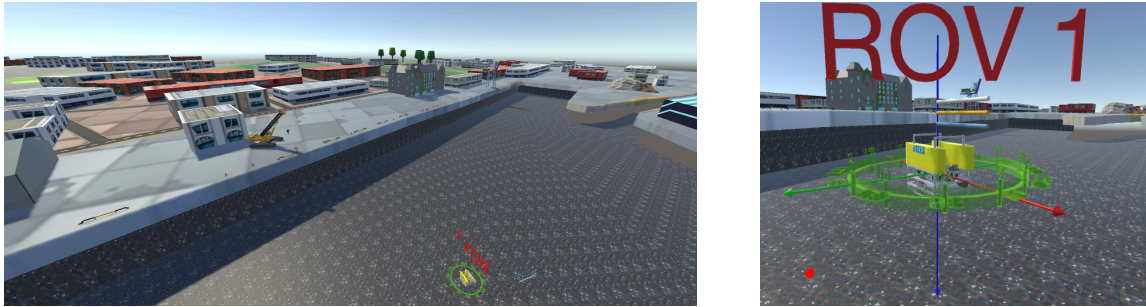
A similar approach of inserting physical reality information into virtual environments is proposed for the supervision of marine systems (Figure 3.3). The system deals with the problem of developing a SCADA VR-based interface that reduces sensory overload and “provides situation awareness while maintaining operator capabilities” (NAĐ; MIŠKOVIĆ; OMERDIC, 2019).

Figure 3.3: VR/AV technology for marine systems.

(a) Classical SCADA Interface.



(b) VR Avatar with attitude, speed and location indicators.



Source: Adapted from (NAĐ; MIŠKOVIĆ; OMERDIC, 2019).

Concerning power tele-assisted substations, video monitoring systems with automatic image analysis are important inspection tools. Color images can be submitted to algorithms capable of detecting people (KIM et al., 2018), fire (SHI et al., 2017; LIU et al., 2015), people climbing ladders in forbidden areas (Figure 3.4) (WANG; AN, et al., 2017), as well as oil leakage in power transformers and unwanted objects left in their nearby (CHANGFU; BIN; FENGBO, 2017).

Figure 3.4: Forbidden ladder climbing detection.

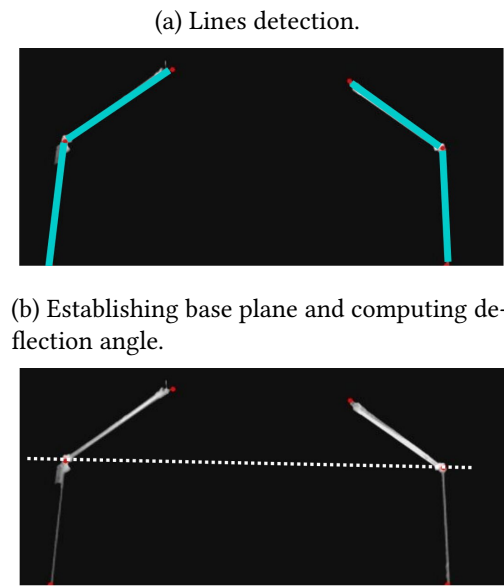


Source: (WANG; AN, et al., 2017).

Another possibility is to automatically infer equipment state from their images. In this manner, Pereira et al. (2016) proposed a way of inferring disconnecter switches states, by using a method

that consists of (i) extracting a region of interest, comprising the mobile parts of the device and the axes supporting them; (ii) applying a threshold, so that the background is removed; (iii) establishing line equations through linear regression; and (iv) checking the deflection angles to judge as either opened or closed (Figure 3.5). A more advanced method using convolutional neural networks is also being tested by the researchers. A similar work, but targeting devices with lower voltages, has been published by Nassu et al. (2018). The solution proposed in our work uses the images and the inferred states from the system developed by Pereira et al. (2016).

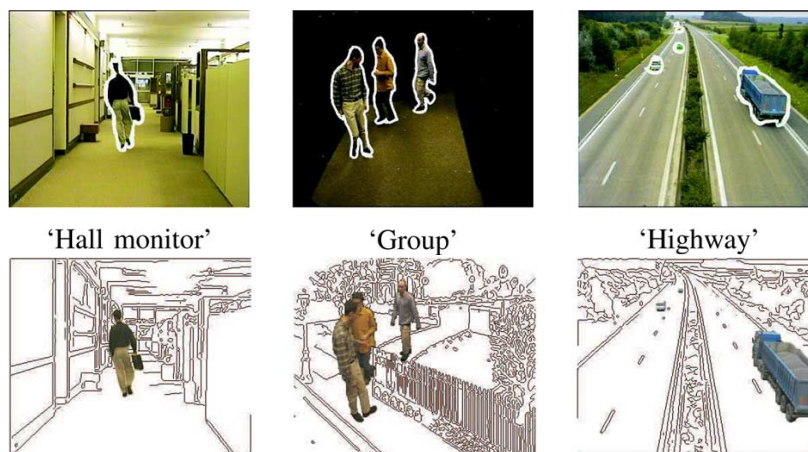
Figure 3.5: Checking deflection angle.



Source: Adapted from (PEREIRA et al. 2016).

One common strategy for the detection of irregularities is the foreground-background segmentation (Figure 3.6).

Figure 3.6: Foreground-background segmentation



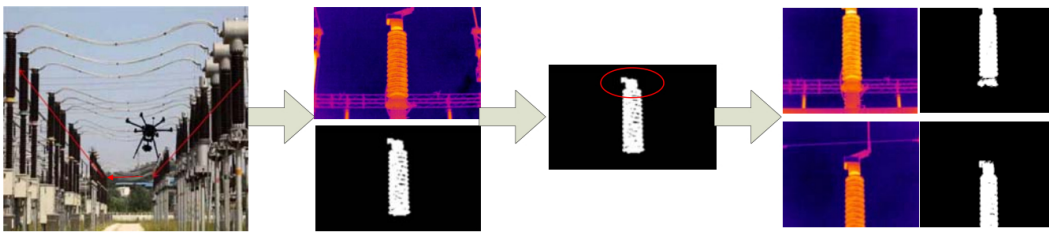
Source: (GELASCA; EBRAHIMI, 2009)

Image segmentation is the computer vision process in which an image is “broken into some non-overlapped meaningful regions” (SEVAK et al., 2017). In particular, the foreground-background segmentation is, for videos, the separation of what is moving from what is static (LI; ZHANG; LIANG, 2018), while as for images, the system queries a historical image database to define what is considered to be the image background (CHANGFU; BIN; FENGBO, 2017).

This technique can be used to detect motion in an equipment nearby region or objects that, according to historical data, should not be there. Some researchers even suggest the adoption of actions such as deactivating remote control in an area whenever an object’s motion is detected there (LUO; TU, 2005, p. 64).

Thermal images are equally important, since “the thermal effect of power devices is one of the major reasons leading to faults” (XIAOMING; SHAOSHENG; BING, 2012). Drones with thermal cameras have already been deployed to scan faults in substations (Figure 3.7), storing pictures of insulators, which are later on processed for failure diagnostics (LV et al., 2017).

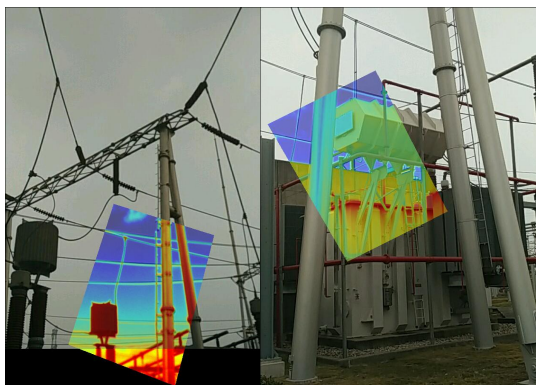
Figure 3.7: Drone-based remote inspection.



Source: (LV et al., 2017).

Also, a recent work published by Jiang et al. (2020) describes image-to-image registration for combining infrared and visible images of power equipment (Figure 3.8). The fusion of infrared and visible light images is also discussed by Sun, Qiu, and Sui (2020).

Figure 3.8: Image-to-image registration applied to substation monitoring.



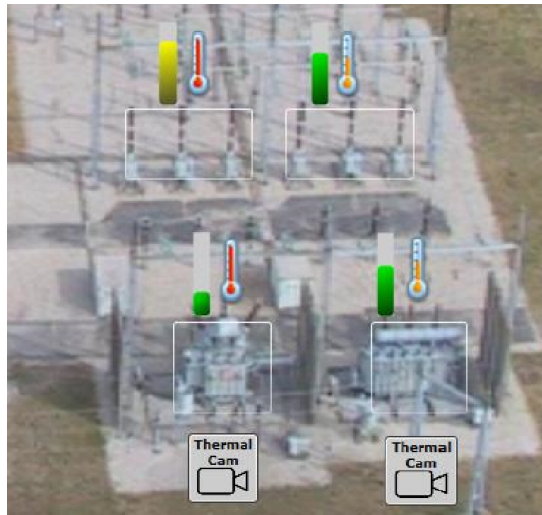
Source: (JIANG et al., 2020).

Alternatively, field images can be augmented with thermal sensor data (BUHAGIAR et al., 2016),

as shown in Figure 3.9. However, since telemetry-data is error-prone, in this case, the inspection will be restricted to failures visible in the captured color images. It should be noted that, even with this limitation, the proposed system integrates SCADA data with video monitoring.

Augmented reality can also be used by field operators to better visualize contextual SCADA data (ANTONIJEVIĆ; SUČIĆ; KESERICA, 2018), although the scope of this work is the opposite integration: having more field information in the operations centers.

Figure 3.9: Video monitoring image augmented with virtual thermometers.



Source: Adapted from (BUHAGIAR et al., 2016).

Equipment inspection can also be realized by inspector robots (XIAO-LE et al., 2017). Considerably more complex, these systems (Figure 3.10) combine (i) the needed advanced techniques for the design of autonomous robots, such as route planning, collision detection, battery management, environment mapping, and information fusion, (ii) machine learning and (iii) failure detection using computer vision routines.

Figure 3.10: Substation inspection robot.



Source: (LU; ZHANG; SU, 2018).

Finally, when dealing with cameras equipped with PTZ (pan-tilt-zoom) control, a common scenario is to capture multiple views, multiplexed in time, so that more than one asset can be

monitored (NASSU et al., 2018). This approach suffers from a limitation: servomotors' motion, periodically changing their setpoints to allow the different poses, generate cumulative errors that must be constantly compensated. Online camera calibration has been already evaluated for substation video monitoring (CAI et al., 2017), as well as 2D–2D registration for coordinates correction (NASSU et al., 2018).

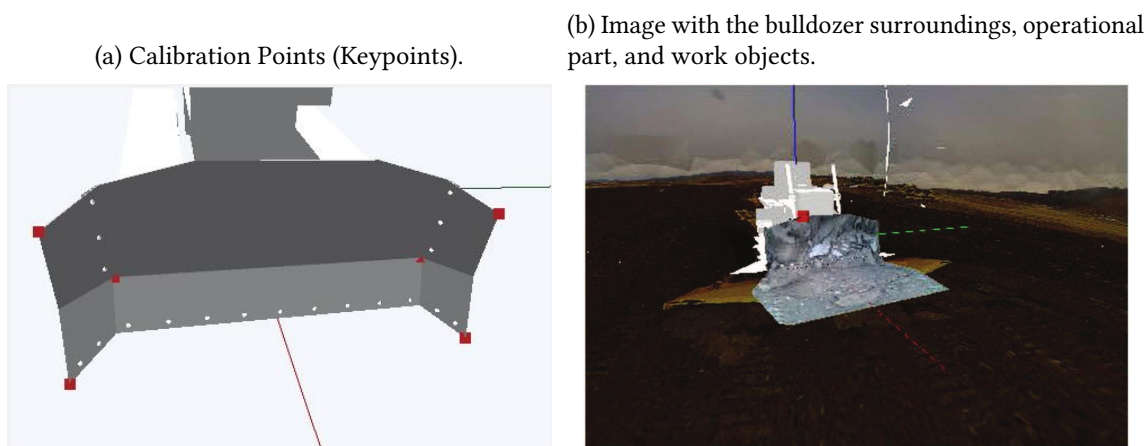
More on the subject of image-based power equipment monitoring can be found in the work published by Liang et al. (2020).

### 3.3 2D–3D Spatial Registration

The technique of superimposing images into a three-dimensional model and matching the positions from both spaces into the mixed environment is called 2D–3D spatial registration. If the camera's pose is known at the instant the photo is taken, as well as its intrinsic parameters, such as focal length and optical center, then one can take advantage of an analytical method of inserting the image into the three-dimensional scene, like shown in Figure 1.4b.

Vagvolgyi et al. (2017) has applied texture-mapping to overlay real photos in a 3D object surface, for robot teleoperation purposes. The application uses matching natural landmarks (keypoints) on the photo and the virtual environment and allows a dynamic change of the point-of-view. Another teleoperation application, concerning construction machines, is proposed by Iwataki et al. (2016). The system combines many images capturing the surrounding of a Bulldozer, its operational part, captured by a perspective camera, and work objects, captured by cameras with fisheye lenses. The registration of the operational part is aided with predefined keypoints.

Figure 3.11: 2D–3D registration for construction machines teleoperation.



Source: Adapted from (IWATAKI et al., 2016).

The spatial mapping (registration) of live actors and entities (LAE) into 360° photos (Figure 3.12) is presented by Chheang et al. (2018). The system uses a webcam and a Kinect tracker and applies the image to a cylindrical surface.

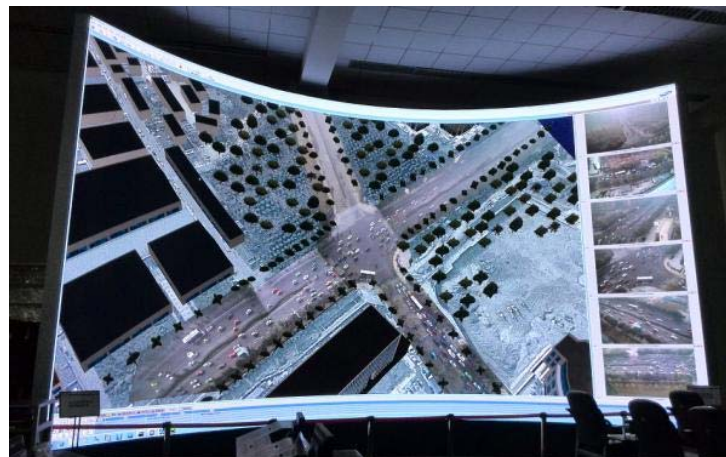
Figure 3.12: Embedding live actors and entities into a 360° photo, with and without chroma key.



Source: Chheang et al. (2018).

Alternatively, some research has been made considering the use of geometrical restrictions to aid the pose estimation. For instance, in outdoor images with humans either walking or stationary, one can (i) assume that they're reasonably aligned with the world's vertical direction, (ii) process the image to obtain the corresponding line equations, and (iii) use this information to estimate the camera's pose (FÜHR; JUNG, 2017). In the case of power substations, static elements such as power disconnectors' insulators and porticos can be used for the same goal. The 2D–3D registration techniques may also be applied in systems with multiple cameras. Wu et al. (2015) propose a framework for the fusion of large-scale surveillance images with an associated virtual environment. The system combines, in the same view, a mosaic with images captured by the surveillance cameras, a panoramic image sent by satellite equipment, and the corresponding three-dimensional model (Figure 3.13).

Figure 3.13: Video surveillance image fusion.



Source: (WU et al., 2015).

Bui et al. (2015) present an augmented virtual environment with the 2D–3D registration of videos in a model generated by the Light Detection and Ranging (LIDAR) method. The system integrates videos submitted by civilians into the scene, providing for better response to disasters, helping authorities define the damage extension and how to access the affected areas.

The game market also presents a potential for Augmented Virtuality development. From a user perspective, there is a negative effect of not seeing his own body or the others in a virtual environment, suggesting the insertion of this data into the game (NAHON; SUBILEAU; CAPEL, 2015). Remote users can play an immersive game in which the virtual world is augmented with realistic elements (Figure 3.14), including other users' virtual replica by real-time 3D reconstruction (KARAKOTTAS et al., 2018).

Figure 3.14: Augmented virtuality game experience.



Source: (KARAKOTTAS et al., 2018).

### 3.4 Comparative Analysis

The papers presented in Section 3.2 have many methods and techniques for the inspection of environments and equipment, by image processing and computer vision. Image registration applications, especially 2D–3D registration, are given in Section 3.3. Considering the objectives of the present research work, the following system features were analyzed:

**Camera Mobility** Some applications are best suited for mobile cameras while others are restricted to fixed cameras.

**Color Model** Although most 2D–3D registration applications handle RGB images, thermal image analysis can be critical to some inspection systems, such as power substations.

**SCADA Integration** To reduce sensory overload from the operator, it is desired that, whenever an equipment's state is inferred from an image, this data is automatically compared against the real-time state reported by SCADA, triggering alarms when appropriate.

**Computer Vision Failure Detection** Image registration is an important part of many Augmented Virtuality systems, but remote inspection doesn't need to be done by the operators without the help of computer vision systems.

**Multiple Scenes and Multiple Cameras** The proof of concept for the solution is power substations remote inspection. Thus, the system's architecture must allow multiple scenes (multiple substation installations) and multiple cameras for each scene.

Table 3.3 shows an overview of these features for the two groups of applications and the insertion of the proposed solution in contrast to the existing applications.

Table 3.3: Features table.

	Camera Mobility	Color Model	SCADA	AR/AV	CV Failure Detection	Multiple Cameras / Scenes
<b>Remote Inspection</b>						
(LV et al., 2017)	Mobile	Thermal	No	None	Yes	No
(LU; ZHANG; SU, 2018)	Mobile	RGB / Thermal	No	None	Yes	No
(JIANG et al., 2020)	Fixed	RGB / Thermal	No	None	No	
(BUHAGIAR et al., 2016)	Fixed	RGB/Thermal	Yes	AR	No	Yes
<b>2D–3D Registration</b>						
(VAGVOLGYI et al., 2017)	Mobile	RGB	No	AV	No	No
(IWATAKI et al., 2016)	Fixed <sup>a</sup>	RGB	No	AV	No	No
(WU et al., 2015)	Fixed	RGB	No	AV	No	Yes
<b>This work</b>	Fixed	RGB /Thermal	Yes	AV	No <sup>b</sup>	Yes

<sup>a</sup>The camera's relative position to the vehicle is fixed.

<sup>b</sup>Although the feature is not implemented on our solution, it can be easily added by using special SCADA nodes (see Section 5.6.3)

## 3.5 Conclusions

Video monitoring and surveillance systems have been widely used and evolved to fulfill tele-assisted substations requirements. The acquired images, besides being used in manual inspection in Operation Centers, can be fed to image processing algorithms for the detection of equipment state and faults. At the same time, advanced human-machine interfaces, based on Virtual Reality and 2D–3D Registration techniques have been evaluated in several research works. However, despite these works indirectly indicate the benefits of augmented virtuality, the case studies are limited to civilian monitoring, some other outdoor environments, and

indoor mobile robot teleoperation. There is a clear opportunity of evaluating augmented virtuality for remote inspection of power substations, associated with SCADA systems.

Additionally, the integration of video-monitoring systems with the substation virtual environments represents a considerable opportunity in usability. Contextual information provided by the physical layout, which is not available neither in the image limited scope nor in the traditional operation diagrams, allied to the features of automatic inspection, improves the communication with field personnel and may result in better response times for the electrical system reestablishment.

# Problem-Solving Methodology and System Architecture

This chapter describes the method applied to achieve the proposed research goals.

One main concern of mixed reality applications is the registration process. All other system features may become useless if good registration is not achieved. In augmented virtuality applications with camera pose estimation, many steps are needed to finally get the image well-positioned and oriented in the virtual environment. In terms of implementation, debugging becomes much less complex if each step is tested and validated separately before the next one is developed.

With this approach in mind, the problem was split into smaller prototypes, each one serving as a solid foundation for developing the next. The proposed method has 5 phases, briefly enumerated in the following sections.

## 4.1 Mathematical Model of the 2D–3D Registration

The main concern of Perspective-n-Point (PNP) solvers is to estimate the camera’s extrinsic parameters, namely the joint rotation-translation matrix  $[R|t]$  from Equation (2.2). However, the rectangular region in which the captured field images will be rendered must still be positioned and oriented. Thus, it is reasonable to start the development by constructing a mathematical model of the full 2D–3D registration. This has some advantages, considering that “when the system model (or part of it) can be solved with analytical methods, considerable gains in terms of efficiency, accuracy, and understanding are usually obtained” (POHJOLAINEN; HEILÖ, et al., 2016).

## 4.2 PNP-based Perspective Projection and Inverse Perspective Transform

Before implementing the full 2D–3D registration algorithm, the PNP solver must be validated. A minimalist approach to this assertion consists of the following steps:

1. create an abstract and minimalist virtual environment with some geometrical entities to define keypoints for a Perspective-n-Point study;
2. extract keypoints in world coordinates;
3. position the virtual environment's camera at a given pose and take a screenshot;
4. extract keypoints in image coordinates;
5. develop a GUI application suitable for running Perspective-n-Point experiments;
6. run the pose estimation;
7. use the estimated matrix in perspective projection transformations, having the keypoints in world space as inputs;
8. assert that the obtained pixel coordinates are close enough to the ones collected in step 4, by applying distance metrics.

The intentions for the suggested software prototype (step 5) are (i) to make easier and less error-prone the execution of the next steps and (ii) to already have some partial implementation of the mathematical model previously constructed. Finally, some implementation regarding the inverse perspective transform, i.e., mapping from image pixels to world lines of sight, might also be added to the prototype.

## 4.3 Loopback Registration Experiment

Once the camera's joint rotation-translation matrix estimation is validated, another interesting prototype is a solution in which a screenshot is taken from the virtual environment camera view and then handled as if it was an actual field photo. That can be accomplished with the steps below:

1. create a virtual environment with a single instance of a virtual model and a rectangular region for receiving the image overlay;
2. define a convention for naming the keypoints, so that no ambiguity exists;

3. extract keypoints in world coordinates, adding the metadata to the virtual environment application;
4. run the VR application and position the virtual camera in some arbitrary position and orientation;
5. take a screenshot and extract keypoints in image coordinates, with the help of an *ad hoc* GUI desktop application, following the convention established in step 2;
6. trigger the 2D–3D registration request from the VR application;
7. automatically put the virtual camera and the rectangular region (overlay) in their estimated poses;
8. evaluate errors in keypoints’ projections by using the same metric defined in the previous experiment;
9. compare camera poses in terms of Euclidean distance between positions and some distance function between rotations;
10. evaluate errors in the image rendered by the augmented virtual environment’s camera, considering the pixel coordinates of both the virtual model keypoints and the overlay.

Although steps 8 and 10 may benefit from using the same quantitative metrics, they do not refer to the same study. Let’s consider that the PNP solver finds a good perspective projection transform for a given camera model, having small errors while mapping from world coordinates to image pixels. That camera model might have different intrinsic parameters from the target’s (virtual environment) camera, notably regarding the field of view, resulting in bad registration quality. Therefore, small errors in step 8 are required but not sufficient to achieve high-quality 2D–3D registration.

## 4.4 Field Photo Registration – Augmented Virtuality Prototype

This module is the first solution overlaying a field photo in the virtual environment. The implemented prototype serves as a foundation for the development of more features, like multiple cameras, equipment state inference, and SCADA integration.

For that goal, the following actions are performed:

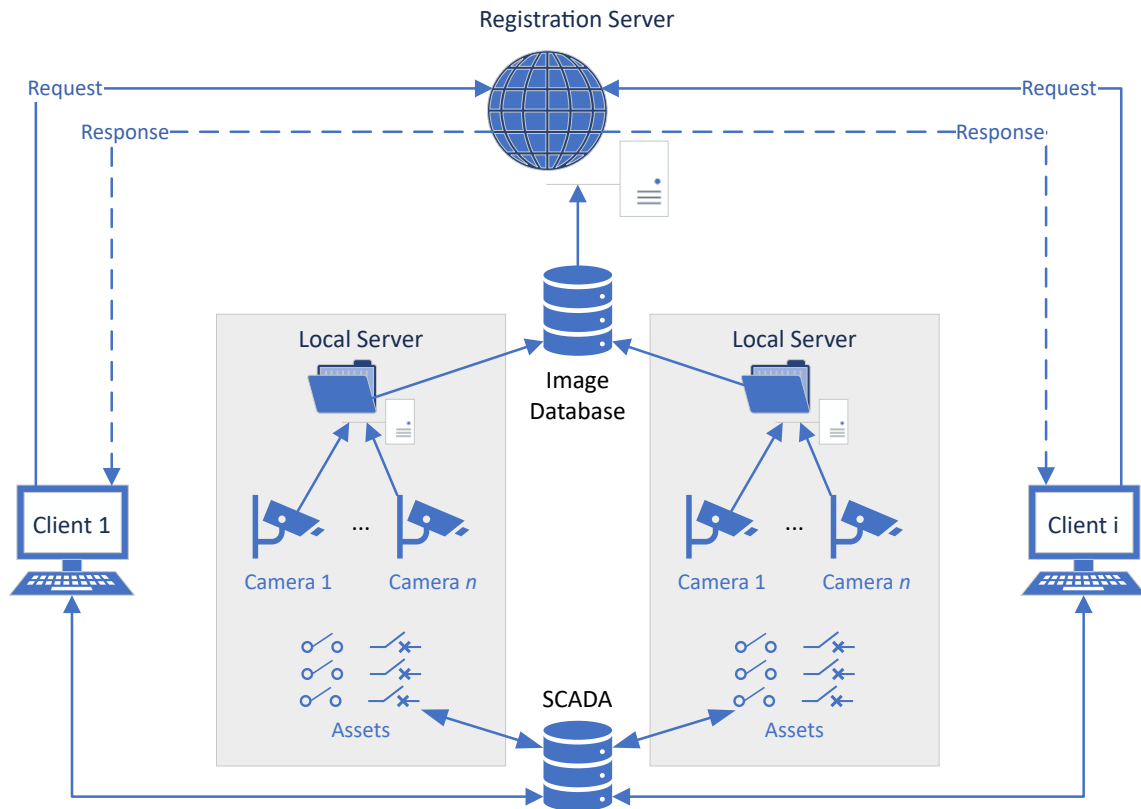
1. construct or acquire a virtual environment for the experiment, for which one or more field photos are available;

2. extract keypoints world space;
3. select a target field photo for the experiment, extracting the keypoints coordinates in image space;
4. perform the 2D–3D registration using the same mechanisms of the previous solutions;
5. evaluate the quality of the perspective projection model and the image overlay.

## 4.5 Augmented Virtuality for Remote Inspection

This last solution is aimed at features that will enable the handling of multiple scenes and cameras, combined with an alarm interface for a quick inspection. A basic component view for this approach is presented in Figure 4.1.

Figure 4.1: Augmented virtuality for remote inspection -- system architecture.

