Analysis of Affine Motion-Compensated Prediction and its Application in Aerial Video Coding

Von der Fakultät für Elektrotechnik und Informatik

der Gottfried Wilhelm Leibniz Universität Hannover

zur Erlangung des akademischen Grades

Doktor-Ingenieur

(abgekürzt: Dr.-Ing.)

genehmigte

Dissertation

von

Dipl.-Ing. Holger Meuel

geboren am 27. Februar 1983 in Lübeck

2019

Hauptreferent:	Prof. Dr.-Ing. Jörn Ostermann
Korreferent:	Prof. Dr.-Ing. André Kaup
Vorsitzender:	Prof. Dr.-Ing. Hans-Georg Musmann

Tag der Promotion: 5. August 2019

Acknowledgement

This thesis was written during my time at the Institut für Informationsverarbeitung (TNT) of the Gottfried Wilhelm Leibniz Universität Hannover.

My special thank goes to Prof. Dr.-Ing. Jörn Ostermann who provided the possibility to work at the institute. He continuously supported me financially and scientifically. Particularly, I would like to thank for the intense and valuable discussions and supervision during the development of this thesis and of course for the evaluation of my thesis as first examiner. I also would like to thank Prof. Dr.-Ing. André Kaup for being the second examiner of this thesis, his helpful comments and the discussions at several opportunities. I also cordially thank Prof. Dr.-Ing. Hans-Georg Musmann for taking over the chair of the examination board and his continuous scientific support during my time at the TNT. For the inspiring discussions I like to thank Prof. Dr.-Ing. Bodo Rosenhahn who offered friendly support at all times.

Moreover, I especially thank all my colleagues. In particular, I owe my deep gratitude to Dr.-Ing. Marco Munderloh and Dr.-Ing. Ulrike Pestel-Schiller. Thanks for the continuous support in any matter from the very beginning until the defense of my thesis in word and deed! My deep appreciation also goes to my room mate Yiqun Liu who supported me relentlessly in any issue. I like to specially thank Stephan Ferenz, Hendrik Hachmann, Florian Kluger, Hanno Ackermann, Ph.D., Dr.-Ing. Aron Sommer, Dr.-Ing. Karsten Vogt, Stella Graßhof, Benjamin Spitschan, Dr.-Ing. Christian Becker, and Yasser Samayoa for plenty of discussions, general and mathematical support, and their encouragement. My acknowledgment also goes to my former room mate Julia Schmidt for her help and advice in business and personal matters. Thanks for all the fruitful cooperations resulting in publications, scientific and personal development and finally this thesis. For their sedulous support I like to thank Matthias Schuh, Doris Jaspers-Göring, Hilke Brodersen, Melanie Huch and the entire former office staff. For their administrative and constant support my acknowledgment goes to Dr.-Ing. Martin Pahl and Thomas Wehberg. Thanks for the great and inspiring time!

I thank my sister Sylvia Nissen for her good wishes and thoughts and my parents Ingrid and Dr. rer. nat. Bernd Meuel for enabling me to study electrical engineering as a basis for this thesis.

Finally, my deepest gratitude goes to my wife Dr. rer. nat. Katharina Neuhäuser for her untiring magnificent support without this thesis would not have been finalized. Thanks for encouraging me over the entire time, the substantial support and for always lighting up my life! I also owe my gratitude to Katharina's parents Emma and Prof. Dr. rer. nat. Hartmut Neuhäuser for their unconditional support in any matter and for integrating me into their family like a son.

This work is dedicated to my wife.

Contents

Abbreviations and Symbols

Abbreviations

AV1	AOMedia Video 1
AVC	Advanced Video Coding (H.264, MPEG-4 part 10)
AWGN	Additive white Gaussian noise
B-frame	Bidirectionally predicted frame
B	Byte
BD	Bjøntegaard delta
BD-PSNR	Bjøntegaard delta PSNR
BD-rate	Bjøntegaard delta rate
CABAC	Context-adaptive binary arithmetic coding
CCR	Comparison category rating (also known as double stimulus comparison or pair comparison method)
CIF	Common Intermediate Format, CIF video sequences have a resolution of 352 × 288 pel and are recorded at 30 fps
CMOS	Complementary metal-oxide-semiconductor
Codec	Coder-decoder
CRF	Corner response function
CTU	Coding tree unit
DCT	Discrete cosine transform
DoF	Degree of Freedom
DPCM	Differential pulse-code modulation
DVB	Digital Video Broadcasting
DVB-C/-C2	Digital Video Broadcasting – Cable (1^{st}/2^{nd} generation)
DVB-S/-S2	Digital Video Broadcasting – Satellite (1^{st}/2^{nd} generation)
DVB-T/-T2	Digital Video Broadcasting – Terrestrial (1^{st}/2^{nd} generation)
FP	False positive (detections)

Fps	Frames per second
GMC	Global motion compensation
GME	Global motion estimation
GOF	Group of frames (for in-loop radial distortion compensation)
GUI	Graphical user interface
HD	High definition (HD resolution equals 1920×1080 pel)
HEVC	High Efficiency Video Coding (H.265, MPEG-H part 2)
HM	HEVC Test Model
I-frame	Intra-coded frame
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
ITU	International Telecommunication Union, former: The International Telegraph and Telephone Consultative Committee (CCITT, from French: Comité Consultatif International Téléphonique et Télégraphique), former International Telegraph Union (ITU)
ITU-T	ITU Telecommunication Standardization Sector
JCT-VC	Joint Collaborative Team on Video Coding
JEM	Joint Exploration Model of JVET
JVET	Joint Video Exploration Team (on Future Video Coding) of ITU-T VCEG and ISO/IEC MPEG founded in October 2015, later transitioned into Joint Video Experts Team (also abbreviated by JVET) in April 2018
kbit	Kilobit
KLT	Kanade-Lucas-Tomasi feature tracker
LD	Low-delay
LDP	Low-delay p
MB	Megabyte
Mbit	Megabit
MC	Motion compensation
MCP	Motion-compensated prediction
ME	Motion estimation
MO	Moving object
MPEG	Motion Picture Experts Group
MPEG-4 ASP	MPEG-4 Advanced Simple Profile
MSE	Mean squared error

MV	Motion vector
MVP	Motion vector prediction
NA	New area
P-frame	Predicted frame
PCM	Pulse-code modulation
Pdf	Probability density function
Pel	Picture element (also known as pixel)
PSD	Power spectral density
PSNR	Peak signal-to-noise ratio
QCIF	Quarter CIF, QCIF video sequences have a resolution of 176 × 144 pel and are recorded at 30 fps
RA	Random-access profile
RANSAC	Random sample consensus
RD	Rate-distortion
RDC	Radial distortion compensation
RDO	Rate-distortion optimization
ROI	Region of interest
ROI-MO	Region of interest – moving object
ROI-NA	Region of interest – new area
ROI-PSNR	PSNR of ROI areas
SAD	Sum of absolute differences
SEI	Supplemental enhancement information
SfM	Structure from motion
SIFT	Scale-Invariant Feature Transform
SNR	Signal-to-noise ratio
s	Second
SSD	Sum of squared differences
TCS	Temporally consistent superpixel
TP	True positive (detections)
TV	Television
UAV	Unmanned aerial vehicle
VCEG	Video Coding Experts Group
VOD	Video on demand
VVC	Versatile Video Coding
x265	Open-source HEVC video encoder software
Y	Luminance component

Symbols

a, b	Parameter of the simplified affine model
$\mathtt{A}$	Affine matrix of size 2×2
A	Auxiliary variable
$\mathtt{A}_f$	Fully affine matrix of size 2×3
a_{ij}	Entries of the fully affine matrix, $i = \{1,2\}$, $j = \{1,2,3\}$
a_i	Entries of the simplified affine matrix, $i = \{a,b,c,f\}$
α	Drop rate of an exponential isotropic (autocorrelation) function
α_x, α_y	Drop rates of exponential (autocorrelation) functions in x- and y-direction
B_{CRF}	Maximum number of feature points per frame
$b_k(\boldsymbol{n})$	Binarized image intensity differences of the frame k
c	Parameter of the simplified affine model (translation in x-direction)
$\boldsymbol{C} = (C_x, C_y, C_z)^\top$	Position of the camera in world coordinates
$c_{\mathrm{size,max}}$, $c_{\mathrm{shape,max}}$	Maximum allowed size and shape change in in-loop radial distortion compensation
c_x, c_y	Thresholds which limit rotations around the x- and y-axis, respectively, in in-loop radial distortion compensation
$\boldsymbol{d}$	Motion vector
D	Maximum allowed average distortion (rate-distortion theory)
$d(u;v)$	General distortion measure between symbols u and v (rate-distortion theory)
d_f	Minimum feature distance
$d_k(\boldsymbol{n})$	Image intensity differences of the frame k
$\boldsymbol{d} = (d_x, d_y)^\top$	Displacement vector
$\boldsymbol{d}_i = (d_{i,x}, d_{i,y})^\top$	Displacement of the i-th feature
$\hat{\boldsymbol{d}}$	Estimate of $\boldsymbol{d}$
$\bar{d}$	Average distortion (rate-distortion theory)
${}^{\mathrm{simp}}D$	Distortion using a simplified affine model (rate-distortion theory)
$\Delta x'$, $\Delta y'$	Displacement estimation error in horizontal (x-) and vertical (y-) direction of the fully affine model

$\Delta x'_{\text{mod}}, \Delta y'_{\text{mod}}$	Displacement estimation error caused by an inappropriate motion model in horizontal (x-) and vertical (y-) direction
$\Delta x'_{\text{s}}, \Delta y'_{\text{s}}$	Displacement estimation error in horizontal (x-) and vertical (y-) direction of the simplified affine model
δ	Dirac delta function
$\boldsymbol{d}'$	Motion vector (for transmission) with limited accuracy
e	Prediction error signal
$E(\cdot)$	Expectation value of $(\cdot)$
$e_k(\boldsymbol{n})$	Binarized image intensity differences of the frame k after erosion
$e_{ij,\text{mod}}$	Error terms caused by the motion model, $i = \{1,2\}$, $j = \{1,2,3\}$
e'	Quantized prediction error signal (residuum)
e_{q}	Quantization error
e_i	Error terms (perturbations of a, b, c, f) of the simplified affine model, $i = \{a,b,c,f\}$
e_{ij}	Error terms (perturbations of a_{ij}) of the fully affine model with $i = \{1,2\}$, $j = \{1,2,3\}$
ϵ	Arbitrarily small error (rate-distortion theory)
f	Frequency (rate-distortion theory)
f	Parameter of the simplified affine model (translation in y-direction)
$\boldsymbol{f}_{i,k}$	Position of the i-th feature in the frame k
f_c	Focal length
$\boldsymbol{g}_{k-1}$	Holds the temporal derivatives of I
$h_{11}, \ldots, h_{33}$	Elements of $\mathtt{H}$
$\mathtt{H}$	Homography matrix of size 3×3
H_{G}	Entropy of a memoryless, time-discrete, amplitude-continuous Gaussian source
i, j	Counter variables
$I(\boldsymbol{n})$	Image intensity at the position $\boldsymbol{n}$
$I_k(\boldsymbol{n})$	Image intensities of the frame k
i_{RDC}	Number of iterations for in-loop radial distortion compensation
I_x, I_y	Partial derivatives of I

k	Frame index
k_{ang}	Constant value in the small-angle approximation
κ_1	Radial distortion parameter
$\kappa_{1,l}$	Radial distortion parameter of group of frames with index l
k_{H}	Harris weighting factor
K	Number of code symbols (rate-distortion theory)
$\mathtt{K}$	Camera calibration matrix of size 3×3
l	Counter variable (for groups of frames in in-loop radial distortion compensation)
L	Number of source symbols emitted by source U (rate-distortion theory)
λ_1, λ_2	Eigenvalues of Harris corner matrix $\mathtt{M}$
Λ	Two-dimensional (2D) spatial frequency vector $\Lambda := (\omega_x, \omega_y)$
m, n	Counter variables
$\mathtt{M}$	Harris corner matrix
M_{CRF}	Minimum distance between feature points
n_{RDC}	Number of frames in a group of frames
$\boldsymbol{n} = (x, y)^\top$	Point on the image plane in image coordinates
$\frac{\boldsymbol{n}_\text{s}}{d_\text{s}}$	Surface normal vector, with d_s being the distance between the camera center and the surface
N_x, N_y	Number of sensor elements in x- and y-direction
$N(f)$	Distortion of a single source in rate-distortion theory
$\mathcal{N}(m_\text{G}; v_\text{G})$	Follows a Gaussian distribution with mean m_G and variance v_G
$N_\text{P}(n_\text{G})$	Power of Gaussian noise n_G
n_G	Gaussian noise
n_{mos}	Frame distance (long-term mosaicking)
ω_x, ω_y	Spatial frequencies in x- and y-direction
$\boldsymbol{p} = (x_c, y_c)^\top$	Point on the image plane in sensor coordinates
$\tilde{\boldsymbol{p}} = (x_d, y_d)^\top$	Point on the image plane with lens distortion
$\boldsymbol{p}_k$	Point on the image plane of camera $\boldsymbol{C}_k$
$\hat{\boldsymbol{p}}_k$	Estimate of $\boldsymbol{p}_k$ through affine motion compensation
$\boldsymbol{P} = (X, Y, Z)^\top$	Point in world coordinates
$\tilde{\boldsymbol{P}} = (X_\text{c}, Y_\text{c}, Z_\text{c})^\top$	Point in camera coordinates

$p_{\Delta X',\Delta Y'}(\Delta x', \Delta y')$	2D probability density function of the displacement estimation error (of the fully affine model)
${}^{\text{simp}}p_{\Delta X'_\text{s},\Delta Y'_\text{s}}(\Delta x'_\text{s}, \Delta y'_\text{s})$	2D probability density function of the displacement estimation error using a simplified affine model
$p(\cdot)$	Probability density function of $(\cdot)$
$p_\bullet(\cdot)$	General form of a probability density function of the random process $\bullet$ with the observations $(\cdot)$
$P(\Lambda)$	Fourier transform of the displacement estimation error
$\boldsymbol{q}(q_1,q_2)^\top, q$	Projective components of the homography
r, r_d	Radii of $\boldsymbol{p}$ and $\tilde{\boldsymbol{p}}$ to the center of distortion
$r_{11} \ldots r_{33}$	Elements of $\mathtt{R}$
$r_k(\boldsymbol{n})$	Pel-wise motion detection results of the frame k
$R(D)$	Bit rate R as a function of the distortion D (rate-distortion theory)
${}^{\text{simp}}R\left({}^{\text{simp}}D\right)$	Bit rate R as a function of the distortion D using a simplified affine model (rate-distortion theory)
$R_\text{G}(D)$	Bit rate R_G of a Gaussian source as a function of the distortion D (rate-distortion theory)
R_{ss}	Autocorrelation function of the video signal s
$R_{ss,\text{iso}}$	Isotropic autocorrelation function of the video signal s
$\rho_{ss,x}$, $\rho_{ss,y}$	Autocorrelation coefficients of the video signal s in x- and y-direction
$\mathtt{R} = \mathtt{R}_\theta\mathtt{R}_\gamma\mathtt{R}_\beta$	Camera orientation matrix of size 3×3
s	Video signal
s_s	Scaling parameter of the simplified affine model
s_w, s_h	Width and height of the camera sensor
s_x, s_y	Width and height of one pel on the image sensor
$\hat{s}$	Predicted signal
s'	Reconstructed video signal
s^*	Preprocessed signal
$\sigma^2_{\Delta x'}$, $\sigma^2_{\Delta y'}$	Variances of $\Delta x'$ and $\Delta y'$ of the fully affine model
$\sigma^2_{\Delta x'_\text{s}}$, $\sigma^2_{\Delta y'_\text{s}}$	Variances of $\Delta x'_\text{s}$ and $\Delta y'_\text{s}$ of the simplified affine model
$\sigma^2_{e_{ij}}$	Variances of the error terms e_{ij}, $i = \{1,2\}$, $j = \{1,2,3\}$
$\sigma^2_{e_{ij,\text{mod}}}$	Variance of the error terms $e_{ij,\text{mod}}$, $i = \{1,2\}$, $j = \{1,2,3\}$, representing the motion model error

σ_u^2	Variance of the source symbols u
σ_x, σ_y	Standard deviations of x and y
S_{CRF}	Threshold of corner response function
S_{ee}	Power spectral density of the prediction error e
${}^{\mathrm{simp}}S_{ee}$	Power spectral density of the prediction error e using a simplified affine model
$S(f)$	Power spectral density
S_{ss}	Power spectral density of the video signal s
t	Time
$\boldsymbol{t}$	Translation vector component of a homography
Θ	Parameter that generates the function $R(D)$ by taking on all positive real values (rate-distortion theory)
θ	Rotation parameter of the simplified affine model
θ_x, θ_y, θ_z	Rotation angles (of the camera)
T_b, T_r	Binarization and erosion thresholds of the noise filter
$u_1, u_2, \ldots, u_L$	Sequence of (unquantized) source symbols (rate-distortion theory)
$\breve{u}$	One specific source symbol (rate-distortion theory)
u,v,$\boldsymbol{u}$, $\boldsymbol{v}$	Arbitrary feature indices and positions
U	Time-discrete, amplitude-continuous source (rate-distortion theory)
$v_1, v_2, \ldots, v_L$	Sequence of (quantized) code symbols (rate-distortion theory)
$\breve{v}$	One specific code symbol (rate-distortion theory)
$\mathtt{W}_x$, $\mathtt{W}_y$, $\mathtt{W}_z$	Skew-symmetric matrices induced by rotation around the X-, Y-, and Z-axis
W	Search window
W_{H}	Window in the Harris corner detector
W_s	Bandwidth of signal s (rate-distortion theory)
x, y	Coordinates in x- and y-direction (in pel)
$\hat{x}$, $\hat{y}$	Perturbed x- and y-value
$\hat{x}'$, $\hat{y}'$	Perturbed x'- and y'-value
$\hat{x}'_{\mathrm{s}}$, $\hat{y}'_{\mathrm{s}}$	Perturbed x'_s- and y'_s-coordinates of the simplified affine model
$\hat{x}_{\mathrm{s}}$, $\hat{y}_{\mathrm{s}}$	Perturbed x- and y-value of the simplified affine model
x', y'	Projected/transformed x- and y-coordinates
x'_{s}, y'_{s}	Projected/transformed x- and y-coordinates of the simplified affine model

Abstract

Motion-compensated prediction is used in video coding standards like *High Efficiency Video Coding* (HEVC) as one key element of data compression. Commonly, a purely translational motion model is employed. In order to also cover non-translational motion types like rotation or scaling (zoom) contained in aerial video sequences such as captured from unmanned aerial vehicles, an affine motion model can be applied.

In this work, a model for affine motion-compensated prediction in video coding is derived by extending a model of purely translational motion-compensated prediction. Using the rate-distortion theory and the displacement estimation error caused by inaccurate affine motion parameter estimation, the minimum required bit rate for encoding the prediction error is determined. In this model, the affine transformation parameters are assumed to be affected by statistically independent estimation errors, which all follow a zero-mean Gaussian distributed probability density function (pdf). The joint pdf of the estimation errors is derived and transformed into the pdf of the location-dependent displacement estimation error in the image. The latter is related to the minimum required bit rate for encoding the prediction error. Similar to the derivations of the fully affine motion model, a four-parameter simplified affine model is investigated. It is of particular interest since such a model is considered for the upcoming video coding standard *Versatile Video Coding* (VVC) succeeding HEVC. As the simplified affine motion model is able to describe most motions contained in aerial surveillance videos, its application in video coding is justified. Both models provide valuable information about the minimum bit rate for encoding the prediction error as a function of affine estimation accuracies.

