FEATURE POINT CLASSIFICATION AND MATCHING

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Avşar Polat Ay

August 2007

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Levent Onural(Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. A. Aydın Alatan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Pınar Duygulu Şahin

Approved for the Institute of Engineering and Sciences:

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

FEATURE POINT CLASSIFICATION AND MATCHING

Avşar Polat Ay
M.S. in Electrical and Electronics Engineering
Supervisor: Prof. Dr. Levent Onural
August 2007

A feature point is a salient point which can be separated from its neighborhood. Widely used definitions assume that feature points are corners. However, some non-feature points also satisfy this assumption. Hence, non-feature points, which are highly undesired, are usually detected as feature points. Texture properties around detected points can be used to eliminate non-feature points by determining the distinctiveness of the detected points within their neighborhoods. There are many texture description methods, such as autoregressive models, Gibbs/Markov random field models, time-frequency transforms, etc. To increase the performance of feature point related applications, two new feature point descriptors are proposed, and used in non-feature point elimination and feature point sorting-matching. To have a computationally feasible descriptor algorithm, a single image resolution scale is selected for analyzing the texture properties around the detected points. To create a scale-space, wavelet decomposition is applied to the given images and neighborhood scale-spaces are formed for every detected point. The analysis scale of a point is selected according to the changes in the kurtosis values of histograms which are extracted from the neighborhood scale-space. By using descriptors, the detected non-feature points are eliminated, feature points are sorted and with inclusion of conventional descriptors feature points are matched. According to the scores obtained in the experiments, the proposed detection-matching scheme performs more reliable than the Harris detector gray-level patch matching scheme. However, SIFT detection-matching scheme performs better than the proposed scheme.

Keywords: Feature Point Elimination, Feature Point Matching, Digital Video Processing, Feature Point Detection, Feature Points.

ÖZET

ÖZNİTELİKLİ NOKTA SINIFLANDIRMASI VE EŞLEMESİ

Avşar Polat Ay
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Prof. Dr. Levent Onural
August 2007

Bir öznitelikli nokta belirgin bir noktadır; etrafındaki diğer noktalardan ayrılabilir. Sıkça kullanılan tanımlarda öznitelikli noktaların köşeler olduğunu varsayılır. Ne yazık ki bazı öznitelikli olmayan noktalar da bu varsayıma uyar. Dolayısıyla, istenmeyen öznitelikli olmayan noktalar da öznitelikli noktalar gibi algılanabilir. Algılanan noktaların etrafındaki doku nitelikleri öznitelikli olmayan noktaların ayıklamasında kullanılabilir. Bunun için noktaların etrafındaki doku özellikleri kullanılarak komşularından ayrılabilirlikleri bulunur. Çok sayıda doku betimleme yöntemi vardır, örneğin, özbağlanım modelleri, Gibbs/Markov rastgele alan modelleri, zaman-sıklık dönüşümleri, vs. Öznitelikli noktalarla ilişkili uygulamaların başarımlarını arttırmak için iki yeni nokta betimleyici önerilmektedir. Bu betimleyiciler öznitelikli olmayan noktaların ayıklanmasında ve öznitelikli nokta sıralanması ile eşlenmesinde kullanılır. Betimleyici algoritmasının bilgiişlem yükünü katlanılabilir bir seviyede tutmak için algılanan noktaların etrafındaki doku özelliklerinin çözümlemesinde tek bir resim çözünürlüğü seçilmiştir. Çözünürlük-uzayı yaratmak için verilen resimlere dalgacık ayrışımı uygulanmıştır ve her nokta için komşuluk çözünürlük-uzayı oluşturulmuştur. Bir noktanın çözümleme çözünürlüğü komşuluk çözünürlük-uzayından çıkartılan histogramların kurtosis değerlerindeki değişimlere göre seçilmektedir. Betimleyici kullanılarak algılanan öznitelikli olmayan noktalar ayıklanmakta, öznitelikli noktalar sıralanmakta ve geleneksel betimleyiciler eklenerek öznitelikli noktalar eşlenmektedir. Deneylerden elde edilen sonuçlara göre, önerilen algılama-eşleme yöntemi Harris algılama ve gri-tonlu bölge betimleme temelli yöntemlerden daha güvenilirdir. Ancak deneylerde SIFT algılama-eşleme yönteminin önerilen yöntemden daha başarılı sonuçlar verdiği gözlenmiştir.

Anahtar Kelimeler: Öznitelikli Nokta Ayıklaması, Öznitelikli Nokta Eşlemesi, Sayısal Video İşleme, Öznitelikli Nokta Algılanması, Öznitelikli Noktalar.

ACKNOWLEDGEMENTS

Contents

1 Introduction 1

1.1 Feature Point Detection Concepts 1

1.2 Problem Statement 7

2 Texture 11

2.1 Texture Definition 11

2.2 Texture Analysis 12

3 Non-Feature Point Elimination And Feature Point Matching 18

3.1 Scale Selection 20

3.2 Texture Description 29

3.3 Feature Point Elimination 34

3.4 Feature Point Matching 35

3.5 Integrating Elimination and Matching Methods to Detectors . . 40

4 Results 42

4.1 Evaluation of the Detector Performances 46

4.2 Evaluation of the Descriptor Performances 50

4.3 Evaluation of the Scale Selection Performances 53

4.4 Comparison of the Harris FPD and the Proposed FPD 56

4.5 Analysis of Detector Performances 77

4.6 Analysis of Descriptor Performances 78

4.7 Analysis of Scale Selection Performances 79

4.8 Analysis of Detection-Matching Performances 79

5 Conclusions 82

A Image Couples Test Results 106

A.1 Detection Results 106

A.2 Scale Selection Results 110

B Video Sequence Results 118

List of Figures

1.1 Selection of the characteristic scale 6

1.2 A corner sample with zoom-in 8

3.1 A scale of a sample image with textures 21

3.2 Another scale of the same image shown in Figure 3.1 21

3.3 Luminance channel of $CIEL^*a^*b^*$ image which is shown in Figure 3.1 23

3.4 Wavelet decomposition of the Luminance channel of $CIEL^*a^*b^*$ image which is shown in Figure 3.1 23

3.5 (a), (b), (c), (d), 31 × 31 patches taken from the image shown in Figure 3.3 from finer scales to coarser scales, respectively, (e) the projection of patches onto the original image. 25

3.6 The histograms of the corresponding patches which are shown in Figure 3.5 26

3.7 Kurtosis values vs scale which are calculated from the histograms of pixel intensities. X-axis shows the scales from 1 (coarsest) to 4 (finest) and Y-axis shows the corresponding kurtosis values. 28

3.8 An illustration of k-means thresholding. 30

3.9 First order neighborhood for the pixel "s". 31

3.10 An illustration of the GMRF parameter estimation method. . . 32

4.1 A set of sample images for each image couple which are used in detector, descriptor and scale selection performance evaluations. 43

4.2 A set of sample frames from the video sequences which are used in comparisons between the Harris feature point detection / gray-level patch description scheme and the proposed feature point detection / description scheme. 44

4.3 A patch from "bike" test image couple. 53

4.4 The detected feature points from the first and second frames of "garden.avi" by using the first scheme. The original garden_sif.yuv sequence is taken from University of California Berkeley Multimedia Research Center. The green and red "+" signs are used for marking the locations of the feature points detected in the first frame and the second frame, respectively. . 61

4.5 The assigned matches for the feature points from the first frame to second frame which are shown in Figure 4.4 by using the first scheme. The green and red "+" signs are used for marking the locations of the feature points detected in the first frame and the second frame, respectively. And the yellow arrows are showing the motion vectors. 62

4.6 A closer look at a set of matched points from the first frame to second frame which are shown in Figure 4.5 by using the first schemes. 63

4.7 The detected feature points from the first and second frames of “garden.avi” by using the second scheme. 64

4.8 The assigned matches for the feature points from the first frame to second frame shown in Figure 4.7 by using the second scheme. 65

4.9 A closer look at a set of matched points from the first frame to second frame which are shown in Figure 4.5 by using the second scheme. 66

4.10 The detected feature points from the first and second frames of “garden.avi” by using the third scheme. 67

4.11 The assigned matches for the feature points from the first frame to second frame which are shown in Figure 4.10 by using the third scheme. 68

4.12 A closer look at a set of matched points from the first frame to second frame which are shown in Figure 4.5 by using the third scheme. 69

4.13 The detection and matching results of “garden.avi” for the first scheme. 70

4.14 The repetability and matching scores of “garden.avi” for the first scheme. 70

4.15 The detection and matching results of “garden.avi” for the second scheme. 71

4.16 The repetability and matching scores of “garden.avi” for the second scheme. 71

4.17 The detection and matching results of “garden.avi” for the third scheme. 72

4.18 The repetability and matching scores of “garden.avi” for the third scheme. 72

4.19 The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for “garden.avi”. 73

4.20 The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for “garden.avi”. 73

A.1 The detected feature points by using the Harris FPD. The original “bike” image couple is taken from Oxford University Visual Geometry Group. 106

A.2 The matched feature points among points detected shown in Figure A.1 by using gray-level patch description. The original “bike” image couple is taken from Oxford University Visual Geometry Group. 107

A.3 The detected feature points by using the proposed FPD. 107

A.4 The matched feature points among points detected shown in Figure A.3 by using gray-level patch description. 108

A.5 The detected feature points by using the SIFT FPD. 108

A.6 The matched feature points among points detected shown in Figure A.5 by using gray-level patch description. 109

A.7 The detected feature points by using the proposed FPD with no scale selection. 110

A.8 The matched feature points among the detected points shown in Figure A.7 by using the proposed matching method. 111

A.9 The detected feature points by using the proposed FPD with the proposed scale selection. 112

A.10 The matched feature points among the detected points shown in Figure A.9 by using the proposed matching method. 113

A.11 The detected feature points by using the proposed FPD with the adjacent finer scale to the proposed scale is selected. 114

A.12 The matched feature points among the detected points shown in Figure A.11 by using the proposed matching method. 115

A.13 The detected feature points by using the proposed FPD with the adjacent coarser scale to the proposed scale is selected. . . . 116

A.14 The matched feature points among the detected points shown in Figure A.13 by using the proposed matching method. 117

B.1 The detection and matching results of "src6_ref_625.avi" for the first scheme. 119

B.2 The repetability and matching scores of "src6_ref_625.avi" for the first scheme. 119

B.3 The detection and matching results of "src6_ref_625.avi" for the second scheme. 120

B.4 The repetability and matching scores of "src6_ref_625.avi" for the second scheme. 120

B.5 The detection and matching results of “src6_ref_625.avi” for the third scheme. 121

B.6 The repetability and matching scores of “src6_ref_625.avi” for the third scheme. 121

B.7 The difference between repetability score of 1^{st} scheme and $2^{nd} or 3^{rd}$ schemes for “src6_ref_625.avi”. 122

B.8 The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for “src6_ref_625.avi”. 122

B.9 The detection and matching results of “src10_ref_625.avi” for the first scheme. 123

B.10 The repetability and matching scores of “src10_ref_625.avi” for the first scheme. 123

B.11 The detection and matching results of “src10_ref_625.avi” for the second scheme. 124

B.12 The repetability and matching scores of “src10_ref_625.avi” for the second scheme. 124

B.13 The detection and matching results of “src10_ref_625.avi” for the third scheme. 125

B.14 The repetability and matching scores of “src10_ref_625.avi” for the third scheme. 125

B.15 The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for “src10_ref_625.avi”. 126

B.16 The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for “src10_ref_625.avi”. 126

B.17 The detection and matching results of “src13_ref_525.avi” for the first scheme. 127

B.18 The repetability and matching scores of “src13_ref_525.avi” for the first scheme. 127

B.19 The detection and matching results of “src13_ref_525.avi” for the second scheme. 128

B.20 The repetability and matching scores of “src13_ref_525.avi” for the second scheme. 128

B.21 The detection and matching results of “src13_ref_525.avi” for the third scheme. 129

B.22 The repetability and matching scores of “src13_ref_525.avi” for the third scheme. 129

B.23 The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for “src13_ref_525”. 130

B.24 The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for “src13_ref_525”. 130

B.25 The detection and matching results of “src19_ref_525.avi” for the first scheme. 131

B.26 The repetability and matching scores of “src19_ref_525.avi” for the first scheme. 131

B.27 The detection and matching results of “src19_ref_525.avi” for the second scheme. 132

B.28 The repetability and matching scores of “src19_ref_525.avi” for the second scheme. 132

B.29 The detection and matching results of "src19_ref_525.avi" for the third scheme. 133

B.30 The repetability and matching scores of "src19_ref_525.avi" for the third scheme. 133

B.31 The difference between repetability score of 1st scheme and 2nd or 3rd schemes for "src19_ref_525". 134

B.32 The difference between match scores of 1st scheme and 2nd and 3rd schemes for "src19_ref_525". 134

B.33 The detection and matching results of "src20_ref_525.avi" for the first scheme. 135

B.34 The repetability and matching scores of "src20_ref_525.avi" for the first scheme. 135

B.35 The detection and matching results of "src20_ref_525.avi" for the second scheme. 136

B.36 The repetability and matching scores of "src20_ref_525.avi" for the second scheme. 136

B.37 The detection and matching results of "src20_ref_525.avi" for the third scheme. 137

B.38 The repetability and matching scores of "src20_ref_525.avi" for the third scheme. 137

B.39 The difference between repetability score of 1st scheme and 2nd or 3rd schemes for "src20_ref_525". 138

B.40 The difference between match scores of 1st scheme and 2nd and 3rd schemes for "src20_ref_525". 138

B.41 The detection and matching results of “src22_ref_525.avi” for the first scheme. 139

B.42 The repetability and matching scores of “src22_ref_525.avi” for the first scheme. 139

B.43 The detection and matching results of “src22_ref_525.avi” for the second scheme. 140

B.44 The repetability and matching scores of “src22_ref_525.avi” for the second scheme. 140

B.45 The detection and matching results of “src22_ref_525.avi” for the third scheme. 141

B.46 The repetability and matching scores of “src22_ref_525.avi” for the third scheme. 141

B.47 The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for “src22_ref_525”. 142

B.48 The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for “src22_ref_525”. 142

B.49 The detection and matching results of “whaleshark_planetEarth_eps11.avi” for the first scheme. 143

B.50 The repetability and matching scores of “whaleshark_planetEarth_eps11.avi” for the first scheme. 143

B.51 The detection and matching results of “whaleshark_planetEarth_eps11.avi” for the second scheme. 144

B.52 The repetability and matching scores of “whaleshark_planetEarth_eps11.avi” for the second scheme. 144

B.53 The detection and matching results of “whaleshark_planetEarth_eps11.avi” for the third scheme. 145

B.54 The repetability and matching scores of “whaleshark_planetEarth_eps11.avi” for the third scheme. 145

B.55 The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for “whaleshark_planetEarth_eps11.avi”. 146

B.56 The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for “whaleshark_planetEarth_eps11.avi”. 146

B.57 The detection and matching results of ‘goats_planetEarth_eps5.avi” for the first scheme. 147

B.58 The repetability and matching scores of “goats_planetEarth_eps5.avi” for the first scheme. 147

B.59 The detection and matching results of “goats_planetEarth_eps5.avi” for the second scheme. 148

B.60 The repetability and matching scores of “goats_planetEarth_eps5.avi” for the second scheme. 148

B.61 The detection and matching results of “goats_planetEarth_eps5.avi” for the third scheme. 149

B.62 The repetability and matching scores of “goats_planetEarth_eps5.avi” for the third scheme. 149

B.63 The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for “goats_planetEarth_eps5.avi”. 150

B.64 The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for “goats_planetEarth_eps5.avi”. 150

B.65 The detection and matching results of “dolphins_planetEarth_eps9.avi” for the first scheme. 151

B.66 The repetability and matching scores of “dolphins_planetEarth_eps9.avi” for the first scheme. 151

B.67 The detection and matching results of “dolphins_planetEarth_eps9.avi” for the second scheme. 152

B.68 The repetability and matching scores of “dolphins_planetEarth_eps9.avi” for the second scheme. 152

B.69 The detection and matching results of “dolphins_planetEarth_eps9.avi” for the third scheme. 153

B.70 The repetability and matching scores of “dolphins_planetEarth_eps9.avi” for the third scheme. 153

B.71 The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for “dolphins_planetEarth_eps9.avi”. 154

B.72 The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for “dolphins_planetEarth_eps9.avi”. 154

B.73 The detection and matching results of “leopard_planetEarth_eps2.avi” for the first scheme. 155

B.74 The repetability and matching scores of “leopard_planetEarth_eps2.avi” for the first scheme. 155

B.75 The detection and matching results of “leopard_planetEarth_eps2.avi” for the second scheme. 156

B.76 The repetability and matching scores of “leopard_planetEarth_eps2.avi” for the second scheme. 156

B.77 The detection and matching results of “leopard_planetEarth_eps2.avi” for the third scheme. 157

B.78 The repetability and matching scores of “leopard_planetEarth_eps2.avi” for the third scheme. 157

B.79 The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for “leopard_planetEarth_eps2.avi”. 158

B.80 The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for “leopard_planetEarth_eps2.avi”. 158

B.81 The detection and matching results of “container.avi” for the first scheme. 159

B.82 The repetability and matching scores of “container.avi” for the first scheme. 159

B.83 The detection and matching results of “container.avi” for the second scheme. 160

B.84 The repetability and matching scores of “container.avi” for the second scheme. 160

B.85 The detection and matching results of “container.avi” for the third scheme. 161

B.86 The repetability and matching scores of “container.avi” for the third scheme. 161

B.87 The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for “container.avi”. 162

B.88 The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for “container.avi”. 162

B.89 The detection and matching results of “coastguard.avi” for the first scheme. 163

B.90 The repetability and matching scores of “coastguard.avi” for the first scheme. 163

B.91 The detection and matching results of “coastguard.avi” for the second scheme. 164

B.92 The repetability and matching scores of “coastguard.avi” for the second scheme. 164

B.93 The detection and matching results of “coastguard.avi” for the third scheme. 165

B.94 The repetability and matching scores of “coastguard.avi” for the third scheme. 165

B.95 The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for “coastguard.avi”. 166

B.96 The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for “coastguard.avi”. 166

B.97 The detection and matching results of “foreman.avi” for the first scheme. 167

B.98 The repetability and matching scores of “foreman.avi” for the first scheme. 167

B.99 The detection and matching results of “foreman.avi” for the second scheme. 168

B.100The repetability and matching scores of “foreman.avi” for the second scheme. 168

B.101The detection and matching results of “foreman.avi” for the third scheme. 169

B.102The repetability and matching scores of “foreman.avi” for the third scheme. 169

B.103The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for “foreman.avi”. 170

B.104The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for “foreman.avi”. 170

B.105The detection and matching results of “janine1_1.avi and janine1_2.avi” for the first scheme. 171

B.106The repetability and matching scores of “janine1_1.avi and janine1_2.avi” for the first scheme. 171

B.107The detection and matching results of “janine1_1.avi and janine1_2.avi” for the second scheme. 172

B.108The repetability and matching scores of “janine1_1.avi and janine1_2.avi” for the second scheme. 172

B.109The detection and matching results of “janine1_1.avi and janine1_2.avi” for the third scheme. 173

B.110The repetability and matching scores of “janine1_1.avi and janine1_2.avi” for the third scheme. 173

B.111The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for “janine1_1.avi and janine1_2.avi”. 174

B.112The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for “janine1_1.avi and janine1_2.avi”. 174

B.113The detection and matching results of “jungle_1.avi and jungle_2.avi” for the first scheme. 175

B.114The repetability and matching scores of “jungle_1.avi and jungle_2.avi” for the first scheme. 175

B.115The detection and matching results of “jungle_1.avi and jungle_2.avi” for the second scheme. 176

B.116The repetability and matching scores of “jungle_1.avi and jungle_2.avi” for the second scheme. 176

B.117The detection and matching results of “jungle_1.avi and jungle_2.avi” for the third scheme. 177

B.118The repetability and matching scores of “jungle_1.avi and jungle_2.avi” for the third scheme. 177

B.119The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for “jungle_1.avi and jungle_2.avi”. 178

B.120The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for “jungle_1.avi and jungle_2.avi”. 178

B.121The detection and matching results of “cam0_capture5_Deniz.avi and cam1_capture5_Deniz.avi” for the first scheme. 179

B.122The repetability and matching scores of “cam0_capture5_Deniz.avi and cam1_capture5_Deniz.avi” for the first scheme. 179

B.123The detection and matching results of “cam0_capture5_Deniz.avi and cam1_capture5_Deniz.avi” for the second scheme. 180

B.124The repetability and matching scores of “cam0_capture5_Deniz.avi and cam1_capture5_Deniz.avi” for the second scheme. 180

B.125The detection and matching results of “cam0_capture5_Deniz.avi and cam1_capture5_Deniz.avi” for the third scheme. 181

B.126The repetability and matching scores of “cam0_capture5_Deniz.avi and cam1_capture5_Deniz.avi” for the third scheme. 181

B.127The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for “cam0_capture5_Deniz.avi and cam1_capture5_Deniz.avi”. 182

B.128The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for “cam0_capture5_Deniz.avi and cam1_capture5_Deniz.avi”. 182

B.129The detection and matching results of “cam0_capture7_novice_jugglers.avi and cam1_capture7_novice_jugglers.avi” for the first scheme. . . 183

B.130The repetability and matching scores of “cam0_capture7_novice_jugglers.avi and cam1_capture7_novice_jugglers.avi” for the first scheme. . . 183

B.131The detection and matching results of “cam0_capture7_novice_jugglers.avi and cam1_capture7_novice_jugglers.avi” for the second scheme. . 184

B.132The repetability and matching scores of “cam0_capture7_novice_jugglers.avi and cam1_capture7_novice_jugglers.avi” for the second scheme. . 184

B.133The detection and matching results of “cam0_capture7_novice_jugglers.avi and cam1_capture7_novice_jugglers.avi” for the third scheme. . . 185

B.134The repetability and matching scores of “cam0_capture7_novice_jugglers.avi and cam1_capture7_novice_jugglers.avi” for the third scheme. . . 185

B.135The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for "cam0_capture7_novice_jugglers.avi and cam1_capture7_novice_jugglers.avi". 186

B.136The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for "cam0_capture7_novice_jugglers.avi and cam1_capture7_novice_jugglers.avi". 186

List of Tables

4.1 The number of detected and matched points and the match and repetition scores of the Harris, proposed and SIFT detectors for the test image couples. 48

4.2 The number of detected and matched points and the match and repetition scores of the Harris FPD for the test image couples. . 49

4.3 The number of detected and matched points and the match and repetition scores of the Gray-level patch, proposed and SIFT descriptors for the test image couples. 51

4.4 The number of detected and matched points and the match and repetition scores of the SIFT descriptors for the test image couples 52

4.5 The number of detected and matched points and the match and repetition scores of different scale selection schemes for the test image couples. 54

4.6 The number of following, not following, correct matches and wrong matches for the image patch given in Figure 4.3. 55

4.7 The properties of video sequences. 57

4.8 The matching and repetition scores for the schemes. 76

Chapter 1

Introduction

Feature point detection is one of the vital topics in image processing. Advancements in feature point detection techniques directly affect many open to research areas such as 3D scene reconstruction, object recognition, image registration, etc. Though the literature on feature point detection can be dated back to the late 70's, there is still no universal feature point detector. Currently the most widely used detector was designed in the late 80's. It has evolved since then but it is still far from an ideal universal feature point detector.

1.1 Feature Point Detection Concepts

A feature point is a salient point, which can be easily separated from the surrounding pixels. The most intuitive feature (or salient) points are corners in an image. There have been many different detectors to accurately localize such points. They can be initially separated into two groups: the detectors which operate on black and white edge contours, and, the ones which operate on gray-level images. We can also include the detectors which operate on color images but these detectors are essentially extensions of their gray-level counterparts. As a result, color image detectors are not counted as a separate class.

After broadly categorizing the feature point detection algorithms, we are going to introduce the detectors which are the milestones of the feature point detection literature. During the introduction we are going to explain some of the detectors in more detail. We begin with the gray-level feature point detectors.

An early feature point detector in this class was designed by Moravec to detect corners[14]. The detector basically measures the directional variance around a pixel. To measure the directional variance, correlations in the neighboring blocks around pixels are computed. By doing this directional derivatives around pixels are implicitly measured. As a result, this detector is known as the first derivative-based feature point detector or “interest point detector” as Moravec named it.

The Moravec’s detector was followed by Kitchen and Rosenfeld’s detector [15]. They noticed that corners lay along edge intersections. They proposed that the corner detection accuracy might increase if the edges in the image were detected and used for eliminating non-corner points. To implement the detector, they combined a directional derivative based detector with an edge detector. The corner detection performance was increased as a result of this restriction.

The Moravec’s detector had a serious drawback; the detector was limited by a small number of derivative directions. Moravec used 45 degrees shifted windows in the estimation of the correlation. This shifting scheme limits the derivative directionality. As a result, Moravec’s detector misses arbitrarily aligned corners. In the late 80’s Harris and Stephens analytically solved the problem of direction limitation and modified the Moravec’s detector. In their paper [16], Harris and Stephens showed that there was a strict relation between directional variance measure and local autocorrelation of an image patch. Moreover, they also proved that by using the gradients of an image, local autocorrelation function can be approximated for small shifts, instead of 45 degree shifts. Besides, they also showed that the gradient of an image, that can be used in approximation of the local autocorrelation function, could be estimated from both vertical and horizontal first-degree partial derivatives. With all of these observations and approximations, they estimated the directional variance measure and the local autocorrelation function for small shifts according to:

$$E(x,y) = [x\ y]M[x\ y]^t, \tag{1.1}$$

$$M = \begin{pmatrix} A & C \\ C & B \end{pmatrix};\ A = (\frac{\partial I}{\partial x})^2,\ B = (\frac{\partial I}{\partial y})^2,\ C = \frac{\partial I}{\partial x}\frac{\partial I}{\partial y}. \tag{1.2}$$

In the equations x and y represent image coordinates, M represents the Hessian matrix, E represents cumulative directional variance and I represents the image intensity. The details about the variables and derivations can be found in [16].

Due to practical reasons, they used the matrix M to find corners, instead of using $E(x, y)$. The eigenvalues (a, b) of M are proportional to the principle curvature of the local autocorrelation function. The principle curvature and, indirectly, the eigenvalues of M, define whether a pixel is on a flat region, edge or corner. Nevertheless, due to the difficulties in computation of eigenvalues, the determinant and trace of M are used in the computation of the cornerness function R, where

$$R = det(M) - k \times Tr(M),\ det(M) = AB - C^2,\ Tr(M) = A + B. \tag{1.3}$$

In this equation $det(M)$ and $Tr(M)$ are determinant and trace of matrix M, respectively; k is a constant which depends on various properties of image such as its contrast. The value of R, which is the cornerness measure, is computed for each pixel of an image. Then the local maxima, which are greater than a positive threshold, are chosen as corners.

Due to the analytical and computational limitations, as previously explained, they left the detector incomplete. Three years after Harris and Stephens published the detector, Tomasi and Kanade finished the incomplete work [17]. In the Harris corner detector eigenvalues were not computed due to computational complexity and reliability. However, in the gradient-based corner detection, the eigenvalues of the Hessian matrix give the most important cues about the cornerness of a pixel. So, in the Tomasi-Kanade corner detector, primarily, the Harris corner detector is applied to the image in order to reduce the computational load. Then, for each candidate corner, the eigenvalues of the matrix M are computed. After that a histogram is generated from the minimum of the eigenvalue pairs. Consequently, the histogram is examined for a breaking point. The breaking point defines the threshold for candidate pixels elimination. The detector selects the pixels with the minor eigenvalue greater than the threshold and sorts them in decreasing order. The sorting is very important because when the minimum eigenvalue is higher, the pixel becomes more reliable. Hence, matching starts from the top of the sorted corner list.

The detectors explained above use partial derivatives or derivative approximations to calculate surface curvature of the gray-level images. It should be noted that there are also other algorithms which use similar techniques to solve the feature point detection problem, given in [18], [19], [20], [21], [22], [23], [24].

In the mid 90's Lindeberg introduced a semi-automatic derivative-based multi-resolution feature point detection algorithm [27]. Mainly the detector creates a Gaussian pyramid from an image. Then it chooses points which are maximum along the scale direction. With this detection scheme one can not only detect a feature point but also select a validity scale in which matching should be done. This was one of the leading multi-resolution feature point detection schemes and inspired many researchers.

In the early 2000, Scale Invariant Feature Transform (SIFT), another multi-resolution feature point detector, was introduced by Lowe[25]. SIFT is a multi-scale feature point detection scheme. In SIFT, all of the scales are searched to find the interest points which are both spatial and scale local extrema.

The method starts with creating a form of scale-space. The SIFT's scale-space is produced from octaves. Octave intervals are either half or double of the previous or next octave, respectively. Octaves are generated by convolving the original image with a Gaussian Kernel (or basis). Each octave interval is divided into "s" many scales by convolving with another Gaussian kernel, but scales are not downsampled after the convolution. All of the images in an octave interval have the same resolution. Images are downsampled when moving from one octave to a lower one. After the scale-space is created, the adjacent scales in an octave are subtracted from each other to form the difference of Gaussian image. After that the feature points are detected by comparing the points with their spatial and scale neighbors. The points, which are local extrema in both scale and space, are selected as feature points. To accurately localize feature points a 3D quadratic function is fitted to the neighborhood of the detected corners. The quadratic function also helps to achieve subpixel resolution.

After the SIFT detector another multi-resolution feature point detector, Harris-Laplacian corner detector, was introduced by Mikolajczyk et.al [26]. The Harris-Laplacian corner detector treats values of each pixel locations on an image as a 1D signal in the scale dimension. Initially, Harris-Laplacian feature point detector detects corners by using Harris corner detector. Then, scale properties of each corner are determined by convolving each scale with a specific function and investigating 1D response with respect to scale as shown in Figure 1.1. Then the scale that the maximum of 1D signal occurred is selected as the characteristic scale, which is used for setting a common scale between the different views of the same scene as Lindeberg showed in [27]. At the end, corners and their characteristic scales, in which the feature point is valid, are determined. In this method, both Harris detector and Laplacian function are used since they are the optimal operators for the specified cases according to the experiments in the paper by Mikolajczyk et.al [26].

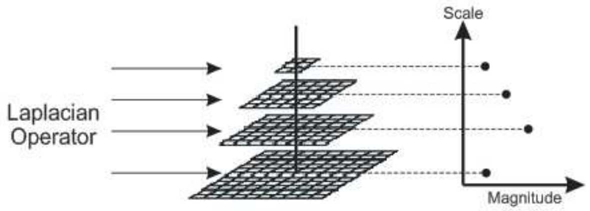


Figure 1.1: Selection of the characteristic scale

Morphological operators are also one of the important tools in signal processing. Various morphological detectors were proposed for feature point detection [28], [29], [30], [31], [32]. Main advantage of these detectors is the low computation load which allows real-time operations.

Some transformation based methods were also proposed. Basically, transformations represent the image data in a simpler and/or more useful way which makes the detection process easier (and faster). Various different transformations are used for feature point detection, such as Hilbert transform [40], wavelet transform [41], [42], matched filtering [43], various filter banks [44], and the phase congruency [38], [39], etc. Each of these detectors have different properties.

Finally, there are also some custom detectors which are applied to gray-level images. Some of these are based on non-linear operators [36], [37], some assigns cost functions to each pixel and minimizes it [35], some uses probabilistic models to detect corners [23] and some of them uses models to detect corners [33], [34].

Boundary based feature point detection is analogous to the gray-level detection. For example, derivative based techniques, as in the gray-level case, were proposed to detect the curvature of the contour (or edge) [1], [2], [3], [4], [5]. Transform based boundary techniques were also proposed in [7], [13]. Also custom boundary methods were proposed using various techniques, such as neural-networks to estimate curvature [6], correlation or $L_2 norm$ to find similarity [1], [11], chain codes to simplify the representation [8], [9], covariance metrics along the contour [10] and probabilistic approaches [12].

1.2 Problem Statement

As shown in the previous section, the aim of the most widely used feature detection algorithms is to detect corners. This approach leads to many problems as shown in various comparisons [45],[46]. The origin of the problems is that there is no robust definition of the corner concept. And some of the non-corner points also satisfy the existing cornerness criteria. Moreover, real-life cases make it even harder. For example, in Figure 1.2 we can see that due to downsampling, misleading corner structures are formed and these causes detectors to extract non-corner points as corners, even without aliasing effect during downsampling. This figure also shows us that the number of corners changes with respect to resolution. For example, when the detector is applied to a finer resolution image and its coarser resolution image, most of the corners, detected in the finer resolution, are lost in the coarser resolution due to the filtering which is used to avoid aliasing after downsampling.

Another problem with the many detectors is they use derivative based methods which are inherently erroneous in noisy environments. Moreover, when quantization is added to this situation, extracting corners accurately becomes a difficult task. Furthermore, in images, there are many non-corner

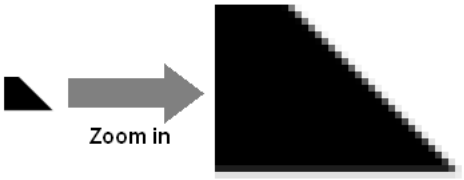


Figure 1.2: A corner sample with zoom-in

pixels which satisfies the multiple high directional derivative test, which is standard in derivative based detectors.

In addition to these drawbacks, the corner detectors have a lot of parameters which vary from one image to another. Most of the detectors even do not have an automatic parameter selection scheme. As each of the parameters change independently according to image properties, such as contrast, the manual detectors become hard to use. Because, to reduce possible errors the user should adjust the parameters for every different case, which significantly changes in real life situations.

The fundamental assumption of most of the feature point detectors, explicitly or implicitly, is "objects in images can be expressed as edges or contours". To have this representation, first of all edges (or contours) in images should be detected. The edges were assumed to represent the object boundaries. And irregularities on the boundaries are used as feature points, because they store more information than others. This basic assumption can be seen clearly in different detectors which operate on contours. However, it is not that obvious in detectors which operate on gray-level images. This is because edge detectors are combined with corner detectors. This can be seen in many algorithms such as [15], [16], [18], [25], [26], [36], etc. There are also some custom detectors which do not integrate an edge detector. But these detectors are also searching for a corner instead of a feature point. So, that custom detectors suffer from almost the same drawbacks as the other ones.

However, as shown in [25], [26], these drawbacks can be lessen by applying multi-resolution analysis in gradient estimation. Moreover, by describing feature points according to deviations around them and not by gray-level patches, SIFT detection-matching scheme has improved its performace more than the others. Hence, the algorithms are stablized and gained immunity to various real-life cases which cause to detection of non-feature point.

Due to these reasons, we propose a post-processing scheme to reduce the number of detected non-corner points. In this approach it is believed that objects in images should be represented by regions instead of edges/contours and each one of the feature points should be represented with respect to its neighborhood. Therefore, we can measure the degree of distinctiveness of a detected feature point among its neighbors. By using this measure non-feature points can be eliminated.

Moreover, we can also use this neighborhood description in feature point matching. A feature point is not an isolated pixel; due to physical constraints, it closely depends on its neighborhood. Therefore, if we use neighborhood conditioned pixel description, we can improve the matching quality.

The proposed post-processing and matching scheme should have some essential properties. For example, the method should be easily implementable and robust under mostly unpredictable, real-life cases. Therefore, the aim of the thesis is to design and implement a post-processing and matching method which is implementable and robust as explained previously, and increases the performance of applications which rely on feature point detection.

In order to have the proposed post-processing and matching scheme, we need to understand the properties of textures. Because, when we are referring to describe a point with respect to its neighbors; we are implicitly representing the texture around a feature point. This is, in fact, one of the crucial steps in our proposed post-processing and matching scheme.

Chapter 2

Texture

Although textures can be easily perceived and distinguished by humans, there is no single definition of texture. For example in analysis sense, we can think of a texture as a region which is homogeneous in some extent. However, the definition of the homogeneity depends on texture and observer. In the synthesis sense, we can think of a texture as an output of a process. Nevertheless, the texture generating process is, in most cases, highly complex and unknown. In almost every situation, the bottleneck of texture studies is the lack of a single definition.

2.1 Texture Definition

As previously mentioned there is no single texture definition and tens of different definitions can be found in the literature. However, there are two basic definitions given by Sklansky et.al [56] and Tamura [55]. These definitions are essential to understand the texture concept.

Texture, according to Sklansky, is a regional attribute whose local properties (statistics) are constant, slowly varying or almost periodic.

On the other hand, texture, according to Tamura, can be expressed by a number of primitives which have various spatial organizations.

We can think the second definition and its variants as a subset of the first one. Because, if we know the repeating primitive, we can extract its statistical attributes and use these to define a homogeneity measure. The homogeneity measure helps to indicate whether a region is locally constant, slowly varying or periodic as stated in the first definition.

As the first definition covers most of the others definitions, we are going to use the first definition which is moderately broad, when compared with the others which are given in [173]. Hence, to be more precise, "texture" term is used to describe a region which is composed of closely blended or mixed elements with properties defined by Sklansky et.al [56].

In order to use this definition in a specific case, we have to extract homogeneity measure (statistical attributes) of the texture by applying texture analysis methods.

2.2 Texture Analysis

In this section, we are going to overview not only texture analysis, but also synthesis and segmentation methods. Because, almost all of the analysis, synthesis and segmentation algorithms rely on similar fundamental assumptions. And as a result, almost all of the texture processing methods (analysis, synthesis and segmentation) give insight about textures and performance of the applied methods.

One of the earliest methods was proposed by Haralick et.al. [57] in 1973. The proposed method was based on second order statistics or in other words "occurrence frequencies" [67], [66] . Basically, they generate a square matrix for a specific separation distance and angle. This separation distance and angle define the distance between the reference pixel and the pixels under investigation. The row and column indices of the matrix represent the gray-level of the current pixel and the pixels under examination, respectively. As a result, the matrix represents the occurrence frequencies of gray-levels pairs at the given separation distance. This matrix form is also known as "co-occurrence matrix". A simple co-occurrence matrix ($P_{d,angle}$) for a binary image is given by

$$P_{d,angle} = \begin{pmatrix} p_{i,j} & p_{i+1,j} \\ p_{i,j+1} & p_{i+1,j+1} \end{pmatrix}; i = 0, j = 0. \tag{2.1}$$

Note that, $p_{i,j}$ denotes the occurrence frequency of a pixel which has value i, separated from another pixel which has value j, with a specific distance d and $angle$. Furthermore, to analyze the given texture, many matrices are computed for different angles and separations [58].

The computational load of co-occurrence matrix can be reasonable for a binary image, but it becomes relatively too large to be used alone, when the number of quantization levels increases. As a result, various different features were defined to represent co-occurrence matrices in simpler forms [60], [64], [65]. Furthermore, the description performance was improved by applying DCT to the co-occurrence matrices to make the components less correlated [63]. What is more important, several fast co-occurrence matrix computation methods were proposed [61], [62].

Nevertheless, as most of the other texture analysis methods, this technique suffers from a huge computational load. As the aim was extracting second order statistics, the computation load increases exponentially with the increasing number of quantization-levels. For example, a co-occurrence matrix generated from a binary image has two dimensions and four elements. On the other hand, a co-occurrence matrix generated from an 8-bit gray-level image has 256 dimensions and 65536 elements and that causes a fairly high computation cost for a single pixel.

Another early approach was based on autoregressive image models. In this approach, autoregressive models (AR) were used to extract the underlying texture statistics [68]. Basically, AR models are used in predicting the future values of time-series as:

$$X_t = c + \sum_{i=1}^{p} \phi_i X_{t-i} + \epsilon_t. \tag{2.2}$$

where X_t is the value to be predicted, X_{t-i} are the past values, p is the order of the prediction, ϵ_t is the prediction error and ϕ_i are the prediction coefficients. As can be seen in the equation, AR models also emphasize the short-range interaction by weighting past values. The weighting is used to define statistical properties of textures.

In image processing, time-series are formed from images in a raster-scan scheme. In the raster-scan, images are transformed from $N \times N$ matrices to $N^2 \times 1$ vectors. As a result, a series is formed from an image which has some basic properties like causality.

There are many different autoregressive models. The main goal of AR methods is finding the best model to represent a given texture. Various AR models were proposed, such as, non-causal AR [69], [76], simultaneous AR [70], [72], [71], multiresolution Gaussian AR [73], circular symmetric AR [74], 2D quarter-plane AR [75], recursive AR [77], generalized circular AR [78], etc.

According to the applied model, prediction coefficients are computed. And based on the coefficients, texture statistics are defined. Abbadeni [79] gives some examples on AR model parameters and their perceptual meaning.

AR models have proved their usefulness in extracting the underlying statistical behavior of time-series. There are many well-known signal processing algorithms which use these models, such as GSM vocoders. Furthermore, with some improvements, such as making them rotation and scale invariant, they become a good texture descriptor.

Another model-based approach is based on Markov (or Gibbs) random fields. Briefly, Markov (or Gibbs) random field (M/GRF) approaches interpret images as random fields. In a Markov/Gibbs random field, the probability of the value of a pixel depends only on the values of the neighboring pixels. This is known as the Markovian property. And the conditional probability distribution function, which is occurrence of a pixel conditioned to its neighborhood, is in exponential form. Further information on M/GRFs can be found in [98] and [97].

Although the M/GRF model and its multi-resolution counterparts, [96], [87], [93], [89], are well-defined and straightforward, estimation of the model parameters is tricky. Moreover, instead of the others, proposed approaches vary in parameter estimation algorithms. For instance, some approaches use dynamic programming [81], [91], [82], [83]; some use pseudo-maximum likelihood [86]; instead of maximum likelihood, some use least squares [84] whereas some others use simulated annealing [85], [80], [94] and [90].

In each of the proposed methods there is a trade-off between computational complexity and parameter estimation accuracy. For example, the parameters can be estimated accurately by using simulated annealing (SA) but the computation load of SA can be overwhelming. On the other hand, the computational load may be low if the least squares estimation methods are used. However, the estimation accuracy will be poorer in this case.

Another well-known model based texture analysis method was proposed by Pentland et.al.[99]. In that paper, the textures are modelled as fractal functions. Fractals are described by their complexity. The complexity parameter is known as “fractal dimension”. As in the parameter estimation of M/GRF model, estimating the fractal dimension is difficult. Many different ways to estimate the fractal dimension were proposed, [102], [103], [104]. Though some of them are useful, it was shown that fractal dimensions alone cannot be used to discriminate different textures [106]. To overcome the lack of discrimination, some modifications were suggested, such as using lacunarity measures [105], [100], extended self-similarity model [101], etc.

Transformation techniques are highly important in signal processing. Basically, transforming a signal from one space to another, where they can be represented according to the needs, has positive effects on performance; such as, increasing accuracy, decreasing computation load, etc. As in the other signal analysis cases, signal transformation techniques can be helpful in texture analysis, too. Because, based on experiments, each texture has a specific signature and transforms can be used to represent textures in a more useful form to easily extract these signatures. Therefore, the purpose of the transform based methods is representing a texture in an appropriate way and finding its characteristic signatures. We have to mention here that these signatures are not unique for every texture. Because, in addition to these structural signatures, textures also have a random characteristic. Nevertheless, extraction of these signatures are generally enough for most cases.

In order to extract the texture signatures, Short Time Fourier Transform is used in [107], [108], [109]. In [110], [111] the Circular-Mellin Transform, which is Fourier Transform in polar coordinates, is used to extract scale-rotation invariant features from textures. Furthermore, also Gabor Transform is utilized in various approaches. Some of the approaches create a filter bank which covers large portion of the space-spatial frequency domain [127], [128], [129], [132], [133], [134], [137] and [138]. Some of them propose adaptive filter parameter (center frequency, orientation, bandwidth) selection schemes for the given input textures [139], [131], [135] and [136]. Moreover, Wigner Ville Distribition based methods were also applied in texture analysis in different ways by [117], [118] and [119].

Another integral transform texture analysis method is called "Local Linear Transform" (LLT). In LLT, the texture is convolved by a custom filter bank. Then, the energy of the channels (filter outputs) are used in feature selection. Various methods were proposed [151], [152], [153], [154], [155], [156], [157], [158], [159], [161] and [160]. One of the famous LLT was proposed by Laws who uses wavelet like basis [159]. The performance of LLT can be improved by using DCT to decorrelate the filter output [153].

There are also some custom texture analysis methods. These methods use various techniques, such as moments [162], [163], [164], histograms [174], metric spaces[175], run-lengths[176], Voronoi polygons[177], dynamic matched filtering [179], local frequency[180], etc.

Chapter 3

Non-Feature Point Elimination And Feature Point Matching

Images and video frames can be used for different purposes and objectives. In some applications, such as 3D scene reconstruction, object recognition, image registration, etc., one may want to uncover point-to-point relations between two images (or frames). There are various ways to find out this relation. One of the popular ways of this is finding matches between the images (or frames). In this type of methods points are represented by some attributes. And the point on the other image with the most similar attributes are assigned as a match. The operation of making point-to-point similarity assignments is known as "Point Matching".

To make better matches between images (or frames) instead of ordinary points, the points with some distinctive features are selected. And the matching is started from these points. The operation of selecting distinctive points is known as "Feature Point Detection". However, the selection of distinctive points is complicated. Images contain variety of pixel combinations and these combinations sometimes fulfill the feature point criteria of detectors and deceive the detection algorithms. As a result of this, the set of detected points contains both feature points and non-feature points.

The false selection of a non-feature point as a feature point is highly undesirable. The inclusion of these points in a set of feature points decreases the reliability of matching step. Because, as these points are not distinctive enough, they may cause mismatches. Due to the inadequate matchings, most of the applications which rely on point-to-point matches may give erroneous and unpredictable results. Hence, to ensure the performance of the applications which rely on feature matching non-feature point should be eliminated.

However, the methods used in detection may not solve the elimination problem, because, non-feature points may have higher detection scores than regular feature points. As a result of this some desired feature points may be discarded while trying to eliminate non-feature points by using the approaches similar to the detection methods. Due to this, a different approach should be utilized in elimination.

As explained in the first chapter, most of the popular feature point detection methods directly or indirectly uses color or intensity attributes of pixels to describe them. However, just color or intensity values are not enough to represent the distinctiveness of a pixel. There should be additional or alternate, descriptors to represent the pixel attributes.

Describing the texture around a point is one of the important elements for representing regional attributes of a point. As explained in chapter 2 there are various ways of describing a texture, basically each of these descriptors uses the neighborhood structure around a pixel to form a regional point representation. As feature points should be distinctive among their neighbors, texture based descriptions are highly useful in pixel classification. Furthermore, these descriptors can also be used in feature matching, too; because, due to physical constraints neighborhoods may preserve their structure. Hence, for every candidate point couple both pixel-wise and regional attributes should match each other. Due to this, better matching results can be obtained by including texture descriptors in point representation.

Although texture description is one of the powerful tools in image processing, it creates some undesired complications in implementation. First of all creating a texture description around each of the pixels of an image is computationally impractical. Because, it requires a large memory space and a high processing power. Even applying texture description only to a set of feature points is computationally overwhelming, due to the scale dependency of textures. Nevertheless, there are ways to bypass these implementation problems.

3.1 Scale Selection

Images can be represented in different ways, one of them is representing images in multi-resolution. In multi-resolution, an image with spatial resolution $N \times M$ per unit area is filtered and downsampled by d to have a coarser image with smaller resolution $N/d \times M/d$ per unit area. The acquired image representation is called a scale. An image and its two scales are given in Figure 3.1 and Figure 3.2, respectively.

The collection of extracted scales is called the “scale-space”. In the proposed method the coarsest and the finest scales of a scale-space is restricted. The range of scales are limited between the sampling rate of the imaging sensor (finest resolution) and a single pixel (coarsest resolution).

Infinitely many scales, in a given range, can be extracted from an image. Furthermore, each of the patches extracted from different scales at relatively

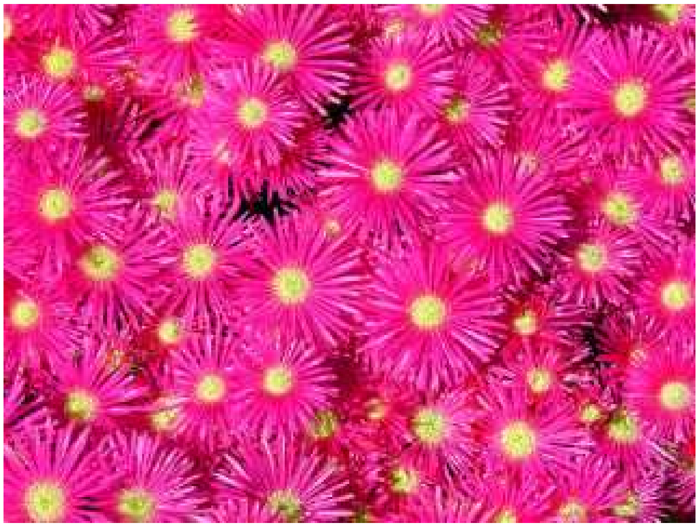


Figure 3.1: A scale of a sample image with textures

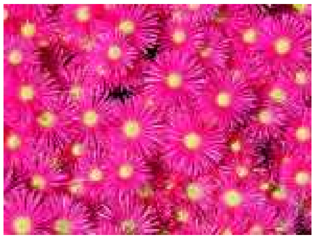


Figure 3.2: Another scale of the same image shown in Figure 3.1

on the same position may have different pixel values. As pixel values inside the patches change from one scale to another, neighborhood structure also changes. Hence, the nature of the texture changes. In order to have a complete texture description every scale of an image should be analyzed and represented.

This scale dependency of textures creates a serious bottleneck in implementation, even for the proposed elimination method which processes relatively a small number of points. One of the solutions to this computation problem is choosing a single scale for analysis according to texture signatures and ignoring the rest of the scale-space.

As explained in the second chapter, every texture has a signature in the scale-space. In the scale selection, the scale which contains the most discriminative texture signatures is desired. Because, if the selected analysis scale does not supply discriminative information, then the extracted texture description will be misleading. And, at the end, the entire approach will be useless.

For example, in the Gabor filter bank case, if one of the filters is tuned to a range which is out of the scale in which the discriminative texture signature lies, then the resultant description may have noise like terms which may give misleading information about the content. This observation also holds for Gibbs/Markov random fields. Basically, GMRF describes a texture by clique potentials which represent pixelwise interactions. These interactions are derived from pixel formations. As pixel formations depend on scale, GMRF structure and model parameters change, too. Hence, if nondiscriminative pixel interactions are modeled, texture may be represented in a misleading way. And the descriptor will be ineffective.

To have an adequate texture description, first of all, a color space should be selected. There are various ways to use a multi-spectral image in multi-resolution analysis. In the proposed method the luminance channel of $CIE\ L^*a^*b^*$ color space is used [189]. The given image, generally in RGB 24-bit raw format, is transform to $CIE\ L^*a^*b^*$ color space based on ITU-R BT.709 using D65 as the white point reference [191]. This color space is selected instead of the many others (such as RGB, YUV, CMYK, CIExyz, etc.), because, $CIE\ L^*a^*b^*$ is a linear color space. A color space is linear, if a unit change in color in any place of the color space makes the same amount of change in its numeric representation. The linearity property is important for color spaces in image processing. Because it avoids bias coming from the comparison of different color values. The luminance channel of an image which is shown in Figure 3.1 is given in Figure 3.3. After representing the image in $CIE\ L^*a^*b^*$ color space, scales are extracted by using the luminance channel.

The scales of an image can be acquired in many ways. Among these we chose the wavelet transform. The flexibility and FIR applicability of wavelet

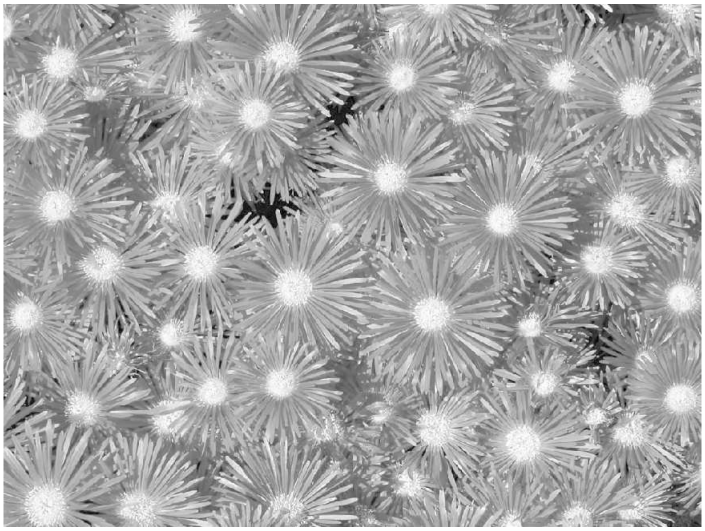


Figure 3.3: Luminance channel of $CIE\ L^*a^*b^*$ image which is shown in Figure 3.1

transform were the main reasons behind this decision. Among wavelet transform schemes, biorthogonal spline wavelet family is used in the proposed method [150]. The biorhogonal spline wavelet family has several nice properties, such as biorhogonality, compact local support (FIR implementability), symmetry, etc. These properties are useful in image processing. A sample wavelet decomposition is shown in Figure 3.4 for the luminance channel which is given in Figure 3.3.

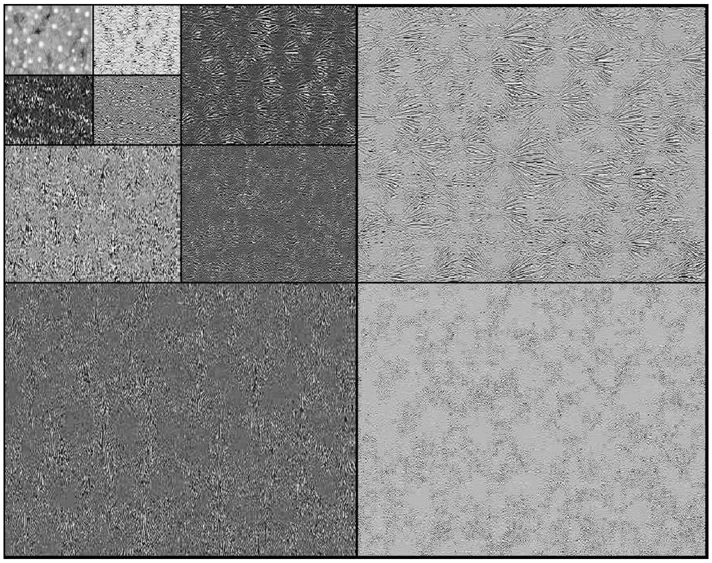


Figure 3.4: Wavelet decomposition of the Luminance channel of $CIE\ L^*a^*b^*$ image which is shown in Figure 3.1

After decomposing the image by using biorthogonal spline wavelets, LL subbands of each of the scales are gathered to form a multi-resolution representation. To select the analysis scale from the collection of scales, a set of patches is extracted in every scale around each of the candidate feature points. We call this subset of scale-space "neighborhood scale-space". Though a pixel is the coarsest scale of a scale-space for a finite image, in the proposed method scales coarser than a resolution threshold are ignored due to the applicability of the adopted texture description method. We call this coarsest resolution bound as "termination resolution".

The termination resolution strictly depends on the adopted texture description algorithm. If the texture descriptor uses an $N \times M$ patch, then the termination resolution should be equal or greater than $N \times M$. Because, in the scales coarser than this resolution the adopted texture description algorithm cannot be applied effectively. However, for some small texture analysis windows with small N and M values, the termination resolution can be much higher than $N \times M$; because, as the resolution decreases, by filtering and downsampling, image pixels merge and feature points disappear, and this decreases the reliability of the elimination method by discarding most of the points without analyzing their neighborhood structure. Therefore, the termination resolution, which does not change from an image to another or from patch to patch, should be at least equal to the maximum neighborhood patch size of the texture descriptor. Nevertheless, during the implementation if the descriptor analysis window size is set to a smaller value, then the termination resolution can be much larger than this. A sample neighborhood scale-space is shown in Figure 3.5 for a 31×31 neighborhood patch size.