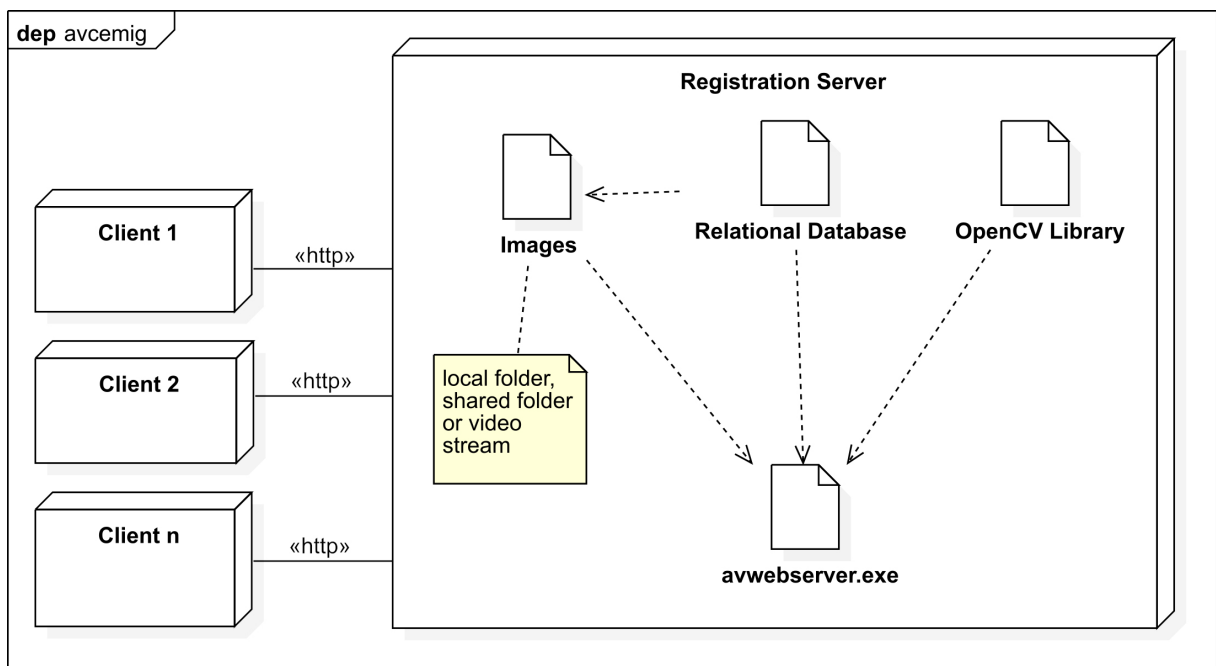


Source: The author.

The solution deals with one or more substations with one or more cameras. Each camera is related to a single asset of interest (circuit breaker, power switch, or transformer). The relation is one-to-many: although many cameras can be associated with the same asset, one single camera does not observe more than one asset. Each substation collects images and other

sensory data, sending the former to an image database and the latter to the SCADA system. Both targets are located in a remote operations center. The remaining nodes of the system architecture are the registration server and clients. The registration server is used to handle 2D--3D registration requests, caching results to improve performance. This mechanism is used to avoid unnecessary processing whenever some client makes a request but no newer image is available for the associated camera. In such a scenario, there is no need to recompute the virtual camera pose again, since the results would be the same. For many cameras and many substations, this is an important requirement, since launching multiple camera pose estimations processes in the server can be expensive and the clients are already taking care of other critical tasks such as real-time rendering and navigation. For fixed cameras, the cache is used almost every time, except during calibration routines or the deployment of new cameras. The registration server node component comprises a relational database, used mainly for storing the remote inspection infrastructure and the estimated poses for the cameras and the overlays, as well as an HTTP server, used to provide an interface to the clients (Figure 4.2). The OpenCV computer vision library (BRADSKI, 2000; KAEHLER; BRADSKI, 2016) is used for running the Perspective-n-Point algorithm.

Figure 4.2: Registration server internal view.

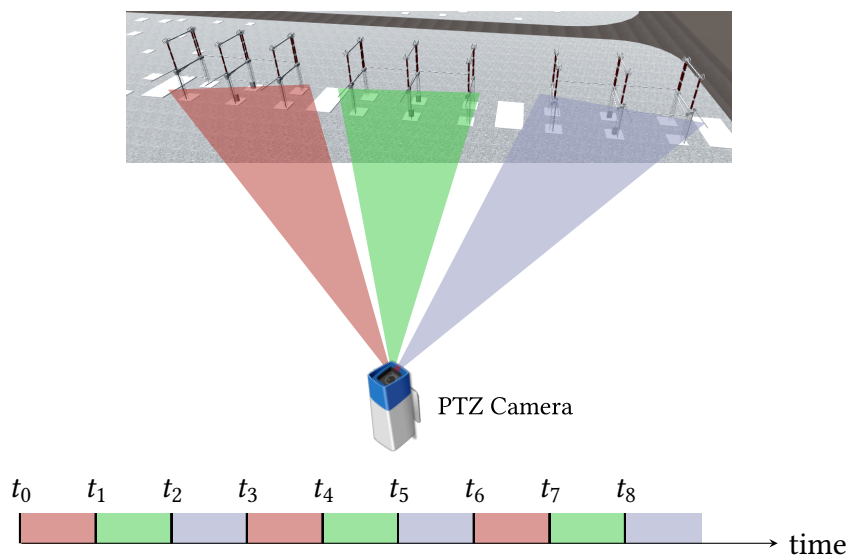


Source: The author.

Also, among the specific needs of this particular application, one critical issue is data transmission between the power substation cameras and the remote operations center. Cameras equipped with pan-tilt-zoom (PTZ) control can monitor multiple areas of interest, resulting in a time-multiplexed scenario (Figure 4.3). Concerning the proposed 2D--3D image registration, though, each pose's preset needs different reference points in the VR space, since they are

associated with distinct assets being monitored. For the remote inspection needs, this means multiple separated overlays for the same source camera. In practice, a single camera acts as if it was many, at the cost of presenting older images but with the benefit of demanding a smaller network bandwidth.

Figure 4.3: Single PTZ-camera with time-multiplexing.



Source: The author.

This kind of multiplexing is adopted in the Nova Ponte's substation, which has one camera (Figure 4.4) for monitoring three distinct disconnectors.

Figure 4.4: Nova Ponte cameras (thermal and RGB).



Source: CEMIG-GT.

Considering the proposed architecture and the objectives for this prototype, the following steps were planned:

1. analyze the solution in terms of system and software, defining an architecture allowing multiple scenes, cameras, field assets, and communication with SCADA;
2. deploy or construct a SCADA simulator to provide and alter equipment states;
3. develop or integrate the system with an image-based state inference engine;
4. implement 2D–3D registration for field images;
5. implement basic alarm functions in the virtual environment, triggered whenever the state inferred from the image differs from the one reported by the SCADA simulator;
6. design a user interface for enabling and configuring remote inspection features through the VR environments;
7. evaluate the impact of using videos (field images at 30 FPS) in the environment’s real-time rendering;
8. evaluate all the registration metrics from the previous solutions, as well as performance-related metrics.

## Development

This chapter presents the mathematical and software development details, considering the steps presented in the previous chapter. The development is directed towards a specific case of 2D–3D registration of electrical substation power disconnect switches, along with the SCADA integration. The registration process begins with the camera pose estimation, followed by an analytical method to place and orient a rectangular region in the virtual environment.

The mathematical model of the 2D–3D registration is presented, along with the prototypes related to each phase of the development, namely:

- *avmath*: a GUI workbench to test perspective projection and inverse perspective transform;
- *avloopback*: 2D–3D registration test solution, treating Virtual Environment screenshots as if they were real photos;
- *avcamera*: augmented virtuality solution for overlaying power disconnect switches photos into the virtual environment;
- *avcemig*: full solution comprising SCADA integration, alarms, multiple scenes, and multiple overlays.

### 5.1 2D–3D Registration Mathematical Model

This section gives details on the math expressions and manipulations used to achieve 2D–3D registration.

Section 2.2.3 described the problem of estimating the camera pose, allowing the retrieval of the complete transformation from world space to image space. Stated another way, the method enables the mapping ( $P_i \mapsto Q_i$ ). The inverse problem, i.e., going from the image space to the world space, is used for augmented virtuality systems and can be done analytically, once the

camera pose is estimated, as shown in this section. It should be noted that more than one camera model can fulfill this mapping by different poses, depending on the focal length elements in (2.3). It is desired, however, to set the image as a texture for a rectangular region overlaid in the virtual environment and to teleport the virtual camera to a pose that is compatible with the VR camera intrinsic parameters, so that the photo and its surrounding virtual environment match optimally.

Let  $\mathbf{p} = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T$  represent the coordinates of a point  $P_i$  in WCS and  $\mathbf{Q} = \begin{bmatrix} q_x & q_y & \lambda \end{bmatrix}^T$  represent a point  $Q_i$  in ICS. It is straightforward manipulate (2.4) to isolate  $\mathbf{p}$ :

$$\begin{aligned} C^{-1} \cdot \mathbf{Q} &= C^{-1}C \cdot R \cdot \mathbf{p} + \mathbf{t} \\ \Rightarrow C^{-1} \cdot \mathbf{Q} - \mathbf{t} &= R \cdot \mathbf{p} \\ \Rightarrow R^{-1} \cdot (C^{-1} \cdot \mathbf{Q} - \mathbf{t}) &= R^{-1} \cdot R \cdot \mathbf{p} \\ \Rightarrow \mathbf{p} &= R^{-1} \cdot (C^{-1} \cdot \mathbf{Q} - \mathbf{t}) \end{aligned} \quad (5.1)$$

Finally, let  $\mathbf{q}$  be defined in ICS to represent the same pixel as  $\mathbf{Q}$ , such that:

$$\mathbf{q} = \eta(\mathbf{Q}) = \frac{1}{\lambda} \mathbf{Q}. \quad (5.2)$$

Here we use the normalization function  $\eta$  described in Section 2.2.1. Then, the right-hand side of (5.1) can be reorganized in the following two terms:

$$\mathbf{p} = \underbrace{R^{-1} \cdot C^{-1} \cdot \mathbf{q} \cdot \lambda}_a - \underbrace{R^{-1} \cdot \mathbf{t}}_b \quad (5.3)$$

Since the scalar  $\lambda$  is not bound to any specific value, the solution set corresponds to a line  $\ell$ , in its parametric form:

$$\ell = \begin{bmatrix} a_x \cdot \lambda - b_x \\ a_y \cdot \lambda - b_y \\ a_z \cdot \lambda - b_z \end{bmatrix}. \quad (5.4)$$

Indeed, many points  $\{P_i\}$  in world space may result in the same projection  $Q_i$ .

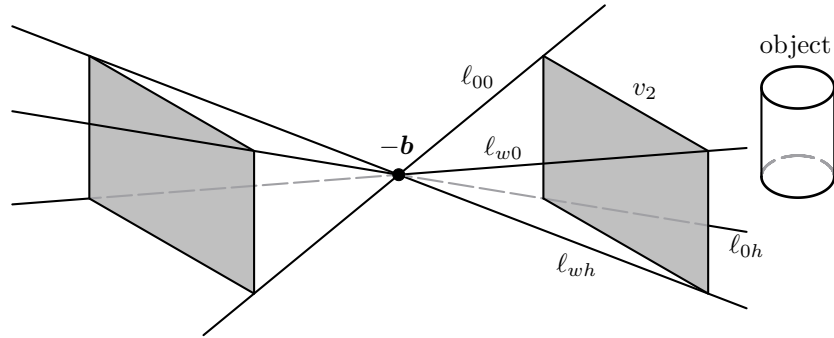
### 5.1.1 Image overlay

The action of overlaying a photo into a virtual environment, for Augmented Virtuality applications, requires determining the pose (position and orientation) of the rectangular region that will display the image in the virtual environment.

Let  $v_1$  be an image of size  $w_1 \times h_1$  dots and  $v_2$  be the rectangular region of size  $w_2 \times h_2$  in world units. Considering that the virtual environment has high geometrical fidelity, we must first assert that the target rectangular region dimensions match the image storage aspect ratio, that is,  $w_1/h_1 = w_2/h_2$ .

Now, from the set of all possible lines extracted from (5.4), by setting values for  $\mathbf{q}$  in (5.3), let us consider  $\ell_{00}$ ,  $\ell_{w0}$ ,  $\ell_{0h}$ ,  $\ell_{wh}$ , obtained by using the image vertices. Finally, let  $\ell_{cc}$  be the line obtained by using the image center point. All these lines, as well as all other lines obtained by (5.1) intercept at  $-\mathbf{b}$ , which corresponds to the estimated position of the camera (Figure 5.1). Indeed, a quick inspection on (5.3) reveals that the term  $-\mathbf{b}$  is equal to  $-\mathbf{R}^{-1} \cdot \mathbf{t}$ , thus it does not depend on the image point  $Q$ .

Figure 5.1: Converging lines and rectangular area  $v_2$ .



Source: The author.

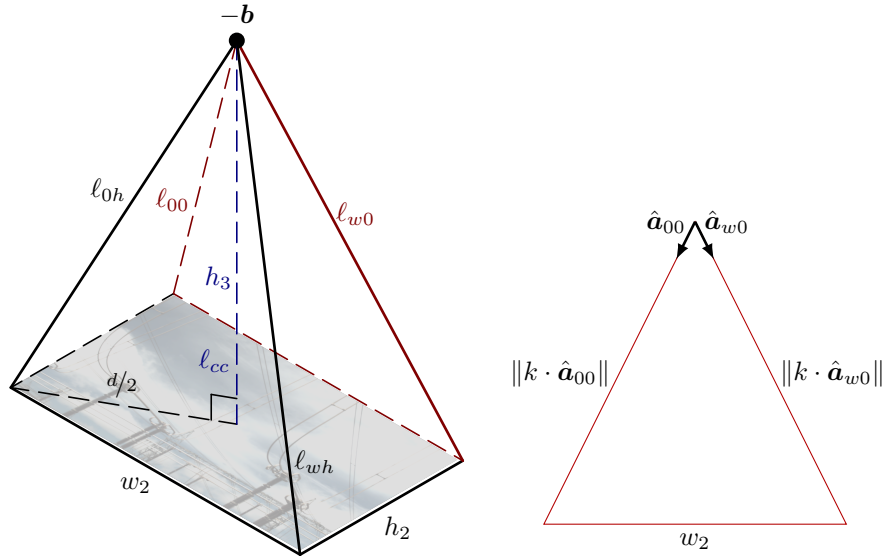
The result is a right bipyramid with a rectangular base and the apex vertex located at  $-\mathbf{b}$ . The relevant pyramid, associated with the camera's viewing frustum, is the one that contains  $v_2$ . Note that this pyramid's base should have the smallest distance to the virtual object. The other pyramid's direction is opposite to the camera and thus is ignored by the solution.

From the known variables and expressions, it is possible to determine the pyramid's height  $h_3$ , as shown in Figure 5.2.

Let  $\ell_{ij}$  denote a line extracted from (5.4), by using some image point  $Q$  specified as the position vector  $\mathbf{q}$  in (5.3). In addition, let  $\mathbf{a}_{ij}$  be the direction vector for line  $\ell_{ij}$  and  $\hat{\mathbf{a}}_{ij}$  be the unit vector obtained from  $\mathbf{a}_{ij}$ . The coordinates of  $-\mathbf{b}$  are also known, from any line  $\ell_{ij}$ . The target rectangular region dimensions,  $w_2$  and  $h_2$ , are also specified. Finally, the pyramid lateral faces are isosceles triangles. It is easy to determine the value of the scalar  $k$  (5.5).

$$\begin{aligned} \|k \cdot \hat{\mathbf{a}}_{00} - k \cdot \hat{\mathbf{a}}_{w0}\| &= w_2 \\ \therefore k &= \frac{w_2}{\|\hat{\mathbf{a}}_{00} - \hat{\mathbf{a}}_{w0}\|} \end{aligned} \quad (5.5)$$

Since the diagonal  $d$  of the rectangular base is given by  $d = \sqrt{w_2^2 + h_2^2}$ , the pyramid's height  $h_3$

Figure 5.2: Elements for calculating  $h_3$ .

Source: The author.

can be equally obtained with the Pythagorean theorem:

$$\begin{aligned} \|k \cdot \hat{a}_{00}\|^2 &= h_3^2 + \left(\frac{d}{2}\right)^2, & h_3 > 0 \\ \therefore h_3 &= \sqrt{\|k \cdot \hat{a}_{00}\|^2 - \left(\frac{d}{2}\right)^2} \end{aligned} \quad (5.6)$$

Then, it is possible to determine the  $v_2$  parameters needed to put it exactly in the pose that the photo was captured, namely the position vector  $\mathbf{u}$  (referring to the point  $U$ ) and the coordinate axes  $\theta_x$ ,  $\theta_y$  and  $\theta_z$  (Figure 5.3).

The direction vector of  $\ell_{cc}$  is normal to the plane that contains  $v_2$  and parallel to the pyramid axis, defining the direction  $\theta_z$ . The other directions can be easily calculated using the unit vectors from the edges. The following equations show all four parameters:

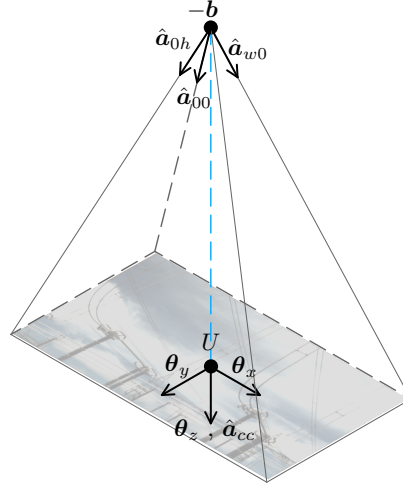
$$\mathbf{u} = -\mathbf{b} + h_3 \cdot \hat{\mathbf{a}}_{cc} \quad (5.7)$$

$$\theta_x = \frac{\hat{\mathbf{a}}_{w0} - \hat{\mathbf{a}}_{00}}{\|\hat{\mathbf{a}}_{w0} - \hat{\mathbf{a}}_{00}\|}, \quad (5.8)$$

$$\theta_y = \frac{\hat{\mathbf{a}}_{0h} - \hat{\mathbf{a}}_{00}}{\|\hat{\mathbf{a}}_{0h} - \hat{\mathbf{a}}_{00}\|} \quad (5.9)$$

$$\theta_z = \hat{\mathbf{a}}_{cc} \quad (5.10)$$

It should be noted that the unit vectors  $\hat{\theta}_x$ ,  $\hat{\theta}_y$  and  $\hat{\theta}_z$  are, in fact, the rows from the estimated

Figure 5.3: Determining the pose of the rectangular region  $v_2$ .

Source: The author.

$R$  matrix. Indeed, the plane containing the rectangular region and the estimated camera have the same orientation.

The obtained position and rotation must be converted to the Game Engine's Coordinate System (GCS), as explained in Section 2.2.1.

That way, the pose of  $v_2$  is determined so that the virtual environment can represent the conditions in which the image was taken. The virtual environment camera is positioned at  $-b$  and oriented according to the  $R$  matrix. Besides eventual distortions and other non-ideal parameters, the physical camera is likely to have a different field-of-view (FoV) from the virtual camera. Hence, the camera matrix is an important source of registration errors. Combined with the Perspective-n-Point solver errors, this can lead to bad quality results. These errors can be reduced by either calibrating the camera or applying some optimization algorithms. The latter approach is discussed in Section 5.6.

## 5.2 Software Tools and Languages

The tools and languages used for developing the prototypes are listed in Table 5.1. The main 2D–3D registration algorithm runs in a Python script, using the function `solvePnp()` from the library OpenCV for the camera's pose estimation, while the Virtual Environment is based on Unity 3D and C#. For the last prototypes, the database was created and managed with the help of SQLite (HIPPI; KENNEDY; MISTACHKIN, 2020).

Table 5.1: Development tools and languages.

	Context
Python 3	Augmented virtuality server
flask	Micro Web framework
OpenCV	Computer-vision library
Sympy	Symbolic mathematics library
Numpy	High-performance numeric processing library
Gtk+	Graphical user interface toolkit (avmath prototype)
tkinter	Graphical user interface toolkit (avloopback prototype)
Unity 3D	Virtual-Environment / Game Engine
C#	Augmented Virtuality Classes for the VR
XML	Markup-language used in HTTP requests and responses
SQLite	Relational database

The virtual environment interacts with a Python-based HTTP server, which performs the 2D–3D registration (or fetches cached results) and then returns a response with the estimated values for the poses of both the camera and the rectangular region. The server was developed using the Flask micro-framework and the SQLAlchemy object-relational mapper (GRINBERG, 2018).

The first prototype (*avmath*) used the Gtk+ toolkit, which was replaced with tkinter for better portability on MS-Windows platforms.

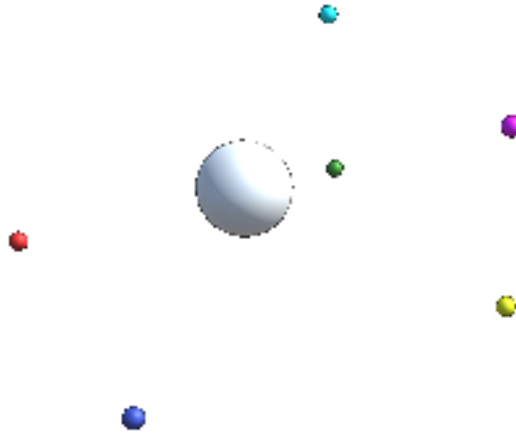
### 5.3 Perspective and Inverse Perspective Transforms Workbench (*avmath*)

The developed solution reads an XML experiment file containing a reference to an image, the keypoints’ coordinates, and the camera parameters (camera matrix, joint rotation-translation matrix).

Once the experiment is loaded, the application enables four tabs. The first one is used to see the photo, while the other three are used to see math equations and expressions and to perform calculations. All formulae in these tabs are dynamically rendered with the  $\text{\LaTeX}$  back end and converted to images, which are then integrated into the interface. The second tab shows the keypoints’ coordinates in image space and world space. The third and fourth tabs are used, respectively, to work with perspective projections and inverse perspective projections.

The 3D model used for the experiment consists of several spheres with different colors. The image to be registered in the environment is shown in Figure 5.4. The image was intentionally cropped and had its background removed for better printing.

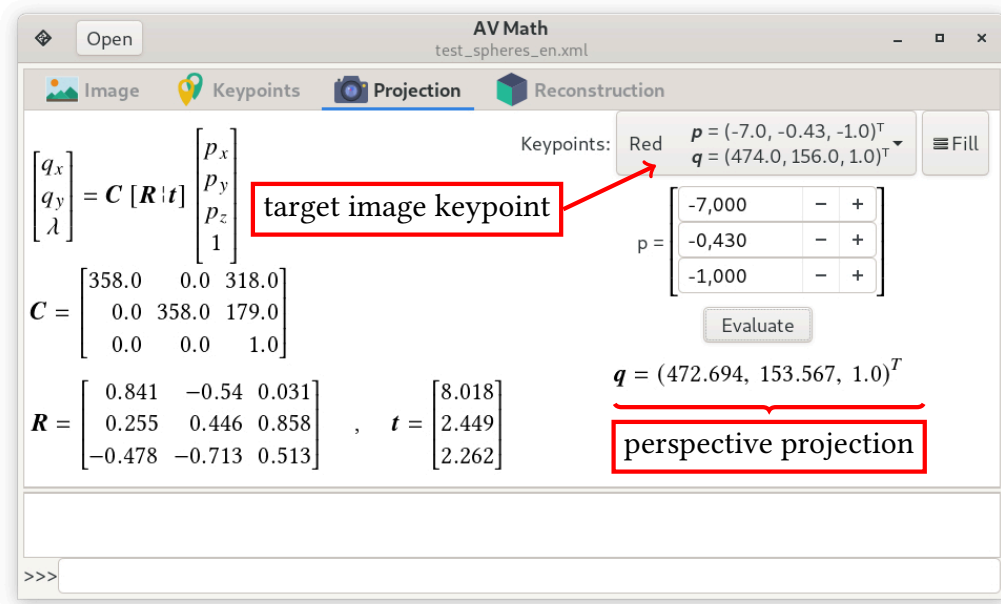
The keypoints are the centers of the 6 small spheres and can be easily detected by image processing, by using circle detection by Hough transforms (ILLINGWORTH; KITTLER 1987),

Figure 5.4: Photo used for the *avmath* experiment.

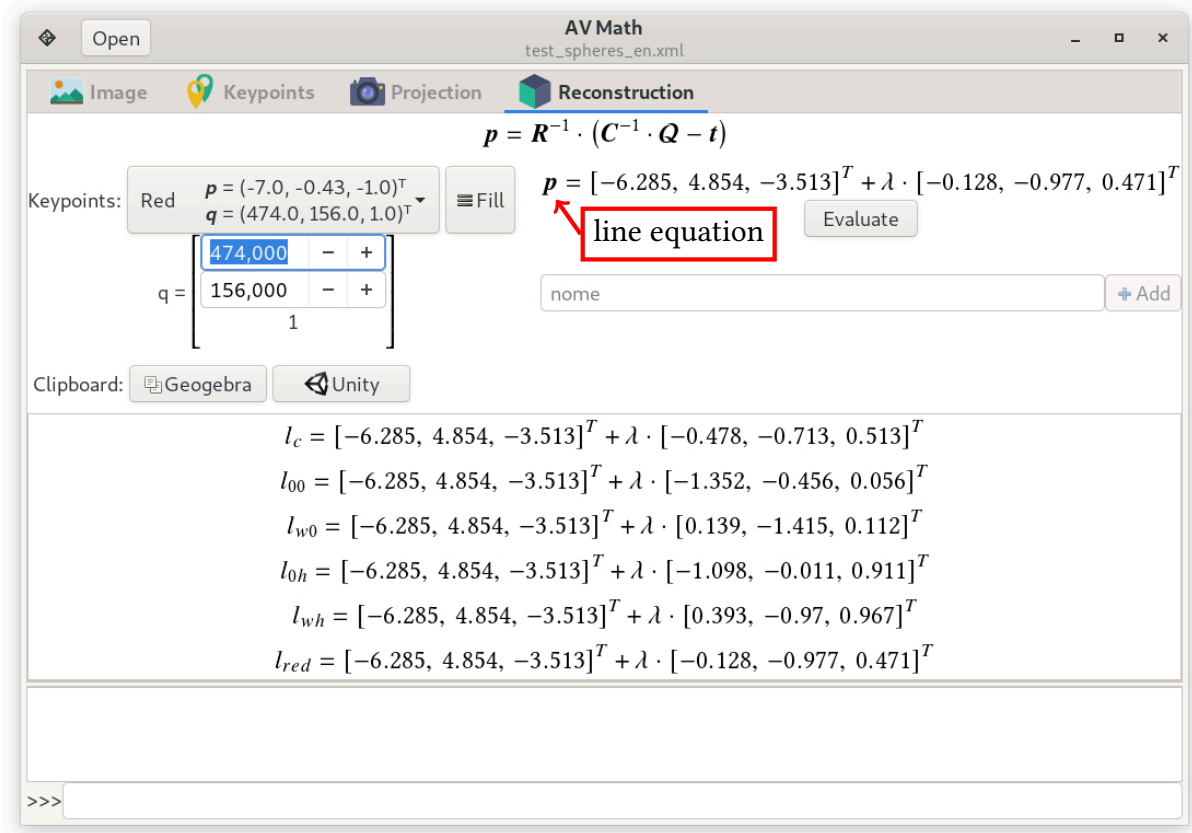
Source: The author.

and then identified by the circle's mean color.