Although the bit rate in motion-compensated prediction can be considerably reduced by using a motion model which is able to describe motion types occurring in the scene, the total video bit rate may remain quite high, depending on the motion estimation accuracy. Thus, at the example of aerial surveillance sequences, a codec independent region of interest- (ROI-) based aerial video coding system is proposed that exploits the characteristic of such sequences. Assuming the captured scene to be planar, one frame can be projected into another using global motion compensation. Consequently, only new emerging areas have to be encoded. At the decoder, all new areas are registered into a so-called mosaic. From this, reconstructed frames are

extracted and concatenated as a video sequence. To also preserve moving objects in the reconstructed video, local motion is detected and encoded in addition to the new areas. The proposed general ROI coding system was evaluated for very low and low bit rates between 100 and 5000 $^{\text{kbit}}/_{\text{s}}$ for aerial sequences of HD resolution. It is able to reduce the bit rate by 90 % compared to common HEVC coding of similar quality. Subjective tests confirm that the overall image quality of the ROI coding system exceeds that of a common HEVC encoder especially at very low bit rates below 1 $^{\text{Mbit}}/_{\text{s}}$.

To prevent discontinuities introduced by inaccurate global motion estimation—as may be caused by radial lens distortion—a fully automatic in-loop radial distortion compensation is proposed. For this purpose, an unknown radial distortion compensation parameter that is constant for a group of frames is jointly estimated with the global motion. This parameter is optimized to minimize the distortions of the projections of frames in the mosaic. By this approach, the global motion compensation was improved by 0.27 dB and discontinuities in the frames extracted from the mosaic are diminished. As an additional benefit, the generation of long-term mosaics becomes possible, constructed by more than 1500 aerial frames with unknown radial lens distortion and without any calibration or manual lens distortion compensation.

Keywords: video coding, affine motion-compensated prediction (MCP), simplified affine motion-compensated prediction, rate-distortion theory, aerial surveillance, global motion compensation (GMC), region of interest- (ROI-) based aerial video coding, moving object detection, long-term mosaicking, radial distortion compensation

Kurzfassung

Bewegungskompensierte Prädiktion wird in Videocodierstandards wie *High Efficiency Video Coding* (HEVC) als ein Schlüsselelement zur Datenkompression verwendet. Typischerweise kommt dabei ein rein translatorisches Bewegungsmodell zum Einsatz. Um auch nicht-translatorische Bewegungen wie Rotation oder Skalierung (Zoom) beschreiben zu können, welche beispielsweise in von unbemannten Luftfahrzeugen aufgezeichneten Luftbildvideosequenzen enthalten sind, kann ein affines Bewegungsmodell verwendet werden.

In dieser Arbeit wird aufbauend auf einem rein translatorischen Bewegungsmodell ein Modell für affine bewegungskompensierte Prädiktion hergeleitet. Unter Verwendung der Raten-Verzerrungs-Theorie und des Verschiebungsschätzfehlers, welcher aus einer inexakten affinen Bewegungsschätzung resultiert, wird die minimal erforderliche Bitrate zur Codierung des Prädiktionsfehlers hergeleitet. Für die Modellierung wird angenommen, dass die sechs Parameter einer affinen Transformation durch statistisch unabhängige Schätzfehler gestört sind. Für jeden dieser Schätzfehler wird angenommen, dass die Wahrscheinlichkeitsdichteverteilung einer mittelwertfreien Gaußverteilung entspricht. Aus der Verbundwahrscheinlichkeitsdichte der Schätzfehler wird die Wahrscheinlichkeitsdichte des ortsabhängigen Verschiebungsschätzfehlers im Bild berechnet. Letztere wird schließlich zu der minimalen Bitrate in Beziehung gesetzt, welche für die Codierung des Prädiktionsfehlers benötigt wird. Analog zur obigen Ableitung des Modells für das voll-affine Bewegungsmodell wird ein vereinfachtes affines Bewegungsmodell mit vier Freiheitsgraden untersucht. Ein solches Modell wird derzeit auch im Rahmen der Standardisierung des HEVC-Nachfolgestandards *Versatile Video Coding* (VVC) evaluiert. Da das vereinfachte Modell bereits die meisten in Luftbildvideosequenzen vorkommenden Bewegungen abbilden kann, ist der Einsatz des vereinfachten affinen Modells in der Videocodierung gerechtfertigt. Beide Modelle liefern wertvolle Informationen über die minimal benötigte Bitrate zur Codierung des Prädiktionsfehlers in Abhängigkeit von der affinen Schätzgenauigkeit.

Zwar kann die Bitrate mittels bewegungskompensierter Prädiktion durch Wahl eines geeigneten Bewegungsmodells und akkurater affiner Bewegungsschätzung stark reduziert werden, die verbleibende Gesamtbitrate kann allerdings dennoch relativ

hoch sein. Deshalb wird am Beispiel von Luftbildvideosequenzen ein *Regionen-von-Interesse-* (ROI-) basiertes Codiersystem vorgeschlagen, welches spezielle Eigenschaften solcher Sequenzen ausnutzt. Unter der Annahme, dass eine aufgenommene Szene planar ist, kann ein Bild durch globale Bewegungskompensation in ein anderes projiziert werden. Deshalb müssen vom aktuellen Bild prinzipiell nur noch neu im Bild erscheinende Bereiche codiert werden. Am Decoder werden alle neuen Bildbereiche in einem gemeinsamen Mosaikbild registriert, aus dem schließlich die Einzelbilder der Videosequenz rekonstruiert werden können. Um auch lokale Bewegungen abzubilden, werden bewegte Objekte detektiert und zusätzlich zu neuen Bildbereichen als ROI codiert. Die Leistungsfähigkeit des ROI-Codiersystems wurde insbesondere für sehr niedrige und niedrige Bitraten von 100 bis 5000 $^{\text{kbit}}/_{\text{s}}$ für Bilder in HD-Auflösung evaluiert. Im Vergleich zu einer gewöhnlichen HEVC-Codierung kann die Bitrate um 90 % reduziert werden. Durch subjektive Tests wurde bestätigt, dass das ROI-Codiersystem insbesondere für sehr niedrige Bitraten von unter 1 $^{\text{Mbit}}/_{\text{s}}$ deutlich leistungsfähiger in Bezug auf Detailauflösung und Gesamteindruck ist als ein herkömmliches HEVC-Referenzsystem.

Um Diskontinuitäten in den rekonstruierten Videobildern zu vermeiden, die durch eine durch Linsenverzeichnungen induzierte ungenaue globale Bewegungsschätzung entstehen können, wird eine automatische Radialverzeichnungskorrektur vorgeschlagen. Dabei wird ein unbekannter, jedoch über mehrere Bilder konstanter Korrekturparameter gemeinsam mit der globalen Bewegung geschätzt. Dieser Parameter wird derart optimiert, dass die Projektionen der Bilder in das Mosaik möglichst wenig verzerrt werden. Daraus resultiert eine um 0.27 dB verbesserte globale Bewegungskompensation, wodurch weniger Diskontinuitäten in den aus dem Mosaik rekonstruierten Bildern entstehen. Dieses Verfahren ermöglicht zusätzlich die Erstellung von Langzeitmosaiken aus über 1500 Luftbildern mit unbekannter Radialverzeichnung und ohne manuelle Korrektur.

Stichwörter: Videocodierung, affine bewegungskompensierte Prädiktion, vereinfachte affine bewegungskompensierte Prädiktion, Raten-Verzerrungs-Theorie, Luftbildüberwachung, globale Bewegungskompensation, Regionen-von-Interesse-(ROI-) basierte Luftbildcodierung, Bewegtobjektdetektion, Langzeitmosaikerstellung, Radialverzeichnungskorrektur

1 Introduction

For aerial surveillance tasks, e. g. for disaster area monitoring as well as for police surveillance operations, unmanned aerial vehicles (UAVs) become more prevalent nowadays. One of the main challenges hereby is the transmission of high resolution video data recorded on-board an UAV over channels with only limited capacities. Taking into account the high resolutions of today's and upcoming camera sensors (4K and above), the demand for multiple or multi-view video streams, and the increasing number of UAVs competing for bandwidth, efficient data compression is of growing interest.

Modern hybrid video coding standards like *Advanced Video Coding* (AVC) [49], or *High Efficiency Video Coding* (HEVC) [51] provide very good video compression capabilities for daily life applications like Digital Video Broadcasting (DVB) [104] over satellite (DVB-S/DVB-S2), cable (DVB-C/DVB-C2) or terrestrial antenna (DVB-T/DVB-T2). Furthermore, video on demand (VOD) applications like Netflix, Amazon Prime Video, Maxdome, or Telekom Entertain TV, and also internet video applications like Youtube depend on high video compression performance. However, those video compression standards are natively optimized for the compression of video sequences as produced by commercial movie production studios or home-brew videos such as captured with a smartphone, camcorder or other digital movie cameras. They reduce the redundancy contained in a video sequence by a combination of motion-compensated prediction (MCP), transform coding with quantization, both typically realized in a *differential pulse-code modulation* (DPCM) loop, and entropy coding (Fig. 1.1) [104]. The usage of DPCM (the closed back-loop in the center of Fig. 1.1) ensures that the prediction, i. e. the motion compensation, is performed on quantized signals. Since a decoder reconstructs the image also on these quantized signals, both reconstructions are exactly the same. Consequently, diverging reconstructions in the en- and decoder are impossible, and thus, error propagation is prevented. MCP exploits that most parts of one video image (further on referred to as *frame*) reoccur in preceding or subsequent frames of the sequence. Instead of a pixel-wise representation of a certain, typically rectangular, image part (called *block*), only a reference to a similar image block is stored (motion vector, MV). For the most often used lossy coding schemes, the remaining pixel-wise prediction error is

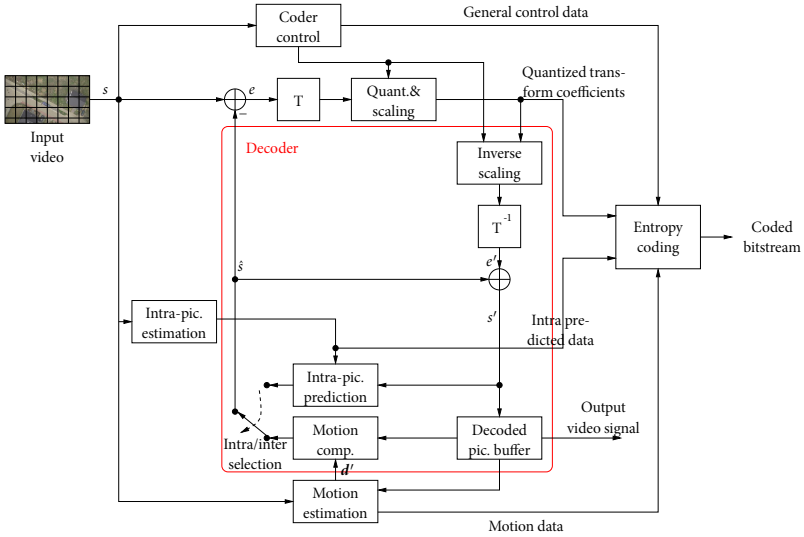


Figure 1.1: Block diagram of a hybrid video coder at the (simplified) example of a HEVC encoder (based on [30, 112]).

transformed ("T" in Fig. 1.1, whereas "T^{-1}" indicates the inverse transform) using a decorrelating transform. Typically, a *discrete cosine transform* (DCT) is applied and the resulting coefficients are quantized ("Quant. & scaling" in Fig. 1.1 and accordingly "Inv. Scaling" indicating inverse scaling) afterwards. The motion information, the quantized transform coefficients as well as additional signaling data needed for video decoding (e. g. video dimensions, frame rate, block partitioning etc.) are entropy encoded, e. g. by using a *context-adaptive binary arithmetic coding* (CABAC). For the first frame of a video sequence, which is intrinsically new, or blocks, for which no appropriate candidate for motion-compensated prediction is found, *intra-frame coding* or just *intra coding* can be applied as an alternative. Intra coding uses only the current frame and thus—in contrast to *inter-frame coding* or just *inter coding* such as applied in MCP—requires no other frames. Depending on the video coding standard, for intra coding different coding modes may be used, e. g. spatial prediction like angular prediction, planar mode or DC mode in HEVC [97], or *pulse-code modulation* (PCM) encoding. Using a rate-distortion optimization (RDO), several encoding possibilities with different block sizes and partitioning as well as coding modes are tested and the one which provides the best bit rate with respect to the introduced distortion is selected for final coding.

1.1 Motion-Compensated Prediction

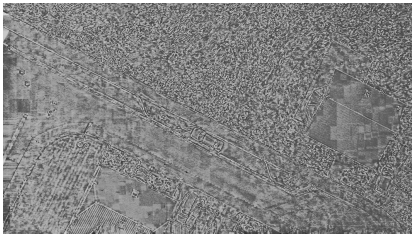
As previously introduced, one of the key elements for data compression in hybrid video coding standards like AVC or HEVC is motion-compensated prediction (MCP). It is based on the idea that the amount of data can be reduced, if for each image block of the current frame only the displacement vector referring to a temporally pre- or succeeding (reference) frame and the remaining error after prediction (*prediction error*) is encoded instead of the content of the block itself. Since for video sequences captured at typical frame rates between 24 and 60 frames per second (fps) the same content is visible in many frames, the coding efficiency using inter frame coding with MCP is much higher compared to that of intra frame coding. More specific, MCP does not attempt to describe the real motion of a block, but rather searches for the corresponding block with the highest similarity, i. e. with the lowest distortion, typically measured as *mean squared error* (MSE) or *sum of absolute differences* (SAD). For a highly accurate prediction, the prediction error is small (or optimally zero) and the entropy of the prediction error is smaller than for an inaccurate prediction. Consequently, also the minimum required bit rate for encoding the prediction error depends on the accuracy of the motion estimation, which can be specified by the variance of the displacement estimation error. The minimum bit rate of the prediction error of motion-compensated prediction as a function of the variance of the displacement estimation error was analyzed by Girod already in 1987 [36]. He assumed that the displacement estimation errors Δx and Δy in x- and y-direction are uncorrelated which only holds true for translational motion. Translational motion is relatively easy to estimate and describes most of the block motion for general videos sufficiently accurately. Consequently, Girod modeled the displacement estimation error for translational motion with two degrees of freedom. Such a motion model was employed in video coding standards like H.261 [52], MPEG-1 [47], MPEG-2 [50], H.263 [53], AVC [49] and HEVC [51].

For video sequences with distinct global motion, affine *global motion compensation* (GMC) was introduced in *MPEG-4 Advanced Simple Profile (MPEG-4 ASP)* [48], which can also cover rotation, scaling (i. e. zooming) and shearing. Since the coding efficiency gains of GMC stayed behind the expectations for general video coding for natural scenes without prevalent global motion, GMC was removed from the MPEG-4 ASP successor AVC again and replaced by an improved *motion vector prediction* (MVP).

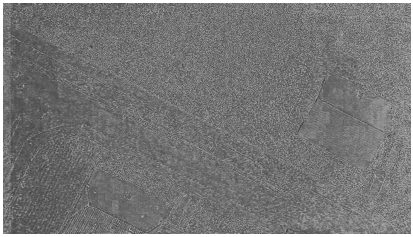
With upcoming small and relatively cheap UAVs like multicopters, aerial video sequences with distinct global motion that cannot be covered by a purely translational motion model, become increasingly important. The importance of such sequences is also reflected in recent test sets, which contain more aerial video sequences than

(a) Frame 6 of the aerial sequence *350 m sequence*

(b) Frame 7 of the aerial sequence *350 m sequence*



(c) Prediction error for frame 7 using HM (QP 43) (log.)



(d) Prediction error between (a) and (b) using GMC as proposed (log.)

Figure 1.2: In (a) and (b) two frames of the video sequence named *350 m sequence* from the TAVT data set [46, 81] are shown. Panel (c) shows the logarithmic (log.) prediction error (for definition see footnote on next page) using the block-based HEVC reference encoder HM and (d) the logarithmic prediction error using (affine) global motion compensation (GMC) as proposed. The prediction error in (c) is much higher and more irregular compared to the prediction error of the proposed GMC-based system in (d). The highest errors occur at non-planar structures (like the house at the right side), which cannot accurately be covered by the motion models in both cases—although much better using GMC in contrast to the translational motion model. Since for very low bit rates an accurate encoding of the prediction error becomes impossible, visible artifacts occur in reconstructed frames. Thus, a more consistent prediction error as shown in (d) is more preferable leading to a better reconstructed frame.

traditional video test sets, e. g. as used for the standardization of AVC or HEVC [13, 135, 136, 46]. For illustration, Fig. 1.2 shows two frames of the aerial video sequence named *350 m sequence* (with reference to the flight altitude from which it was recorded) from the *TNT Aerial Video Testset* (TAVT) data set [46, 81] in (a) and (b). The logarithmic

prediction error[1] using the block-based HEVC reference encoder HM is shown in (c) and the logarithmic prediction error using affine global motion compensation (GMC) in (d) at a similarly low bit rate. The prediction error in (c) is considerably larger and more irregular compared to the prediction error of the proposed affine GMC-based system in (d). The highest errors occur at non-planar structures (like the house at the right-hand side), which cannot accurately be covered by the motion models in both cases—although affine GMC yields much better results compared to the translational motion model. Since for low bit rates an accurate encoding of the prediction error becomes impossible, visible artifacts occur in the reconstructed frame. Thus, a more consistent prediction error as shown in (d) is more preferable leading to a better reconstructed frame.

To improve the processing of such higher-order global motions, the *ITU-T/ISO/IEC Joint Video Exploration Team (JVET)* (on Future Video Coding) incorporated a simplified 4-parameter affine motion model [65] (also referred to as *similarity* with four degrees of freedom, e. g. by Hartley and Zissermann [42]) into their (former) reference software *Joint Exploration Model (JEM)* [55] again [18], whereas in contrast to MPEG-4 ASP, it operates on a block-level. Affine motion compensation is also part of the video codec (coder-decoder) AV1 [96, 5].

First investigations on the common test set [110] (containing no sequences consisting of distinct motion which cannot be covered by a purely translational model) show coding efficiency gains of up to 1.35 % [134, 3]. Larger gains of more than 20 % can be expected for sequences containing more higher-order motions [65].

Although affine global motion compensation has a long tradition in video coding, it has not been theoretically analyzed thoroughly in the context of video coding. Particularly the assumption of Girod of uncorrelated displacement estimation errors $\Delta x'$ and $\Delta y'$ (in the original work called Δx and Δy) in x- and y-direction cannot be applied for non-translational global motion.

Thus, in this work, the rate-distortion function for video coding using affine global motion compensation is derived by extending the work of Girod [36] towards affine motion compensation and correlated displacement estimation errors $\Delta x'$ and $\Delta y'$. For this purpose the displacement estimation error during motion estimation is modeled and the bit rate after application of the rate-distortion theory is obtained (Chapter 3).

[1] The logarithmic prediction error $e_{\log}$ is calculated from the prediction error e as:

$e_{\log} = \operatorname{round}\left(127 + 128 \cdot \frac{\log_{10}\left(1+\operatorname{abs}(e)\right)}{\log_{10}(256)} \cdot \operatorname{sign}(e)\right)$ with "abs" denoting the absolute value of a number, "sign" the signum function and "round" a function rounding its argument towards the nearest integer.

1.2 Challenges for Aerial Surveillance Video Coding

With general video coding employing motion-compensated prediction, the bit rate for encoding high resolution content (full HD resolution of 1920×1080 pel, recorded at a minimum of 24 fps) of several megabit per second for subjectively "good" quality remains quite high. Additionally taking into account the demand for multi-cameras for aerial surveillance, it becomes obvious that a further bit rate reduction is necessary.

1.2.1 Region of interest-based video coding

In order to reduce the bit rate of the video to be coded while maintaining interesting image content, *region of interest* (ROI) coding is commonly applied, spatially dividing each frame of a video sequence into ROIs and non-ROIs. Both, ROIs and non-ROIs, are treated differently during (or before) encoding. Hereby, the quality of the regions of interest remains unaffected. Non-ROI areas of a frame could be blurred in a preprocessing step prior to actual video encoding or coarsely quantized within the video encoder itself to reduce the overall bit rate [59, 28, 19]. A modified or externally controllable block-based hybrid video coder like AVC or HEVC is employed in [68, 128, 67, 127] and [129, 74], respectively, in order to apply different quantization parameters for the coding of ROI and non-ROI blocks. Such encoder internal modifications typically require severe changes and thus are time-consuming and expensive. In already existing hardware implementations, subsequent coding control modifications are even impossible to apply.