In texture description we would like to represent the neighborhood structure with the highest possible detail. Though the finest scale supplies the highest detail, in most of the cases neighborhood in the finest scale may not

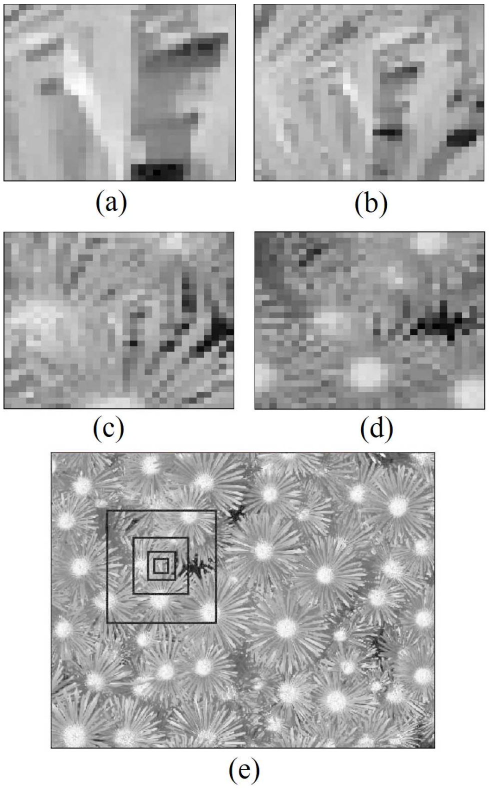


Figure 3.5: (a), (b), (c), (d), 31 × 31 patches taken from the image shown in Figure 3.3 from finer scales to coarser scales, respectively, (e) the projection of patches onto the original image.

be the most convenient one to be used in description due to texture features. As a patch is used in describing the neighborhood sctructure of a point, the neighborhood representation depends on the scale of the analysis. Furthermore, in some cases if the finest scale is used, a small portion of the texture signature can be represented. To avoid mis-representation, the descriptions in coarser scales are desired. However, in coarser scales image details are partially lost due to filtering and sampling. To find a balance between texture description and texture detail, we chose to use the coarser scale after a dramatic change in the patch content. The dramatic change between a coarser scale to a finer scale is counted as a sign of signature loss. And, it is supposed that the scale before the partial loss of texture signature can be used in reliable texture description.

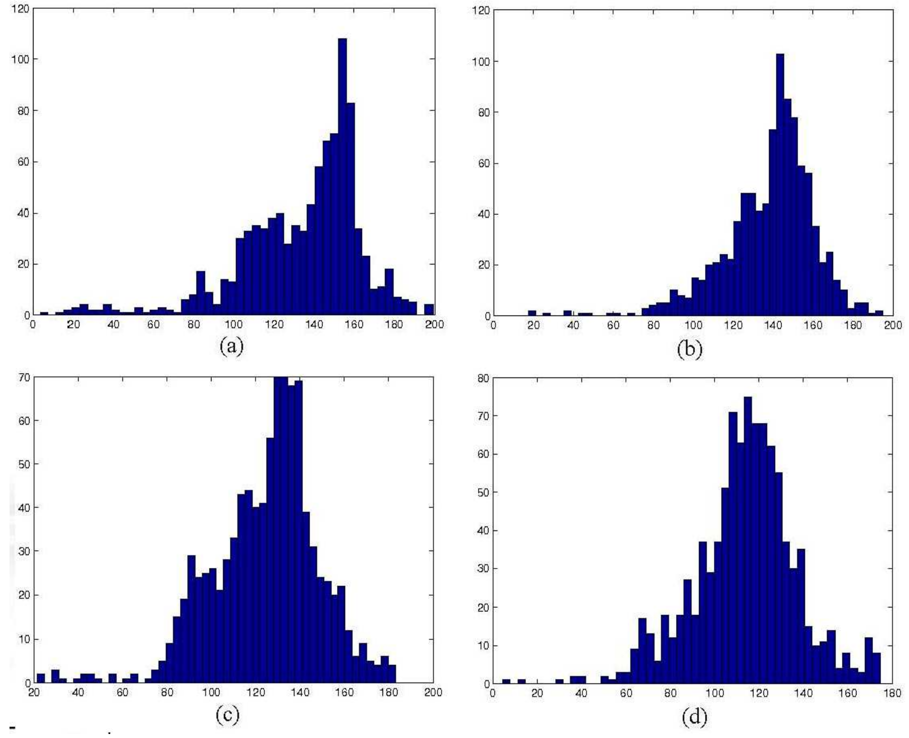


Figure 3.6: The histograms of the corresponding patches which are shown in Figure 3.5

To find the scale in which the desired texture signature lies, we chose to use the kurtosis value of histogram of each patch in the neighborhood scale-space. Kurtosis is defined as

$$Kurt(X) = \frac{\mu_4}{\sigma^4} - 3, \tag{3.1}$$

which is the forth central moment normalized with respect to normal distribution. Sometimes it may be used without normalization as

$$\mu_n = E((X - \mu)^n), \tag{3.2}$$

in which n equals to 4.

Kurtosis measures the amount of deviation from the mean due to the peaks in the histogram. This measure implicitly shows how complicated the given patch is. For example, if the kurtosis value is high, it means that the patch contains some dominant pixels values, and that indicates that the neighborhood is mostly composed of the same pixel value and, due to this, it is expected to have a simple structure. If the kurtosis value is low, it means that there are various different pixel values inside the patch and none of them is dominating that indicates that the neighborhood is composed of a variety of colors or intensities and the structure is expected to be complicated.

In the proposed method deviations in the kurtosis values across the scales are examined for the scale selection. It is believed that if the kurtosis values across the scale-space have a sharp positive change as we go from coarser to finer, it means a dramatic change in the neighborhood structure from a complicated structure to a simpler structure. And this abrupt positive change is counted as a sign of obscure texture feature in the finer level. Because it is supposed that due to the disappearance of a variety of colors in the neighborhood, the neighborhood structure became less complicated and as a result the texture signature is obscured.

Therefore, in scale selection dramatic positive transitions are detected to find the scale which describes the texture in a more reliable way as it contains more complicated neighborhood structure.

A sample scale versus kurtosis curve is calculated from the histograms given in Figure 3.6 for the patches shown in Figure 3.5 which is given in Figure 3.7. In this curve it can be noticed that there is a sharp positive transition while moving from coarser scales to finer scales. This sharp positive transition is a sign of an abrupt change in the patch content. Because, in the coarser scale before the transition there is a complicated neighborhood formation; however, in the adjacent finer scale there is a much simple neighborhood pattern. Hence, the coarser scale before the transition, which is the second scale, is considered as the analysis scale.

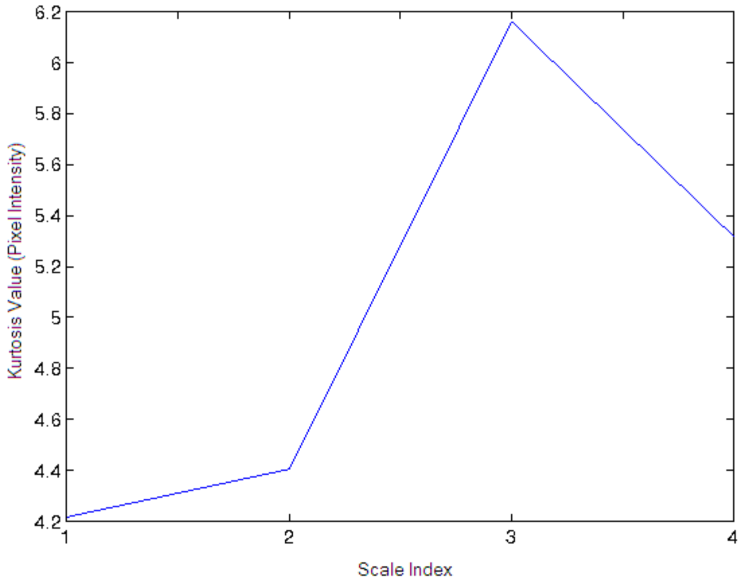


Figure 3.7: Kurtosis values vs scale which are calculated from the histograms of pixel intensities. X-axis shows the scales from 1 (coarsest) to 4 (finest) and Y-axis shows the corresponding kurtosis values.

In the proposed method sharp positive transitions are determined by applying a transition threshold. As it is undesired to set a transition threshold manually for each of the neighborhood scale-space of candidate feature points, an automatic threshold selection method is utilized. The transition threshold is calculated by

$$th_{trans} = \frac{1}{N-1} \sum_{i=1}^{N-1} |Kurt_i - Kurt_{i+1}|, \tag{3.3}$$

for every candidate feature point. In this equation N is the number of extracted scales, i is the scale index and $Kurt_i$ is the kurtosis value of the patch acquired from the i^{th} scale.

In the scale selection, by using kurtosis curve, there can be three different possible situations. In the first one there can be only one sharp positive transition. In this case the coarser scale before the transition is selected as the analysis scale. In the second case there can be more than one sharp positive transitions. In this case the coarser scale before the first sharp positive transition is selected. In the final case no sharp positive transitions are detected. In this case the finest scale is selected for analysis.

3.2 Texture Description

There are different ways of describing a texture as explained in the second chapter. Each of the methods have its own advantages and disadvantages. Among these the method described in [91] is selected for neighborhood structure description. Though this method has some implementation drawbacks, it is the one which fits to our design concerns.

This method describes a texture by using a Gibbs/Markov random field model. Although it can be implemented for an arbitrary number of gray levels, in the proposed method it is implemented for binary images. Because, the computational complexity of estimating binary GMRF parameters is relatively low when compared with multi-level counterparts. Moreover, calculating the probability of a pixel value conditioned to its neighborhoods is relatively easy and straightforward for binary images, when compared to multi-level counterparts. As a result of this choice, primarily, a threshold is applied to the patch extracted around the candidate feature point in the selected scale. However, a single threshold value cannot be used in binarization.

To have a better GMRF description in this method the threshold value should be changed from a point neighborhood to another. Because, neighborhoods of each of the points may contain different gray-level values. Moreover, each of the neighborhood patches belonging to the same point but taken at different scales should be thresholded with different threshold values, because, each of the scales may contain different gray-level values. Due to these limitations, to have an efficient thresholding scheme an automatic threshold selection method is used.

There are many threshold selection methods in the literature. Most of the researches in the field was conducted during the earlier times of image processing. In this era thresholding was generally referring to image segmentation, more specifically, separating a foreground object from a background. The most widely used thresholding techniques and associated experimental results are given in [188]. Among those, because of its nice properties, we selected to use k-means thresholding procedure.

Basically, k-means finds pixel representation, with the minimum description error, for a set of points. For example, if there are "P" number of points in a set and they are going to be represented with "C" number of arbitrary points ($P \geq C$), k-means positions "C" points to minimize the total error due to representing P points with C points. Hence, after k-means it can be said that, mathematically similar pixels are grouped into the same cluster to have the minimum representation error. This is not true for most of the other thresholding methods. For example, in entropy based thresholding, grouping is not done according to pixel similarity (minimum representation error) but according to achieve maximal pixel entropy after thresholding.

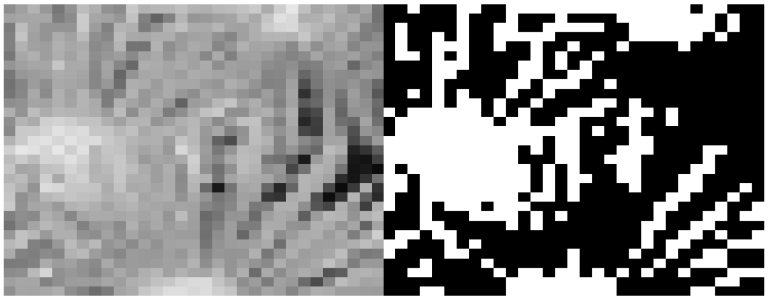
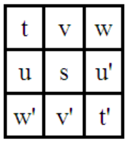


Figure 3.8: An illustration of k-means thresholding.

In the implementation, k-means is used for clustering a given patch into two groups which corresponds to black and white pixels after thresholding. The pixels inside the patch are counted as data points ("P"). At the beginning of the k-means, cluster centers ("C") are set to the maximum pixel value and the minimum pixel value in the neighborhood patch. Then the iteration step is started. At every cycle each of the points are assigned to a cluster whose center is closest to the point on the query. At the end of each cycle, cluster centers are updated in case of addition and/or subtraction of new points. The iteration step is terminated, when there is no point exchange between clusters that means the cluster centers are converged to their final locations, and the binary representation of gray-level image is acquired with the minimum representation error.

After thresholding, coefficient vectors are formed for each of the points inside the patch from their first order neighborhood. A sample coefficient vector is given by Equation 3.4 for the neighborhood shown in Figure 3.9.



t	v	w
u	s	u'
w'	v'	t'

Figure 3.9: First order neighborhood for the pixel "s".

For an $N \times N$ patch N^2 coefficient vectors are observed in a patch since there are N^2 distinct locations. A coefficient vector is formed by

$$x_k = [1 \ \ (u + u') \ \ (t + t') \ \ (v + v') \ \ (w + w')]^T, \tag{3.4}$$

in which $u, u', t, t', v, v', w, w' \in \{0, 1\}$. "0" denotes a black pixel and "1" denotes a white pixel. k denotes a set of neighborhood configurations which yields to the same coefficient vector which is x_k.

The coefficient vectors are extracted from every neighborhood inside a patch. An illustration of a patch is given in Figure 3.10. In this figure, a coefficient vector is calculated for every pixel in the gray region.

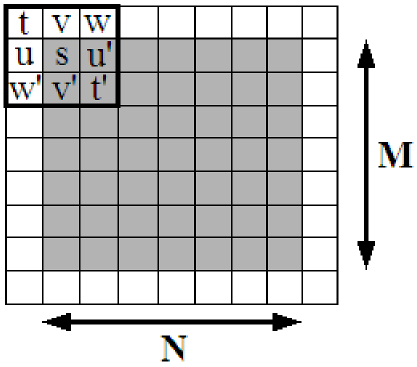


Figure 3.10: An illustration of the GMRF parameter estimation method.

After forming the coefficient vectors, similar coefficient vectors inside the patch are found and each different vector is assigned a distinctive group number k. Then both the number of coefficient vectors belonging to that group and the number of coefficient vectors which are belonging to the same group but from a neighborhood centered around a white pixel are counted. Subsequently the following algebraic equations are formed

$$\overline{x_k}^T \overline{w} = d_k, \tag{3.5}$$

$$\hat{d}_k = ln(\frac{N_{white,k}}{C_k - N_{white,k}}). \tag{3.6}$$

In Equations 3.5 and 3.6, $\overline{x_k}$ is the coefficient vector for the k^{th} group, $\overline{w}$ is the GMRF parameters for the patch, C_k is the number of vectors which are the same as $\overline{x_k}$, and $N_{white,k}$ is the number of vectors in the similarity group k which are from a neighborhood centered around a white pixel. d_k represent the energy function and $\hat{d}_k$ represents the estimate of the energy function for large C_k. Finally, T superscript denotes the matrix transpose operation.

After forming this linear equation the different coefficient vectors which are extracted from the patch are written in a matrix form which is given by

Equation 3.7. Consequently, the formulation of the matrix equation the GMRF parameters are estimated by solving

$$X^T \overline{w} = \overline{d}, \tag{3.7}$$

for $\overline{w}$, regional texture parameters, by calculating the pseudo-inverse of matrix X.

Though for an $N \times N$ patch N^2 coefficient vectors are formed, there can be a maximum 3^4 different coefficient vectors; because,

$$u + u', t + t', v + v', w + w' \in \{0, 1, 2\}, \tag{3.8}$$

$$u, u', t, t', v, v', w, w' \in \{0, 1\}, \tag{3.9}$$

Moreover, vectors which has $N_{1,k} = L_{1,k}$ or $N_{1,k} = 0$ are eliminated; because, these conditions make Equation 3.6 undefined or zero, respectively. Hence, at the end of the vector grouping and elimination, according to experiments, for a 31×31 patch there is a maximum of 25 different coefficient vectors, in other words, groups. Hence, the parameter calculation becomes less complicated.

After calculating the GMRF parameters, the occurrence probability of each of the candidate feature points conditioned to their neighborhoods are calculated by using the conditional probability formula:

$$P(X = i \mid \eta(X)) = \frac{e^{iT}}{1 + e^{T}}, \tag{3.10}$$

$$T = \overline{x}^T \overline{w}, \tag{3.11}$$

in these equations i is the color of the pixel which is either white or black, $\eta(X)$ is the first order neighborhood of the pixel.

3.3 Feature Point Elimination

Due to various reasons images may contain some tricky points which are not feature points but fulfills the feature point criteria of detectors. The inclusion of these points in a set of feature points decreases the reliability of matching step. As a result of this, these points should be eliminated from feature point sets. However, the methods used in detection cannot solve the elimination problem, because, non-feature points may have higher detection scores than regular feature points. As a result of this some desired feature points may be discarded while trying to eliminate non-feature points. Due to this a different approach should be utilized in elimination.

Although non-feature points cannot be distinguished according to pixelwise gray-level attributes, they can be easily separated from valid feature points by checking their neighborhood attributes. The occurrence expectancy of a point is one of the neighborhood attributes. In the proposed method, the occurrence expectancies of candidate points, that can be calculated by Equation 3.10, are used for measuring the distinctiveness of a point. For example, if a point has a high occurrence expectancy, it cannot be a feature point. Furthermore, a point with a low occurrence expectancy should be retained, because, it represents an irregularity. Hence, the points with high occurrence expectancy should be eliminated and the points with low occurrence expectancy should be retained.

This elimination method can discard non-feature points generated by various sources, especially regular texture patterns. Feature points from regular texture patterns are highly undesired in point matching. Though most of the feature points from texture patterns fulfill the feature point detection criteria of detectors, they are non-discriminative points. Because, a textural pattern may be encountered multiple times in a patch; and as a result of textural similarity around these points, they are prone to result in a mismatch.

On the other hand, points from textural irregularities, which are highly desired, are retained. A textural irregularity mainly disturbs the texture structure. Hence, it has a low occurrence expectation. And high probability of making a correct match. In the proposed algorithm though the points from regular texture patterns are eliminated, the points from texture pattern irregularities are retained. Furthermore, the points are sorted according to their accurence expectancy for to increase the reliability of matching by starting from the procedure with the most distinctive point.

In the proposed method, the occurrence expectancy of each of the feature points are calculated according to Equation 3.10. If the conditional probability of a feature point is larger than a predefined threshold value, the feature point is eliminated; if not, it is preserved.

3.4 Feature Point Matching

In the conventional matching algorithms, gray-level patches are taken around each of the detected feature points. After that the patches are correlated with each other and patch-couples which give the lowest absolute difference, which is calculated as

$$e_{i,j} = \sum_{l=1}^{N} |\overline{P_{1i}} - \overline{P_{2ij}}|. \tag{3.12}$$

In Equation 3.12 $e_{i,j}$ denotes the sum of absolute differences between two patches. $\overline{P_{1i}}$ is the patch from the i^{th} point in the first image, $\overline{P_{2j}}$ is the patch from the j^{th} point in the second image. N and l denotes the size of the patches in the vector form and pixel index of the patch, respectively.

In some algorithms a disparity restriction is used before the assignment of match couples. According to this, a point couple can be matched if and only if the Euclidean distance of their relative image coordinates is smaller than a predefined threshold. This restriction ensures that all of the matches are going to be inside a predefined ϵ-neighborhood.

The match quality of conventional matching methods can be determined for some extreme cases. For example, if $e_{i,j}$ is close to zero and the point is not from a regular texture pattern, it can be said that the match is correct due to the almost perfect overlapping between patches. Furthermore, if $e_{i,j}$ has an exceptionally high value, which depends on the patch size and number of gray-levels, it can be said that the match is wrong. However, it is not easy to classify the matches with the errors in between these extreme cases. Because, when there is a medium level of error, the goodness of match is uncertain. We cannot say that it is a correct match or a wrong match. Moreover, most of the matches in an image couple have medium level of errors and as a result of uncertain goodness of match, the performance of matching step is unclear and unpredictable.

To overcome this problem and increase the reliability of matching step, two new point descriptors are proposed. The first point descriptor is a neighborhood structure descriptor. Due to the previously explained physical constraint, for a valid match both point and its neighborhood should match the candidate point and its neighborhood. Gray-level patch correlation partially follows this; however, for uncertain matches neighborhood structure descriptors are needed to increase the reliability of matching. As mentioned previously, texture descriptors are highly useful in describing a neighborhood structure. Hence, the texture parameters extracted in the previous step can be used as the neighborhood structure descriptor of points.

The second proposed descriptor is a point status descriptor. It describes the status of a point inside a neighborhood. This type of descriptor is desired because as mentioned in the texture description, points with different values can be in the center of the exactly same neighborhood structure. To discriminate these points, their status in the neighborhood structure should be resolved. To find out the status of points in their neighborhood, conditional probabilities of the points are calculated by using Equation 3.10 which describes the status as the probability of occurrence with respect to the neighborhood structure.

In the proposed matching method the same conventional matching approach with ϵ-neighborhood restriction is followed. However, instead of starting from an arbitrary point, the matching is started from the point with the lowest conditional probability according to the sorting done in the elimination step. For every point in the first image, their candidate matches within the ϵ-neighborhood are found in the second image. After that each of the similarity measures (gray-level patch similarity, texture parameter similarity and conditional probability similarity) between the point in the first image and the candidate matches in the second image are calculated separately. These similarity values (or, in opposite, differences) are counted as a feature vector for a candidate point-couple. Hence, for each candidate point-couple a 3×1 feature vector is formed as

$$Match\ Features = [e_{i,j}^{gray-level\ patch}\quad e_{i,j}^{GMRF\ parameters}\quad e_{i,j}^{conditional\ probability}]^T. \tag{3.13}$$

In Equation 3.13 $e_{i,j}^{gray-level\ patch}$ denotes the sum absolute differences between gray-level patches that is

$$e_{i,j}^{gray-level\ patch} = \sum_{l=1}^{N}(|\overline{P_{1i}} - \overline{P_{2j}}|), \tag{3.14}$$

in which $\overline{P_{1i}}$, $\overline{P_{2j}}$, l and N denote the patch from the i^{th} point in the first image, the patch from the j^{th} point in the second image, the pixel index of the patch and the size of the patch in vector form.

In Equation 3.13 $e_{i,j}^{GMRF\ parameters}$ denotes the sum absolute differences between GMRF parameters that is

$$e_{i,j}^{GMRF\ parameters} = \sum_{l=1}^{5}(|\overline{w_{1i}} - \overline{w_{2j}}|), \tag{3.15}$$

in which $\overline{w_{1i}}$, $\overline{w_{2j}}$ and l denote the GMRF parameters of the i^{th} point the first image, GMRF parameters of the j^{th} point the second image and the index of GMRF parameters.

In Equation 3.13 $e_{i,j}^{conditional\ probability}$ denotes the absolute difference between conditional probabilities calculated by Equation 3.10 that is

$$e_{i,j}^{conditional\ probability} = |P(X_{1i} = l \mid \eta(X_{1i})) - P(X_{2j} = k \mid \eta(X_{2j}))|, \tag{3.16}$$

in which $\eta(X_{1i})$, $\eta(X_{2j})$, $P(X_{1i} = l \mid \eta(X_{1i}))$, $P(X_{2j} = l \mid \eta(X_{2j})$, l and k denote neighborhood of the i^{th} point in the first image, neighborhood of the j^{th} point in the second image, occurence expectancy of the i^{th} point in the first image conditioned to its neighborhood, occurence expectancy of the j^{th} point in the second image conditioned to its neighborhood, the pixel value of the

i^{th} point in the first image and the pixel value of the j^{th} point in the second image.

To avoid bias in the final measure. The candidate match feature vectors are collected and each of the elements are normalized to [0 1] interval with respect to other elements in its dimension. For example, $e_{i,j}^{gray-level\ patch}$ values are taken from all of the candidate matches and they are normalized to [0 1] interval. This is repeated for every dimension that is calculated as

$$eNorm_{i,j}^{gray-level\ patch} = \frac{e_{i,j}^{gray-level\ patch} - min(e_{i,j}^{gray-level\ patch})}{max(e_{i,j}^{gray-level\ patch})}, \tag{3.17}$$

$$eNorm_{i,j}^{GMRF\ parameters} = \frac{e_{i,j}^{GMRF\ parameters} - min(e_{i,j}^{GMRF\ parameters})}{max(e_{i,j}^{GMRF\ parameters})}, \tag{3.18}$$

$$eNorm_{i,j}^{conditional\ probability} = \frac{e_{i,j}^{conditional\ probability} - min(e_{i,j}^{conditional\ probability})}{max(e_{i,j}^{conditional\ probability})}, \tag{3.19}$$

in these equations $i \in [-P, P]$ and $j \in [-P, P]$. The minimum and maximum values are calculated from the set spaned by i and j.

At the end of the normalization, the average of three similarity measures are taken as

$$eNorm_{i,j}^{Average} = \frac{eNorm_{i,j}^{gray-level\ patch} + eNorm_{i,j}^{GMRF\ parameters} + eNorm_{i,j}^{conditional\ probability}}{3}. \tag{3.20}$$

and the candidate point-couple with the lowest average difference (or, in opposite, highest average similarity) is assigned as a valid match.

$$Match_{P_{1i}} = \arg_{i,j} min(eNorm_{i,j}^{Average}). \qquad (3.21)$$

And the procedure is continued to the following point with the lowest conditional probability after the current one.

3.5 Integrating Elimination and Matching Methods to Detectors

As we explained earlier most of the detectors have various parameters to be adjusted. This is referred as a drawback in the previous chapters, because, these parameters directly affect the accuracy of the feature detection and depend on image properties. As a result, they should be tailored carefully to have an accurate feature detection. However, in the proposed post-processing and matching scheme it is not a drawback; because, in the proposed method, it is assumed that the detectors inherently make inaccurate detections due to concentrating only on color or intensity similarity. So, it is not necessary to tailor the parameters to find a good configuration; even, it is better to set them loosely. Because, while carefully setting the parameters, some desired feature points may be lost. Hence, the adjustment of parameters are not important and not recommended.

In the implementation, first of all, the candidate points should be detected with loosely set parameters. Then the extracted points should be fed into the elimination algorithm. At the end of the elimination the sorted points should be matched by using the proposed matching method starting from the point with the lowest occurrence expectancy.

As the proposed detection-elimination-matching scheme is designed to avoid manual adjustments, most of the parameters are adjusted automatically during the feature analysis. Only the parameters from the feature point detection can be adjusted manually, but as explained previously loosely setting these parameters are enough for this scheme. Hence, in the proposed detection-elimination-matching scheme none of the parameter values are changed from image to image.

Chapter 4

Results

In this chapter four different test are done to evaluate the performance of the proposed scheme. Initially the performances of the proposed detector and the descriptor are evaluated by using a set of image couples. The proposed detector performance is compared with the Harris and SIFT feature point detectors (FPDs). The proposed descriptor performance is compared with the conventional gray-level patch and SIFT descriptors. Then, the proposed scale selection method performance is evaluated by using different scale selection methods. Finally, the performance of the proposed detection-matching scheme is compared with the Harris FPD and the gray-level patch matching scheme.

A set of image couples and a set of video sequences which are given in the supplementary CD are used in test procedures. A set of sample images from test image couples are given in Figure 4.1. Furthermore, a set of sample images from test video sequences are given in Figure 4.2. Note that the test image couples also include a frame couple from “garden” video sequence, whose sample image is shown in the test video sequences.

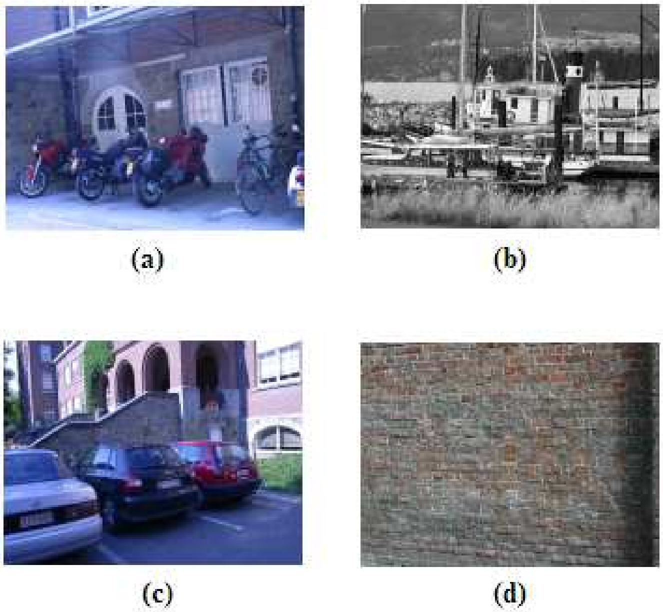


Figure 4.1: A set of sample images for each image couple which are used in detector, descriptor and scale selection performance evaluations.

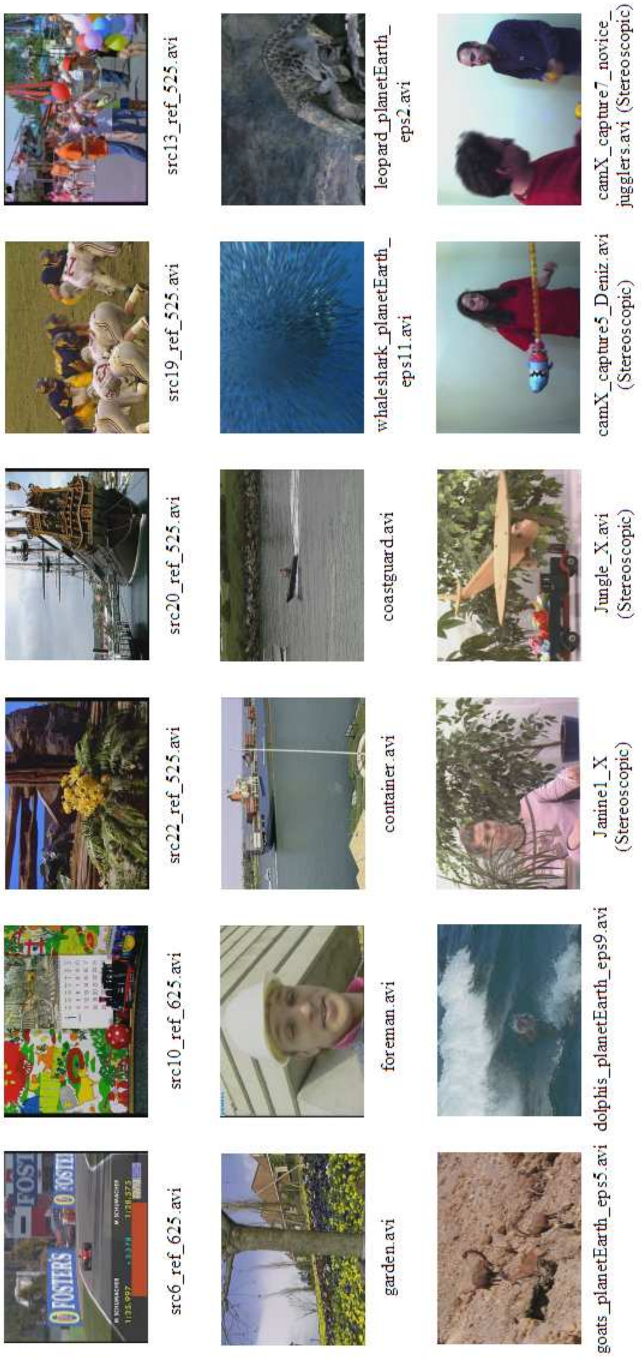

src6_ref_625.avi
src10_ref_625.avi
src22_ref_525.avi
src20_ref_525.avi
src19_ref_525.avi
src13_ref_525.avi
garden.avi
foreman.avi
container.avi
coastguard.avi
whaleshark_planetEarth_eps11.avi
leopard_planetEarth_eps2.avi
goats_planetEarth_eps5.avi
dolphis_planetEarth_eps9.avi
Janine1_X (Stereoscopic)
Jungle_X.avi (Stereoscopic)
camX_capture5_Deniz.avi (Stereoscopic)
camX_capture7_novice_jugglers.avi (Stereoscopic)

Figure 4.2: A set of sample frames from the video sequences which are used in comparisons between the Harris feature point detection / gray-level patch description scheme and the proposed feature point detection / description scheme.

Two different measures are used to evaluate the performance of the elimination and matching methods. The first measure which was proposed in [190] evaluates the repeatability of the detected feature points in a frame couple. In this measure, the repeatability denotes the matching of a detected point in one image to detected point within the ϵ-neighborhood in the other image. If a detected point in one image is not matched to a detected point within the ϵ-neighborhood in the other image, then it is supposed that the point is not detected in the other image; hence, not repeated. If a detected point is matched to a detected point within the ϵ-neighborhood in the other image, it is supposed that the point is also detected in the other image and as a result repeated.

This measure uses the minimum number of detected feature points in either of the frames and the number of matches within the ϵ-neighborhood over the entire frame. Then it assigns the ratio of two values as the repetition score:

$$r_i^{rep}(\epsilon) = \frac{|M(\epsilon)|}{min(n_i, n_{i+1})}. \tag{4.1}$$

In Equation 4.1, r_i^{rep} is the repetition score for the i^{th} frame couple which contains i^{th} and $(i+1)^{th}$ frames. $M(\epsilon)$ denotes the number of matched feature points within the ϵ-neighborhood over the frame. $min(n_i, n_{i+1})$ is the minimum number of detected feature points in the i^{th} frame and in the $(i+1)^{th}$ frame.

If the repetition score equals to 1, it means that each of the detected feature points are matched. If the score equals to 0, it means that none of the detected points are matched. Furthermore, if the repetition score is greater than 1, we can say that there are definite wrong matches, because number of matched points are greater than the number of minimum detected points. Hence, there are multiple match assignments to some points.

In this measure the scores close to but not higher than 1 are desired. However, this measure only checks the repetition of points within the ϵ-neighborhood in a given image couple, it does not give any insights about the matching quality. For example, the score can be equal to 1 by making totally correct matches, totally wrong matches or anything in between. The exact match quality is uncertain. So, the score should not be interpreted as a goodness of match measure.

To determine the goodness of match another measure is used. The measure which was proposed in [25] as

$$r_i^{match} = \frac{|P_{bestMatch} - P_{current}|}{|P_{secondMatch} - P_{current}|}, \tag{4.2}$$

evaluates the matching confidence by comparing the best match and the second best match of a point. r_i^{match} is the match score for the i^{th} point. $P_{current}$, $P_{bestMatch}$ and $P_{secondMatch}$ are the feature vectors, which is shown in Equation 3.13, of the current feature point, the assigned feature point to the current feature point and the second feature point which can be assigned to the current feature point after the assigned feature point, respectively.

It supposes that if the score is small, the best match is valid; because, there is a clear distinction between the first two candidate points of top matches. If the score is high, it indicates that the match could be wrong; because, the first two candidate points of top matches are similar to each other. Hence, the match can be incorrect also.

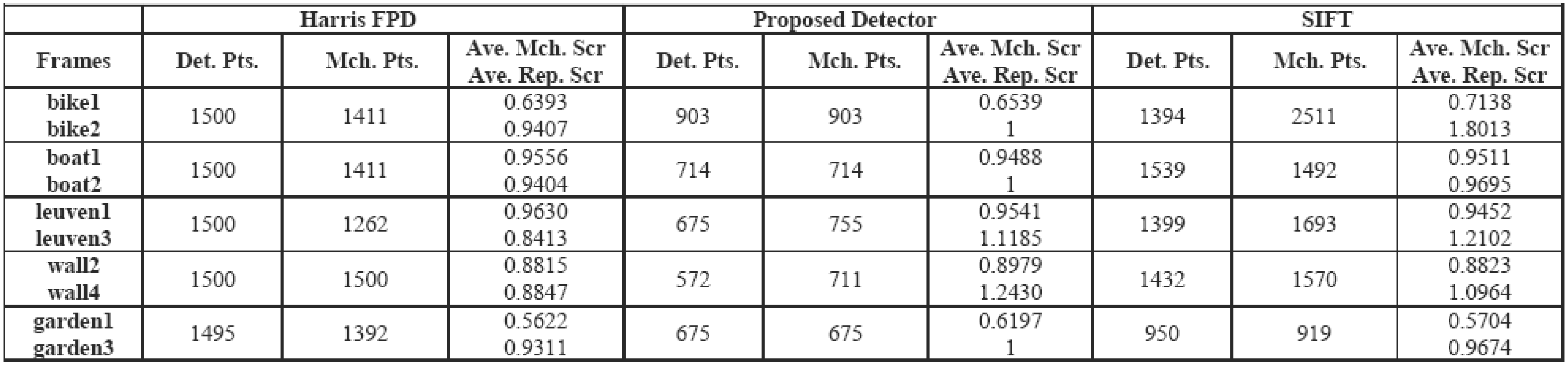
Four different approaches are followed in the performance evaluation. In the first two steps the detection and description performance of the Harris FPD, proposed FPD and SIFT are evaluated. After that the proposed scale selection method is evaluated. And finally, the conventional and the proposed detection-matching scheme performances are compared over a set of video sequences.

4.1 Evaluation of the Detector Performances

In the evaluation of the detector performances, feature points from the test image couples are detected by using the Harris, proposed and SIFT FPDs. After that the detected points are matched by using the gray-level patch matching method. At the end, the match and repetition scores are calculated.

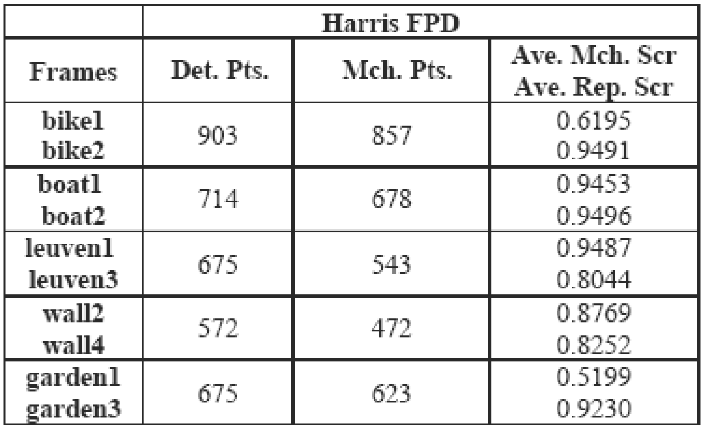
To have a better insight, initially nearly 1500 feature points are detected by using the Harris and SIFT FPDs. After that the points detected by the Harris FPD are fed into the proposed elimination method. Then the remaining points are counted as the points detected by the proposed method. Finally, the same number of points detected by the proposed method is detected by the Harris FPD and the match and repetition scores of all four cases are calculated which are given in Table 4.2.

Moreover, the detected and matched points by using the Harris, proposed and SIFT FPDs are given in Appendix A.1 for two different patches from “Bike” test image couple.



	Harris FPD			Proposed Detector			SIFT		
Frames	**Det. Pts.**	**Mch. Pts.**	**Ave. Mch. Scr** **Ave. Rep. Scr**	**Det. Pts.**	**Mch. Pts.**	**Ave. Mch. Scr** **Ave. Rep. Scr**	**Det. Pts.**	**Mch. Pts.**	**Ave. Mch. Scr** **Ave. Rep. Scr**
bike1 **bike2**	1500	1411	0.6393 0.9407	903	903	0.6539 1	1394	2511	0.7138 1.8013
boat1 **boat2**	1500	1411	0.9556 0.9404	714	714	0.9488 1	1539	1492	0.9511 0.9695
leuven1 **leuven3**	1500	1262	0.9630 0.8413	675	755	0.9541 1.1185	1399	1693	0.9452 1.2102
wall2 **wall4**	1500	1500	0.8815 0.8847	572	711	0.8979 1.2430	1432	1570	0.8823 1.0964
garden1 **garden3**	1495	1392	0.5622 0.9311	675	675	0.6197 1	950	919	0.5704 0.9674

Table 4.1: The number of detected and matched points and the match and repetition scores of the Harris, proposed and SIFT detectors for the test image couples.



	Harris FPD		
Frames	Det. Pts.	Mch. Pts.	Ave. Mch. Scr Ave. Rep. Scr
bike1 bike2	903	857	0.6195 0.9491
boat1 boat2	714	678	0.9453 0.9496
leuven1 leuven3	675	543	0.9487 0.8044
wall2 wall4	572	472	0.8769 0.8252
garden1 garden3	675	623	0.5199 0.9230

Table 4.2: The number of detected and matched points and the match and repetition scores of the Harris FPD for the test image couples.

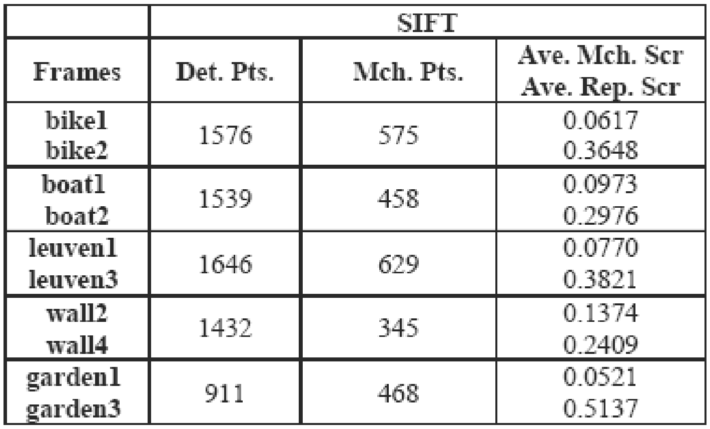
4.2 Evaluation of the Descriptor Performances

To evaluate the descriptor performances, initially 1500 feature points are detected by using the Harris FPD and they are fed into the proposed elimination algorithm. Consequently, the remaining points are matched by using the conventional gray-level patch, proposed and SIFT descriptors. Finally, the match and repetition scores which are given in Table 4.3 are calculated.

Furthermore, to have an unbiased evaluation the SIFT detector and descriptor are used alone and their match and repetition scores which are given in Table 4.4 are calculated.

	Gray-Level Patch Descriptor			Proposed Descriptor			SIFT Descriptor		
Frames	Det. Pts.	Mch. Pts.	Ave. Mch. Scr Ave. Rep. Scr	Det. Pts.	Mch. Pts.	Ave. Mch. Scr Ave. Rep. Scr	Det. Pts.	Mch. Pts.	Ave. Mch. Scr Ave. Rep. Scr
bike1 bike2	903	903	0.6539 1	903	667	0.5427 0.7386	277	1	0.1887 0.0108
boat1 boat2	714	714	0.9488 1	714	613	0.7658 0.8585	270	1	0.1195 0.0111
leuven1 leuven3	675	755	0.9541 1.1185	675	613	0.7129 0.9081	219	3	0.0933 0.0137
wall2 wall4	572	711	0.8979 1.2430	572	556	0.7097 0.9720	238	1	0.1640 0.0126
garden1 garden3	675	675	0.6197 1	675	438	0.5345 0.6489	160	17	0.0625 0.1083

Table 4.3: The number of detected and matched points and the match and repetition scores of the Gray-level patch, proposed and SIFT descriptors for the test image couples.



	SIFT		
Frames	Det. Pts.	Mch. Pts.	Ave. Mch. Scr Ave. Rep. Scr
bike1 bike2	1576	575	0.0617 0.3648
boat1 boat2	1539	458	0.0973 0.2976
leuven1 leuven3	1646	629	0.0770 0.3821
wall2 wall4	1432	345	0.1374 0.2409
garden1 garden3	911	468	0.0521 0.5137

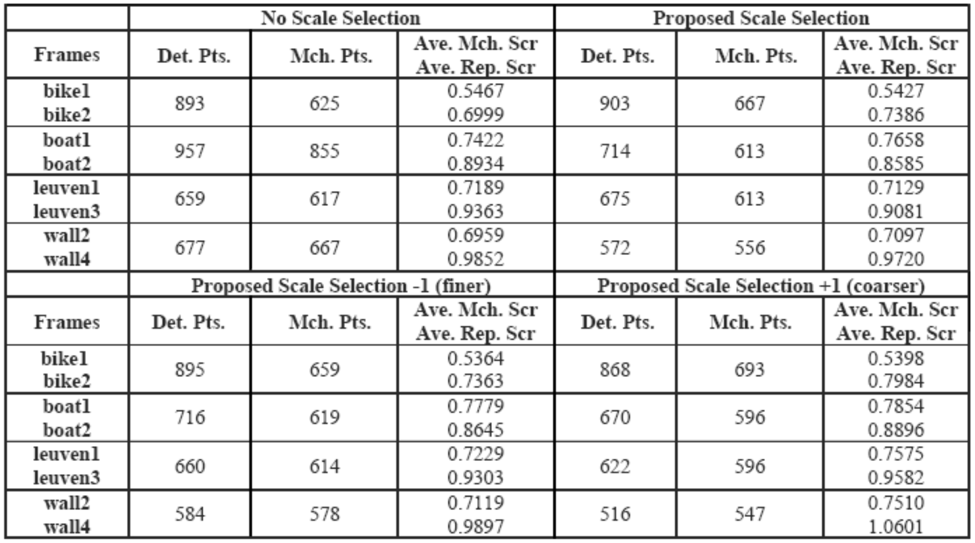
Table 4.4: The number of detected and matched points and the match and repetition scores of the SIFT descriptors for the test image couples

4.3 Evaluation of the Scale Selection Performances

In the evaluation of the proposed scale selection method initially 1500 feature points are detected by using the Harris FPD. Then, they are processed by using four different scale selection schemes. In the first scheme no scale selection is applied in elimination and matching. In the second scheme the proposed scale selection is applied in elimination and matching. In the third scheme the scale which is one scale finer than the selected scale of a feature point is used in elimination and matching. In the fourth scheme the scale which is one scale coarser than the selected scale of a feature point is used in elimination and matching.

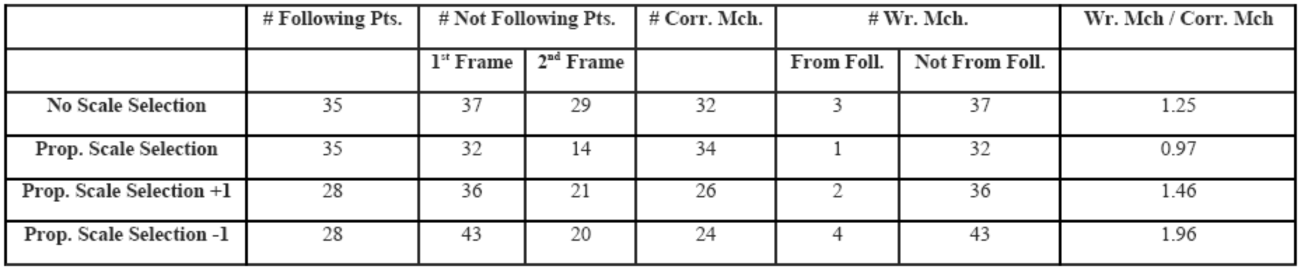
Moreover, to have a better insight the detected points are analyzed for a patch from the "bike" image couple. The number of following points which are detected inside the patch in both of the images, the number of non-following points inside the patches are counted. Finally the correct and wrong matches are counted and their ratio is calculated for each scheme. A set of image patches which are showing the detected and matched points for all of the four schemes are given in Appendix A.2 for "Bike" image couple.

Figure 4.3: A patch from "bike" test image couple.



	No Scale Selection			Proposed Scale Selection		
Frames	Det. Pts.	Mch. Pts.	Ave. Mch. Scr Ave. Rep. Scr	Det. Pts.	Mch. Pts.	Ave. Mch. Scr Ave. Rep. Scr
bike1 bike2	893	625	0.5467 0.6999	903	667	0.5427 0.7386
boat1 boat2	957	855	0.7422 0.8934	714	613	0.7658 0.8585
leuven1 leuven3	659	617	0.7189 0.9363	675	613	0.7129 0.9081
wall2 wall4	677	667	0.6959 0.9852	572	556	0.7097 0.9720
	Proposed Scale Selection -1 (finer)			Proposed Scale Selection +1 (coarser)		
Frames	Det. Pts.	Mch. Pts.	Ave. Mch. Scr Ave. Rep. Scr	Det. Pts.	Mch. Pts.	Ave. Mch. Scr Ave. Rep. Scr
bike1 bike2	895	659	0.5364 0.7363	868	693	0.5398 0.7984
boat1 boat2	716	619	0.7779 0.8645	670	596	0.7854 0.8896
leuven1 leuven3	660	614	0.7229 0.9303	622	596	0.7575 0.9582
wall2 wall4	584	578	0.7119 0.9897	516	547	0.7510 1.0601

Table 4.5: The number of detected and matched points and the match and repetition scores of different scale selection schemes for the test image couples.



	# Following Pts.	# Not Following Pts.		# Corr. Mch.	# Wr. Mch.		Wr. Mch / Corr. Mch
		1st Frame	2nd Frame		From Foll.	Not From Foll.	
No Scale Selection	35	37	29	32	3	37	1.25
Prop. Scale Selection	35	32	14	34	1	32	0.97
Prop. Scale Selection +1	28	36	21	26	2	36	1.46
Prop. Scale Selection -1	28	43	20	24	4	43	1.96

Table 4.6: The number of following, not following, correct matches and wrong matches for the image patch given in Figure 4.3.

4.4 Comparison of the Harris FPD and the Proposed FPD

In the comparison of the Harris feature point detector with threshold set to maximum 1500 feature points per frame and the proposed method, three different matching schemes are used. In the first scheme, which is used as a baseline, the feature points detected by the Harris FPD are directly fed into the conventional gray-level matching algorithm and the evaluation scores are calculated. In the second scheme the proposed feature point elimination method is applied before the conventional gray-level patch matching algorithm and then the evaluation scores are calculated. In the third scheme, again the proposed elimination method is utilized but after that instead of the conventional matching method the proposed matching method is applied and the evaluation scores are calculated.

The schemes are evaluated over fourteen monoscopic and four stereoscopic video sequences. The video sequences are selected according to the motion types (static background, camera tilt/pan/zoom/slide, multiple object motion, etc.), capturing device types (raster scan or progressive scan), level of occlusions etc. The properties of each video sequence is given in Table 4.7.

Name of the Video Sequence	Seqeunce Type	Capturing Device	Resolution	Camera Motion	Object Motion	Occlusions
src6_ref_625.avi	Monoscopic	Raster Scan	720×526	Tilt, pan, zoom	Single object, simple motion	Scene Cut
src10_ref_625.avi	Monoscopic	Raster Scan	720×526	Pan	Multiple objects, simple motions	Multiple small occlusions
src13_ref_525.avi	Monoscopic	Raster Scan	720×486	Static	Multiple objects, complex motions	Scene Cut, multiple small occlusions
src19_ref_525.avi	Monoscopic	Raster Scan	720×486	Tilt, pan, zoom	Multiple objects, complex motions	Multiple large occlusions
src20_ref_525.avi	Monoscopic	Raster Scan	720×486	Static	Multiple objects, simple motions	Multiple small occlusions
src22_ref_525.avi	Monoscopic	Raster Scan	720×486	Zoom	Multiple objects, complex motions	Multiple small occlusions
garden.avi	Monoscopic	Raster Scan	352×240	Sliding left	No object Motion	Multiple small occlusions
foreman.avi	Monoscopic	Raster Scan	352×288	Tilt and pan	Single object, simple motion	Small occlusion
container.avi	Monoscopic	Raster Scan	352×288	Static	Multiple objects, simple motions	Small occlusion
coastguard.avi	Monoscopic	Raster Scan	352×288	Arbitrary slide	Multiple objects, simple motions	Small occlusion
whaleshark_planetEarth_eps11.avi	Monoscopic	Progressive Scan	640×352	Pan	Multiple objects, complex motions	Multiple large occlusions
leopard_planetEarth_eps2.avi	Monoscopic	Progressive Scan	640×352	Pan	Single object, simple motion	Large occlusions
goats_planetEarth_eps5.avi	Monoscopic	Progressive Scan	640×352	Tilt	Multiple objects, simple motions	Multiple small occlusions
dolphis_planetEarth_eps9.avi	Monoscopic	Progressive Scan	640×352	Tilt and pan	Multiple objects, complex motions	Multiple large occlusions
Janine1_X (Stereoscopic)	Stereoscopic	Progressive Scan	1024×768	Static	Multiple objects, simple motions	Multiple small occlusions
Jungle_X.avi (Stereoscopic)	Stereoscopic	Progressive Scan	1024×768	Static	Multiple objects, simple motions	Multiple small occlusions
camX_capture5_Deniz.avi (Stereoscopic)	Stereoscopic	Progressive Scan	640×480	Static	Single object, complex motion	Multiple small occlusions
camX_capture7_novice_jugglers.avi (Stereoscopic)	Stereoscopic	Progressive Scan	640×480	Static	Multiple objects, copmlex motions	Multiple large occlusions

Table 4.7: The properties of video sequences.

In the monoscopic video sequences the detected points on a frame is matched with the detected points on the following frame. In the stereoscopic video sequences the detected points in a frame of a monoscopic subsequence is matched with the detected points on the same frame in the other monoscopic subsequence. And the scores are calculated accordingly.

A set of sample matches is given for each of the schemes by using the first two frames of "garden_sif.yuv" sequence which is taken from University of California, Berkeley Multimedia Research Center. In this sequence the camera is sliding from left to right. As a result of this there is a motion from right to left whose rate changes according to the distance of physical objects to the camera.

In Figures 4.4, 4.7 and 4.10 the detected points according to the first, second and third schemes are shown, respectively. In these figures the green and red "+" signs are used for marking the location of the feature points detected in the first frame and the second frame, respectively.

In Figures 4.5, 4.8 and 4.11 the detected points according to the first, second and third schemes are shown, respectively. In these figures the green and red "+" signs are used for marking the location of the feature points detected in the first frame and the second frame, respectively. And the yellow arrows are used for showing the motion vectors calculated from point matches.

In Figures 4.6, 4.9 and 4.12 some of the detected points according to the first, second and third schemes are shown in detail, respectively. In these figures the green and red "+" signs are used for marking the location of the feature points detected in the first frame and the second frame respectively. And the yellow arrows are used for showing the motion vectors calculated from point matches.

The results of each of the video sequences for every scheme are illustrated in a set of two figures. The first figure comprises the plots of number of detected feature points versus frame number and number of matched feature points versus frame number. The second figure shows the plot of repetition and match scores versus frame number. Furthermore, there is also a set of two figures for each sequence to show the difference between each of the schemes in a single plot. In the first plot the difference between repetition score of the schemes with elimination and without elimination are plotted. In the second plot the difference between the match score of the first, second and third schemes are plotted. In both of the figures the dashed line at $y = 0$ shows the base value which is the first scheme and other schemes are plotted with their differences to the baseline. The the repetition score differences are calculated as

$$\triangle^{rep}_{second\backslash third\,scheme}(frame) = r^{rep_{second\backslash third}}_{frame} - r^{rep_{first}}_{frame}, \tag{4.3}$$

$$r^{rep_x}_{frame} = \frac{1}{N}\sum_{i=1}^{N} r^{rep_x}_{i}. \tag{4.4}$$

In Equation 4.4 N is the number of matched point couples in a $frame$ and x subscript denotes the used scheme.

The match score differences are calculated as

$$\triangle^{match}_{second\,scheme}(frame) = r^{match_{second}}_{frame} - r^{match_{first}}_{frame}, \tag{4.5}$$

$$\triangle^{match}_{third\,scheme}(frame) = r^{match_{third}}_{frame} - r^{match_{first}}_{frame}, \tag{4.6}$$

$$r_{frame}^{match_x} = \frac{1}{N} \sum_{i=1}^{N} r_i^{match_x}. \tag{4.7}$$

In Equation 4.7 N is the number of matched point couples in a $frame$ and x subscript denotes the used scheme.

In the evaluation, the average of the repetition scores over the feature points on the entire frame is taken as the repetition score of the frame and the average of the match scores over the feature points on the entire frame is taken as the match score of the frame.