The user can calculate projections for any point in world space, including the keypoints (Figure 5.5). The last tab is used for calculating the line equations for  $\ell_{00}$ ,  $\ell_{w0}$ ,  $\ell_{0h}$ ,  $\ell_{wh}$  (see Section 5.1) and any other line corresponding to some point in image space. In Figure 5.6, the line of sight referring to the red keypoint is highlighted.

Figure 5.5: Perspective projection tab of the *avmath* GUI application.

Source: The author.

Figure 5.6: Inverse perspective projection tab of the *avmath* GUI application.

Source: The author.

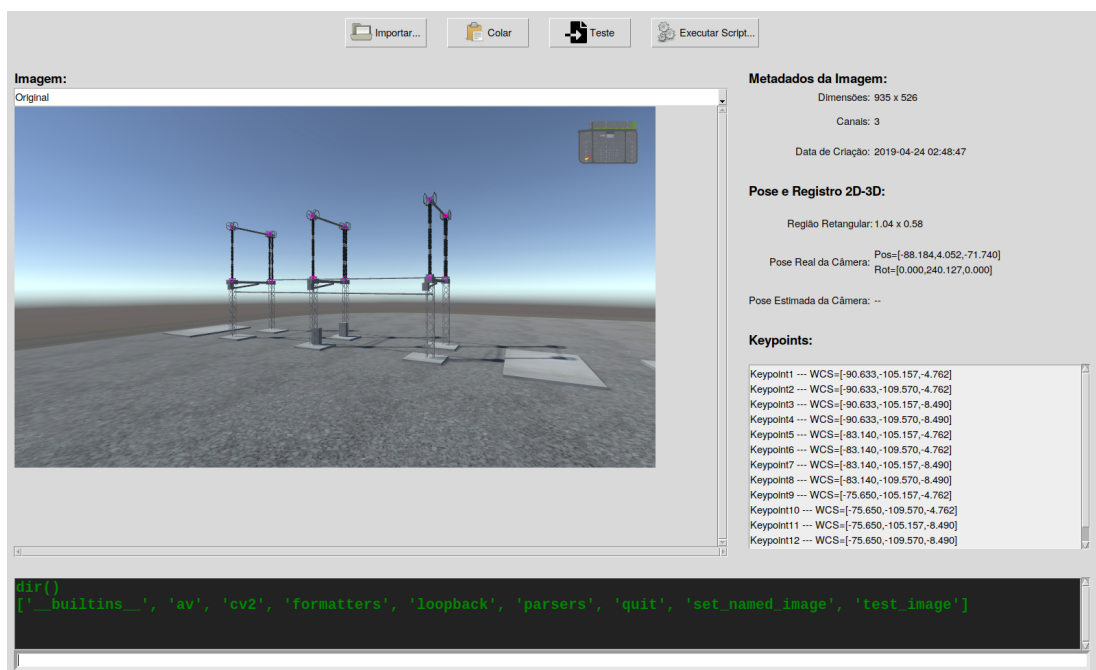
## 5.4 Loopback Experiment (avloopback)

Similar to the *avmath* solution, the *avloopback* GUI application reads an XML file with the image for the experiment, encoded in base 64, and the keypoints coordinates in WCS. The chosen virtual model was a 500kV power disconnector, shown in Figure 5.7 inside the solution's main window.

It is possible to get the coordinates in image space either with mouse clicks or by applying some image processing with the help of an interactive scripting console. For the loopback experiment, those coordinates were manually extracted, considering the location of the equipment insulators.

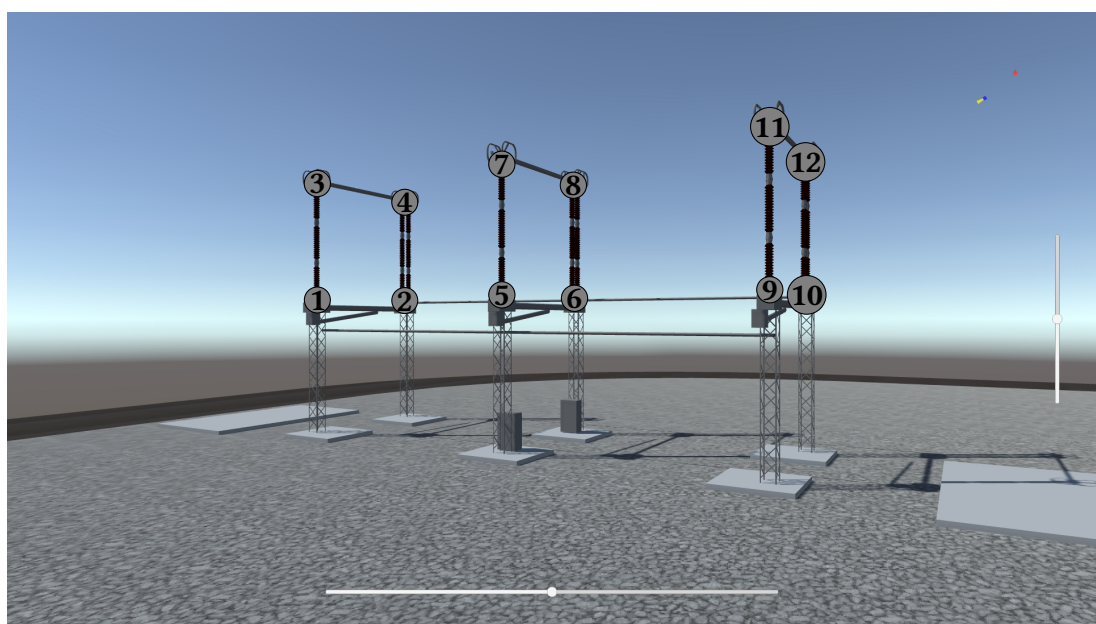
Instead of the simple case of color spheres, this prototype needs a convention for naming the keypoints, used both in the image and in world space. The convention used for this model is illustrated in Figure 5.8.

Figure 5.7: GUI application to extract keypoints.



Source: The author.

Figure 5.8: Nova Ponte’s substation power disconnector keypoints convention.

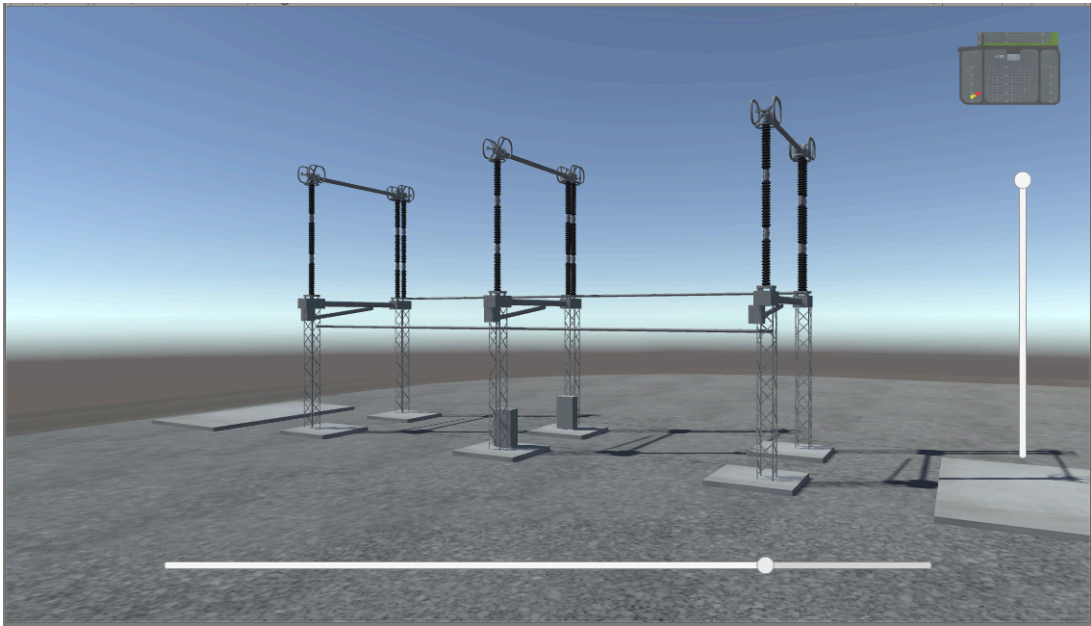


Source: The author.

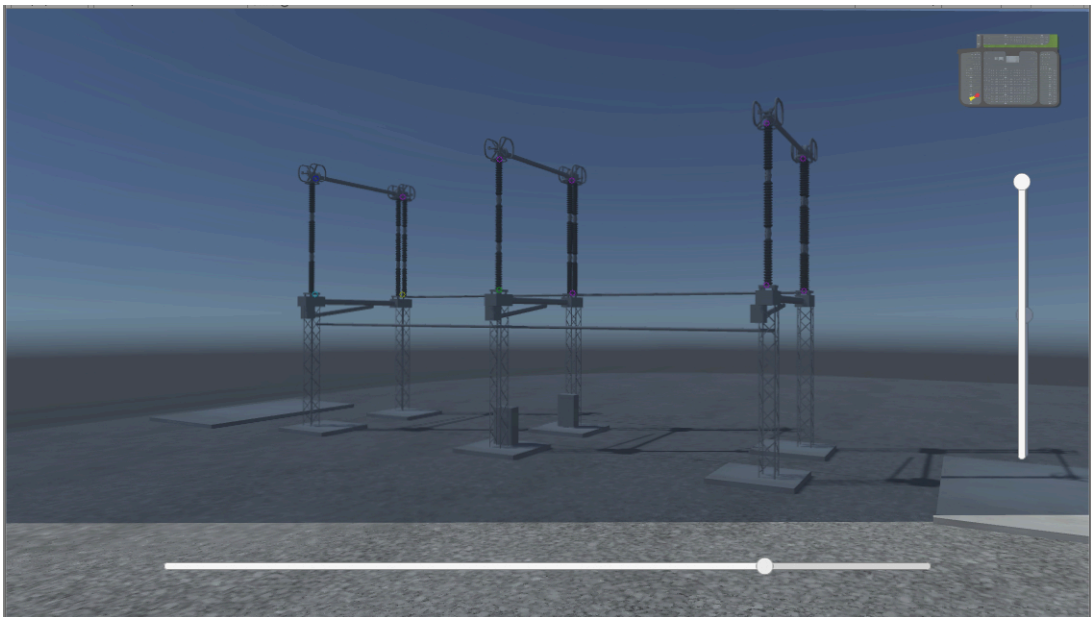
The prototype’s Virtual Environment and the effect of the 2D–3D registration request are given in Figure 5.9.

Figure 5.9: Virtual environment and 2D–3D registration for the *avloopback* prototype.

(a) Virtual camera teleported to the estimated pose.



(b) 2D–3D registration of the loopback image.



Source: The author.

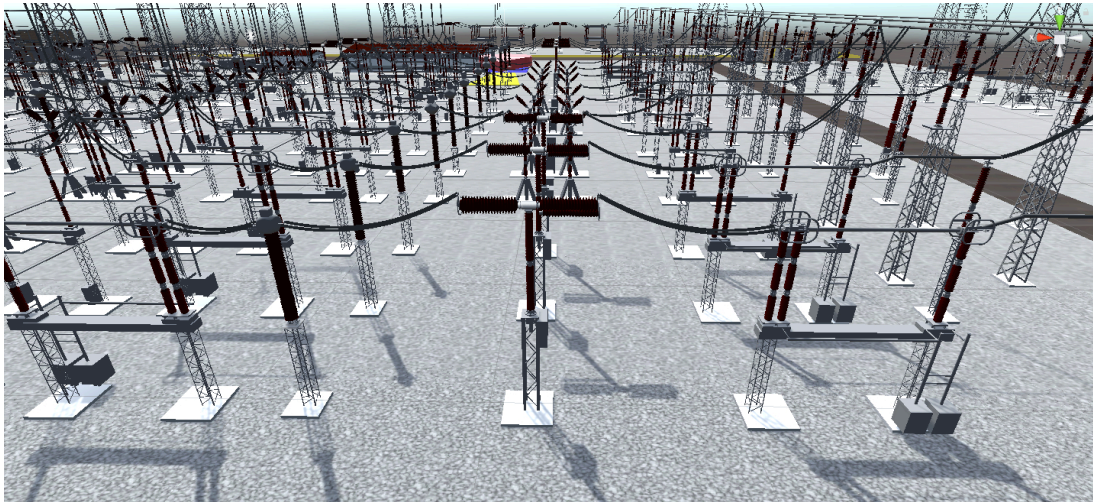
The virtual environment has two levers for adjusting the camera’s focal length and the rectangular region scale. Both adjustments work by applying some small variation to the initial parameter values and improve the result of the overlay.

This is useful because, even if the estimated camera model can fit world keypoints into image pixels, the intrinsic parameters of the target camera (virtual environment’s camera) might differ from the obtained model.

## 5.5 Augmented Virtuality Prototype (avcamera)

The Virtual Environment used for the experiment refers to one of the power substations under the responsibility of CEMIG-GT (Minas Gerais Energy Corporation), namely the Nova Ponte substation (Figure 5.10).

Figure 5.10: Nova Ponte power substation.



Source: The author.

The photo used in the experiment is the thermal photo shown in Figure 5.11. As explained in Chapter 3, thermal photos are particularly useful in substation remote inspection systems. Indeed, Nova Ponte Substation is currently tele-assisted only during the night period, restricting the usage of RGB photos.

Figure 5.11: Field photo used in the *avcamera* prototype.

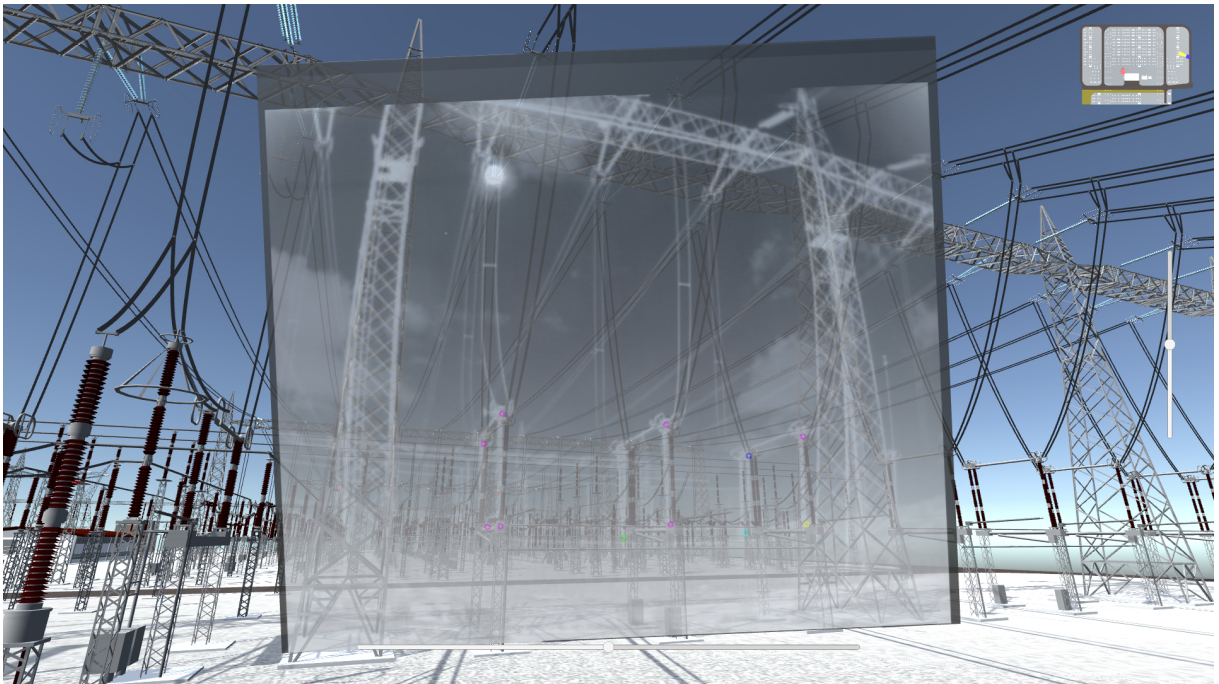


Source: The author.

While the user is navigating in the VR environment, it can emulate an Augmented Reality request for a given asset identifier with a keyboard shortcut. Then, the image is processed and overlaid in the scene using the same registration mechanism as the previous prototypes.

The camera is teleported to the estimated pose, from which we can see the environment augmentation (Figure 5.12). In the figure, some transparency was intentionally added to the rectangular region only for visualization purposes. The overlay visibility can be toggled by the user.

Figure 5.12: 2D–3D registration in a transparent rectangular region for the *avcamera* prototype.



Source: The author.

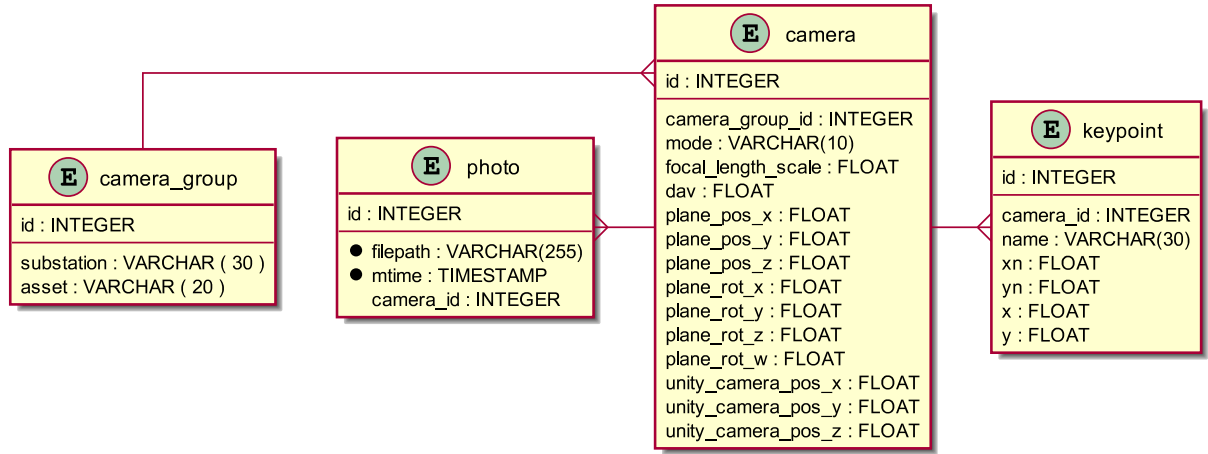
## 5.6 Augmented Virtuality for Remote Inspection (avcemig)

This section describes the last implemented prototype, featuring an architecture for supporting multiple cameras in multiple substations, an alarm system, and the proposed user interface. Besides, some methods to improve the overall registration quality are also outlined.

### 5.6.1 Database model

The augmented virtuality server stores data referring to the photos fetched from the image database and the 2D–3D registration metadata. A simplified data model is presented in Figure 5.13.

Figure 5.13: Back end simplified database model.



Source: The author.

Each substation has zero or more camera groups, which are sets of fixed-positioned cameras aimed at the monitoring of one asset. As explained in the previous section, a single camera with different position presets, multiplexed in time, is treated as if it were multiple fixed cameras. For each camera, a set of keypoints is defined. These are pixel coordinates  $(x, y)$  and normalized coordinates  $(x_n, y_n)$  of the points  $Q_i$  from the mappings  $(P_i \mapsto Q_i)$  explained in Section 2.2.3. These keypoints can be stored in the database, for fixed cameras, or detected in real time by some auxiliary computer vision system, such as the one described by Pereira et al. (2016) or, more recently, the approach proposed by Nassu et al. (2018). However, our tests concern only fixed cameras.

Regarding the images, the photo table is used to provide high-level access to the dataset files. If a real-time communication channel is available for streaming videos from the field, this table is not used.

The poses for both the virtual camera and the target rectangular region are also stored within the camera record. For optimal matching, the overlay plane's orientation will be the same as the virtual camera's (see Figure 5.3), so there is no need to store two distinct orientations. The rotation is stored as a quaternion  $\left(\omega, \begin{bmatrix} x & y & z \end{bmatrix}\right)$ , thus using four floating-point fields in the database. The other six fields are used to store the positions for the camera and the overlay region. Considering the case of fixed cameras, this means that the Perspective-n-Point problem does not need to be executed on each request, but only during the calibration process.

Finally, the camera table also stores the values for both an intrinsic parameter (*focal\_length\_scale* field) and a quality metric (*dav* field). The former can be obtained with a ternary-search algorithm, detailed in the next section, whereas the latter is explained in the next chapter.

### 5.6.2 Focal length scale auto-set method

The RI cameras used in this work have either no mobility or fixed position presets, which is why the keypoints' coordinates are static. A good consequence of this constraint is that there is no need for running the Perspective-n-Point algorithm for every new image acquired. Once the VR environment knows the parameters for well positioning the overlay image and the virtual camera, the real-time operation consists only of fetching the new images and updating the overlay (a texture).

Taking into account the difficulties of having and maintaining intrinsic parameters calibration for each remote camera of the system, it is reasonable to consider methods that do not require procedures *in loco*. Thus, an iterative method for discovering the optimal focal length  $f_x = f_y$  was applied. Once good results are achieved, the parameters' values are stored in a database, described in Section 5.6.1.

The values for  $f_x$  and  $f_y$  are obtained by multiplying the image height,  $h_1$ , by some scale factor  $f_{x,y}$ , mapping from pixels to meters. The algorithm's goal is to find the optimal value for  $f_{x,y}$  inside a numeric range. Our method uses a mean distance metric for the objective function (to be minimized) and a ternary-search variant.

Let us recall the symbols used in Sections 2.2.2 and 2.2.3, considering a set of keypoints coordinates defined in the image,  $\{Q_i \in \mathbb{I} | 1 \leq i \leq n\}$  and in the world  $\{P_i \in \mathbb{W} | 1 \leq i \leq n\}$ , such that  $(P_i \mapsto Q_i)$ . Using the homogeneous coordinates normalization function (2.1), the mean Euclidean distance can be formally stated as:

$$\overline{d_{PNP}} = \frac{1}{n} \sum_{i=1}^n \|\eta(q_i) - \eta(\rho_f(p_i))\|, \quad (5.11)$$

where:

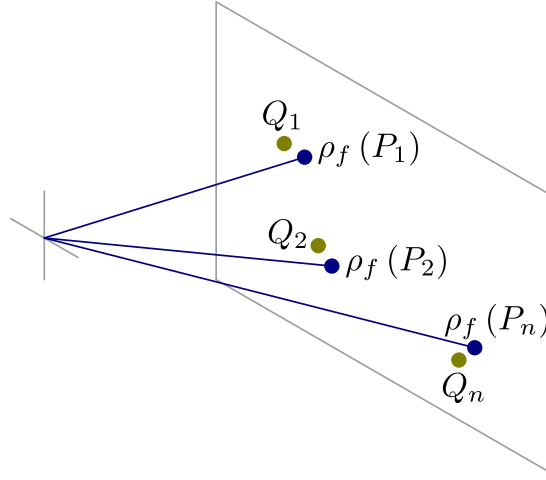
$n$  is the number of keypoints,

$q$  is a vector with key point coordinates in the image coordinate system, and

$\rho_f(p_i)$  is the result of the perspective projection of the point  $P_i$  located at  $p_i$  in the overlay rectangular region, considering the camera pose obtained with  $f_x = f_y = f_{x,y} \cdot h_1$ .

The pixels and their projections (with errors due to inaccurate pose estimation) are represented in Figure 5.14.

Figure 5.14: Image space keypoints and projections from world space keypoints.



Source: The author.

The goal of Algorithm 5.1 is to find the optimal  $f_{x,y} \in [f_{min}, f_{max}]$ , such that  $\overline{d_{PNP}}$  is minimum:

$$f_{x,y} = \arg \min \frac{1}{n} \sum_{i=1}^n \|\eta(q_i) - \eta(\rho_f(p_i))\| \quad (5.12)$$

---

**Algorithm 5.1** Ternary-search focal length scale factor.

---

```

1: procedure FINDOPTIMALF( $f_{min}, f_{max}$ )
2:    $f_1 \leftarrow f_{min}, f_4 \leftarrow f_{max}$ 
3:   while ( $f_4 - f_1 > \epsilon$ ) or iterations limit reached do
4:      $f_2 \leftarrow f_1 + (f_4 - f_1)/3$ 
5:      $f_3 \leftarrow f_1 + 2 * (f_4 - f_1)/3$ 
6:      $error_2 \leftarrow \overline{d_{PNP}}$  metric using values  $R_2$  and  $T_2$ 
7:      $error_3 \leftarrow \overline{d_{PNP}}$  metric using values  $R_3$  and  $T_3$ 
8:     if  $error_2 < error_3$  then
9:        $f_4 \leftarrow f_3$ 
10:    else
11:       $f_1 \leftarrow f_2$ 
12:    end if
13:  end while
14:  if  $error_2 < error_3$  then
15:    return  $f_2$ 
16:  end if
17:  return  $f_3$ 
18: end procedure

```

---

In each iteration, the search range is subdivided into four uniformly spaced values  $f_1, \dots, f_4$ , and then the metric is evaluated in the intermediate points  $f_2$  and  $f_3$ , giving conditions to narrow the search range to either  $[f_1, f_3]$  or  $[f_2, f_4]$ . The algorithm runs with a fixed number

of iterations, as shown above, or until a considerably small error is found.

### 5.6.3 SCADA integration

The SCADA database stores both analog values, such as the voltage in a transformer and digital (i.e., “open” or “closed”) device states. The latter is used in our solution, specifically for the case of disconnecter switches.

In this sense, the VR environment application makes HTTP requests to a middleware, periodically fetching a report with the digital states of all disconnecter switches. For our RI goals, this telemetry data is combined with the on-line images, providing the automatic detection of discrepancies among these two sources. This is especially useful in cases where the switch is only partially opened, a condition not detected by standard telemetry instrumentation devices. Experiments with the system have been successfully done, by using a web-based equipment state controller (Figure 5.15) to safely simulate the role of the SCADA database used in production.

Figure 5.15: Equipment state simulator.

## Estados de Equipamentos

☒ Listas Pré-Definidas: NPON

☐ Arquivo Externo: Browse... No file selected.

Carregar... Salvar...

Nome	Valor	Ações
UHNPNON_S4@A	0	<input type="text" value="0.000"/> <span>Alterar</span>
UHNPNON_S4@Hz	60	<input type="text" value="60.000"/> <span>Alterar</span>
UHNPNON_S4@MVAR	0	<input type="text" value="0.000"/> <span>Alterar</span>
UHNPNON_S4@MW	0	<input type="text" value="0.000"/> <span>Alterar</span>
UHNPNON_S4@kV	0	<input type="text" value="0.000"/> <span>Alterar</span>
UHNPNON_S6@A	0	<input type="text" value="0.000"/> <span>Alterar</span>
UHNPNON_S6@Hz	60	<input type="text" value="60.000"/> <span>Alterar</span>
UHNPNON_S6@MVAR	0	<input type="text" value="0.000"/> <span>Alterar</span>
UHNPNON_S6@MW	0	<input type="text" value="0.000"/> <span>Alterar</span>
UHNPNON_S6@kV	0	<input type="text" value="0.000"/> <span>Alterar</span>
UHNPNON_10U3	Fechado	<input checked="" type="radio"/> Fechado <input type="radio"/> Aberto <span>Alterar</span>
UHNPNON_10U4	Fechado	<input checked="" type="radio"/> Fechado <input type="radio"/> Aberto <span>Alterar</span>
UHNPNON_10U5	Fechado	<input checked="" type="radio"/> Fechado <input type="radio"/> Aberto <span>Alterar</span>
UHNPNON_10U8	Fechado	<input checked="" type="radio"/> Fechado <input type="radio"/> Aberto <span>Alterar</span>

Source: The author.