The drawback of typical ROI coding approaches as discussed above is the degradation of non-ROI areas that cannot be reconstructed at full quality at the decoder. To overcome this limitation and to provide high resolution and quality over the entire reconstructed frame, it is proposed to only encode and transmit new emerging image content (*new areas*, ROI-NAs) for each of the frames. Since only small parts of each frame have to be encoded, this ROI coding system is capable of providing a high image quality at low bit rates. The new areas are stitched together in a mosaicking step at the decoder to reconstruct the static parts of the scene (background) by means of global motion compensation. From this panoramic image, a video sequence can be reconstructed [75, 79] (Section 4.1).

The quality of such a panoramic image—and consequently of the reconstructed parts of the video frames as well—may be impaired by lens distortions like radial distortion, since non-fitting new areas lead to visible artifacts. Especially radial distortion is a common lens characteristic for zoom and wide-angle cameras like used in aerial surveillance, and thus should be considered during mosaicking.

To also retain local motion not conforming with the global motion, such areas have to be detected, additionally transmitted and appropriately handled at the decoder.

1.2.1.1 Moving object detection

Although, theoretically, ROIs can be arbitrarily defined, e. g. in the center of the image or by detecting skin color in a teleconferencing system like in [115], more context-sensitive approaches are desirable. Depending on the specific task, dedicated detectors may be used to find areas containing interesting objects or subjects in the video, e. g. cars, houses, faces, people, etc., which could be later-on defined as ROIs. For aerial surveillance scenarios, *moving objects* (MO) are often considered as ROI, further on referred to as ROI-MO. Popular approaches rely on global motion compensation of the background pixels (pixels are also referred to as *pels* for *picture elements* as in this work) due to the camera movement prior to calculation of the pel-wise image differences (difference image) between two frames of the video sequence or between the current frame and a reconstructed background reference image [56, 107, 17, 45]. More efficient detectors were proposed, which exploit parallax effects [58], utilize block matching motion vectors [33], cluster moving image features [117], or use an optical flow analysis in order to detect moving objects [131, 90]. In [62] and [116] extensive overviews on recent publications in the field of aerial surveillance with a moving camera and appropriate moving object detection methods are provided.

Since the focus of this work lies on *efficient aerial video coding* on-board an UAV with limited energy and computational resources, a simple, yet effective difference image-based moving object detector is used here. Due to the modular concept of the proposed detection and coding framework, the moving object detector can easily be replaced (Section 4.2).

1.2.1.2 Radial distortion in aerial video sequences

For motion-compensated prediction induced by global motion of the camera, camera aberrations may impair the accurate estimation of the motion, which leads to an increased prediction error and thus finally results in an increased bit rate. Moreover, the generation of overview panoramic images from several subsequent frames, which is one common way of visualizing aerial video sequences, becomes impossible without lens distortion correction [94, 133, 130].

Radial distortion has been determined as one of the most important aberrations [124, 26]. There has been plenty of research about radial distortion and radial distortion compensation [8, 121, 26, 31, 76]. Also in computer vision, radial distortion has to be compensated depending on specific application requirements [114]. Most correction methods rely on some kind of test pattern to calibrate a lens at a given

focal length. However, calibration pattern based methods like [31] can be applied only for known cameras. In aerial surveillance, the camera type and parameters are often unknown and thus have to be estimated from the video sequence. In [122], it was proposed to estimate the complete camera matrix including the radial distortion. This method is based on the estimation of projective homographies from corresponding image feature points, but it is restricted to static scenes and limited degrees of freedom and thus not appropriate for aerial surveillance applications with a moving camera. In contrast to that, in [26] an approach to estimate the radial distortion based on edge detection and subsequent polygonal approximation was proposed in order to first detect straight lines. In the second step, the distortion error of different estimated radial distortion parameters is iteratively minimized while taking the straightness of detected lines in the image into account. However, in aerial surveillance applications, it cannot be guaranteed that straight lines *are* in the image and that those lines are indeed exactly *straight*. Consequently, a method not relying on specific image structures is more preferable. For an accurate global motion estimation between two frames affected by unknown (and theoretically) different radial distortions, the radial distortion parameters have to be jointly estimated with the global motion. A frame-to-frame-based approach was proposed and combined with *Random Sample Consensus* (RANSAC) for noise robustness of camera-captured signals [61]. However, for image sequences with more than two frames, a frame-to-frame-based method tends to estimate different radial distortions for different pairs of subsequent images, especially for noisy signals. Since changing radial distortion parameters from frame to frame negatively influence the global motion estimation accuracy, it is desirable to keep the radial distortion parameters constant as long as possible. Moreover, a constant radial distortion reflects the property of a real camera, where the radial distortion for one specific focal length is constant (Section 4.1.2.1). In order to estimate constant radial distortions for a high number of subsequent frames, the joint estimation of homographies for several frames with one common radial distortion is proposed (Section 4.1.3).

1.3 Contributions

The contributions of this work are as follows:

1. The first contribution of this work is the analysis of motion-compensated prediction using an affine motion model. For a fully affine motion model with six degrees of freedom, the prediction error after motion compensation as a function of the affine transformation parameter accuracy is analytically

derived. The affine parameters are assumed to be independently estimated and, as a worst-case assumption, independently perturbed by zero-mean Gaussian noise. Using the rate-distortion theory [7], the minimum required bit rate for encoding the prediction error is derived.

Similar considerations are made for a simplified affine motion model with only four degrees of freedom (rotation, scaling, translation) as employed in JEM. Since the assumption of independently estimated affine transformation parameters cannot be met for the simplified model, the correlation between the estimated parameters has to be specifically considered.

The derivations for both models were previously published in [87] for the fully affine model and in [88] for the simplified affine model.

2. A region of interest-based video coding system (*ROI-based coding system*) for aerial video sequences is introduced. Exploiting the special characteristic of (predominant) planarity of aerial videos, global motion compensation is employed to reconstruct areas of each frame, which are already known to the encoder. Only new emerging areas (*new areas*, NA or ROI-NA) are encoded. At the decoder-side, NAs are stitched together and video frames are reconstructed from the resulting mosaic. Areas containing local motion (ROI-MO) are detected on-board, additionally encoded, transmitted and properly inserted into the reconstructed video. In contrast to common video coding standards, errors introduced by global motion compensation due to non-planar ground structures like trees or buildings are not encoded, but are tolerated in favor of a reduced bit rate. Thus, the bit rate for encoding aerial sequences is highly reduced compared to a common HEVC video encoding without subjectively negatively impairing the image quality.

   The ROI coding system including the simple moving object detector was previously published in [75] using a modified AVC video encoder. A similar system employing a HEVC encoder instead was published in [89, 79, 81]. A codec-independent general ROI-coding approach is presented which enables the use of the proposed ROI-based coding system for aerial videos with arbitrary video codecs. Since no encoder modification is necessary, general ROI coding facilitates the easy replacement of the video encoder itself to exploit latest efficiency improvements. The general ROI coding approach was previously published in [85].

   Task-dependent moving object detector improvements for the proposed system were published in [77, 78, 81] and are shortly summarized in this work.

3. A long-term mosaicking approach is presented, which is robust against unknown radial distortion as well as smaller violations of the planarity assumption, as caused by 3D structures like houses or trees. A model for the joint estimation of several homographies and one constant radial distortion is developed. Due to the computational complexity of the solution, a fast, iterative algorithm is proposed. Based on geometric constraints, the projection of a jointly estimated group of frames (GOF) is regularized. Thereby the radial distortion parameter is not necessarily optimized to match the correct radial distortion but to provide a decent projection of the frames into the mosaic.

   The long-term mosaicking approach was previously published in [83].

1.4 Outline

This thesis is organized as follows: in Chapter 2, basic principles are introduced. Aiming at aerial surveillance video coding, camera models with their extrinsic and intrinsic parameters as well as projection models are summarized. After a review of general hybrid video coding with a focus on motion-compensated prediction, the rate-distortion theory is revisited as far as used in this work, before region of interest-based video coding is introduced. In Chapter 3, the efficiency of motion-compensated prediction is analyzed for a fully as well as a simplified affine motion model and compared to the efficiency of a purely translational motion model using the example of aerial sequences containing distinct global motions. A ROI-based coding system for aerial video sequences exploiting the special characteristics of such sequences is presented in Chapter 4. By use of global motion compensation of already known content, the bit rate is reduced below the bit rate which standardized common video coders can provide at a subjectively comparable quality. It is explained how the global motion is estimated at the encoder-side and compensated at the decoder-side by means of a (short-term) mosaic. To retain also locally moving objects like cars or pedestrians, a moving object detector suitable for UAV on-board processing is incorporated into the system. Experimental results are presented in Chapter 5: the model from Chapter 3 is experimentally validated in Section 5.1 by measurements of the prediction error bit rate for inaccurate affine motion estimation (Section 5.1.1). Operational rate-distortion diagrams for real-world sequences encoded with and without affine motion-compensated prediction are presented in Section 5.1.2. The ROI coding system from Chapter 4 is evaluated in Section 5.2. It is shown that the ROI coding system outperforms state-of-the-art video coding systems in terms of objective and subjectively perceived quality. In Section 5.2.3 finally results of the in-loop radial distortion compensation as introduced in Section 4.1.3 are presented. Chapter 6 summarizes and concludes this work.

2 Basics

In this chapter, the fundamentals of this work are introduced. First, the scene and camera model (Section 2.1 and 2.2, respectively) as used here are described. The latter comprises perspective projection (Section 2.2.1), a lens model including radial distortion (Section 2.2.2), the sensor model (Section 2.2.3) as well as the mathematical essentials of homogeneous coordinates and the mapping from world to camera coordinates as far as relevant for this work (Section 2.2.4 and 2.2.5, respectively). Later on, the projective transformation and the basics of homography mappings are introduced in Section 2.3. Motion estimation from image sequences is explained in Section 2.4, covering feature detection, feature tracking and RANSAC outlier removal. The idea of mosaicking of aerial video sequences is shortly presented in Section 2.5. Hybrid video coding incorporating motion-compensated prediction and also global motion compensation is encompassed in Section 2.6, prior to discussion of the rate-distortion theory in Section 2.7 as a basis for the affine motion-compensated prediction in video coding in the next chapter. Finally, region of interest-based coding is reviewed in Section 2.8. The Sections 2.1–2.4 are developed and partly quoted from the work of Munderloh [90]. The Subsections 2.4.2–2.4.3 are based on [15] and [90]. The Section 2.6 is based on the work of Klomp [60] and Section 2.7 is based on [92].

2.1 Scene Model

The landscape model used in this work assumes the surface of the earth to be planar. This holds true as long as the camera is located high enough above the ground, but not so high that the curvature of the earth becomes significant. Moreover the focal length of the camera needs to be sufficiently small (Fig. 2.1). This is given for small and medium UAVs with a fixed, downwards-facing camera (nadir view) of a full-frame equivalent focal length between 50 and several hundred millimeters, and the flight altitude is expected to be between approximately 100 and 2000 meters. Furthermore, it is assumed that the predominant area of each video frame represents the surface of the earth and that the heights of 3D objects in the scene are small compared to the flight altitude. Such assumptions are met for typical drone missions in rural or suburban regions. Even hilly terrain is sufficiently flat in the above sense, since at

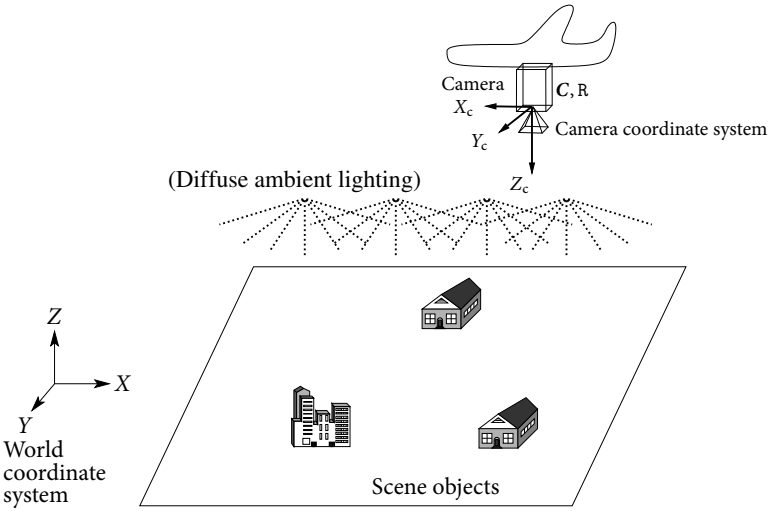


Figure 2.1: Scene model (based on [90], buildings from [21]).

typical surveillance video frame rates of about 24–60 fps the assumption of scene planarity is still valid between several subsequent frames. Without loss of generality, the illumination is assumed to be a constant, diffuse ambient lighting. Hence a scene without shadowing, reflection and other lighting effects is assumed.

The coordinate systems are identically defined as in [90]: the world coordinate system (X, Y, Z) is a fixed, global coordinate system which can be used to uniquely describe every point within the world. The local camera coordinate system (X_c, Y_c, Z_c) moves with the camera. The origin of this local camera system is set to the center of projection, also known as the camera center [90]. It is assumed that the X- and Y-axes of the local camera system are aligned to the camera sensor and the Z-axis of the right-handed orthogonal coordinate system is pointing downwards through the center of the lens towards the scene. The mapping of camera coordinates to world coordinates can be performed by applying a rotation $\mathtt{R}$ which indicates the local orientation of the camera coordinate system with respect to the world coordinate system, and the position of the camera center $\boldsymbol{C}$ in world coordinates [90].

2.2 Camera Model

The camera model in this work is the same as in [90]. It explains the projection of a 3D scene onto a 2D image plane of a camera. It is described as a combination of a perspective projection model, a lens model, and a sensor model [120, 90]. A scene point $\tilde{\boldsymbol{P}}$ in camera coordinates is projected into the image point $\boldsymbol{n}$ on the image plane in image coordinates by using the camera model (Fig. 2.2).

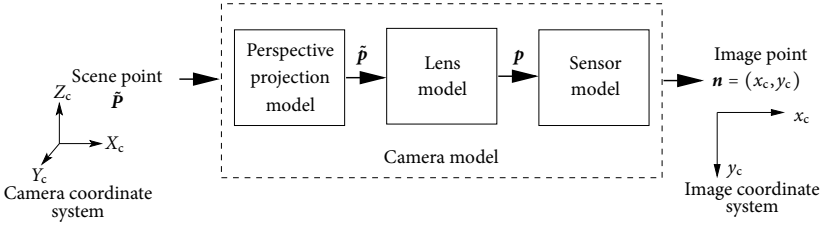


Figure 2.2: Camera mapping model (based on [120] and [90]).

2.2.1 Perspective projection

A perspective projection describes the mapping of a 3D object point $\tilde{\boldsymbol{P}}$ to a 2D point $\boldsymbol{p}$ in camera coordinates with an ideal pinhole camera (Fig. 2.3). The distance between

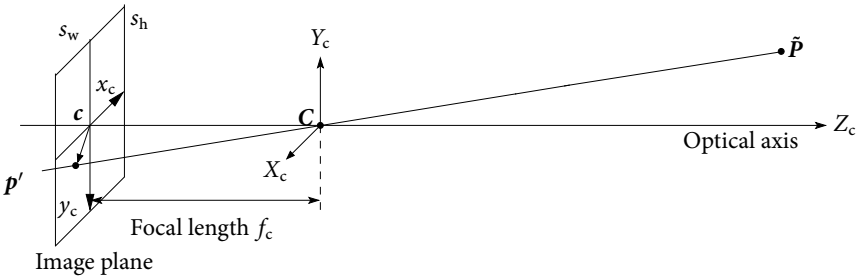


Figure 2.3: Pinhole camera model (based on [90]).

the camera center and the image plane is called focal length f_c. The point on the image plane intersected by the optical axis is called the principal point $\boldsymbol{c}$. It is often assumed to be the origin of the coordinate system of the image plane, as it holds

also true in this work. Without loss of generality the image plane can be mirrored to the other side of the pinhole at an equidistant distance f_c, resulting in the geometry as shown in Fig. 2.4. This results in uninverted image coordinates of the projected scene points. As all rays intersect at this central point, this representation is called central projection. Then the camera coordinates of the projection $\boldsymbol{p} = (x_c, y_c, f_c)^\top$ of the object point $\tilde{\boldsymbol{P}} = (P_x, P_y, P_z)^\top$ in camera coordinates on the image plane can be determined using the intercept theorems [120, 22, 90]

$$\begin{pmatrix} x_c \\ y_c \end{pmatrix} = \frac{f_c}{P_z} \cdot \begin{pmatrix} P_x \\ P_y \end{pmatrix} . \tag{2.1}$$

In Equation (2.1), the projection is carried out by assuming a plane at distance P_z from the camera center $\boldsymbol{C}$, parallel to the image plane at distance f_c. The mapping is performed by scaling the remaining two space coordinates of the point at distance P_z by the ratio of their distances $\frac{f_c}{P_z}$ [90].

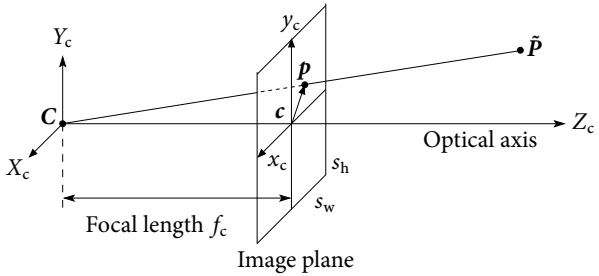


Figure 2.4: Central projection [90].

2.2.2 Lens model

For real cameras, however, pinhole cameras can neither be realized nor would they provide enough light for a proper projection. Thus, camera lenses are used instead of a pinhole. Although the paths of rays are different in an ideal lens (dashed lines in Fig. 2.5) compared to those of a pinhole camera, the rays converge at the same spot on the image plane and the projection is equal to that of a pinhole camera.

In contrast to ideal lenses, real lenses are affected by several different distortions. In the camera model, this is treated in the lens model block (Fig. 2.2). The main geometric distortion is the radial distortion [26, 124]. Following the common notation

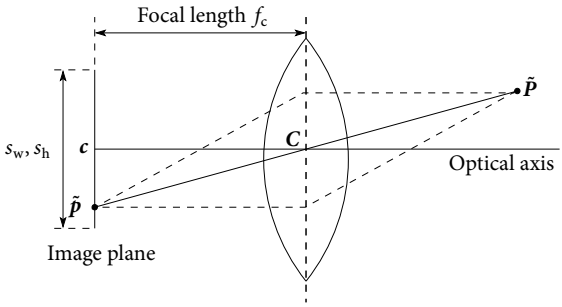


Figure 2.5: Simplified camera and lens model [90].

and description of [90], the non-linear projection between the perspective mapping coordinates $\tilde{\boldsymbol{p}} = (x_d, y_d)^\top$ and the real image coordinates on the image plane $\boldsymbol{p}$ can be approximated by a Taylor expansion

$$r = r_d(1 + \kappa_1 r_d^2 + \kappa_2 r_d^4 + \cdots) , \tag{2.2}$$

with $r_d = \sqrt{(x_d^2 + y_d^2)}$ being the distorted and $r = \sqrt{(x^2 + y^2)}$ being the undistorted distance of one point on the image plane from the center of distortion. Without loss of generality, the center of distortion is often assumed to be located in the origin of the camera coordinate system. For several applications it is a sufficiently accurate approximation to only consider the second order radial distortion with its coefficient κ_1 [124, 8]. Then, (2.2) can be simplified to

$$\boldsymbol{p} = (1 + \kappa_1 r_d^2)\, \tilde{\boldsymbol{p}} , \tag{2.3}$$

with $r_d^2 = x_d^2 + y_d^2$ being the squared distance of $\tilde{\boldsymbol{p}}$ from the origin [90].