For “garden.avi” video sequence the analysis results are given in Figures between 4.13 and 4.20. The analysis results of the remaining video sequence are given in Appendix B.

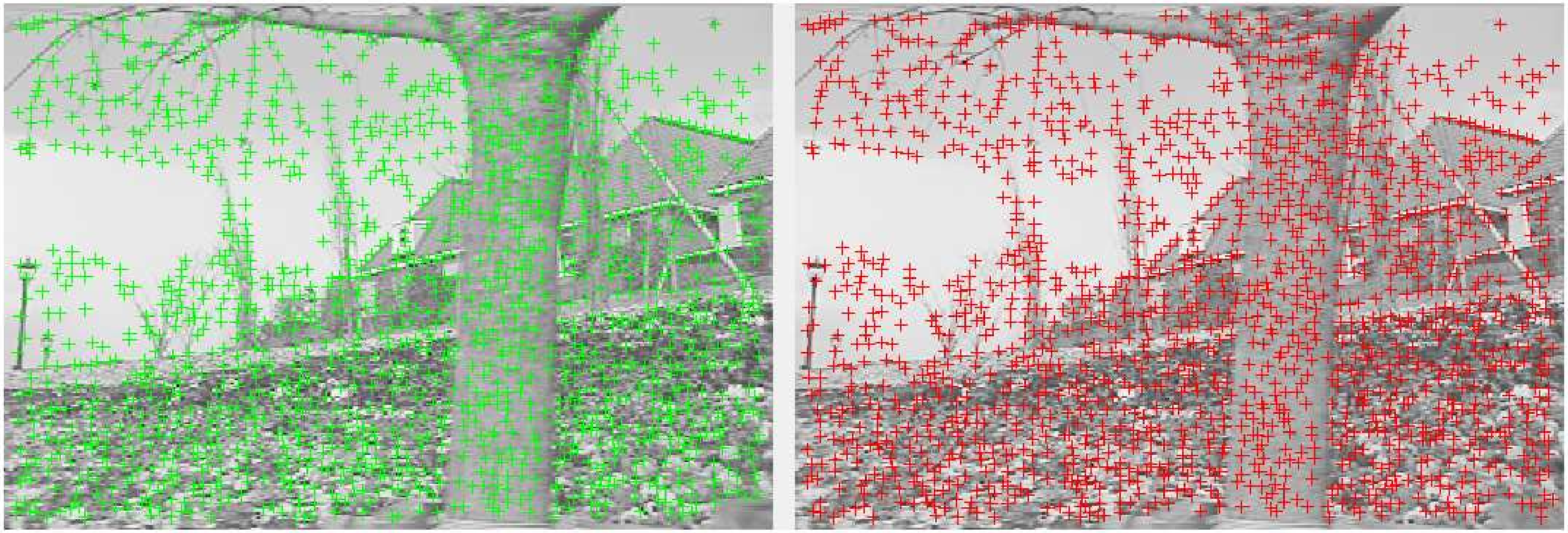


Figure 4.4: The detected feature points from the first and second frames of "garden.avi" by using the first scheme. The original garden_sif.yuv sequence is taken from University of California Berkeley Multimedia Research Center. The green and red "+" signs are used for marking the locations of the feature points detected in the first frame and the second frame, respectively.

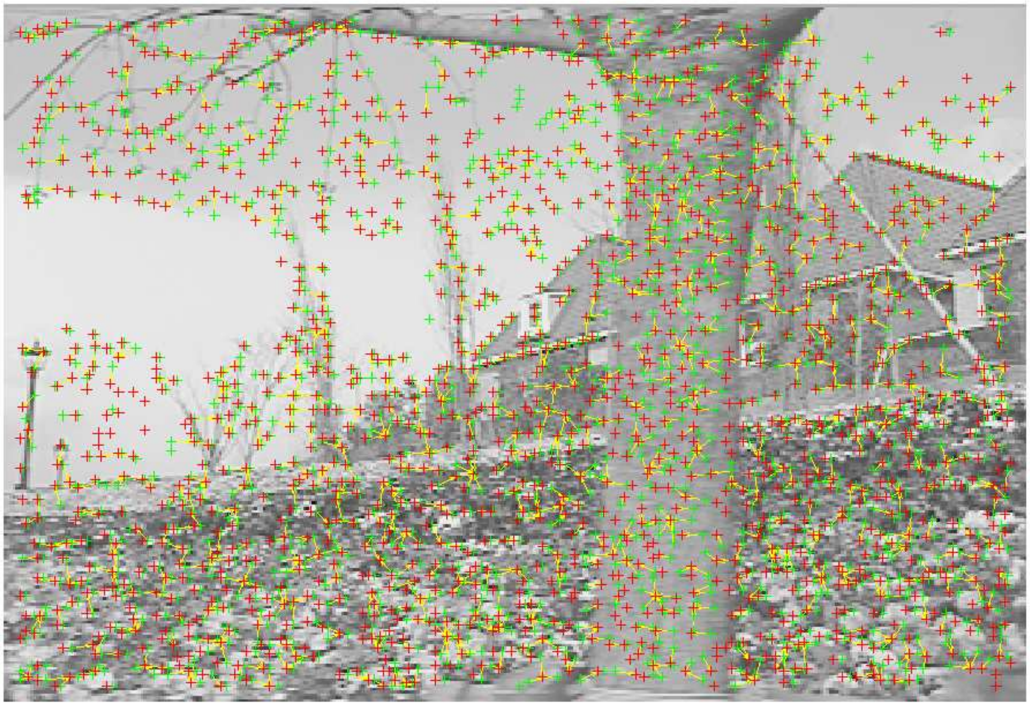


Figure 4.5: The assigned matches for the feature points from the first frame to second frame which are shown in Figure 4.4 by using the first scheme. The green and red "+" signs are used for marking the locations of the feature points detected in the first frame and the second frame, respectively. And the yellow arrows are showing the motion vectors.

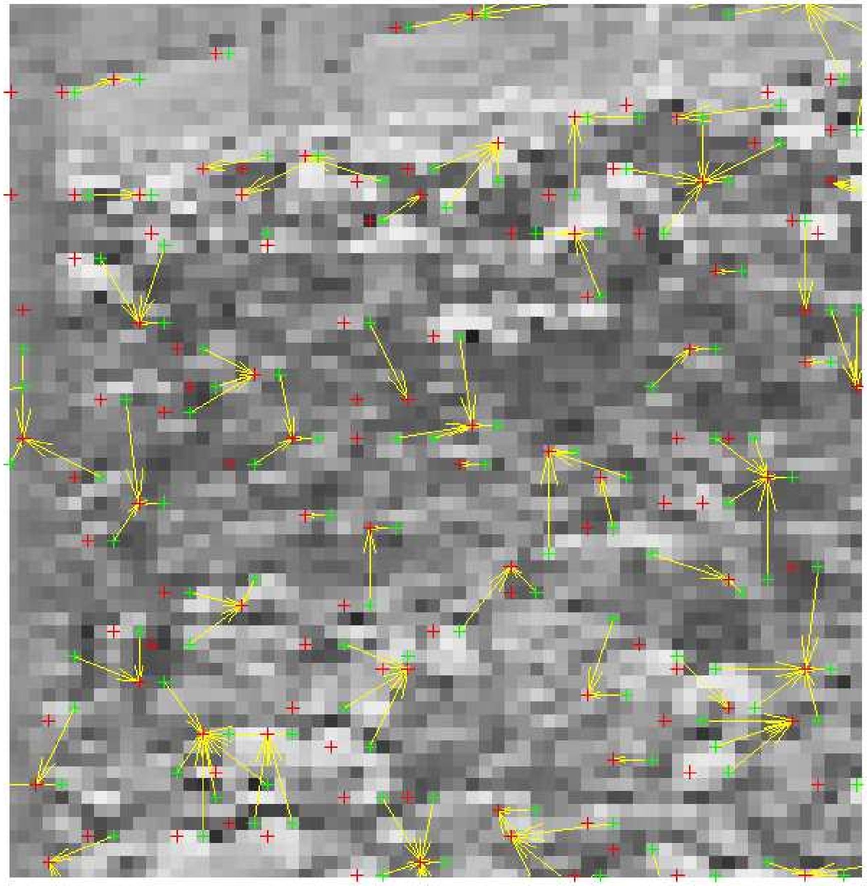


Figure 4.6: A closer look at a set of matched points from the first frame to second frame which are shown in Figure 4.5 by using the first schemes.

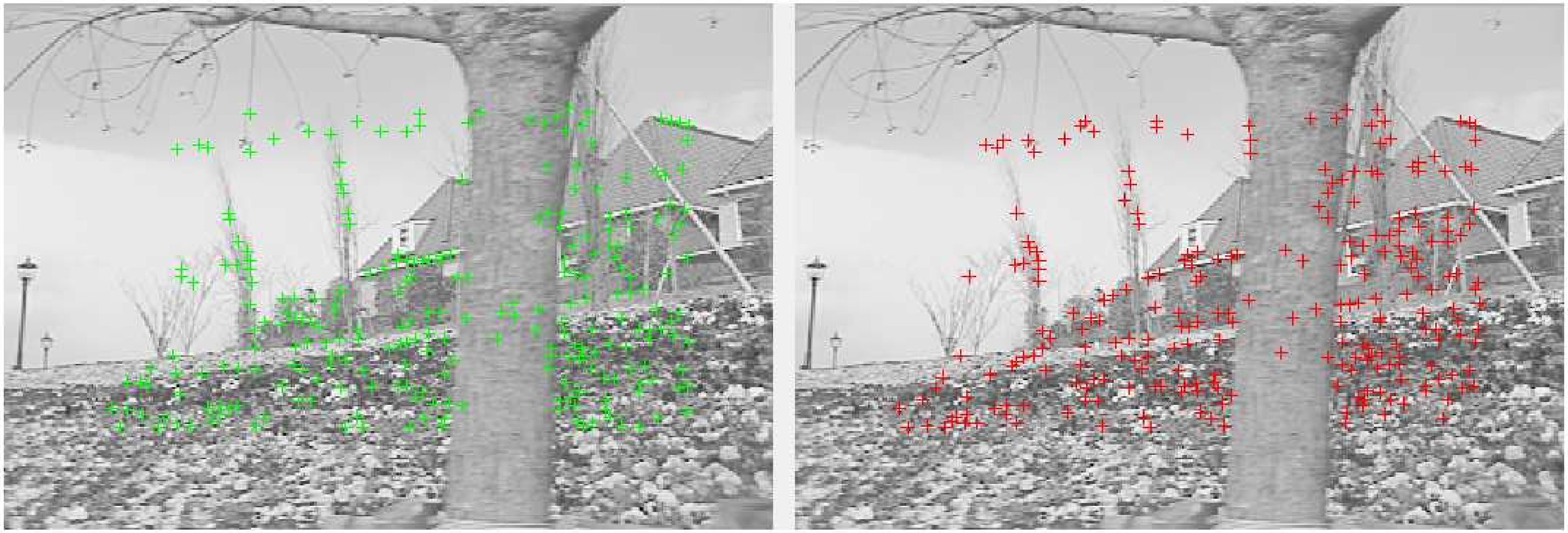


Figure 4.7: The detected feature points from the first and second frames of “garden.avi” by using the second scheme.

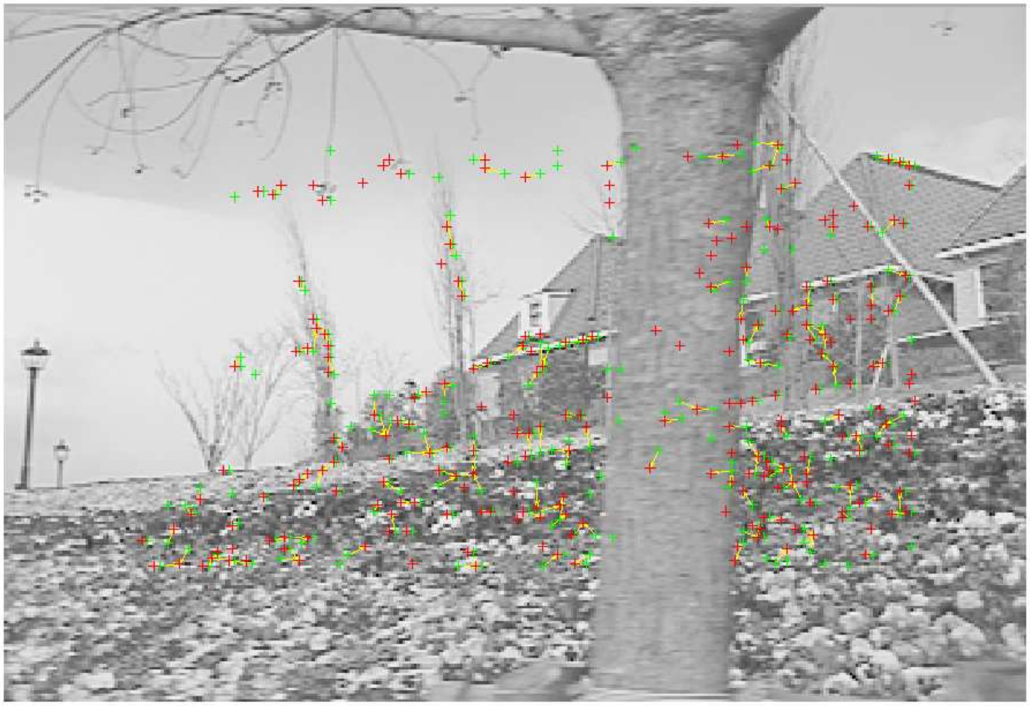


Figure 4.8: The assigned matches for the feature points from the first frame to second frame shown in Figure 4.7 by using the second scheme.

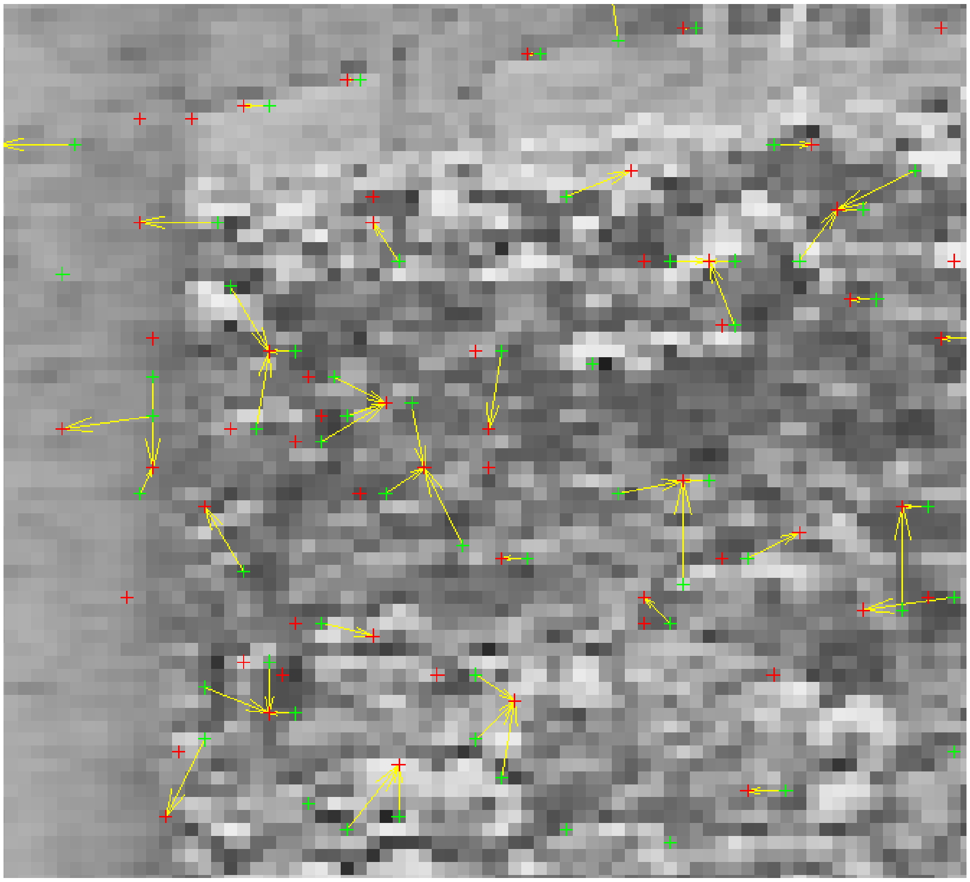


Figure 4.9: A closer look at a set of matched points from the first frame to second frame which are shown in Figure 4.5 by using the second scheme.

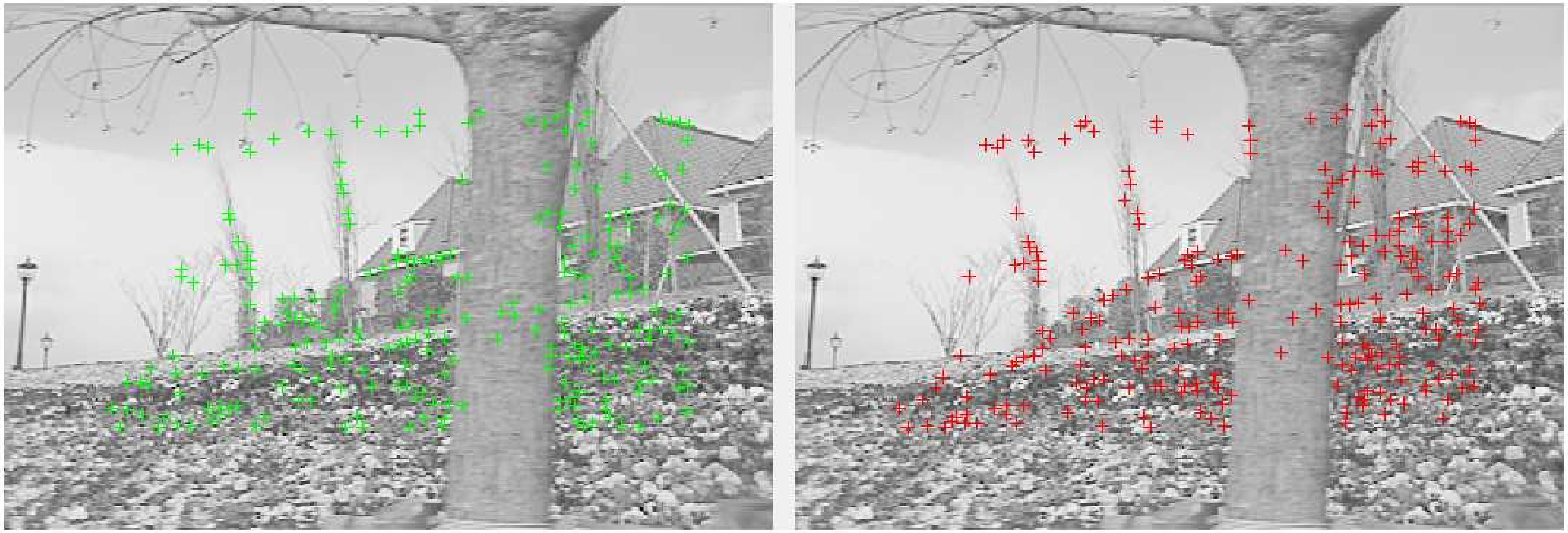


Figure 4.10: The detected feature points from the first and second frames of "garden.avi" by using the third scheme.

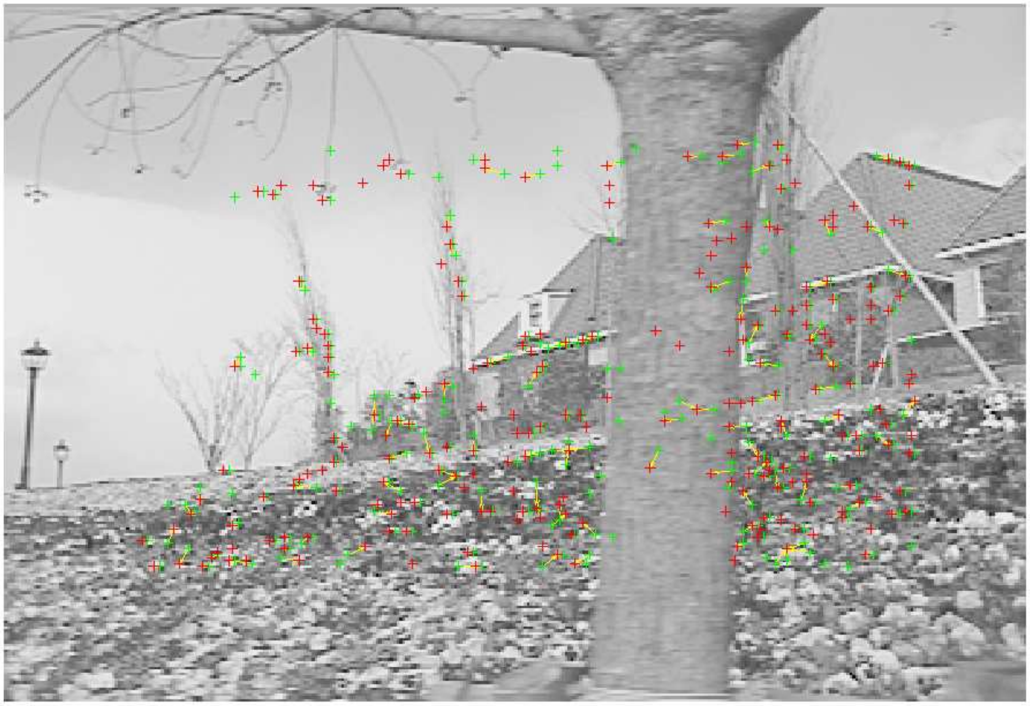


Figure 4.11: The assigned matches for the feature points from the first frame to second frame which are shown in Figure 4.10 by using the third scheme.

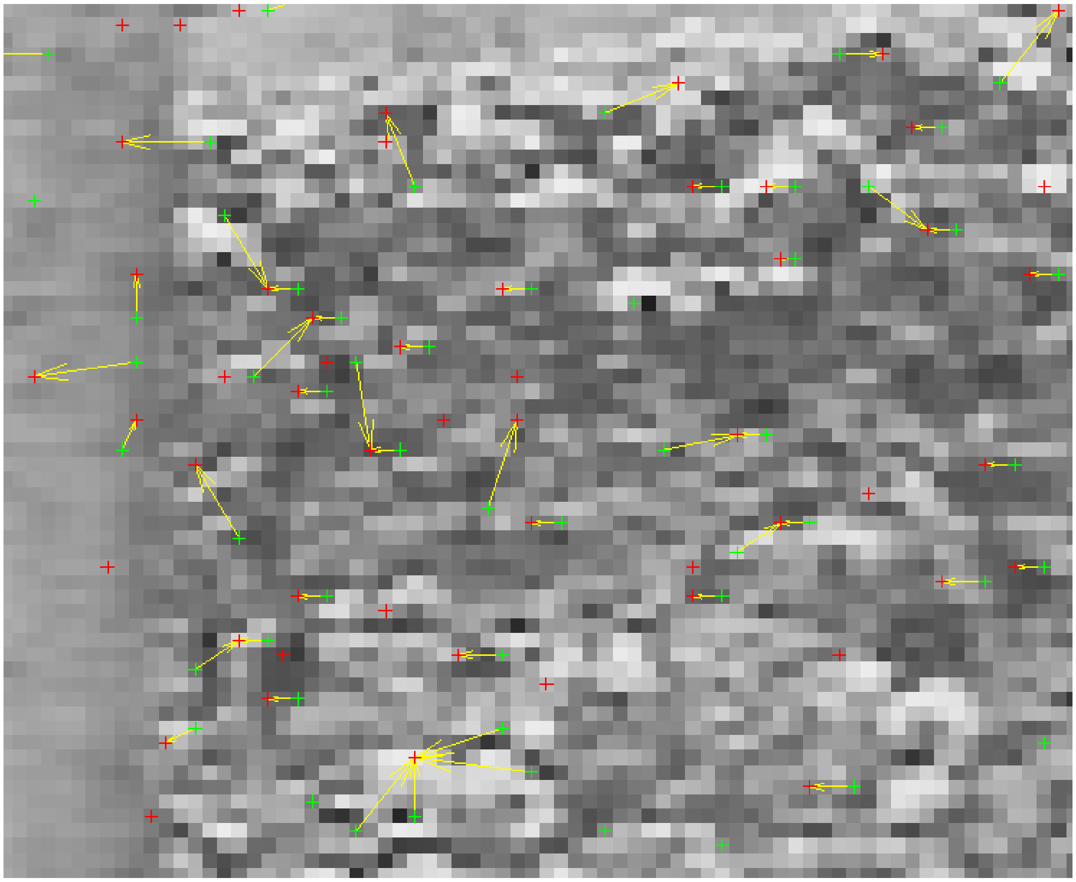


Figure 4.12: A closer look at a set of matched points from the first frame to second frame which are shown in Figure 4.5 by using the third scheme.

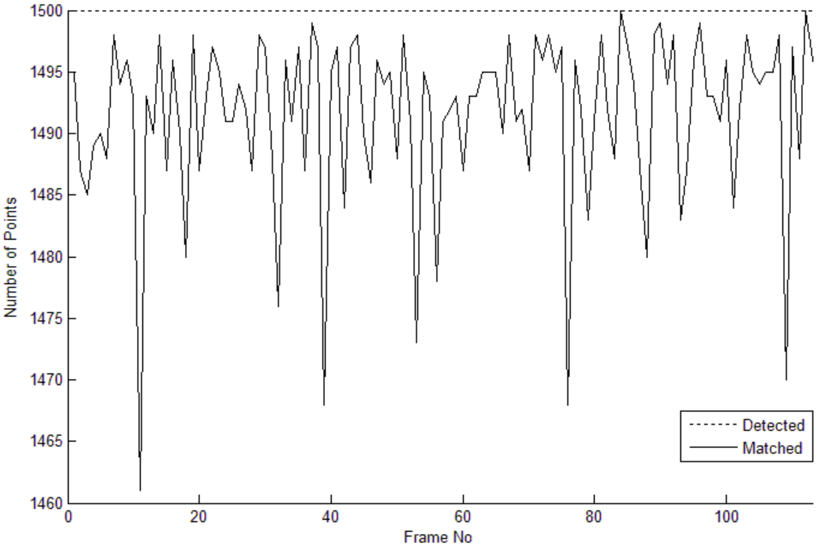


Figure 4.13: The detection and matching results of “garden.avi” for the first scheme.

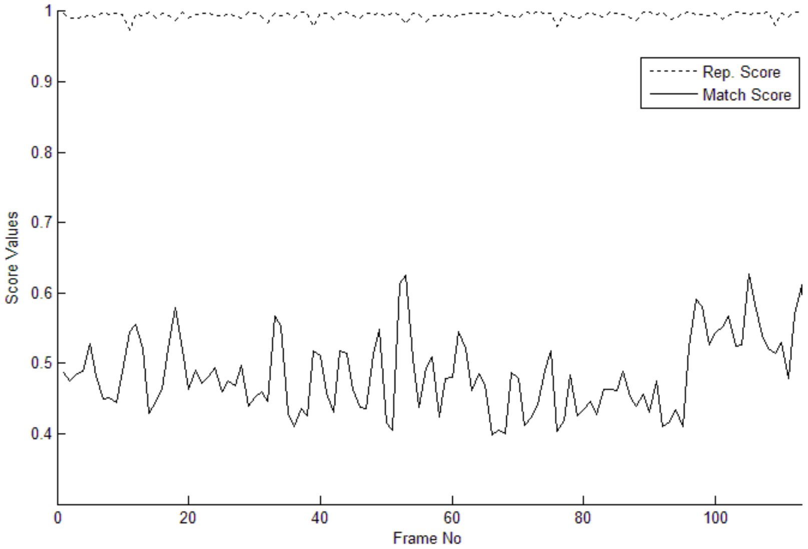


Figure 4.14: The repetability and matching scores of “garden.avi” for the first scheme.

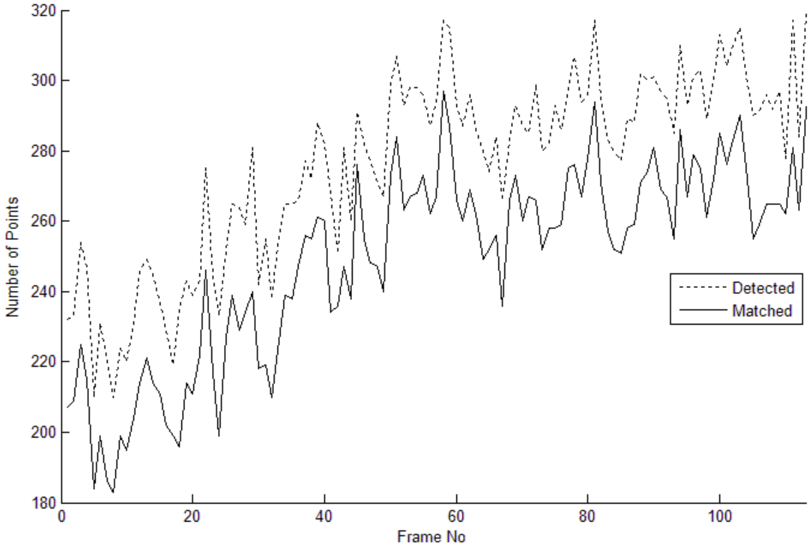


Figure 4.15: The detection and matching results of “garden.avi” for the second scheme.

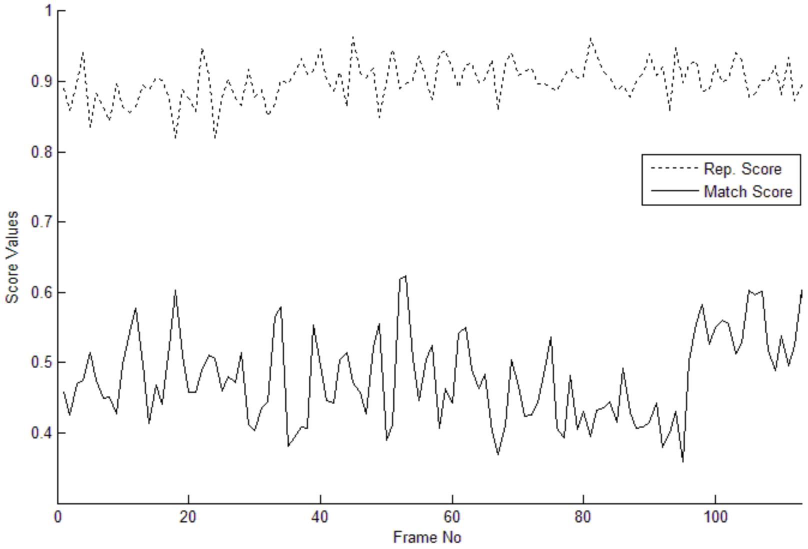


Figure 4.16: The repetability and matching scores of “garden.avi” for the second scheme.

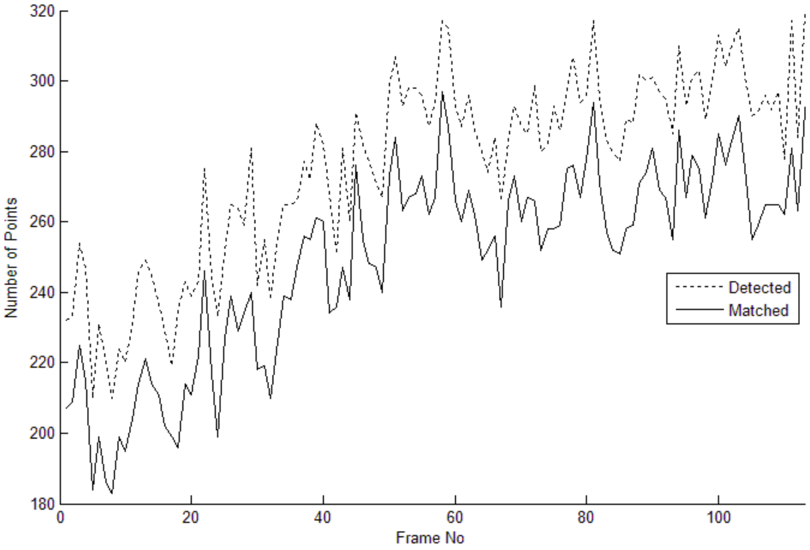


Figure 4.17: The detection and matching results of “garden.avi” for the third scheme.

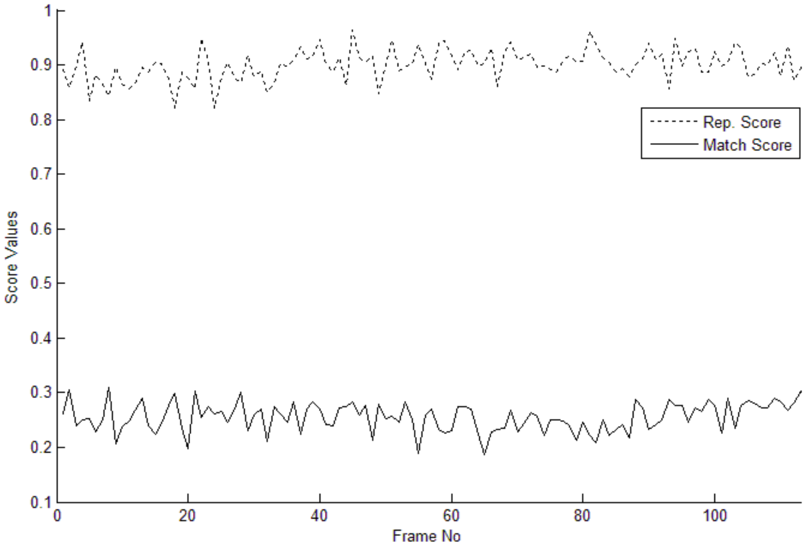


Figure 4.18: The repetability and matching scores of “garden.avi” for the third scheme.

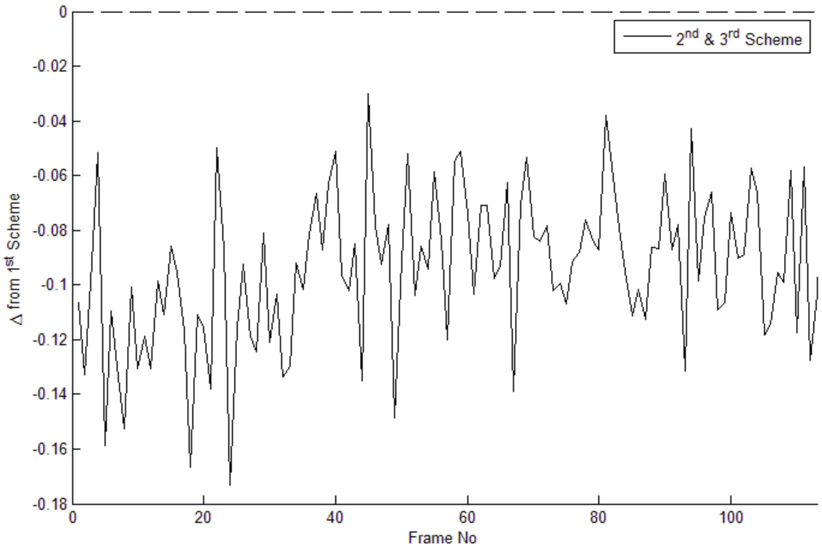


Figure 4.19: The difference between repetability score of 1st scheme and 2nd or 3rd schemes for “garden.avi”.

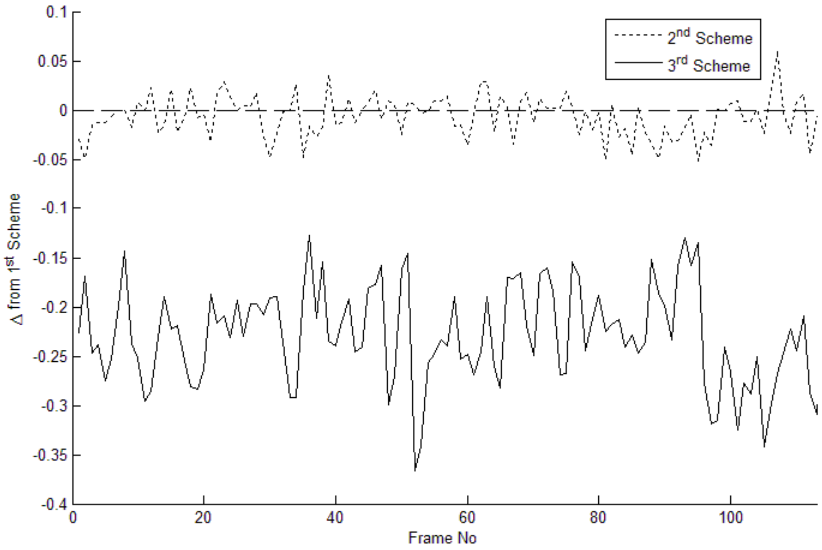


Figure 4.20: The difference between match scores of 1st scheme and 2nd and 3rd schemes for “garden.avi”.

Furthermore, to better visualize the average scores of the schemes over each of the video sequences, the mean scores of the second and the third scheme are given in Table 4.8 with respect to the first scheme for each of the sequences. In Table 4.8 the first column shows the mean matching scores of the first scheme for each of the video sequences. In the second and third columns the ratio of matching scores which are calculated as

$$Ratio\ of\ Matching\ Score = \frac{Mean\ Matching\ Score_{second\ or\ third\ scheme}}{Mean\ Matching\ Score_{first\ scheme}}, \tag{4.8}$$

$$Mean\ Matching\ Score = \frac{1}{N}\sum_{i=1}^{N} r_{frame}^{match_{mean}}, \tag{4.9}$$

$$r_{frame}^{match_{mean}} = \frac{1}{M}\sum_{i=1}^{M} r_{i}^{match}. \tag{4.10}$$

are given. In Equation 4.9 and 4.10, $r_{frame}^{match_{mean}}$ and r_{i}^{match} denotes the mean match score for a specific $frame$ and for a specific matched point couple, respectively. N and M are the total number of frames and the total number of matched point couples, respectively.

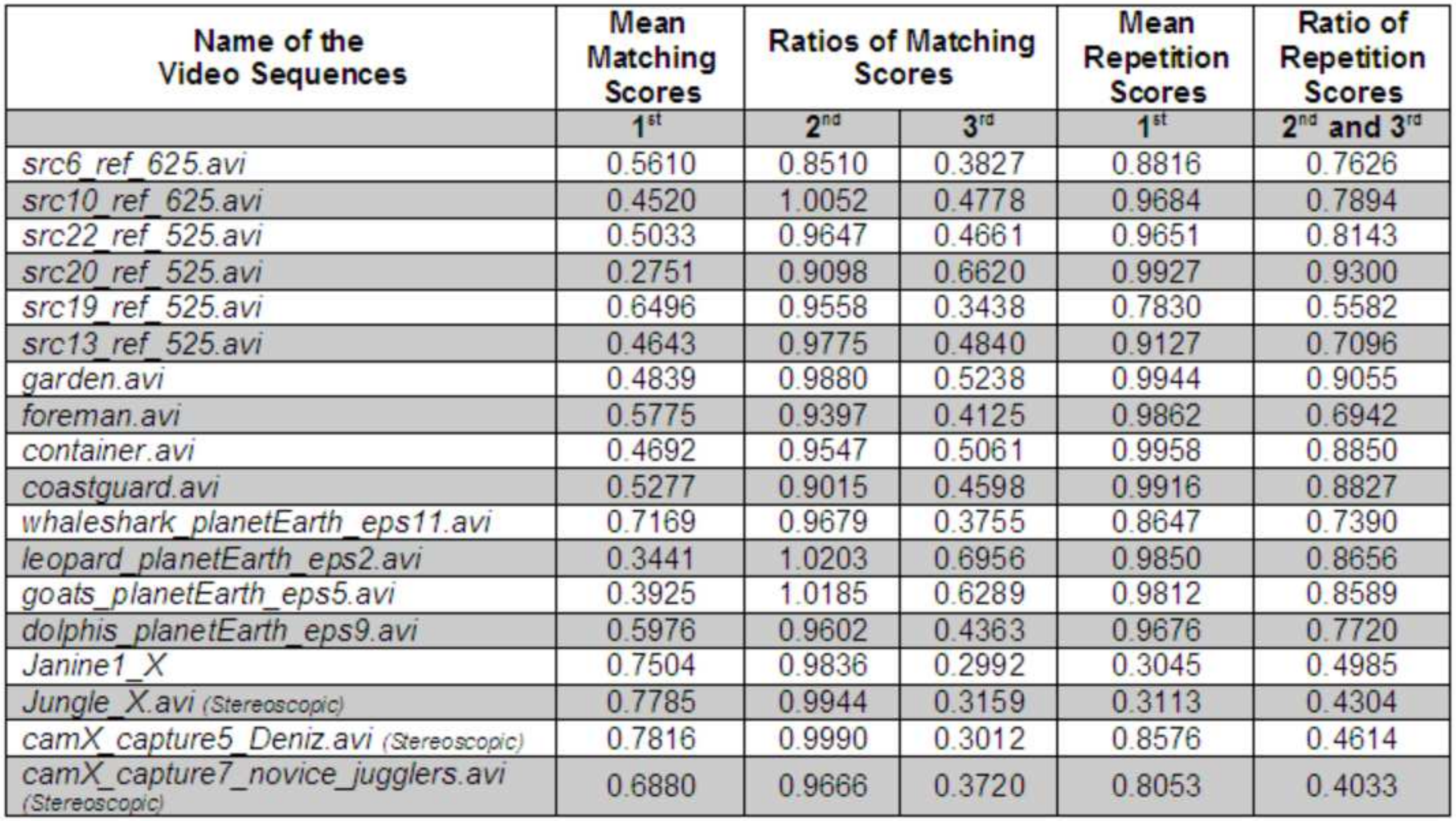
In the forth column of the table the repetition scores of sequences according to the first scheme are given. And at the final column, the ratio of repetition scores are given as a ratio of mean matching score of the second or third scheme by the first scheme. The ratio is calculated as

$$Ratio\ of\ Repetition\ Score = \frac{Mean\ Repetition\ Score_{second\ or\ third\ scheme}}{Mean\ Repetition\ Score_{first\ scheme}}. \tag{4.11}$$

$$Mean\ Repetition\ Score = \frac{1}{N}\sum_{i=1}^{N} r_{frame}^{rep_{mean}}. \tag{4.12}$$

$$r_{frame}^{rep_{mean}} = \frac{1}{M}\sum_{i=1}^{M} r_{i}^{rep}. \tag{4.13}$$

In Equation 4.12 and 4.13, $r_{frame}^{rep_{mean}}$ and r_i^{rep} denotes the mean repetition score for a specific $frame$ and for a specific matched point couple, respectively. N and M are the total number of frames and the total number of matched point couples, respectively.



Name of the Video Sequences	Mean Matching Scores	Ratios of Matching Scores		Mean Repetition Scores	Ratio of Repetition Scores
	1st	2nd	3rd	1st	2nd and 3rd
src6_ref_625.avi	0.5610	0.8510	0.3827	0.8816	0.7626
src10_ref_625.avi	0.4520	1.0052	0.4778	0.9684	0.7894
src22_ref_525.avi	0.5033	0.9647	0.4661	0.9651	0.8143
src20_ref_525.avi	0.2751	0.9098	0.6620	0.9927	0.9300
src19_ref_525.avi	0.6496	0.9558	0.3438	0.7830	0.5582
src13_ref_525.avi	0.4643	0.9775	0.4840	0.9127	0.7096
garden.avi	0.4839	0.9880	0.5238	0.9944	0.9055
foreman.avi	0.5775	0.9397	0.4125	0.9862	0.6942
container.avi	0.4692	0.9547	0.5061	0.9958	0.8850
coastguard.avi	0.5277	0.9015	0.4598	0.9916	0.8827
whaleshark_planetEarth_eps11.avi	0.7169	0.9679	0.3755	0.8647	0.7390
leopard_planetEarth_eps2.avi	0.3441	1.0203	0.6956	0.9850	0.8656
goats_planetEarth_eps5.avi	0.3925	1.0185	0.6289	0.9812	0.8589
dolphis_planetEarth_eps9.avi	0.5976	0.9602	0.4363	0.9676	0.7720
Janine1_X	0.7504	0.9836	0.2992	0.3045	0.4985
Jungle_X.avi (Stereoscopic)	0.7785	0.9944	0.3159	0.3113	0.4304
camX_capture5_Deniz.avi (Stereoscopic)	0.7816	0.9990	0.3012	0.8576	0.4614
camX_capture7_novice_jugglers.avi (Stereoscopic)	0.6880	0.9666	0.3720	0.8053	0.4033

Table 4.8: The matching and repetition scores for the schemes.

4.5 Analysis of Detector Performances

In Table 4.1 and 4.2 the scores for three of the detectors are given. According to the score a clear distinction among there of the detectors seems not possible. Their match scores are similar to each other. And, their repetition scores are different due to the differences in number of detected and matched points.

To evaluate the performance we should have a closer look at the images. In Figures between A.1 and A.6 the detected points inside a patch from "bike" image couple is given for three of the detectors.

In Figures A.1 and A.2 the detected and matches points for the Harris FPD are given. Though all of the points are matched, more than half of them are wrong matches. If Figure A.1 is inspected, it can be seen that only 8 points could be matched correctly. Because, only 8 points were detected in both of the images. And the remaining 9 matches are going to be incorrect.

In Figures A.3 and A.4 the detected and matches points for the proposed FPD are given. In these images 5 points are detected in both of the images. Hence, there can be maximum 5 correct matches and there can be minimum 5 wrong matches due to the non-repeating points.

In Figures A.5 and A.6 the detected and matches points for the SIFT FPD are given. In these images we can see 3 points are detected in both of the images. Hence, there can be maximum 3 correct matches and there can be minimum 12 wrong matches due to the non-repeating points.

According to the tables and figures a distinction between three of the detectors is not apparent. The proposed detector without the proposed descriptor is almost same as the others. However, according to the repetition scores the Harris FPD makes less number of wrong matches when gray-level patch descriptor is used.

4.6 Analysis of Descriptor Performances

In Table 4.3 the scores for three of the descriptors are given. According to the results the proposed descriptor makes better matches for the points detected by the proposed detector.

If we check the repetition scores of the gray-level patch and proposed descriptors, the gray-level patch descriptor has a repetition score greater than 1 which means that there are definite wrong matches. On the other hand, the repetition score of the proposed descriptor is less than 1.

Moreover, the match score of the proposed descriptor is smaller than the gray-level patch descriptor. The smaller match score is a sign of better point description which results in better matching results.

However, more interestingly, SIFT descriptor rejects almost one-fifth of the detected points detected and makes maximum 17 number of matches. According to the results we can say that the Harris detector detects unreliable points according to the SIFT descriptor and most of the points are rejected. Moreover, among the remaining points only a few of them are assigned as valid matches.

To check the detection-matching scheme of the SIFT detection and matching method, only the SIFT detection-description scheme is applied to the test image couples. The results are given in Table 4.4. According to the tables SIFT detector-descriptor achieves the best scores among the tested methods. We should also mention that the detectors should be used with their specific descriptor.

4.7 Analysis of Scale Selection Performances

According to the scores in Table 4.5, it is hard to make a performance distinction between each of the selection scheme. Because, as a result of a change in description, the number of detected and matched points change from one scale selection scheme to another. However, if we examine Table 4.6 which shows the point detection and matching performance of the scale selection scheme for the image patch which is shown in Figure 4.3, we can say that by using the proposed scale selection method the matching performance increases by 25% for the given test image set on average.

4.8 Analysis of Detection-Matching Performances

In Figure 4.4 feature points detected with the first scheme are given. In this scheme, the threshold is set to give 1500 feature points. The green and red "+" signs are used for marking the locations of the feature points detected in the first frame and the second frame, respectively, by using the first scheme. As it can be seen in the frames, there are many non-distinctive feature points which are in the textured regions of the frames. For example, the points in the sky, which has a simple texture, are hard to discriminate from each other as the ones on the flower field, which has a complicated texture.

The detected points and motion vectors are shown in Figure 4.5 with green, red "+" signs and yellow arrows. It can be seen that though the most of the points are matched, the match vectors are highly contradicting with each other. Hence, intuitively we can say that most of the matches made by the first scheme are highly unreliable.

If we have a closer look at the matches, we can verify the previous observation. As it is seen in Figure 4.6, most of the motion vectors are pointing to different directions. There is no regularity in vectors, although such regularity is expected, due to the content of “garden.avi” sequence.

In Figure 4.7, we can see the remaining 250 points after elimination of the points. As it is seen, most of the non-distinctive points are eliminated. For example, the points on the sky and most of the points on the flower field are eliminated. However, the points in the sky-branch transition region and some of the points, which have low occurrence expectancy, in the flower field are retained.

As it can be seen in Figure 4.8 though the elimination seems successful, due to the conventional matching scheme there are still unreliable matches between the frame couples. This observation can also be verified by having a closer look at the matches which are shown in Figure 4.9.

In Figure 4.10 we can see the same remaining 250 points which are previously shown in Figure 4.7. In Figures 4.11 and 4.12, we can see there are less number of mismatches due to the introduction of regional descriptors and we can say that the matching performance and reliability is increased.

Moreover, the intuitive results can be verified by examining Figures between 4.13 and 4.20. In Figures 4.13, 4.15 and 4.17, we can see the detected points, matched points and the evaluation scores over the entire “garden.avi” sequence by using the first, second and third schemes. In these plots we can see that the number of matches of the first scheme almost three times higher that the other schemes, due to the applied elimination method which discards two-thirds of the detected points. Furthermore, due to the number of detected and matched points in the first scheme is higher than the second and the third schemes, the repetition score for the first scheme is also higher that the other two as shown in Figures 4.14, 4.16 and 4.18.

Though the repetition score of the first scheme is higher than the others, the quality of matches are poor when compared with the other two. If we examine the plots shown in Figures 4.14, 4.16 and 4.18 the quality of the matches in the first scheme is almost half of the third scheme. This situation was also shown in the individual frame couple analysis which are shown in Figures 4.5, 4.8 and 4.11. According to the figures and tables best match quality is achieved by the third scheme.

Moreover, we can also verify these observation by using comparative score plots, which are shown in Figures 4.19 and 4.20 for match score and repetition scores. In these figures we can see that due to the elimination the repetition score of the first scheme is higher than the other two. However, the match score of the third scheme is always better than the first and second schemes that use conventional matching algorithms.

The same results are also obtained in Figures between B.1 and B.136 for a variety of sequences.

Furthermore, if we examine the average results which are given in Table 4.8 for each sequence, we can verify the previous findings. In the table, the best average matching is always made by the third scheme. The matching quality changes between 30% and 70% of the first scheme for the given video sequence set according to match results in Table 4.8. This observation also shows that when conventional methods are used the match quality is unreliable in most of the cases, though it has high repetition score.

Chapter 5

Conclusions

In this thesis a new post-processing method is studied to improve the reliability and robustness of the conventional feature point detection and matching schemes, independent of the a priori used feature point detection method.

The proposed method is based on the idea that the feature points should be distinctive among their neighbors and the distinctiveness of a point can be measured by using regional attributes of a point. However, due to the computational complexity of extracting regional attributes, it is computationally overwhelming to extract regional attributes of every single pixel in an image. Instead of that the feature points, detected by conventional detectors, are counted as candidate feature points and they are eliminated according to their regional attributes. Due to this, the computational load is decreased to an acceptable level.

To find out the distinctiveness of a point, first of all a scale is selected from a neighborhood scale-space for the point. After that a patch is extracted around the point in the selected scale and binarized. The regional attributes of the point are calculated by using a binary GMRF model. According to the model parameters occurrence expectancy (distinctiveness) of the point is calculated.

The points with high occurrence expectancies are eliminated and the remaining points are sorted according to their distinctiveness. In the final step, the points are matched starting from the point with the lowest occurrence expectancy to the one with the highest occurrence expectancy by using averaged normalized parameters of pixel-wise and regional point descriptors.

According to the detector and descriptor performance evaluation results, the SIFT detector-descriptor combination performs better than the Harris detector gray-level descriptor scheme and proposed detector-descriptor scheme. Moreover, we should also add that each of the descriptors can perform better with a specific descriptor. For example, if the Harris FPD is used in detection, the gray-level patches should be used in description. Moreover, if the proposed detector or SIFT is used in detection than the proposed or SIFT descriptor should be used, respectively.

Moreover, if the scale selection results are analyzed in Tables 4.5 and 4.6, the proposed scale selection scheme performs better than the one with no scale selection or with disturbed scale selection.

According to the comparison results which are shown in Table 4.8 and Figures between 4.4 and B.136, it can be seen that the proposed method increases both the matching reliability and robustness for the given video sequence set. According to the Table 4.8, the average match score of the third scheme is between 30% and 70% of the first scheme for the given video seqeunces. Hence, according to the ratings the match quality of the third scheme is expected to be better than the first scheme. The same conclusion can also be derived by checking the sequence results, which are shown in Figures between B.1 and B.136. In these figures the match scores of the third scheme is better than the first and second schemes in most of the cases. It can also be verified intuitively with the given set of sample matches, which are shown in Figures 4.6, 4.9 and

4.12. In these figures the third scheme gives the more reliable matches than the others.

The match score of the third scheme is better than the first and the second schemes. Because, the neighborhood structure features are included with the new descriptors, so, the detected points are described more distinctively. However, the associated repetition scores are not good in variety of cases. Because, one third of the detected points are eliminated on the third scheme. As a result of the elimination, the repetition scores are decreased.

If we examine the repetition scores, we can say that the first scheme is better than the others. However, the repetition score does not measure the goodness of match. As it measure the number of matched points not the quality, totally wrong matches can also increase this measure. To check the goodness of match we have the examine the match scores. According to the experiments, the match scores of the first scheme is slightly less than the third scheme. This observation indicates that tough the first scheme makes more matches, it makes more wrong matches than the third scheme, according to the experiments.

Bibliography

[1] J. Cooper, S. Venkatesh and L. Kitchen, “The dissimilarity corner detector”, IEEE Proceedings on Image Processing, pp. 1377-1382, 1991.

[2] J. Cooper, S. Venkatesh, L. Kitchen, “Early Jump-Out Corner Detectors”, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 15, pp. 823-828, 1993.

[3] B.K. Ray and K.S. Ray, “Scale-Space Analysis and Corner Detection on Digital Curves Using a Discrete Scale-Space Kernel”, Pattern Recognition, Vol. 30 , No. 9, pp. 1463-1474, September 1997.

[4] F. Mokhtarian and R. Suomela, “Robust Image Corner Detection through Curvature Scale Space”, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 20, No. 12, pp. 1376-1381, 1998.

[5] H.T. Sheu and W.C. Hu, “A Rotationally Invariant Two-Phase Scheme for Corner Detection”, Pattern Recognition, Vol. 29, pp. 819-828, 1996.

[6] D. M. Tsai, “Boundary-based corner detection using neural networks”, Pattern Recognition, Vol. 30, No. 1, pp. 85-97, 1997.

[7] Hyun-Hwa Oh , Sung-Il Chien, “Exact corner location using attentional generalized symmetry transform”, Pattern Recognition Letters, Vol. 23 No. 11, pp. 1361-1372, September 2002.

[8] F. Arrebola, P. Camacho, A. Bandera and F. Sandoval, “Corner detection and curve representation by circular histograms of contour chain code”, Electronics Letters, Vol. 35, No. 13, pp. 1065-1067, 1999.

[9] F. Arrebola, F. S. Hernndez, “Corner detection and curve segmentation by multiresolution chain-code linking”, Pattern Recognition Vol. 38, No. 10, pp. 1596-1614, 2005.

[10] D. Tsai, H. Hou and H. Su, “Boundary-based corner detection using eigenvalues of covariance matrices”, Pattern Recogn. Lett., Vol. 20, pp. 31-40, Jan. 1999.

[11] D.S. Guru and R. Dinesh and P. Nagabhushan, “Boundary based corner detection and localization using new ’cornerity’ index: a robust approach”, CRV04, pp. 417-423, 2004.