No action is taken if no error is detected. Otherwise, the system displays an alarm and gives the option to perform the 2D–3D registration. The alarm system dialogs will be discussed in Section 5.6.4.

An alternative design is to store the state inferred from the image in the same database used to store the standard telemetry data. In such an approach, the state inference machine sends the results directly to the SCADA database, using nodes with special identifiers. This is the solution

used by the CEMIG-GT company. The SCADA nodes referring to the computer vision system bring the suffix VM (from video-monitoring). Both types of nodes, standard and image-based, were added into the equipment state simulator.

#### 5.6.4 User interface

This section describes the user interface for the remote inspection system. The environment is semi-immersive, receiving input only from standard devices (keyboard, mouse, and optionally a joystick). Since the system can be used for long periods, this characteristic may prevent or reduce cybersickness, which is commonly caused by the use of special devices, such as head-mounted displays (HMDs) (VENKATAKRISHNAN; VENKATAKRISHNAN; BHARGAVA, et al., 2020; VENKATAKRISHNAN; VENKATAKRISHNAN; ANARAKY, et al., 2020).

##### Interacting with the monitored assets

When the remote inspection is active, the assets related to the available camera groups are decorated with a transparent sphere with an alternate vertical motion (Figure 5.16). The movement is dynamically switched to a rotation through the vertical axis when the mouse is hovering the sphere. Except for some animations related to the opening or closing of power disconnectors, the equipment is static, so representative objects with motion are almost reserved to indicate interactive objects.

Figure 5.16: Assets decorated with interactive 3D objects.


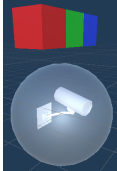



*Source:* The author.

The symbol inside the sphere varies according to the current configuration for the associated asset and the virtual camera's pose. The possible states are explained in Table 5.2.

One of the most notable features of the system is the ability to decide whether the spatial registration is adequate for the current point of view, according to some quantitative metric

Table 5.2: Representative icons for spatial registration.

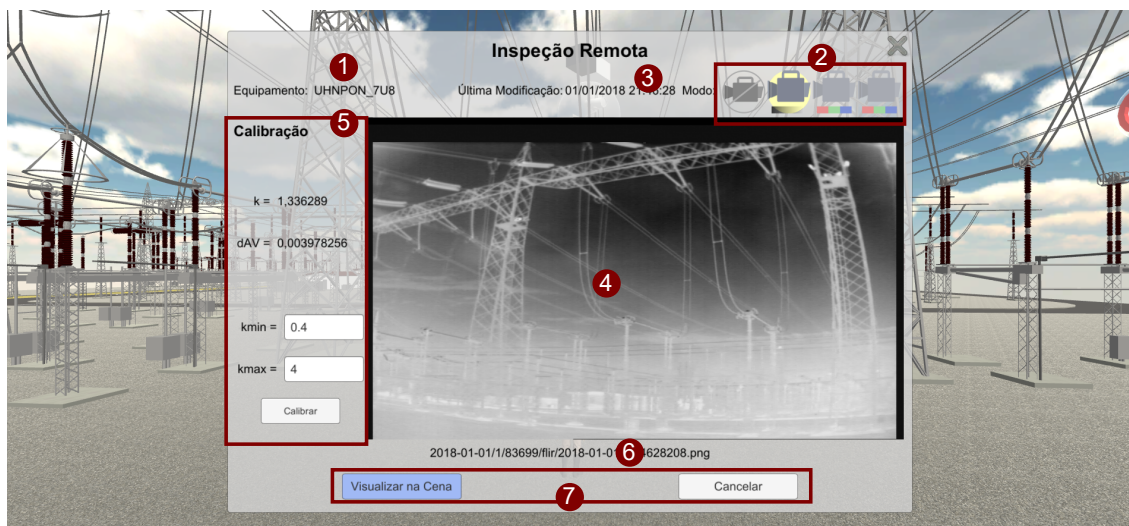
Icon	Description
	Spatial registration disabled.
	Spatial registration enabled for a thermal camera. In this scenario, the icon is shown instead of the field image since the virtual camera's current position is not suitable for the image overlay (registration error would exceed a user-defined threshold).
	Spatial registration enabled for a color camera. In this scenario, the icon is shown instead of the field image since the virtual camera's current position is not suitable for the image overlay (registration error would exceed a user-defined threshold).

and a configurable threshold. In this sense, the representative icons are displayed whenever the spatial registration is disabled for that asset, which can occur either because of configuration or inappropriate point of view.

## 2D–3D spatial registration configuration dialog

When either the overlaid field image or the icon is clicked, a configuration dialog is shown (Figure 5.17).

Figure 5.17: 2D–3D spatial registration configuration dialog.



Source: The author.

The following information is presented:

1. the asset name;
2. buttons for choosing the current field camera (or none);
3. the timestamp referring to the latest image available for the currently selected camera;
4. the last field camera;
5. a calibration panel for defining some spatial registration parameters;
6. the file relative path in the dataset (when real-time images are unavailable);
7. action buttons for performing the 2D–3D registration or closing the dialog.

### Alarm system

As discussed in Section 5.6.3, alarms are generated whenever there is any inconsistency between the reported digital states from two sources: standard telemetry devices and an image-based inference machine. Alarms are promptly presented in a dialog window (Figure 5.18).

Figure 5.18: Alarm dialog.



Source: The author.

The main elements of the dialog are:

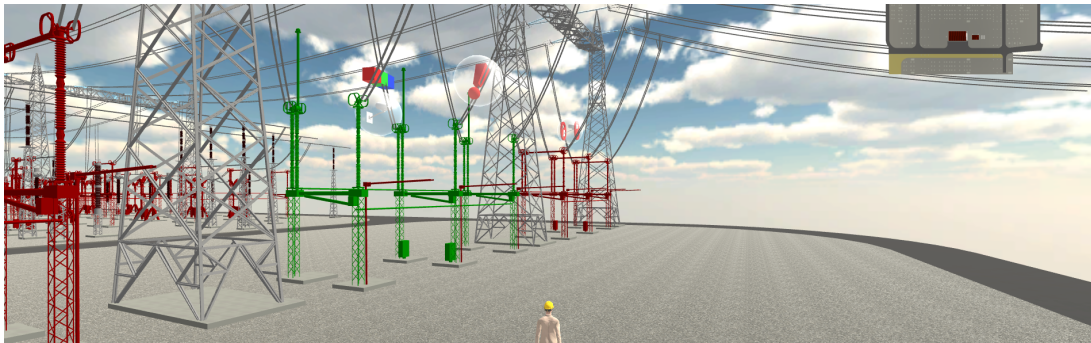
1. the name of the asset related to the abnormal condition;
2. the digital state understood by the image-based inference machine, which is stored as a special node in the SCADA historical database;
3. the digital state measured by standard telemetry devices;

4. the timestamp at which the condition was first detected<sup>1</sup>;
5. the last timestamp at which the condition was detected (this field is updated dynamically);
6. the last image acquired during the alarm;
7. action buttons.

In addition to the somewhat standard “Acknowledge” and “Acknowledge and Dismiss” buttons, there is a button used to trigger the 2D–3D spatial registration and teleport the virtual camera to the appropriate point of view. This is a relevant feature for remote inspection usages: the ability to immediately evaluate the situation with the support of a bigger context, specifically, a virtual model of the region nearby the problem’s location.

Acknowledged alarms are enqueued for further actions, through the same dialog. Again, an interactive object is rendered to represent this situation (Figure 5.19).

Figure 5.19: Alarm queue icon.



Source: The author.

Finally, it should be noted that the alarm’s life cycle ends either when the condition ceases to exist or when the user chooses to dismiss. Therefore, if the same discrepancy responsible for triggering an alarm in the past occurs again, a new (and different in terms of identity) alarm will be generated.

### 5.6.5 API specification

All communication between the registration server and the VR/AV application is done through HTTP requests. The main endpoints and their purposes are summarized below. More complete specifications are provided in Appendix A.

**GET /cameras/query/<substation>** Retrieve the list of cameras for a given substation. This is the first request made by the clients, so they can be aware of the available remote inspection structure for the virtual environment being displayed.

<sup>1</sup>the start time was fixed at 10 October 2017 during our tests to make the system compatible with the dataset images, instead of real-time operation.

**GET /cameras/<id>/last\_image** Query the last available image for a certain camera.

**POST /cameras/<id>/calibrate** Perform server-side camera calibration by using the algorithm described in Section 5.6.2. The keypoints in WCS are specified by the client since it is that software component that is aware of the 3D models.

**POST /cameras/<id>/set-params** Save camera parameters obtained by the calibration request.

**GET /cameras/<id>/video-feed.mjpg** Get the real-time video from some camera.

**GET /cameras/<int:camera\_id>/image** Get the first historical image in a time interval. This request is particularly useful for the alarm system, which can present the first image related to an abnormal condition, instead of being limited only to the current (last available) image.

## 5.7 Conclusions

This chapter described the mathematical model of the 2D–3D registration, as well as the implementation and usage details of the developed prototypes. Each application served as a foundation for the next, gradually reaching the final solution’s features.

The next chapter presents some quality metrics related to the image registration and system performance.

## Results

This chapter discusses the 2D-3D registration results, considering both the quantitative metrics and qualitative analysis.

### 6.1 Metrics for Quality Evaluation

To validate the feasibility of the proposed solution, different metrics might be put in place. The metrics concern the registration quality and the performance impact on real-time rendering and navigation.

#### 6.1.1 2D–3D Registration Metrics

##### Mean Euclidean distance and relative reprojection errors

For the Perspective-n-Point routine, we can evaluate the mean Euclidean distance (5.11), described in Section 5.6.2, considering the keypoints in the image space and the perspective transformation of world keypoints.

However, this metric measures the mean distance in pixels, which might not be an intuitive unit for representing errors. A more generic alternative, then, is to obtain the relative errors for each axis, resulting in values that do not depend on the image dimensions. The mean relative reprojection errors are given by:

$$\overline{e_{x\%}} = \frac{1}{n \cdot w_1} \sum_{i=1}^n \left| \eta(\mathbf{q}_i)_x - \eta(\rho_f(\mathbf{p}_i))_x \right| \cdot 100\% \quad (6.1)$$

$$\overline{e_{y\%}} = \frac{1}{n \cdot h_1} \sum_{i=1}^n \left| \eta(\mathbf{q}_i)_y - \rho_f(\mathbf{p}_i)_y \right| \cdot 100\% \quad (6.2)$$

where

$n$  is the number of keypoints,

$w_1$  is the image width,

$h_1$  is the image height,

$\eta$  is the normalization function for homogeneous coordinates,

$q_{i,x}, q_{i,y}$  are the coordinates of the pixel related to the  $i$ -th keypoint and

$\rho_f(\mathbf{p}_i)_x, \rho_f(\mathbf{p}_i)_y$  are the first two coordinates of the perspective projection of  $\mathbf{p}_i$ , using the estimated camera pose.

### Distance between virtual and estimated cameras

For the loopback solution, since the virtual environment camera's pose is known, it can be compared against the estimated pose. For this matter, two additional metrics are established: the Euclidean distance between the virtual and estimated cameras' poses and the rotation variation among these cameras. The former can be expressed by:

$$d_{CAMS} = \|\mathbf{u}_{VR} - \mathbf{u}_{PNP}\|, \quad (6.3)$$

where

$\mathbf{u}_{VR}$  is the position of the virtual environment's camera in the world coordinate system and

$\mathbf{u}_{PNP}$  is the estimated position obtained from the PNP solution, i.e.,  $\mathbf{u}_{PNP} = -\mathbf{b}$ , which is the intersection for all reconstruction lines given by (5.4).

The rotation variation should also be taken into account. There are many alternatives for metrics aimed at comparing rotations, such as:

- the norm of the difference of quaternions;
- the inner product of unit quaternions;
- the deviation from the identity matrix.

Still, some studies on the subject suggest that unit quaternions are both spatially and computationally more efficient (HUYNH, 2009). This metric is formally stated as:

$$\Delta R = \phi(\hat{\mathbf{r}}_{VR}, \hat{\mathbf{r}}_{PNP}) = 1 - \|\hat{\mathbf{r}}_{VR} \cdot \hat{\mathbf{r}}_{PNP}\| \quad (6.4)$$

where

$\hat{\mathbf{r}}_{VR}$  is the unit quaternion referring to the virtual camera rotation and

$\hat{\mathbf{r}}_{PNP}$  is the unit quaternion referring to the estimated camera rotation.

To obtain a quaternion from a given rotation matrix  $\mathbf{R}$ , we use the method proposed by Sarabandi and Thomas (2019).

### Distance between keypoints in the rendered AV image

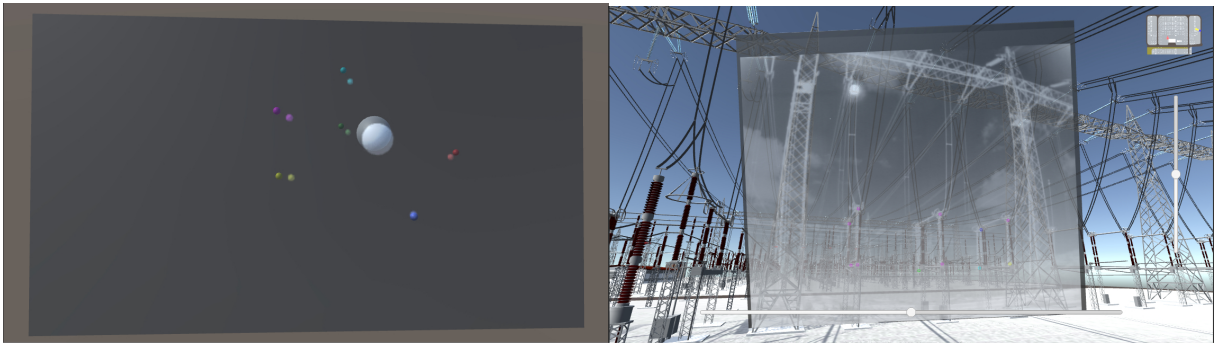
Another metric, especially useful for the cases in which the real camera's pose is not known, consists of analyzing the keypoints positions once the virtual camera is teleported to the estimated pose and the rectangular region is textured with the field image.

The Unity 3D Scripting API exposes a method to map a point in world space to the corresponding pixel related to the current camera viewport. This can be used to extract the virtual model keypoints coordinates in the rendered image.

As for the rectangular region with the image overlay, there is also a simple process to extract the keypoints coordinates in the rendered frame. Invisible objects can be added as children of the rectangular region and positioned to match the photo keypoints. Once the plane is positioned and oriented after the 2D–3D registration, the same world-to-screen utility method can be applied to these objects to extract their positions in the rendered image.

Figure 6.1 shows rendered AV images for the avmath and avcamera prototypes, with the addition of some transparency to the inserted images for better visualization.

Figure 6.1: Transparent rectangular region for better inspecting image and world keypoints.



Source: The author.

Again, the mean Euclidean distance can be used, considering such images rendered by the AV application. After 2D–3D registration succeeds, the distance between corresponding keypoints are computed and used in the metric expression:

$$\overline{d_{AV}} = \frac{1}{n} \sum_{i=1}^n \left\| \eta \left( \mathbf{q}_{i,AV} \right) - \eta \left( \mathbf{q}_{i,M} \right) \right\|, \quad (6.5)$$

where

$n$  is the number of keypoints,

$\eta$  is the normalization function for homogeneous coordinates,

$\mathbf{q}_{i,AV}$  is the  $i$ -th key point pixel coordinates in the rectangular region, and

$\mathbf{q}_{i,M}$  is the  $i$ -th key point pixel coordinates in the virtual model.

Relative errors for this metric are given below:

$$\overline{\delta_x\%} = \frac{1}{n \cdot w_1} \sum_{i=1}^n \left| \eta \left( \mathbf{q}_{i,AV} \right)_x - \eta \left( \mathbf{q}_{i,M} \right)_x \right| \cdot 100\% \quad (6.6)$$

$$\overline{\delta_y\%} = \frac{1}{n \cdot h_1} \sum_{i=1}^n \left| \eta \left( \mathbf{q}_{i,AV} \right)_y - \eta \left( \mathbf{q}_{i,M} \right)_y \right| \cdot 100\% \quad (6.7)$$

These metrics are evaluated automatically by the VR application, in real-time.

### 6.1.2 Performance Impact

For measuring performance impact, one metric is related to the time interval between the instant just before the request, from the virtual environment, and the moment after which the 2D–3D registration is complete. This metric named henceforth  $\Delta t_{2D,3D}$  is especially useful when multiple VR environments and inspection cameras are considered.

The real-time rendering should also be analyzed. For that matter, the engine's reported *Frames Per Second* (FPS) metric should be sampled in a log file for further analysis. We're interested in the FPS variation during the 2D–3D registration requests.

Finally, the focal length scale factor optimization algorithm is evaluated. For our main test case, covering only fixed cameras, this leads to measuring the time needed for the calibration HTTP request, aimed at computing the optimal poses for the overlay plane and the virtual camera. This metric is named  $\Delta t_{calib}$ .

### 6.1.3 Prototype-Metrics Table

The previous sections described the quality metrics regarding registration and performance. Table 6.1 clarify the use of these metrics in each prototype, considering their specific goals.

The next sections present the details and results for each prototype. The metrics  $\overline{d_{PNP}}$ ,  $\overline{d_{AV}}$  and  $\Delta t_{2D,3D}$  can be evaluated in real-time directly by the augmented virtuality server and the game engine. This can be used to assess the registration quality and disable or notify the user

whenever a bad 2D–3D registration result is detected. The feature is used in the final prototype, *avcemig*.

Table 6.1: Prototype-metrics table.

	<b>avmath</b>	<b>avloopback</b>	<b>avcamera</b>	<b>avcemig</b>
<b>2D–3D Registration</b>				
$\overline{d_{PNP}}, \overline{e_{x\%}}, \overline{e_{y\%}}$	✓	✓	✓	✓
$\overline{d_{CAMS}}, \overline{\Delta R}$		✓		
$\overline{d_{AV}}, \overline{\delta_{x\%}}, \overline{\delta_{y\%}}$		✓	✓	✓
<b>Performance</b>				
$\Delta t_{2D,3D}$				✓
FPS				✓
$\Delta t_{calib}$				✓

## 6.2 Perspective-n-Point Prototype (avmath)

As explained in Chapter 4, the first prototype uses an abstract virtual model consisting of several color spheres. The experiment parameters values are presented in Table 6.2, while the keypoints coordinates are given in Table 6.3.

Table 6.2: *avmath* experiment parameters.

<b>Description</b>	<b>Symbol</b>	<b>Value</b>
Image width (pixels)	$w_1$	636
Image height (pixels)	$h_1$	358
Rectangular Region width (VR units)	$w_2$	6.36
Rectangular Region height (VR units)	$h_2$	3.58
VR Camera Position	$\mathbf{u}_{VR}$	$[-5.777126 \quad 6.372389 \quad -4.378578]^T$

The camera matrix is constructed so that the principal point is located in the image center and  $f_x = f_y = h_1$ :

$$C = \begin{bmatrix} 358 & 0 & 318 \\ 0 & 358 & 179 \\ 0 & 0 & 1 \end{bmatrix}. \quad (6.8)$$

Table 6.3: *avmath* prototype keypoints.

Identifier	WCS	ICS
Red	$\mathbf{p}_1 = [-7.0 \quad -0.43 \quad -1.0]^T$	$\mathbf{q}_1 = [474 \quad 156 \quad 1]^T$
Green	$\mathbf{p}_2 = [-9.0 \quad -0.43 \quad -1.0]^T$	$\mathbf{q}_2 = [353 \quad 128 \quad 1]^T$
Blue	$\mathbf{p}_3 = [-7.0 \quad 1.57 \quad -1.0]^T$	$\mathbf{q}_3 = [430 \quad 223 \quad 1]^T$
Yellow	$\mathbf{p}_4 = [-9.0 \quad 1.57 \quad -1.0]^T$	$\mathbf{q}_4 = [288 \quad 181 \quad 1]^T$
Cyan	$\mathbf{p}_5 = [-9.0 \quad -0.43 \quad -2.0]^T$	$\mathbf{q}_5 = [356 \quad 70 \quad 1]^T$

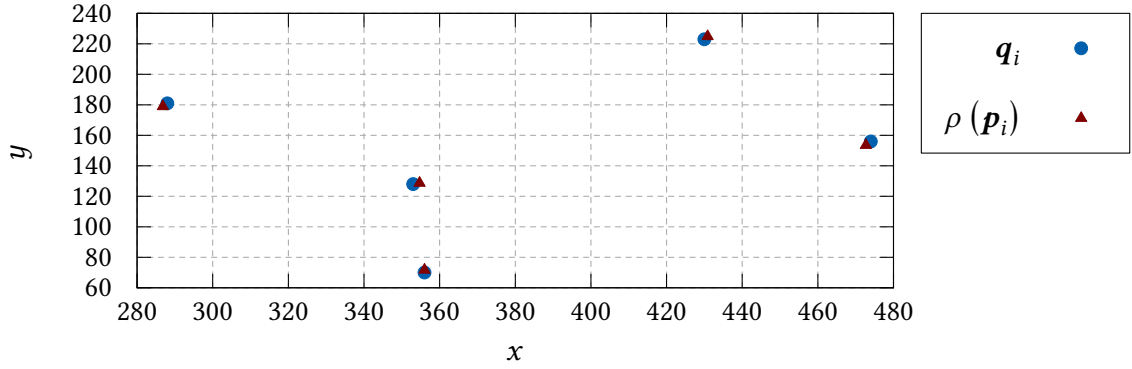
Then, the estimated joint rotation-translation, as given by the PNP solver, is:

$$\mathbf{R} = \begin{bmatrix} 0.8408 & -0.5404 & 0.0314 \\ 0.2547 & 0.4461 & 0.8579 \\ -0.4777 & -0.7133 & 0.5128 \end{bmatrix}, \mathbf{t} = \begin{bmatrix} 8.018 \\ 2.449 \\ 2.262 \end{bmatrix} \quad (6.9)$$

With all these symbols bounded to numeric values, the metrics  $\overline{d_{PNP}}$ ,  $\overline{e_{x\%}}$  and  $\overline{e_{y\%}}$  can be calculated (Table 6.4). The image keypoints and their associated projections are plotted in Figure 6.2.

Table 6.4: *avmath* PNP errors.

Identifier	Image	Projection	Distance	Relative Error	
			$d_{PNP}$	$ e_{x\%} $	$ e_{y\%} $
Red	$[474 \quad 156]^T$	$[472.694 \quad 153.567]^T$	2.762	0.205 %	0.680 %
Green	$[353 \quad 128]^T$	$[354.707 \quad 128.693]^T$	1.842	0.268 %	0.194 %
Blue	$[430 \quad 223]^T$	$[430.847 \quad 224.847]^T$	2.032	0.133 %	0.516 %
Yellow	$[288 \quad 181]^T$	$[286.817 \quad 178.954]^T$	2.363	0.186 %	0.571 %
Cyan	$[356 \quad 70]^T$	$[356.002 \quad 71.704]^T$	1.704	0.000 %	0.476 %
Mean Values			2.141	0.159%	0.487%

Figure 6.2: Keypoints and projections for the *avmath* prototype.

Source: The author.

Finally, it is easy to obtain the camera estimated position,  $-\mathbf{b}$ :

$$\mathbf{u}_{PNP} = -\mathbf{b} = -\mathbf{R}^{-1} \cdot \mathbf{t} = \begin{bmatrix} -6.2849 \\ 4.8546 \\ -3.5130 \end{bmatrix} \quad (6.10)$$

The errors are mainly due to the approximation for  $f_x = f_y = h_1$ . However, this prototype served its purpose of asserting that the *solvePNP* routine is properly running for the approximated camera matrix  $\mathbf{C}$ .

### 6.3 Loopback Prototype (avloopback)

The loopback prototype was developed considering a virtual environment with a single instance of a power disconnect switch 3D model.

Table 6.5 presents the experiment parameters. One difference from the previous prototype is that the image size depends on the VR environment window size since a screenshot will serve as input. Nevertheless, the Unity 3D project was configured to work with a fixed aspect ratio of 16:9.

Table 6.5: *avloopback* experiment parameters.

Description	Symbol	Value
Image width (pixels)	$w_1$	1920 (dynamic)
Image height (pixels)	$h_1$	1080 (dynamic)
Rectangular region width (VR units)	$w_2$	10.39
Rectangular region height (VR units)	$h_2$	5.84
VR camera position	$\mathbf{u}_{VR}$	$[-71.7399 \quad -88.1842 \quad -4.0516]^T$

In the case of manually specifying the keypoints, they may be normalized to the range  $[0, 1]$ ,

for example, to become independent from the image size. Absolute values are obtained by multiplying these coordinates with the image dimensions, whenever appropriate. This is also applicable to field cameras which have been reconfigured for a different resolution. Hence, the normalized coordinates were added to the database model, as shown in Section 5.6.1. Table 6.6 shows the keypoints coordinates' absolute values, considering the image with the dimensions  $1920 \times 1080$ .

Table 6.6: *avloopback* prototype keypoints.

Identifier	WCS	ICS
Keypoint 1	$\mathbf{p}_1 = [-90.633 \quad -105.157 \quad -4.762]^T$	$\mathbf{q}_1 = [547 \quad 507 \quad 1]^T$
Keypoint 2	$\mathbf{p}_2 = [-90.633 \quad -109.57 \quad -4.762]^T$	$\mathbf{q}_2 = [701 \quad 508 \quad 1]^T$
Keypoint 3	$\mathbf{p}_3 = [-90.633 \quad -105.157 \quad -8.49]^T$	$\mathbf{q}_3 = [547 \quad 302 \quad 1]^T$
Keypoint 4	$\mathbf{p}_4 = [-90.633 \quad -109.57 \quad -8.49]^T$	$\mathbf{q}_4 = [701 \quad 335 \quad 1]^T$
Keypoint 5	$\mathbf{p}_5 = [-83.14 \quad -105.157 \quad -4.762]^T$	$\mathbf{q}_5 = [872 \quad 498 \quad 1]^T$
Keypoint 6	$\mathbf{p}_6 = [-83.14 \quad -109.57 \quad -4.762]^T$	$\mathbf{q}_6 = [1000 \quad 504 \quad 1]^T$
Keypoint 7	$\mathbf{p}_7 = [-83.14 \quad -105.157 \quad -8.49]^T$	$\mathbf{q}_7 = [872 \quad 268 \quad 1]^T$
Keypoint 8	$\mathbf{p}_8 = [-83.14 \quad -109.57 \quad -8.49]^T$	$\mathbf{q}_8 = [998 \quad 305 \quad 1]^T$
Keypoint 9	$\mathbf{p}_9 = [-75.65 \quad -105.157 \quad -4.762]^T$	$\mathbf{q}_9 = [1344 \quad 489 \quad 1]^T$
Keypoint 10	$\mathbf{p}_{10} = [-75.65 \quad -109.57 \quad -4.762]^T$	$\mathbf{q}_{10} = [1409 \quad 498 \quad 1]^T$
Keypoint 11	$\mathbf{p}_{11} = [-75.65 \quad -105.157 \quad -8.49]^T$	$\mathbf{q}_{11} = [1344 \quad 202 \quad 1]^T$
Keypoint 12	$\mathbf{p}_{12} = [-75.65 \quad -109.57 \quad -8.49]^T$	$\mathbf{q}_{12} = [1407 \quad 264 \quad 1]^T$

To better analyze the influence of the camera matrix approximation, two sliders have been added to the virtual environment, allowing to multiply  $f_x$  and  $f_y$  with the same factor  $k_f \in [0.7, 1.3]$  and to apply a scale  $k_{rect} \in [0, 2]$  to the rectangular region. The next sections present the results considering the scenarios with and without this manual adjustment step. It should be noted that, for fixed cameras, this adjustment does not need to be done for every request: the optimal values can be stored as a *per-camera* configuration dictionary. Again, the database contains fields for storing such individual parameters.