2.2.3 Sensor model

Modern image sensors are typically realized as active pixel sensors manufactured using CMOS (complementary metal-oxide semiconductor) technology. Each picture element (pel, pixel) has a light sensitive photo element in addition to an (eponymous) active amplifier and can be read-out individually in theory. The numbers of pels of a sensor in horizontal and vertical direction are defined as N_x (columns, image width) and N_y (rows, image height), respectively. Pels are typically counted starting

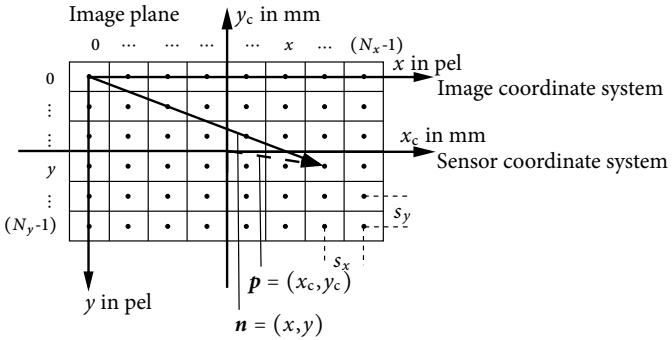


Figure 2.6: Camera sensor model (based on [120, 90]).

from the upper left (Fig. 2.6). A full HD resolution image sensor has $N_x = 1920$ columns and $N_y = 1080$ rows. The sensor has its own coordinate system which is assumed to be aligned to the local camera coordinate system with its center in the center of the sensor (Fig. 2.6). The information of the pels is quantized and stored. Assuming an 8-bit quantization, the amount of data of one entire frame (only luminance) is $N_x \times N_y$ bytes (unit byte, abbreviation B) or for a full HD resolution image $1920 \times 1080 \approx 2$ megabytes (MB). Since the dimensions of a pel in horizontal and vertical direction may be different, two scaling factors are defined by

$$s_x = \frac{s_\mathrm{w}}{N_x} \quad \text{and} \quad s_y = \frac{s_\mathrm{h}}{N_y} \,, \tag{2.4}$$

with s_w and s_h being the width and the height of the sensor. Then, the mapping from image plane into sensor coordinates is defined as in (2.5) and from the sensor coordinates into image coordinates as in (2.6) [90].

$$\begin{aligned} x_\mathrm{c} &= s_x \cdot x - 0.5(N_x - 1)s_x \,, \\ y_\mathrm{c} &= s_y \cdot y - 0.5(N_y - 1)s_y \,; \end{aligned} \tag{2.5}$$

$$\begin{aligned} x &= {}^1\!/_{s_x} \cdot x_\mathrm{c} + 0.5(N_x - 1) \,, \\ y &= {}^1\!/_{s_y} \cdot y_\mathrm{c} + 0.5(N_y - 1) \,. \end{aligned} \tag{2.6}$$

2.2.4 Homogeneous coordinates

Homogeneous coordinates are used in projective geometry to describe the projection between two planes in 3D space. In projective geometry, geometric operations like affine transformations become simple matrix-vector multiplications which often are more convenient than the component-wise calculation.

A 2D image point $(a,b)^\top$ can be represented by its homogeneous form $(a,b,1)^\top$. The projection line through the camera center $\boldsymbol{C}$ and the 3D space point $(a,b,1)^\top$ intersects the 2D point in the image plane. Any other point located on this line is also a valid homogeneous representation of the 2D point $(a,b)^\top$ and thus, $(a,b,1)^\top \simeq (\frac{a}{k},\frac{b}{k},k)^\top$ (e. g. line connecting $\boldsymbol{C}$ and $\tilde{\boldsymbol{P}}$ in Fig. 2.4), where $\simeq$ represents projective identity [90].

The points $\boldsymbol{p} = (x_\mathrm{c},y_\mathrm{c},f_\mathrm{c})$ and $\tilde{\boldsymbol{P}} = (P_x,P_y,P_z)$ in Fig. 2.4 are located at the same vector starting at $\boldsymbol{C}$ and have the same homogeneous coordinate:

$$\boldsymbol{p} = \begin{pmatrix} x_\mathrm{c} \\ y_\mathrm{c} \\ f_\mathrm{c} \end{pmatrix} \simeq \begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix} = \tilde{\boldsymbol{P}} \,. \tag{2.7}$$

$\boldsymbol{P}$ and $\boldsymbol{p}$ hereby are equivalent up to a scalar multiple.

To project the homogeneous coordinate back to 2D, the homogeneous coordinate has to be normalized by its third component:

$$\frac{1}{f_\mathrm{c}} \begin{pmatrix} x_\mathrm{c} \\ y_\mathrm{c} \\ f_\mathrm{c} \end{pmatrix} = \frac{1}{P_z} \begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix} . \tag{2.8}$$

The 2D space coordinates now equal the homogeneous 3D coordinate without its third dimension [90]:

$$\boldsymbol{p} = \mathrm{K}\,\tilde{\boldsymbol{P}} \,, \tag{2.9}$$

with the matrix K which contains the inner camera parameters. The inner camera parameters are the focal length f_c and the image plane offset $\boldsymbol{c} = (c_x, c_y)^\top$. K therefore is referred to as the camera calibration matrix [90]:

$$\mathrm{K} = \begin{bmatrix} f_\mathrm{c} & 0 & c_x \\ 0 & f_\mathrm{c} & c_y \\ 0 & 0 & 1 \end{bmatrix} . \tag{2.10}$$

2.2.5 World coordinates to camera coordinates

The mapping of a point in camera coordinates is modeled with (2.9). For the projection of an arbitrary scene point $\boldsymbol{P} = (X,Y,Z)^\top$ in world coordinates, its coordinates have to be converted into corresponding camera coordinates $\tilde{\boldsymbol{P}}$. This is carried out by rotation and translation of the coordinate axes as follows: first, the world coordinate system is moved so that the camera center becomes the new origin by subtracting the position of the camera center $\boldsymbol{C}$ in the world coordinate system from the observed scene point $\boldsymbol{P}$. Next, the orientation of this translated axis and the camera coordinate system are aligned by rotation around each of the axes. The angles of rotation around the X-, Y-, and Z-axis are denoted by θ_x, θ_y, and θ_z, respectively. A 3×3 rotation matrix $\mathtt{R}$ can be used to describe all three rotations, with the order of rotation Y, X, Z:

$$\mathtt{R} = \mathtt{R}_\theta \mathtt{R}_\gamma \mathtt{R}_\beta = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.11}$$

With the orientation $\mathtt{R}$, the mapping from $\boldsymbol{P}$ to $\tilde{\boldsymbol{P}}$ is

$$\tilde{\boldsymbol{P}} = \mathtt{R}(\boldsymbol{P} - \boldsymbol{C}) \,. \tag{2.12}$$

Inserting (2.12) into (2.9) leads to the projection of a point $\boldsymbol{P}$ in world coordinates into a point $\boldsymbol{p}$ on the image plane

$$\boldsymbol{p} \quad = \quad \mathtt{K}\mathtt{R}(\boldsymbol{P} - \boldsymbol{C}) \,. \tag{2.13}$$

The offset of the camera center $\boldsymbol{C}$ and the rotation matrix $\mathtt{R}$ describe the position and orientation of the camera in the world coordinate system and thus are called the extrinsic camera parameters [90].

2.3 Projective Transformation and Homography

A projective transformation is an invertible linear mapping from projective space to itself that maps straight lines to straight lines. A projective transformation, also

referred to as homography, can be described by a 3×3 matrix $\mathtt{H}$ [125]:

$$\begin{pmatrix} x_1' \\ x_2' \\ x_3' \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \tag{2.14}$$

or in matrix notation:

$$\boldsymbol{x}' = \mathtt{H}\,\boldsymbol{x}\,. \tag{2.15}$$

Image coordinates can be obtained through dividing by the last dimension, assuming $x_1 = x$, $x_2 = y$ and $x_3 = 1$:

$$x' = \frac{x_1'}{x_3'} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \qquad y' = \frac{x_2'}{x_3'} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}. \tag{2.16}$$

A homography can be decomposed into its components [42]

$$\mathtt{H} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ q_1 & q_2 & q \end{bmatrix} = \begin{bmatrix} \mathtt{A} & \boldsymbol{t} \\ \boldsymbol{q}^\top & q \end{bmatrix}, \tag{2.17}$$

with $\boldsymbol{t}$ being the translational vector, $\mathtt{A}$ an affine two dimensional scaling and rotation and q a scaling factor which is commonly 1 after normalization of $\mathtt{H}$. The vector $\boldsymbol{q} = (q_1, q_2)^\top$ describes non-linear properties of a projectivity. The matrix $\mathtt{H}$ has eight degrees of freedom (DoFs).

In computer vision applications, projective transformations can be used to describe the mapping of points between two planes in space. For instance, points on a planar surface in space can be projected onto the image plane of a camera. Projective transformations can also be employed to map corresponding 2D points obtained by projected 3D points located on a planar surface in space (Fig. 2.7). For the latter case, the projective transformation between two image planes can be considered as two concatenated projective transformations ($\mathtt{H}_1$ and $\mathtt{H}_2$) [42] as illustrated in Fig. 2.7:

$$\boldsymbol{x}' = \mathtt{H}_2\mathtt{H}_1^{-1}\boldsymbol{x} = \mathtt{H}_{21}\boldsymbol{x}\,. \tag{2.18}$$

$\mathtt{H}_{21}$ has eight degrees of freedom, thus four independent, non-collinear point correspondences are necessary to estimate the projective transformation between two planes (see Section 2.4).

For the special case of $\boldsymbol{q} = (0,0)^\top$ in (2.17), the projective transformation is an affine

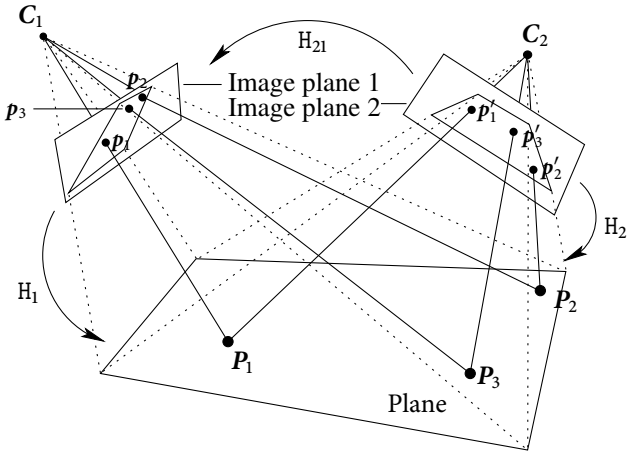


Figure 2.7: Homography mappings $\mathtt{H}_1$ and $\mathtt{H}_2$ between the points $\boldsymbol{P}_1$, $\boldsymbol{P}_2$ and $\boldsymbol{P}_3$ located on a plane in 3D space and two different image planes to the points $\boldsymbol{p}_1$, $\boldsymbol{p}_2$ and $\boldsymbol{p}_3$ on image plane 1 and $\boldsymbol{p}'_1$, $\boldsymbol{p}'_2$ and $\boldsymbol{p}'_3$ on image plane 2, respectively (based on [90]).

transformation with six degrees of freedom. The inhomogeneous affine transformation equations are

$$x' = h_{11}x + h_{12}y + h_{13} \quad \text{and} \quad y' = h_{21}x + h_{22}y + h_{23}\,. \tag{2.19}$$

For several real-world applications, a projective transformation can be approximated by an affine transformation as will be also performed in this work (Chapter 3). For the estimation of the affine transformation (6 DoFs) between two planes, three independent, non-collinear point correspondences are required [90].

2.4 Motion Estimation from Image Sequences

The estimation of the global motion of an image sequence is a common issue in computer vision. Typically, it is performed in a three-step approach: suitable image regions (features) are located and selected in the current frame s_k (Section 2.4.1), relocated in the next frame s_{k+1} (tracking, Section 2.4.2), and pruned of potentially false feature correspondences (outlier removal, Section 2.4.3) in the third step. This

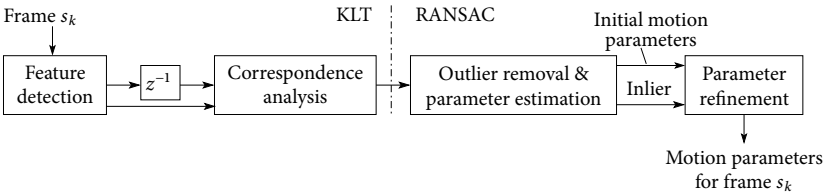


Figure 2.8: Feature tracking workflow used in this work (based on [90]).

results in a sparse motion vector field which can be further used, e. g. for motion compensation.

This procedure can either be performed by relocation of the features from the frame with largest temporally distance but still located inside the current frame (often used in structure from motion (SfM) applications), or on a frame-to-frame basis. The latter technique (as shown in Fig. 2.8) is applied in this work as it is known to increase the accuracy compared to the previously described approach [22, 90].

2.4.1 Feature detection

A feature is a specific descriptor at a specified position in an image. It is designed to uniquely describe a certain image region even in noisy environments. The position is specified in image coordinates of the current frame s_k. In computer vision, common feature types are corner features [41], *scale-invariant feature transform* (SIFT) features [69] and *histogram of oriented gradients* (HOG) features [24]. Corner features are defined by high luminance gradients in two orthogonal directions (commonly applied in a surrounding window) and thus, they are largely invariant against rotation, perspective distortions or illumination changes. SIFT features consist of 128 dimensions and are designed to occur at local extrema in a scale space of a difference of Gaussians (DoG), providing additional robustness against scale changes. HOG features are similar to SIFT features but are computed on an uniformly spaced grid [90]. For this work, a high feature localization accuracy is of highest importance for the best possible (global) motion estimation. Thus, corner features were selected as they provide the best localization accuracy [22][90].

2.4.1.1 Harris & Stephens / Shi & Tomasi corner detector

For the Harris & Stephens corner detector, features are defined as high gradients in horizontal as well as vertical direction. These features can be used to establish reliable

correspondences between images. To calculate the image gradient at a position $\boldsymbol{n} = (x,y)^\top$ in the image, the partial derivatives I_x and I_y of the image intensity I are determined:

$$\nabla I(\boldsymbol{n}) = \begin{pmatrix} I_x(\boldsymbol{n}) \\ I_y(\boldsymbol{n}) \end{pmatrix} = \begin{pmatrix} I_x(x,y) \\ I_y(x,y) \end{pmatrix} = \begin{pmatrix} \frac{\partial I(x,y)}{\partial x} \\ \frac{\partial I(x,y)}{\partial y} \end{pmatrix} \approx \begin{pmatrix} \frac{I(x+1,y)-I(x-1,y)}{2} \\ \frac{I(x,y+1)-I(x,y-1)}{2} \end{pmatrix}. \tag{2.20}$$

Next, the "cornerness" $\mathtt{M}$ of the image point $\boldsymbol{n}$ is determined, considering the local neighborhood W_H (WH typically is a squared window with 7×7 pel) centered around $\boldsymbol{n}$:

$$\mathtt{M}(\boldsymbol{n}) = \begin{bmatrix} \sum\limits_{\boldsymbol{n}\in W_\mathrm{H}} I_x^2(\boldsymbol{n}) & \sum\limits_{\boldsymbol{n}\in W_\mathrm{H}} I_x(\boldsymbol{n}) I_y(\boldsymbol{n}) \\ \sum\limits_{\boldsymbol{n}\in W_\mathrm{H}} I_x(\boldsymbol{n}) I_y(\boldsymbol{n}) & \sum\limits_{\boldsymbol{n}\in W_\mathrm{H}} I_y^2(\boldsymbol{n}) \end{bmatrix}. \tag{2.21}$$

Analyzing the eigenvalues λ_1 and λ_2 of $\mathtt{M}(\mathtt{n})$ results in structural information of $\boldsymbol{n}$ and its surrounding W as follows:

$$\begin{array}{lcl} \lambda_1 \approx \lambda_2 \approx 0 & : & \text{no structure} \\ \lambda_1 \approx 0, \lambda_2 \gg 0 & : & \text{edge} \\ \lambda_1 \gg 0, \lambda_2 \approx 0 & : & \text{edge} \\ \lambda_1 \gg 0, \lambda_2 \gg 0 & : & \text{corner}\,. \end{array} \tag{2.22}$$

A corner can be considered, if the eigenvalues λ_1 and λ_2 are large. To quantify the quality of edges, a *corner response function* (CRF) was additionally introduced as

$$\begin{aligned} \mathrm{CRF}(\boldsymbol{n}) &= \det \mathtt{M}(\boldsymbol{n}) - k_\mathrm{H} \left(\operatorname{trace} \mathtt{M}(\boldsymbol{n})\right)^2 \\ &= \lambda_1\lambda_2 - k_\mathrm{H} \left(\lambda_1 + \lambda_2\right)^2, \end{aligned} \tag{2.23}$$

with k_H being an empirically optimized value (according to [120] a typical value of 0.04) [90]. All points exceeding a predefined CRF threshold S_CRF are sorted by their CRF value and stored in a list. Additionally, a distance check for all already processed features is performed and features, which are located in a predefined minimal surrounding M_CRF are discarded to enable a more uniform distribution of the features within the image. Finally, the B_CRF best points of the list are chosen as image features, where B_CRF represents the maximum number of feature points per frame [22]. In [108], Shi and Tomasi proposed to approximate the CRF in (2.23) by the simpler $\min(\lambda_1,\lambda_2)$ operation [90].