[12] X. Zhang, R.M. Haralick, R. Ramesh, and A.S. Bedekar, “A Bayesian Corner Detector: Theory and Performance Evaluation”, IUE Workshop, 1995.

[13] E.R. Davies, “Application of generalized Hough transform to corner detection”, IEE-PE : 135, No. 1, pp. 49-54, 1988.

[14] H. Moravec, “Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover”, Tech. report CMU-RI-TR-80-03, Robotics Institute, Carnegie Mellon University & doctoral dissertation, Stanford University, September, 1980.

[15] L. Kitchen and A. Rosenfeld, “Gray-level corner detection”, Pattern Recognition Letters, Vol. 1, pp. 95-102, 1982.

[16] C.G. Harris and M. Stephens, “A combined corner and edge detector”, 4th Alvey Vision Conference, pp. 147-151, 1988.

[17] C. Tomasi and T. Kanade, “Detection and tracking of point features”, Technical Report CMU-CS-91-132, Carnegie Mellon University Technical, 1991

[18] C. Achard and E. Bigorgne, “A Sub-Pixel and Multispectral Corner Detector”, In Proceedings of the international Conference on Pattern Recognition, Vol. 3 $September 03 - 08, 2000$, ICPR - IEEE Computer Society, Washington, DC, 3971.

[19] Z. Zheng, H. wang and E. K. Teoh. “Analysis of gray level corner detection”, Pattern Recognition Letters, Vol.20, pp. 149-162, 1999.

[20] S. Ando, “Image field categorization and edge/corner detection from gradient covariance”, IEEE Trans. Pattern Anal. Machine Intell., Vol. 22, No. 2, pp. 179190, 2000.

[21] H. Wang and M. Brady, “A practical solution to corner detection”, Proc International Conference on Image Processing, Vol. 1, pp. 919-923, 1994.

[22] T. Stammberger, M. Michaelis, M. Reiser, and K. Englmeir, “A hierarchical filter scheme for efficient corner detection”, Pattern Recogn. Lett. Vol. 19, No. 8, pp. 687-700, Jun. 1998.

[23] E.Sojka, “A new and efficient algorithm for detecting the corners in digital images”, Proc. 24th DAGM Symposium, Springer, LNCS 2449, Berlin, NY, pp. 125-132, 2002.

[24] B. Telle and M.J. Aldon, “Interest point detection in color images”, MVA’02 : IARP Workshop on Machine Vision and Applications, Nara-Japan, pp.558-563, December 11-13, 2002.

[25] D. G. Lowe, “Object recognition from local scale-invariant features”, in IEEE Int. Conf. Computer Vision, pp. 1150-1157, 1999

[26] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points", Proceedings of the International Conference on Computer Vision, Vancouver, Canada, pp. 525-531, 2001.

[27] T. Lindeberg, "Junction detection with automatic selection of detection scales and localization scales", Proc. First Int. Conf. on Image Processing, Austin, Texas, Vol. 1, pp. 924-928, Nov. 1994.

[28] R. Laganiere, "A morphological operator for corner detection", Pattern Recognition, Vol. 31, No. 11, pp. 1643-1652, 1998.

[29] Rey-Sern Lin , Chyi-Hwa Chu and Yuang-Cheh Hsueh, "A modified morphological corner detector", Pattern Recognition Letters, Vol.19 No.3-4, pp.279-286, March 1998

[30] F. Y. Shih, C. Chuang and V. Gaddipati, "A modified regulated morphological corner detector", Pattern Recogn. Lett., Vol. 26, No. 7, pp.931-937, May. 2005.

[31] L. Wen-Yu, L. Hua and Z. Guang-Xi, "A fast algorithm for corner detection using the morphologic skeleton", Pattern Recognition Letters, Vol. 22, No. 8, pp. 891-900, June 2001.

[32] J.A. Noble, "Images as Functions and Sets", Image and Vision Computing, Vol. 10, No. 1, pp.19-29, January-February 1992.

[33] C.M. Orange and F.C.A Groen, "Model based corner detection", In Proceedings of the Conference on Computer Vision and Pattern Recognition, New York, USA, pp.690-691, June 1993.

[34] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection", Proceedings of European Conference on Computer Vision, Vol.1, pp.430-443, 2006.

[35] X. Xie, R. Sudhakar and H. Zhuang, “Corner Detection by a Cost Minimization Approach”, Pattern Recognition, Vol. 26, No. 8, pp.1235-1243, August 1993.

[36] S.M. Smith and J.M. Brady, “SUSAN - a new approach to low level image processing”, Int. Journal of Computer Vision, Vol.23 No.1, pp.45-78, 1997

[37] M. Trajkovic and M. Hedley, “Fast corner detection”, Image Vision Comput. 16, pp.75-87, 1998.

[38] P. Kovesi, “Phase congruency detects corners and edges”, In DICTA, Sydney, December 2003.

[39] B. Robbins and R. A. Owens, “2D feature detection via local energy”, Image Vis. Computing, Vol. 15, pp.353-368, 1997.

[40] K. Kohlmann., “Corner Detection in Natural Images Based on the 2-D Hilbert Transform”, Signal Processing, Vol.48, pp.225–234, 1996.

[41] A. Pedersini and S. Tubaro, “Multi-Resolution Corner Detection”, Proc. ICIP-2000, Vancuver, Canada, 10-13 September 2000.

[42] YL. Hao, DH. Shen, B. Gong, “Image salient point detection based on Wavelet”, Proceedings of the 2004 International Conference on Intelligent Mechatronics and Automation, International Conference on Intelligent Mechatronics and Automation,AUG 26-31, 2004

[43] K. Rangarajan, M. Shah, and D. Brackle, “Optimal corner detector”, Computer Vision, Graphics and Image Processing, Vol.48, pp.230-245, 1989.

[44] L. Rosenthaler, F. Heitger, O. K ubler, and R. v. d. Heydt, “Detection of general edges and keypoints”, In G. Sandini, editor, European Conference on Computer Vision *ECCV*, LNCS 588, pages 78–86, Santa Margherita Ligure, Italy, May 1992.

[45] P. Tissainayagam and D. Suter, “Assessing the performance of corner detectors for point feature tracking applications”, IVC, Volume 22, Issue 8, Pages 663-679, August 2004.

[46] P.I. Rockett, “Performance assessment of feature detection algorithms: a methodology and case study on corner detectors”, IP, Volume 12, Issue 12, Pages:1668-1676, December 2003.

[47] T. Tuytelaars, “A survey on local invariant features”, ECCV 2006 Tutorial.

[48] J. Shi and C. Tomasi, “Good Features to Track”, 1994 IEEE Conference on Computer Vision and Pattern Recognition CVPR’94, pp. 593 - 600, 1994.

[49] X. Zhang, R.M. Haralick, R. Ramesh, and A.S. Bedekar, “A Bayesian Corner Detector: Theory and Performance Evaluation”, IUE Workshop, 1995.

[50] M. Brown and D. G. Lowe, “Invariant features from interest point groups”, British Machine Vision Conference, BMVC 2002, pp. 656-665, Cardiff.

[51] J. Matas, O. Chum, M. Urban and T. Pajdla, “Robust Wide Baseline Stereo from Maximally Stable Extremal Regions”, British Machine Vision Conference, BMVC 2002, Cardiff.

[52] T. Tuytelaars and L. Van Gool. “Wide baseline stereo matching based on local, affinely invariant regions”. In British Machine Vision Conference BMVC’2000.

[53] A. Baumberg, “Reliable feature matching across widely separated views,” Conference on Computer Vision and Pattern Recognition, Hilton Head, South Carolina, pp. 774781, June 2000.

[54] A. Turina, T. Tuytelaars, T. Moons, and L. Van Gool, “Grouping via the matching of repeated patterns”, In Proc. CAPR, pages 250259, 2001.

[55] H. Tamura, Mori S. and Yamawaki T., “Textural Features Corresponding to Visual Perceptron”, IEEE Trans. on Systems, Man and Cybernetics, Vol.8, No.6, pp.460-473.

[56] J. Sklansky, “Image segmentation and feature extraction”, IEEE Trans. on System, Man and Cybernetics, Vol.8, No.4, pp.237-247, 1978.

[57] R. Haralick, K. Shanmugam and I. Dinstein, “Texture features for image classification”, IEEE Transactions on Systems, Man, and Cybernetics, Vol.3, No.6, pp.610-621, 1973.

[58] V.V. Starovoitov, S.-Y. Jeong and R.-H. Park, “Texture periodicity detection: features, properties, andcomparisons”, IEEE Transactions on Systems, Man and Cybernetics, Part A, Vol.28, No.6, pp.839-849, 1998

[59] K. Selkainaho, J. Parkkinen and E. Oja, “Comparison of and statistics in finding signal χ^2 and κ picture periodicity”, Proc. 9th Int. Conf. on Pattern recognition, pp.1221-1224, 1988.

[60] W.R. Schwartz and H. Pedrini, “Texture classification based on spatial dependence features using co-occurrence matrices and Markov random fields”, 2004 International Conference on Image Processing, Vol. 1, pp.239-242, 2004.

[61] G. Oh, S. Lee and S. Y. Shin, “Fast determination of textural periodicity using distance matching function”, Pattern Recogn. Lett., Vol.20, No.2, 1999.

[62] M. Hauta-Kasari, J. Parkkinen, T. Jaaskelainen and R. Lenz, “Generalized co-occurrence matrix for multispectral texture analysis”, Proceedings of the 13th International Conference on Pattern Recognition, Vol.2, pp.785-789, 1996.

[63] K. Valkealahti and E. Oja, Reduced Multidimensional Cooccurrence Histograms in Texture Classification, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.20, No.1, pp.90-94, 1998.

[64] D. Chetverikov, “Texture Analysis Using Feature Based Pairwise Interaction Maps”, Pattern Recognition, Vol.32, pp.487-502, 1999.

[65] M. Unser, “Sum and difference histograms for texture classification”, IEEE Trans. Pattern Anal. Mach. Intell., Vol.8, No.1, pp.118-125, 1986.

[66] M. Partio, B. Cramariuc, and M. Gabbouj, Texture similarity evaluation using ordinal co-occurrence, in Proceedings of International Conference on Image Processing (ICIP ’04), vol. 3, pp. 15371540, Singapore, October 2004.

[67] H. Yan-fang and S. Peng-fei, ‘Mean shift texture surface detection based on WT and COM feature image selection”, Journal of Zhejiang University - Science A, Vol. 7, No 6, pp.969-975, 2006.

[68] P. De Souza, “Texture recognition via autoregression”, Pattern Recognition, Vol.15, No.6, pp.471-475, 1982.

[69] R. Chellappa and R.L. Kashyap, “Texture Synthesis Using 2-D Noncausal Autoregressive Models”, IEEE Trans. Acoustics, Speech, Signal Processing, Vol.33, No.1, pp. 194-203, 1985.

[70] I. Claude and A. Smolarz, “A new textured image segmentation algorithm by autoregressive modelling and multiscale block classification”, 6th International Conference on Image Processing and its Applications (CP443), pp.586-590, 1997.

[71] Y. Hu and T.J. Dennis, “Textured Image Segmentation by Context Enhanced Clustering”, In IEE Proc.-Vis. Image Signal Process., Vol.141, pp.413-421, December 1994.

[72] J. Mao, A.K. Jain, "Texture classification and segmentation using multiresolution simultaneous autoregressive models", Pattern Recognition, Vol.25, No.2, pp.173-188, 1992.

[73] M. L. Comer and E. J. Delp, "Segmentation of textured images using a multiresolution Gaussian autoregressive model", IEEE Trans. on Image Processing, Vol.8, No.3, pp.408420, March 1999.

[74] D.P. Mital and G.W. Leng, "Autoregressive approach to surface texture analysis", Proceedings of the 1992 International Conference on Industrial Electronics, Control, Instrumentation, and Automation, 1992. 'Power Electronics and Motion Control, Vol.3, pp.1309-1312, 1992.

[75] O. Alata and C. Ramananjarasoa, "Unsupervised textured image segmentation using 2-D quarter plane autoregressive model with four prediction supports", Pattern Recogn. Lett. Vol.26, No.8, pp.1069-1081, 2005.

[76] A. Sarkar, K.M.S. Sharma and R.V. Sonak, "A new approach for subset 2-D AR model identification fordescribing textures", IEEE Transactions on Image Processing, Vol.6, No.3, pp.407-413, 1997.

[77] S.R. Kadaba, S.B. Gelfand, R.L. Kashyap, "Recursive estimation of images using non-Gaussian autoregressivemodels", IEEE Transactions on Image Processing, Vol.7, No.10, pp.1439-1452, 1998.

[78] K. Eom, Generalized circular autoregressive models for modeling isotropic and anisotropic textures, Proceedings of International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, pp.129-132, 2002.

[79] N. Abbadeni, "Perceptual meaning of the estimated parameters of the autoregressive model", IEEE International Conference on Image Processing, Vol.3, pp.1164-1167, 2005.

[80] R.C. Dubes, A.K. Jain, S.G. Nadabar and C.C. Chen, “MRF model-based algorithms for image segmentation”, In Proceedings IEEE of the 10th International Conference on Pattern Recognition, Vol.1, pp.808-814, 1990.

[81] H. Elliott, H. Derin, R. Cristi and D. Geman, “Application of the Gibbs distribution to image segmentation”, In Proceedings of the International Conference on Acoustic, Speech and Signal Processing, pp. 32.5.1-32.5.4, 1984.

[82] M. I. Gurelli and L. Onural, “On a Parameter Estimation Method for Gibbs-Markov Random Fields”, IEEE Trans. Pattern Anal. Mach. Intell. Vol.16, No.4, 1994.

[83] B. Calder, L. Linnett, S. Clarke and D. Carmichael, “Improvements in MRF parameter estimation”, IEE Colloquium on Multiresolution Modelling and Analysis in Image Processing and Computer Vision, pp.3/1-3/6, 1995.

[84] K.H. Kim, B.S. Sharif and E.G. Chester, “Unsupervised texture analysis using a robust stochastic image model”, Sixth International Conference on Image Processing and Its Applications, Vol.2, pp.613-617, 1997.

[85] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.6, No.6, pp.721-741, 1984.

[86] S. Lakshmanan and H. Derin, “Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing”, IEEE Trans. Pattern Anal. Machine Intell., Vol.11, No.8, pp.799-813, 1989.

[87] S. Krishnamachari and R. Chellappa, “Multiresolution Gauss-Markov random field models for texture segmentation”, IEEE Trans’ on Image Processing, Vol.6, No.2, pp.251-267, 1997.

[88] I. M. Elfadel and R. W. Picard, Gibbs random fields, cooccurences, and texture modeling, IEEE Trans. Pattern Anal. Machine Intell., Vol.16, No.1, pp.24-37 , 1994.

[89] G.L. Gimel'farb and A.V. Zalesny, "Probabilistic Models of Digital Region Maps Based on Markov Random Fields with Short- and Long-Range Interaction", Pattern Recognition Letters, Vol.14, No.10, pp.789-797, 1993.

[90] F. S. Cohen and D. B. Cooper, "Simple parallel hierarchical and relaxation algorithms for segmenting noncausal Markovian random fields", IEEE Trans. Pattern Anal. Machine Intell., Vol.9, No.2, pp.195-219, March 1987.

[91] H. Derin and H. Elliot, "Modeling and segmentation of noisy and textured images using Gibbs random fields", IEEE Trans. Pattern Anal. Machine Intell., Vol.9, No.1, pp.39-55, 1987.

[92] G. Cross and A. Jain, "Markov Random Field Texture Models", IEEE Trans. On PAMI, Vol.5, No.1, pp.25-39, 1983.

[93] R. Chellapa and S. Chatterjee, "Classification of Texture Using Gaussian Markov Random Fields", IEEE Trans on ASSP, Vol.33, No.4, pp. 959-963, 1985.

[94] P. Andrey and P. Tarroux, "Unsupervised segmentation of Markov random field modeled texturedimages using selectionist relaxation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.20, No.3, pp.252-262, 1998.

[95] S. Lakshmanan and H. Derin, "Gaussian Markov random fields at multiple resolutions", In R. Chellappa, editor, Markov Random Fields: Theory and Applications, pages 131–157. Academic Press, New York, 1993.

[96] F. C. Jeng, Subsampling of Markov random fields, J. Visual Commun. and Image Repres., Vol. 3, No. 3, pp.225-229, 1992.

[97] I. U. Rozanov, “Markov random fields”, Springer-Verlag, 1982.

[98] R. Chellappa and A. Jain, “Markov random fields : theory and application”, Academic Press, 1991.

[99] A. P. Pentland, “Fractal-based description of natural scenes”, IEEE Trans. Pattern Anal. Machine Intell., Vol.6, No.6, pp.661-675, 1984.

[100] M. P. Dubuisson and R. C. Dubes, “Efficacy of fractal features in segmenting images of natural textures”, Pattern Recognition Letter, Vol.15, pp.419-431, 1994.

[101] L. M. Kaplan and C.-C. Jay Kuo, “Texture Roughness Analysis and Synthesis via Extended Self-Similar (ESS) Model”, IEEE Trans. Pattern Anal. Mach. Intell., Vol.17, No.11, 1043-1056, 1995.

[102] B. B. Chaudhuri and N. Sarkar, “Texture Segmentation Using Fractal Dimension”, IEEE Trans. Pattern Anal. Mach. Intell. Vol.17, No.1, pp.72-77, 1995.

[103] R.C. Luo, H. Potlapalli, D.W. Hislop, “Natural scene segmentation using fractal based autocorrelation”, Proceedings of the 1992 International Conference on Industrial Electronics, Control, Instrumentation, and Automation, Vol.2, pp.700-705, 1992.

[104] S.-C. Liu and S. Chang, “Dimension estimation of discrete-time fractional Brownian motionwith applications to image texture classification”, IEEE Transactions on Image Processing, Vol.6, No.8, pp.1176-1184, 1997.

[105] J. M. Keller and S. Chen, “Texture description and segmentation through fractal geometry”, Computer Vision, Graphics and Image Processing, Vol.45, pp.150-166, 1989.

[106] J. Garding, “Properties of fractal intensity surfaces”, Pattern Recognition Letters, Vol.8, No.5, pp.319-324, 1988.

[107] S.L. Tanimoto, “An Optimal Algorithm for Computing Fourier Texture Descriptors”, IEEE Transactions on Computers, Vol.C-27, No.1, pp.81-84, 1978.

[108] Dong-Chen He and Li Wang, “Unsupervised textural classification of images using the texture spectrum”, Pattern Recognition, Vol.25, No.3, pp.247-255, 1992.

[109] R. Azencott, J. Wang and L. Younes, “Texture classification using windowed fourier filters”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.19, No.2, pp.148-153, 1997.

[110] G. Ravichnadran and M. M. Trivedi, “Circulant-Mellin features for texture segmentation”, IEEE Transactions on Image Processing, Vol.4, No.12, 1995.

[111] M.S. Sutaone, P.P. Bartakke, V.S. Vyas and N.B. Pasalkar, “Rotation and scale invariant feature extractors”, Conference on Convergent Technologies for Asia-Pacific Region, Vol.1, pp.235-238, 2003.

[112] S. Qian, “Introduction to Time-Frequency and Wavelet Transforms”, Prentice Hall PTR, 2002.

[113] A. V. Oppenheim, A. S. Willsky and S. H. Nawab, “Signals and Systems Second Edition”, Prentice Hall, 1996.

[114] H.P. Robertson, ”The uncertainty principle”, Phys. Rev., Vol.34, No.1, pp. 163164, 1929.

[115] E.Schrodinger, “About Heisenberg uncertainty relation”, Proc.Prussian Acad.Sci., Phys.Math. Section, Vol.XIX, pp.293, 1930.

[116] L. Cohen, “Time-Frequency Analysis”, Prentice Hall PTR, 1995.

[117] T. Reed and H. Wechsler, “Segmentation of textured images and gestalt organization using spatial/spatial-frequency representation”, IEEE Trans. Pattern Anal. Mach. Intell. Vol.12, No.1, pp.1-12, 1990.

[118] Z. Huang, K. L. Chan and Y. Huang , “Local spectra features extraction based on 2D pseudo-Wignerdistribution for texture analysis”, Proceedings of 2000 International Conference on Image Processing, Vol.3, pp.917-920, 2000.

[119] C.-Y. Wen and R. Acharya, “Self-similar texture characterization using Wigner-Villedistribution”, Proceedings of International Conference on Image Processing 1996, Vol.3, pp.141-144, 1996.

[120] H. Choi and W. J. Williams, “Improved Time-Frequency Representation of Multicomponent Signals Using Exponential Kernels”, IEEE Trans. Acoust. Speech, and Signal Processing, Vol.37, No.6, pp.862-871, June 1989.

[121] L. Durak, ”Novel Time-Frequency Analysis Techiniques For Deterministic Signals”, Doctoral Thesis Dissertation, Bilkent University, Dept. of Electrical and Electronics Engineering, December 2003.

[122] G. Matz and F. Hlawatsch, “Wigner distributions (nearly) everywhere: Time-frequency analysis of signals, systems, random processes, signal spaces, and frames”, Signal Processing, Vol.83, No.7 pp.13551378, 2003.

[123] S. Qian and D. Chen, “Joint time-frequency analysis”, IEEE Signal Process. Mag., Vol.16, No.2, pp.52-67, March 1999.

[124] W. Mecklenbruker and F. Hlawatsch, “The Wigner Distribution: Theory and Applications in Signal Processing”, Elsevier, 1997.

[125] A. Mertins, “Signal Analysis: Wavelets, Filter Banks, Time-Frequency Transforms and Applications”, John Wiley & Sons Lt, 1999.

[126] A.J.E.M. Janssen, “Optimality property of the Gaussian window spectrogram”, IEEE Trans. SP. Vol.39 No.1, pp. 202-204, 1991.

[127] J. Bigun and J. M. H. Du Buf, “N-folded symmetries by complex moments in Gabor space and their application to unsupervised texture segmentation”, IEEE Trans. on PAMI, Vol.16, No.1, pp.80-87, 1994.

[128] A.K. Jain, F. Farrokhnia, “Unsupervised Texture Segmentation Using Gabor Filters”, Pattern Recongition, Vol.24, No.12, pp.1167-1186, 1991.

[129] T. Hofmann, J. Puzicha and J.M. Buhmann, “Unsupervised segmentation of textured images by pairwise dataclustering”, Proceedings of International Conference on Image Processing, Vol.3, pp.137-140, 1996.

[130] S.E. Grigorescu, N. Petkov and P. Kruizinga, “Comparison of texture features based on Gabor filters”, IEEE Transactions on Image Processing, Vol.11, No.10, 2002.

[131] J. Chen, T. N. Pappas, A. Mojsilovic and B. Rogowitz, “Image Segmentation by Spatially adaptive color and texture features”, Proceedings of IEEE International Conference on Image Processing, Vol.1, pp.1005-1008, 2003.

[132] B. J. Super and A. C. Bovik, “Localized Measurement of Image Fractal Dimension Using Gabor Filters”, Journal of Visual Communication and Image Representation, Vol.2, No.2, pp.114-128, June 1991.

[133] M. Clark, A. C. Bovik and W. S. Geisler, “Texture segmentation using Gabor modulation/demodulation”, Pattern Recognition Letters, Vol.6, pp.261267, Sept. 1987.

[134] A. C. Bovik, M. Clark and W. S. Geisler, “Multichannel texture analysis using localized spatial Filters”, IEEE Trans. Pattern Anal. Mach. Intell., Vol.12, No.1, pp.55-73, 1990

[135] A. Teuner, O. Pichler and B.J. Hosticka, “Unsupervised Texture Segmentation of Images Using Tuned Matched Gabor Filters”, IEEE Trans On.Image Processing, Vol.4, No.6, pp.863-870, June 1995.

[136] T. P. Weldon, W. E. Higgins and D. F. Dunn, “Gabor filter design for multiple texture segmentation”, Optical Engineering SPIE, Vol.35, pp.2852-2863, October 1996.

[137] M. Turner, “Texture discrimination by Gabor functions”, Biological Cybernetics, Vol.55, pp.71-82, 1986.

[138] D. Dunn, W.E. Higgins and J Wakeley, “Texture Segmentation Using 2-D Gabor Elementary Functions”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.16, No.2, pp.130-149, 1994.

[139] A. K. Jain and K. Karu, “Learning texture discrimination masks”, IEEE Trans. Pattern Anal. Mach. Intell., vol. 18, no. 2, pp. 195–205, Feb. 1996.

[140] P. Scheunders, S. Livens, G. van de Wouwer, P. Vautrot and D. van Dyck, “Wavelet-based texture analysis”, International journal on computer science and information management, Vol1:2, pp.22-34, 1998.

[141] N. Sebe, M.S. Lew, “Wavelet Based Texture Classification”, 15th International Conference on Pattern Recognition, Vol.3, pp.947-950, 2000.

[142] F. Destrempes and M. Mignotte, “Unsupervised Texture Segmentation Using a Statistical Wavelet-Based Hierarchical Multi Data Model”, Proc. 10th IEEE Int’l Conf. Image Processing, Vol.2, pp.1053-1056, 2003.

[143] T. Randen and J.H. Husy, “Multichannel Filtering for Image Texture Segmentation”, Optical Eng., Vol.33, pp. 2617-2625, 1994.

[144] A. Laine and J. Fan, “Texture Classification by Wavelet Packet Signatures”, IEEE Trans On. PAMI, Vol.15, No.11, pp.1186-1191, 1993.

[145] R. Porter and N. Canagarajah, “A Robust Automatic Clustering Scheme foe Image Segmentation Using Wavelets”, IEEE Trans. on Image Processing, Vol.5, No.4, pp.662-665, 1996.

[146] T. Chang and C.-C. J. Kuo, Texture analysis and classification using tree-structured wavelet transform, IEEE Trans. Image Processing, Vol.2, No.4, pp.429441, Oct. 1993.

[147] J. Portilla and E. P. Simoncelli, A parametric texture model based on joint statistics of complex wavelet coefficients, International Journal of Computer Vision, Vol.40, No.1, pp.4971, Dec 2000.

[148] M. Vetterli and C. Herley, “Wavelets and Filter Banks: theory and design”, IEEE Transactions on Signal Processing, Vol.40, No.9, pp.2207-2232, 1992.

[149] G. Strang and T. Nguyen, “Wavelets and Filter Banks”, Wellesly-Cambridge Press, 1997.

[150] S.G. Mallat, “A Wavelet Tour of Signal Processing”, 2nd ed., Academic Press, 1999.

[151] M. Unser, M. Eden, “Multiresolution feature extraction and selection for texture segmentation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.11, No.7, pp.717-728, 1989.

[152] M. Unser and M. Eden, “Nonlinear operators for improving texture segmentation based on features extracted by spatial filtering”, IEEE Transactions on Systems, Man, Cybernetics, Vol.20, No.4, pp.804-815, 1990.

[153] I. Ng, T. Tan and J. Kittler, “On Local Linear Transform and Gabor Filter Representation of Texture”, in Proc. 11th IAPR Intl. Conf. Pattern Recognition, Vol.3, pp.627-631, 1992.

[154] P. Kruizinga and N. Petkov, “Nonlinear operator for oriented texture”, IEEE Transactions on Image Processing, Vol.8, No.10, pp. 1395-1407, 1999.

[155] M. Unser, “Local linear transform for texture measurements”, Signal Processing, Vol.11, pp.61-79, 1986.

[156] J. M. Coggins and A. K. Jain, “A spatial filtering approach to texture analysis”, Pattern Recognition Letters, Vol.3, No.3, pp.195-203, 1985.

[157] X. Gong N-K Huang, “Texture segmentation using iterative estimate of energy states”, 9th International Conference on Pattern Recognition, pp.51-55, 1988.

[158] M. Pietikainen, A. Rosenfeld and L. S. Davis, ”Experiments with texture classification using averages of local pattern matches”, IEEE Trans. Systems, Man and Cybernetics, Vol.13, No.3, pp.421-426, 1983.

[159] K. I. Laws, ”Rapid texture identification”, Proc. SPIE, Vol.238, pp. 376-380, 1980.

[160] J. Y. Hsiao and A. A. Sawchuk, “Unsupervised textured image segmentation using feature smoothing and probabilistic relaxation techniques”, Comput. Vision Graph. Image Process., Vol.48, pp.1-21, 1989.

[161] P. Cohen, C. LeDinh and V. Lacasse, Classification of Natural Textures by Means of Two-Dimensional Orthogonal Masks, IEEE Trans. Acoustics, Speech and Signal Processing, Vol.37, No.1, pp.125-128, 1989.

[162] C. W. Tyler, “Theory of texture discrimination of based on higher-order perturbations in individual texture samples”, Vision Research, Vol.44, No.18, pp.2179-2186, 2004.

[163] M. Amadasun and R. King, “Textural Features Corresponding to Textural Properties”, IEEE Transactions on Systems, Man, and Cybernetics, Vol.19, No.5, pp.12641274, 1989.

[164] M. Tuceryan, “Moment-Based Texture Segmentation”, Pattern Recognition Letters, Vol.15, pp. 659-668, 1994.

[165] T. Ojala, M. Pietikainen and D. Harwood, “A comparative study of texture measures with classification based on feature distribution”, Pattern Recognition, Vol.29, No.1, pp.51-59, 1996.

[166] J.M.H. Buf, M. Kardan and M. Spann, “Texture feature performance for image segmentation”, Pattern Recognition, Vol.23, No.3/4, pp.291-309, 1990.

[167] J. G. Zhang and T. N. Tan, “Brief Review of Invariant Texture Analysis Methods”, Pattern Recognition, Vol.25, No.3, pp.735-747, 2002.

[168] P.P. Ohanian and R.C. Dubes, “Performance evaluation for four classes of textural features”, Pattern Recognition, Vol.25, No.8, pp.819-833, 1992.

[169] T. Reed and J. Du Buf, “A review of recent texture segmentation and feature extraction techniques”, CVGIP Image Understanding, Vol.57, No.3, pp.359-372, 1993.

[170] R.M. Haralick, “Statistical and structural approaches to texture”, Proc. of IEEE., Vol.67, No.5, pp.786-804, 1979.

[171] T. Randen and J. Husy, “Filtering for texture classification: a comparative study”, T-PAMI Vol.21, No.4, pp.291-310, 1999.

[172] O. Pichler, A. Teuner and B. J. Hosticka, “A comparison of texture feature extraction using adaptive gabor filtering, pyramidal and tree structured wavelet transforms”, Pattern Recognition, Vol.29, No.5, pp.733-742, 1996.

[173] M. Tuceryan and A. K. Jain, “Texture Analysis”, In The Handbook of Pattern Recognition and Computer Vision (2nd Edition), by C. H. Chen,

L. F. Pau, P. S. P. Wang (eds.), pp. 207-248, World Scientific Publishing Co., 1998.

[174] G. E. Lowitz. "Can a local histogram really map texture information?", Pattern Recognition, Vol.16, No.2, pp.141-147, 1983.

[175] S. Y. Doh and R-H. Pang, Segmentation of Statistical Texture Images Using the Metric Space Theory, Signal Processing, Vol.53, pp.27-34, 1996.

[176] D. Wang, S.N. Shihari, "Classification of newspaper image blocks using texture analysis", Computer Vision Graphics Image Process. Vol.47, pp.327-352, 1989.

[177] M. Tuceryan and A. K. Jain, Texture Segmentation Using Voronoi Polygons, in IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. PAMI-12, pp. 211216, February, 1990.

[178] D. Chetverikov, "Pattern regularity as a visual key", Image and Vision Computing, Vol.18, pp.975986, 2000.

[179] M. K. Tsatsanis and G. B. Giannakis, "Object and texture classification using higher order statistics", IEEE Trans. Pattern Anal. Mach. Intell., Vol.14, No.7, pp.733750, 1992.

[180] J. Strand and T. Taxt, "Local frequency features for texture classification", Pattern Recognition, Vol.27, No.10, pp.1397-1406, 1994.

[181] M.E. Jernigan and F. D'astous, "Entropy-based texture analysis in the spatial frequency domain", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.6, No.2, pp.237-243, 1984.

[182] Y. Rubner, J. Puzicha, C. Tomasi and J. M. Buhmann, "Empirical Evaluation of Dissimilarity Measures for Color and Texture", Computer Vision and Image Understanding, Volume 84, Issue 1, Pages 25-43, 2001.

[183] P. Torr, “Structure and motion toolkit in Matlab”, Department of Computing, Oxford Brookes University, England. Available from: <http://cms.brookes.ac.uk/staff/PhilipTorr/ >

[184] E. Rosten, “FAST Corner Detection”, Engineering Department, Machine Intelligence Laboratory, University of Cambridge. Available from: <http://mi.eng.cam.ac.uk/ er258/index.html>

[185] S. Ettinger, “SIFT Matlab Implementation”, Intel. Available from: <http://robots.stanford.edu/cs223b/MatlabSIFT.zip>

[186] “CIPR SIF Sequences”, Center for Image Processing Research, ECSE Dept, Rensselaer Polytechnic Institute, Troy, NY 12180, Available from: <http://www.cipr.rpi.edu/resource/sequences/sif.html>

[187] L. Onural, M. Kocatepe, “Family of scaling chirp functions, diffraction and holography”, IEEE Transactions on Signal Processing, 43, pp.1568-1578, 1995.

[188] P. K. Sahoo, S. Soltani, A. K.C. Wong, Y. C. Chen, “A survey of thresholding techniques”, Computer Vision Graph. Image Process, Vol.41, No.2, pp.233-260, 1988.

[189] Anil K. Jain, “Fundamentals of Digital Image Processing”, New Jersey, United States of America: Prentince-Hall Inc., p. 68, 71, 73., 1989.

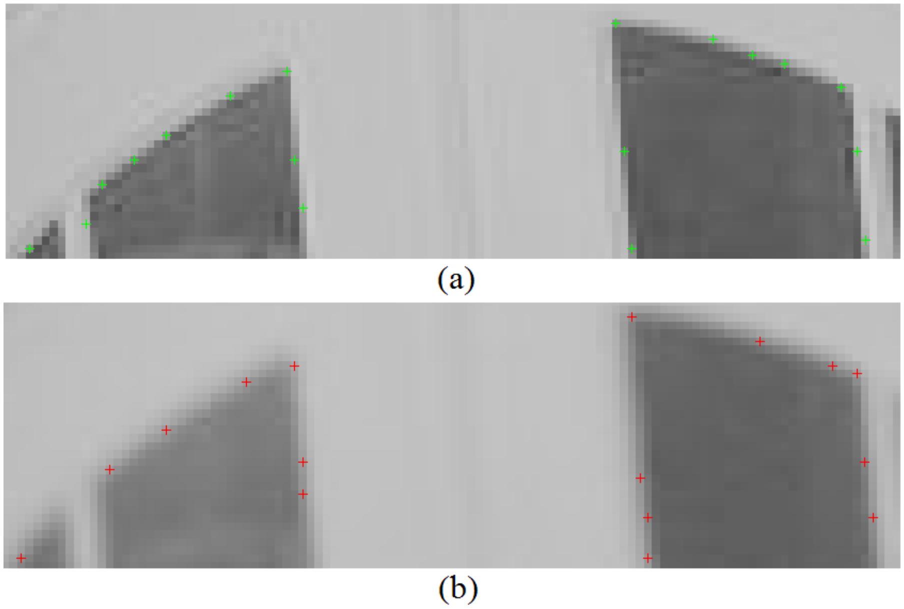
[190] C. Schmid, R. Mohr, C. Bauckhage,”Comparing and Evaluating Interest Points”, ICCV98, pp.230-235, 1998.

[191] ITU-R BT.709-5 (04/02) Parameter values for the HDTV standards for production and international programme exchange, 2002.

Appendix A

Image Couples Test Results

A.1 Detection Results



(a)

(b)

Figure A.1: The detected feature points by using the Harris FPD. The original "bike" image couple is taken from Oxford University Visual Geometry Group.

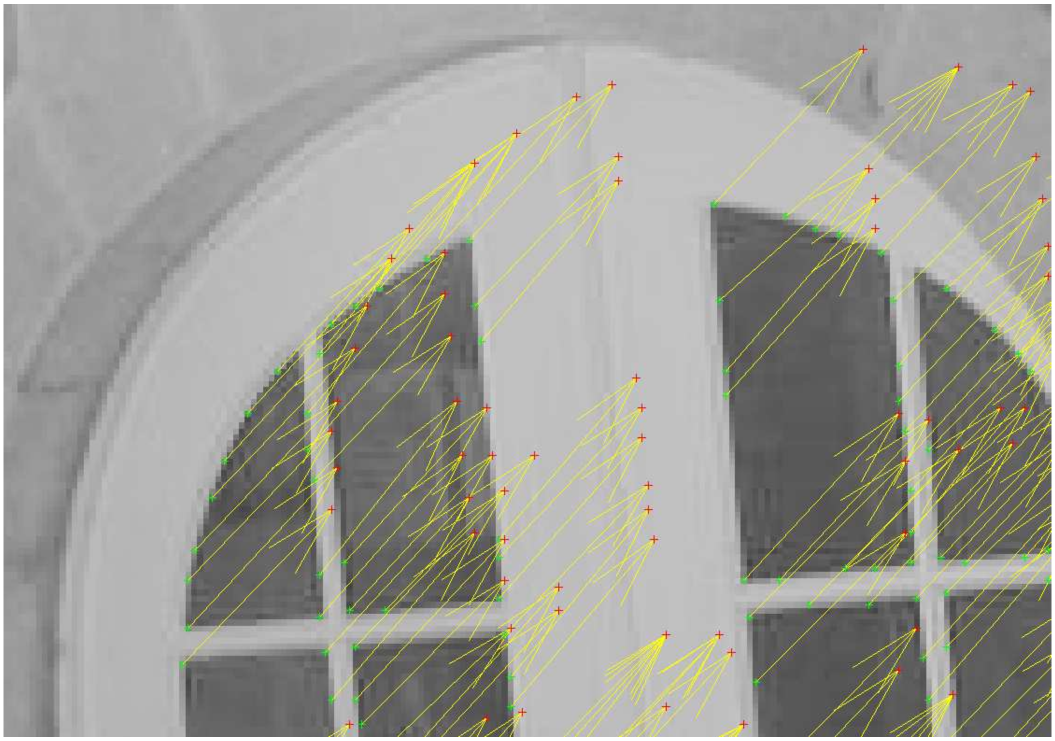


Figure A.2: The matched feature points among points detected shown in Figure A.1 by using gray-level patch description. The original "bike" image couple is taken from Oxford University Visual Geometry Group.

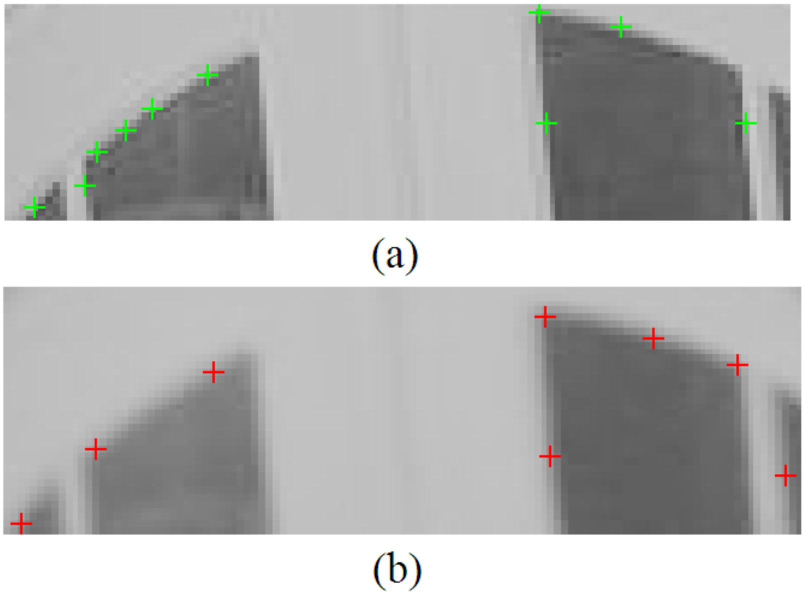


Figure A.3: The detected feature points by using the proposed FPD.

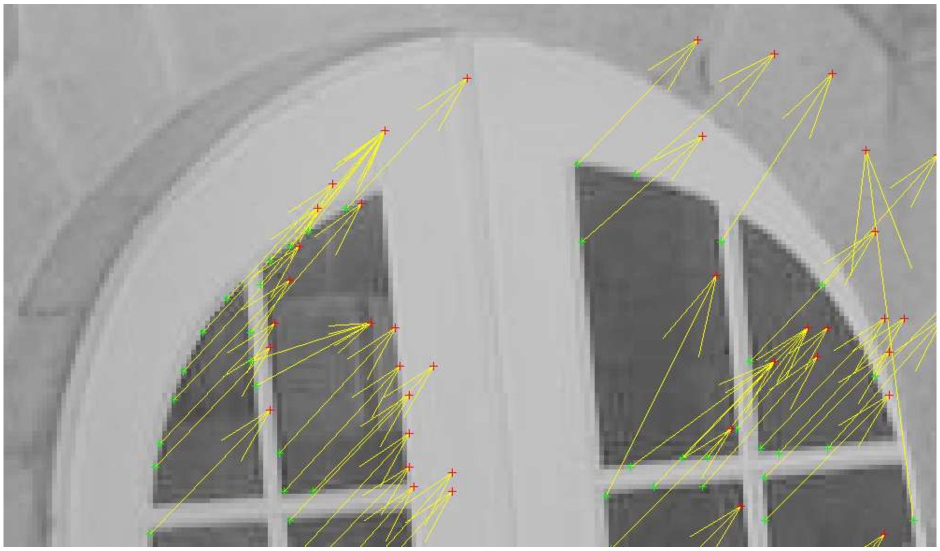


Figure A.4: The matched feature points among points detected shown in Figure A.3 by using gray-level patch description.

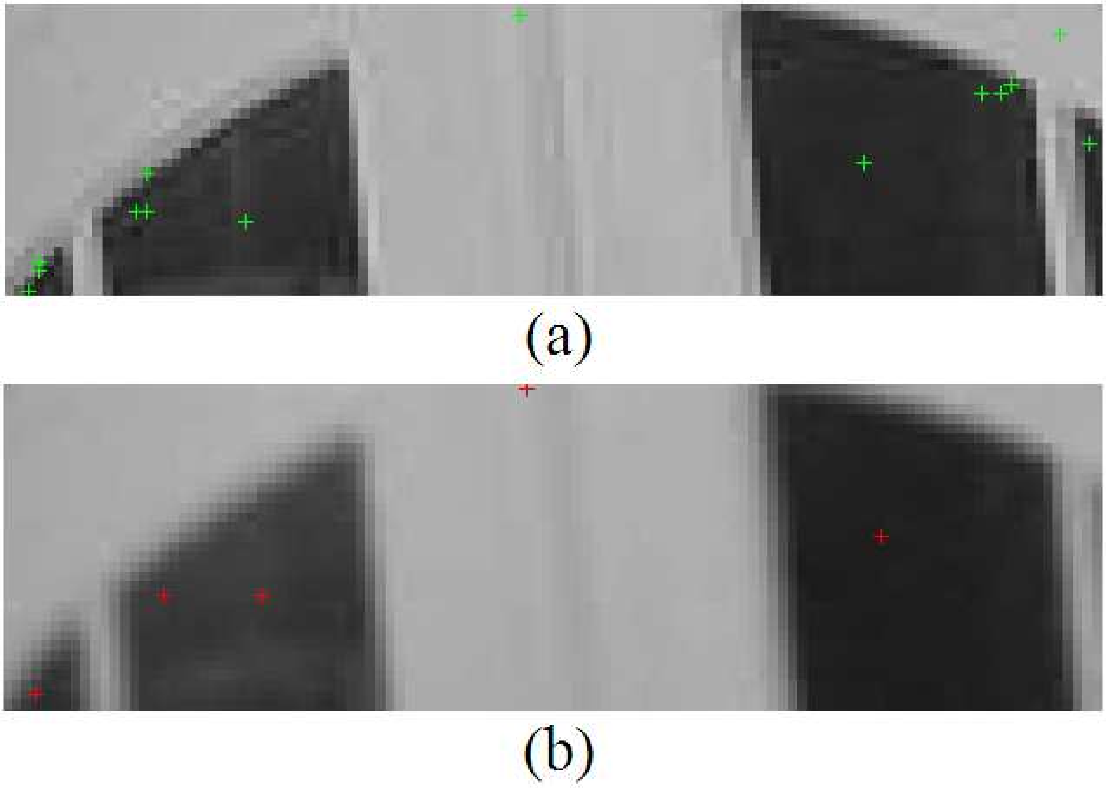


Figure A.5: The detected feature points by using the SIFT FPD.

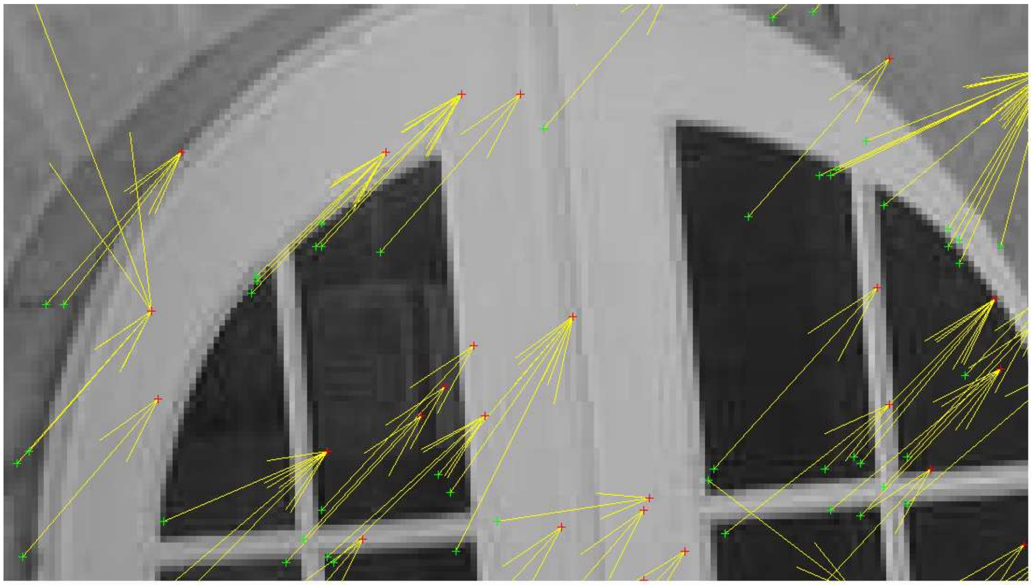
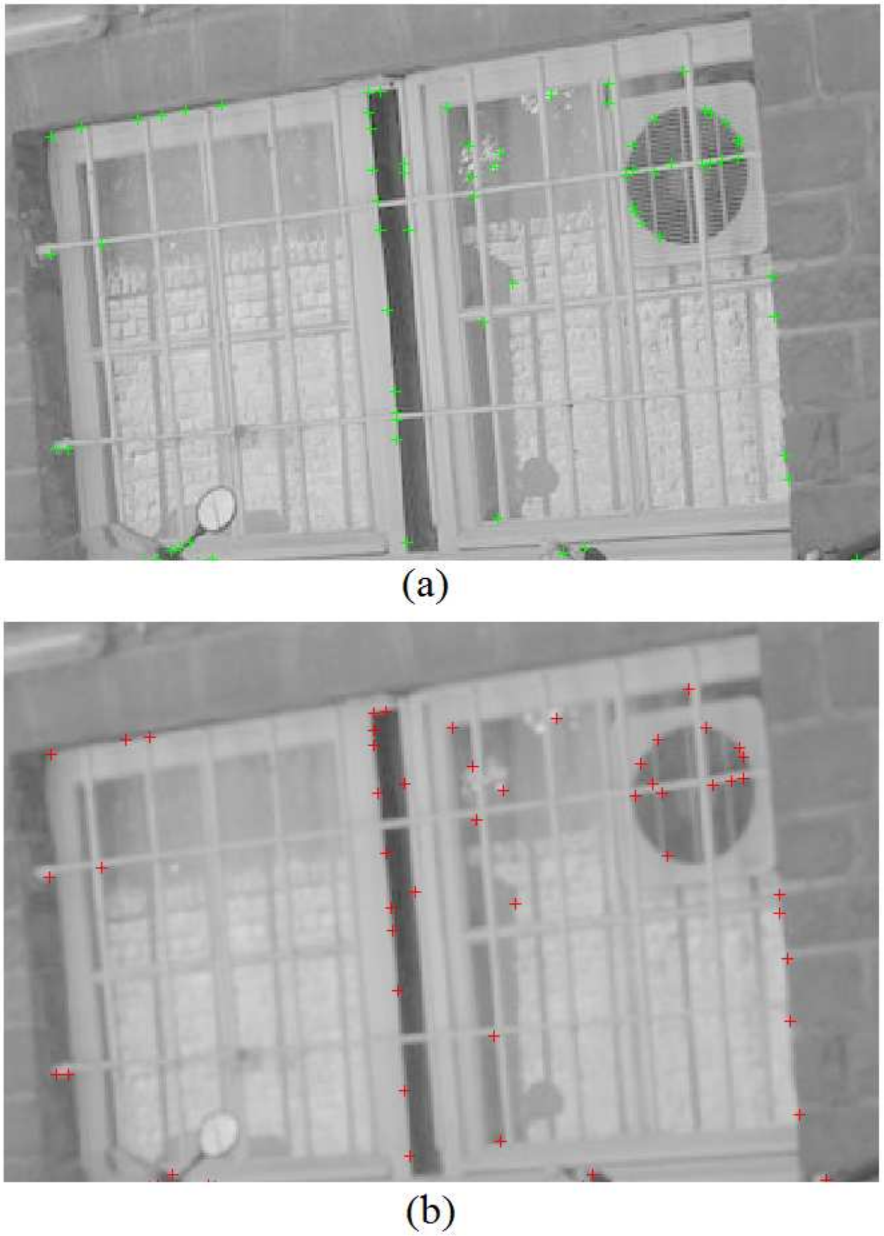


Figure A.6: The matched feature points among points detected shown in Figure A.5 by using gray-level patch description.

A.2 Scale Selection Results



(a)

(b)

Figure A.7: The detected feature points by using the proposed FPD with no scale selection.

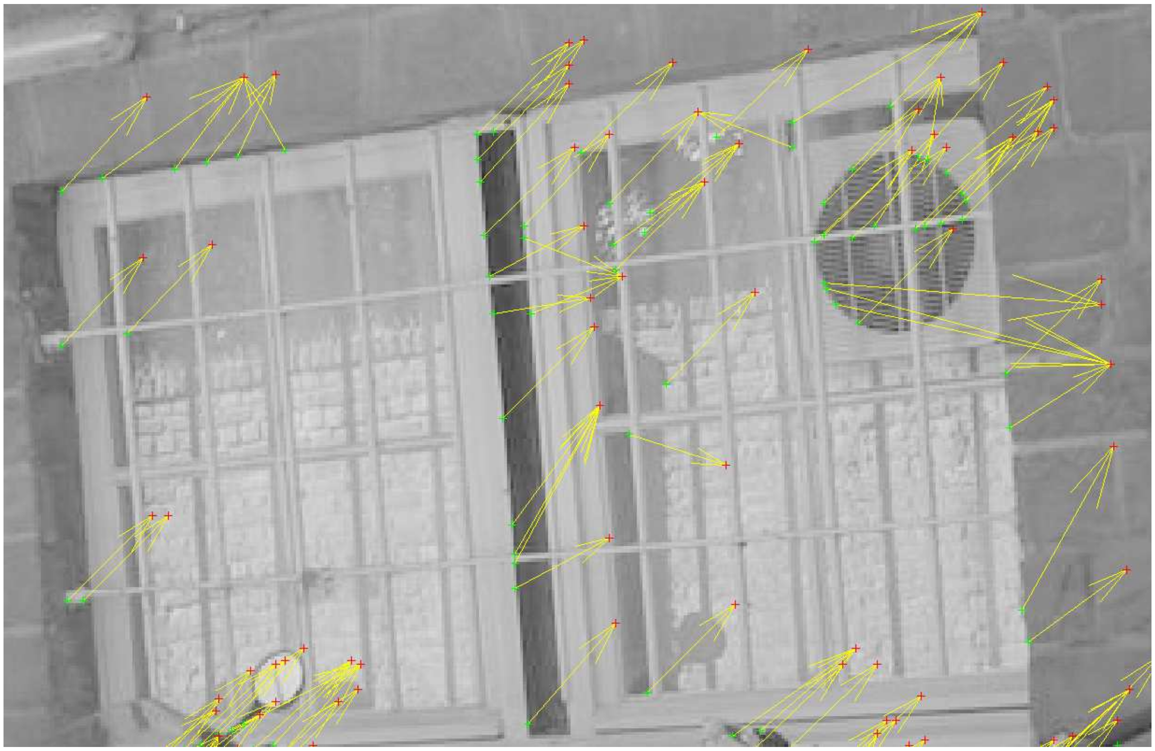
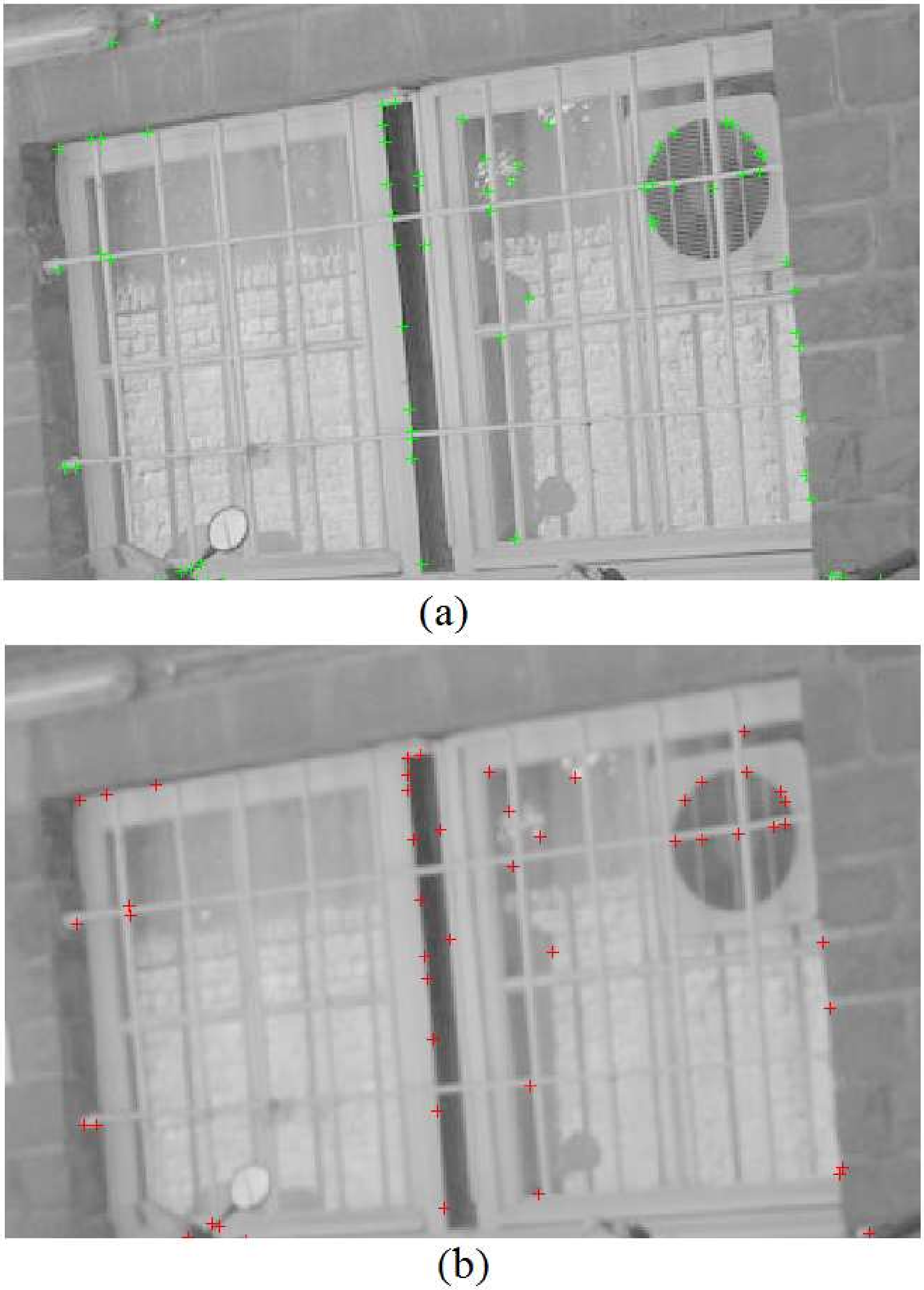


Figure A.8: The matched feature points among the detected points shown in Figure A.7 by using the proposed matching method.