### 6.3.1 Scenario 1: No adjustments

In this scenario, the camera matrix is constructed so that the principal point is located in the image center and  $f_x = f_y = h_1$ :

$$\mathbf{C} = \begin{bmatrix} 1080 & 0 & 960 \\ 0 & 1080 & 540 \\ 0 & 0 & 1 \end{bmatrix}.$$

The pose obtained from the PNP estimation is:

$$R = \begin{bmatrix} 0.867788 & -0.496924 & 0.003259 \\ 0.011587 & 0.026791 & 0.999574 \\ -0.496799 & -0.867380 & 0.029007 \end{bmatrix}, t = \begin{bmatrix} 18.38573 \\ 7.86446 \\ -114.93329 \end{bmatrix}$$

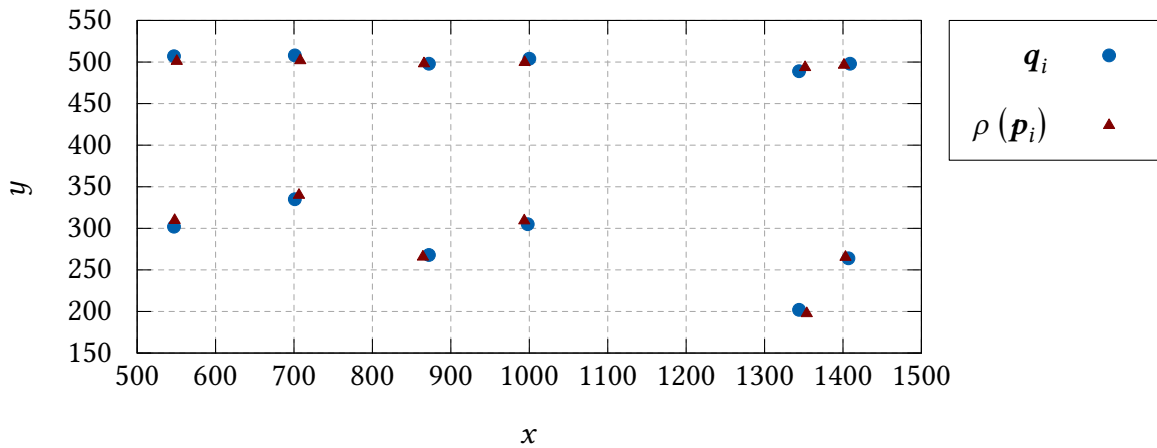
### Mean Euclidean Distance and Relative Errors

Table 6.7 shows the results for  $\overline{d_{PNP}}$ ,  $\overline{e_x\%}$  and  $\overline{e_y\%}$ . The image keypoints and their associated projections are also plotted in Figure 6.3.

Table 6.7: *avloopback* scenario 1 – PNP errors.

Identifier	Image	Projection	Distance			Relative Error	
			$d_{PNP}$	$ e_x\% $	$ e_y\% $		
Keypoint1	$[547 \ 507]^T$	$[550.525 \ 501.067]^T$	6.901	0.184 %	0.549 %		
Keypoint2	$[701 \ 508]^T$	$[707.993 \ 501.921]^T$	9.266	0.364 %	0.563 %		
Keypoint3	$[547 \ 302]^T$	$[547.799 \ 309.751]^T$	7.792	0.042 %	0.718 %		
Keypoint4	$[701 \ 335]^T$	$[706.371 \ 340.035]^T$	7.362	0.280 %	0.466 %		
Keypoint5	$[872 \ 498]^T$	$[865.726 \ 498.134]^T$	6.275	0.327 %	0.012 %		
Keypoint6	$[1000 \ 504]^T$	$[994.031 \ 499.665]^T$	7.377	0.311 %	0.401 %		
Keypoint7	$[872 \ 268]^T$	$[864.381 \ 265.717]^T$	7.953	0.397 %	0.211 %		
Keypoint8	$[998 \ 305]^T$	$[993.585 \ 309.293]^T$	6.158	0.230 %	0.397 %		
Keypoint9	$[1344 \ 489]^T$	$[1351.708 \ 493.612]^T$	8.982	0.401 %	0.427 %		
Keypoint10	$[1409 \ 498]^T$	$[1401.222 \ 496.454]^T$	7.930	0.405 %	0.143 %		
Keypoint11	$[1344 \ 202]^T$	$[1353.855 \ 197.637]^T$	10.778	0.513 %	0.404 %		
Keypoint12	$[1407 \ 264]^T$	$[1403.206 \ 265.449]^T$	4.062	0.198 %	0.134 %		
Mean Values			7.570	0.304 %	0.369 %		

Figure 6.3: Keypoints and projections for the *avloopback* prototype (scenario 1).



Source: The author.

### Distance Between Virtual and Estimated Cameras

Again, it is easy to obtain the camera estimated position,  $-\mathbf{b}$ :

$$\mathbf{u}_{PNP} = -\mathbf{b} = -\mathbf{R}^{-1} \cdot \mathbf{t} = \begin{bmatrix} -73.1469634 \\ -90.76721396 \\ -4.58755769 \end{bmatrix}$$

Thus, the distance between the VR camera and the estimated is:

$$\begin{aligned} \mathbf{u}_{VR} - \mathbf{u}_{PNP} &= \begin{bmatrix} -71.7399 \\ -88.1842 \\ -4.0516 \end{bmatrix} - \begin{bmatrix} -73.1448 \\ -90.7653 \\ -4.5871 \end{bmatrix} = \begin{bmatrix} 1.4050 \\ 2.5811 \\ 0.5355 \end{bmatrix} \\ d_{CAMS} &= \|\mathbf{u}_{VR} - \mathbf{u}_{PNP}\| = 2.9871 \end{aligned} \quad (6.11)$$

The unit quaternions for the virtual environment's camera before the loopback request,  $\phi_{VR}$ , and the estimated pose,  $\phi_{PNP}$  are:

$$\begin{aligned} \phi_{VR} &= 0.6832 + 0.6832\hat{\mathbf{i}} + 0.1823\hat{\mathbf{j}} + 0.1823\hat{\mathbf{k}} \\ \phi_{PNP} &= 0.6935 + 0.6731\hat{\mathbf{i}} + 0.1803\hat{\mathbf{j}} + 0.1833\hat{\mathbf{k}} \end{aligned}$$

Or, in vector form:

$$\begin{aligned} \hat{\mathbf{r}}_{VR} &= \begin{bmatrix} 0.6832 & 0.6832 & 0.1823 & 0.1823 \end{bmatrix}^T \\ \hat{\mathbf{r}}_{PNP} &= \begin{bmatrix} 0.6935 & 0.6731 & 0.1803 & 0.1833 \end{bmatrix}^T \end{aligned}$$

The distance between the two rotations is:

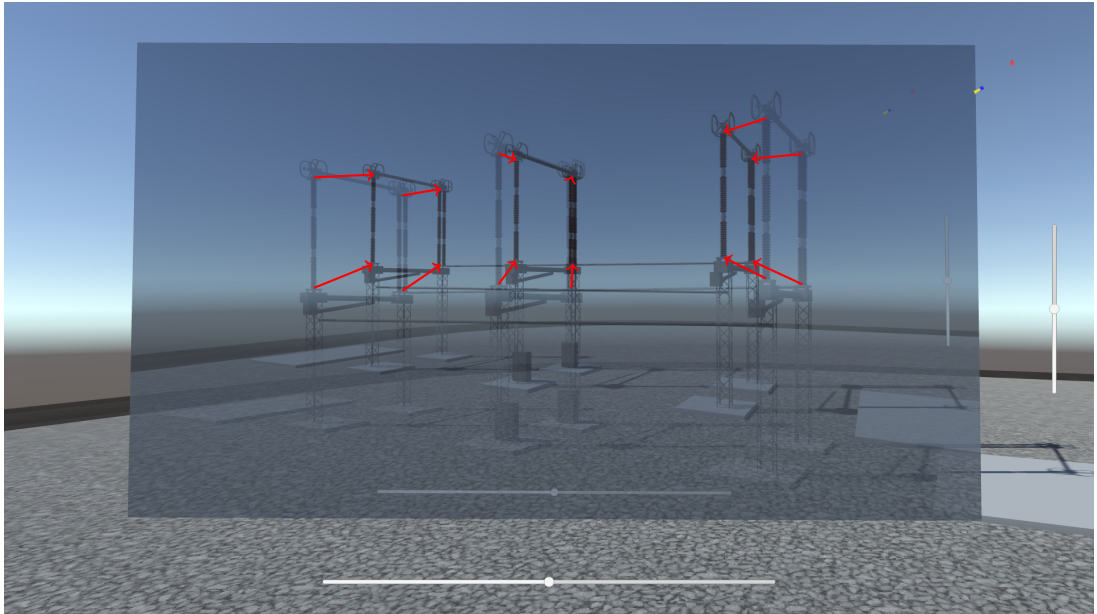
$$\Delta R = 1 - \|\hat{\mathbf{r}}_{VR} \cdot \hat{\mathbf{r}}_{PNP}\| = 1.067 \times 10^{-4}$$

### Distance Between keypoints in the Rendered AV Image

Finally, Table 6.8 shows the results for  $\overline{d_{AV}}$ ,  $\overline{\delta_{x\%}}$  and  $\overline{\delta_{y\%}}$ . The distances between the keypoints in the virtual models and in the overlay image after the rendering process is shown in Figure 6.4.

Table 6.8: *avloopback* scenario 1 – errors in the rendered image.

Identifier	Scene	Overlay	Distance $d_{AV}$	Relative Error	
				$ \delta_{x\%} $	$ \delta_{y\%} $
Keypoint1	$[546 \ 577]^T$	$[650.764 \ 619.439]^T$	112.946	5.453 %	3.922 %
Keypoint2	$[701 \ 572]^T$	$[769.804 \ 618.202]^T$	82.238	3.544 %	4.277 %
Keypoint3	$[546 \ 771]^T$	$[652.428 \ 776.636]^T$	106.483	5.540 %	0.456 %
Keypoint4	$[701 \ 739]^T$	$[770.852 \ 751.375]^T$	70.019	3.598 %	1.054 %
Keypoint5	$[871 \ 583]^T$	$[902.319 \ 625.268]^T$	51.556	1.601 %	3.833 %
Keypoint6	$[999 \ 576]^T$	$[1002.045 \ 619.923]^T$	43.037	0.114 %	3.980 %
Keypoint7	$[871 \ 814]^T$	$[903.174 \ 802.342]^T$	33.708	1.645 %	1.090 %
Keypoint8	$[999 \ 770]^T$	$[1000.952 \ 773.858]^T$	3.243	0.057 %	0.283 %
Keypoint9	$[1342 \ 593]^T$	$[1268.602 \ 630.713]^T$	82.830	3.859 %	3.428 %
Keypoint10	$[1406 \ 583]^T$	$[1319.276 \ 623.863]^T$	95.899	4.535 %	3.722 %
Keypoint11	$[1342 \ 875]^T$	$[1267.816 \ 851.648]^T$	78.581	3.900 %	2.206 %
Keypoint12	$[1406 \ 812]^T$	$[1316.941 \ 804.338]^T$	89.806	4.656 %	0.786 %
Mean Values			70.862	3.209 %	2.420 %

Figure 6.4: Keypoints in the rendered AV image for the *avloopback* prototype (scenario 1).

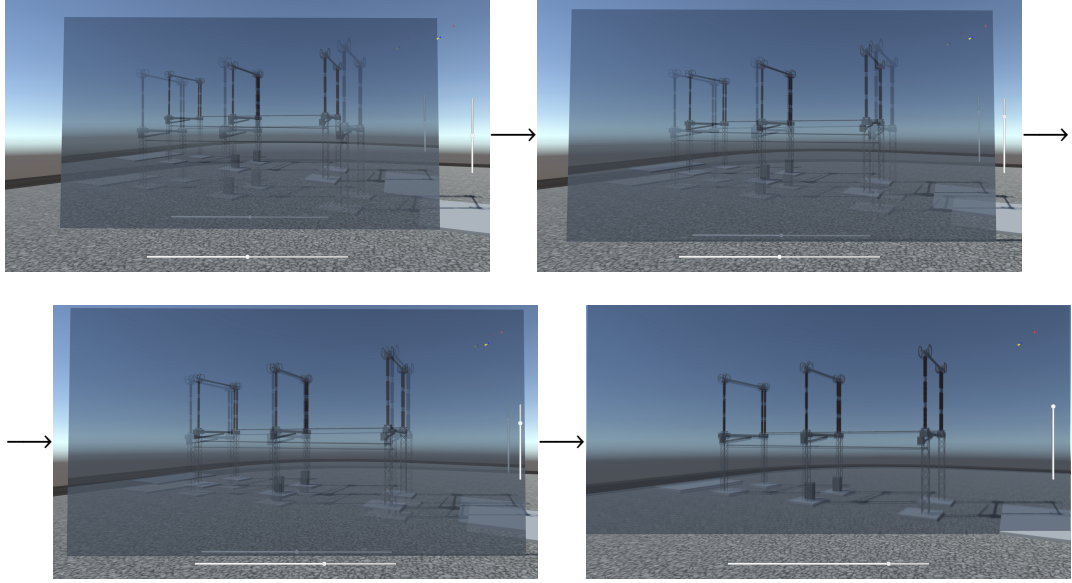
Source: The author.

The rectangular region's opacity was intentionally changed to improve error visualization. Even though the PNP solver can find a camera model which maps world keypoints to image keypoints with reasonably good results, considering  $\overline{d_{PNP}} = 7.57$  for a  $1920 \times 1080$  image, the overall 2D–3D registration is bad when setting  $C$  elements  $f_x = f_y = h_1$ .

### 6.3.2 Scenario 2: Adjustments for focal length and rectangular region size

In this scenario, the user moves the sliders to adjust the focal length and to apply a scale factor to the rectangular region, achieving optimal results (Figure 6.5).

Figure 6.5: Focal length and rectangular region size adjustments.



Source: The author.

The final adjustment for this experiment was:

$$f_x = f_y = 1.181538 \cdot 1080 = 1276.06104$$

resulting in:

$$C = \begin{bmatrix} 1276.0610 & 0 & 960 \\ 0 & 1276.0610 & 540 \\ 0 & 0 & 1 \end{bmatrix}.$$

The pose obtained from the PNP estimation is:

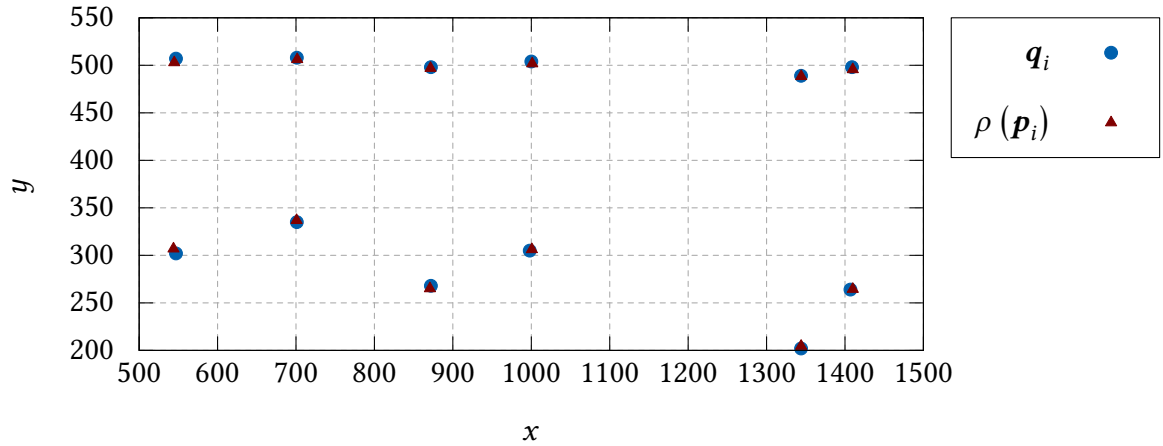
$$R = \begin{bmatrix} 0.865329 & -0.501196 & 0.002722 \\ 0.001771 & 0.008489 & 0.999962 \\ -0.501201 & -0.865292 & 0.008234 \end{bmatrix}, t = \begin{bmatrix} 17.82752 \\ 5.10908 \\ -112.07246 \end{bmatrix}$$

#### Mean Euclidean distance and relative errors

Table 6.9 shows the results for  $\overline{d_{PNP}}$ ,  $\overline{e_{x\%}}$  and  $\overline{e_{y\%}}$ . The image keypoints and their associated projections are also plotted in Figure 6.6.

Table 6.9: *avloopback* scenario 2 – PNP errors.

Identifier	Image	Projection	Distance $d_{PNP}$	Relative Error	
				$ e_x\% $	$ e_y\% $
Keypoint1	$\begin{bmatrix} 547 & 507 \end{bmatrix}^T$	$\begin{bmatrix} 544.788 & 502.934 \end{bmatrix}^T$	4.628	0.115 %	0.376 %
Keypoint2	$\begin{bmatrix} 701 & 508 \end{bmatrix}^T$	$\begin{bmatrix} 701.520 & 506.267 \end{bmatrix}^T$	1.809	0.027 %	0.160 %
Keypoint3	$\begin{bmatrix} 547 & 302 \end{bmatrix}^T$	$\begin{bmatrix} 543.730 & 306.921 \end{bmatrix}^T$	5.908	0.170 %	0.456 %
Keypoint4	$\begin{bmatrix} 701 & 335 \end{bmatrix}^T$	$\begin{bmatrix} 700.776 & 336.901 \end{bmatrix}^T$	1.914	0.012 %	0.176 %
Keypoint5	$\begin{bmatrix} 872 & 498 \end{bmatrix}^T$	$\begin{bmatrix} 871.536 & 496.985 \end{bmatrix}^T$	1.117	0.024 %	0.094 %
Keypoint6	$\begin{bmatrix} 1000 & 504 \end{bmatrix}^T$	$\begin{bmatrix} 1001.220 & 501.763 \end{bmatrix}^T$	2.548	0.064 %	0.207 %
Keypoint7	$\begin{bmatrix} 872 & 268 \end{bmatrix}^T$	$\begin{bmatrix} 870.772 & 265.087 \end{bmatrix}^T$	3.161	0.064 %	0.270 %
Keypoint8	$\begin{bmatrix} 998 & 305 \end{bmatrix}^T$	$\begin{bmatrix} 1000.740 & 306.256 \end{bmatrix}^T$	3.014	0.143 %	0.116 %
Keypoint9	$\begin{bmatrix} 1344 & 489 \end{bmatrix}^T$	$\begin{bmatrix} 1344.185 & 488.378 \end{bmatrix}^T$	0.649	0.010 %	0.058 %
Keypoint10	$\begin{bmatrix} 1409 & 498 \end{bmatrix}^T$	$\begin{bmatrix} 1409.933 & 495.621 \end{bmatrix}^T$	2.555	0.049 %	0.220 %
Keypoint11	$\begin{bmatrix} 1344 & 202 \end{bmatrix}^T$	$\begin{bmatrix} 1344.116 & 204.539 \end{bmatrix}^T$	2.541	0.006 %	0.235 %
Keypoint12	$\begin{bmatrix} 1407 & 264 \end{bmatrix}^T$	$\begin{bmatrix} 1409.975 & 264.449 \end{bmatrix}^T$	3.009	0.155 %	0.042 %
Mean Values			2.738	0.070%	0.201%

Figure 6.6: Keypoints and projections for the *avloopback* prototype (scenario 2).

Source: The author.

### Distance between virtual and estimated cameras

The camera estimated pose is:

$$\mathbf{u}_{PNP} = -\mathbf{b} = -\mathbf{R}^{-1} \cdot \mathbf{t} = \begin{bmatrix} -71.6065 \\ -88.0837 \\ -4.2346 \end{bmatrix}$$

Thus, the distance between the VR camera and the estimated is:

$$\begin{aligned} \mathbf{u}_{VR} - \mathbf{u}_{PNP} &= \begin{bmatrix} -71.7399 \\ -88.1842 \\ -4.0516 \end{bmatrix} - \begin{bmatrix} -71.6065 \\ -88.0837 \\ -4.2346 \end{bmatrix} = \begin{bmatrix} -0.1334 \\ -0.1005 \\ 0.1830 \end{bmatrix} \\ d_{CAMS} &= \|\mathbf{u}_{VR} - \mathbf{u}_{PNP}\| = 0.24775 \end{aligned} \quad (6.12)$$

The unit quaternions for the virtual environment's camera before the loopback request,  $\hat{\mathbf{r}}_{VR}$ , and the estimated pose,  $\hat{\mathbf{r}}_{PNP}$ , are:

$$\begin{aligned} \hat{\mathbf{r}}_{VR} &= \begin{bmatrix} 0.6832 & 0.6832 & 0.1823 & 0.1823 \end{bmatrix}^T \\ \hat{\mathbf{r}}_{PNP} &= \begin{bmatrix} 0.6859 & 0.6798 & 0.1837 & 0.1833 \end{bmatrix}^T \end{aligned}$$

The distance between the two rotations is:

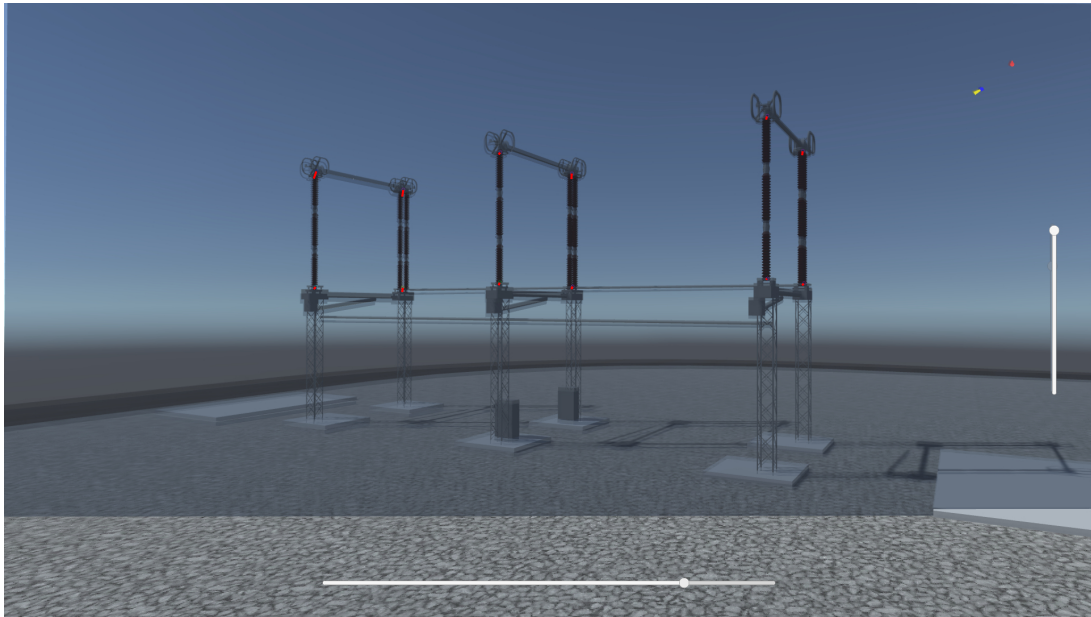
$$\Delta R = 1 - \|\hat{\mathbf{r}}_{VR} \cdot \hat{\mathbf{r}}_{PNP}\| = 1.103 \times 10^{-5}$$

### Distance between keypoints in the rendered AV image

Finally, Table 6.10 shows the results for  $\overline{d_{AV}}$ ,  $\overline{\delta_{x\%}}$  and  $\overline{\delta_{y\%}}$ . The distances between the keypoints in the virtual models and in the overlay image after the rendering process is shown in Figure 6.7. The rectangular region scale factor was adjusted to 1.5285.

Table 6.10: *avloopback* scenario 2 – errors in the rendered image.

Identifier	Scene	Overlay	Distance    Relative Error		
			$d_{AV}$	$ \delta_{x\%} $	$ \delta_{y\%} $
Keypoint1	$\begin{bmatrix} 546 & 577 \end{bmatrix}^T$	$\begin{bmatrix} 549.368 & 580.992 \end{bmatrix}^T$	5.118	0.172 %	0.362 %
Keypoint2	$\begin{bmatrix} 701 & 572 \end{bmatrix}^T$	$\begin{bmatrix} 702.839 & 579.449 \end{bmatrix}^T$	7.517	0.056 %	0.689 %
Keypoint3	$\begin{bmatrix} 546 & 771 \end{bmatrix}^T$	$\begin{bmatrix} 550.484 & 784.324 \end{bmatrix}^T$	13.368	0.230 %	1.168 %
Keypoint4	$\begin{bmatrix} 701 & 739 \end{bmatrix}^T$	$\begin{bmatrix} 703.616 & 751.544 \end{bmatrix}^T$	11.700	0.096 %	1.070 %
Keypoint5	$\begin{bmatrix} 871 & 583 \end{bmatrix}^T$	$\begin{bmatrix} 873.456 & 588.618 \end{bmatrix}^T$	5.100	0.097 %	0.439 %
Keypoint6	$\begin{bmatrix} 999 & 576 \end{bmatrix}^T$	$\begin{bmatrix} 1001.740 & 581.771 \end{bmatrix}^T$	5.183	0.098 %	0.447 %
Keypoint7	$\begin{bmatrix} 871 & 814 \end{bmatrix}^T$	$\begin{bmatrix} 874.244 & 817.498 \end{bmatrix}^T$	4.306	0.138 %	0.314 %
Keypoint8	$\begin{bmatrix} 999 & 770 \end{bmatrix}^T$	$\begin{bmatrix} 1000.350 & 780.552 \end{bmatrix}^T$	9.759	0.026 %	0.902 %
Keypoint9	$\begin{bmatrix} 1342 & 593 \end{bmatrix}^T$	$\begin{bmatrix} 1344.072 & 595.776 \end{bmatrix}^T$	2.494	0.072 %	0.193 %
Keypoint10	$\begin{bmatrix} 1406 & 583 \end{bmatrix}^T$	$\begin{bmatrix} 1409.012 & 586.981 \end{bmatrix}^T$	4.253	0.139 %	0.307 %
Keypoint11	$\begin{bmatrix} 1342 & 875 \end{bmatrix}^T$	$\begin{bmatrix} 1344.204 & 881.092 \end{bmatrix}^T$	5.820	0.078 %	0.521 %
Keypoint12	$\begin{bmatrix} 1406 & 812 \end{bmatrix}^T$	$\begin{bmatrix} 1407.118 & 819.806 \end{bmatrix}^T$	7.023	0.040 %	0.646 %
<b>Mean Values</b>			6.803	0.104 %	0.588 %

Figure 6.7: Keypoints in the rendered AV image for the *avloopback* prototype (scenario 1).