2.4.2 Correspondence analysis by Kanade-Lucas-Tomasi feature tracking

For the determination of correspondences, detected feature points can be tracked using a Kanade-Lucas-Tomasi (KLT) feature tracker [108]. This method is used to estimate correspondences between the feature points from one frame s_{k-1} to another frame s_k with sub-pel accuracy. For the underlying Lucas-Kanade [71] method the displacement $\boldsymbol{d} = (d_x, d_y)^\top$ of pels between two subsequent frames is assumed to be small and purely translational, which is realistic for typical video frame rates of 24 fps or larger. In this case the optical flow [43] between feature points (including their surroundings) can be described by the 2D representation of the optical flow equation

$$I_k(\boldsymbol{n} + \boldsymbol{d}) = I_{k-1}(\boldsymbol{n}) \; . \tag{2.24}$$

To determine the displacement $\boldsymbol{d}$ of a feature point $\boldsymbol{n}$ between the frames $k-1$ and k, the sum of squared differences (SSD) of image intensities inside the block W (e. g. of size 7×7) is calculated and the cost function

$$\epsilon_d(\boldsymbol{d}) = \sum_{\boldsymbol{n} \in W} \left(I_k(\boldsymbol{n} + \boldsymbol{d}) - I_{k-1}(\boldsymbol{n}) \right)^2 \tag{2.25}$$

has to be minimized by variation of $\boldsymbol{d}$. This results in the estimated displacement

$$\hat{\boldsymbol{d}} = \arg\min_{d} \left(\epsilon_d(\boldsymbol{d}) \right) . \tag{2.26}$$

To enable sub-pel accuracy, the image signal at position $\boldsymbol{n} + \boldsymbol{d}$ at frame k is expressed as spatially continuous signal:

$$I_k(\boldsymbol{n} + \boldsymbol{d}) \approx I_{k-1}(\boldsymbol{n}) + \nabla^\top I_{k-1}(\boldsymbol{n}) \boldsymbol{d} + \frac{\delta I_{k-1}(\boldsymbol{n})}{\delta t} t_k \; . \tag{2.27}$$

The derivative of the image signal with respect to time can be approximated by ([15])

$$\frac{\delta I_{k-1}(\boldsymbol{n})}{\delta t} \approx \frac{I_k(\boldsymbol{n}) - I_{k-1}(\boldsymbol{n})}{t_k} \; . \tag{2.28}$$

The cost function (2.25) is minimized via insertion of (2.27) by using (2.28):

$$\epsilon_d(\boldsymbol{d}) = \sum_{\boldsymbol{n} \in W} \left(\nabla^\top I_{k-1}(\boldsymbol{n}) \boldsymbol{d} + I_k(\boldsymbol{n}) - I_{k-1}(\boldsymbol{n}) \right)^2 \; . \tag{2.29}$$

To calculate the extrema of (2.29), (2.29) is partially differentiated with respect to dx and dy and set to zero. This leads to the linear equation system

$$\mathtt{M}_{k-1}\hat{\boldsymbol{d}} = \boldsymbol{g}_{k-1}\,, \tag{2.30}$$

where, analogous to (2.20) and (2.21),

$$\begin{aligned}
\mathtt{M}_{k-1} &= \sum_{\boldsymbol{n}\in W} \nabla I_{k-1}(\boldsymbol{n})\, \nabla^{\top} I_{k-1}(\boldsymbol{n}) \\
&= \begin{bmatrix} \sum\limits_{\boldsymbol{n}\in W} I_{x\,k-1}^{2}(\boldsymbol{n}) & \sum\limits_{\boldsymbol{n}\in W} I_{x\,k-1}(\boldsymbol{n})\, I_{y\,k-1}(\boldsymbol{n}) \\ \sum\limits_{\boldsymbol{n}\in W} I_{x\,k-1}(\boldsymbol{n})\, I_{y\,k-1}(\boldsymbol{n}) & \sum\limits_{\boldsymbol{n}\in W} I_{y\,k-1}^{2}(\boldsymbol{n}) \end{bmatrix} \qquad (2.31) \\
\boldsymbol{g}_{k-1} &= -\sum_{\boldsymbol{n}\in W} \big(I_k(\boldsymbol{n}) - I_{k-1}(\boldsymbol{n})\big)\, \nabla I_{k-1}(\boldsymbol{n}) \\
&= -\sum_{\boldsymbol{n}\in W} \big(I_k(\boldsymbol{n}) - I_{k-1}(\boldsymbol{n})\big) \begin{pmatrix} I_{x\,k-1}(\boldsymbol{n}) \\ I_{y\,k-1}(\boldsymbol{n}) \end{pmatrix}. \qquad (2.32)
\end{aligned}$$

M_{k-1} describes the spatial derivatives of the image intensity function and $\boldsymbol{g}_{k-1}$ contains the temporal derivatives. By transformation of (2.30), the estimated displacement value $\hat{\boldsymbol{d}}$ of a feature point is determined to

$$\hat{\boldsymbol{d}} = \mathtt{M}_{\mathtt{k-1}}^{-1}\, \boldsymbol{g}_{k-1}\,. \tag{2.33}$$

Since the image signal was linearly approximated, this result holds only true for small displacements $\boldsymbol{d}$. To overcome this issue, the Newton-Raphson method [102] can be employed for an iterative displacement estimation as in [70]. The spatially continuous image signal is used to adjust the linear approximation of the image signal to the current estimated displacement. The spatially continuous image signal is generated from four adjacent sample values using a bilinear interpolation filter [32]. The displacement of the iteration $\hat{\boldsymbol{d}}_i$ is therefore calculated from the previous iteration as

$$\hat{\boldsymbol{d}}_{i+1} = \hat{\boldsymbol{d}}_i + \mathtt{M}_{\mathtt{k-1}}^{-1} \sum_{\boldsymbol{n}\in W} \left(\Big(I_{k-1}(\boldsymbol{n}) - I_k\big(\boldsymbol{n} + \hat{\boldsymbol{d}}_i\big)\Big)\, \nabla I_{k-1}(\boldsymbol{n}) \right), \tag{2.34}$$

until the change in $\boldsymbol{d}$ is smaller than a threshold (commonly 0.01 pel) or the maximum number of iterations is reached.

To improve the estimation of larger displacements of the feature points between

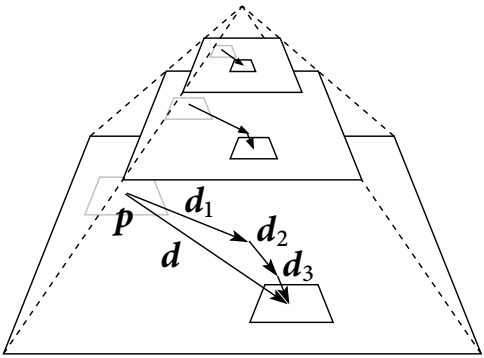


Figure 2.9: Image pyramid for handling of large displacements [90].

two frames, hierarchical estimation can be applied [40, 9]. As illustrated in Fig. 2.9, an image resolution pyramid is employed with several low-pass filtered and down-sampled representations of the image intensities. The number of resolution stages depends on the original intensity image resolution and the maximum displacement which is likely to occur. The displacement estimation using (2.33) starts at the lowest resolution stage and is refined in the iteration steps employing (2.34) and using the higher resolution stage. To achieve this, the feature position $\boldsymbol{n}$ and displacement $\boldsymbol{d}$ need to be appropriately scaled to each pyramid stage (see Fig. 2.9) [90]. Without loss of generality, the translational motion model can be replaced by an affine motion model as proposed by [108], although it is less stable due to the higher number of degrees of freedom [90].

2.4.3 Outlier removal: random sample consensus (RANSAC)

For the calculation of an accurate H matrix, reliable point correspondences are required. Since the estimated feature point correspondences (see Section 2.4.2) may be partly wrong, e. g. due to features located on moving objects, or are affected by noise (called outlier), an outlier detector has to be employed in order to remove these false correspondences from the set of detected correspondences (inliers). As outlier detector the *random sample consensus* (RANSAC) method [35] is suitable. It consists of 3 steps: first, a random set of correspondences of minimal size for the required motion model is drawn (e. g. three points with two coordinates each for an affine model) from all available correspondences. Second, the transformation is calculated

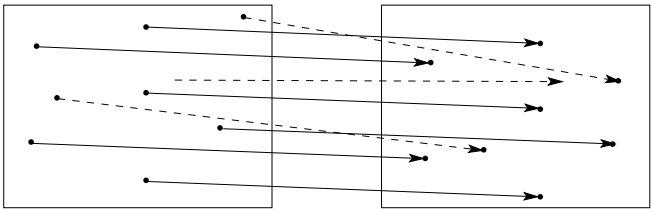


Figure 2.10: RANSAC outlier removal. Dashed correspondences are removed as they do not conform to the majority determined model (solid arrows) [90].

according to the motion model, e. g. using the *direct linear transform* (DLT) [42]. Third, all remaining correspondences are tested for conformity with the current estimated projection. The calculated transformation is applied to each of the remaining correspondences not involved in the transformation parameter estimation from step two above. Each mapping located outside a predefined environment ($\epsilon > \epsilon_{\max}$) is considered as an outlier, otherwise it is kept as an inlier (Fig. 2.10). These three steps are repeated until the mean squared mapping error (MSE) of all inliers is minimal and the amount of inliers is maximal [90].

In a final step, the sum of squared Euclidean distances is minimized using a least squares approach [123, 15, 90].

2.5 Mosaicking of Aerial Videos

For the stitching of a panoramic image from an aerial video sequence, also called mosaicking in this context, the global motion between the video frames has to be estimated. For aerial sequences recorded from sufficiently high flight altitudes with a common focal length for aerial surveillance between 50 and several hundred millimeters (as defined in Section 2.1), the assumption of planarity is valid for the surface of the earth. Thus, the global motion estimation can be performed as described in Section 2.4. One common possibility for the generation of a panoramic image is to use the coordinate system of one of the frames (e. g. the first in the sequence) as a reference coordinate system and registering all other frames with respect to this reference coordinate system (called *flat panorama* in [113]). For non integer-pel global motion displacements, an interpolation of the image intensity values based on their adjacent pels has to be performed, e. g. using bilinear [102] or Lanczos filtering [29].

2.6 Hybrid Video Coding

General hybrid video coding is based on two principles. First, the frame s_k to be encoded is predicted from already decoded data. Since also the decoder is able to reconstruct the same predicted frame, only the difference to the original frame (called *prediction error*) has to be encoded and transmitted. Second, the prediction error is transformed to exploit spatial correlations between neighboring pels and to allow for a perceptual adapted quantization.

The simplified block diagram of a hybrid video encoder has already been shown in Fig. 1.1 (page 2). Intra prediction transform coding and quantization as well as entropy coding have been shortly reviewed at the beginning of Chapter 1. Since they are not relevant for this work, they will not be considered in more detail.

Typically, a frame is divided into small blocks and each block is encoded sequentially. If the prediction of a block is based only on already encoded content of the current frame, e. g. neighboring blocks, this block is called *intra* coded and if all blocks of one frame are intra encoded, this frame is called an intra frame (I-frame). In contrast to that in *inter* coding also information from other, already encoded frames is employed. Inter coded frames are distinguished in unidirectionally predicted frames (P-frames) and bidirectionally predicted frames (B-frames). Blocks in P-frames only use information from one encoded frame, whereas in B-frames more than one encoded frame is used for the prediction of a block in the current frame.

In a first step, for every block the motion $\boldsymbol{d}$ is estimated between the current frame s_k to be encoded and the already encoded frames of the decoded picture buffer (also known as reference image buffer, block "Decoded Pic. Buffer" in Fig. 1.1). Using the determined motion vectors $\boldsymbol{d}'$, a motion-compensated frame $\hat{s}_k$ is calculated from the reference images. The motion vectors $\boldsymbol{d}'$ have to be encoded and transmitted from the encoder to the decoder. Next, the prediction error e is calculated as the difference between s_k and $\hat{s}_k$, transformed and quantized. The quantized prediction error e' is called residuum for a clear differentiation. Finally, the quantized transform coefficients are encoded using an entropy encoding like CABAC.

As indicated by the red rectangle in Fig. 1.1, a decoder is also integrated in the encoder. This ensures that both, encoder and decoder, reconstruct identical frames s'_k by combining the residuum e' with the motion-compensated frame $\hat{s}_k$. This frame is also saved in the reference image buffer for the decoding of subsequent images (Fig. 1.1) [60].

2.6.1 Motion-compensated prediction

In contrast to intra coding, where one block may only be predicted from the current frame, any reference image can be employed for motion-compensated prediction in inter coding (Fig. 2.11). For the encoding of one block the encoder signalizes the

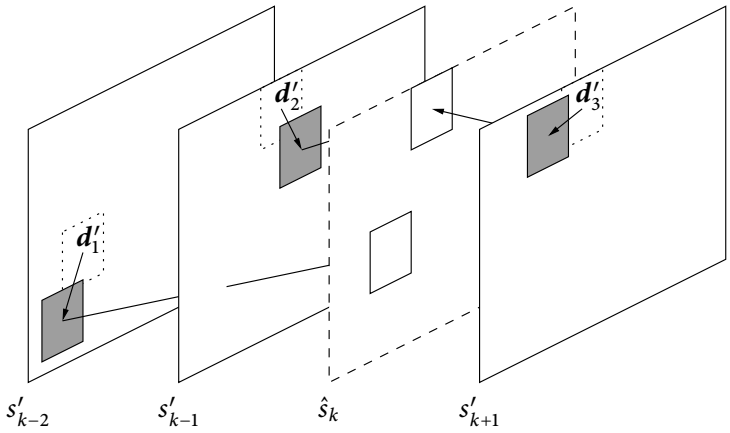


Figure 2.11: Inter prediction of $\hat{s}_k$ from the gray blocks of the already encoded reference images. Dashed blocks visualize the corresponding positions of the blocks in the current frame [60].

position of the block within the reference image, which is used for the prediction. In addition to the motion vector $\boldsymbol{d}'_i$ it has to be known at the decoder which reference image was used. Finally the residuum is added to compensate the prediction error e.

In the current video coding standard HEVC a quarter-pel accuracy is employed for the motion vectors [112].

Since for each block (at least) one motion vector has to be stored and transmitted, the block size should be as large as possible to reduce the total number of motion vectors of one frame. However, for larger blocks the prediction accuracy is decreased and the residual signal increases. To find the optimal ratio between residual signal bit rate and motion vector bit rate, each block can be independently split into smaller blocks like shown in Fig. 2.12 [60].

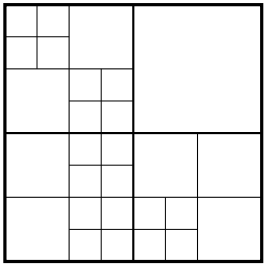


Figure 2.12: Example of a block splitting [60].

2.6.2 Global motion compensation

As an alternative to block-based motion compensation as described in Section 2.6.1, motion compensation can also be applied on entire frames to compensate global motion introduced by camera motion. It was already employed for general video coding in MPEG-4 ASP [100]. Later on it was removed due to the superior coding performance of the improved motion vector prediction in AVC.

With emerging new application scenarios like drone-captured videos with a prevalent global motion, (affine as well as homographic) GMC has recently been reconsidered as a coding tool in the video codec AV1 [96, 5]. Moreover, depending on the application scenario, GMC may also be beneficial in combination with HEVC [27] and presumably as well with its potential successor *Versatile Video Coding* (VVC) [14, 132].

2.7 Rate-Distortion Theory

The rate-distortion (RD) theory enables the calculation of a lower bound for the bit rate of a given source, which allows the reconstruction of the message by complying to a predefined maximum permitted average distortion D. The most important benefit of the RD theory is that the rate of a specific distortion serves as a lower bound independent of any particular implementation, e. g. of quantization. The following introduction in RD theory is based on [92].

Let $u_1, u_2, \ldots, u_L$ be a sequence of L source symbols, where each u represents any arbitrary symbol $\breve{u}$ of a time-discrete, amplitude-continuous source U. The source symbol $\breve{u}$ originates from a continuous range of values $-\infty \leq \breve{u} \leq +\infty$. Let $q(u)$ be the probability density function of the source symbols u. By source coding, the original sequence of source symbols may be represented by a sequence of K code symbols

$v_1, v_2, \ldots, v_K$ with the probability density function $p(v)$, where each v represents any arbitrary symbol $\check{v}$. A distortion measure $d(u;v)$ is defined which assigns a numerical value to any pair of source symbols u and code symbols v. The distortion is assumed to be caused solely by the source encoder, i. e. the quantization. Assuming that $d(u;v) \geq 0$ for all u and v, then a large value of $d(u;v)$ corresponds to a large distortion of the symbol u.

The rate-distortion function now is defined for a time-discrete, amplitude-continuous source as

$$R(D) = \min_{p(v|u)} \int_u \int_v q(u) \cdot p(v|u) \cdot \log_2 \frac{p(v|u)}{p(v)} \, \mathrm{d}u \, \mathrm{d}v \tag{2.35}$$

and

$$\bar{d} = \int_u \int_v q(u) \cdot p(v|u) \cdot d(u;v) \, \mathrm{d}u \, \mathrm{d}v \tag{2.36}$$

with mean distortion $\bar{d} \leq D$ [92]. It can be shown that without exceeding a maximum allowed average distortion D by more than an arbitrarily small amount ϵ, a bit rate $R(D)$ can be realized for a source. Thus, $R(D)$ is considered as the lower bound of the bit rate of a source and a given distortion D [92].

Assuming a memoryless, time-discrete, amplitude-continuous Gaussian source with entropy $H_\mathrm{G}(U) = \frac{1}{2} \log_2(2\pi e \sigma_u^2)$ and variance σ_u^2, the rate-distortion theory yields

$$R_\mathrm{G}(D) = \frac{1}{2} \log_2 \frac{\sigma_u^2}{D} \tag{2.37}$$

for a squared deviation $d(u;v) = (v-u)^2$ as distortion measure [92].

In image and video coding, a stationary Gaussian source with memory is typically assumed due to correlations between neighboring pixels. Such a source with memory can be decomposed into a sum of memoryless sources with white power spectrum and variance $S(f) \, \mathrm{d}f$, assuming $S(f)$ to be the power spectral density of the Gaussian source with memory (Fig. 2.13). Let $N(f) \, \mathrm{d}f$ be the distortion of one single Gaussian source. Then, with (2.37) the average bit rate of a Gaussian source with memory follows as

$$R(D) = \frac{1}{2W_s} \int_{-W_s}^{+W_s} \frac{1}{2} \log_2 \frac{S(f)}{N(f)} \, \mathrm{d}f \, , \tag{2.38}$$

with frequency f and bandwidth W_s of the signal s, given that $S(f) > N(f)$ [92]. The distortion $N(f)$ of every single source must consist of a constant power density in

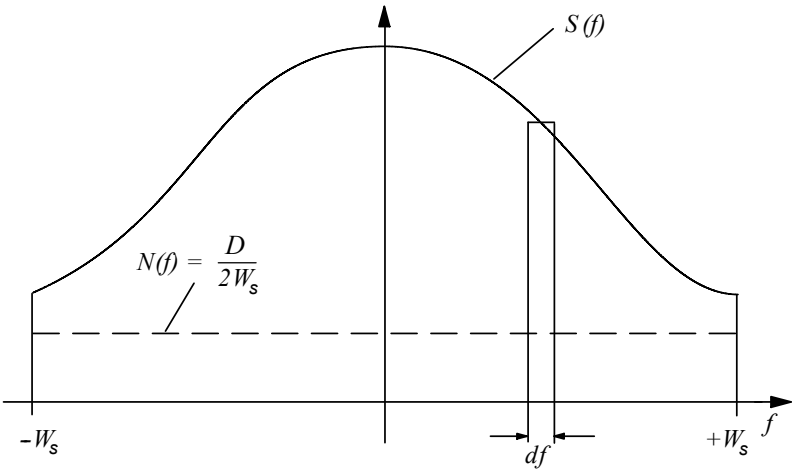


Figure 2.13: Decomposition of a Gaussian source with memory into a sum of memoryless Gaussian sources [92].

order to provide the minimum bit rate $R(D)$ for a given distortion D according to (2.35), where

$$\int_{-W_s}^{+W_s} N(f)\,\mathrm{d}f = D \tag{2.39}$$

has to apply [92].

Using variational calculus, $R(D)$ can be minimized as a function of $N(f)$, where (2.39) applies for $N(f)$. This leads to the rate-distortion function for Gaussian sources with memory

$$R(D) = \frac{1}{2W_s} \int_{-W_s}^{+W_s} \frac{1}{2} \log_2 \frac{S(f) \cdot 2W_s}{D}\,\mathrm{d}f\,. \tag{2.40}$$

Here, $N(f)$ is a constant according to

$$D = 2W_s \cdot N(f) = 2W_s \cdot \Theta\,, \tag{2.41}$$

where Θ represents the power spectral density of the noise [92]. Since the Gaussian distribution has the highest entropy among all infinite distributions with the same

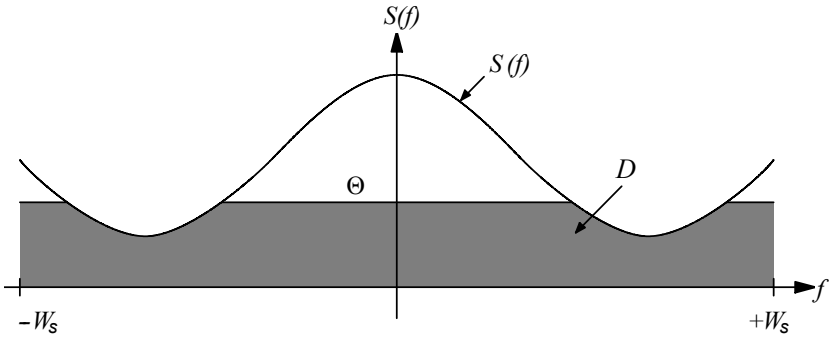


Figure 2.14: Illustration of the integrals D and $R(D)$ [92].

(finite) mean and variance [23], the rate-distortion function in (2.40) represents an upper bound for $R(D)$ for non-Gaussian sources with memory and same power spectral density [92].

The distortion D in a coding system is typically defined by the quantization error e_q or its variance σ_e^2, respectively. The quantization error e_q is the difference of the quantized prediction error e' and the unquantized prediction error e. For uniform quantization with step size Δx, the average distortion D is approximately

$$D = 2W_s \cdot \Theta = \sigma_e^2 \approx \frac{\Delta x^2}{12} \,. \tag{2.42}$$

In case of a coarse quantization, the step sizes Δx are large and Θ may become larger than $S(f)$ (Fig. 2.14). For those frequency intervals, the output signal v is set to 0. This results in the generation of a distortion D matching the power spectral density $S(f)$ and the corresponding bit rate $R(D)$ with

$$D = \int\limits_{f:\, S(f)>\Theta} \Theta \, \mathrm{d}f + \int\limits_{f:\, S(f)\le\Theta} S(f) \, \mathrm{d}f \tag{2.43}$$

and

$$R(D) = \frac{1}{2W_s} \int\limits_{f:\, S(f)>\Theta} \frac{1}{2} \log_2 \frac{S(f)}{\Theta} \, \mathrm{d}f \,. \tag{2.44}$$

$R(D)$ is obtained in bit per sample [92].