(a)

(b)

Figure A.9: The detected feature points by using the proposed FPD with the proposed scale selection.

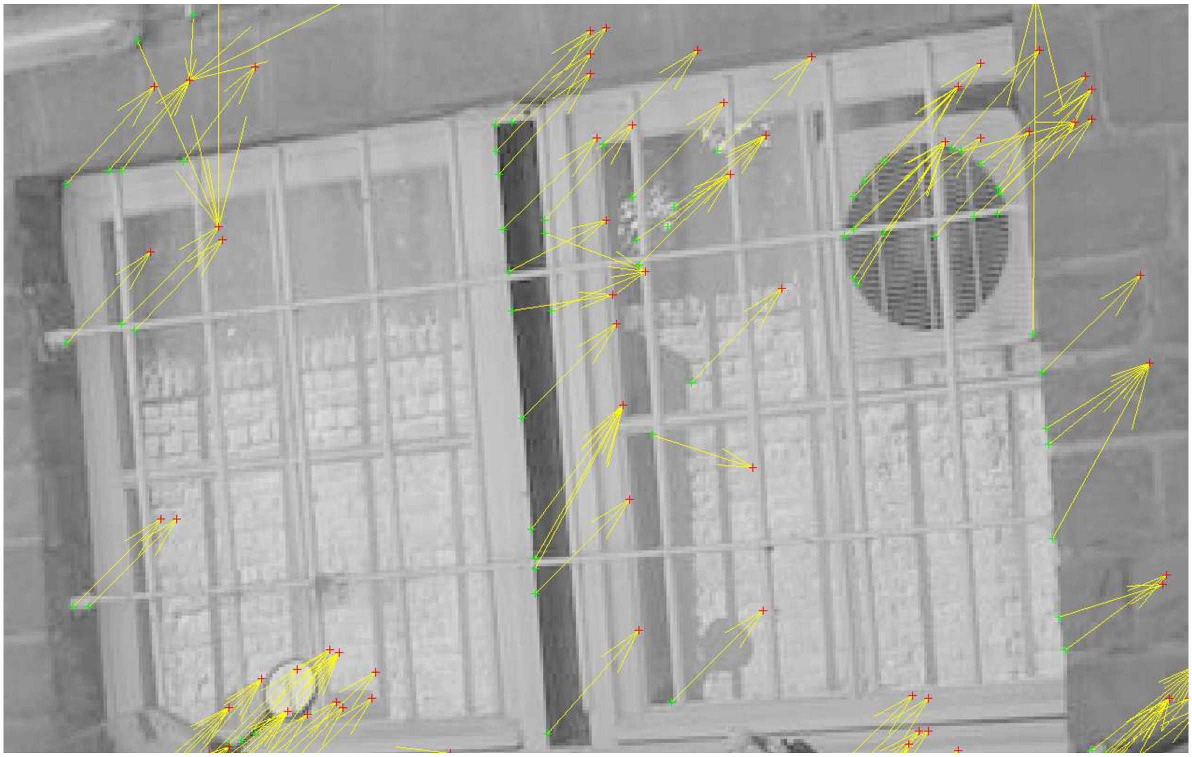
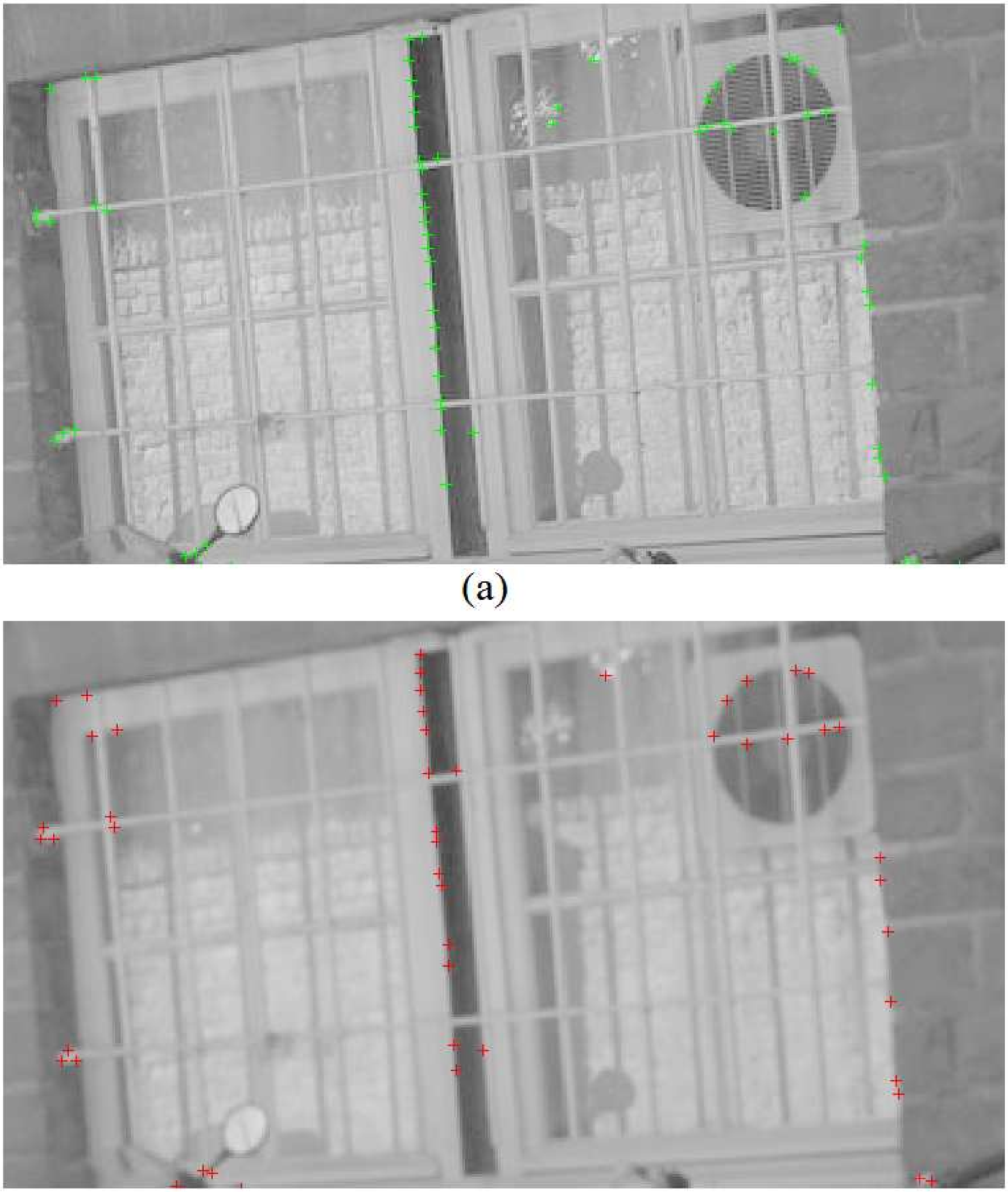


Figure A.10: The matched feature points among the detected points shown in Figure A.9 by using the proposed matching method.



(a)

(b)

Figure A.11: The detected feature points by using the proposed FPD with the adjacent finer scale to the proposed scale is selected.

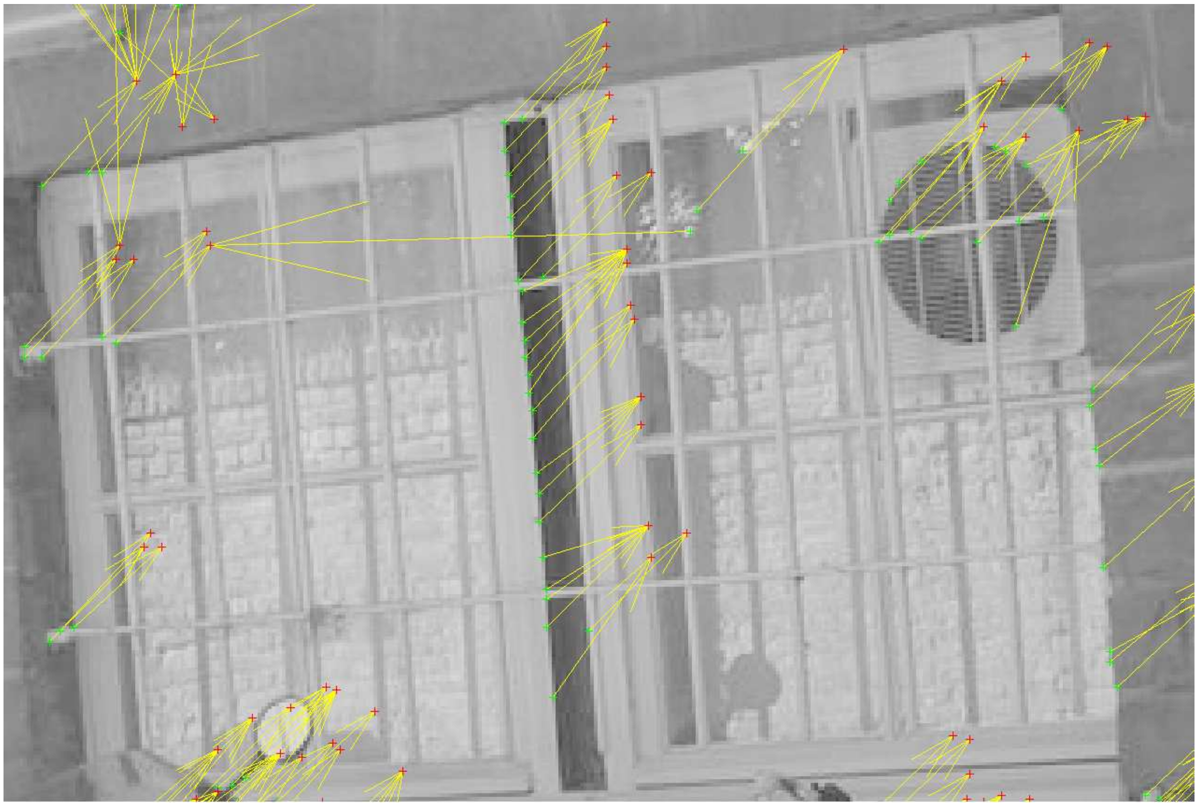
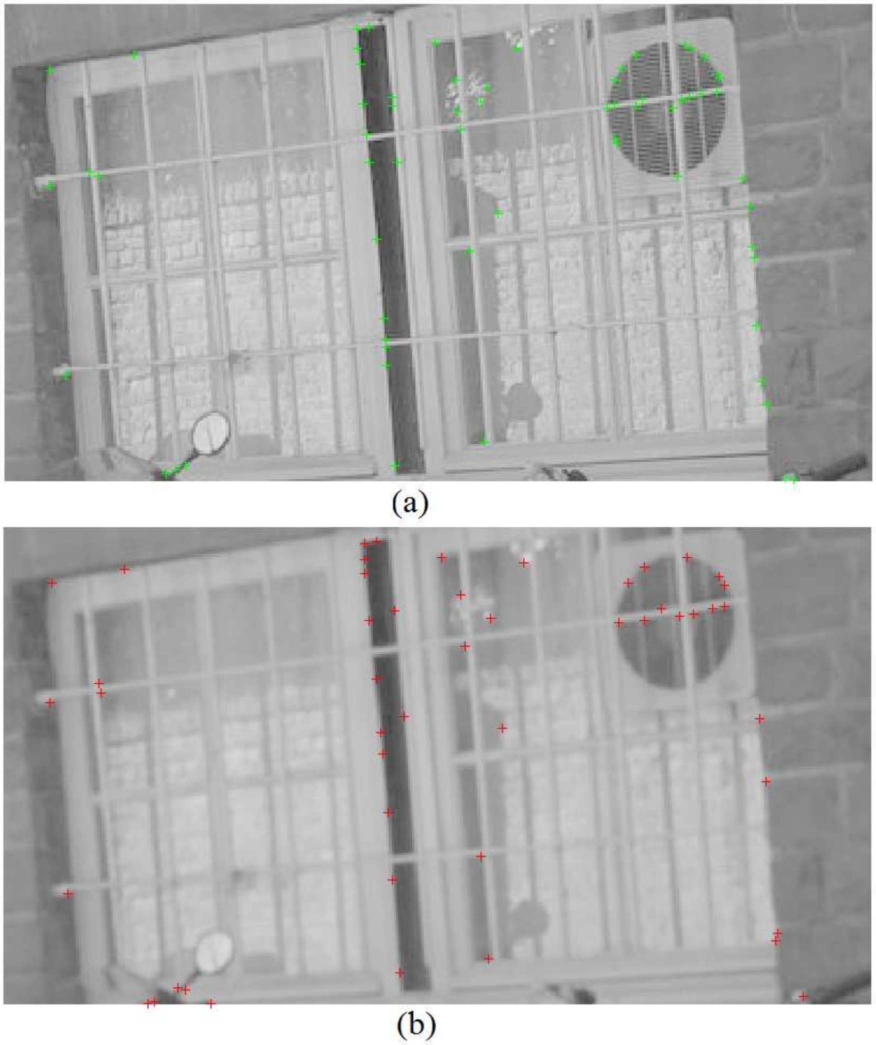


Figure A.12: The matched feature points among the detected points shown in Figure A.11 by using the proposed matching method.


(a)

(b)

Figure A.13: The detected feature points by using the proposed FPD with the adjacent coarser scale to the proposed scale is selected.

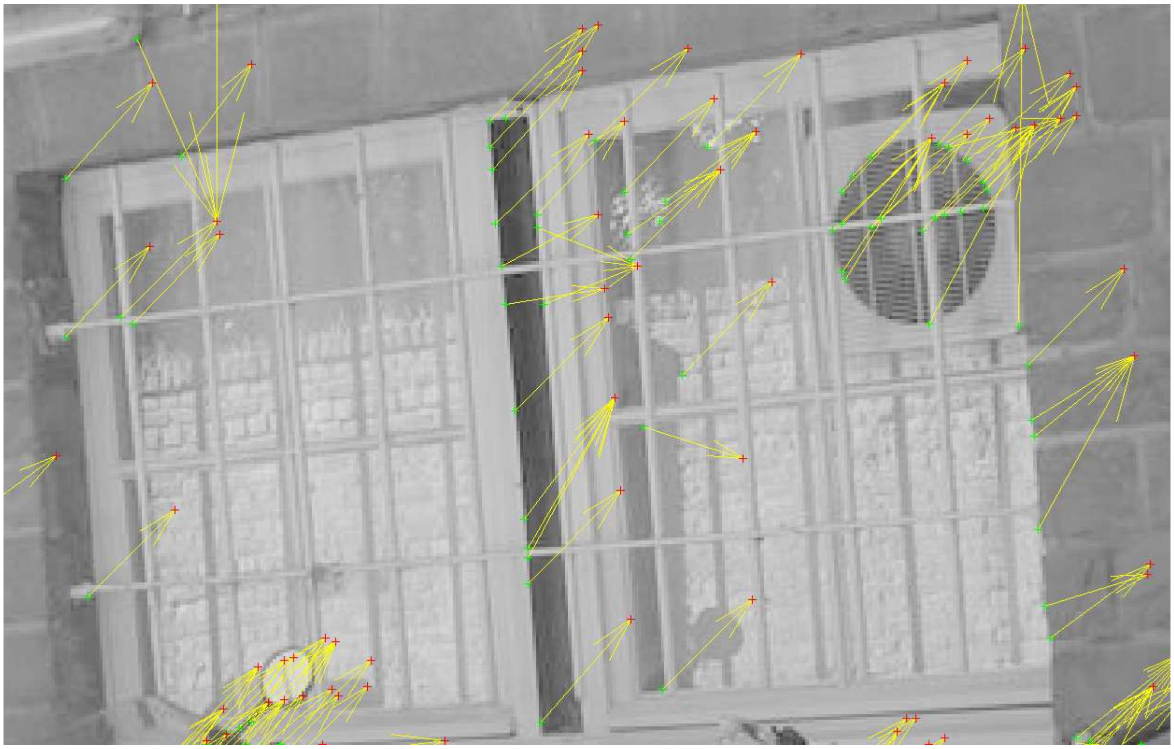


Figure A.14: The matched feature points among the detected points shown in Figure A.13 by using the proposed matching method.

Appendix B

Video Sequence Results

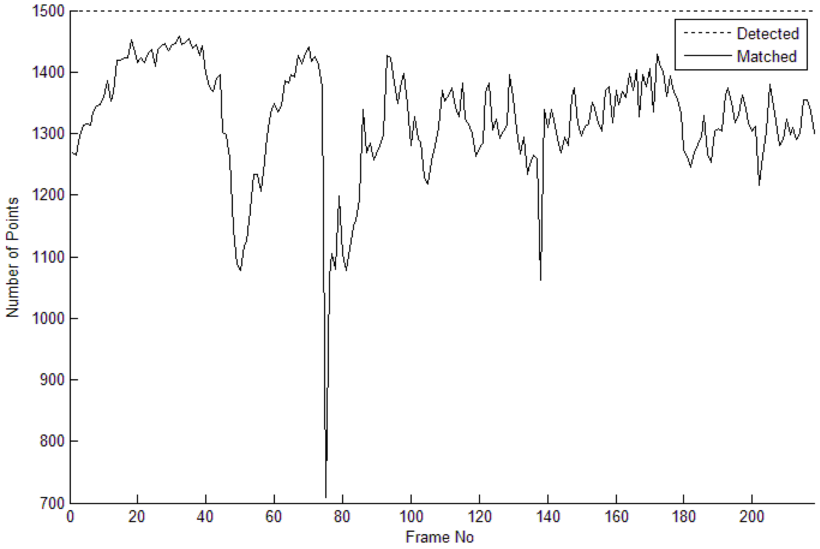


Figure B.1: The detection and matching results of "src6_ref_625.avi" for the first scheme.

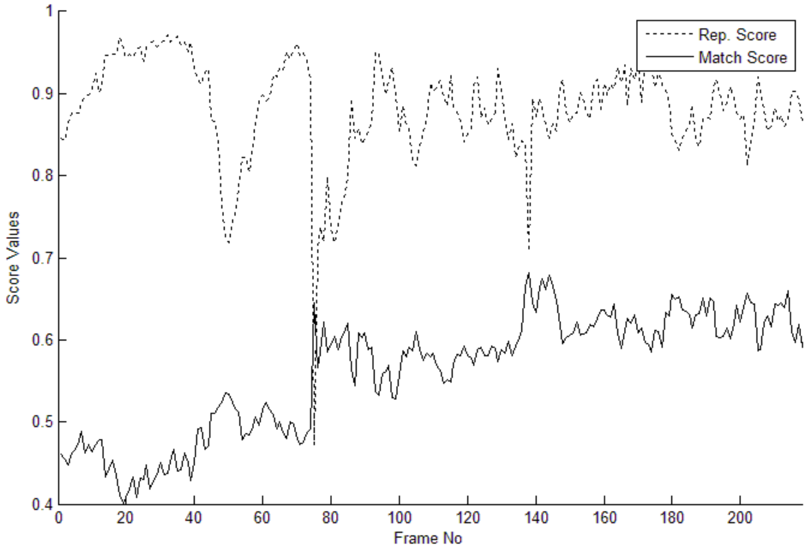


Figure B.2: The repetability and matching scores of "src6_ref_625.avi" for the first scheme.

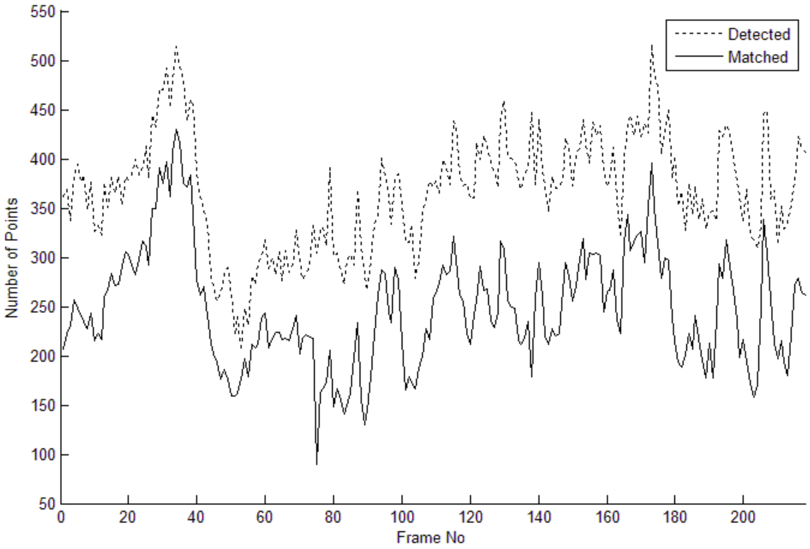


Figure B.3: The detection and matching results of "src6_ref_625.avi" for the second scheme.

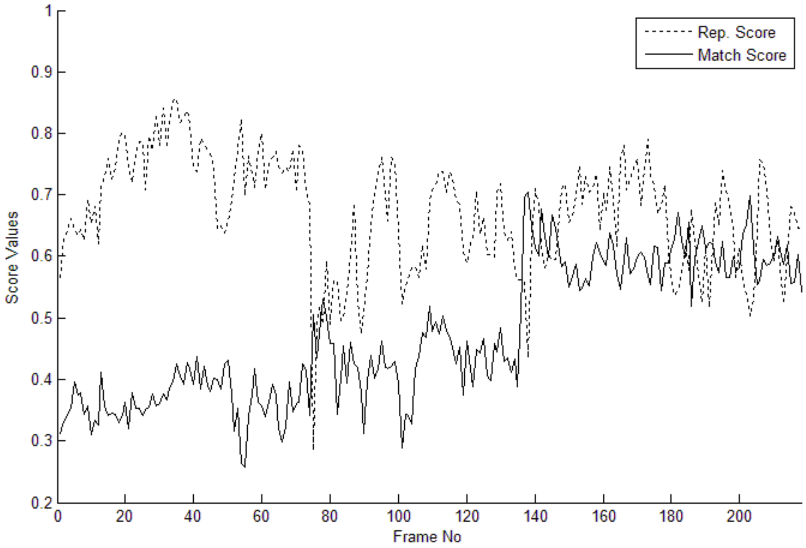


Figure B.4: The repetability and matching scores of "src6_ref_625.avi" for the second scheme.

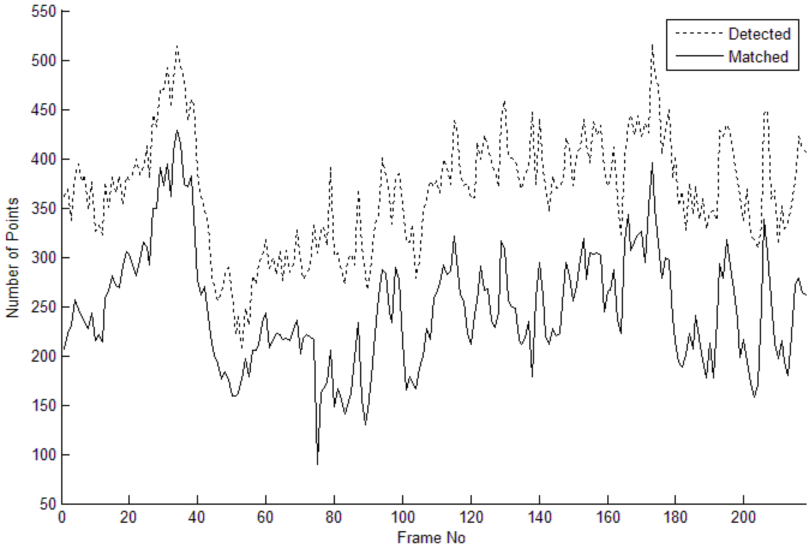


Figure B.5: The detection and matching results of "src6_ref_625.avi" for the third scheme.

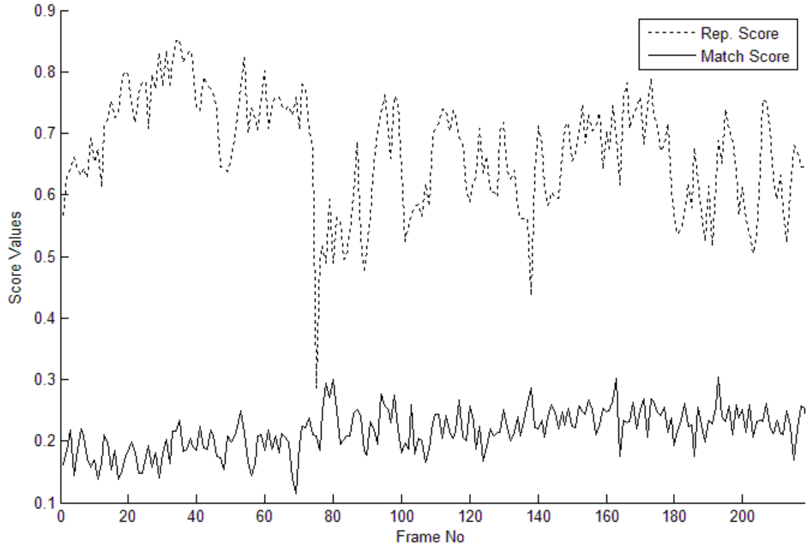


Figure B.6: The repetability and matching scores of "src6_ref_625.avi" for the third scheme.

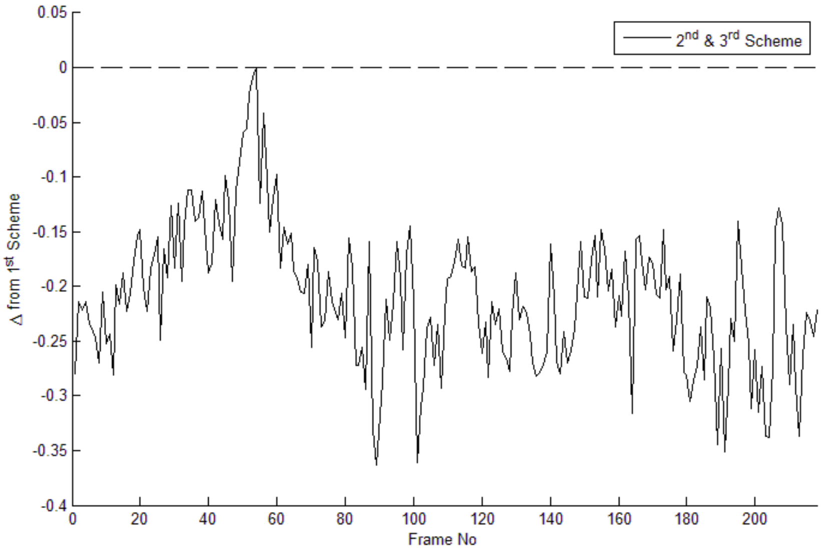


Figure B.7: The difference between repetability score of 1^{st} scheme and $2^{nd} or 3^{rd}$ schemes for "src6_ref_625.avi".

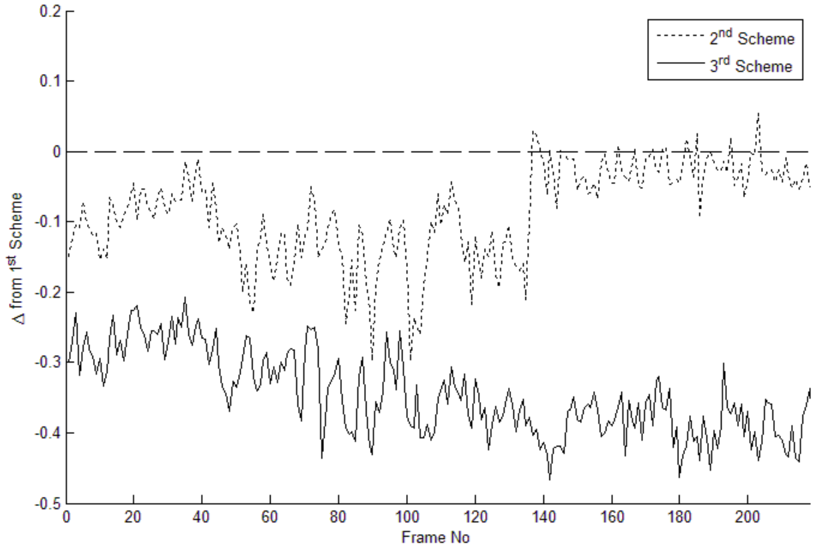


Figure B.8: The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for "src6_ref_625.avi".

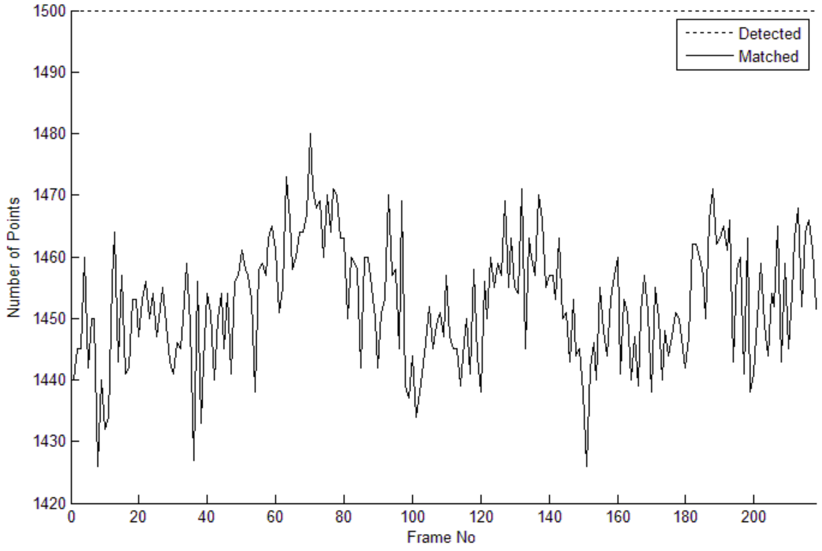


Figure B.9: The detection and matching results of "src10_ref_625.avi" for the first scheme.

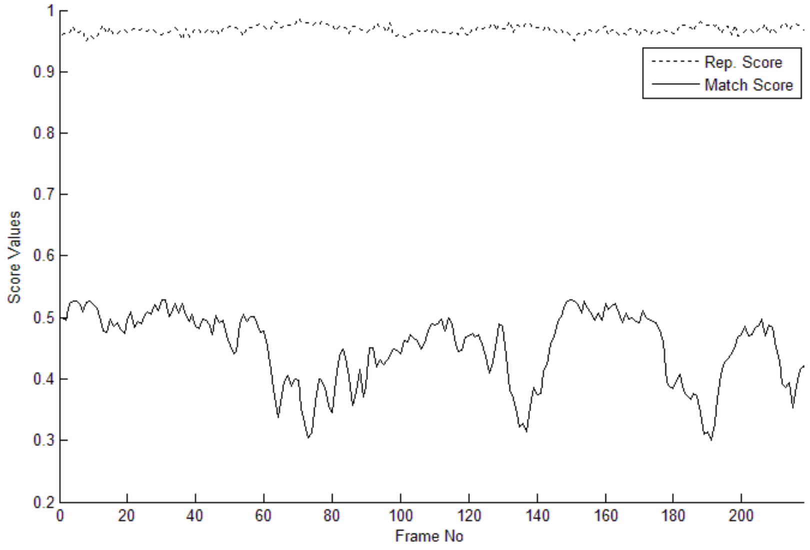


Figure B.10: The repetability and matching scores of "src10_ref_625.avi" for the first scheme.

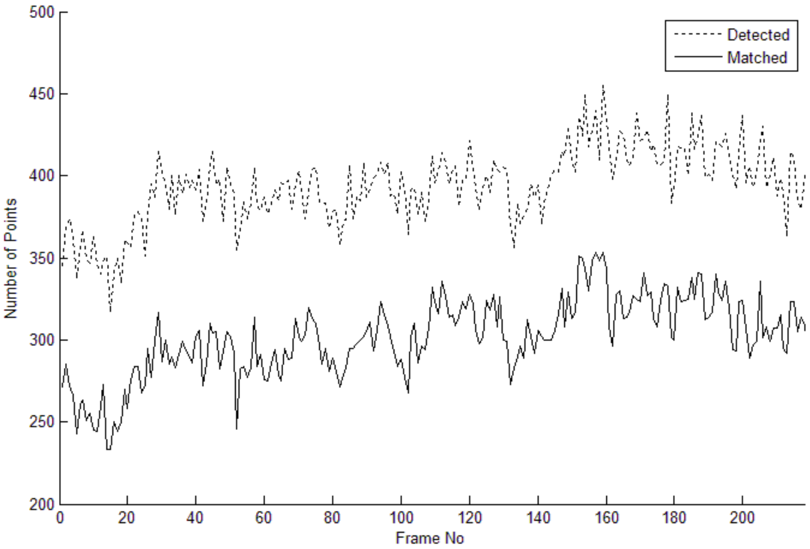


Figure B.11: The detection and matching results of “src10_ref_625.avi” for the second scheme.

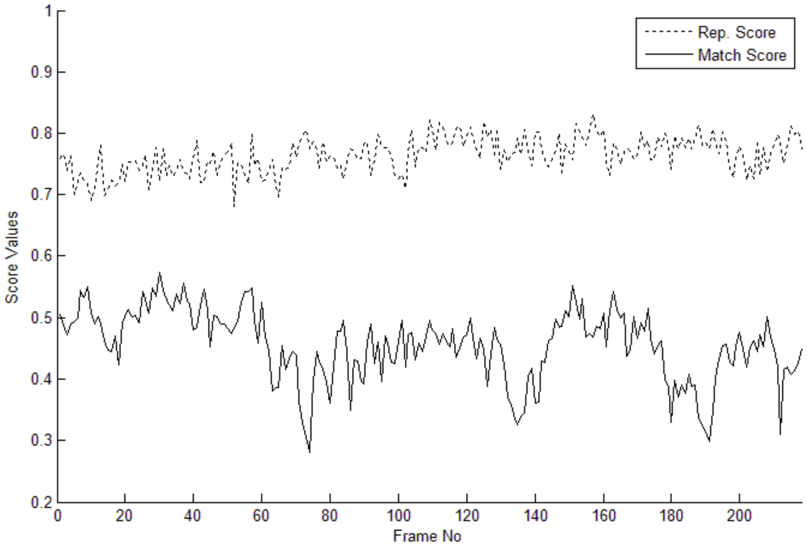


Figure B.12: The repetability and matching scores of “src10_ref_625.avi” for the second scheme.

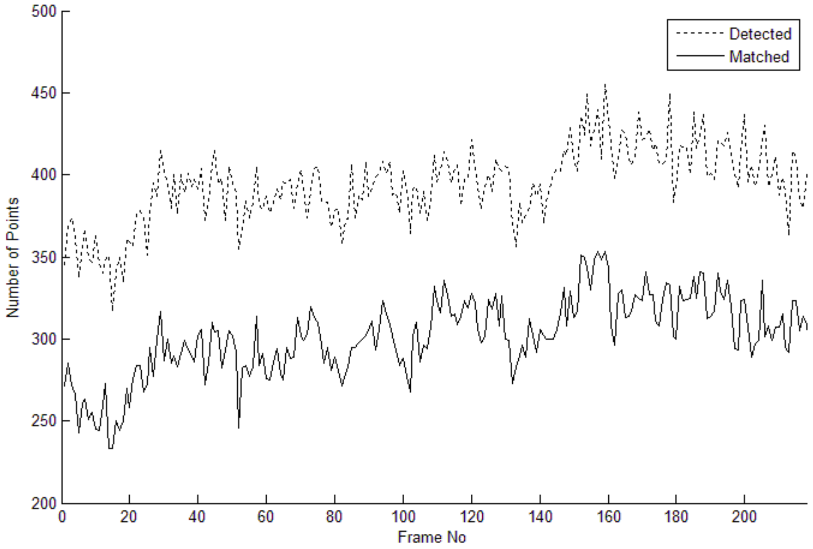


Figure B.13: The detection and matching results of “src10_ref_625.avi” for the third scheme.

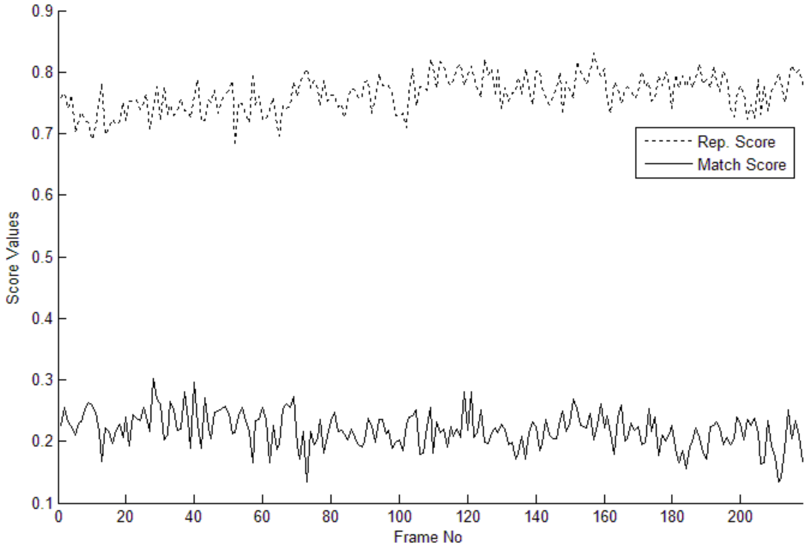


Figure B.14: The repetability and matching scores of “src10_ref_625.avi” for the third scheme.

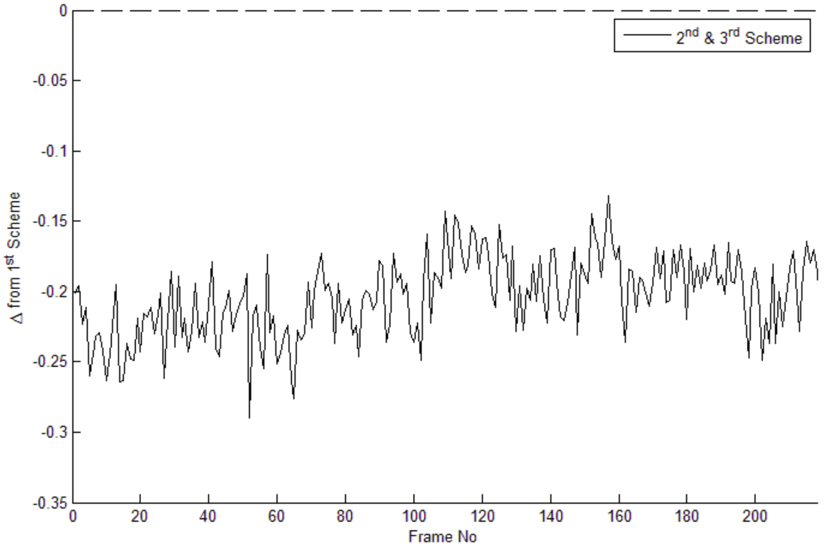


Figure B.15: The difference between repetability score of 1st scheme and 2nd or 3rd schemes for "src10_ref_625.avi".

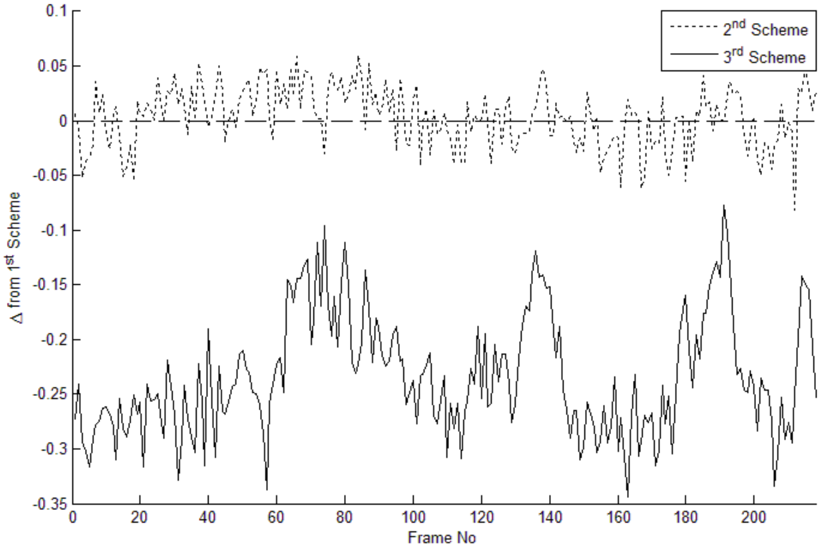


Figure B.16: The difference between match scores of 1st scheme and 2nd and 3rd schemes for "src10_ref_625.avi".

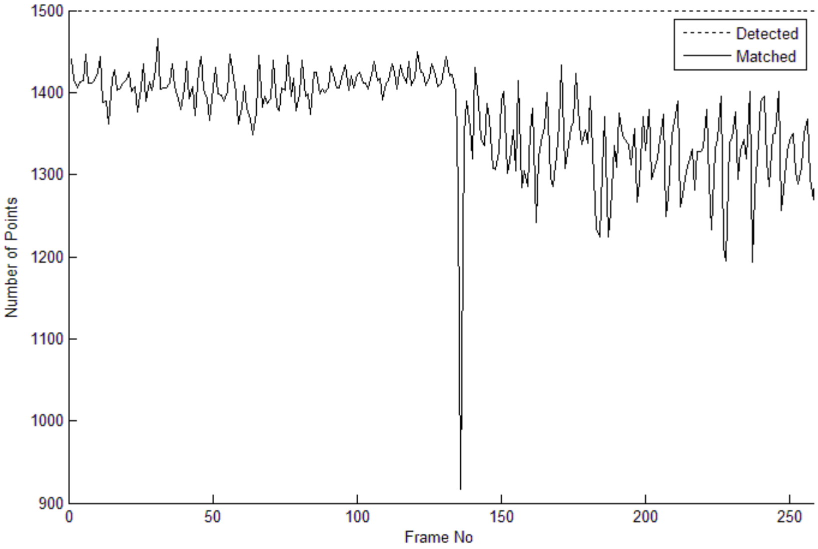


Figure B.17: The detection and matching results of “src13_ref_525.avi” for the first scheme.

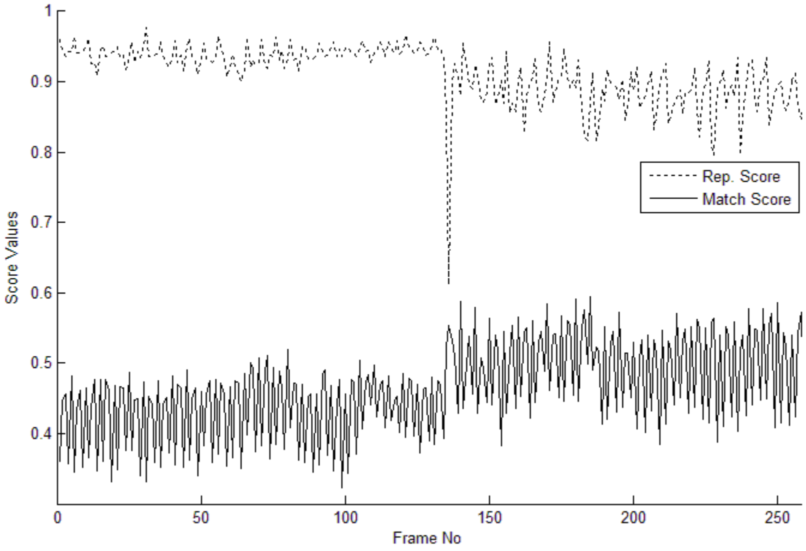


Figure B.18: The repetability and matching scores of “src13_ref_525.avi” for the first scheme.

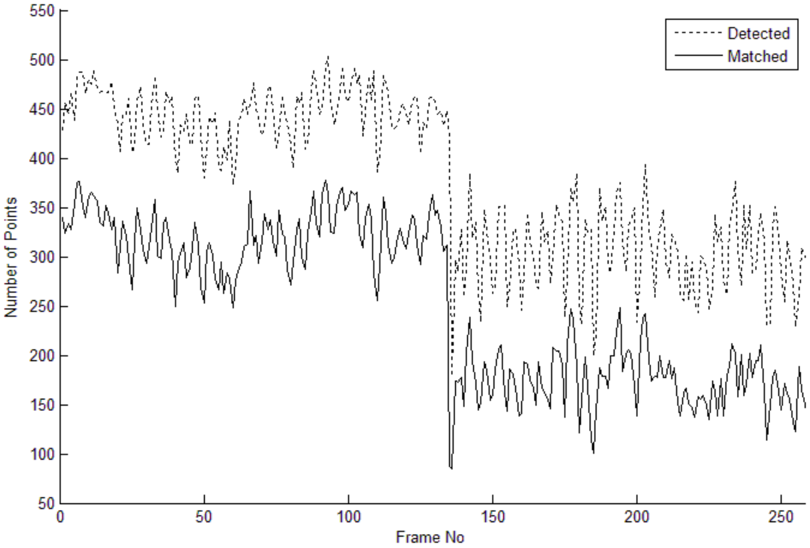


Figure B.19: The detection and matching results of “src13_ref_525.avi” for the second scheme.

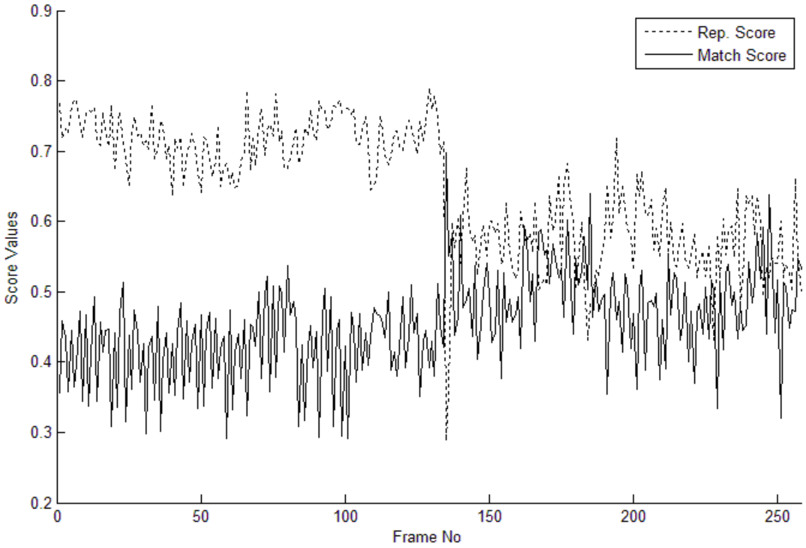


Figure B.20: The repetability and matching scores of “src13_ref_525.avi” for the second scheme.

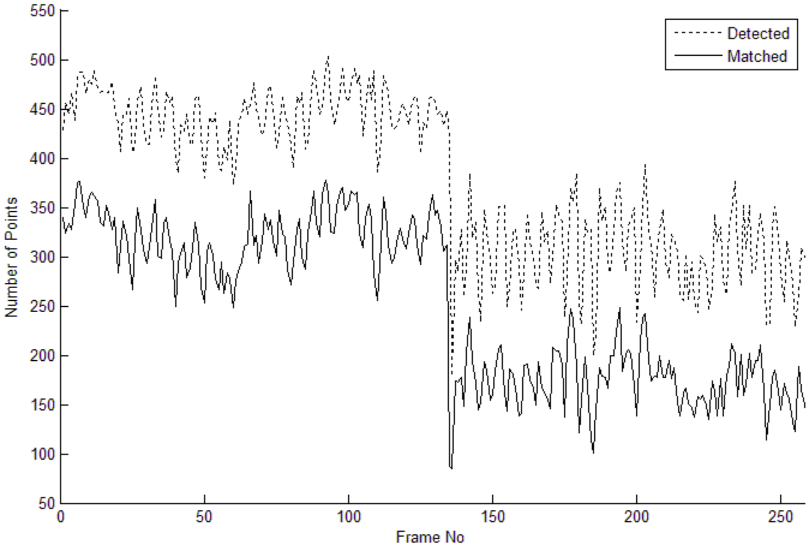


Figure B.21: The detection and matching results of “src13_ref_525.avi” for the third scheme.

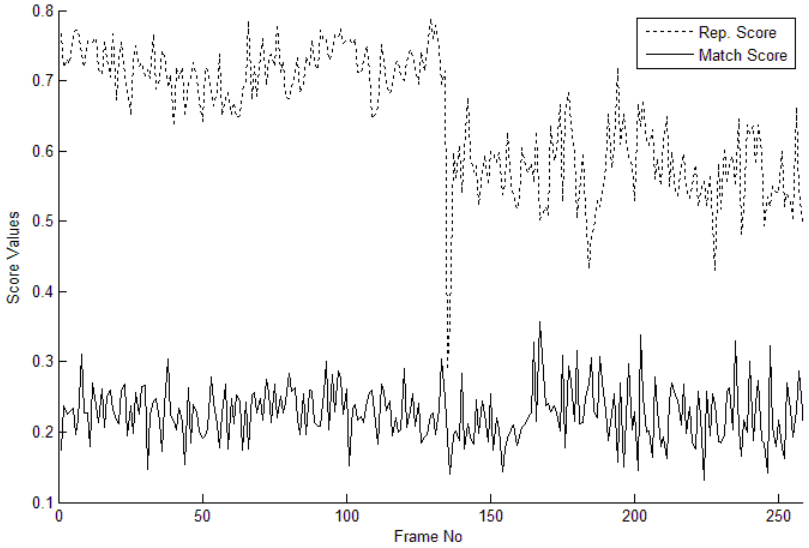


Figure B.22: The repetability and matching scores of “src13_ref_525.avi” for the third scheme.

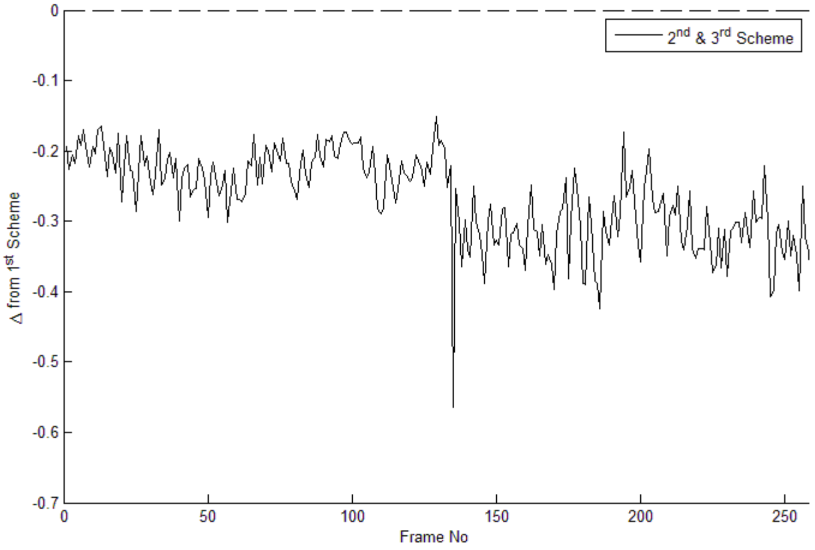


Figure B.23: The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for "src13_ref_525".

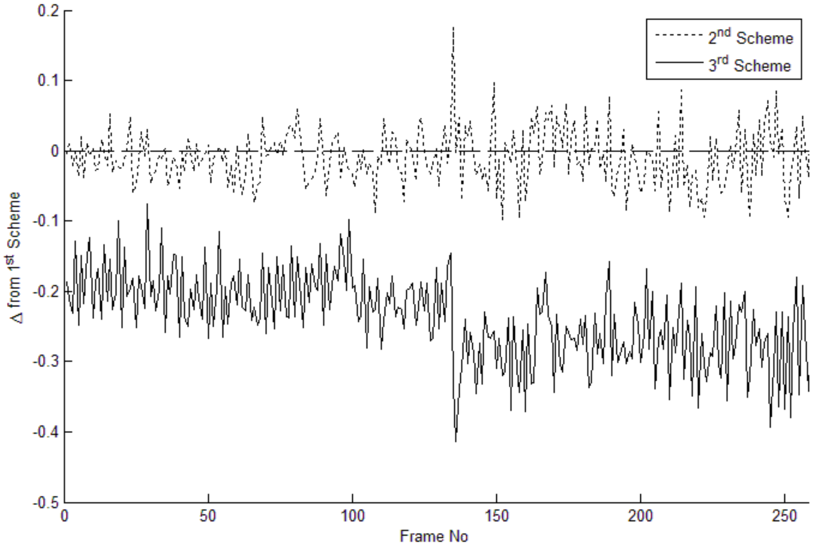


Figure B.24: The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for "src13_ref_525".

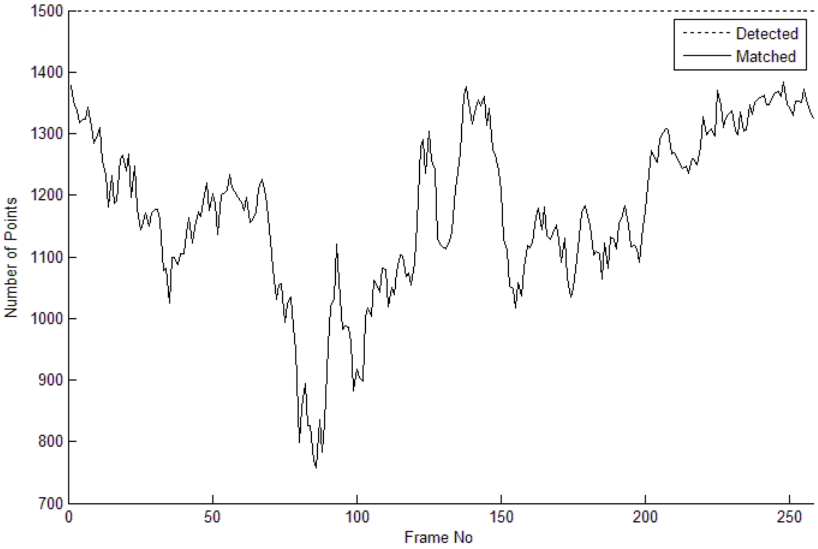


Figure B.25: The detection and matching results of “src19_ref_525.avi” for the first scheme.

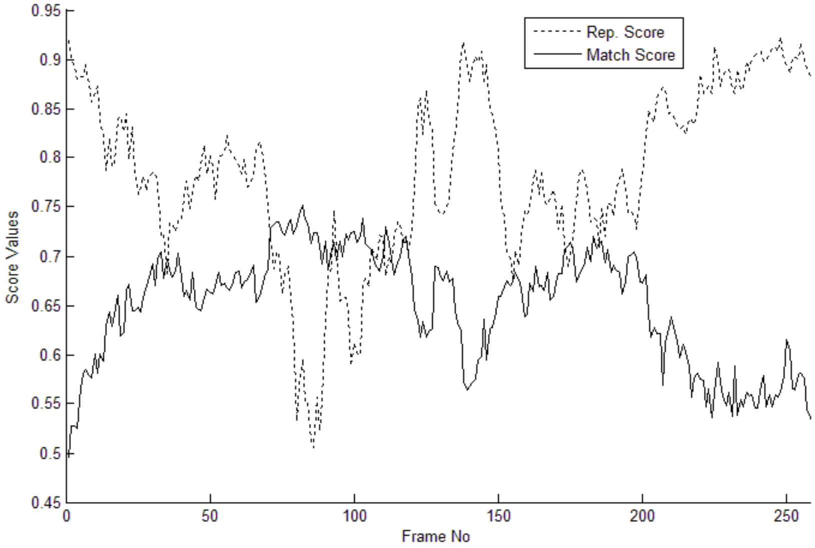


Figure B.26: The repetability and matching scores of “src19_ref_525.avi” for the first scheme.