Source: The author.

With the proposed adjustment, much better results can be achieved, in comparison to the previous scenario. This is particularly useful for fixed cameras or PTZ cameras with presets, for which the intrinsic parameters are not available and when calibration routines *in loco* might not be a viable option. Besides, this method can be used to combine images from cameras with different FoVs (field of view), as is the case for the virtual camera and the field camera.

## 6.4 Field Camera Prototype (*avcamera*)

This prototype uses a field image targeting the same type of power disconnecter switch used in the loopback solution.

Table 6.11 presents the experiment parameters. In this prototype, there's no ground-truth value for the virtual camera target position. However, the VR application knows which asset is associated with that specific photo. In this prototype, that metadata is stored within the VR application settings.

Table 6.11: *avcamera* experiment parameters.

Description	Symbol	Value
Image width (pixels)	$w_1$	720
Image height (pixels)	$h_1$	624
Rectangular Region width (VR units)	$w_2$	14.4
Rectangular Region height (VR units)	$h_2$	12.48
Equipment ID	$g_{ID}$	7U8

This prototype uses the same keypoints naming convention of the previous solution (see Figure 5.8). However, this time, the camera is located behind the power disconnecter, mirroring the coordinates pattern on the  $x$ -axis. For this particular model, there's no ambiguity, since the part with double insulators gets even keypoint numbers.

The keypoints coordinates in the virtual environment and the image are shown in Table 6.12.

Table 6.12: *avcamera* prototype keypoints.

Identifier	WCS	ICS
Keypoint 1	$\mathbf{p}_1 = [60.293 \quad 140.799 \quad -4.459]^T$	$\mathbf{q}_1 = [497 \quad 518 \quad 1]^T$
Keypoint 2	$\mathbf{p}_2 = [60.293 \quad 145.243 \quad -4.459]^T$	$\mathbf{q}_2 = [562 \quad 512 \quad 1]^T$
Keypoint 3	$\mathbf{p}_3 = [60.293 \quad 140.758 \quad -8.306]^T$	$\mathbf{q}_3 = [504 \quad 437 \quad 1]^T$
Keypoint 4	$\mathbf{p}_4 = [60.293 \quad 145.243 \quad -8.306]^T$	$\mathbf{q}_4 = [562 \quad 418 \quad 1]^T$
Keypoint 5	$\mathbf{p}_5 = [52.803 \quad 140.799 \quad -4.459]^T$	$\mathbf{q}_5 = [367 \quad 518 \quad 1]^T$
Keypoint 6	$\mathbf{p}_6 = [52.803 \quad 145.243 \quad -4.459]^T$	$\mathbf{q}_6 = [418 \quad 505 \quad 1]^T$
Keypoint 7	$\mathbf{p}_7 = [52.803 \quad 140.758 \quad -8.306]^T$	$\mathbf{q}_7 = [367 \quad 424 \quad 1]^T$
Keypoint 8	$\mathbf{p}_8 = [52.803 \quad 145.243 \quad -8.306]^T$	$\mathbf{q}_8 = [418 \quad 399 \quad 1]^T$
Keypoint 9	$\mathbf{p}_9 = [45.310 \quad 140.799 \quad -4.459]^T$	$\mathbf{q}_9 = [223 \quad 499 \quad 1]^T$
Keypoint 10	$\mathbf{p}_{10} = [45.336 \quad 145.243 \quad -4.459]^T$	$\mathbf{q}_{10} = [238 \quad 499 \quad 1]^T$
Keypoint 11	$\mathbf{p}_{11} = [45.310 \quad 140.758 \quad -8.306]^T$	$\mathbf{q}_{11} = [223 \quad 412 \quad 1]^T$
Keypoint 12	$\mathbf{p}_{12} = [45.310 \quad 145.243 \quad -8.306]^T$	$\mathbf{q}_{12} = [245 \quad 381 \quad 1]^T$

The camera matrix is constructed so that the principal point is located in the image center and  $f_x = f_y = h_1$  (no adjustment is made):

$$\mathbf{C} = \begin{bmatrix} 624 & 0 & 360 \\ 0 & 624 & 312 \\ 0 & 0 & 1 \end{bmatrix}$$

The pose obtained from the PNP estimation is:

$$\mathbf{R} = \begin{bmatrix} 0.909636 & 0.415194 & -0.013279 \\ 0.162019 & -0.325165 & 0.931675 \\ 0.382508 & -0.849636 & -0.363051 \end{bmatrix}, \mathbf{t} = \begin{bmatrix} -105.97104 \\ 50.51812 \\ 126.85941 \end{bmatrix}$$

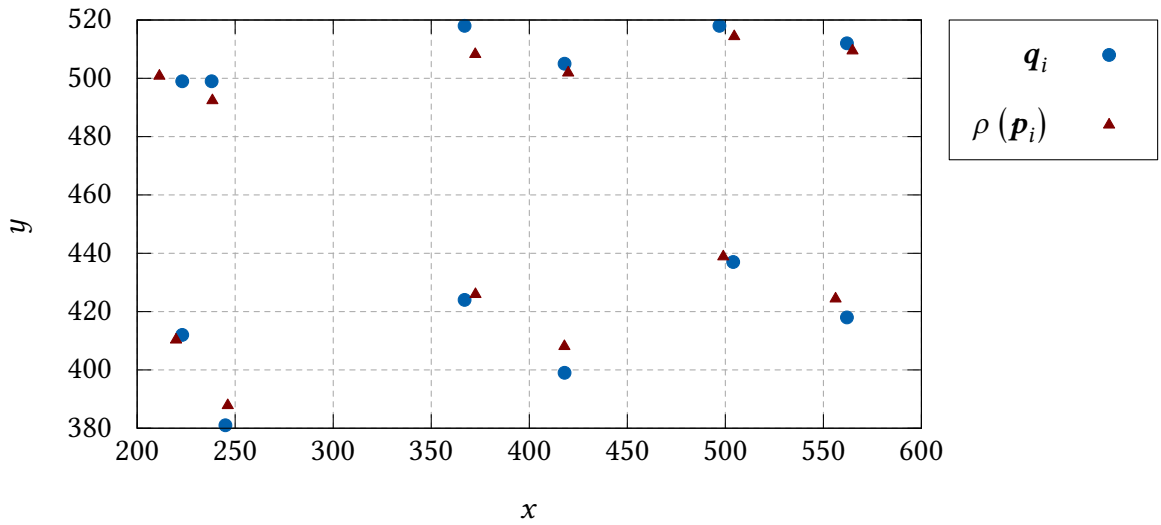
### 6.4.1 Mean Euclidean distance and relative errors

Table 6.13 shows the results for  $\overline{d_{PNP}}$ ,  $\overline{e_{x\%}}$  and  $\overline{e_{y\%}}$ . The image keypoints and their associated projections are also plotted in Figure 6.8.

Table 6.13: *avcamera* PNP errors.

Identifier	Image	Projection	Distance	Relative Error	
			$d_{PNP}$	$ e_{x\%} $	$ e_{y\%} $
Keypoint1	$[497 \ 518]^T$	$[504.534 \ 514.366]^T$	8.364	1.046 %	0.582 %
Keypoint2	$[562 \ 512]^T$	$[564.848 \ 509.475]^T$	3.806	0.396 %	0.405 %
Keypoint3	$[504 \ 437]^T$	$[498.966 \ 438.855]^T$	5.365	0.699 %	0.297 %
Keypoint4	$[562 \ 418]^T$	$[556.240 \ 424.409]^T$	8.617	0.800 %	1.027 %
Keypoint5	$[367 \ 518]^T$	$[372.430 \ 508.256]^T$	11.155	0.754 %	1.561 %
Keypoint6	$[418 \ 505]^T$	$[419.846 \ 501.898]^T$	3.609	0.256 %	0.497 %
Keypoint7	$[367 \ 424]^T$	$[372.544 \ 425.934]^T$	5.872	0.770 %	0.310 %
Keypoint8	$[418 \ 399]^T$	$[417.907 \ 408.091]^T$	9.092	0.013 %	1.457 %
Keypoint9	$[223 \ 499]^T$	$[211.345 \ 500.807]^T$	11.794	1.619 %	0.290 %
Keypoint10	$[238 \ 499]^T$	$[238.393 \ 492.417]^T$	6.594	0.055 %	1.055 %
Keypoint11	$[223 \ 412]^T$	$[219.821 \ 410.326]^T$	3.593	0.441 %	0.268 %
Keypoint12	$[245 \ 381]^T$	$[246.197 \ 387.836]^T$	6.940	0.166 %	1.095 %
Mean Values			7.067	0.585%	0.737%

Figure 6.8: Keypoints and projections for the *avcamera* prototype.



Source: The author.

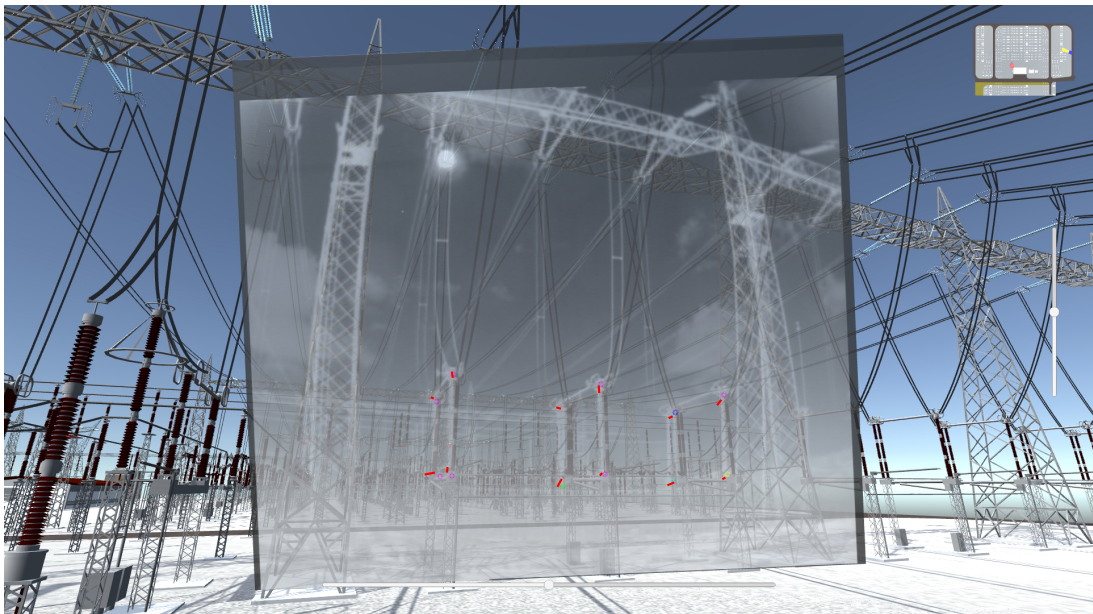
### 6.4.2 Distance between keypoints in the rendered AV image

Finally, Table 6.14 shows the results for  $\overline{d_{AV}}$ ,  $\overline{\delta_{x\%}}$  and  $\overline{\delta_{y\%}}$ . The distances between the keypoints in the virtual models and in the overlay image after the rendering process is shown in Figure 6.9.

Table 6.14: *avcamera* – errors in the rendered image.

Identifier	Scene	Overlay	Distance			Relative Error	
			$d_{AV}$	$ \delta_{x\%} $	$ \delta_{y\%} $		
Keypoint1	$[1180 \ 239]^T$	$[1168.928 \ 234.836]^T$	12.984	0.625 %	0.460 %		
Keypoint2	$[1271 \ 248]^T$	$[1265.727 \ 245.521]^T$	6.206	0.286 %	0.268 %		
Keypoint3	$[1170 \ 352]^T$	$[1178.018 \ 356.315]^T$	7.854	0.367 %	0.321 %		
Keypoint4	$[1256 \ 375]^T$	$[1263.796 \ 385.519]^T$	12.216	0.380 %	0.907 %		
Keypoint5	$[982 \ 246]^T$	$[975.068 \ 232.129]^T$	16.004	0.403 %	1.297 %		
Keypoint6	$[1053 \ 256]^T$	$[1050.199 \ 251.847]^T$	5.988	0.184 %	0.447 %		
Keypoint7	$[981 \ 369]^T$	$[973.116 \ 372.126]^T$	8.519	0.422 %	0.242 %		
Keypoint8	$[1048 \ 397]^T$	$[1047.992 \ 410.514]^T$	13.316	0.044 %	1.231 %		
Keypoint9	$[741 \ 253]^T$	$[759.275 \ 257.120]^T$	18.334	0.940 %	0.302 %		
Keypoint10	$[781 \ 267]^T$	$[780.815 \ 257.421]^T$	9.622	0.041 %	0.888 %		
Keypoint11	$[752 \ 389]^T$	$[757.436 \ 387.787]^T$	5.741	0.283 %	0.172 %		
Keypoint12	$[791 \ 423]^T$	$[789.094 \ 434.909]^T$	11.168	0.102 %	1.018 %		
Mean Values			10.663	0.340 %	0.629 %		

Figure 6.9: Keypoints in the rendered AV *image* for the *avcamera* prototype .



Source: The author.

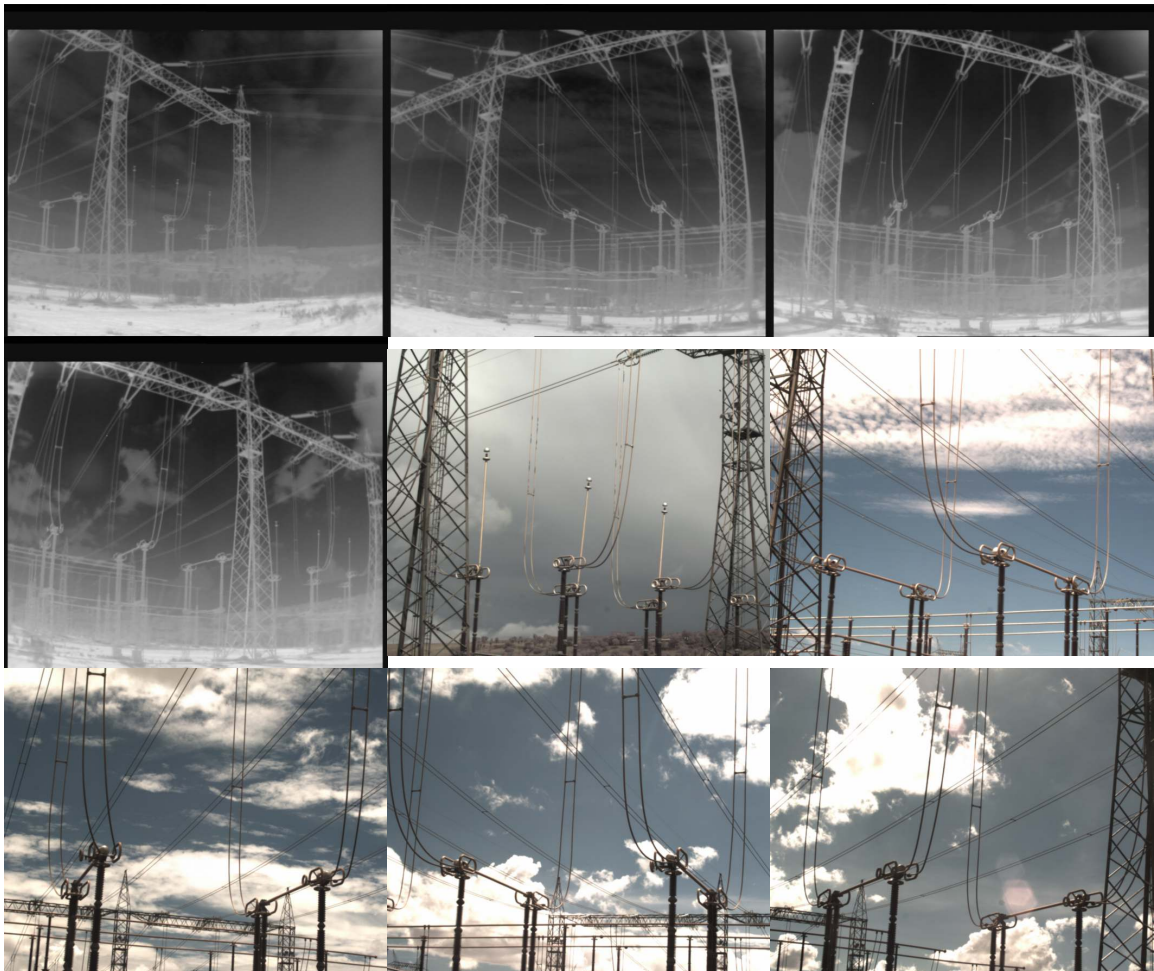
As with the previous prototype, the rectangular region's opacity was intentionally changed to improve error visualization.

## 6.5 Final Prototype (avcemig)

The final prototype used the same virtual environment of the Nova Ponte's substation (Figure 5.10). The substation has a pair of cameras (RGB and Thermal) with PTZ control. They have five adjustments presets, which, properly multiplexed in time, can be used to monitor three distinct power disconnectors. The thermal camera always captures, in a single pose, the full geometry of the asset of interest. The color cameras, however, are adjusted to zoom levels requiring more than one pose to capture some of the assets.

The CEMIG-GT company kindly provided a dataset with 561 images captured by these cameras. From this set, 123 images were ignored since they correspond to images taken by RGB cameras without any favorable light conditions, at night. A sample of the dataset is presented in Figure 6.10.

Figure 6.10: Thermal and RGB image dataset sample.



Source: The author.

The images also have metadata for their timestamps, ranging from 31 December 2017 23:03 and 1 January 2018 22:46. This allows emulation of real-time data, by applying some time offset in the system clock. The thermal camera images' size is  $720 \times 624$  pixels, whereas RGB images' dimensions are  $1280 \times 1024$ .

Regarding the keypoints coordinates in image space, they were manually specified for each camera's pose, using a single photo captured at that pose. Hence, mobile cameras or PTZ cameras with significant repositioning errors were not considered in our experiments.

### 6.5.1 Experiments details

Both calibration and registration were evaluated for eight different combinations of cameras and poses. Due to zoom levels, the RGB camera needs more poses for capturing the full geometry of some assets. Table 6.15 summarizes the conditions for each experiment.

Table 6.15: Experiments codes.

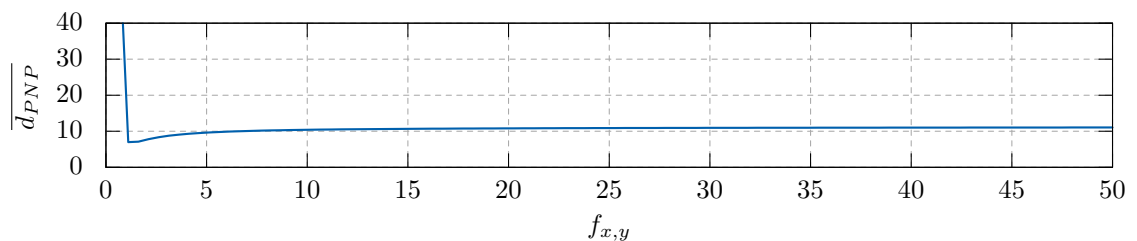
Code	Camera Type	Asset	Detail	Keypoints
T1	thermal	1	full	12
T2	thermal	2	full	12
T3	thermal	3	full	12
C1A	color	1	lines A and B	4
C1B	color	1	lines B and C	4
C2A	color	2	lines A and B	4
C2B	color	2	lines B and C	4
C3	color	3	full	6

### 6.5.2 Mean Euclidean distance and relative errors

This prototype uses the ternary-search algorithm for obtaining the optimal focal length scale factor. For this reason, the values for these metrics are changed during the algorithm's execution.

The range for the search algorithm was determined empirically, from the inspection of the  $\overline{d_{PNP}}$  values in a wide range, as shown in Figure 6.11 for experiment T1.

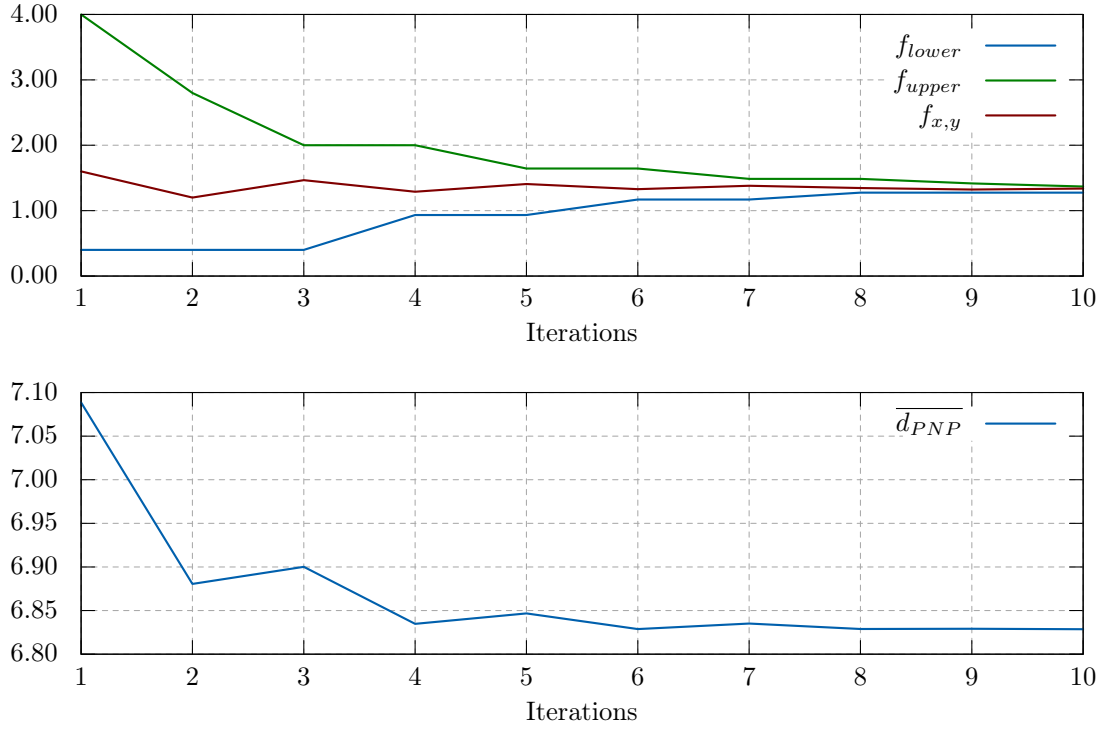
Figure 6.11: Focal length scale factor impact on  $\overline{d_{PNP}}$  metric.



Source: The author.

Figure 6.12 shows values during the execution of 10 iterations, for the experiment T1 and the range  $f_{x,y} \in [0.4, 4]$ .

Figure 6.12: Ternary-search iterations



Source: The author.

Data for other poses, along with some collected metrics, are summarized in Table 6.16. The algorithm was parameterized for running at most 50 iterations, also stopping in the  $n$ -th iteration whenever  $f_4 - f_1 < 10^{-5}$ . Finally, these metrics were also computed considering the set of keypoints from all experiments at once. The last row in the table contains the standard deviation ( $\sigma$ ) associated with each metric.

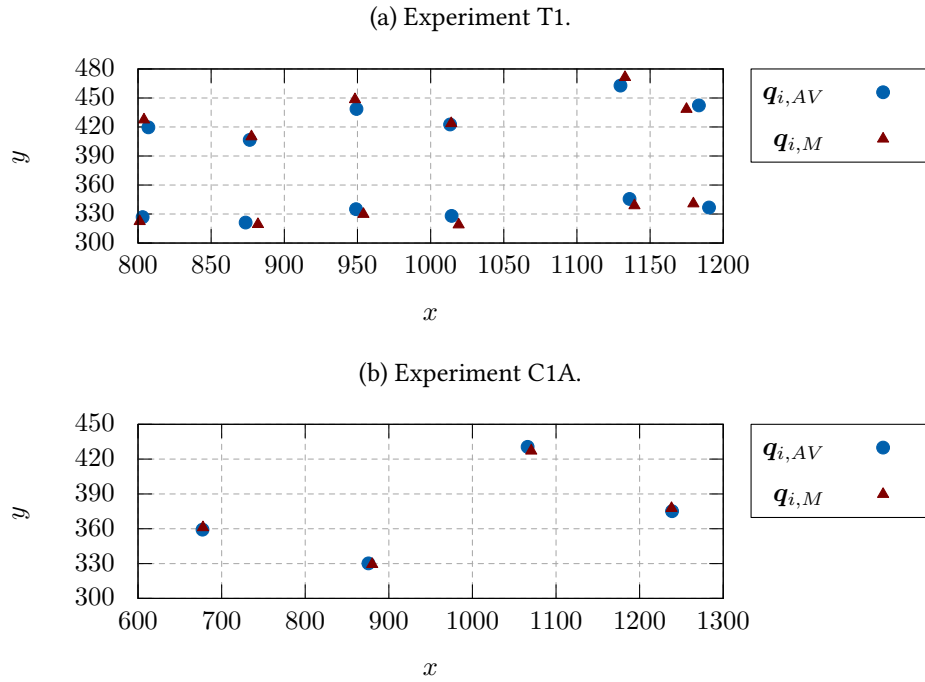
Table 6.16: *Perspective-n-Point* and calibration results.

Exp.	$n$	$f$	$\overline{d_{PNP}}$	$\overline{e_{x\%}}$	$\overline{e_{y\%}}$
T1	32	1.34	6.83	0.526 %	0.806 %
T2	32	0.94	6.97	0.590 %	0.727 %
T3	30	1.10	6.88	0.432 %	0.838 %
C1A	32	1.29	6.39	0.344 %	0.448 %
C1B	32	1.79	0.71	0.018 %	0.065 %
C2A	32	1.80	2.83	0.060 %	0.266 %
C2B	32	2.13	5.40	0.234 %	0.437 %
C3	30	2.00	16.10	1.206 %	0.277 %
<b>All keypoints</b>			7.00	0.490 %	0.603 %
$\sigma$			5.27	0.506 %	0.471 %

### 6.5.3 Distance between keypoints in the rendered AV image

Concerning the resulting rendered image after the registration, the keypoints' coordinates in pixels were extracted for both the rectangular region (overlay) and the virtual model instances. Figure 6.13 shows the resulting coordinates for experiments T1 and C1A. For the quality metrics, Table 6.17 shows the collected values for  $\overline{d_{AV}}$ ,  $\overline{\delta_{x\%}}$ , and  $\overline{\delta_{y\%}}$  for all experiments. Again, the overall values for mean and standard deviation, considering the set of all keypoints from all cameras, are also presented.

Figure 6.13: Keypoints discrepancy.



Source: The author.

Table 6.17: *avcemig* – final registration results.