For evaluation of the rate-distortion theory, RD plots are typically employed which show bit rate over distortion. The latter is often represented as the peak signal-to-noise ratio (PSNR) in video coding:

$$\mathrm{PSNR} = 10 \cdot \log_{10} \frac{\mathrm{MAX}^2}{\mathrm{MSE}} , \tag{2.45}$$

with MAX being the maximum possible amplitude value of a pel and the mean squared error MSE. It should be noted that such measurement-based RD plots, which are often presented in the context of video coding, are technically only operational rate-distortion curves. Consequently, they do not represent a theoretical lower bound.

2.8 Region of Interest- (ROI-) based Video Coding

As explained in the previous section, hybrid video coding aims at the reconstruction of each entire frame contained in a video sequence. Limited by the rate-distortion theory (Section 2.7) it is only possible to achieve a specific image quality, often assessed in subjective tests or approximated by measuring the *peak signal-to-noise ratio* (PSNR), for a given rate. If the rate-constraints cannot be complied, e. g. because the transmission channels are too small, the image quality has to be reduced or otherwise the video cannot be transmitted.

One solution to this problem is region of interest (ROI) video coding, where only specific, interesting regions in each frame of a video sequence are encoded in high quality, whereas all non-ROI parts of the frame are encoded in a lower quality. Since low quality typically is accompanied by a smaller bit rate, the overall bit rate of a video sequence can be reduced while maintaining the high quality in the interesting regions.

2.8.1 ROI definition and detection

The definition of ROIs is highly application-specific. In the context of aerial video coding, ROIs are typically defined as moving objects. As already introduced in Section 1.2.1, plenty of related literature exists for sophisticated detection of moving objects. However, since the focus of this work is not on highly optimized moving object detection, a simple moving object detector is used. Consequently, in the next subsection the moving object detection by background subtraction is introduced (based on [90]).

2.8.1.1 Moving object detection by background subtraction

Background subtraction is one common possibility of detecting moving objects. It is based on the idea that if from a video frame containing moving objects and static objects (background) all the static objects are subtracted, only the moving objects remain. To model the background, simple approaches like in [4, 20] or more sophisticated approaches such as in [2, 1, 109, 101, 57] can be applied. If the camera is not static as assumed in the previously mentioned approaches, the ego-motion of the camera has to be compensated prior to the generation of the background model either once or continuously updated during runtime. These background-motion compensation algorithms are required to have a high accurate background-motion estimation, since an incorrect estimation would cause high differences when the wrongly compensated background is pel-wise subtracted from the real background in the frame. In such a case, not existing moving objects would be detected (false positive detections).

As it will be apparent later, for the proposed system false positive detections only slightly degrade the overall performance (as long as not too many false positive detections occur) and thus a simple background subtraction-based system is used. Moreover, a computational efficient algorithm is preferable over a more accurate but more computationally intensive algorithm for the desired purpose. Without loss of generality the moving object detector can be replaced by any other detector if necessary.

As illustrated in Fig. 2.15, the absolute pel-wise difference between a motion-compensated previous frame $\hat{s}_{k-\Delta k}$ and the current frame s_k (difference image) is computed.

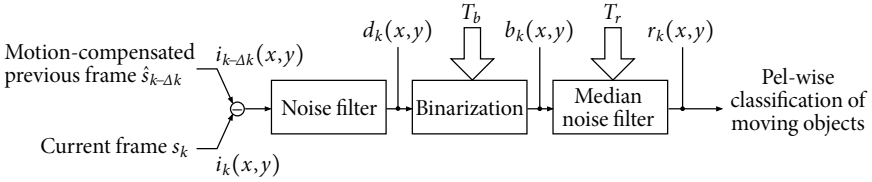


Figure 2.15: Background subtraction-based motion detection system [90].

As a noise filter the pel-wise differences are summarized in a 3×3 sliding window and binarized. To achieve this, the pel-wise sum is compared to a threshold T_b and converted to the pel-wise binary decision b_k:

$$
\begin{aligned}
b_k(x,y) &= \begin{cases} 1 & \text{for } d_k(x,y) \geq T_b \\ 0 & \text{for } d_k(x,y) < T_b \end{cases}, \text{with} \\
d_k(x,y) &= \sum_{y-1}^{y+1} \sum_{x-1}^{x+1} i_k(x,y) - i_{k-\Delta k}(x,y)
\end{aligned}
\tag{2.46}
$$

with $i_k(x,y)$ being the luminance intensity at the image coordinate (x,y) in frame s_k and $i_{k-\Delta k}(x,y)$ the luminance intensity in the frame $\hat{s}_{k-\Delta k}$ [90].

In the next step, a similarly working median filter is applied in a 16×16 window. If the energy e_k in the current window exceeds a predefined threshold T_r, the current pel is classified as (part of) a moving object ([90]):

$$
\begin{aligned}
r_k(x,y) &= \begin{cases} 1 & \text{for } e_k(x,y) \geq T_r \\ 0 & \text{for } e_k(x,y) < T_r \end{cases}, \text{with} \\
e_k(x,y) &= \sum_{y-\frac{W}{2}}^{y+\frac{W}{2}} \sum_{x-\frac{W}{2}}^{x+\frac{W}{2}} b_k(x,y) \; .
\end{aligned}
\tag{2.47}
$$

2.8.2 ROI encoding

Common general-purpose video encoders are optimized to provide a comparable visual quality over the entire frame or—in other words—to equally distribute the unavoidable error for lossy video coding over the entire frame. In contrast to that ROI coding distinguishes between regions of interest (ROI) and non-regions of interest (non-ROI) of a frame, which can be arbitrarily defined (Section 2.8.1). Based on the classification in ROI and non-ROI, the visual quality or, more precisely, the approximated quality by a quality measure like PSNR, is increased at the expense of the quality of non-ROI areas. A common way to realize such a ROI encoder is the alteration of the quantization parameter on a block-level, e. g. in an AVC encoder [68, 128, 67, 127] or in a HEVC encoder [129, 74], respectively. Either case typically results in a standard compliant bit stream but with highly degraded image quality for non-ROI areas.

2.8.2.1 Quality evaluation of ROI encoded videos

The judgment of the overall image quality may be challenging. It is a widely unsolved question how the low image quality of non-ROI areas compared to ROI areas influences

the overall image quality. Moreover, it is highly dependent on the specific quality-distribution as well as the application scenario. As the overall image quality over the entire frame can only be poorer for ROI coding compared to common general-purpose coding, it is common practice to only judge the quality of the ROI, assuming that non-ROI areas in fact contain no relevant information for the viewer and thus are negligible. Similar evaluation measures can be found in related literature like [103, 39, 38]. Also in this work, the image quality in the context of ROI coding is objectively evaluated by measuring the PSNR of ROIs only (ROI-PSNR). For the judgment of the overall image qualities, subjective tests may be additionally performed, as also done in this work.

3 Rate-Distortion Theory for Affine Motion Compensation in Video Coding

The largest contribution to the overall data rate of an encoded video stream in hybrid video coding is caused by the prediction error [60]. Thus, Bernd Girod modeled the minimum required bit rate for encoding the prediction error as a function of the motion estimation accuracy in his early work from [36]. In his work, Girod modeled the bit rate for a translational motion model and thus only for uncorrelated displacement estimation errors $\Delta x'$ and $\Delta y'$. With upcoming new application scenarios with video sequences containing distinct global and non-translational motion like aerial videos, it is beneficial to consider additional—non purely translational—motion models [134, 3, 65] as currently applied in the upcoming video coding standards VVC [14, 132] and AV1 [96, 5].

Although there is a long tradition of using higher-order motion models in video coding, no thorough theoretical analysis in the context of video coding currently exists. Thus, in this chapter an efficiency analysis of motion-compensated prediction is performed for a fully affine model with six degrees of freedom as well as for a simplified affine motion model. The latter motion model is of particular interest as it was and is investigated in the course of the standardization of the new video coding standard VVC in the draft evaluation software JEM [134, 3, 65, 64] by JVET [14, 132].

To model the minimum required bit rate for encoding the prediction error, two different influences have to be distinguished.

On the one hand, the model error itself has to be considered. The model error describes motions contained in the scene which cannot be covered by the selected motion model, i. e. an affine or simplified affine motion model in this case. Such model errors may occur for aerial videos recorded with cameras not facing vertically downwards in nadir view. Such global motion may be described only by a projective model. Moreover, in the case of global motion estimation, the estimation accuracy may be negatively influenced by local motion, e. g. due to features located on cars not recognized as outliers in aerial surveillance scenarios. However, for typical sequences, an affine motion model is a sufficiently accurate approximation. Furthermore, from a video coding point of view, the additional parameters of a more complex motion model do not necessarily justify its possible benefits. This is evidenced by the integra-

tion of a simplified model, which needs two parameters less to be encoded, instead of a fully affine motion model, in the next video coding standard. Hence, as a good compromise between a model which is able to perfectly describe a scene and coding of the additional parameters, the fully affine as well as the simplified affine motion models are considered in this work.

On the other hand, the estimation error of the motion estimation itself has to be considered. The estimation error of course depends on the specific implementation and restrictions like motion vector accuracy in common hybrid video coding—as analyzed in [36].

Both aspects will be considered in this work. As for the rate-distortion analysis the source of the perturbations does not matter, the derivations for both are the same and thus are conducted only once. Parts of this chapter including the derivations for the fully and simplified affine models have been already published in [87, 88].

This chapter is organized as follows: in Section 3.1 the prediction error bit rate as a function of the affine motion estimation accuracy is derived. In Section 3.2 similar derivations are performed for a simplified affine model with only four degrees of freedom as used in JEM. In Section 3.3 the findings from Sections 3.1 and 3.2 are summarized and conclusions are drawn.

3.1 Efficiency Analysis of Fully Affine Motion Compensation

The overview flow diagram in Fig. 3.1 illustrates the connections between the different components of the analysis within this section. The working steps are structured as follows:

1. First, the affine motion and the error model as used for further derivations are introduced (Section 3.1.1).

2. Second, the 2D probability density function (pdf) $p_{\Delta X', \Delta Y'}(\Delta x', \Delta y')$ of the displacement estimation errors in x- ($\Delta x'$) and y-direction ($\Delta y'$) is derived (right part in Fig. 3.1). Here, $\Delta X'$ and $\Delta Y'$ denote the random processes generating the $\Delta x'$ and $\Delta y'$. The Fourier transform of $p_{\Delta X', \Delta Y'}(\Delta x', \Delta y')$ is $P(\Lambda)$, which will be used for subsequent derivations. Λ here abbreviates the two-dimensional (2D) spatial frequency vector $\Lambda := (\omega_x, \omega_y)$ for reasons of clarity (Sections 3.1.2 and for the simplified affine model 3.2.1).

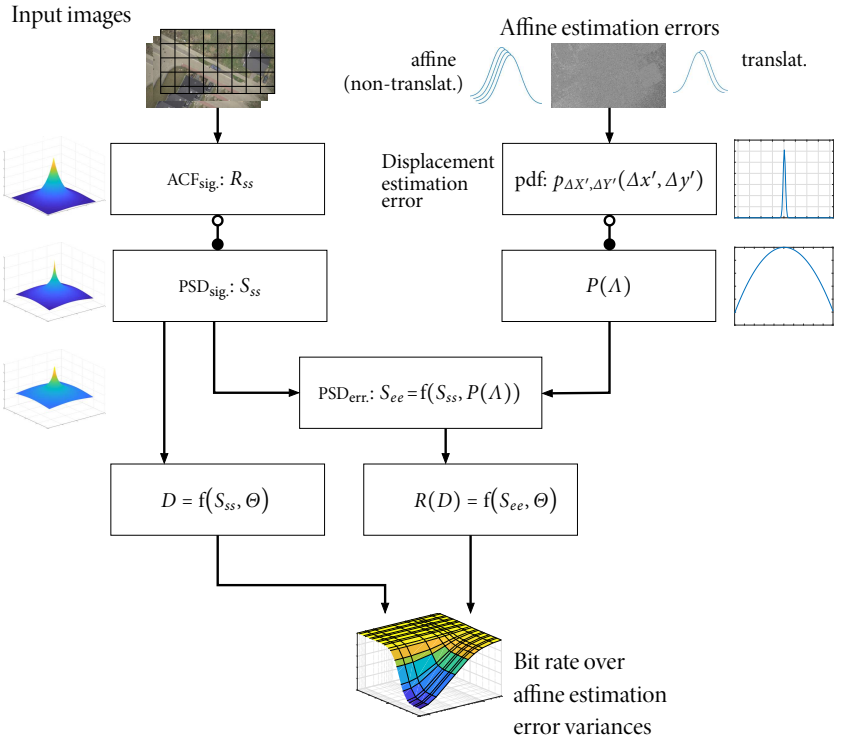


Figure 3.1: Flowchart of the analysis.

3. In a third step, the autocorrelation function (ACF) $R_{ss}(\Delta x', \Delta y')$ is modeled for typical input video sequences. According to the Wiener-Khinchin theorem, the power spectral density (PSD) of the signal $S_{ss}(\Lambda)$ is the Fourier transform of this autocorrelation function $R_{ss}(\Delta x', \Delta y')$ (left part in Fig. 3.1, Section 3.1.3).

4. Combining the PSD of the signal $S_{ss}(\Lambda)$ and the Fourier transform of the probability density function of the displacement estimation error $P(\Lambda)$ by exploiting the findings from Girod [36], the PSD of the prediction error $S_{ee}(\Lambda)$ is derived (middle in Fig. 3.1, Section 3.1.4).

5. In the last step, the rate-distortion theory is applied to derive a distortion D and the corresponding bit rate $R(D)$ of the prediction error signal (lower part in Fig. 3.1, Sections 3.1.5 and for the simplified affine model 3.2.2).

3.1.1 Affine motion and error model

Assuming a fully affine motion model with six degrees of freedom, the x- and y-coordinates x' and y' in the source frame can be computed from the affine parameter matrix

$$\mathtt{A}_f = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \tag{3.1}$$

and the homogeneous coordinate $(x,y,1)^\top$ in the current (destination) frame in component notation by backwards prediction:

$$x' = a_{11}\cdot x + a_{12}\cdot y + a_{13}\,; \quad y' = a_{21}\cdot x + a_{22}\cdot y + a_{23}\,. \tag{3.2}$$

The parameters a_{13} and a_{23} describe the translational part of a motion, whereas the parameters a_{11}, a_{12}, a_{21}, a_{22} express the rotation, scaling and shearing, respectively. The latter four parameters are further referred to as (purely) "affine parameters". It is assumed that each parameter a_{ij} with $i=\{1,2\}$, $j=\{1,2,3\}$ is perturbed by an independent error term e_{ij}, caused by inaccurate parameter estimation. Consequently, the perturbed coordinates $\hat{x}'$ and $\hat{y}'$ can be expressed as $\hat{x}' = \hat{a}_{11}x + \hat{a}_{12}y + \hat{a}_{13}$ and $\hat{y}' = \hat{a}_{21}x + \hat{a}_{22}y + \hat{a}_{23}$, leading to displacement estimation errors $\Delta x'$ and $\Delta y'$ (in

pel) in horizontal and vertical direction of:

$$\begin{aligned}\Delta x' = \hat{x}' - x' &= \underbrace{(\hat{a}_{11} - a_{11})}_{e_{11}} \cdot x + \underbrace{(\hat{a}_{12} - a_{12})}_{e_{12}} \cdot y + \underbrace{(\hat{a}_{13} - a_{13})}_{e_{13}} \\ &= e_{11} \cdot x + e_{12} \cdot y + e_{13}\,, \qquad (3.3)\\ \Delta y' &= e_{21} \cdot x + e_{22} \cdot y + e_{23}\,. \qquad (3.4)\end{aligned}$$

3.1.2 Probability density function of the displacement estimation error

With the assumption that each error term e_{ij} is zero-mean Gaussian distributed, the probability density functions $p(e_{ij})$ of the error terms e_{ij} are

$$p(e_{ij}) = \frac{1}{\sqrt{2\pi\sigma_{e_{ij}}^2}} \cdot \exp\left(-\frac{e_{ij}^2}{2\sigma_{e_{ij}}^2}\right), \qquad (3.5)$$

with $i = \{1,2\}$, $j = \{1,2,3\}$ and the variances $\sigma_{e_{ij}}^2$ of the error terms.

For statistically independent variables the joint pdf $p_{E_{11},\ldots,E_{23}}(e_{11},\ldots,e_{23})$ for the random variables $E_{11},\ldots,E_{23}$ generating the observations $e_{11},\ldots,e_{23}$ is:

$$p_{E_{11},\ldots,E_{23}}(e_{11},\ldots,e_{23}) = p(e_{11}) \cdot \ldots \cdot p(e_{23})\,. \qquad (3.6)$$

To convert the pdf $p_{E_{11},\ldots,E_{23}}(e_{11},\ldots,e_{23})$ to the desired pdf $p_{\Delta X',\Delta Y'}(\Delta x',\Delta y')$ with the random processes $\Delta X'$, $\Delta Y'$ generating the resulting displacement estimation errors $\Delta x'$ and $\Delta y'$ as caused by affine parameter estimation errors, the transformation theorem for pdfs is used ([99, 93]):

$$\begin{aligned}p_{\mathcal{Y}_1,\ldots,\mathcal{Y}_M}(y_1,\ldots,y_M) &= \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} p_{\mathcal{X}_1,\ldots,\mathcal{X}_N}(\xi_1,\ldots,\xi_N)\\ &\cdot \prod_{m=1}^{M} \delta\big(y_m - g_m(\xi_1,\ldots,\xi_N)\big)\mathrm{d}\xi_1\ldots\mathrm{d}\xi_N\,, \qquad (3.7)\end{aligned}$$

with $\delta(\cdot)$ denoting the Dirac delta function, $g_1,\ldots,g_M$ being functions $y_1 = g_1(x_1,\ldots,x_N),\ldots,y_M = g_M(x_1,\ldots,x_N)$, $\mathcal{X}_1,\ldots,\mathcal{X}_N$ and $\mathcal{Y}_1,\ldots,\mathcal{Y}_M$ representing random processes and $p_{\mathcal{Y}_1,\ldots,\mathcal{Y}_M}(y_1,\ldots,y_M)$ being the joint pdf. With (3.3) and

(3.4) this yields

$$\begin{aligned} p_{\Delta X',\Delta Y'}(\Delta x', \Delta y'|x, y) = &\int_{\mathbb{R}^6} p_{E_{11},\ldots,E_{23}}(e_{11}, \ldots, e_{23}) \\ &\cdot \delta\big(\Delta x' - (xe_{11} + ye_{12} + e_{13})\big) \\ &\cdot \delta\big(\Delta y' - (xe_{21} + ye_{22} + e_{23})\big)\, \mathrm{d}e_{11} \ldots \mathrm{d}e_{23}\,, \end{aligned} \tag{3.8}$$

with a dependency on the location coordinates x and y in the current frame. By using the properties of the delta function and substituting e_{13} and e_{23}, the integrals

$$\begin{aligned} &p_{\Delta X',\Delta Y'}(\Delta x', \Delta y'|x, y) \\ &= \int_{\mathbb{R}^4} p_{E_{11},\ldots,E_{22}}(e_{11}, e_{12}, \Delta x' - xe_{11} - ye_{12}, e_{21}, e_{22}\,, \\ &\quad \Delta y' - xe_{21} - ye_{22})\, \mathrm{d}e_{11}\mathrm{d}e_{12}\mathrm{d}e_{21}\mathrm{d}e_{22} \end{aligned} \tag{3.9}$$

are solved. Exploiting the statistical independence from (3.6), the integrands are separated, which leads to

$$\begin{aligned} &p_{\Delta X',\Delta Y'}(\Delta x', \Delta y'|x, y) \\ &= \int_{\mathbb{R}^2} p_{E_{11},E_{12},E_{13}}(e_{11}, e_{12}, \Delta x' - xe_{11} - ye_{12})\, \mathrm{d}e_{11}\mathrm{d}e_{12} \\ &\quad \cdot \int_{\mathbb{R}^2} p_{E_{21},E_{22},E_{23}}(e_{21}, e_{22}, \Delta y' - xe_{21} - ye_{22})\, \mathrm{d}e_{21}\mathrm{d}e_{22}\,. \end{aligned} \tag{3.10}$$