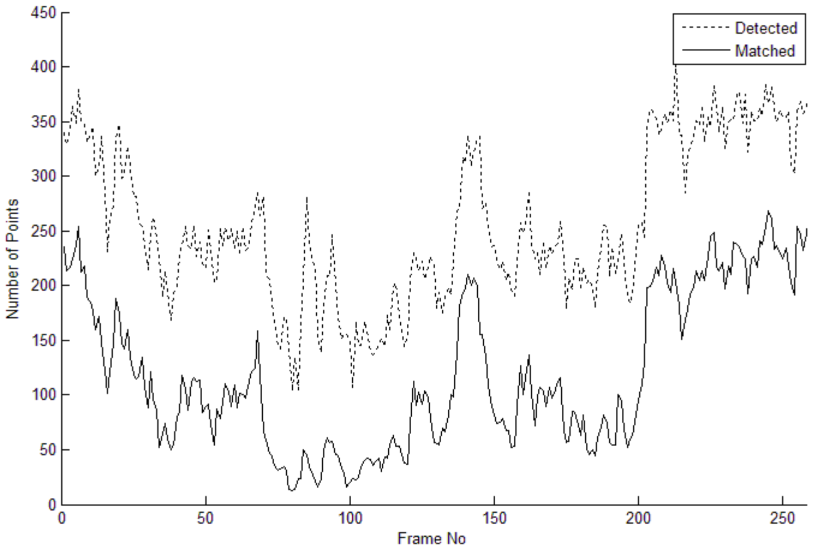


Figure B.27: The detection and matching results of “src19_ref_525.avi” for the second scheme.

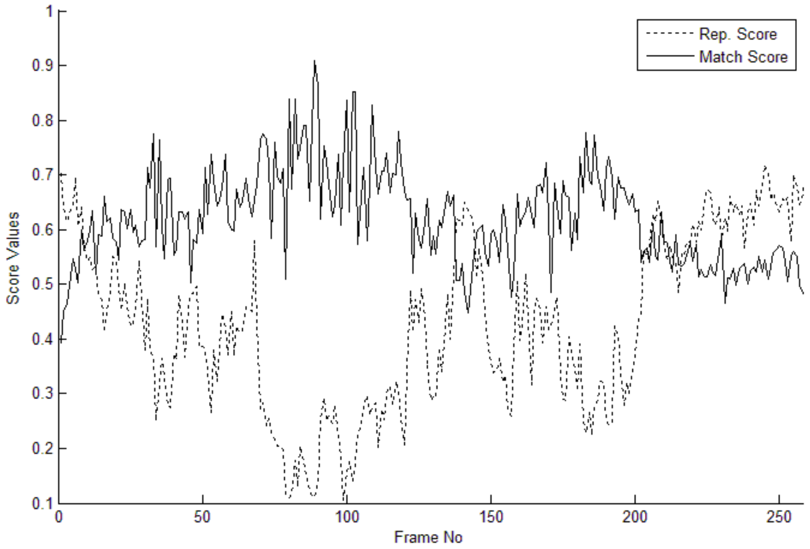


Figure B.28: The repetability and matching scores of “src19_ref_525.avi” for the second scheme.

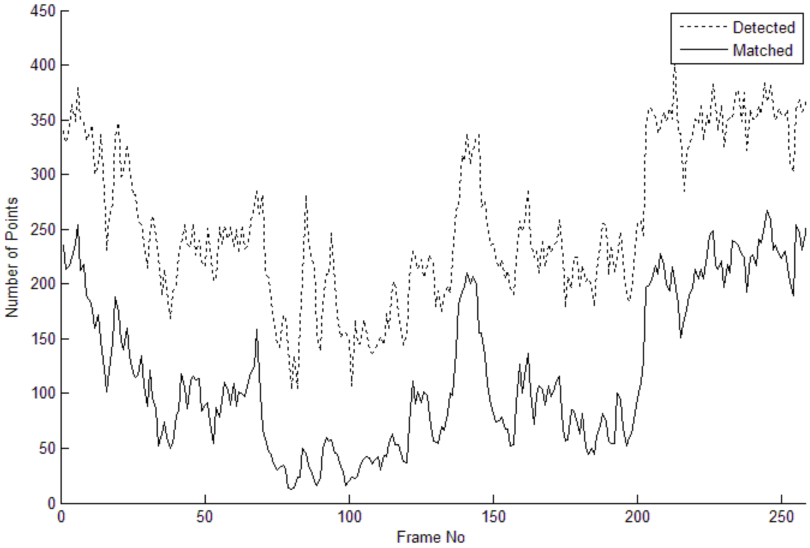


Figure B.29: The detection and matching results of “src19_ref_525.avi” for the third scheme.

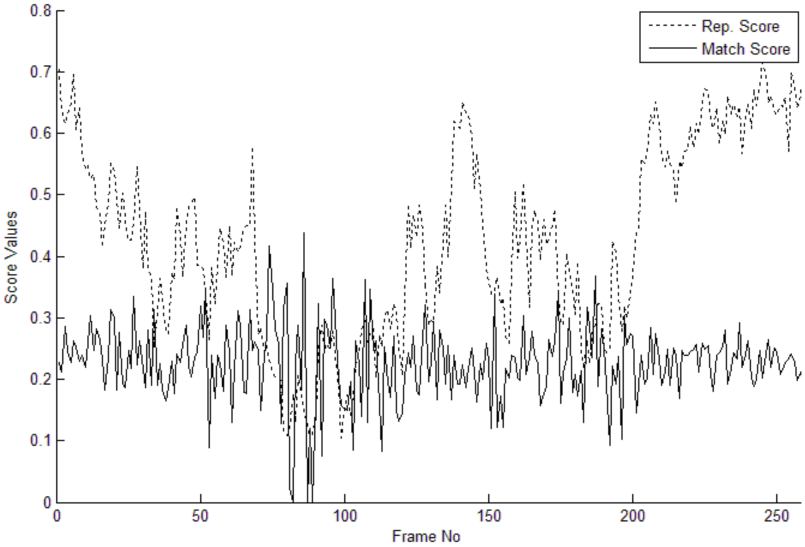


Figure B.30: The repetability and matching scores of “src19_ref_525.avi” for the third scheme.

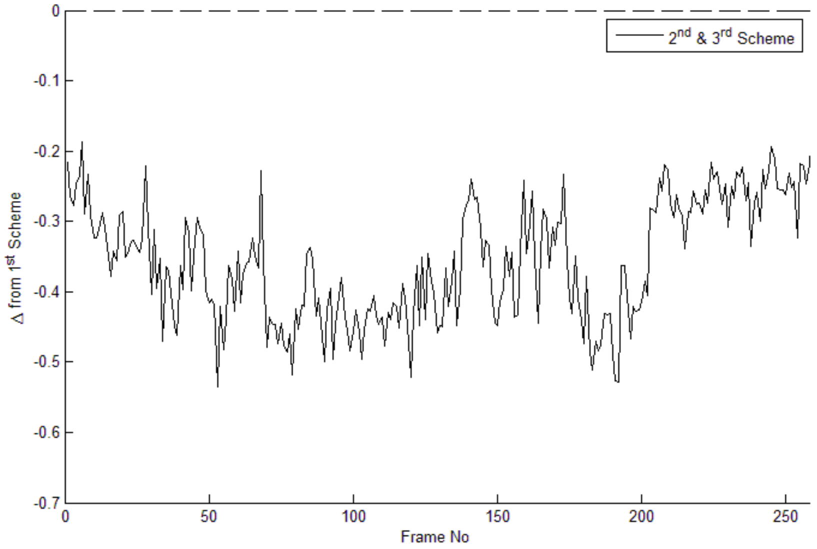


Figure B.31: The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for "src19_ref_525".

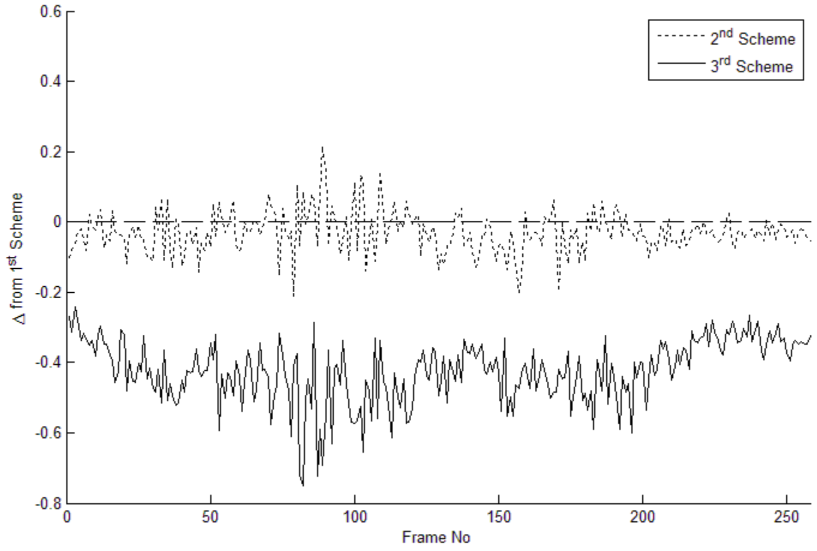


Figure B.32: The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for "src19_ref_525".

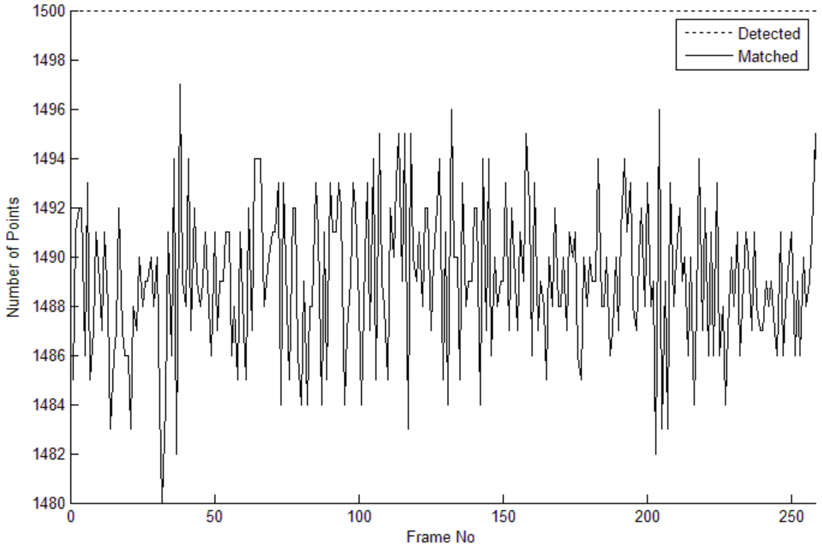


Figure B.33: The detection and matching results of “src20_ref_525.avi” for the first scheme.

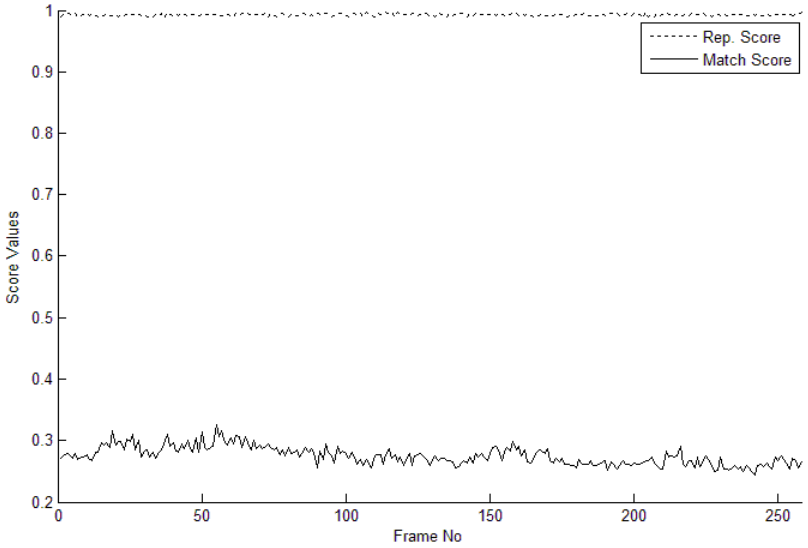


Figure B.34: The repetability and matching scores of “src20_ref_525.avi” for the first scheme.

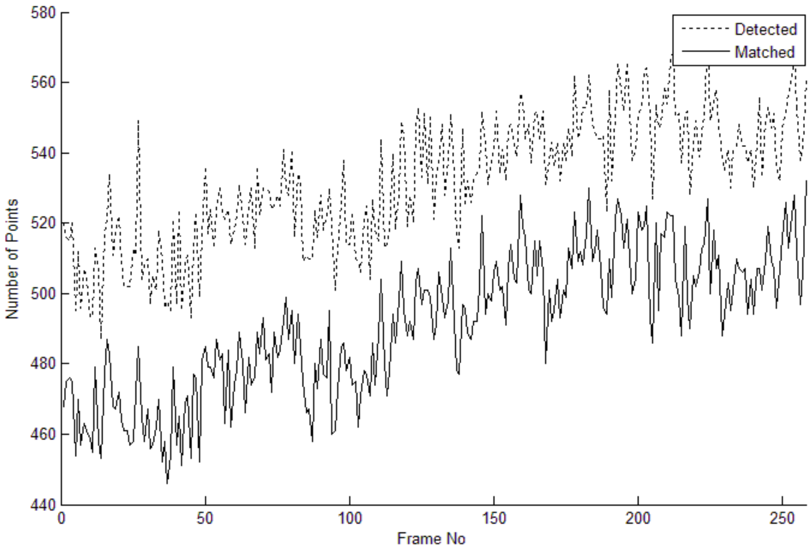


Figure B.35: The detection and matching results of “src20_ref_525.avi” for the second scheme.

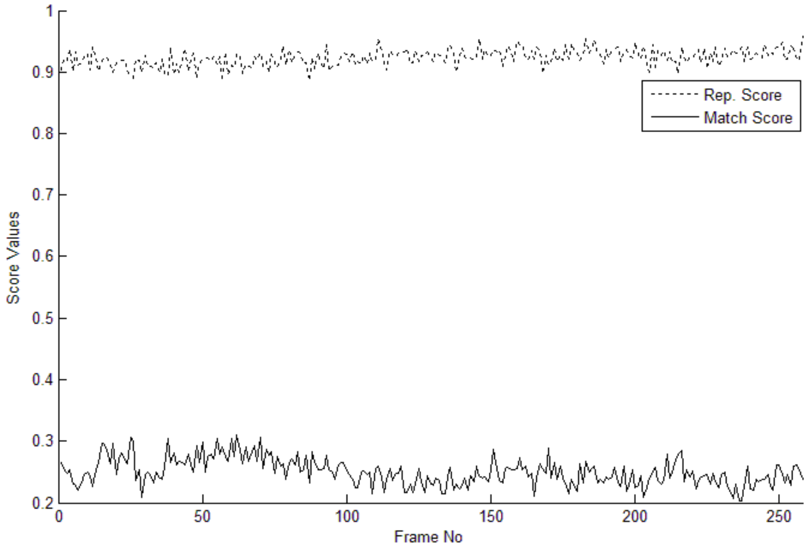


Figure B.36: The repetability and matching scores of “src20_ref_525.avi” for the second scheme.

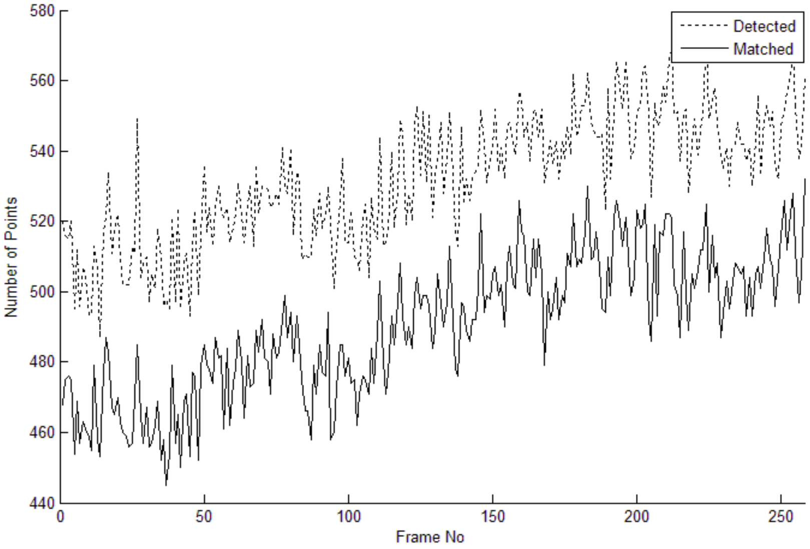


Figure B.37: The detection and matching results of "src20_ref_525.avi" for the third scheme.

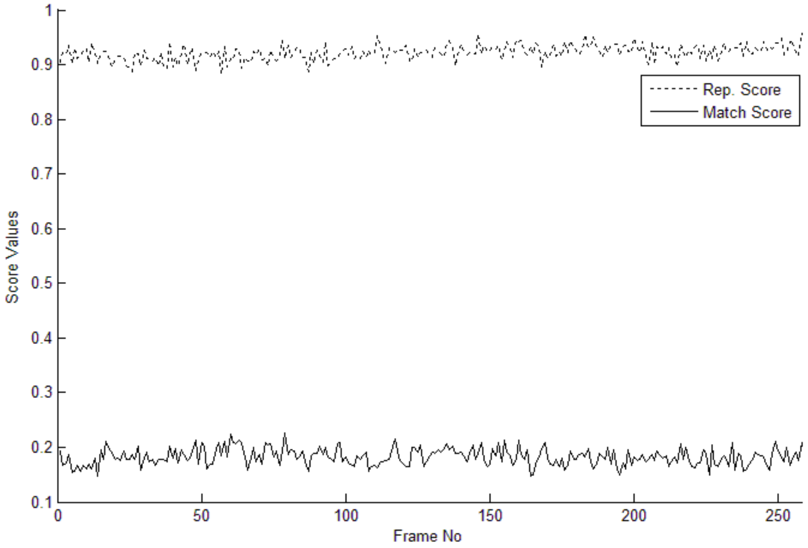


Figure B.38: The repetability and matching scores of "src20_ref_525.avi" for the third scheme.

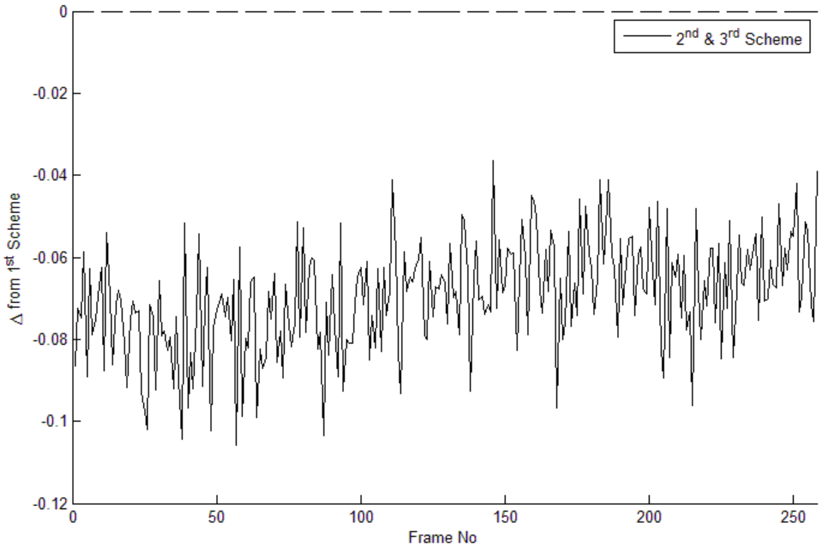


Figure B.39: The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for "src20_ref_525".

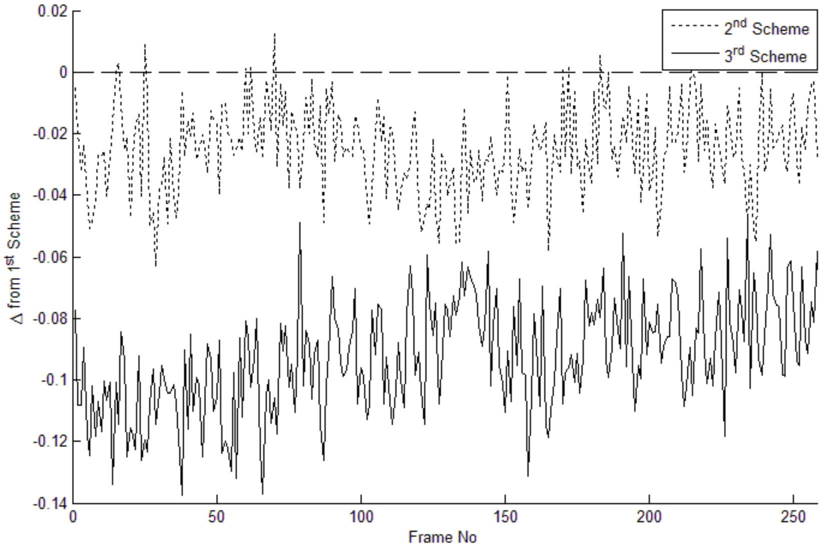


Figure B.40: The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for "src20_ref_525".

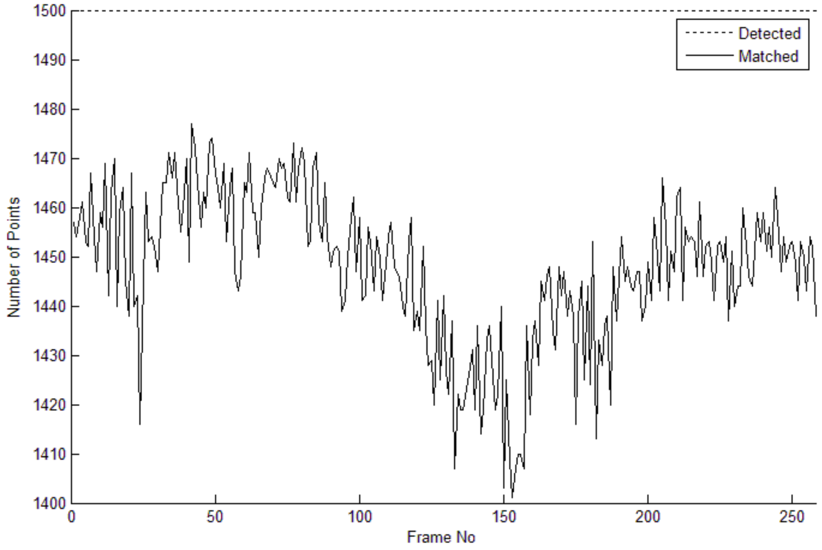


Figure B.41: The detection and matching results of "src22_ref_525.avi" for the first scheme.

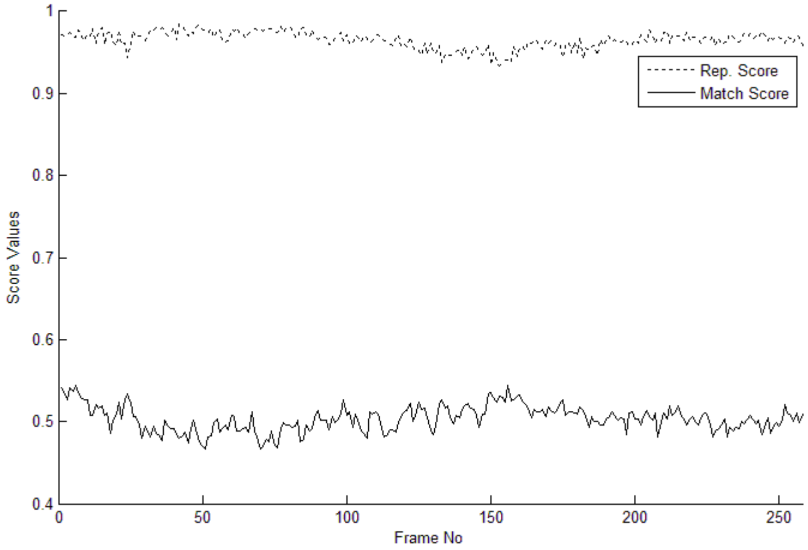


Figure B.42: The repetability and matching scores of "src22_ref_525.avi" for the first scheme.

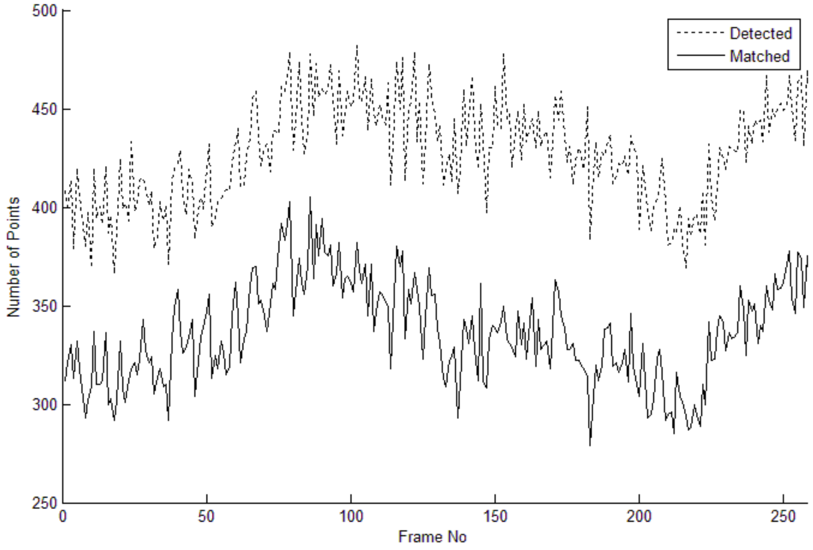


Figure B.43: The detection and matching results of "src22_ref_525.avi" for the second scheme.

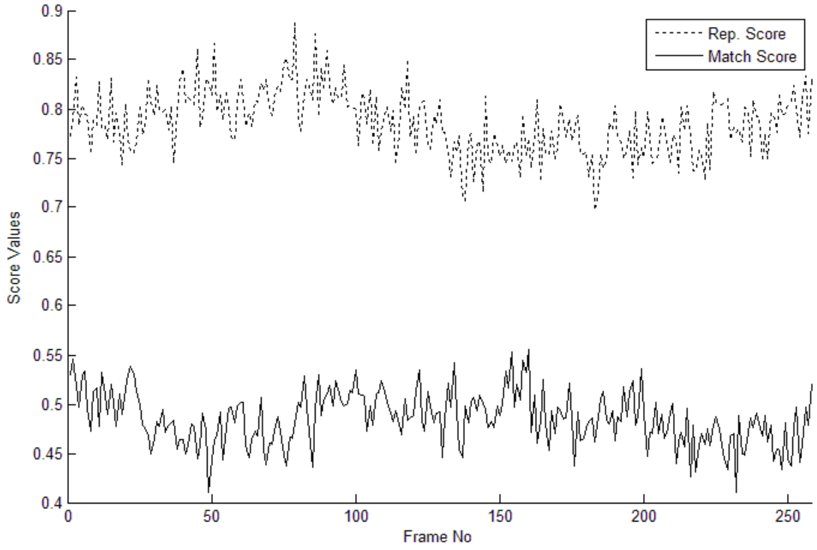


Figure B.44: The repetability and matching scores of "src22_ref_525.avi" for the second scheme.

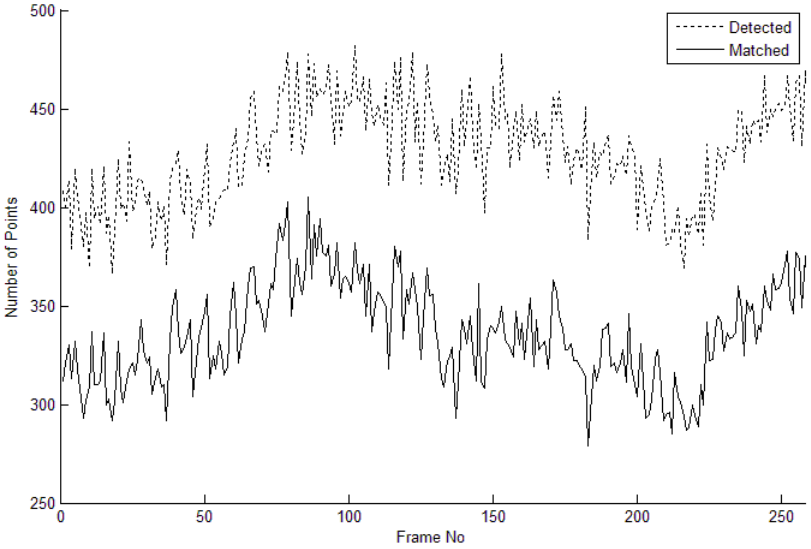


Figure B.45: The detection and matching results of “src22_ref_525.avi” for the third scheme.

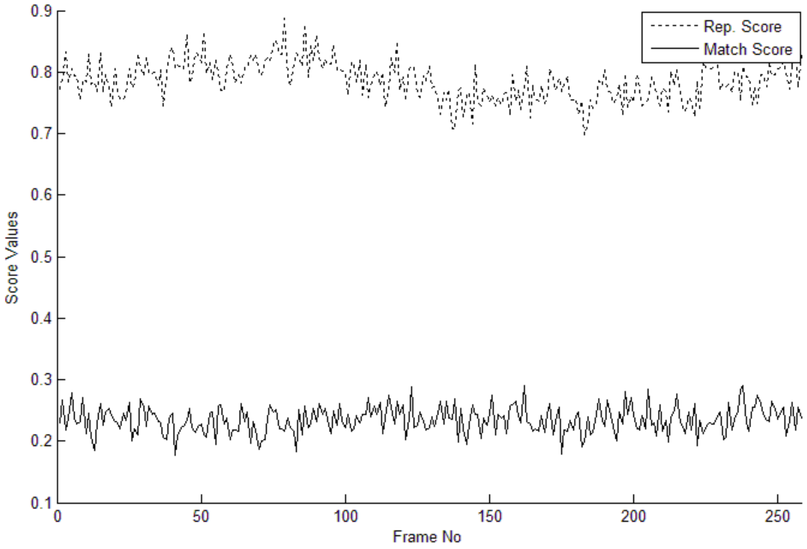


Figure B.46: The repetability and matching scores of “src22_ref_525.avi” for the third scheme.

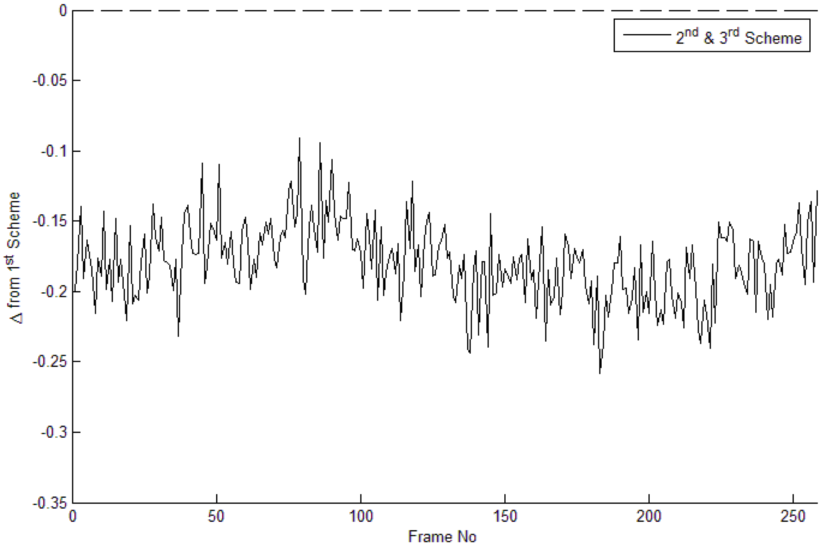


Figure B.47: The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for “src22_ref_525”.

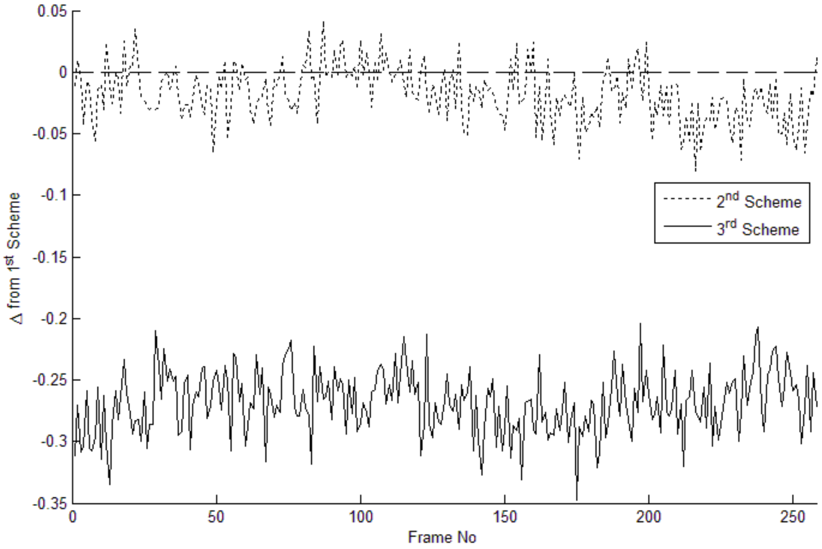


Figure B.48: The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for “src22_ref_525”.

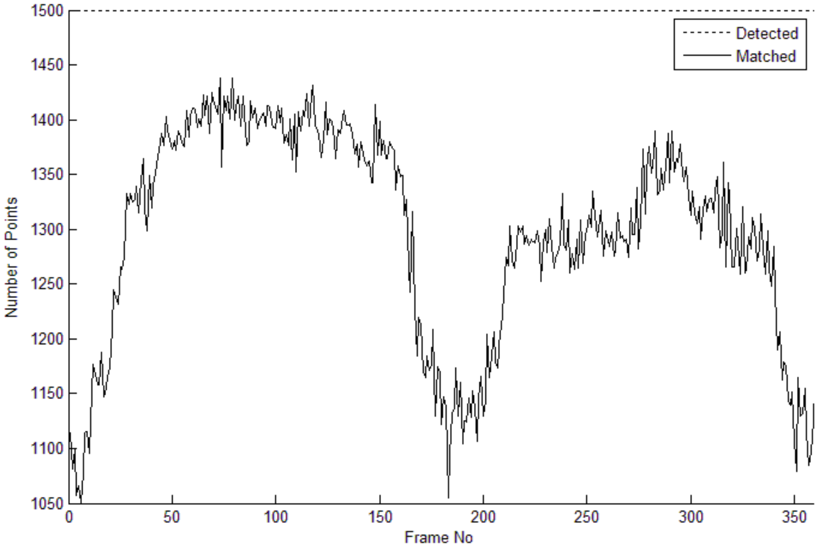


Figure B.49: The detection and matching results of "whale-shark_planetEarth_eps11.avi" for the first scheme.

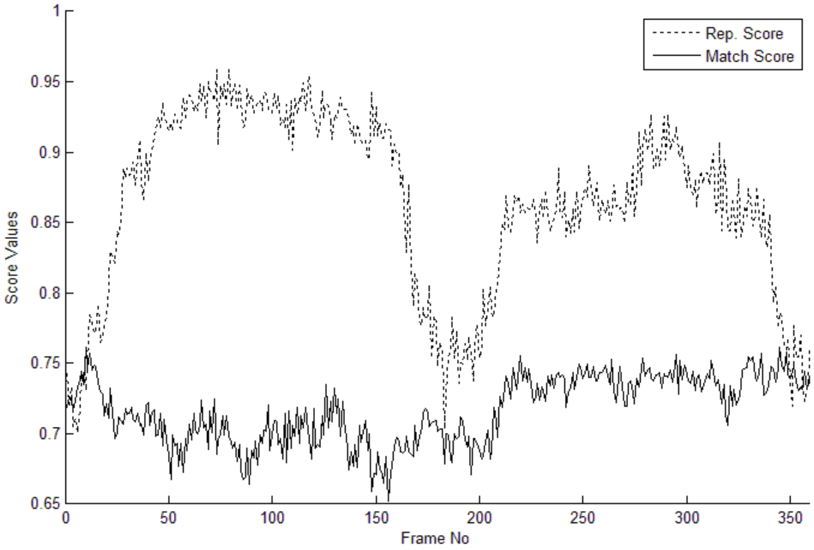


Figure B.50: The repetability and matching scores of "whale-shark_planetEarth_eps11.avi" for the first scheme.

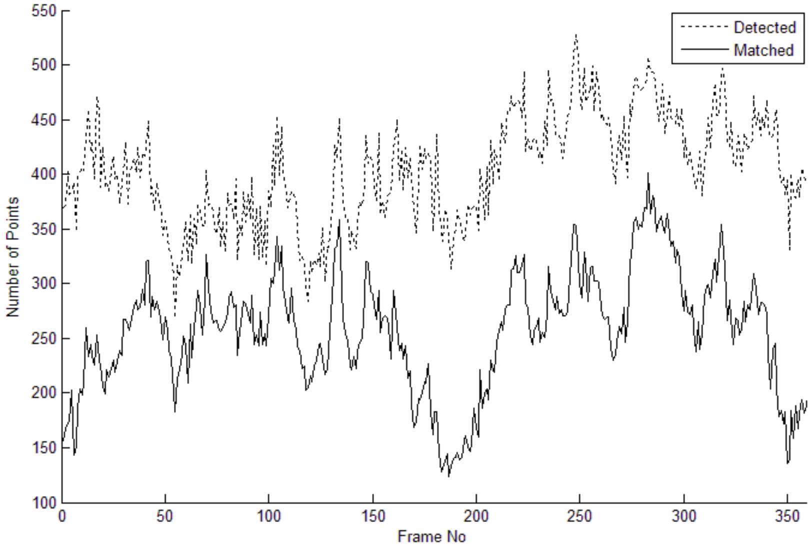


Figure B.51: The detection and matching results of "whale-shark_planetEarth_eps11.avi" for the second scheme.

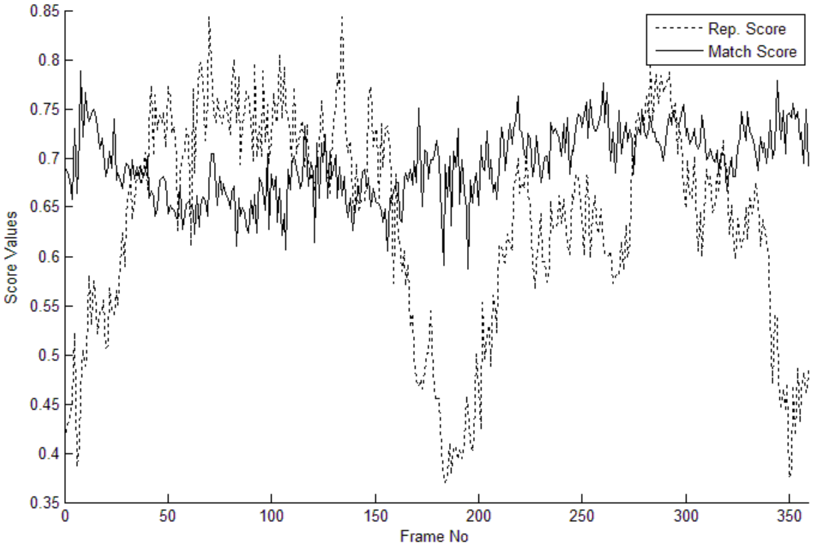


Figure B.52: The repetability and matching scores of "whale-shark_planetEarth_eps11.avi" for the second scheme.

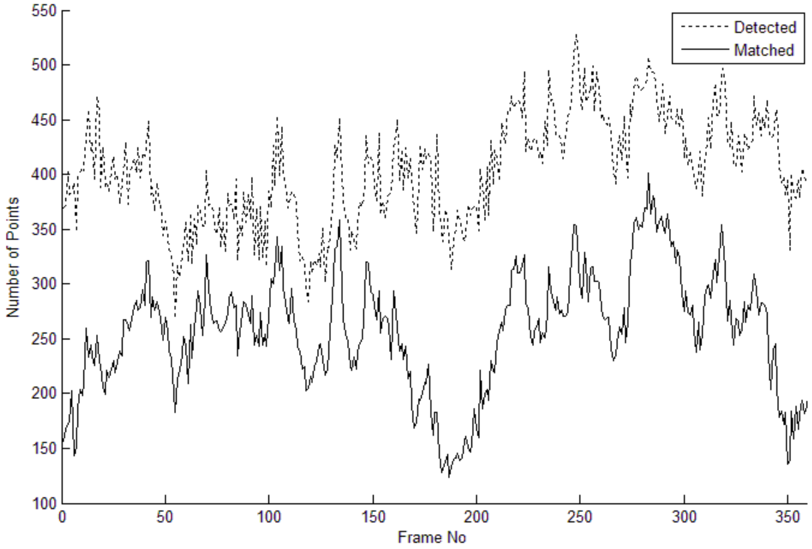


Figure B.53: The detection and matching results of "whale-shark_planetEarth_eps11.avi" for the third scheme.

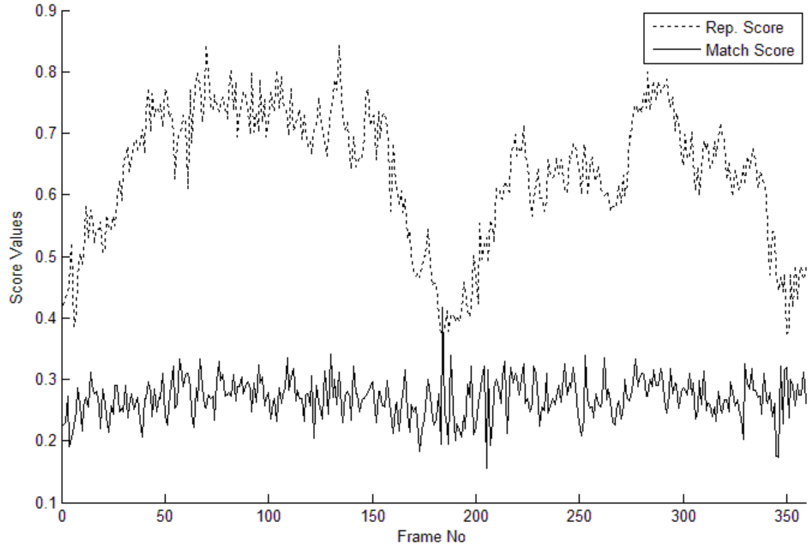


Figure B.54: The repetability and matching scores of "whale-shark_planetEarth_eps11.avi" for the third scheme.

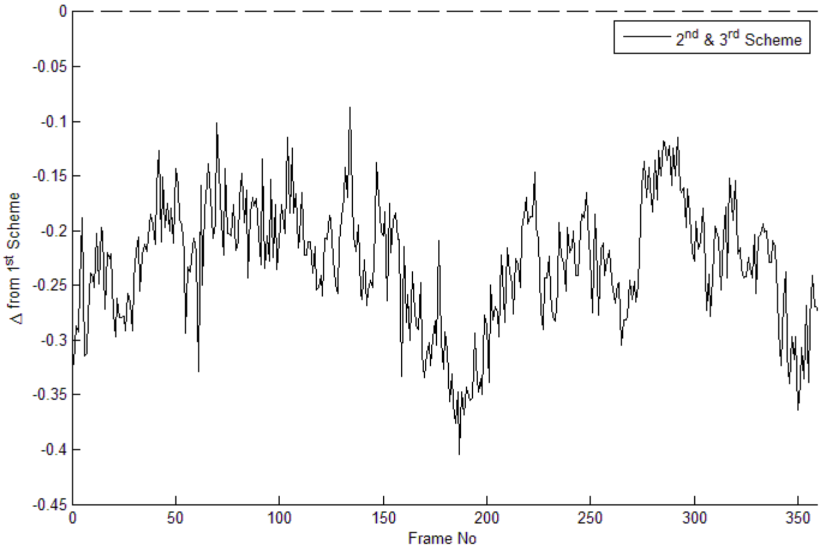


Figure B.55: The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for "whaleshark_planetEarth_eps11.avi".

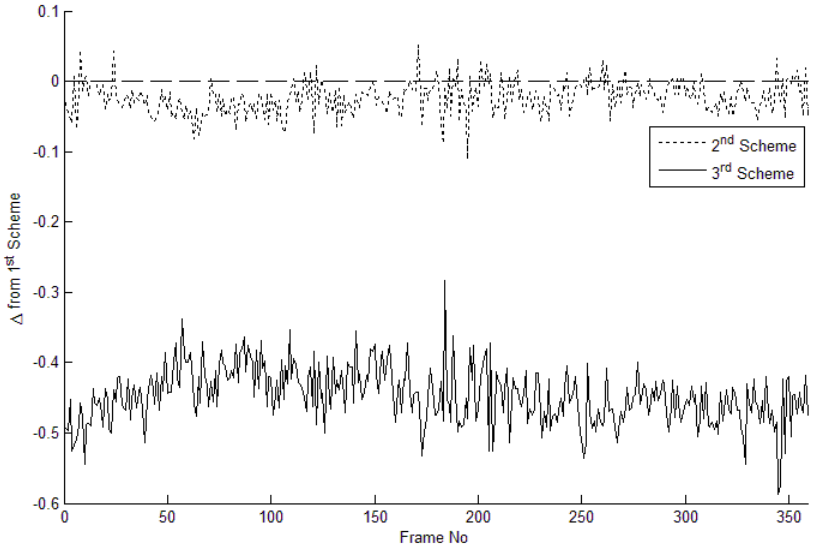


Figure B.56: The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for "whaleshark_planetEarth_eps11.avi".

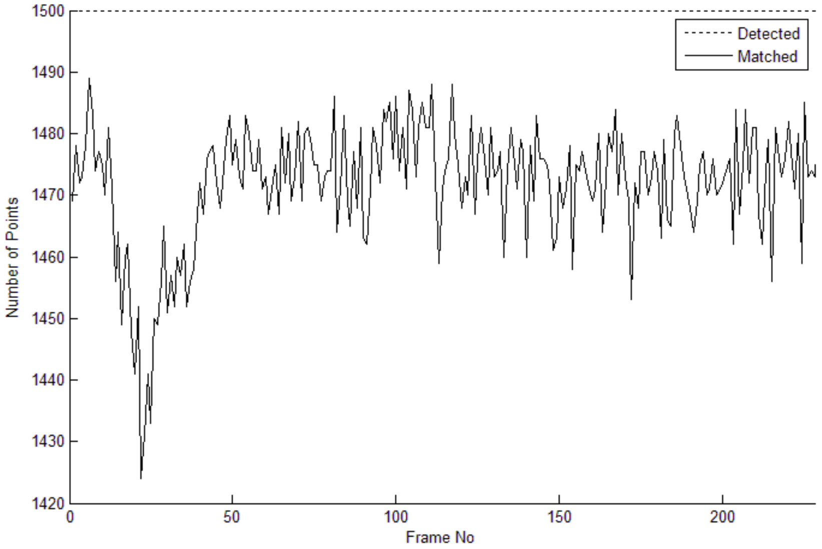


Figure B.57: The detection and matching results of ‘goats_planetEarth_eps5.avi” for the first scheme.

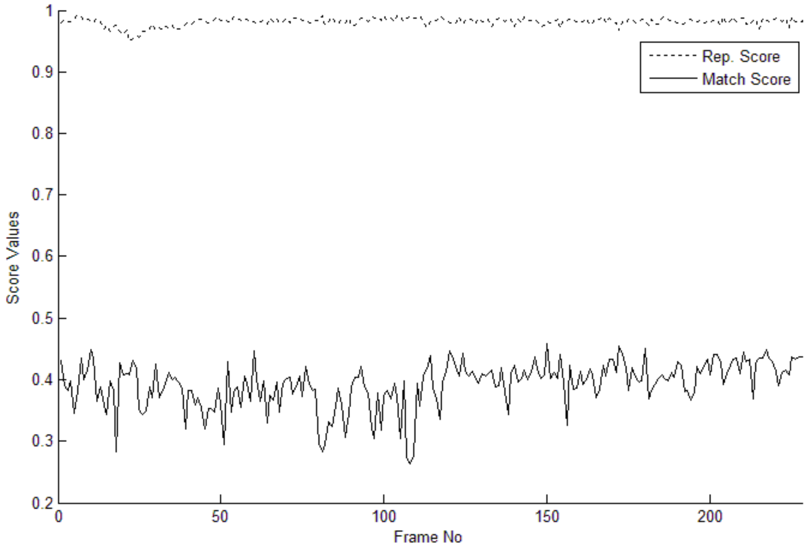


Figure B.58: The repetability and matching scores of “goats_planetEarth_eps5.avi” for the first scheme.

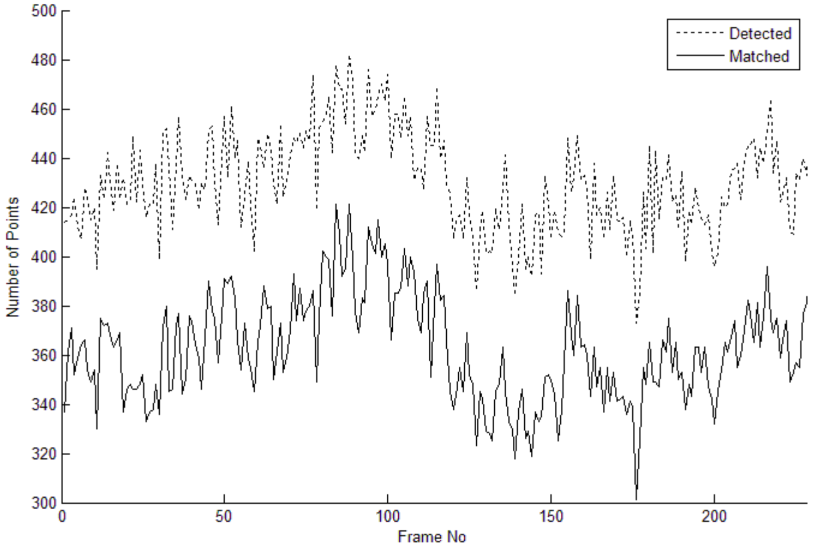


Figure B.59: The detection and matching results of “goats_planetEarth_eps5.avi” for the second scheme.

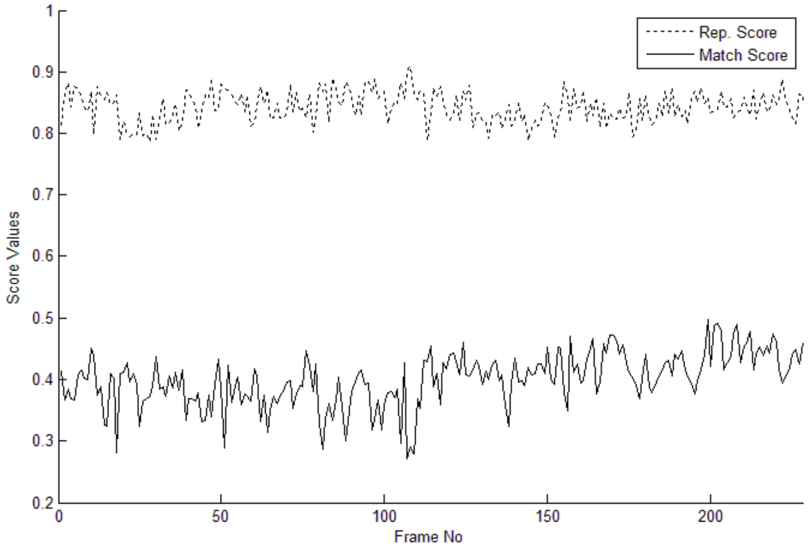


Figure B.60: The repetability and matching scores of “goats_planetEarth_eps5.avi” for the second scheme.

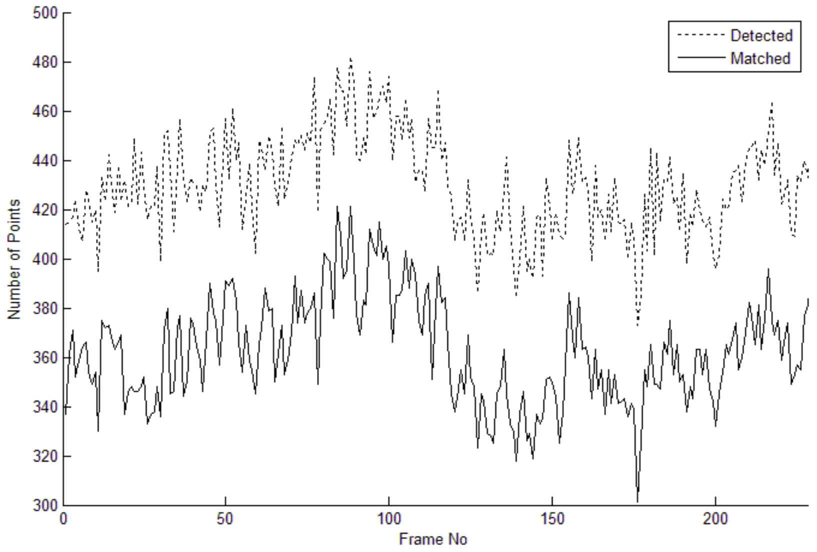


Figure B.61: The detection and matching results of "goats_planetEarth_eps5.avi" for the third scheme.

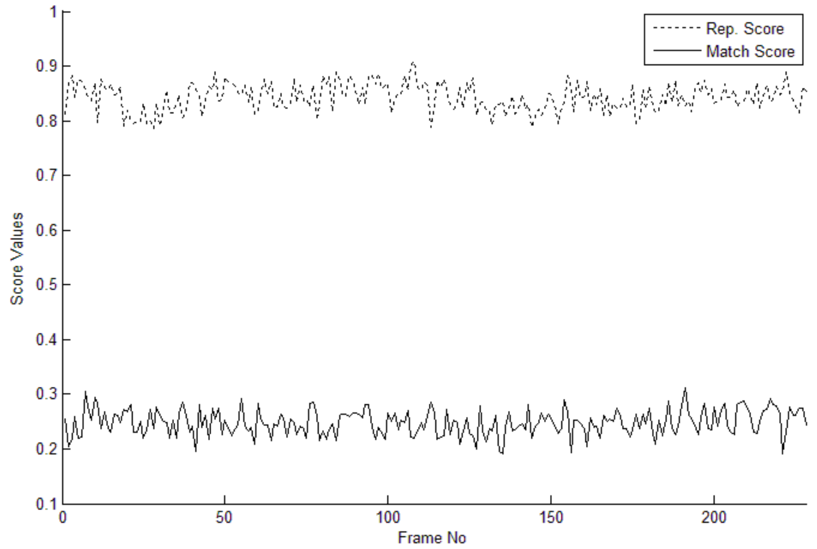


Figure B.62: The repetability and matching scores of "goats_planetEarth_eps5.avi" for the third scheme.

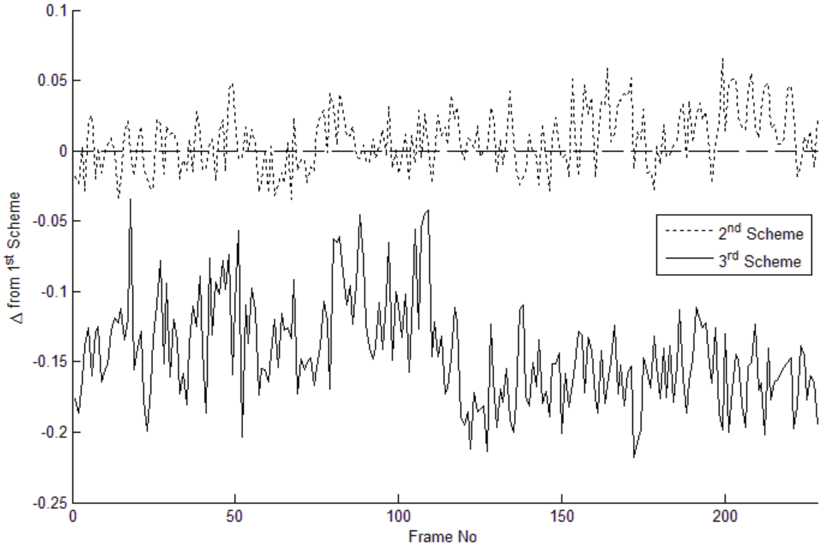


Figure B.63: The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for “goats_planetEarth_eps5.avi”.