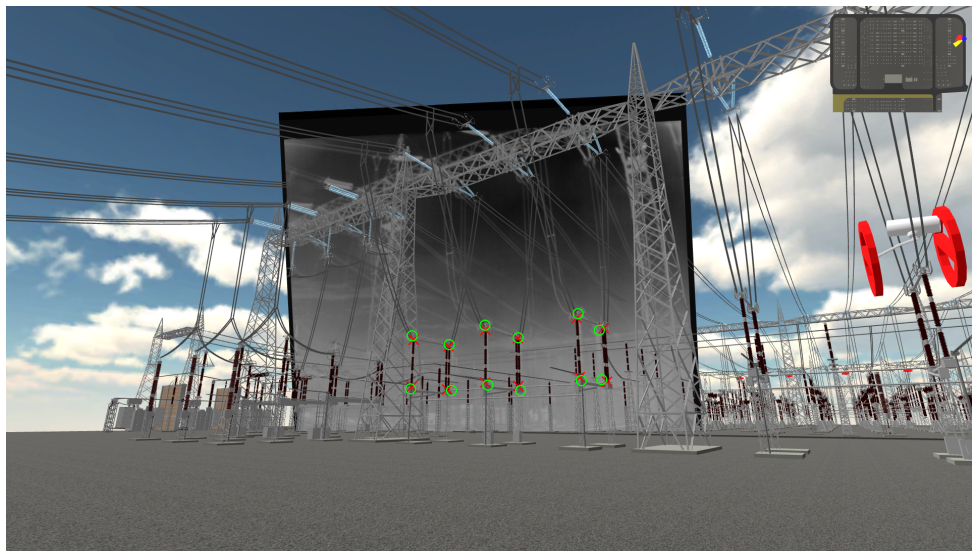
Exp.	$\overline{d_{AV}}$	$\overline{\delta_{x\%}}$	$\overline{\delta_{y\%}}$
T1	7.64	0.226 %	0.509 %
T2	11.27	0.370 %	0.657 %
T3	10.74	0.254 %	0.735 %
C1A	3.68	0.130 %	0.200 %
C1B	1.77	0.050 %	0.132 %
C2A	1.76	0.045 %	0.137 %
C2B	2.78	0.092 %	0.153 %
C3	6.39	0.318 %	0.143 %
<b>All keypoints</b>	7.483	0.231 %	0.451 %
$\sigma$	5.099	0.224 %	0.418 %

Despite having more keypoints, errors were higher for the thermal camera. This is related with the images' curvature and the fact that, due to zoom levels, the keypoints are far away from the camera. In this scenario, small deviations in the pixels coordinates corresponds to arguably big distances in world space.

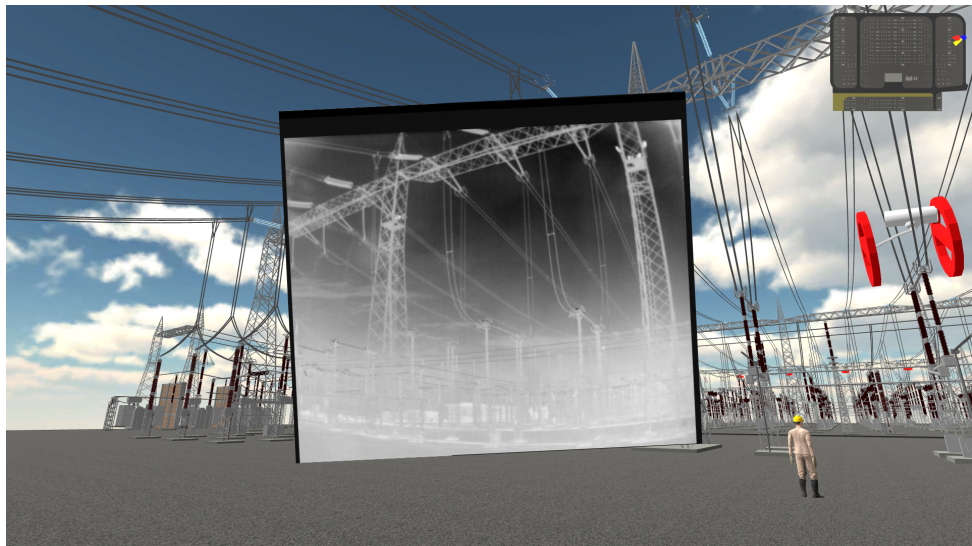
A custom shader was applied to the overlay plane to hide VR objects within its region. Figure 6.14 shows the rendered images for both standard and custom shaders. In the former, keypoints are highlighted with red crosses (virtual model) and green circles (overlay rectangular region).

Figure 6.14: 2D–3D registration for experiment T1.

(a) Keypoints discrepancy.



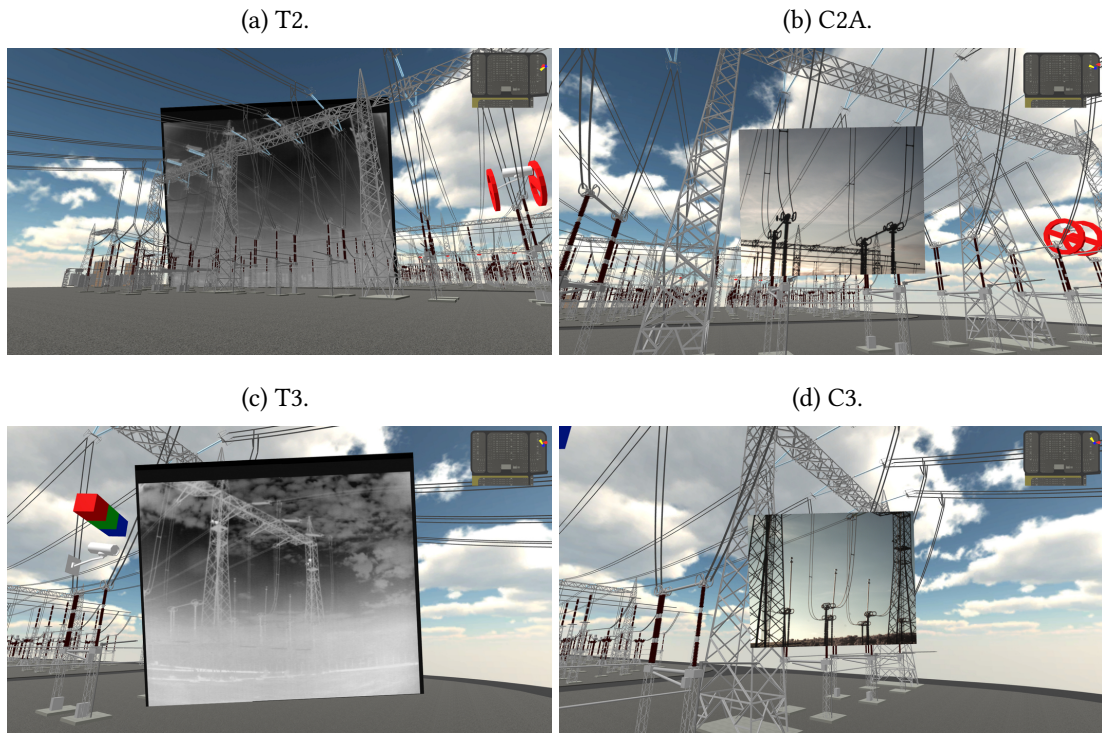
(b) Custom shader overlay.



*Source:* The author.

Some other registrations are shown in Figure 6.15.

Figure 6.15: Power disconnector registrations for other experiments.



Source: The author.

#### 6.5.4 System performance

The performance metrics were measured in two scenarios. In the first one, a single machine was used as client and server, for measuring the average duration of calibration routines and image requests. The second scenario involves three clients making alternate and simultaneous requests. The machines' configuration is shown in Table 6.18.

##### Scenario A – One client

In this scenario, the same physical machine *A* was used as the registration server and the AV client. Thus, it is expected that network-related delays are minimal. For each combination of cameras and poses, the calibration request was performed three times, using the same criterion described before for stopping the algorithm, i.e., a limit of 50 iterations or  $f_4 - f_1 < 10^{-5}$ , whichever occurs first. The mean values are summarized in the column  $\overline{\Delta t_{calib}}$  of Table 6.19.

For the 2D–3D spatial registration, HTTP requests were separated by an interval of at least 50 milliseconds. The mean value for each pose is also shown in Table 6.19, considering 40 consecutive requests.

Table 6.18: Hardware used in the performance experiments.

Name	Role(s)	Type	Specifications
A	Server and Client	Desktop	<b>CPU:</b> Core i5-7400 <b>System memory:</b> 16 GB DDR4-2400 <b>Video card:</b> NVIDIA GeForce GT 1030 2GB GDDR5 <b>Network connection:</b> Ethernet <b>Operating System:</b> Windows 10 Home
B	Client	Laptop	<b>CPU:</b> Core i5-7200U <b>System memory:</b> 8 GB DDR4-2133 <b>Video card:</b> NVIDIA GeForce 940MX <b>Network connection:</b> Ethernet <b>Operating System:</b> Windows 10 Home
C	Client	Laptop	<b>CPU:</b> Core i7-4500U <b>System memory:</b> 8 GB DDR3-1600 <b>Video card:</b> Intel HD Graphics 4400 <b>Network connection:</b> Wi-Fi <b>Operating System:</b> Windows 10 Home

Table 6.19: *avcemig* – registration performance.

Exp.	$\overline{\Delta t_{calib}}$ (ms)	$\overline{\Delta t_{2D,3D}}$ (ms)
T1	586.92	272.27
T2	553.92	289.14
T3	594.24	275.77
C1A	582.61	336.50
C1B	566.97	333.91
C2A	587.85	336.57
C2B	571.45	335.67
C3	561.77	337.36
<b>All</b>	575.72	314.65
$\sigma$	14.28	29.87

The values obtained for  $\overline{\Delta t_{2D,3D}}$  may be adequate for some real-time remote inspection applications. The opening or closing of a power disconnecter, for instance, is relatively slow motion, so that a few frames per second is more than enough to follow the events. After a command to toggle the state of a power disconnecter, triggered remotely, the operations center might assert that the device was properly switched.

In contrast, for very fast events, such as an explosion, this structure is inadequate. In this case, making one HTTP request per image would be very expensive, and so a dedicated video streaming channel should be used instead. Still, the main value of performing 2D–3D registration for such a disaster is not for real-time operation, but post-event analysis (BUI et al., 2015). It should be noted that even the time needed for the operator to correctly interpret the situation, after the 2D–3D registration is performed, might be considerably longer than just a

fraction of a second.

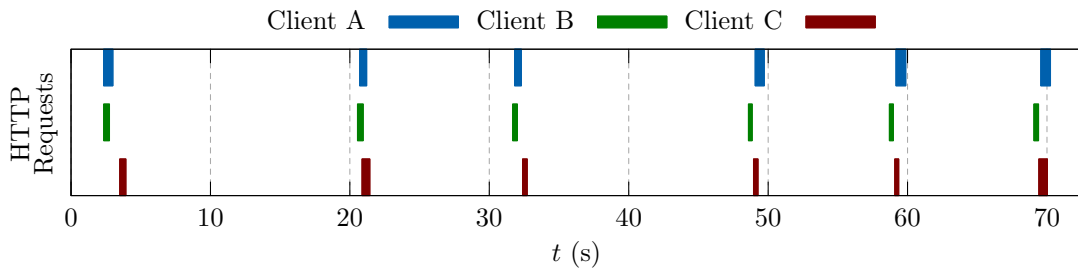
As expected, requests related to RGB cameras took longer. Although these cameras provide images with smaller dimensions ( $720 \times 624$  instead of  $1280 \times 1024$ ), each pixel is stored within 24 bits, one byte per channel. Besides, the use of Base64 encoding in the XML response contributed to bigger packet sizes being sent over HTTP.

Finally, considering the time needed for running the calibration algorithm, the measured values suggest that, depending on the application, this might also be performed in real-time, for cameras with some mobility. If the camera motion is sufficiently slow, one option is to periodically redefine the camera matrix by running the optimization algorithm according to some fixed-rate (one run per  $n$  frames). However, this would still imply the detection of the keypoints in real-time, which can make use of techniques such as the one presented by Pereira et al. (2016).

### Scenario B – Three clients

In this scenario, three clients were used for testing the calibration and registration requests. Figure 6.16 shows the timeline for the calibration requests. The mean and standard deviation values for each client and all requests at once are summarized in Table 6.20.

Figure 6.16: Multiple calibration requests.



Source: The author.

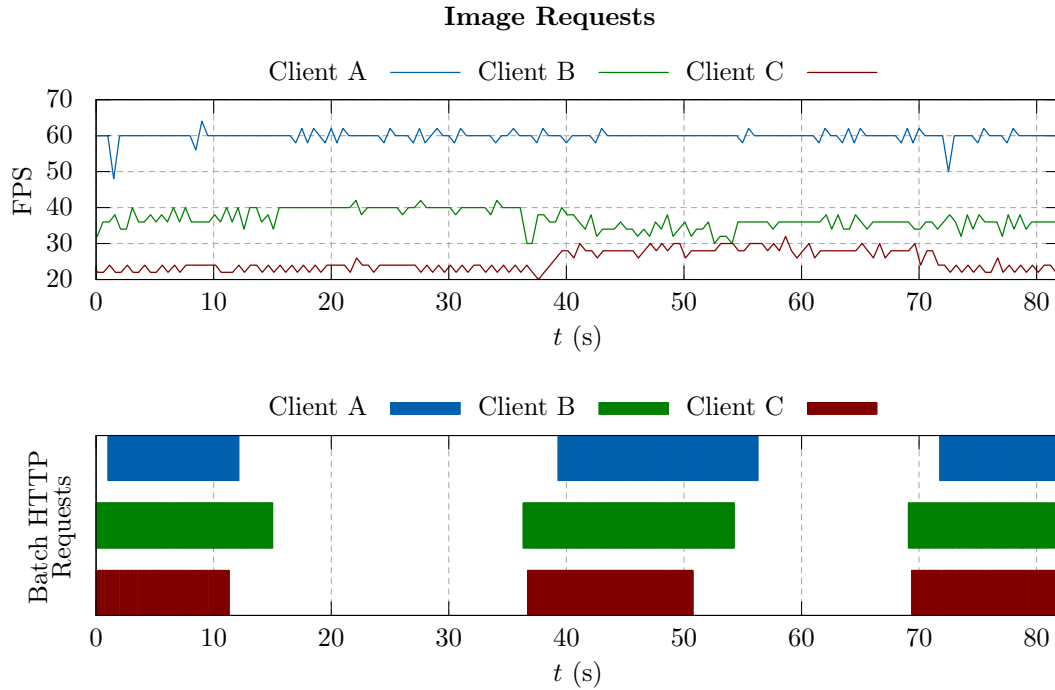
Table 6.20: Mean and standard deviation values for multiple calibration requests.

Client	$\overline{\Delta t_{calib}}$ (ms)	$\sigma$
A	615.62	98.63
B	337.72	60.45
C	424.25	136.85
<b>All</b>	459.19	154.01

Concerning the time needed for the registration requests, the experiment session's timeline is presented in Figure 6.17. The renderer's frame rate was measured during all the session, allowing to check whether these requests have any impact on the real-time rendering. For

clients A and B, the screen resolution was  $1920 \times 1080$ , in contrast with client C's smaller resolution of  $1366 \times 768$ .

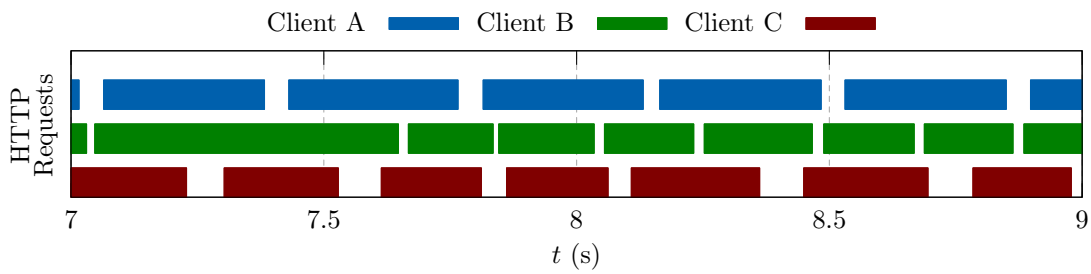
Figure 6.17: Multiple registration requests.



Source: The author.

The HTTP requests were separated in the timeline by, at least, 50 milliseconds. A detailed view of an interval is shown in Figure 6.18.

Figure 6.18: Detailed view of the registration requests.



Source: The author.

The mean and standard deviation values for each client are summarized in Table 6.21. For most requests, no significant impact on the frame rate was observed.

### Scenario C – Video streaming

Although the system primarily targeted low-rate photo-based monitoring, an interesting improvement is the ability to handle real-time video streaming. For that matter, having one

Table 6.21: Mean and standard deviation values for the experiment with multiple registration requests.

Client	$\Delta t_{2D3D}$ (ms)		FPS	
	Mean	$\sigma$	Mean	$\sigma$
A	291.80	37.95	59.86	2.14
B	164.50	112.24	37.29	4.74
C	412.38	177.53	26.41	6.19
<b>All</b>	245.67	152.67	41.36	14.61

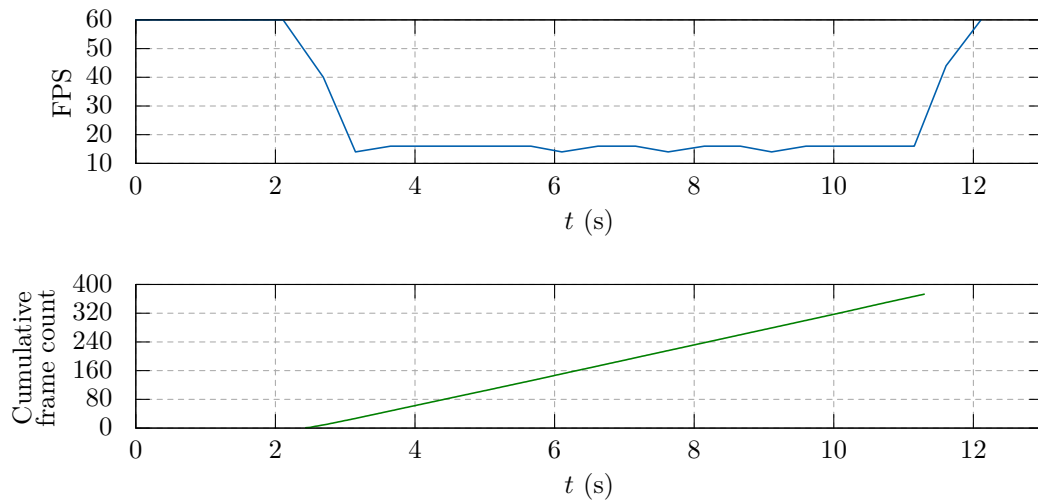
separate HTTP request for each frame of the video is not an interesting solution, since the time spent on network handshaking would compromise the maximum frame rate allowed. Thus, another endpoint was implemented exclusively for testing for this feature.

The video was emulated by using an additional dataset containing 373 images referring to the pose C3. The server application sends the video to the client encoded in Motion JPEG (APPLECOMPUTER, INC, 2001), a format commonly used by IP Cameras. This technique uses the special HTTP multipart requests, which keep the connection active and sends frames delimited by a given boundary. All the images were previously cached in memory, by the server, so that it works as if it had access to a very fast capture device: the next frame is available simply by indexing a long buffer.

Upon retrieval, the contents of each frame replace the texture in the overlay plane. As explained throughout this text, no real-time pose estimation is needed for fixed cameras.

This test was run in Client A. Initially, we tested the scenario of the server streaming the content without any frame rate limit. Figure 6.19 shows the frames at the time they were received by the client, as well as the collected FPS.

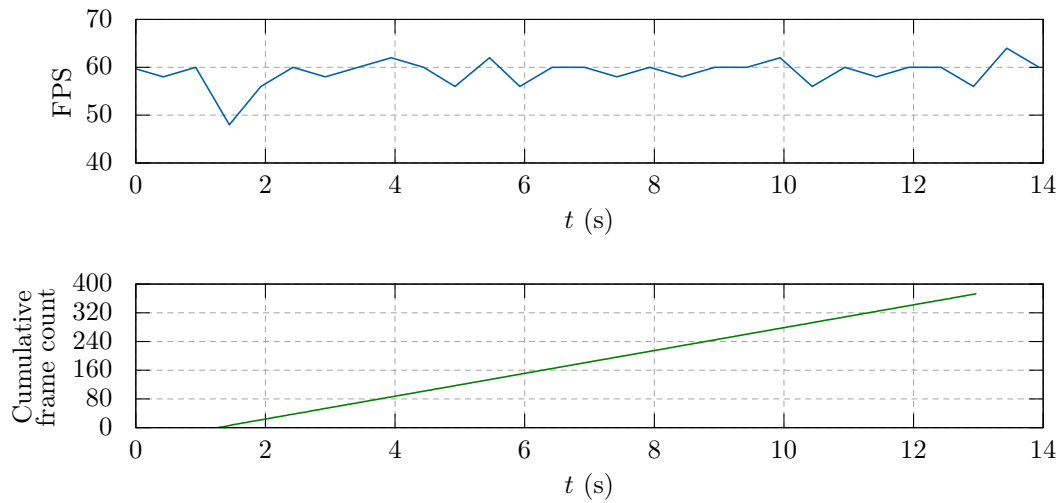
Figure 6.19: Video streaming – no delay between frames.



Source: The author.

Although a reasonably high frame rate was obtained for the video (373 frames in 8.86 seconds, i.e., 42.1 FPS), it compromised the VR engine's real-time renderer. Thus, a second test was run, this time adding a small delay, on the server-side, between the frames ( $\Delta t = 25$  milliseconds). The new behaviour is depicted in Figure 6.20.

Figure 6.20: Video streaming – small delay between frames.



Source: The author.

With this new approach, the frame rates for the video alone and the game engine's rendering became more balanced. The video was streamed at 31.92 FPS, whereas the application was rendered at 58.69 FPS.

## 6.6 Results Overview

All the registration quality metrics values extracted from the prototypes are organized in Table 6.22. For the *avcemig* solution, the overall values (all keypoints from all cameras at once) are used.

Although the collected values are reasonably small, it should be noted that the substation's virtual model used in the *avcamera* and *avcemig* solutions has not the exact dimensions of real equipment, cables, and porticos. Such objects were modeled with incomplete computer-aided design (CAD) data instead of more precise methods such as 3D scanning. This limitation directly affects the registration quality. Thus, the quantitative metrics are focused on the keypoints and the reprojection errors. Still, the geometrical model of the power disconnecter is arguably good, considering the low errors obtained from both the Perspective-n-Point solver and the 2D–3D registration routines.

Table 6.22: Results for the first prototypes.

	avmath	avloopback Scenario 1	avloopback Scenario 2	avcamera	avcemig
$\overline{d_{PNP}}$	2.141	7.570	2.738	7.067	7.00
$\overline{e_{x\%}}$	0.159 %	0.304 %	0.0698 %	0.585 %	0.490 %
$\overline{e_{y\%}}$	0.487 %	0.369 %	0.201 %	0.737 %	0.603 %
$d_{CAMS}$		2.9871	0.24775	N/A	N/A
$\Delta R$		$1.067 \times 10^{-4}$	$1.103 \times 10^{-5}$	N/A	N/A
$\overline{d_{AV}}$		70.86	6.80	10.66	7.48
$\overline{\delta_{x\%}}$		3.21 %	0.104 %	0.340 %	0.231 %
$\overline{\delta_{y\%}}$		2.42 %	0.588 %	0.629 %	0.451 %

Regarding the metrics related to performance, Table 6.23 summarizes the collected values.

Table 6.23: Performance metrics values.

<b>Multi-client</b>		
Metric	Mean	$\sigma$
$\Delta t_{calib}$ (ms)	459.19	154.01
$\Delta t_{2D3D}$ (ms)	245.67	152.67
Renderer FPS	41.36	14.61
<b>Video Streaming</b>		
Metric	Mean	
Renderer FPS	58.69	
Video FPS	31.92	

Although the FPS mean and standard value for the multi-client scenario is included in the table, for completeness, it should be noted that, as expected, its value depends mainly on the machine hardware, in particular the video card.

Finally, for the user interface and the SCADA integration, some qualitative metrics are still needed. This is a planned action once the system is deployed.

The next chapter presents the final considerations and future work, considering these results and the development directions.

## Conclusion

From the results presented in the last chapter, we can conclude that it is possible to perform 2D–3D registration of real-time field images into power substations virtual environments with satisfactory levels of quality, even without applying *in loco* calibration algorithms. Instead of this expensive method, we can use the ideal camera matrix and adjust the elements related to the focal length according to an optimization algorithm, which evaluates the quality metric for many possible configurations.

One immediate issue, however, is the extraction of keypoints coordinates. While the virtual environment can easily store this metadata within the virtual models, the same is not true for the field images: PTZ cameras may vary their preset positions and orientations due to servomotors precision issues, and mobile cameras cannot be used if these keypoints coordinates must be manually added every time.

Hence, without computer-vision methods to extract the keypoints, the solution is limited to fixed inspection cameras, although some mobility can be handled with the aid of correction mechanisms such as the one proposed by Nassu et al. (2018). With automatic keypoints detection available, one interesting approach would be to run the camera calibration routine periodically, according to some fixed or dynamic rate. In this case, the calibration would run on the client-side, for best performance, and the server would stream video instead of images too distant in time. Also, the metric  $\overline{d_{AV}}$ , computed in real-time, could guide the decision process of running the calibration again.

Indeed, the ability to check the final registration quality, given by  $\overline{d_{AV}}$ , after minimizing reprojection errors, i.e., minimizing  $\overline{d_{PNP}}$ , is of great value, since the intrinsic parameters of virtual and real cameras may differ, as well as their point of view. Our solution uses the  $\overline{d_{AV}}$  metric to decide whether the field image should be rendered, according to some error threshold. This is an interesting feature for remote inspection, since bad registration due to a misplaced image overlay may lead to wrong interpretations from the user. Alternatively, some new methods like the one proposed by Brown, Windridge, and Guillemaut (2019) can be evaluated for this application, resulting in a keypoint-free approach.

Since we have evaluated the case of fixed inspection cameras, the client-server architecture has the benefit of caching the estimated camera poses, computed during the calibration requests, so that only images are transferred for performing 2D–3D registration. Besides, this architecture allows the deployment of new inspection cameras without having to recompile the software components on the clients’ workstations. Although the image keypoints must be informed manually, the operation is done only once per camera, upon insertion on the database. Multiple servers might be set up to leverage the load or to separate substations into groups.

We have defined a convention for establishing keypoints in a particular type of power disconnector. Other devices may be used, as long as such conventions are defined for them and their world space keypoints coordinates are stored either within the virtual environment application, as was the case in our solution, or into the image registration database.

Finally, two papers were published during this research work: one in a journal (MATTIOLI; CARDOSO; LAMOUNIER, 2020) and another in a conference (MATTIOLI; CARDOSO; LAMOUNIER; SOUSA RAMOS, 2019).

## **7.1 Ongoing and future work**

This section describes some features currently being added to the system, as well as some other ideas for future work.

### **7.1.1 Automatic image keypoints extraction**

Although we can manually specify the image keypoints coordinates for fixed cameras, assuming that they will not vary with time, this certainly is not the case for mobile cameras. Thus, the application of computer vision routines for extracting such coordinates is of great importance for extending the scope of the solution. For example, a camera zoom motion might imply in moving the virtual environment accordingly. Since, in this case, the navigation would occur in through the physical counterpart representation, the result could arguably be considered an Augmented Reality solution.

### **7.1.2 Sepia effect**

We are currently exploring the idea of applying a sepia effect to indicate old images, according to a user-defined setting. This is particularly useful to avoid wrong conclusions based on outdated information.