For simplicity, (3.10) is separated into its x- and y-components and the following derivation is presented for the x-component only. The y-component can be calculated accordingly. From (3.10) with (3.5) the pdf of $\Delta x'$ is determined:

$$
\begin{aligned}
p_{\Delta X'}(\Delta x'|x,y) &= \int_{\mathbb{R}^2} p_{E_{11},E_{12},E_{13}}(e_{11},e_{12},\Delta x' - xe_{11} - ye_{12})\,\mathrm{d}e_{11}\mathrm{d}e_{12} \\
&= \underbrace{\frac{1}{\sqrt{2\pi\sigma_{e_{11}}^2}} \cdot \frac{1}{\sqrt{2\pi\sigma_{e_{12}}^2}} \cdot \frac{1}{\sqrt{2\pi\sigma_{e_{13}}^2}}}_{A} \\
&\quad \cdot \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\left(-\frac{e_{11}^2}{2\sigma_{e_{11}}^2}\right) \cdot \exp\left(-\frac{e_{12}^2}{2\sigma_{e_{12}}^2}\right) \\
&\quad \cdot \exp\left(-\frac{(\Delta x' - xe_{11} - ye_{12})^2}{2\sigma_{e_{13}}^2}\right) \mathrm{d}e_{11}\mathrm{d}e_{12} \\
&= A \cdot \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\Bigg(-\frac{1}{2\sigma_{e_{11}}^2\sigma_{e_{12}}^2\sigma_{e_{13}}^2} \\
&\quad \cdot \Big[\sigma_{e_{12}}^2\sigma_{e_{13}}^2 e_{11}^2 + \sigma_{e_{11}}^2\sigma_{e_{13}}^2 e_{12}^2 \\
&\quad + \sigma_{e_{11}}^2\sigma_{e_{12}}^2 (\Delta x' - xe_{11} - ye_{12})^2\Big]\Bigg) \mathrm{d}e_{11}\mathrm{d}e_{12}\,. \qquad (3.11)
\end{aligned}
$$

Integration results in

$$
\begin{aligned}
p_{\Delta X'}(\Delta x'|x,y) &= \frac{1}{\sqrt{2\pi\left(\sigma_{e_{11}}^2 x^2 + \sigma_{e_{12}}^2 y^2 + \sigma_{e_{13}}^2\right)}} \\
&\quad \cdot \exp\left(-\frac{\Delta x'^2}{2\cdot\left(\sigma_{e_{11}}^2 x^2 + \sigma_{e_{12}}^2 y^2 + \sigma_{e_{13}}^2\right)}\right). \qquad (3.12)
\end{aligned}
$$

The step-by-step integration can be found in Appendix A.1.

After calculating the y-component accordingly, the resulting displacement estimation error pdf is obtained as

$$p_{\Delta X',\Delta Y'}(\Delta x',\Delta y'|x,y) = \frac{1}{2\pi\sigma_{\Delta x'}\sigma_{\Delta y'}}\cdot\exp\left(-\frac{\Delta x'^2}{2\sigma^2_{\Delta x'}}\right)\cdot\exp\left(-\frac{\Delta y'^2}{2\sigma^2_{\Delta y'}}\right) \tag{3.13}$$

$$\text{with } \sigma^2_{\Delta x'} = \sigma^2_{e_{11}}x^2 + \sigma^2_{e_{12}}y^2 + \sigma^2_{e_{13}} \tag{3.14}$$

$$\text{and } \sigma^2_{\Delta y'} = \sigma^2_{e_{21}}x^2 + \sigma^2_{e_{22}}y^2 + \sigma^2_{e_{23}}\,. \tag{3.15}$$

It is obvious that the variances $\sigma^2_{\Delta x'}$ and $\sigma^2_{\Delta y'}$ depend on the location in the frame. For simplicity $p_{\Delta X',\Delta Y'}(\Delta x',\Delta y'|x,y)$ is abbreviated as $p_{\Delta X',\Delta Y'}(\Delta x',\Delta y')$ further on and accordingly.

3.1.3 Power spectral density of the signal

The power spectral density $S_{ss}(\omega_x,\omega_y)$ is modeled according to O'Neal and Girod [95, 36]. There it was assumed that the statistics of each frame of the video sequence can be represented by the isotropic autocorrelation function

$$\begin{aligned} R_{ss,\text{iso}}(\Delta x',\Delta y') &= E\left[s(x',y')\cdot s(x'-\Delta x',y'-\Delta y')\right] \\ &= \exp\left(-\alpha\sqrt{\Delta x'^2+\Delta y'^2}\right) \end{aligned} \tag{3.16}$$

with $\Delta x'$ and $\Delta y'$ denoting the shift in x- and y-direction, respectively. Based on measurements in this work the autocorrelation function is assumed to be non-isotropic, leading to the general form

$$R_{ss}(\Delta x',\Delta y') = \exp\left(-\sqrt{\alpha_x^2\Delta x'^2+\alpha_y^2\Delta y'^2}\right). \tag{3.17}$$

The exponential drop rates α_x and α_y in x- and y-direction can be determined as the negative logarithm of the correlations between horizontally and vertically adjacent pels $\alpha_x = -\ln(\rho_{ss,x})$ and $\alpha_y = -\ln(\rho_{ss,y})$ [95]. For this, the autocorrelation coefficients [98, 93] $\rho_{ss,x}$ and $\rho_{ss,y}$ are calculated line- and column-wise, respectively. The power spectral density $S_{ss}(\Lambda)$ now is the Fourier transform of (3.17) (Wiener-Khinchin theorem).

3.1.4 Power spectral density of the displacement estimation error

To derive the bit rate for coding the prediction error in motion-compensated video coding, the findings from Girod are used [36]. He related the displacement estimation error pdf $p_{\Delta X', \Delta Y'}(\Delta x', \Delta y')$ to the prediction error e as follows: given a displacement estimation error pdf $p_{\Delta X', \Delta Y'}(\Delta x', \Delta y')$, the power spectral density of the prediction error

$$S_{ee}(\Lambda) = 2\, S_{ss}(\Lambda) \left[1 - \mathrm{Re}\{P(\Lambda)\}\right] + \Theta \tag{3.18}$$

is obtained, where $S_{ss}(\Lambda)$ denotes the power spectral density of the video signal s, Λ the two-dimensional (2D) spatial frequency vector $\Lambda := (\omega_x, \omega_y)$, $P(\Lambda)$ the 2D Fourier transform of the probability density function of the displacement estimation error (Appendix A.3), $\mathrm{Re}\{P(\Lambda)\}$ the real part of $P(\Lambda)$, and Θ a parameter that generates the rate-distortion function $R(D)$ (see next subsection) by taking on all positive real values ([36], Equation (28)). By variation of Θ the distortion and the corresponding rate for encoding the prediction error are determined, whereby one specific Θ yields one distinct distortion and a corresponding rate.

3.1.5 Rate-distortion function

Applying the rate-distortion theory [7] finally results in the minimum required bit rate for encoding the prediction error. The distortion D as well as the corresponding minimum bit rate $R(D)$ are derived from the rate-distortion function for a given mean-squared error (Equations (19), (20) in [36], and [7]):

$$D = \frac{1}{4\pi^2} \iint_{\Lambda} \min\left[\Theta, S_{ss}(\Lambda)\right] \mathrm{d}\Lambda \,, \tag{3.19}$$

$$R(D) = \frac{1}{8\pi^2} \iint_{\substack{\Lambda:\, \left(S_{ss}(\Lambda) > \Theta \\ \text{and } S_{ee}(\Lambda) > \Theta\right)}} \log_2 \left[\frac{S_{ee}(\Lambda)}{\Theta}\right] \mathrm{d}\Lambda \text{ bit} \,. \tag{3.20}$$

It is noteworthy that in contrast to the derivations from Girod for a purely translational motion model $\sigma^2_{\Delta x'}$ and $\sigma^2_{\Delta y'}$ are location-dependent for an affine

motion model, since they are functions of the coordinates x and y. Consequently, $p_{\Delta X', \Delta Y'}(\Delta x', \Delta y')$, $P(\Lambda)$ and $S_{ee}(\Lambda)$, and finally $R(D)$ are also location-dependent.

Using the idea of generating the rate-distortion function for translative motion like explained by Girod [36] and the results from Sections 3.1.1–3.1.4, the rate-distortion function for affine motion can be evaluated, which is done at the end of the next subsection (Section 3.1.6.3).

3.1.6 Rate-distortion analysis of affine global motion-compensated prediction

In this subsection, the minimum bit rate R (Equation (3.20)) for encoding the prediction error as a function of the estimation parameter variances $\sigma^2_{e_{11}}$, $\sigma^2_{e_{12}}$, $\sigma^2_{e_{13}}$, $\sigma^2_{e_{21}}$, $\sigma^2_{e_{22}}$, $\sigma^2_{e_{23}}$ is evaluated using a fully affine motion model with 6 degrees of freedom. Without loss of generality, the computations in this subsection are carried out for *global* motion-compensated prediction, which is justified by the fact that in aerial videos from small and medium UAVs the camera-motion induced global motion is the predominant motion in each frame.

Computations for block-based motion-compensated prediction are additionally carried out in Section 3.2. There, a simplified affine motion model is assumed as it is currently explored in the course of the standardization of the upcoming next video coding standard VVC by JVET.

Due to the findings of (3.14) and (3.15), the variances of the displacement estimation error $\sigma^2_{\Delta x'}$ and $\sigma^2_{\Delta y'}$ depend on the location in the frame. Consequently, also the resulting minimum achievable bit rate is location-dependent. To obtain the total bit rate for encoding one frame, the bit rate is calculated for each pel over the entire frame and subsequently summed up. Also according to (3.14) and (3.15), the variances of the displacement estimation errors $\sigma^2_{\Delta x'}$ and $\sigma^2_{\Delta y'}$ additionally depend on the variances of the error terms $\sigma^2_{e_{11}}$, $\sigma^2_{e_{12}}$, $\sigma^2_{e_{13}}$ for $\sigma^2_{\Delta x'}$ and on $\sigma^2_{e_{21}}$, $\sigma^2_{e_{22}}$, $\sigma^2_{e_{23}}$ for $\sigma^2_{\Delta y'}$, respectively.

3.1.6.1 Displacement estimation error variances, scene "affinity" and motion model error

To receive viable values for the minimum bit rate R for encoding the prediction error, realistic variances $\sigma^2_{e_{11}}, \ldots, \sigma^2_{e_{23}}$ are determined (Equations (3.13)–(3.20)). Therefore, the affine estimation error variances of the affine motion estimation implementation [10] are measured. A video sequence in full HD resolution of 1920×1080 pel was

Table 3.1: Measured estimation error variances $\sigma^2_{e_{ij}}$ in the artificial aerial video sequence generated from the *Hannover* [63] aerial image as provided by the applied affine motion estimation implementation [10]. The values represent the accuracy limits of the implementation.

$\sigma^2_{e_{11}}$	$\sigma^2_{e_{12}}$	$\sigma^2_{e_{13}}$	$\sigma^2_{e_{21}}$	$\sigma^2_{e_{22}}$	$\sigma^2_{e_{23}}$
$3.27{\cdot}10^{-10}$	$6.73{\cdot}10^{-10}$	$3.06{\cdot}10^{-5}$	$6.61{\cdot}10^{-10}$	$3.19{\cdot}10^{-10}$	$2.83{\cdot}10^{-5}$

extracted from the aerial image *Hannover* [63] with a resolution of 10000×10000 pel (see examples in Fig. 5.1 on page 89 in the experimental chapter). The signal characteristic and the ground resolution of $0.2\,\mathrm{m/pel} \times 0.2\,\mathrm{m/pel}$ of the sequence represent realistic conditions for aerial surveillance missions. Each frame of the video sequence was generated by affine transformation (Equation (3.2)) of the still image *Hannover* whereas each affine parameter follows a Gaussian distribution with given means and variances, denoted as $\mathcal{N}(\text{mean; variance})$, of:

$$\begin{aligned} A_{11} &\sim \mathcal{N}(1;\, 10^{-5})\,; & A_{12} &\sim \mathcal{N}(0;\, 10^{-5})\,; & A_{13} &\sim \mathcal{N}(15;\, 100)\,; \\ A_{21} &\sim \mathcal{N}(0;\, 10^{-5})\,; & A_{22} &\sim \mathcal{N}(1;\, 10^{-5})\,; & A_{23} &\sim \mathcal{N}(\ 0;\ \ 10)\,. \end{aligned} \tag{3.21}$$

$A_{11}, \ldots, A_{23}$ represent the random processes generating $a_{11}, \ldots, a_{23}$. A Lanczos filter [29] was applied as interpolation filter. The introduced motion covers typical motion types like rotation and shearing. This sequence was used as ground truth. The variances of the estimation parameter errors of the generated artificial video sequence are presented in Table 3.1. These values represent the accuracy of the motion estimation implementation [10].

To analyze the overall benefit of the application of affine global motion-compensated prediction in video coding, the affine global motion parts, the "affinities", can be determined. Here, "affinity" means the inherent purely affine parts of the motion contained in a sequence which cannot be described in principle by a translational motion model.

If a translational motion model is used for a sequence containing a distinct affinity, the motion model error can be expressed as displacement estimation errors $\Delta x'_{\mathrm{mod}}$

and $\Delta y'_{\mathrm{mod}}$ in x- and y-direction as

$$\Delta x'_{\mathrm{mod}} = x'_{\mathrm{trans}} - x'_{\mathrm{aff}} \,, \tag{3.22}$$

$$\Delta y'_{\mathrm{mod}} = y'_{\mathrm{trans}} - y'_{\mathrm{aff}} \,. \tag{3.23}$$

In these two equations, x'_{trans}, y'_{trans} are the estimated displacements and x'_{aff}, y'_{aff} are the real displacements in the sequence caused by a fully affine motion inherently contained in the scene. With a fully affine motion according to (3.2) (page 40) and a purely translational motion model

$$x' = x + a_{13} \,; \qquad y' = y + a_{23} \tag{3.24}$$

(3.22) and (3.23) yield

$$\begin{aligned} \Delta x'_{\mathrm{mod}} &= \underbrace{(1 - a_{11})}_{e_{11,\mathrm{mod}}} \cdot x \qquad - \underbrace{(a_{12})}_{e_{12,\mathrm{mod}}} \cdot y \\ &= e_{11,\mathrm{mod}} \cdot x \qquad + e_{12,\mathrm{mod}} \cdot y \,, \end{aligned} \tag{3.25}$$

$$\Delta y'_{\mathrm{mod}} = e_{21,\mathrm{mod}} \cdot x \qquad + e_{22,\mathrm{mod}} \cdot y \,. \tag{3.26}$$

The parameters $a_{11}, \ldots, a_{23}$ in (3.24)–(3.26) are assumed to be perfectly estimated for the calculation of the motion model error, since estimation errors have already been considered separately (Table 3.1). This means that the purely affine motion model errors $e_{11,\mathrm{mod}}$, $e_{12,\mathrm{mod}}$, $e_{21,\mathrm{mod}}$, $e_{22,\mathrm{mod}}$ are solely caused by motion contained in the scene which cannot be covered by a translational motion model.

The Equations (3.25) and (3.26) have the same structure as (3.3) and (3.4). Consequently, (3.13)–(3.15) also describe the motion model error if the variances of the motion model errors $\sigma^2_{e_{11,\mathrm{mod}}}$, $\sigma^2_{e_{12,\mathrm{mod}}}$, $\sigma^2_{e_{21,\mathrm{mod}}}$, $\sigma^2_{e_{22,\mathrm{mod}}}$ are inserted in (3.14)–(3.15) instead of the estimation error variances $\sigma^2_{e_{11}}$, $\sigma^2_{e_{12}}$, $\sigma^2_{e_{21}}$, $\sigma^2_{e_{22}}$. Purely translational model errors $e_{13,\mathrm{mod}}$ and $e_{23,\mathrm{mod}}$, or e_{13} and e_{23} in (3.14)–(3.15), respectively, are non-existent and thus set to zero.

As shown above, in case of a translational motion model, the entire "affinity" of a sequence can be considered as estimation error, since it cannot be covered by the motion model.

The affinities of four representative camera-captured aerial sequences from the TAVT data set (set 1) [46, 81] were measured. Hereby, the purely affine motion types (rotation, shearing, scaling) were assumed to be zero between two consecutive frames in a video sequence recorded at 30 fps and with a prevalent straight forward motion

Table 3.2: Measured variances $\sigma^2_{e_{ij}}$ of non-translational affine transformation parameters ("affinity") of aerial videos from the TAVT data set (set 1) [46, 81]. The sequence (seq.) names refer to the flight altitudes they were recorded at.

Seq. name	$\sigma^2_{e_{11}}$	$\sigma^2_{e_{12}}$	$\sigma^2_{e_{21}}$	$\sigma^2_{e_{22}}$	Mean $(\sigma^2_{e_{11}}, \sigma^2_{e_{22}})$	Mean $(\sigma^2_{e_{12}}, \sigma^2_{e_{21}})$
350 m seq.	$2.03 \cdot 10^{-7}$	$6.03 \cdot 10^{-7}$	$6.59 \cdot 10^{-7}$	$2.24 \cdot 10^{-7}$	$\mathbf{2.13 \cdot 10^{-7}}$	$\mathbf{6.31 \cdot 10^{-7}}$
500 m seq.	$1.94 \cdot 10^{-7}$	$5.09 \cdot 10^{-7}$	$3.63 \cdot 10^{-7}$	$1.94 \cdot 10^{-7}$	$\mathbf{1.94 \cdot 10^{-7}}$	$\mathbf{4.35 \cdot 10^{-7}}$
1000 m seq.	$1.74 \cdot 10^{-7}$	$4.05 \cdot 10^{-7}$	$4.13 \cdot 10^{-7}$	$2.12 \cdot 10^{-7}$	$\mathbf{1.93 \cdot 10^{-7}}$	$\mathbf{4.09 \cdot 10^{-7}}$
1500 m seq.	$3.19 \cdot 10^{-7}$	$3.80 \cdot 10^{-7}$	$3.69 \cdot 10^{-7}$	$3.46 \cdot 10^{-7}$	$\mathbf{3.33 \cdot 10^{-7}}$	$\mathbf{3.75 \cdot 10^{-7}}$
Mean	$\mathbf{2.23 \cdot 10^{-7}}$	$\mathbf{4.74 \cdot 10^{-7}}$	$\mathbf{4.51 \cdot 10^{-7}}$	$\mathbf{2.44 \cdot 10^{-7}}$	$\mathbf{2.33 \cdot 10^{-7}}$	$\mathbf{4.63 \cdot 10^{-7}}$

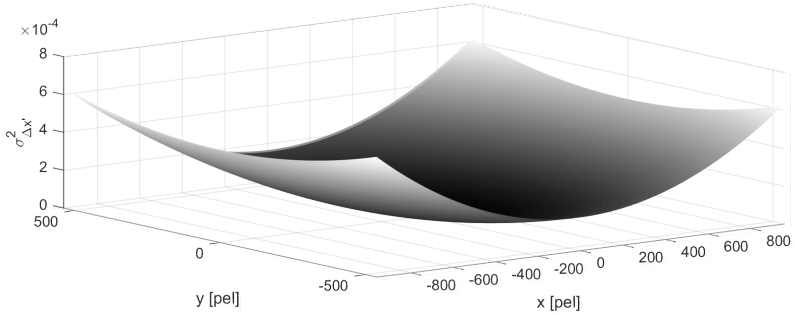
of the camera. This results in the affinities of the TAVT data set sequences as shown in Table 3.2.