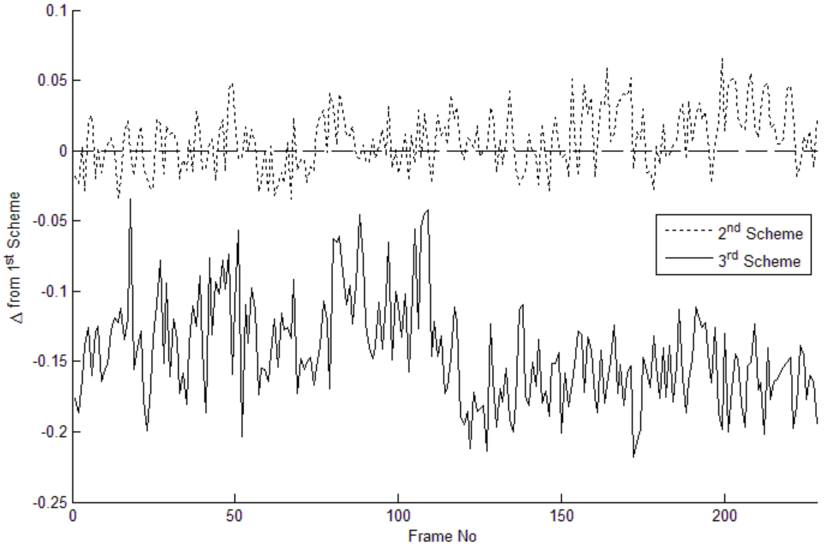


Figure B.64: The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for “goats_planetEarth_eps5.avi”.

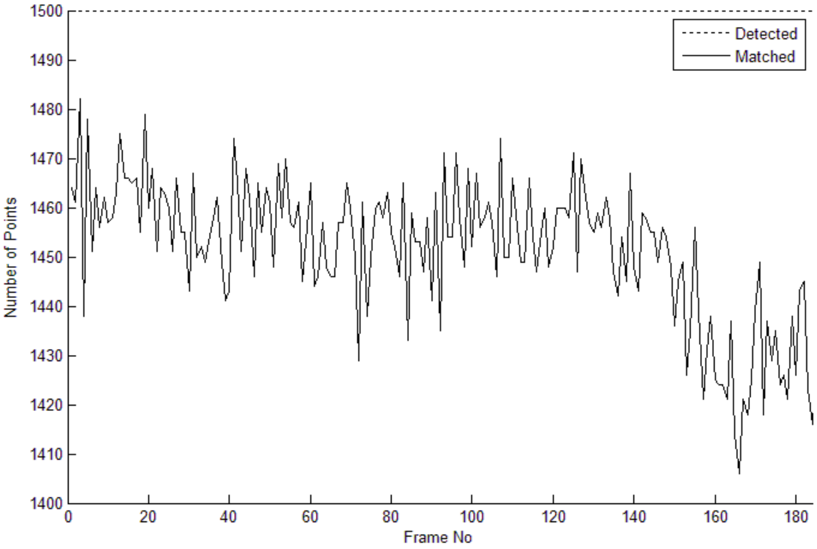


Figure B.65: The detection and matching results of "dolphins_planetEarth_eps9.avi" for the first scheme.

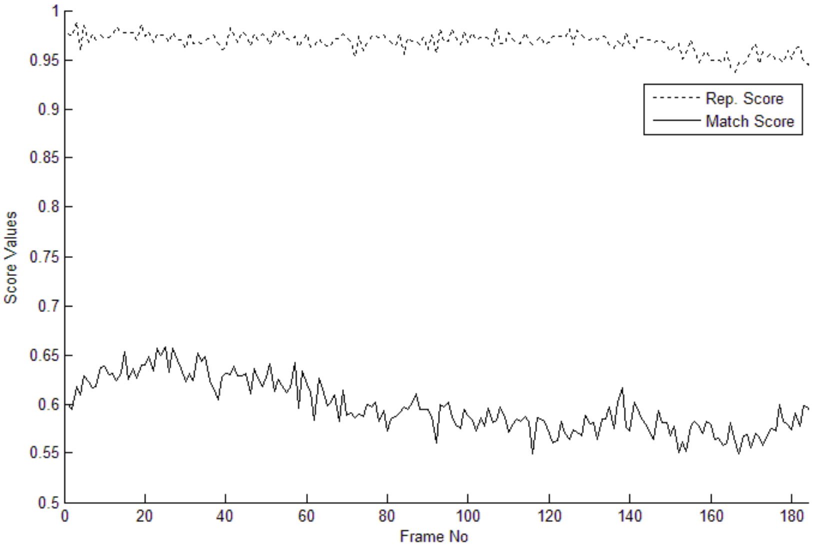


Figure B.66: The repetability and matching scores of "dolphins_planetEarth_eps9.avi" for the first scheme.

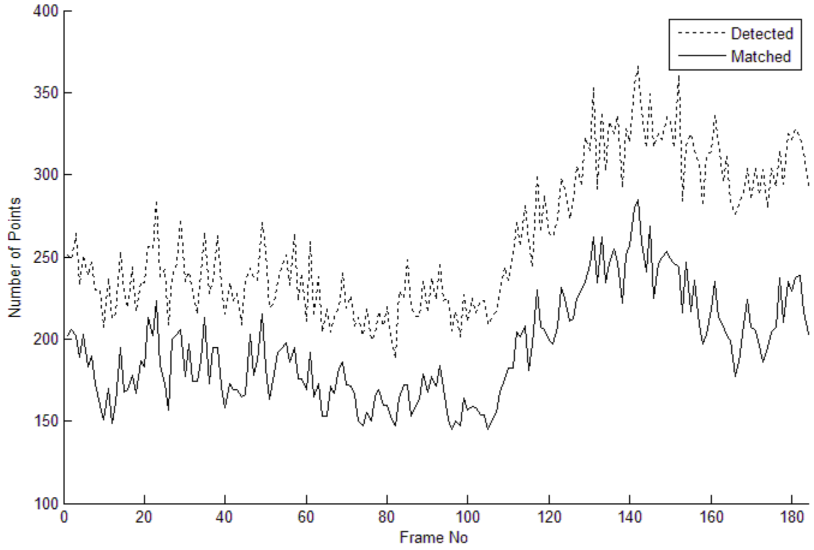


Figure B.67: The detection and matching results of "dolphins_planetEarth_eps9.avi" for the second scheme.

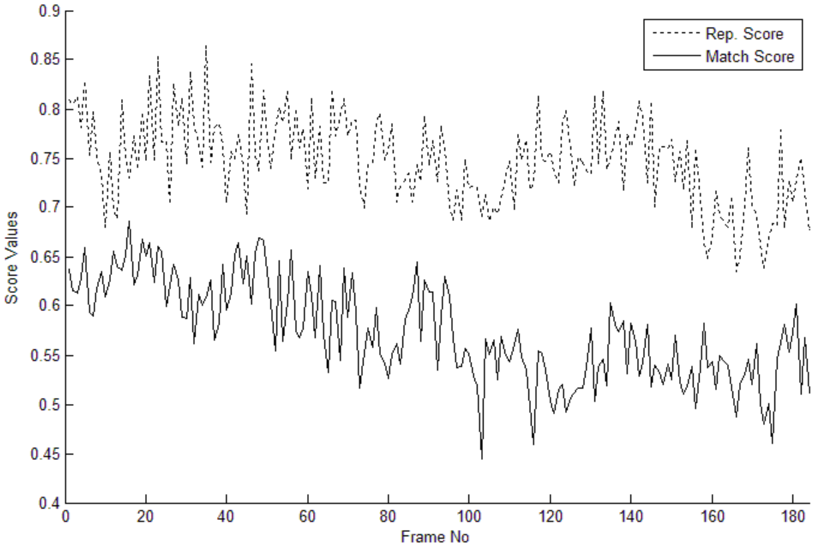


Figure B.68: The repetability and matching scores of "dolphins_planetEarth_eps9.avi" for the second scheme.

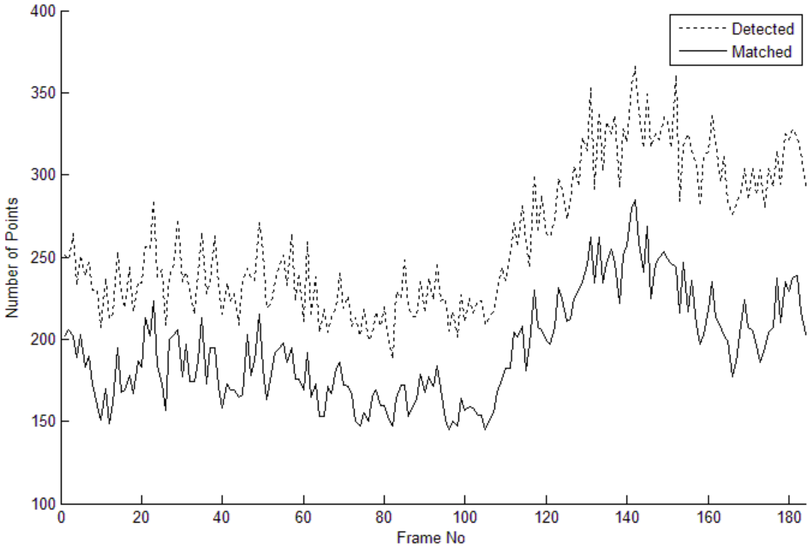


Figure B.69: The detection and matching results of "dolphins_planetEarth_eps9.avi" for the third scheme.

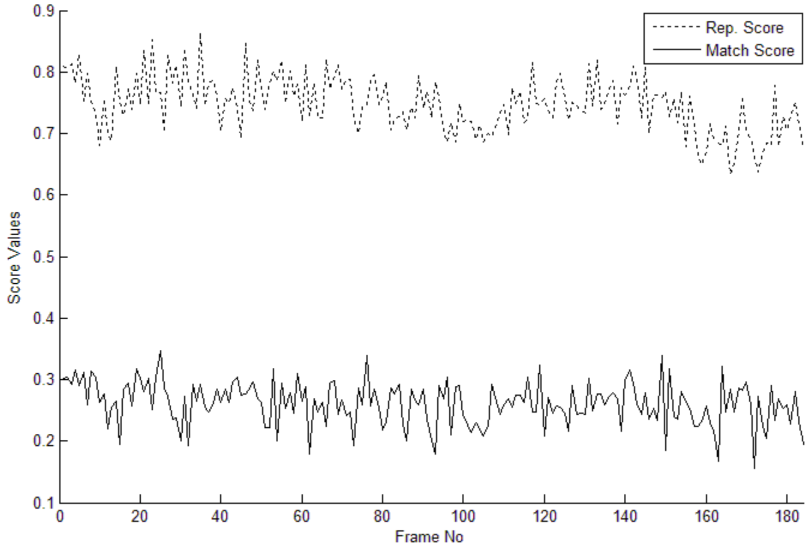


Figure B.70: The repetability and matching scores of "dolphins_planetEarth_eps9.avi" for the third scheme.

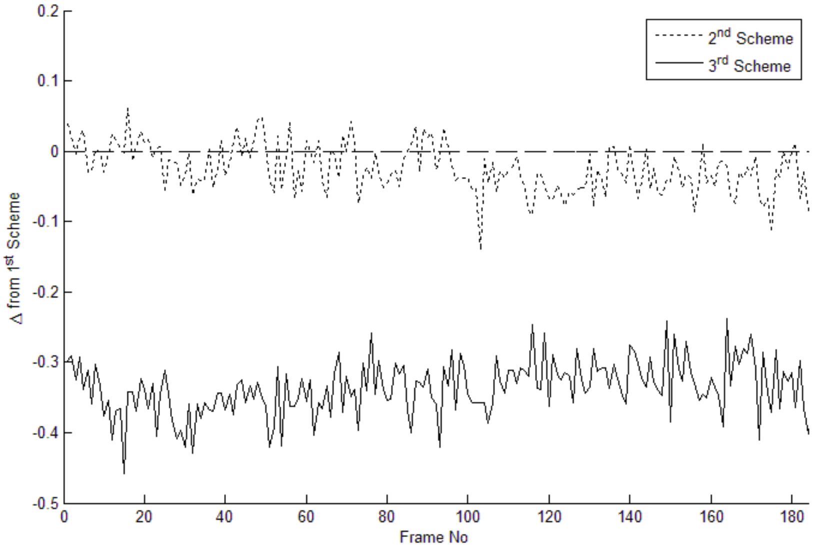


Figure B.71: The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for "dolphins_planetEarth_eps9.avi".

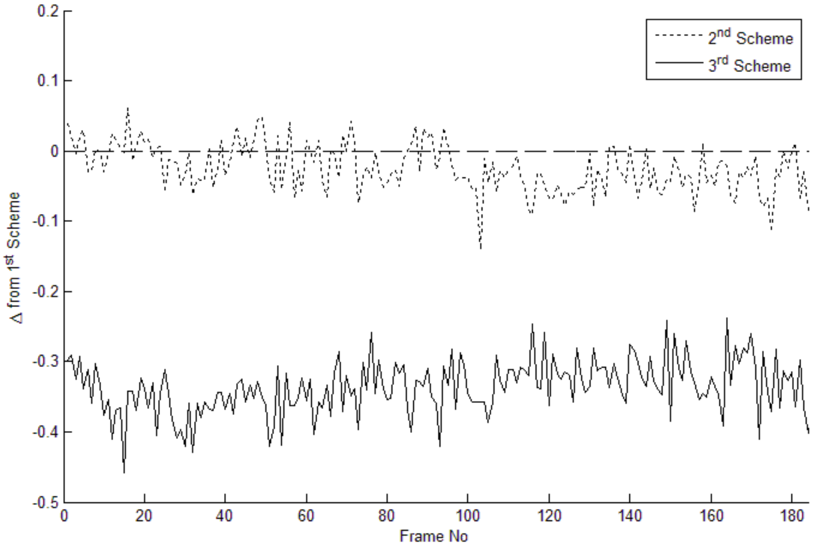


Figure B.72: The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for "dolphins_planetEarth_eps9.avi".

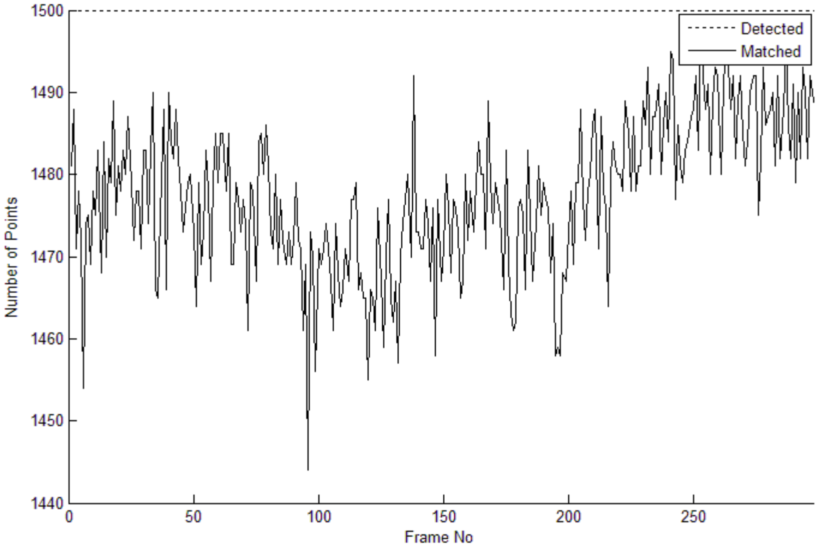


Figure B.73: The detection and matching results of "leopard_planetEarth_eps2.avi" for the first scheme.

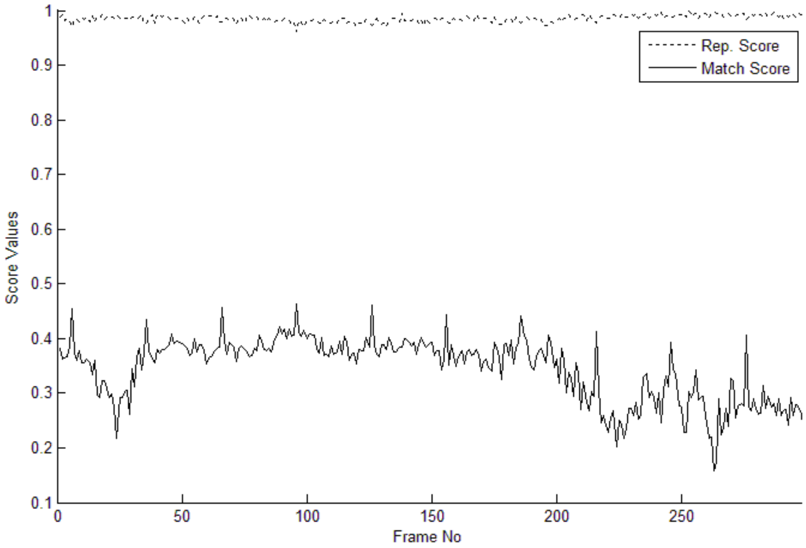


Figure B.74: The repetability and matching scores of "leopard_planetEarth_eps2.avi" for the first scheme.

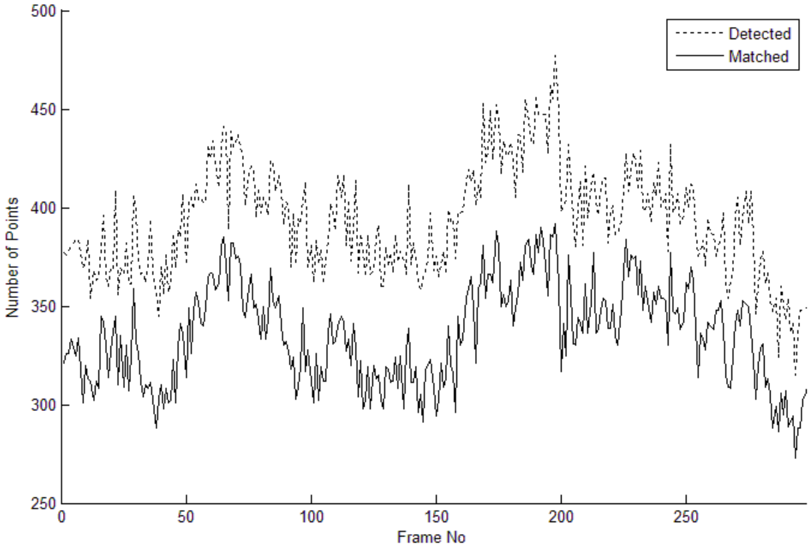


Figure B.75: The detection and matching results of "leopard_planetEarth_eps2.avi" for the second scheme.

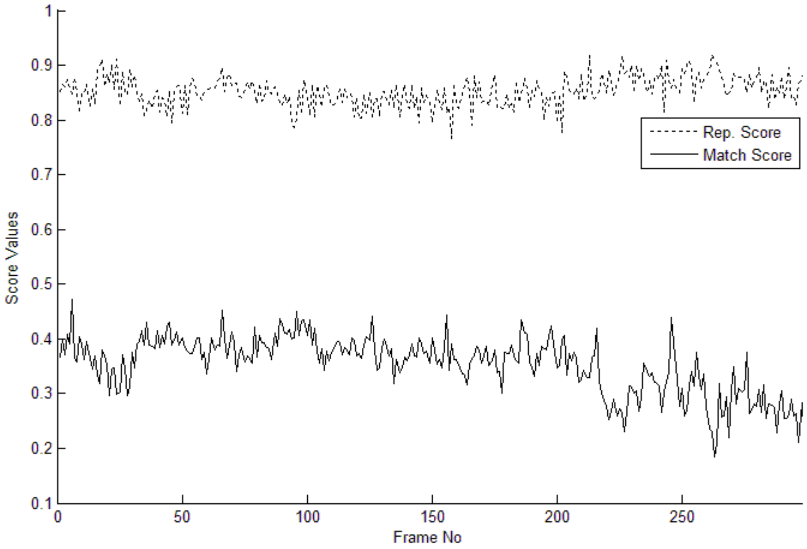


Figure B.76: The repetability and matching scores of "leopard_planetEarth_eps2.avi" for the second scheme.

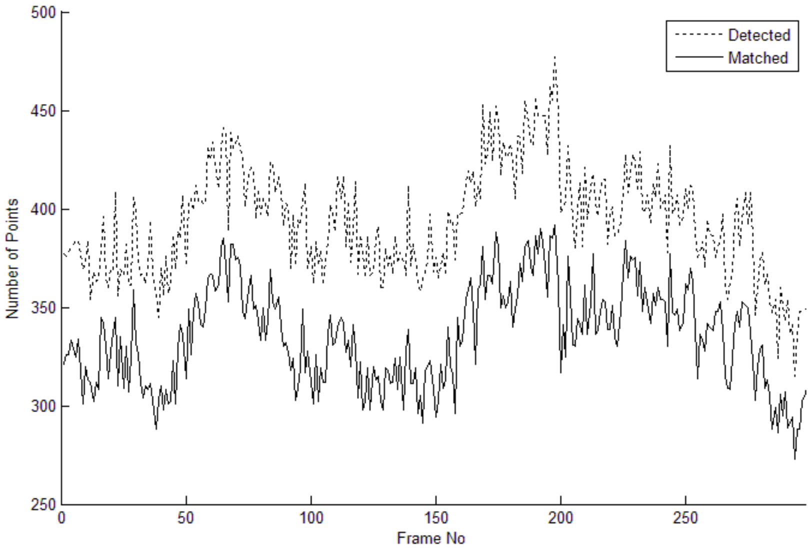


Figure B.77: The detection and matching results of “leopard_planetEarth_eps2.avi” for the third scheme.

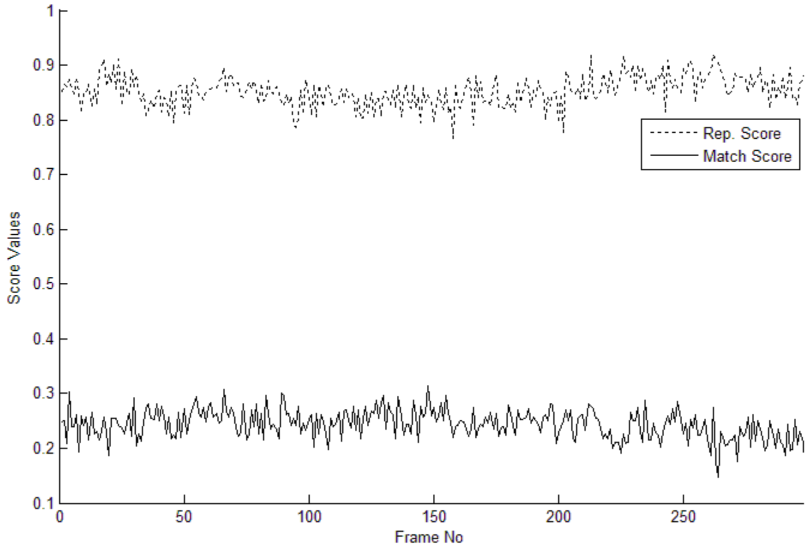


Figure B.78: The repetability and matching scores of “leopard_planetEarth_eps2.avi” for the third scheme.

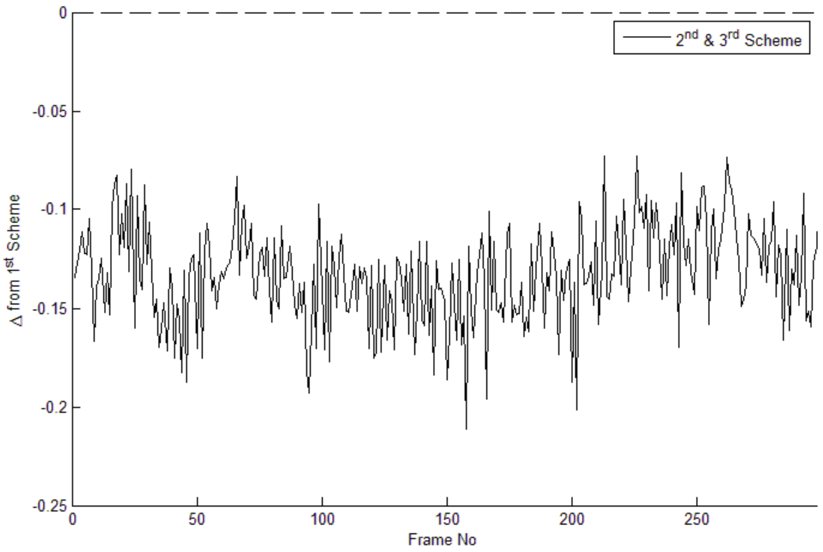


Figure B.79: The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for "leopard_planetEarth_eps2.avi".

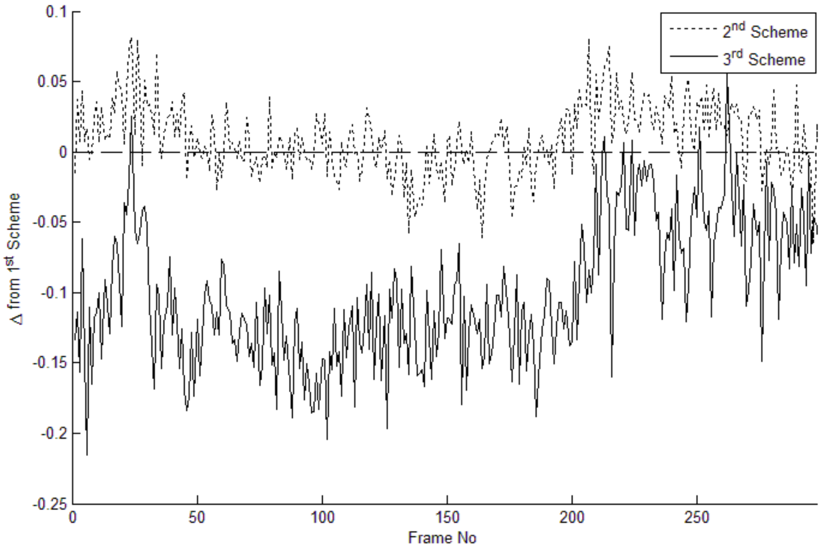


Figure B.80: The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for "leopard_planetEarth_eps2.avi".

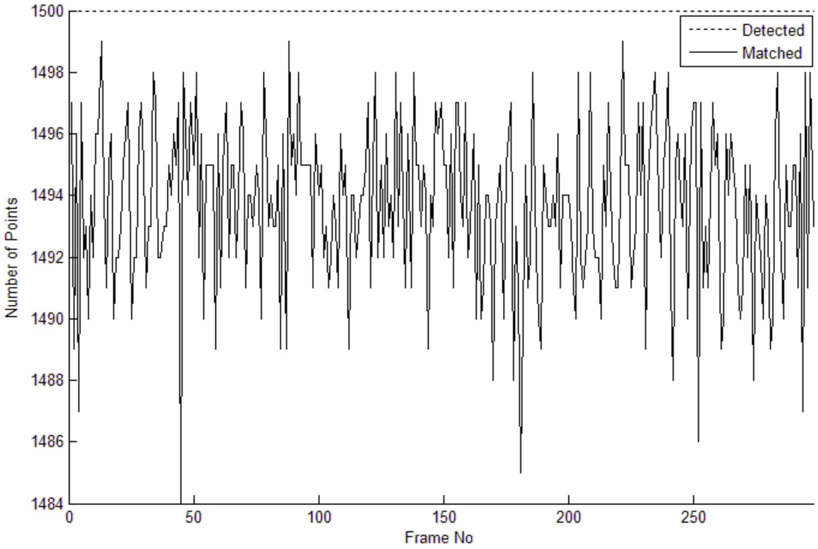


Figure B.81: The detection and matching results of “container.avi” for the first scheme.

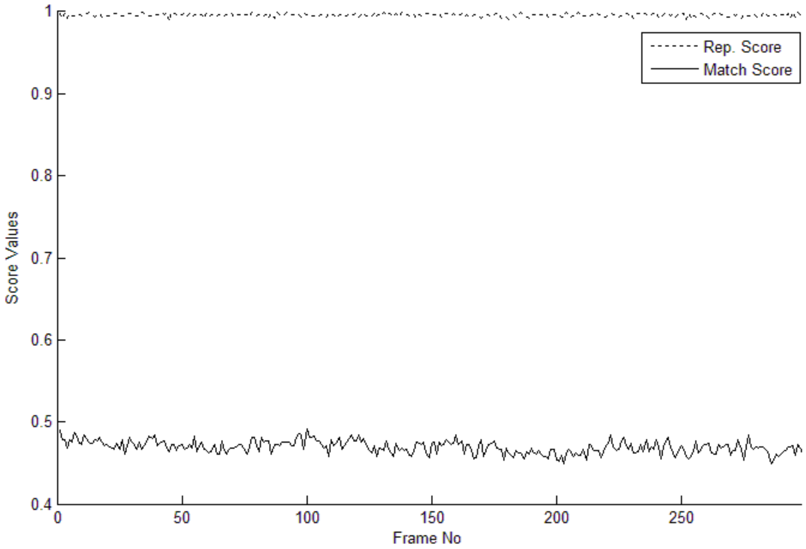


Figure B.82: The repetability and matching scores of “container.avi” for the first scheme.

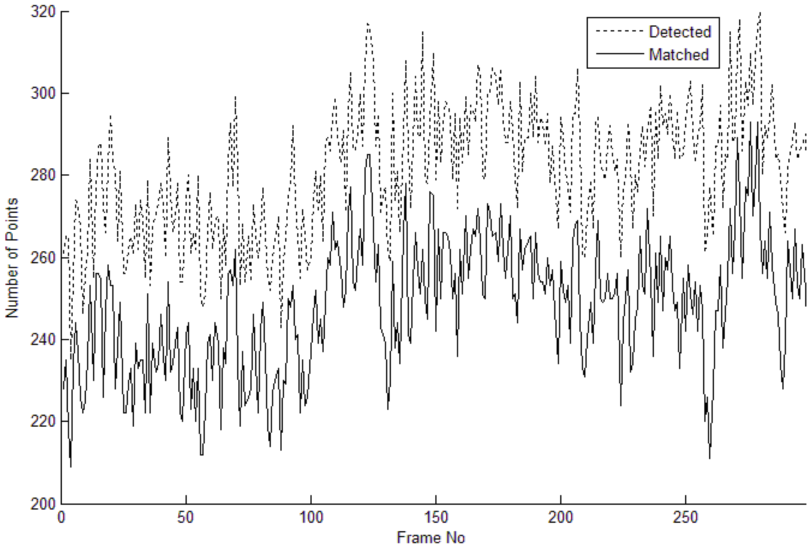


Figure B.83: The detection and matching results of “container.avi” for the second scheme.

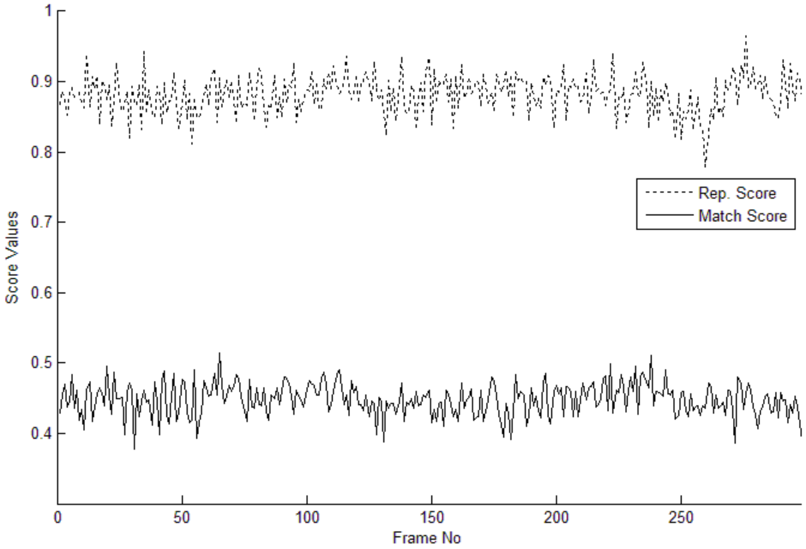


Figure B.84: The repetability and matching scores of “container.avi” for the second scheme.

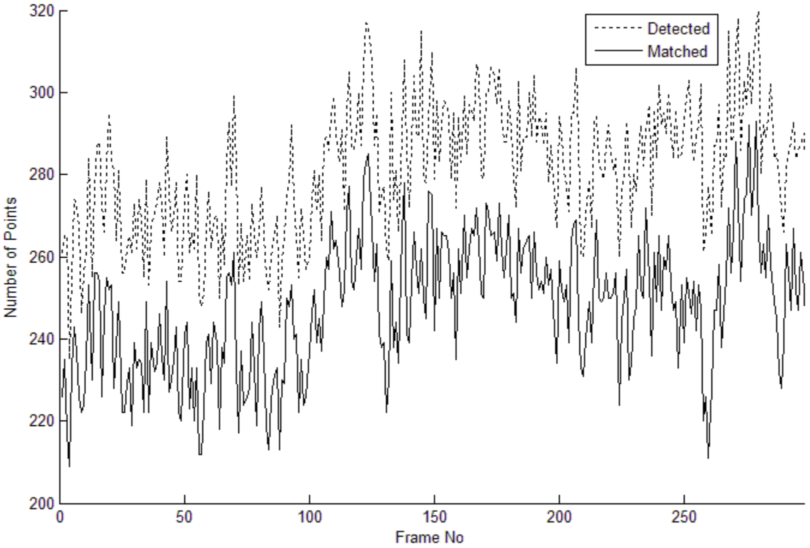


Figure B.85: The detection and matching results of "container.avi" for the third scheme.

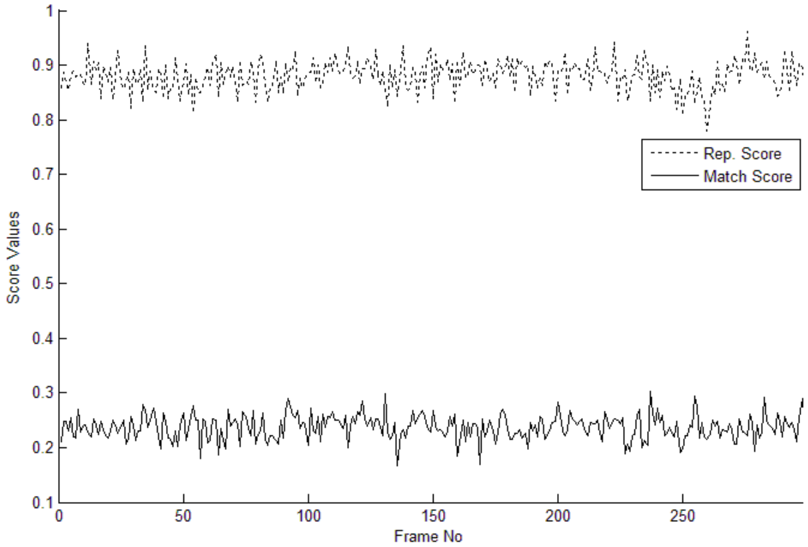


Figure B.86: The repetability and matching scores of "container.avi" for the third scheme.

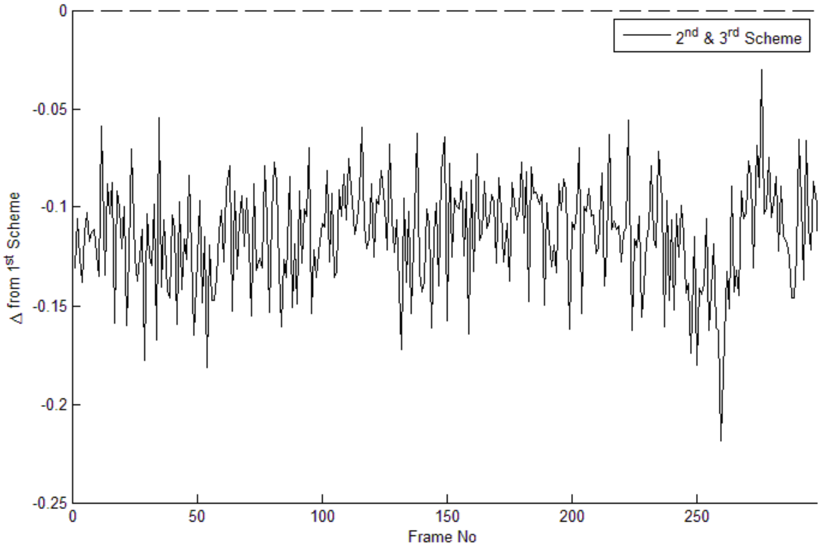


Figure B.87: The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for "container.avi".

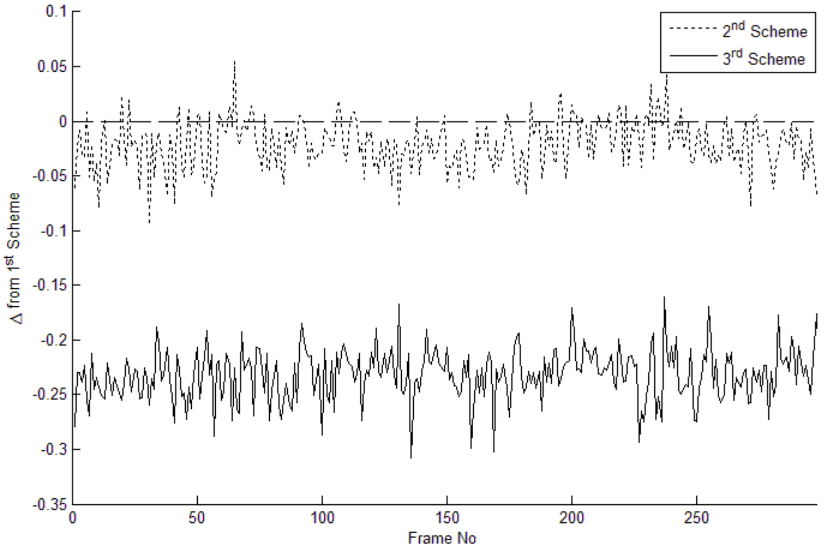


Figure B.88: The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for "container.avi".

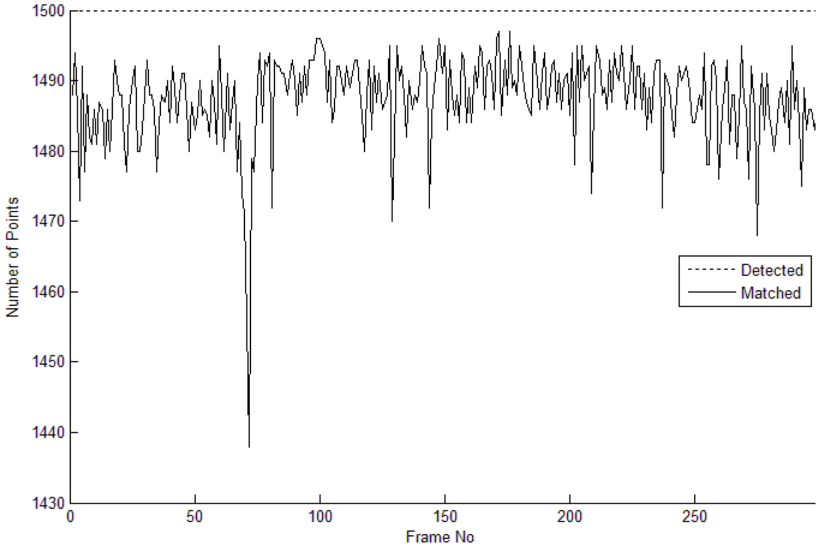


Figure B.89: The detection and matching results of “coastguard.avi” for the first scheme.

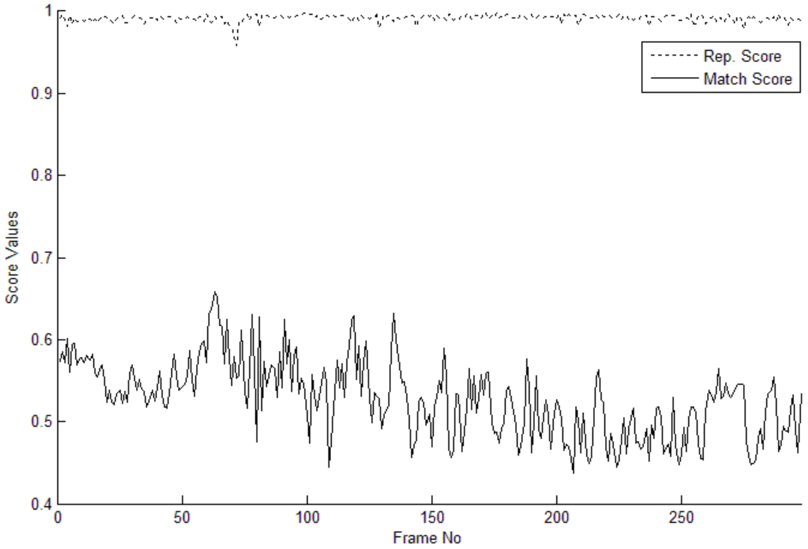


Figure B.90: The repetability and matching scores of “coastguard.avi” for the first scheme.

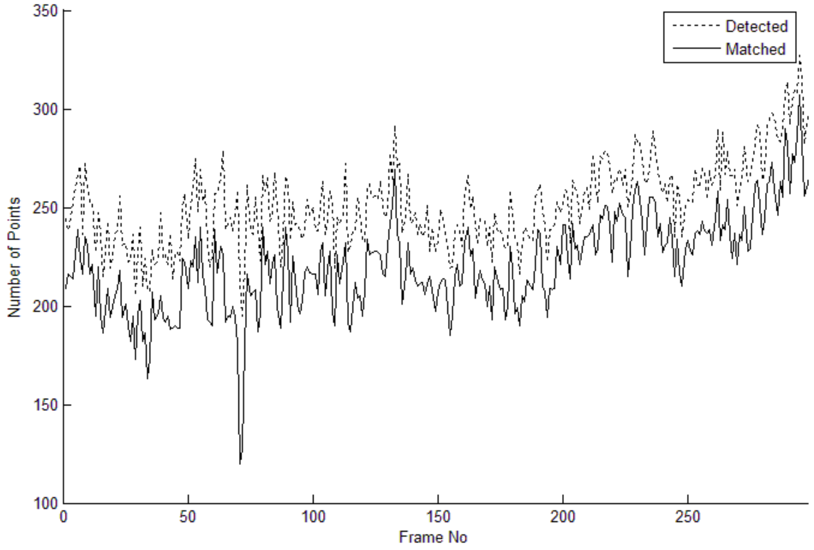


Figure B.91: The detection and matching results of "coastguard.avi" for the second scheme.

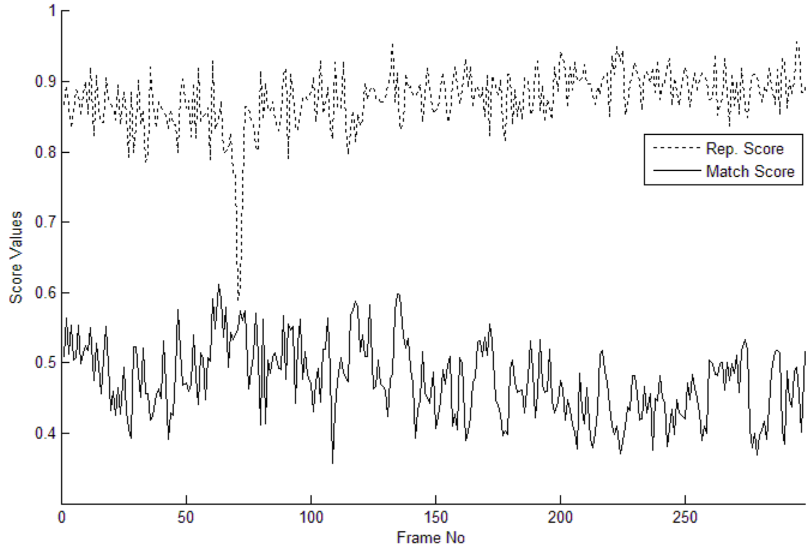


Figure B.92: The repetability and matching scores of "coastguard.avi" for the second scheme.

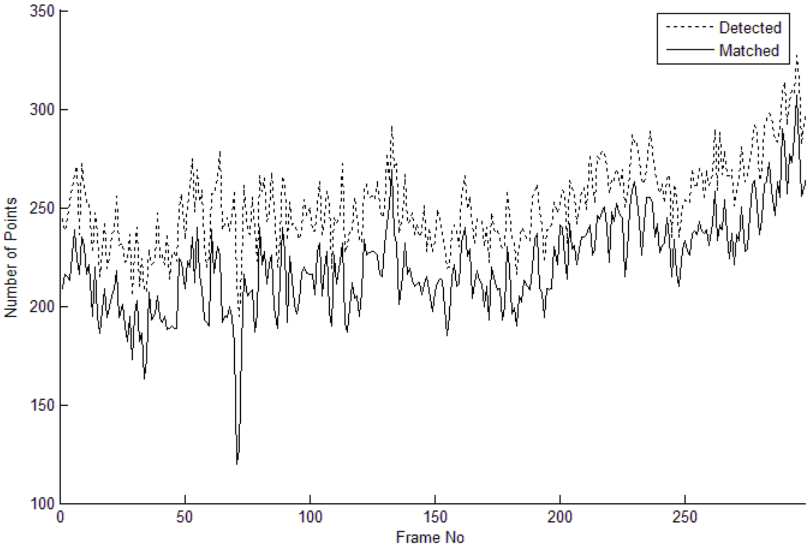


Figure B.93: The detection and matching results of “coastguard.avi” for the third scheme.

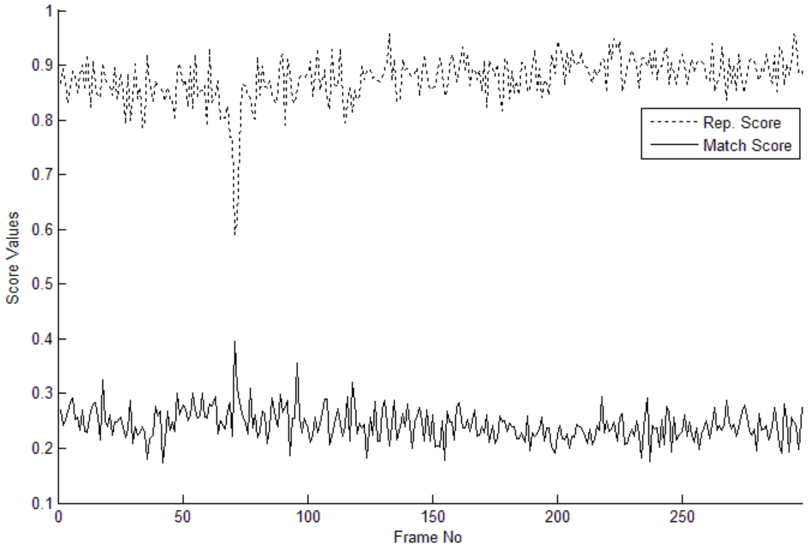


Figure B.94: The repetability and matching scores of “coastguard.avi” for the third scheme.

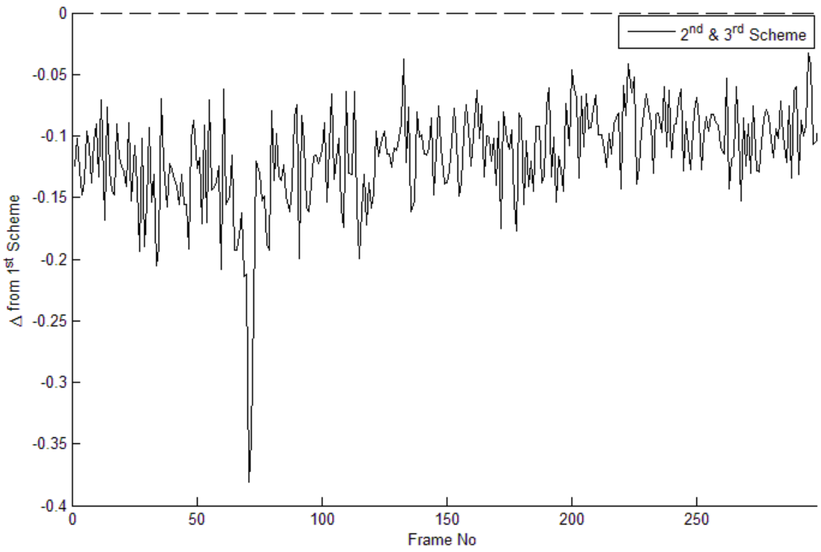


Figure B.95: The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for "coastguard.avi".

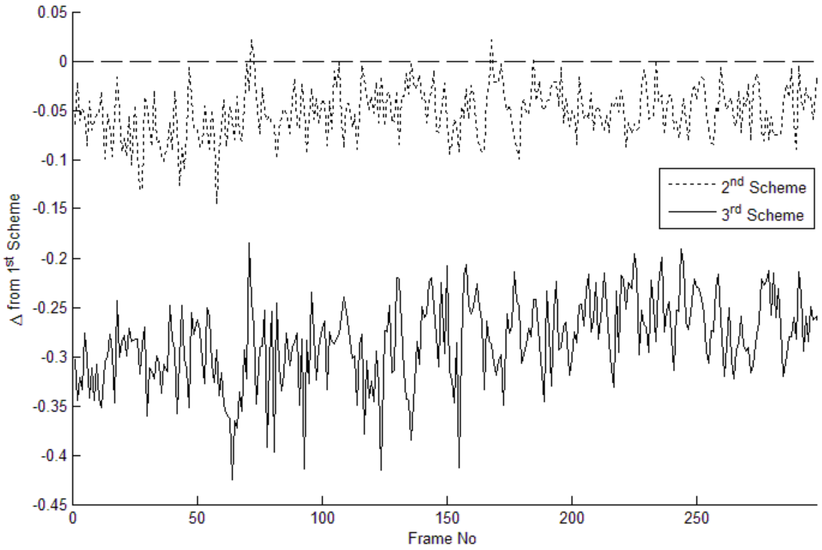


Figure B.96: The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for "coastguard.avi".

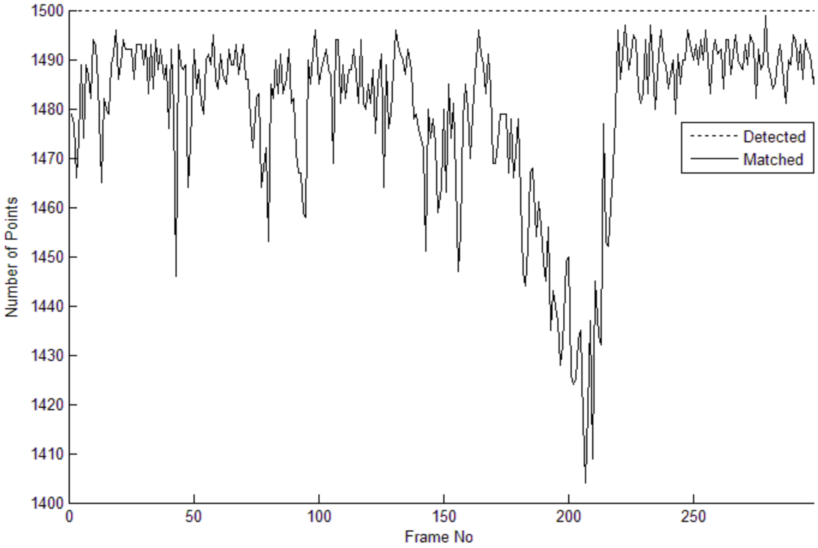


Figure B.97: The detection and matching results of “foreman.avi” for the first scheme.

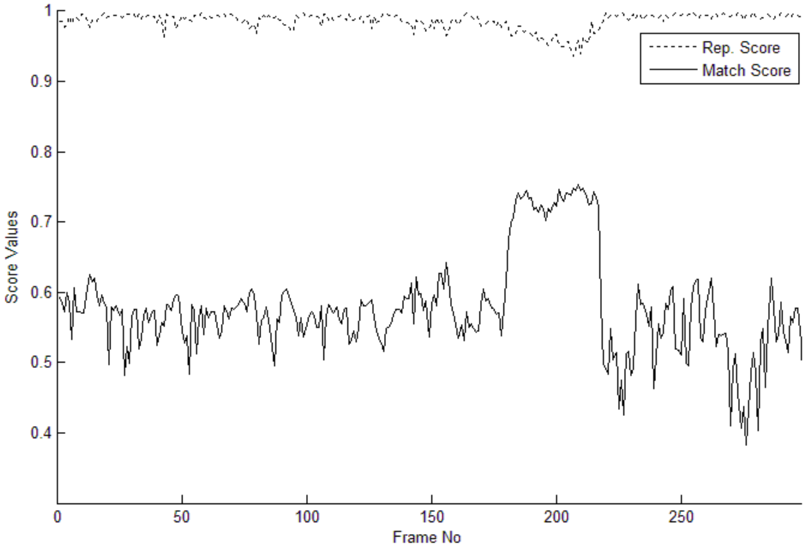


Figure B.98: The repetability and matching scores of “foreman.avi” for the first scheme.

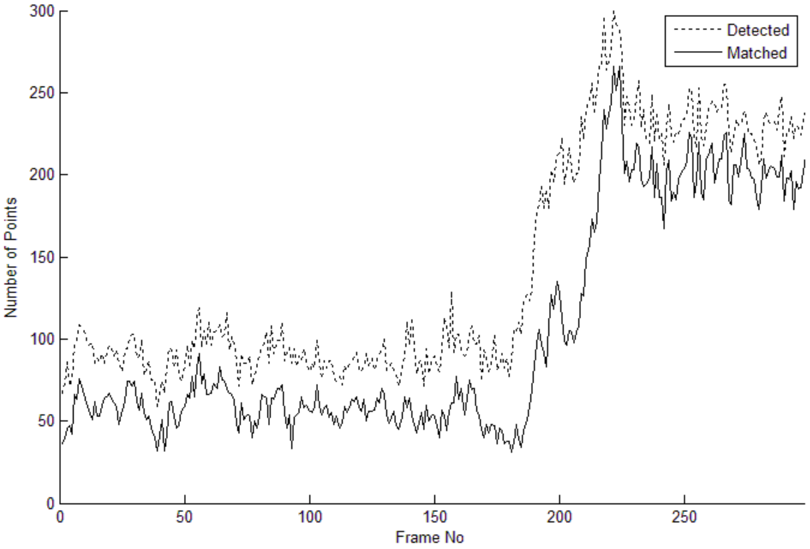


Figure B.99: The detection and matching results of “foreman.avi” for the second scheme.

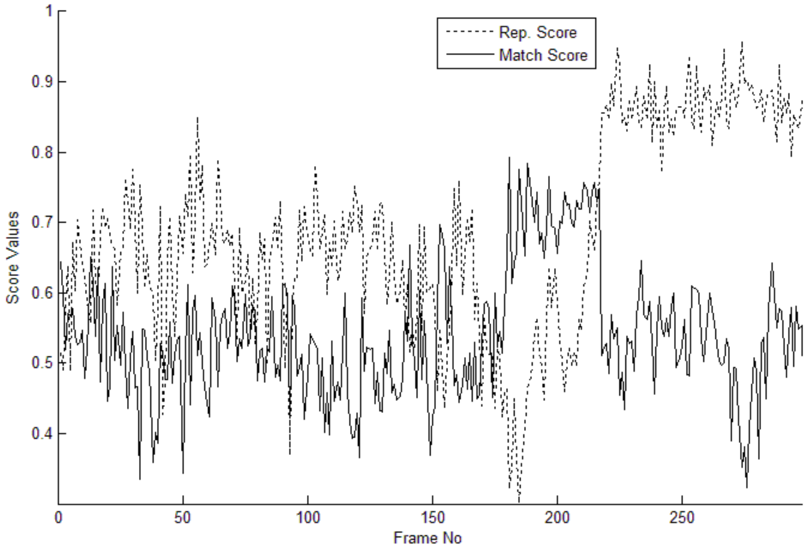


Figure B.100: The repetability and matching scores of “foreman.avi” for the second scheme.