### 7.1.3 Registration-oriented modeling

During our experiments with field images, we have observed that our virtual models' geometry have some irregularities, mainly related to components modeled without precise CAD data. For instance, we have cases of thermal images having very low discrepancies among keypoints, but visibly with a poor matching for porticos. This fact has lead to the idea of using 2D–3D registration to validate and improve the geometry of virtual models. In such a scenario, we would consider a model having high confidence and low confidence parts. The former would be used to define keypoints, whereas the latter would be adjusted according to registration results. This is an interesting field of research, also proposed as future work.

### 7.1.4 Foreground-background segmentation

Another technique related to registration is the foreground-background segmentation (SEVAK et al., 2017; GELASCA; EBRAHIMI, 2009), described in Chapter 3. Applied to the case of power disconnecter images, this may filter the information presented to the user, so that only the relevant parts of the image are rendered, and useless objects such as clouds are removed. However, it should be noted that, for some occurrences, such as an explosion, the complete image is still important.

## 7.2 Final considerations

Considering the growing need of monitoring remote installations, and the concept of cyber-physical systems, 2D–3D spatial registration can be a valuable technique to aid in decision-making processes as well as cognitive-related issues.

We have evaluated the particular case of power disconnectors in substations, and a method was developed to put field images on the corresponding virtual environments and to dynamically evaluate the registration quality, without the need of traveling to the remote sites to perform calibration and without knowing the exact camera specifications.

The solution is adequate for photo-based monitoring, but can also be used with real-time video, depending on the desired frame rate for both the rendered environment and the field images. Moreover, an alarm system was implemented, to emit alerts under the condition of inconsistent equipment states reported by different sources (image and meters).

Finally, taking into account the trends of the oncoming Industrial Internet of Things (IIoT), as discussed by Boyes et al. (2018), an immediate implication is that most automatic actions won't be performed only with non-contextual, restricted-scope sensorial information. Each component can request other components' states to have a bigger background before acting. Similarly, substation dispatchers, either human or computer, may benefit from analyzing the surroundings of an occurrence.

In fact, the solution described in this work is aligned with the concept of cyber-physical systems (CPS), one of the main features of Industry 4.0 (VAIDYA; AMBAD; BHOSLE, 2018), providing seamless integration of remote inspection data with other sources of information, such as the SCADA database, the substation's physical layout and the equipment topology.

## Web API Specification

### A.1 Requests for cameras and images

#### GET /cameras/query/<substation>

*retrieve the list of cameras for a given substation*

##### Parameters

substation (string) substation name

##### Response

*application/xml*

200 OK

**TAG cameras**

**TAG camera-group** ..... zero or more camera groups

**ATTR asset** (string) ..... asset name

**TAG camera** ..... zero or more cameras

**ATTR mode** (string) ..... camera mode (e.g.: thermal, RGB)

Attributes for already calibrated cameras:

**ATTR focal-length-scale** (opcional, float) ..... focal length scale factor

**ATTR dav** (opcional, float) ..... quality metric (mean error value)

**ATTR plane-pos, plane-rot** (optional, float[]) ..... overlay plane pose

**ATTR unity-camera-pos** (opcional, float[]) ..... virtual camera position [x,y,z]

**TAG keypoints**

**TAG keypoint** ..... zero or more keypoints

**ATTR name** (string) ..... keypoint name (e.g.: Keypoint12, Isolator1)

**ATTR x, y** (int) ..... image coordinates

**ATTR xn, yn** (float) .... normalized image coordinates (between 0 and 1)

**GET /cameras/<id>/last\_image***query the last available image for certain camera***Parameters**

id (int) camera id

**Response***application/xml***200** OK**TAG** photo**ATTR** creation-time (long) ..... UNIX timestamp (epoch)**ATTR** dataset-path (string) ..... image relative path on dataset or server**ATTR** width, height (int) ..... image dimensions, in pixels**TAG** pngdata ..... PNG image encoded in Base 64**GET /cameras/<id>/image?start=<t1>&end***get the first historical image in a time interval***Parameters**

id (int) camera id

t1 (long) UNIX timestamp for the lower limit

t2 (long) UNIX timestamp for the upper limit

**Response***application/xml***200** OK**TAG** photo

(Same structure as the /cameras/&lt;id&gt;/last\_image endpoint.)

## A.2 Request for video streaming

**GET /cameras/<id>/video-feed.mjpg***get real-time video from some camera***Parameters**

id (int) camera id

**Response***multipart/x-mixed-replace***200** OK

The video frames encoded as JPG (Motion JPEG).

## A.3 Requests for camera calibration

### POST /cameras/<id>/calibrate

*server-side calibration algorithm*

#### Parameters

id (int) camera identifier

#### Request

*application/xml*

**calibration-request**

**camera-id** (int) ..... camera identifier

**keypoints**

**keypoint** ..... one or more keypoints

**name** (string) ..... keypoint name or code

**wcs** (float[]) ..... keypoint coordinates in  $\mathbb{R}^3 [x,y,z]$

**focal-length-scale**

**ternary-search** ..... other methods may be implemented in the future

**min** (float) ..... minimum value for  $k$

**max** (float) ..... maximum value for  $k$

**iterations** (int) ..... number of iterations

#### Response

*application/xml*

**calibration-response**

**solutions**

**solution** ..... one or more solutions

**k** (float) ..... focal length scale factor

**pyramid**

**apex** (float[]) ..... pyramid apex in WCS

**axis** (float[]) ..... pyramid axis in WCS

**line** ..... projection lines (pyramid edges and central line)

**name** (string) ..... line name (l00, lw0, lwh, lc, ...)

**coeffs** (float[]) ..... line equations coefficients

**metrics**

**metric** ..... one or more server-side quality metrics

**name** (string) ..... metric name (e.g.: dpnp)

**value** (float[]) ..... metric value

POST /cameras/<id>/set-params		
save camera parameters (calibration and spatial registration)		
Parameters		
id	(int)	camera identifier
Request		application/xml
TAG camera-params		
ATTR focal-length-scale (float) ..... focal length scale factor		
ATTR dav (float) ..... registration quality metric		
ATTR plane-pos (float[]) ..... overlay plane position [x,y,z]		
ATTR plane-rot (float[]) ..... overlay plane rotation [x,y,z,w]		
ATTR unity-camera-pos (float[]) ..... virtual camera position [x,y,z]		
Response		text/plain
200	OK	
OK		
404	Not Found	
Camera not found.		

## Publications

During doctoral studies, the following papers were published:

MATTIOLI, Leandro; CARDOSO, Alexandre; LAMOUNIER, Edgard. 2D–3D Spatial Registration for Remote Inspection of Power Substations. **Energies**, v. 13, n. 23, 2020. ISSN 1996-1073. DOI: 10.3390/en13236209. Available from: <<https://www.mdpi.com/1996-1073/13/23/6209>>

MATTIOLI, Leandro Resende; CARDOSO, Alexandre; LAMOUNIER, Edgard; SOUSA RAMOS, Daniel de. Inspeção de Equipamentos de Subestações de Energia Elétrica por Videomonitoramento e Virtualidade Aumentada. In: XVIII ENCONTRO REGIONAL IBERO-AMERICANO DO CIGRÉ - ERIAC. **Anais do XVIII Encontro Regional Ibero-americano do Cigré - ERIAC**. ed. by Editora do CIGRE INTERNACIONAL. Foz do Iguaçu, PR, Brazil: Editora do CIGRE Internacional, 2019. v. 1

# References

- ANTONIJEVIĆ, Miro; SUČIĆ, Stjepan; KESERICA, Hrvoje. Augmented Reality Applications for Substation Management by Utilizing Standards-Compliant SCADA Communication. **Energies**, MDPI AG, v. 11, n. 3, p. 599, Mar. 2018. DOI: 10.3390/en11030599.
- APPLECOMPUTER, INC (Ed.). **QuickTime File Format**. [S.l.: s.n.], 1 Mar. 2001.
- AUTODESK. **3D Studio Max**. [S.l.: s.n.]. Available from: <<https://www.autodesk.com.br/products/3ds-max/overview>>.
- AVEVA. **Wonderware InTouch**. [S.l.: s.n.]. Available from: <<https://www.wonderware.com/pt-br/hmi-scada/intouch/>>. Visited on: 15 Jan. 2020.
- BAILEY, David; WRIGHT, Edwin. **Practical SCADA for industry**. [S.l.]: Elsevier, 2003.
- BASTOS, M. R.; MACHADO, M. S. L. Visual, real-time monitoring system for remote operation of electrical substations. In: 2010 IEEE/PES Transmission and Distribution Conference and Exposition: Latin America (T D-LA). [S.l.: s.n.], Nov. 2010. p. 417–421. DOI: 10.1109/TDC-LA.2010.5762915.
- BOYES, Hugh et al. The industrial internet of things (IIoT): An analysis framework. **Computers in Industry**, Elsevier BV, v. 101, p. 1–12, Oct. 2018. DOI: 10.1016/j.compind.2018.04.015.
- BRADSKI, G. The OpenCV Library. **Dr. Dobb's Journal of Software Tools**, 2000.
- BROWN, Mark; WINDRIDGE, David; GUILLEMAUT, Jean-Yves. A family of globally optimal branch-and-bound algorithms for 2D–3D correspondence-free registration. **Pattern Recognition**, v. 93, p. 36–54, 2019. ISSN 0031-3203. DOI: 10.1016/j.patcog.2019.04.002. Available from: <<http://www.sciencedirect.com/science/article/pii/S0031320319301426>>.
- BUHAGIAR, T. et al. Poste intelligent - The next generation smart substation for the French power grid. In: 13TH International Conference on Development in Power System Protection 2016 (DPSP). [S.l.: s.n.], Mar. 2016. p. 1–4. DOI: 10.1049/cp.2016.0007.

- BUI, G. et al. LIDAR-based virtual environment study for disaster response scenarios. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). [S.l.: s.n.], May 2015. p. 790–793. DOI: 10.1109/INM.2015.7140377.
- CAI, D. et al. A practical preset position calibration technique for unattended smart substation security improvement. In: 2017 IEEE Power Energy Society General Meeting. [S.l.: s.n.], July 2017. p. 1–5. DOI: 10.1109/PESGM.2017.8274375.
- CARDOSO, Alexandre et al. VRCEMIG: a novel approach to power substation control. In: ACM SIGGRAPH 2016 Posters. [S.l.]: Association for Computing Machinery, 2016. 3:1–3:2. DOI: 10.1145/2945078.2945081.
- CATER-STEEL, Aileen; TOLEMAN, Mark; RAJAEIAN, Mohammad Mehdi. Design Science Research in Doctoral Projects: An Analysis of Australian Theses. **Journal of the Association for Information Systems**, Association for Information Systems, p. 1844–1869, 2019. DOI: 10.17705/1jais.00587.
- CÉSAR, Vinícius Miranda et al. Avaliação de Algoritmos de Estimativa de Pose para Reconstrução 3D e Realidade Aumentada. In: VIII Workshop de Realidade Virtual e Aumentada - WRVA 2011. Uberaba, Brazil: [s.n.], 2011.
- CHANGFU, X.; BIN, B.; FENGBO, T. Research of Substation Equipment Abnormity Identification Based on Image Processing. In: 2017 International Conference on Smart Grid and Electrical Automation (ICSGEA). [S.l.: s.n.], May 2017. p. 411–415. DOI: 10.1109/ICSGEA.2017.54.
- CHHEANG, V. et al. Natural embedding of live actors and entities into 360° virtual reality scenes. English. **Journal of Supercomputing**, Springer New York LLC, 2018. cited By 0; Article in Press. ISSN 0920-8542. DOI: 10.1007/s11227-018-2615-z. Available from: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85053789504%5C&doi=10.1007%2fs11227-018-2615-z%5C&partnerID=40%5C&md5=bd0121e77a600fe65760dbc93c123240>>.
- DEMENTHON, Daniel F.; DAVIS, Larry S. Model-based Object Pose in 25 Lines of Code. **Int. J. Comput. Vision**, Kluwer Academic Publishers, Hingham, MA, USA, v. 15, n. 1-2, p. 123–141, June 1995. ISSN 0920-5691. DOI: 10.1007/BF01450852.
- FISCHLER, Martin A.; BOLLES, Robert C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. **Commun. ACM**, ACM, New York, NY, USA, v. 24, n. 6, p. 381–395, June 1981. ISSN 0001-0782. DOI: 10.1145/358669.358692.
- FÜHR, G.; JUNG, C. R. Camera Self-Calibration Based on Nonlinear Optimization and Applications in Surveillance Systems. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 27, n. 5, p. 1132–1142, May 2017. ISSN 1051-8215. DOI: 10.1109/TCSVT.2015.2511812.

- GELASCA, E. D.; EBRAHIMI, T. On Evaluating Video Object Segmentation Quality: A Perceptually Driven Objective Metric. **IEEE Journal of Selected Topics in Signal Processing**, v. 3, n. 2, p. 319–335, Apr. 2009. ISSN 1932-4553. DOI: 10.1109/JSTSP.2009.2015067.
- GRINBERG, Miguel. **Flask Web Development: Developing Web Applications with Python**. [S.l.]: O'Reilly Media, Inc., 2018.
- GUTIÉRREZ-MARROQUÍN, W.; JIMÉNEZ, C. H.; FERNÁNDEZ-FERNÁNDEZ, M. Web access for flexible manufacturing system. In: 2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON). [S.l.: s.n.], Oct. 2017. p. 1–6. DOI: 10.1109/CHILECON.2017.8229618.
- HARTLEY, Richard. **Multiple view geometry in computer vision**. Cambridge, UK New York: Cambridge University Press, 2004. ISBN 978-0-521-54051-3.
- HIPP, Dwayne Richard; KENNEDY, D.; MISTACHKIN, J. **SQLite**. Ed. by SQLite Development Team. [S.l.: s.n.], 2020. <https://www.sqlite.org/>. Available from: <<https://www.sqlite.org/>>. Visited on: 15 Jan. 2020.
- HONGKAI, Chen et al. Computer vision-based detection and state recognition for disconnecting switch in Substation automation. **International Journal of Robotics and Automation**, v. 32, Feb. 2017. DOI: 10.2316/Journal.206.2017.1.206-4624.
- HU, Yu; WU, Wei; ZHOU, Zhong. Video Driven Pedestrian Visualization with Characteristic Appearances. In: PROCEEDINGS of the 21st ACM Symposium on Virtual Reality Software and Technology. Beijing, China: ACM, 2015. (VRST '15), p. 183–186. ISBN 978-1-4503-3990-2. DOI: 10.1145/2821592.2821614.
- HUYNH, Du Q. Metrics for 3D Rotations: Comparison and Analysis. **Journal of Mathematical Imaging and Vision**, Springer Science and Business Media LLC, v. 35, n. 2, p. 155–164, June 2009. DOI: 10.1007/s10851-009-0161-2.
- ILLINGWORTH, J.; KITTLER, J. The Adaptive Hough Transform. **IEEE Trans. Pattern Anal. Mach. Intell.**, IEEE Computer Society, Washington, DC, USA, v. 9, n. 5, p. 690–698, May 1987. ISSN 0162-8828. DOI: 10.1109/TPAMI.1987.4767964. Available from: <<https://doi.org/10.1109/TPAMI.1987.4767964>>.
- ISO. **Information technology — Computer graphics, image processing and environmental data representation — Mixed and augmented reality (MAR) reference model**. v. 2019. [S.l.], 2019.
- IWATAKI, S. et al. Visualization of the surrounding environment and operational part in a 3DCG model for the teleoperation of construction machines. English. In: p. 81–87. cited By 6. ISBN 9781467372428. DOI: 10.1109/SII.2015.7404958. Available from: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0->

84963717989%5C&doi=10.1109%2fSII.2015.7404958%5C&partnerID=40%5C&md5=2221f4e796079d98c066dc6d805f69f9>.

JIANG, Q. et al. A Contour Angle Orientation for Power Equipment Infrared and Visible Image Registration. **IEEE Transactions on Power Delivery**, p. 1–1, 2020. ISSN 1937-4208. DOI: 10.1109/TPWRD.2020.3011962.

KAEHLER, Adrian; BRADSKI, Gary. **Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library**. 1st. [S.l.]: O'Reilly Media, Inc., 2016. ISBN 1491937998.

KARAKOTTAS, A. et al. Augmented VR. In: 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). [S.l.: s.n.], Mar. 2018. p. 1–1. DOI: 10.1109/VR.2018.8446561.

KIM, Chulyeon et al. A hybrid framework combining background subtraction and deep neural networks for rapid person detection. **Journal of Big Data**, Springer, v. 5, n. 1, p. 1, 10 July 2018. ISSN 2196-1115. DOI: 10.1186/s40537-018-0131-x.

LABONTE, D.; BOISSY, P.; MICHAUD, F. Comparative Analysis of 3-D Robot Teleoperation Interfaces With Novice Users. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, v. 40, n. 5, p. 1331–1342, Oct. 2010. ISSN 1083-4419. DOI: 10.1109/TSMCB.2009.2038357.

LEPETIT, Vincent; MORENO-NOGUER, Francesc; FUA, Pascal. EPnP: An Accurate O(n) Solution to the PnP Problem. **International Journal Of Computer Vision**, Springer Verlag, v. 81, p. 155–166, 2009. DOI: 10.1007/s11263-008-0152-6.

LI, Haitao; ZHANG, Yongting; LIANG, Guojian. Application of Foreground Detection Technology in Intelligent Video Monitoring System of Substation. In: PROCEEDINGS of the 2Nd International Conference on Computer Science and Application Engineering. Hohhot, China: ACM, 2018. (CSAE '18), 111:1–111:5. ISBN 978-1-4503-6512-3. DOI: 10.1145/3207677.3278013.

LIANG, Haiyang et al. Computer Vision based Automatic Power Equipment Condition Monitoring and Maintenance: A Brief Review. In: 2020 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES). [S.l.]: IEEE, Oct. 2020. DOI: 10.1109/dcabes50732.2020.00045.

LIU, Y. et al. Fire Detection in Radiant Energy Domain for Video Surveillance. In: 2015 International Conference on Virtual Reality and Visualization (ICVRV). [S.l.: s.n.], Oct. 2015. p. 1–8. DOI: 10.1109/ICVRV.2015.54.

LU, S.; ZHANG, Y.; SU, J. Mobile robot for power substation inspection: a survey. **IEEE/CAA Journal of Automatica Sinica**, p. 1–18, 2018. ISSN 2329-9266. DOI: 10.1109/JAS.2017.7510364.

- LUO, Y.; DAI, J.; QI, L. Fault-Tolerant Video Analysis Cloud Scheduling Mechanism. In: 2013 International Conference on Virtual Reality and Visualization. [S.l.: s.n.], Sept. 2013. p. 119–126. DOI: 10.1109/ICVRV.2013.27.
- LUO, Yi; TU, Guangyu. Who's watching the unattended substation [substation television system]. **IEEE Power and Energy Magazine**, v. 3, n. 1, p. 59–66, Jan. 2005. ISSN 1540-7977. DOI: 10.1109/MPAE.2005.1380236.
- LV, L. et al. An approach for fault monitoring of insulators based on image tracking. In: 2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP). [S.l.: s.n.], Nov. 2017. p. 1–6. DOI: 10.1109/M2VIP.2017.8211434.
- MATTIOLI, Leandro; CARDOSO, Alexandre; LAMOUNIER, Edgard. 2D–3D Spatial Registration for Remote Inspection of Power Substations. **Energies**, v. 13, n. 23, 2020. ISSN 1996-1073. DOI: 10.3390/en13236209. Available from: <https://www.mdpi.com/1996-1073/13/23/6209>.
- MATTIOLI, Leandro Resende; CARDOSO, Alexandre; LAMOUNIER, Edgard; SOUSA RAMOS, Daniel de. Inspeção de Equipamentos de Subestações de Energia Elétrica por Videomonitoramento e Virtualidade Aumentada. In: XVIII ENCONTRO REGIONAL IBERO-AMERICANO DO CIGRÉ - ERIAC. **Anais do XVIII Encontro Regional Ibero-americano do Cigré - ERIAC**. Ed. by Editora do Cigre Internacional. Foz do Iguaçu, PR, Brazil: Editora do CIGRE Internacional, 2019. v. 1.
- MILGRAM, Paul; COLQUHOUN, Herman. A Taxonomy of Real and Virtual World Display Integration, Jan. 2001. DOI: 10.1007/978-3-642-87512-0\_1.
- NAĐ, Đ.; MIŠKOVIĆ, N.; OMERDİC, E. Multi-Modal Supervision Interface Concept for Marine Systems. In: OCEANS 2019 - Marseille. [S.l.: s.n.], June 2019. p. 1–5. DOI: 10.1109/OCEANSE.2019.8867226.
- NAHON, D.; SUBILEAU, G.; CAPEL, B. “Never Blind VR” enhancing the virtual reality headset experience with augmented virtuality. In: 2015 IEEE Virtual Reality (VR). [S.l.: s.n.], Mar. 2015. p. 347–348. DOI: 10.1109/VR.2015.7223438.
- NASSU, B. T. et al. Image-Based State Recognition for Disconnect Switches in Electric Power Distribution Substations. In: 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). [S.l.: s.n.], Oct. 2018. p. 432–439. DOI: 10.1109/SIBGRAPI.2018.00062.
- PAL, D. et al. Real-time condition monitoring of substation equipment using thermal cameras. **IET Generation, Transmission Distribution**, v. 12, n. 4, p. 895–902, 2018. DOI: 10.1049/iet-gtd.2017.0096.

- PEREIRA, R. F. L. E. C. et al. Sistema para Videomonitoramento Operacional de Chaves Seccionadoras de Subestações de Energia [A System for Operational Videomonitoring of Power Substations Disconnecter Switches]. Portuguese. In: 21 Nov.–232016. 14TH Encontro para Debates de Assuntos de Operação (XIV EDAO). São Paulo, BrazilSão Paulo, Brazil: [s.n.], Nov. 2016.
- POHJOLAINEN, Seppo; HEILIÖ, Matti, et al. **Mathematical modelling**. [S.l.]: Springer, 2016.
- POSTOLKA, Barbara et al. Evaluation of an intensity-based algorithm for 2D/3D registration of natural knee videofluoroscopy data. **Medical Engineering & Physics**, v. 77, p. 107–113, 2020. ISSN 1350-4533. DOI: 10 . 1016/j . medengphy . 2020 . 01 . 002. Available from: <<http://www.sciencedirect.com/science/article/pii/S1350453320300047>>.
- REGENBRECHT, H. et al. An augmented virtuality approach to 3D videoconferencing. In: THE Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings. [S.l.: s.n.], Oct. 2003. p. 290–291. DOI: 10 . 1109/ISMAR . 2003 . 1240725.
- SARABANDI, Soheil; THOMAS, Federico. Accurate Computation of Quaternions from Rotation Matrices. In: \_\_\_\_\_. **Advances in Robot Kinematics 2018**. Cham: Springer International Publishing, 2019. p. 39–46. ISBN 978-3-319-93188-3.
- SEBE, Ismail Oner et al. 3D Video Surveillance with Augmented Virtual Environments. In: FIRST ACM SIGMM International Workshop on Video Surveillance. Berkeley, California: ACM, 2003. (IWVS '03), p. 107–112. ISBN 1-58113-780-X. DOI: 10 . 1145/982452 . 982466.
- SEVAK, J. S. et al. Survey on semantic image segmentation techniques. In: 2017 International Conference on Intelligent Sustainable Systems (ICISS). [S.l.: s.n.], Dec. 2017. p. 306–313. DOI: 10 . 1109/ISS1 . 2017 . 8389420.
- SHI, Lifeng et al. Video-Based Fire Detection with Saliency Detection and Convolutional Neural Networks. **Advances in Neural Networks - ISNN 2017**, Springer, 1 Jan. 2017. DOI: 10 . 1007/978-3-319-59081-3\_36.
- SUN, Guobing; QIU, Yongsheng; SUI, Shengchun. Research on Optimization Method for Enhancing Night Surveillance Image Based on Fusion of Infrared and Visible Light Image. In: PROCEEDINGS of the 2020 12th International Conference on Computer and Automation Engineering. Sydney, NSW, Australia: Association for Computing Machinery, 2020. (ICCAE 2020), p. 98–102. ISBN 9781450376785. DOI: 10 . 1145/3384613 . 3384644.
- UNITY TECHNOLOGIES. **Unity 3D**. [S.l.: s.n.]. <https://unity.com/>. Available from: <<https://www.unity.com/>>. Visited on: 15 Jan. 2020.
- VAGVOLGYI, B. et al. Augmented virtuality for model-based teleoperation. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). [S.l.: s.n.], Sept. 2017. p. 3826–3833. DOI: 10 . 1109/IROS . 2017 . 8206233.

- VAIDYA, Saurabh; AMBAD, Prashant; BHOSLE, Santosh. Industry 4.0 – A Glimpse. **Procedia Manufacturing**, v. 20, p. 233–238, 2018. ISSN 2351-9789. DOI: 10.1016/j.promfg.2018.02.034. Available from: <<http://www.sciencedirect.com/science/article/pii/S2351978918300672>>.
- VENKATAKRISHNAN, R.; VENKATAKRISHNAN, R.; ANARAKY, R. G., et al. A Structural Equation Modeling Approach to Understand the Relationship between Control, Cybersickness and Presence in Virtual Reality. In: 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). [S.l.: s.n.], 2020. p. 682–691. DOI: 10.1109/VR46266.2020.00091.
- VENKATAKRISHNAN, R.; VENKATAKRISHNAN, R.; BHARGAVA, A., et al. Comparative Evaluation of the Effects of Motion Control on Cybersickness in Immersive Virtual Environments. In: 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). [S.l.: s.n.], 2020. p. 672–681. DOI: 10.1109/VR46266.2020.00090.
- VINCE, John. **Mathematics for computer graphics**. London: Springer, 2006. ISBN 978-1-84628-034-4.
- WANG, T.; AN, Q., et al. Vision-based illegal human ladder climbing action recognition in substation. In: 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI). [S.l.: s.n.], Feb. 2017. p. 189–194. DOI: 10.1109/ICACI.2017.7974507.
- WANG, X.; CHEN, R. An empirical study on augmented virtuality space for tele-inspection inspection of built environments. **Tsinghua Science and Technology**, v. 13, S1, p. 286–291, Oct. 2008. DOI: 10.1016/S1007-0214(08)70163-9.
- WU, Y. et al. Real-Time 3D Road Scene Based on Virtual-Real Fusion Method. **IEEE Sensors Journal**, v. 15, n. 2, p. 750–756, Feb. 2015. ISSN 1530-437X. DOI: 10.1109/JSEN.2014.2354331.
- XIAO-LE, H. et al. 500 kV substation robot patrol system. In: 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC). [S.l.: s.n.], Oct. 2017. p. 105–109. DOI: 10.1109/ITOEC.2017.8122390.
- XIAOMING, S.; SHAOSHENG, F.; BING, Y. Implementation of infrared measuring temperature on remote image monitoring and control system in transformer substation. In: 2012 International Conference on Image Analysis and Signal Processing. [S.l.: s.n.], Nov. 2012. p. 1–4. DOI: 10.1109/IASP.2012.6425048.
- XIE, J. et al. Automatic Path Planning for Augmented Virtual Environment. In: 2016 International Conference on Virtual Reality and Visualization (ICVRV). [S.l.: s.n.], Sept. 2016. p. 372–379. DOI: 10.1109/ICVRV.2016.69.
- ZHENG, J. et al. Accelerated RANSAC for Accurate Image Registration in Aerial Video Surveillance. **IEEE Access**, p. 1–1, 2021. ISSN 2169-3536. DOI: 10.1109/ACCESS.2021.3061818.