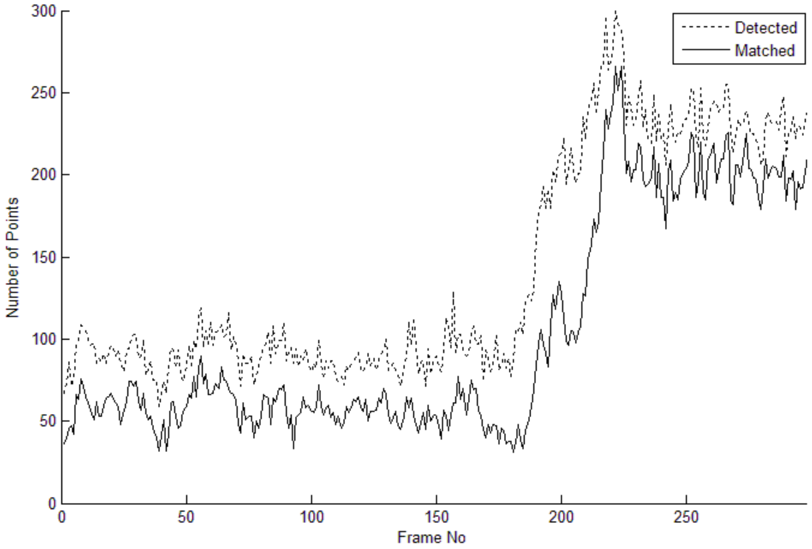


Figure B.101: The detection and matching results of "foreman.avi" for the third scheme.

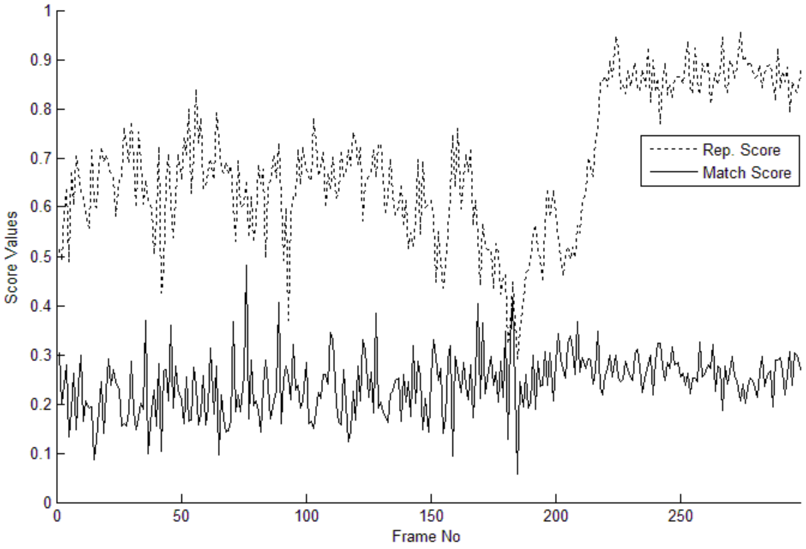


Figure B.102: The repetability and matching scores of "foreman.avi" for the third scheme.

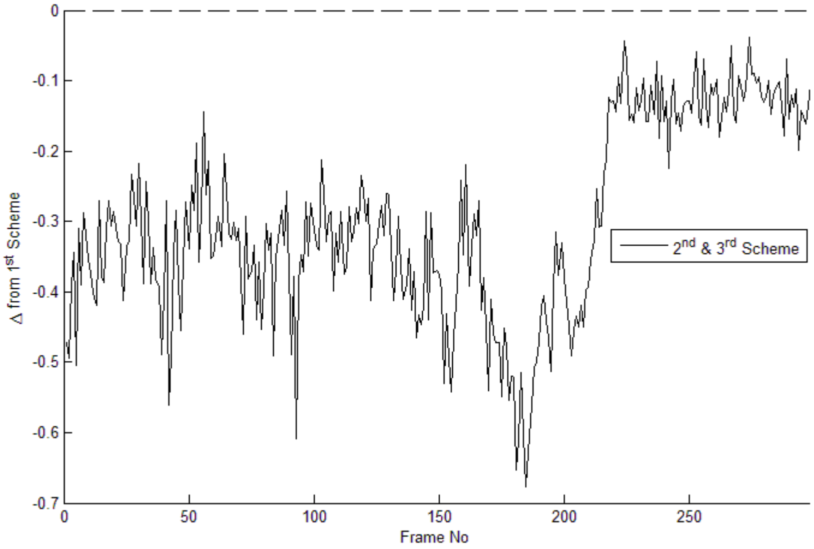


Figure B.103: The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for “foreman.avi”.

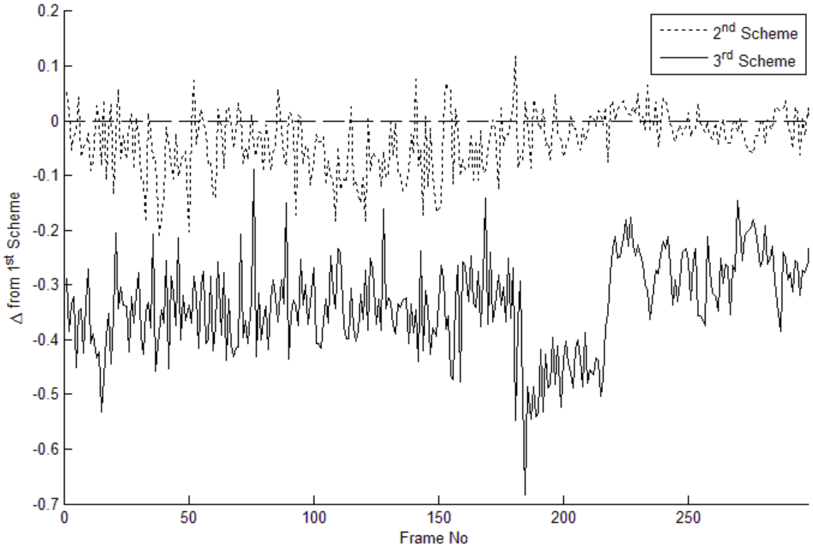


Figure B.104: The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for “foreman.avi”.

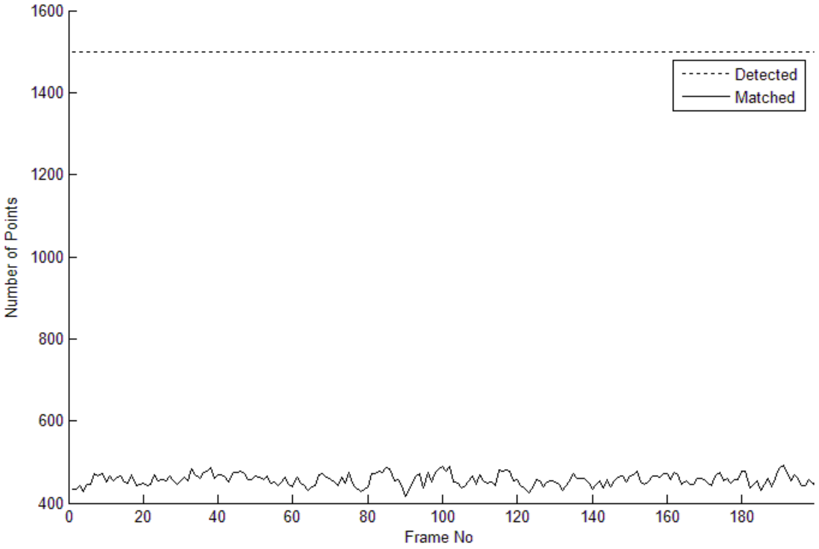


Figure B.105: The detection and matching results of "janine1_1.avi and janine1_2.avi" for the first scheme.

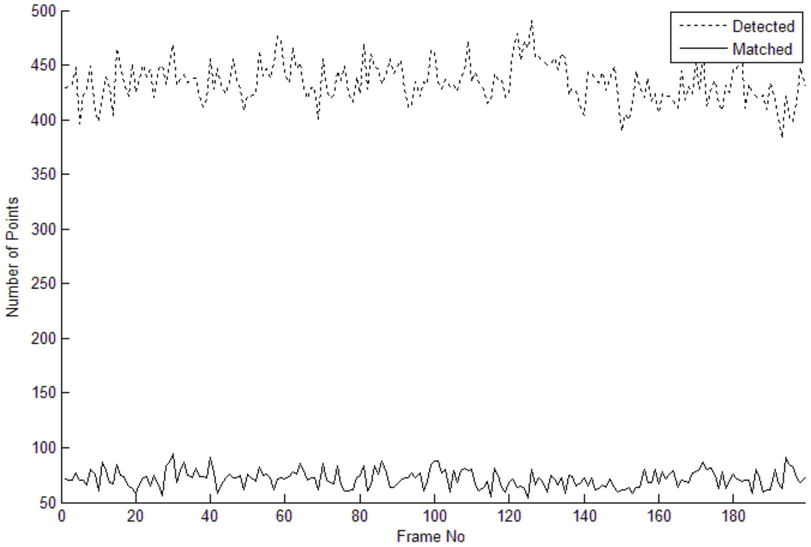


Figure B.106: The repetability and matching scores of "janine1_1.avi and janine1_2.avi" for the first scheme.

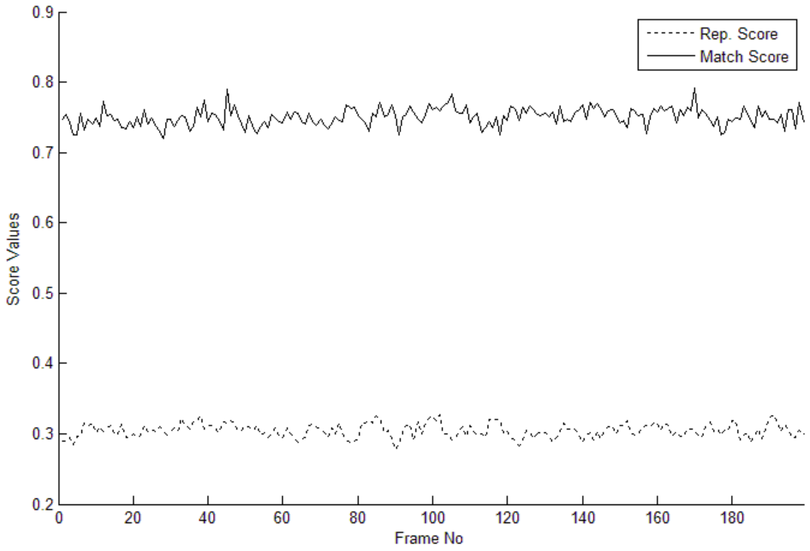


Figure B.107: The detection and matching results of “janine1_1.avi and janine1_2.avi” for the second scheme.

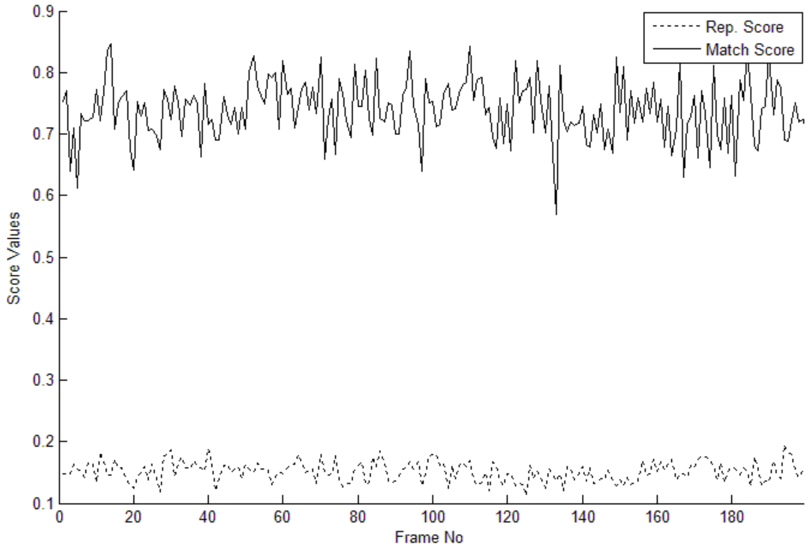


Figure B.108: The repetability and matching scores of “janine1_1.avi and janine1_2.avi” for the second scheme.

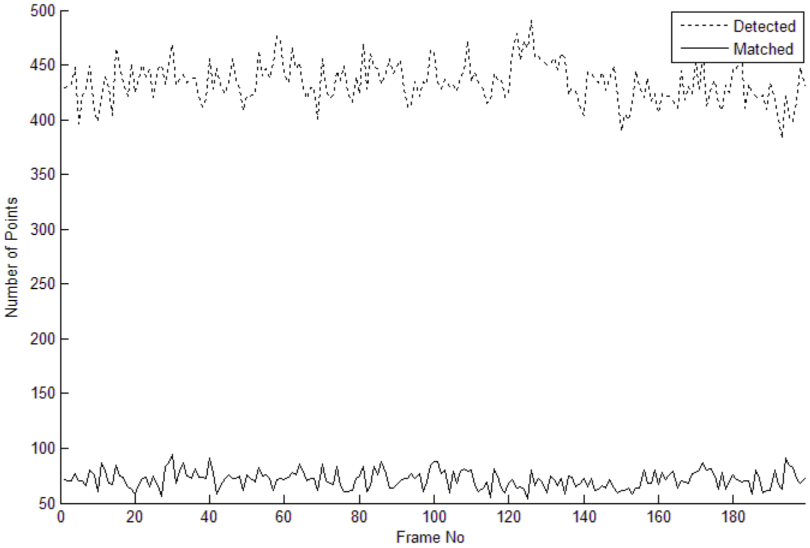


Figure B.109: The detection and matching results of "janine1_1.avi and janine1_2.avi" for the third scheme.

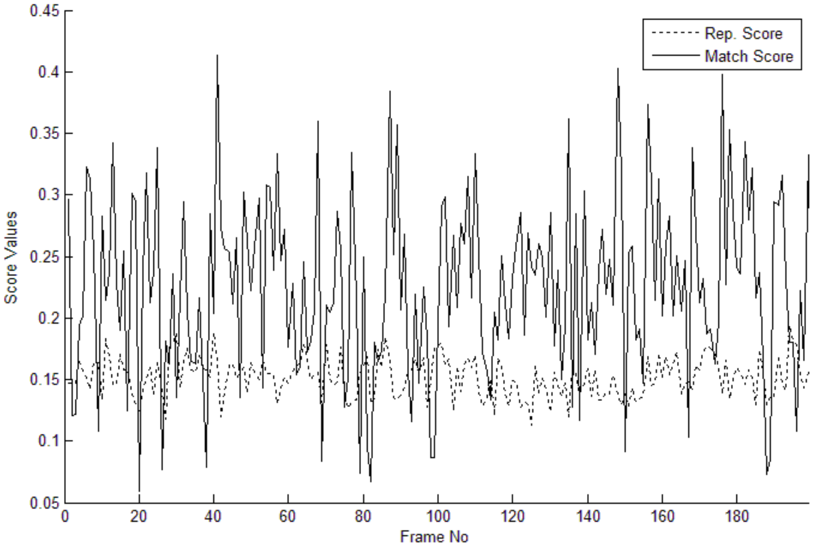


Figure B.110: The repetability and matching scores of "janine1_1.avi and janine1_2.avi" for the third scheme.

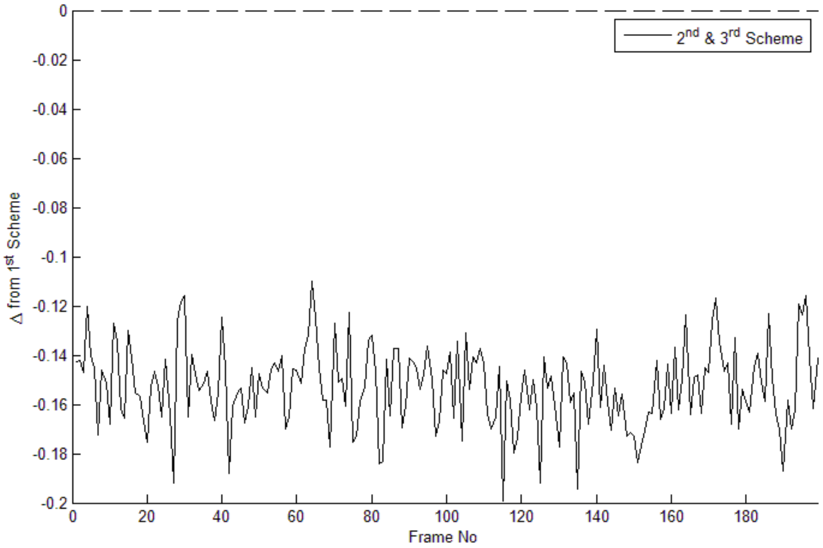


Figure B.111: The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for “janine1_1.avi and janine1_2.avi”.

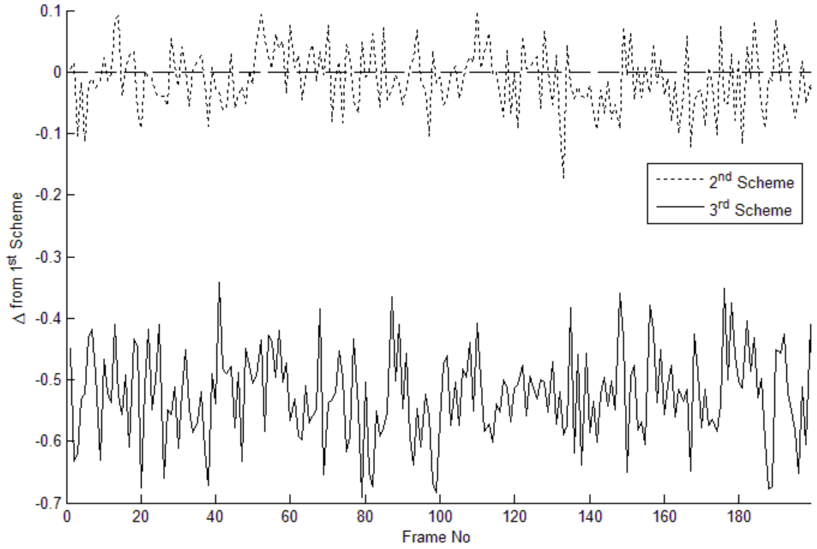


Figure B.112: The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for “janine1_1.avi and janine1_2.avi”.

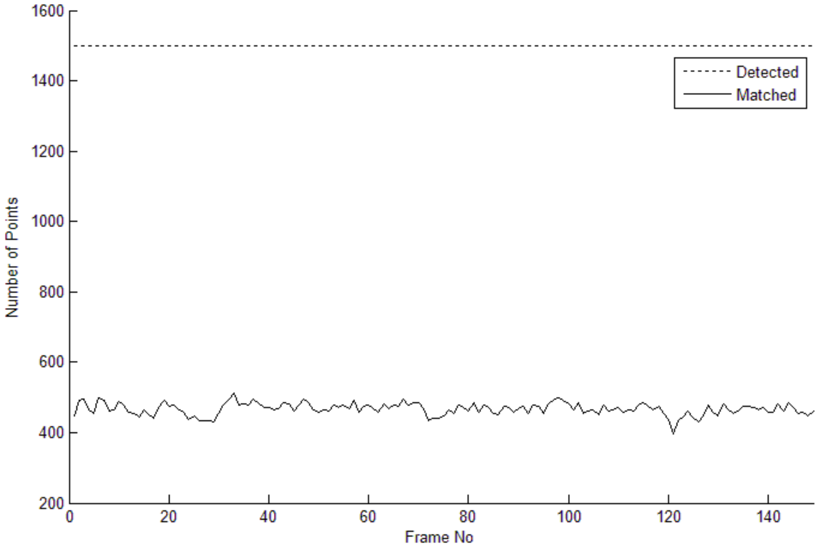


Figure B.113: The detection and matching results of “jungle_1.avi and jungle_2.avi” for the first scheme.

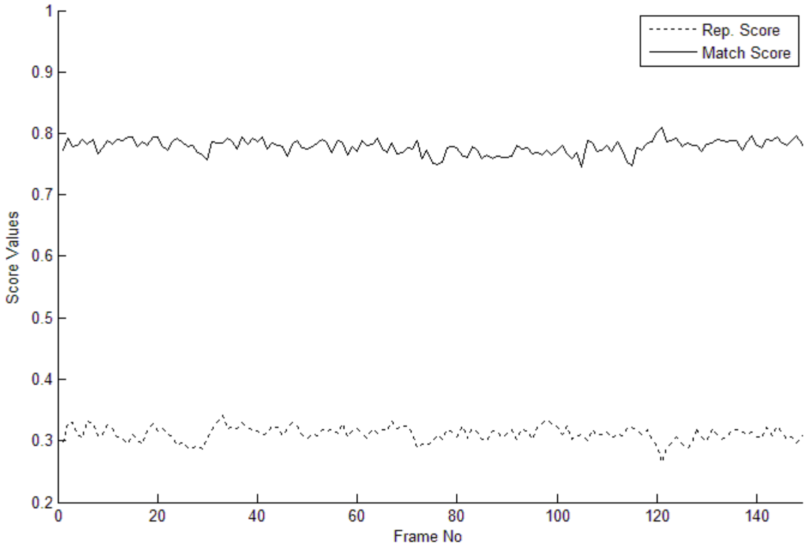


Figure B.114: The repetability and matching scores of “jungle_1.avi and jungle_2.avi” for the first scheme.

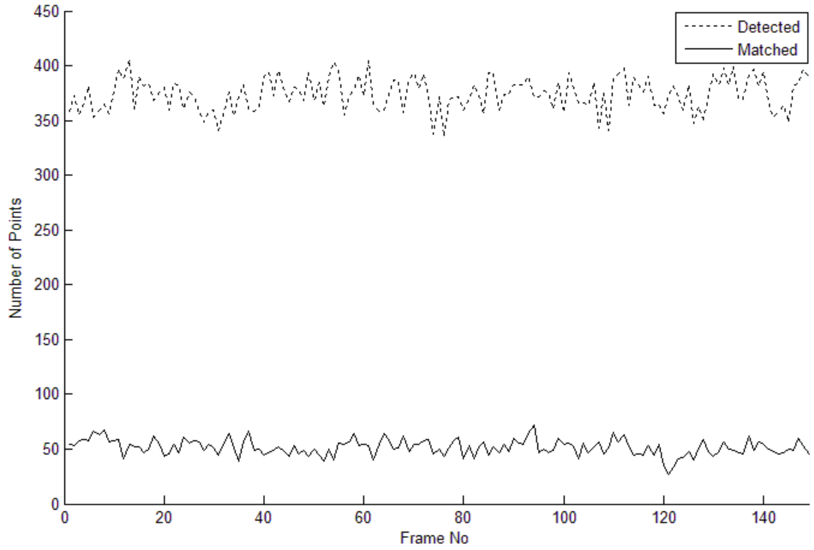


Figure B.115: The detection and matching results of “jungle_1.avi and jungle_2.avi” for the second scheme.

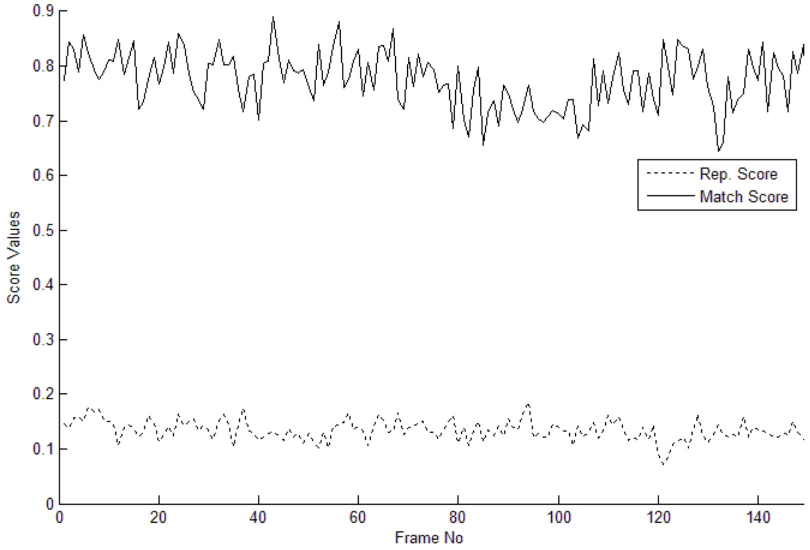


Figure B.116: The repetability and matching scores of “jungle_1.avi and jungle_2.avi” for the second scheme.

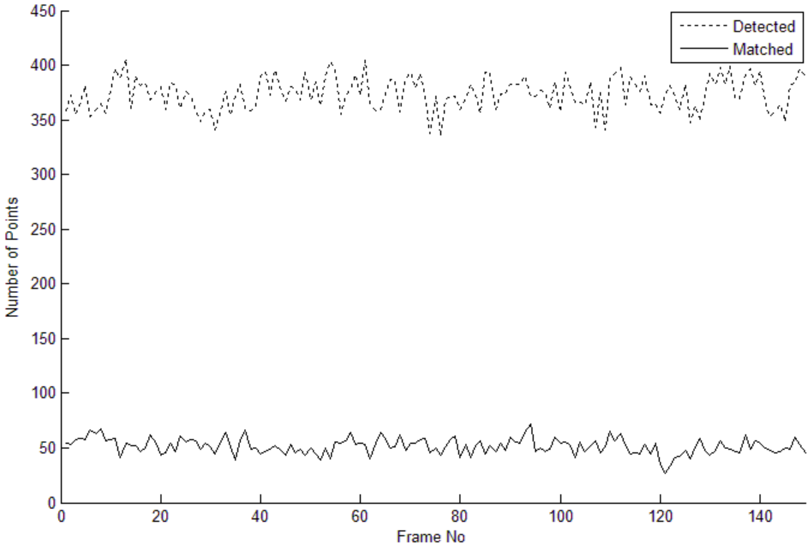


Figure B.117: The detection and matching results of "jungle_1.avi and jungle_2.avi" for the third scheme.

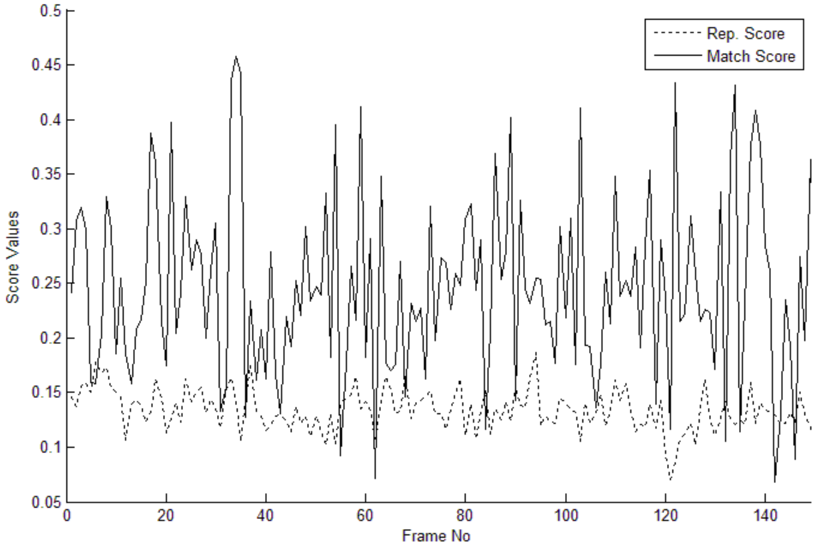


Figure B.118: The repetability and matching scores of "jungle_1.avi and jungle_2.avi" for the third scheme.

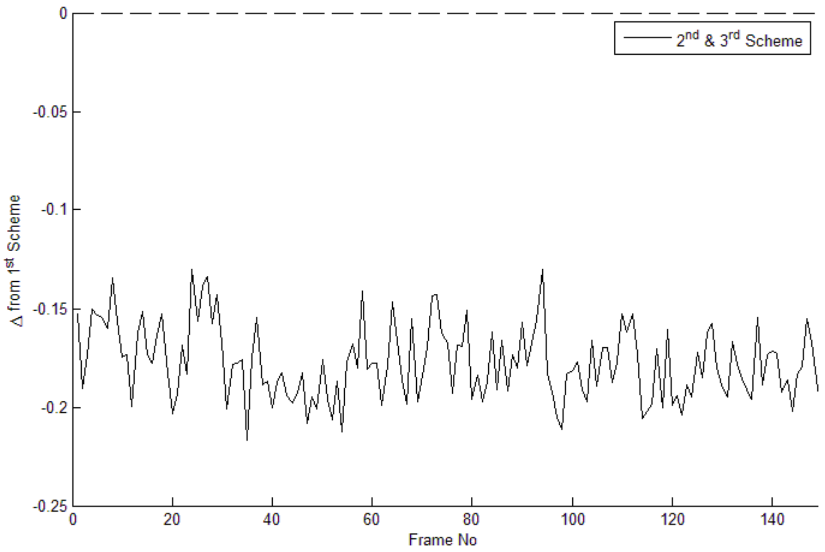


Figure B.119: The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for "jungle_1.avi and jungle_2.avi".

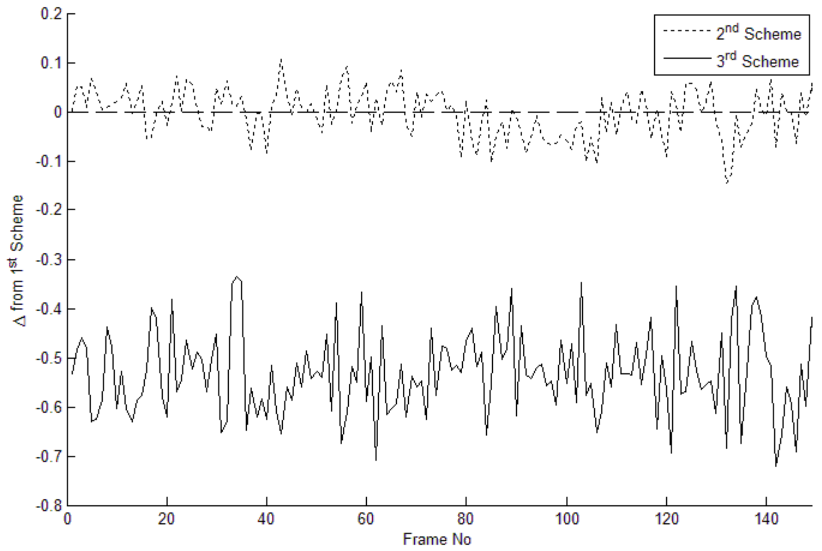


Figure B.120: The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for "jungle_1.avi and jungle_2.avi".

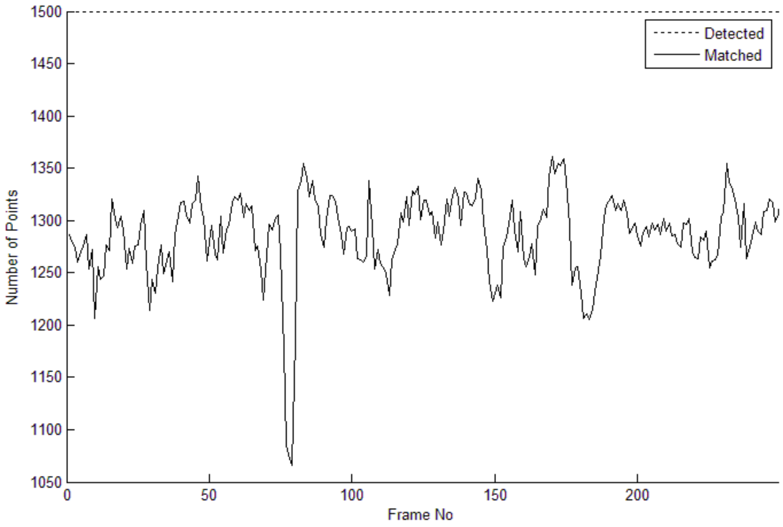


Figure B.121: The detection and matching results of "cam0_capture5_Deniz.avi and cam1_capture5_Deniz.avi" for the first scheme.

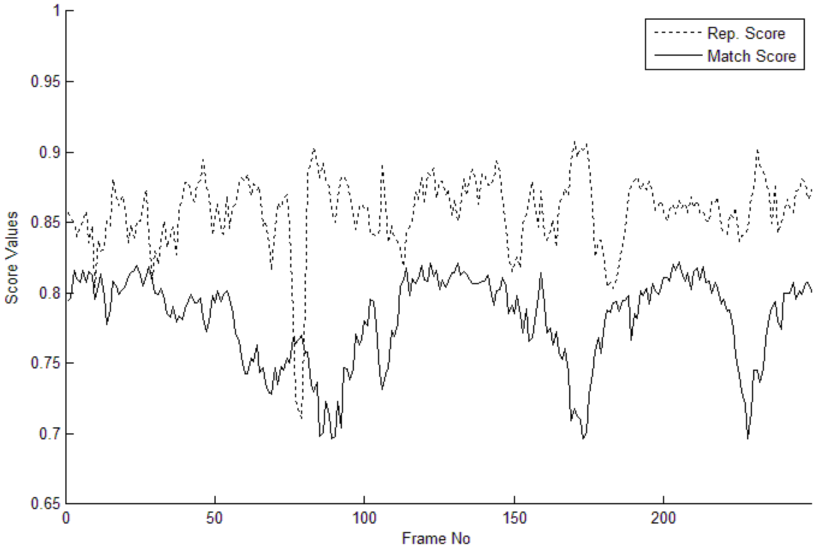


Figure B.122: The repetability and matching scores of "cam0_capture5_Deniz.avi and cam1_capture5_Deniz.avi" for the first scheme.

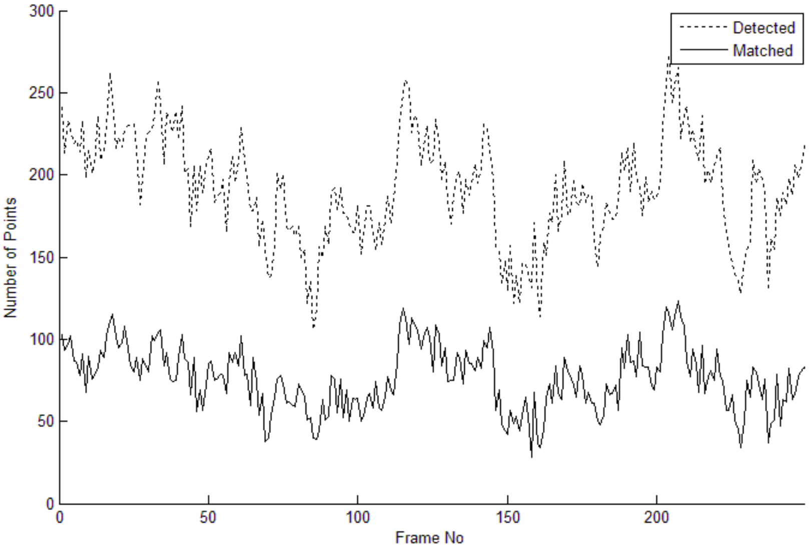


Figure B.123: The detection and matching results of "cam0_capture5_Deniz.avi and cam1_capture5_Deniz.avi" for the second scheme.

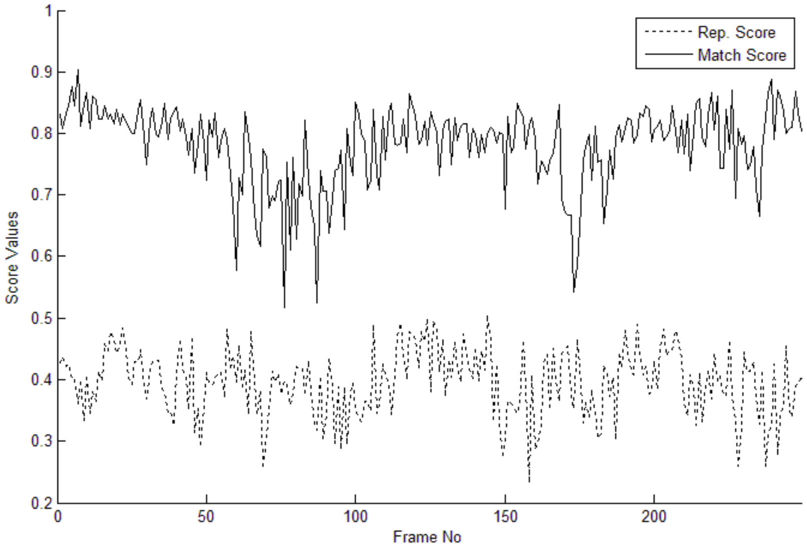


Figure B.124: The repetability and matching scores of "cam0_capture5_Deniz.avi and cam1_capture5_Deniz.avi" for the second scheme.

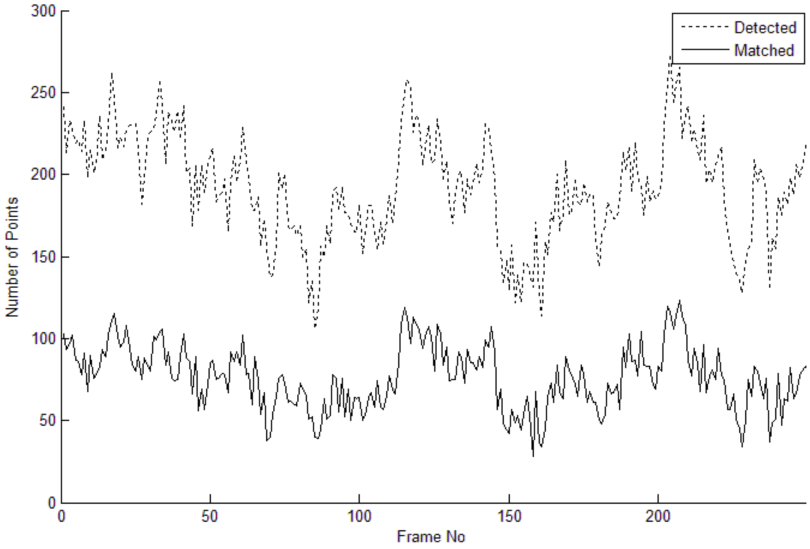


Figure B.125: The detection and matching results of "cam0_capture5_Deniz.avi and cam1_capture5_Deniz.avi" for the third scheme.

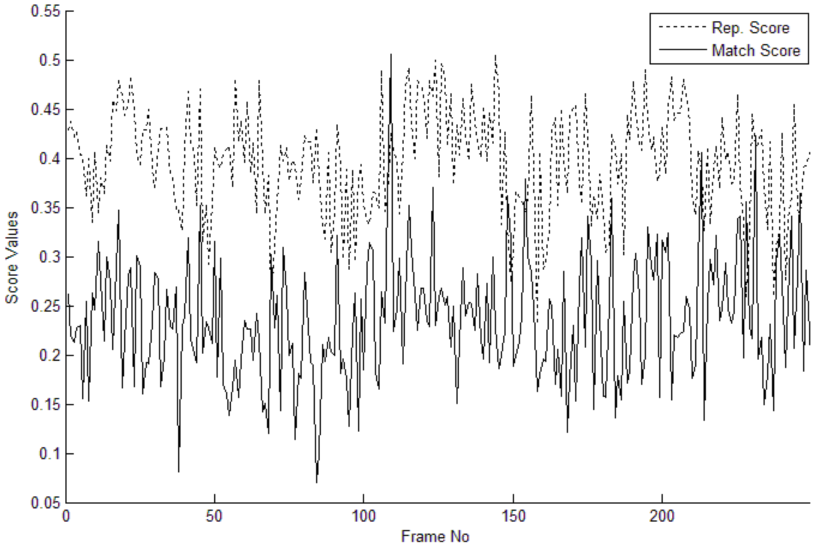


Figure B.126: The repetability and matching scores of "cam0_capture5_Deniz.avi and cam1_capture5_Deniz.avi" for the third scheme.

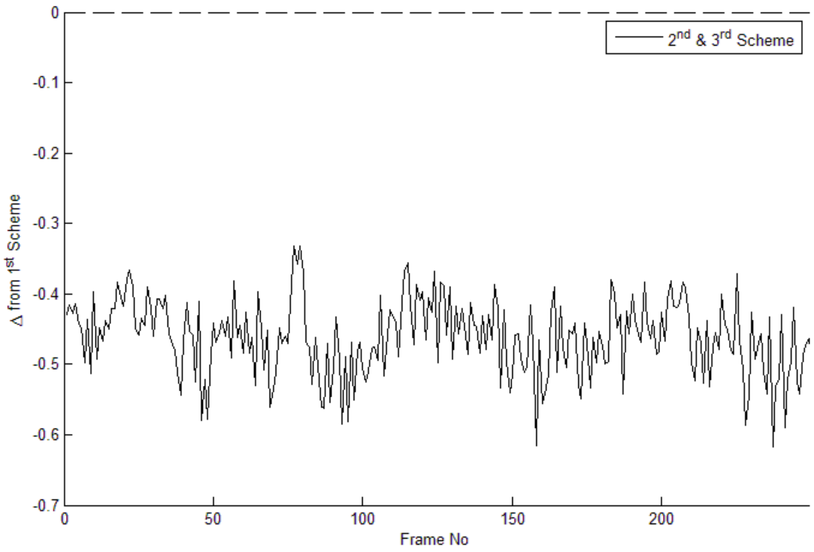


Figure B.127: The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for "cam0_capture5_Deniz.avi and cam1_capture5_Deniz.avi".

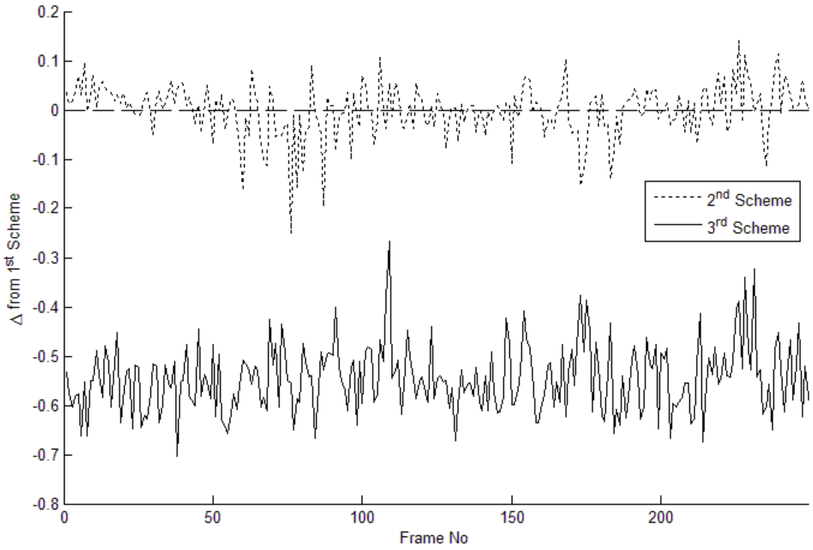


Figure B.128: The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for "cam0_capture5_Deniz.avi and cam1_capture5_Deniz.avi".

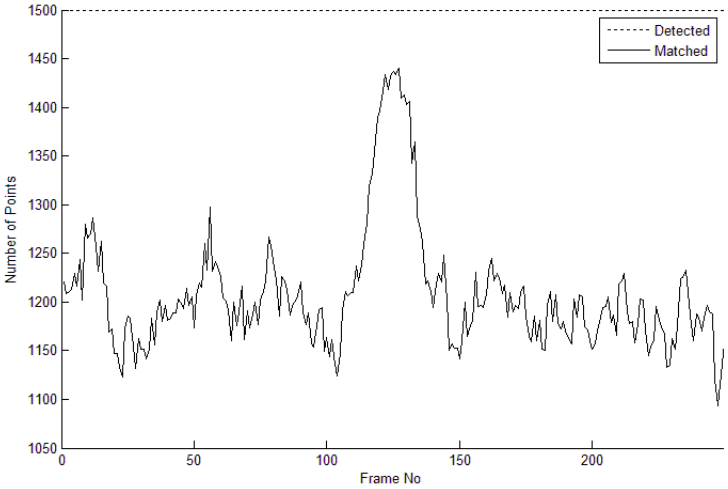


Figure B.129: The detection and matching results of "cam0_capture7_novice_jugglers.avi and cam1_capture7_novice_jugglers.avi" for the first scheme.

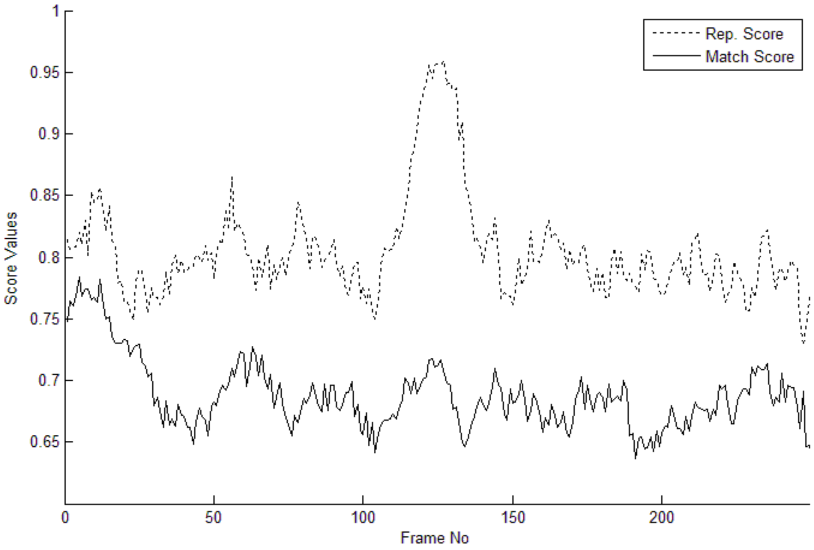


Figure B.130: The repetability and matching scores of "cam0_capture7_novice_jugglers.avi and cam1_capture7_novice_jugglers.avi" for the first scheme.

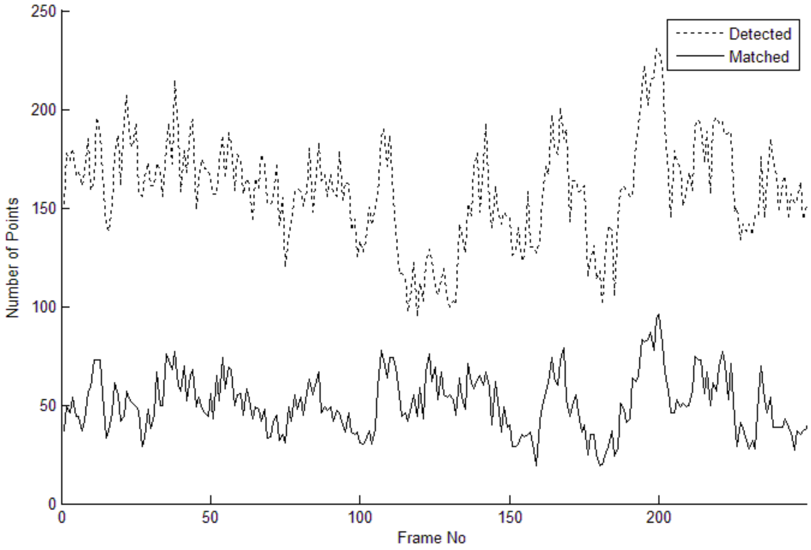


Figure B.131: The detection and matching results of "cam0_capture7_novice_jugglers.avi and cam1_capture7_novice_jugglers.avi" for the second scheme.

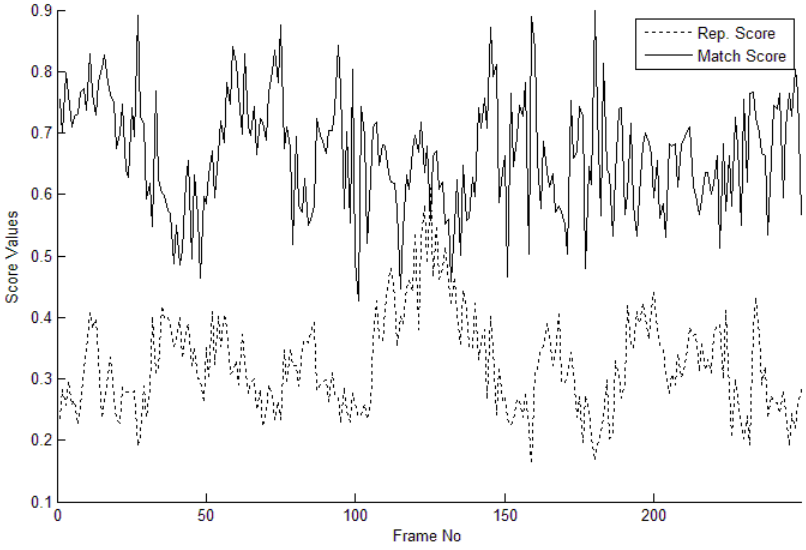


Figure B.132: The repetability and matching scores of "cam0_capture7_novice_jugglers.avi and cam1_capture7_novice_jugglers.avi" for the second scheme.

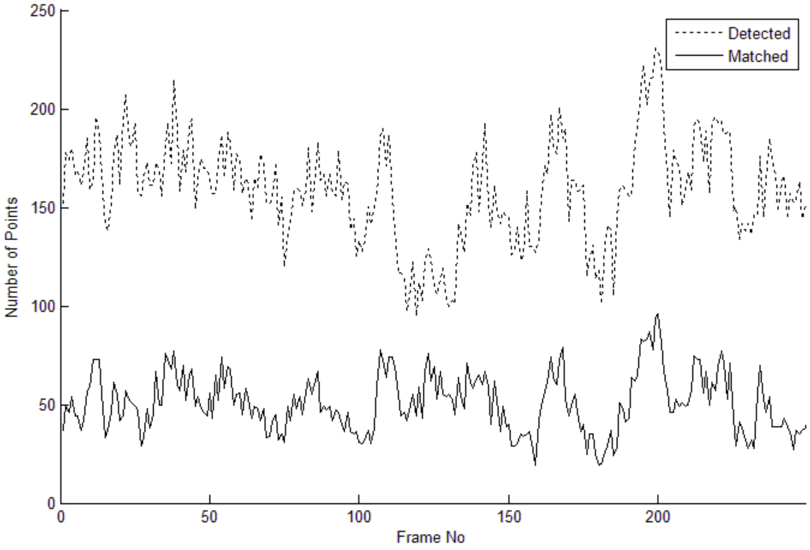


Figure B.133: The detection and matching results of “cam0_capture7_novice_jugglers.avi and cam1_capture7_novice_jugglers.avi” for the third scheme.

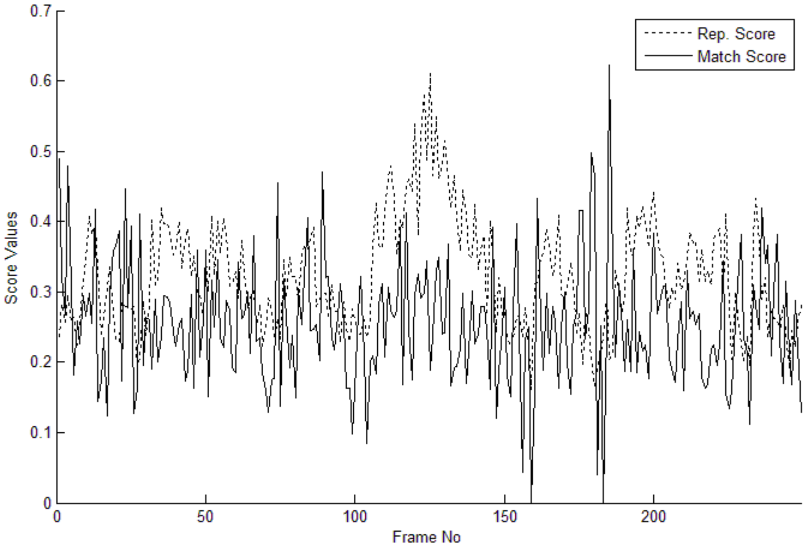


Figure B.134: The repetability and matching scores of “cam0_capture7_novice_jugglers.avi and cam1_capture7_novice_jugglers.avi” for the third scheme.

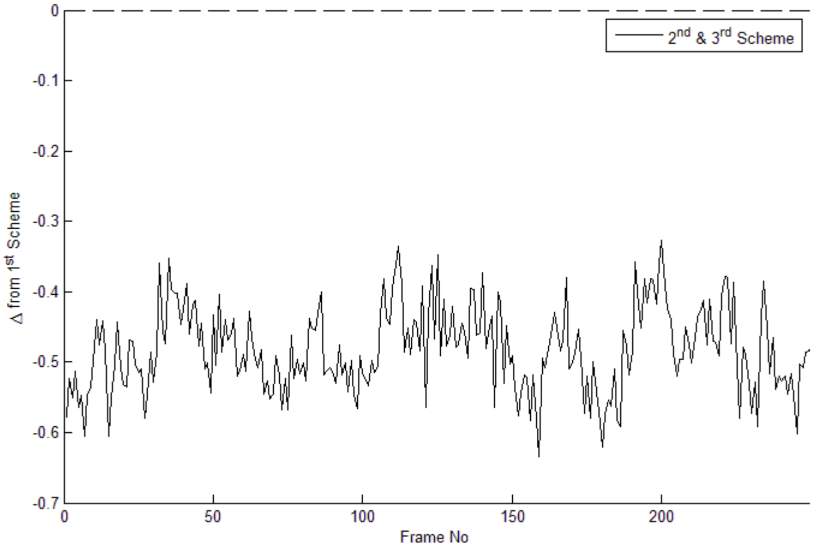


Figure B.135: The difference between repetability score of 1^{st} scheme and 2^{nd} or 3^{rd} schemes for "cam0_capture7_novice_jugglers.avi and cam1_capture7_novice_jugglers.avi".

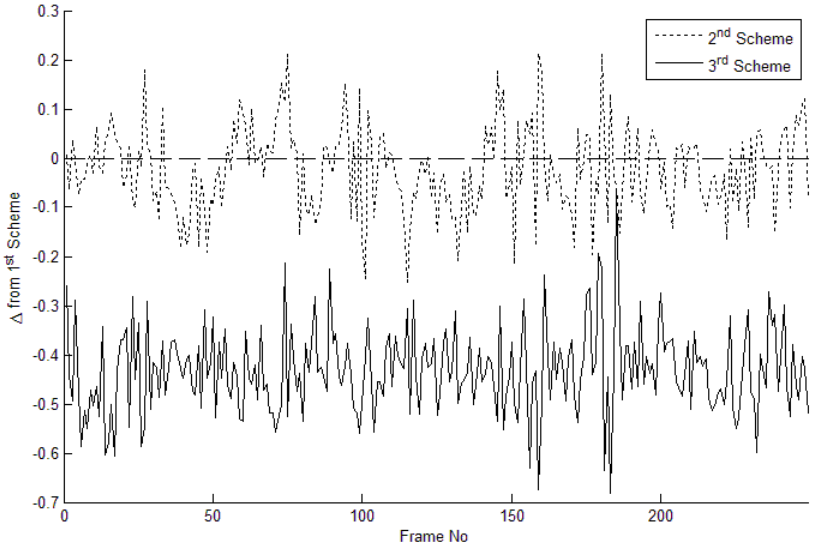


Figure B.136: The difference between match scores of 1^{st} scheme and 2^{nd} and 3^{rd} schemes for "cam0_capture7_novice_jugglers.avi and cam1_capture7_novice_jugglers.avi".