From the measured results in Table 3.2 it is obvious that the variances $\sigma^2_{e_{11}}$ and $\sigma^2_{e_{22}}$ as well as $\sigma^2_{e_{12}}$ and $\sigma^2_{e_{21}}$ are pairwise similar. This can be explained by the fact that the affine motion parts are predominantly caused by a physical rotation of the camera and the skew-symmetry of a 2D rotation matrix. Justified by these findings, it is assumed that $\sigma^2_{e_{11}} = \sigma^2_{e_{22}}$ as well as $\sigma^2_{e_{12}} = \sigma^2_{e_{21}}$ and the averaged values $2.33 \cdot 10^{-7}$ and $4.63 \cdot 10^{-7}$ (see Table 3.2), respectively, are used for further computations.

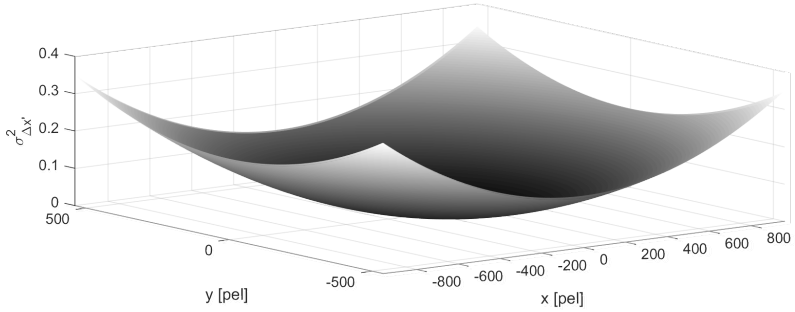
The location-dependent variance $\sigma^2_{\Delta x'}$ is shown in Fig. 3.2 for a full HD resolution image. In Fig. 3.2a the affine estimation error variances which can be provided by maximum accurate estimation (Table 3.1) are applied, and in Fig. 3.2b the inherent affinities contained in real camera-captured sequences as measured in the TAVT video sequences (Table 3.2), are used. It had to be expected that the variances of the model error in the range of 10^{-7} exceed the estimation error variances (approximately $5 \cdot 10^{-10}$) by several orders of magnitude. This is caused by the fact that any non-translational motion like rotation of the UAV causes a global rotation in the frame (for a camera in nadir-view) which cannot be covered by a translational motion model. Although the TAVT sequences contain prevalently straightforward motion, small rotations are also included. As a consequence also the variances of the displacement estimation errors vary by three orders of magnitude.

3.1.6.2 Power spectral density of the video signal

For the calculation of the power spectral density S_{ss} of the video signal, the exponential drop rates α_x and α_y of the autocorrelation function are required (Section 3.1.3,



(a) $\sigma^2_{\Delta x',\text{affine}}$ for $\sigma^2_{e_{11}} = \sigma^2_{e_{12}} = 5 \cdot 10^{-10}$ and $\sigma^2_{e_{13}} = 3 \cdot 10^{-5}$



(b) $\sigma^2_{\Delta x',\text{affine}}$ for $\sigma^2_{e_{11}} = 2.3 \cdot 10^{-7}$, $\sigma^2_{e_{12}} = 4.6 \cdot 10^{-7}$ and $\sigma^2_{e_{13}} = 0.0052$ ($\hat{=}$ $\frac{1}{4}$-pel translational accuracy)

Figure 3.2: Location-dependent variances $\sigma^2_{\Delta x'}$ assuming Gaussian distributed displacement estimation error pdfs for a frame in full HD resolution (a) for affine estimation error variances which can be provided by maximum accurate estimation as measured for the artificially generated video sequence and (b) for the averaged scene affinities from the real-world video sequences from the TAVT data set (set 1) [46, 81] and $\sigma^2_{e_{13}} = 0.0052$ (equals $\frac{1}{4}$-pel translational motion estimation accuracy).

Equation (3.16)). Thus, the mean correlations [98] of horizontally and vertically adjacent pels of several video sequences from the JCT-VC test set [66] were calculated. Results are presented in Table 3.3. It can be observed that the correlations between adjacent pels are larger for higher resolution sequences (HD) compared to lower resolution sequences as those used by Girod. This result had to be expected, since the video characteristics have not fundamentally changed and comparable focal lengths were used for capturing. Thus, much more pels represent one object in a HD

Table 3.3: Measured horizontal and vertical correlations [98] between adjacent pels for typical test sequences (*: 100 frames each, HD refers to a resolution of 1920×1080, SD refers to a resolution of 720×576).

Sequence	Corr. $\rho_{ss,x}$	Corr. $\rho_{ss,y}$
Values from Girod [36]	0.928	0.934
OldTownCross* (SD)	0.9780	0.9407
CrowdRun* (SD)	0.9257	0.9378
ParkJoy* (SD)	0.8731	0.9084
DucksTakeOff* (SD)	0.9563	0.8739
InToTree* (SD)	0.9792	0.9722
BasketballDrive* (HD) [13]	0.9782	0.9488
BQTerrace* (HD) [13]	0.9680	0.9659
Cactus* (HD) [13]	0.9741	0.9812
Kimono* (HD) [13]	0.9883	0.9900
ParkScene* (HD) [13]	0.9634	0.9518
Mean of SD sequences	0.9425	0.9266
Mean of HD sequences	**0.9744**	**0.9677**

sequence than in a low resolution sequence (e. g. QCIF, CIF, or SD) and consequently, the correlations between pels have to be higher for HD sequences.

3.1.6.3 Application of the rate-distortion theory

The evaluation of the rate-distortion theory (Equations (3.19) and (3.20) in Section 3.1.5) yields the minimum required bit rate R for a distortion D.

The location-dependent bit rate is visualized in Fig. 3.3 for a HD resolution frame with purely affine (non-translational) estimation error variances of $\sigma^2_{e_{11}} = \sigma^2_{e_{12}} = \sigma^2_{e_{21}} = \sigma^2_{e_{22}} = 5 \cdot 10^{-10}$ (see Table 3.1), translational estimation error variances $\sigma^2_{e_{13}} = \sigma^2_{e_{23}} = 0$, and Θ selected to yield a signal-to-noise ratio (SNR) of 30 dB. In Fig. 3.4 the bit rate is plotted versus the translational variances on one axis ($\sigma^2_{e_{13}}$, $\sigma^2_{e_{23}}$) and the non-translational affine variances ($\sigma^2_{e_{11}}$, $\sigma^2_{e_{12}}$, $\sigma^2_{e_{21}}$, $\sigma^2_{e_{22}}$) on the other axis. For visualization both translational and all purely affine error variances are assumed to be equal. Isolines are marked by data tips in the 3D plot in Fig. 3.4a for a translational half-pel resolution (data tip for "transl. var.: 0.0208") as well as quarter-pel resolution (data tips with "transl. var.: 0.0052") and non-translational affine estimation error variances of $\sigma^2_{e_{11}} = \sigma^2_{e_{12}} = \sigma^2_{e_{21}} = \sigma^2_{e_{22}} = 5 \cdot 10^{-10}$ (see Table 3.1). In Fig. 3.4b, 2D cuts of the surface

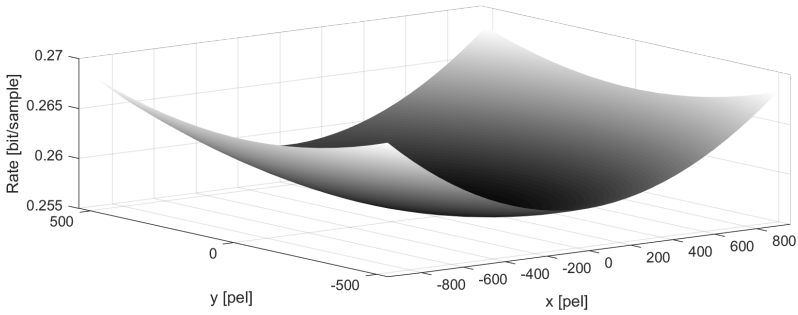


Figure 3.3: Location-dependent bit rate for Gaussian distributed displacement estimation error pdfs for a HD frame for purely affine estimation error variances of $\sigma^2_{e_{11}} = \sigma^2_{e_{12}} = \sigma^2_{e_{21}} = \sigma^2_{e_{22}} = 5 \cdot 10^{-10}$ and translational quarter-pel resolution ($\sigma^2_{e_{13}} = \sigma^2_{e_{23}} = 0.0052$).

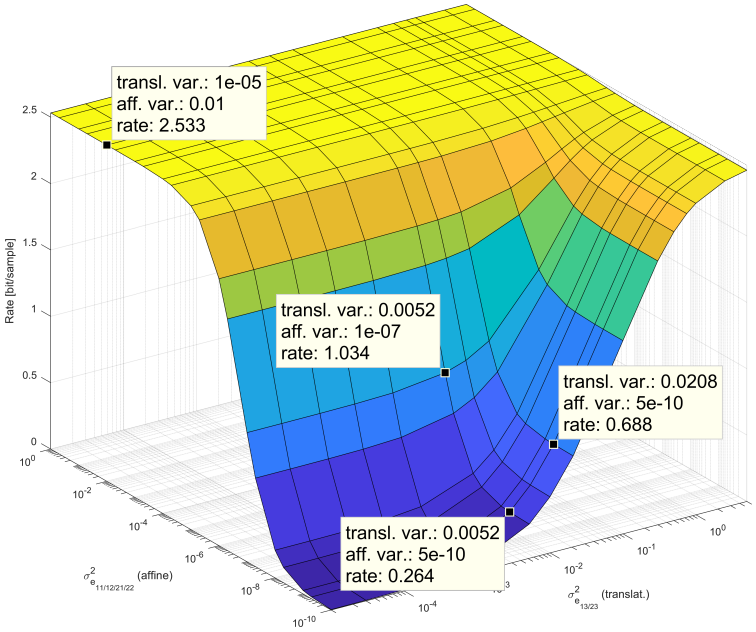
are plotted for HD resolution and different non-translational affine estimation error variances.

3.1.7 Conclusions for the fully affine motion model for global motion compensation

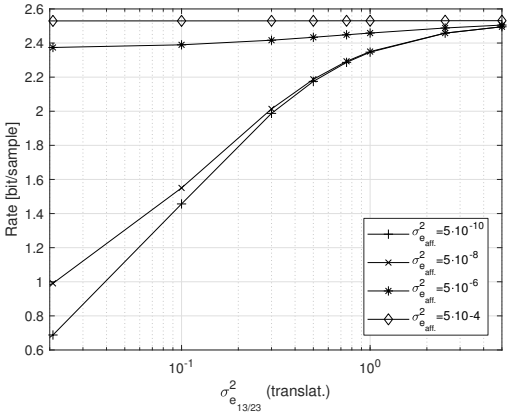
From the results it can be inferred:

1. The variances of the estimation errors of the purely affine parameters ($\sigma^2_{e_{11}}$, $\sigma^2_{e_{12}}$, $\sigma^2_{e_{21}}$, $\sigma^2_{e_{22}}$) have to be magnitudes smaller than the variances of the translational parameters ($\sigma^2_{e_{13}}$, $\sigma^2_{e_{23}}$) to yield reasonably small bit rates. For a potential quantization of the purely affine parameters for encoding purposes this fact should be taken into account. This result had to be expected, since the error variances as well as the bit rates are location-dependent which becomes important for non (purely) translational motion like rotation.

2. The isoline with all purely affine error variances equal to zero (not printed in the logarithmic plot in Fig. 3.4a) describes the bit rate for encoding the prediction error for a translational motion model (which is identical to the results from Girod [36] for same correlations). Purely affine variances unequal to zero obviously can only occur if an affine model is employed. In such a case, affine motions contained in a scene can be matched much better than with a purely



(a) Isolines for translational motion estimation error variances of $\sigma^2_{e_{13}} = \sigma^2_{e_{23}} = 0.0052$ and $\sigma^2_{e_{13}} = \sigma^2_{e_{23}} = 0.0208$ are marked by data tips



(b) 2D cuts of the surface shown in (a)

Figure 3.4: Minimum required bit rate versus variances $\sigma^2_{e_{ij}}$ for a distortion of SNR = 30 dB assuming $\sigma^2_{e_{11}} = \sigma^2_{e_{12}} = \sigma^2_{e_{21}} = \sigma^2_{e_{22}}$ and $\sigma^2_{e_{13}} = \sigma^2_{e_{23}}$ for HD resolution. (a) shows a 3D plot, in (b) corresponding 2D cuts are plotted.

translational motion model, i. e. the operating point moves towards the dark blue plateau in Fig. 3.4a. Using an affine motion model is especially beneficial in the case that high amounts of non-translational motions are contained in a scene.

3. For a sequence with a specific degree of purely affine motion ("affinity"), which cannot be described by a translational motion model, the minimum bit rate is limited along the (non-translational) affine-variances-axis (directing from the origin leftwards in Fig. 3.4a). As an example, a HD sequence with an "affinity" of 10^{-7} is assumed (see Table 3.2). The additional estimation error is negligible in this example since it is three orders of magnitude smaller (see Table 3.1) and consequently also the contribution of the estimation error to the bit rate is negligible. For the example above the minimum bit rate for encoding the prediction error using a purely translational motion estimation with the small estimation error variances of $\sigma^2_{e_{13}} = \sigma^2_{e_{23}} = 0.0052$ is 1.034 $^{\text{bit}}/_{\text{sample}}$ (upper left data tip in Fig. 3.4a). In contrast to that the minimum bit rate is only 0.26 $^{\text{bit}}/_{\text{sample}}$ for an accurate *affine* motion estimation with estimation error variances of $\sigma^2_{e_{11}} = \sigma^2_{e_{12}} = \sigma^2_{e_{21}} = \sigma^2_{e_{22}} = 5 \cdot 10^{-10}$ and the same translational accuracy of $^1/_4$-pel resolution (lower data tip in Fig. 3.4a).

4. From the example given in 3., it can be generalized that the minimum required bit rate is reached, if the motion model covers the real motion contained in the scene, *and* if the affine estimation is highly accurate. The feasibility of this requirement is shown in this work.

5. As it is obvious from (3.13)–(3.15) (and Fig. 3.2), $\sigma^2_{\Delta x'}$ and $\sigma^2_{\Delta y'}$ increase for large image dimensions. For block-based motion compensation, the "frame dimensions" are equal to the block dimensions. A block-based affine motion-compensated prediction is analyzed in the following Section 3.2.

3.2 Efficiency Analysis of Simplified Affine Motion Compensation

An efficiency analysis of a fully affine motion model has been presented in the previous section (Section 3.1). In contrast to that, a simplified affine motion model as investigated in JEM and in the course of the standardization of VVC is assumed here. Although "simplified" in the name suggests that also the theoretical analysis is

simplified, additional dependencies between the parameters of the model have to be considered.

However, the basic structure of the derivation remains the same and only the modeling of the probability density function ${}^{\text{simp}}p_{\Delta X'_\text{s},\Delta Y'_\text{s}}\left(\Delta x'_\text{s}, \Delta y'_\text{s} | x, y\right)$ is different.

3.2.1 Derivation of the probability density function of the displacement estimation error for a simplified affine model

A simplified affine model with four parameters like proposed by Li et al. [65] and used in the JEM software is assumed in this subsection. The effect of global motion parameter inaccuracies employing such a simplified model has been investigated by Dane and Nguyen [25]. In their work, they introduced probabilistic rotational, scale and translational errors and derived that doubling the accuracy of the motion parameter estimates enables a theoretical gain of up to 6 dB in the prediction error variance. However, they did not relate their results to other motion models (e. g. purely translational, fully affine), albeit the bit rate for encoding the prediction error highly depends on the applied motion model. Moreover, the bit rate for encoding the prediction error was not considered in their work.

With the rotation angle θ, the scaling factor s_s in both, horizontal and vertical direction, and the translational parameters c and f (which correspond to the parameters a_{13} and a_{23} in the fully affine model in Section 3.1.1), the relationship between the coordinates x and y before and x'_s and y'_s after the transformation is described in [65] as

$$\begin{aligned} x'_\text{s} &= s_\text{s}\cos\theta\cdot x + s_\text{s}\sin\theta\cdot y + c\,; \\ y'_\text{s} &= -s_\text{s}\sin\theta\cdot x + s_\text{s}\cos\theta\cdot y + f\,. \end{aligned} \tag{3.27}$$

Replacing

$$(s_\text{s}\cos\theta) \text{ by } (1+a) \quad \text{and} \quad (s_\text{s}\sin\theta) \text{ by } b\,, \tag{3.28}$$

respectively, (3.27) can be expressed as

$$\begin{aligned} x'_\text{s} &= (a+1)\cdot x + b\cdot y + c\,; \\ y'_\text{s} &= -b\cdot x + (a+1)\cdot y + f\,. \end{aligned} \tag{3.29}$$

Each parameter a, b, c, f is assumed to be perturbed by an independent error term e_i, with $i = \{a, b, c, f\}$, caused by inaccurate parameter estimation. The perturbed coordinates $\hat{x}_\text{s}$, $\hat{y}_\text{s}$ lead to displacement estimation errors in horizontal and vertical

direction of $\Delta x'_\mathrm{s}$ and $\Delta y'_\mathrm{s}$ (in pel)

$$\begin{aligned} \Delta x'_\mathrm{s} &= \hat{x}' - x' &&= e_a \cdot x &&+ e_b \cdot y &&+ e_c\,; \\ \Delta y'_\mathrm{s} &= \hat{y}' - y' &&= -e_b \cdot x &&+ e_a \cdot y &&+ e_f\,. \end{aligned} \tag{3.30}$$

Assuming each error term e_i to be zero-mean Gaussian distributed leads to the probability density functions (pdfs)

$$p(e_i) = \frac{1}{\sqrt{2\pi\sigma_{e_i}^2}} \cdot \exp\left(-\frac{e_i^2}{2\sigma_{e_i}^2}\right) \tag{3.31}$$

with $i = \{a,b,c,f\}$.

For statistically independent variables, the joint pdf $p_{E_a,E_b,E_c,E_f}(e_a,e_b,e_c,e_f)$ for the random processes E_a, E_b, E_c, E_f and the observations e_a, e_b, e_c, e_f is:

$$p_{E_a,E_b,E_c,E_f}(e_a,e_b,e_c,e_f) = p(e_a) \cdot p(e_b) \cdot p(e_c) \cdot p(e_f)\,. \tag{3.32}$$

In order to convert the pdf $p_{E_a,E_b,E_c,E_f}(e_a,e_b,e_c,e_f)$ to the desired pdf ${}^{\mathrm{simp}}p_{\Delta X'_\mathrm{s},\Delta Y'_\mathrm{s}}(\Delta x'_\mathrm{s},\Delta y'_\mathrm{s}|x,y)$, the transformation theorem for pdfs can be used again (Equation (3.7) on page 41). Here, $\Delta X'_\mathrm{s}$, $\Delta Y'_\mathrm{s}$ are the random processes generating the displacement estimation errors $\Delta x'_\mathrm{s}$, $\Delta y'_\mathrm{s}$ (in pel) caused by affine parameter estimation errors. With (3.30) this yields

$$\begin{aligned} {}^{\mathrm{simp}}p_{\Delta X'_\mathrm{s},\Delta Y'_\mathrm{s}}(\Delta x'_\mathrm{s},\Delta y'_\mathrm{s}|x,y) = \int_{\mathbb{R}^4} & p_{E_a,\ldots,E_f}(e_a,\ldots,e_f) \\ & \cdot \delta\big(\Delta x'_\mathrm{s} - (e_a x + e_b y + e_c)\big) \\ & \cdot \delta\big(\Delta y'_\mathrm{s} - (-e_b x + e_a y + e_f)\big)\,\mathrm{d}e_a\mathrm{d}e_b\mathrm{d}e_c\mathrm{d}e_f\,, \end{aligned} \tag{3.33}$$

with a dependency on the location coordinates x, y in the current frame. Using the properties of the delta function results in

$$\begin{aligned} &{}^{\mathrm{simp}}p_{\Delta X'_\mathrm{s},\Delta Y'_\mathrm{s}}(\Delta x'_\mathrm{s},\Delta y'_\mathrm{s}|x,y) \\ &\quad = \int_{\mathbb{R}^2} p_{E_a,E_b,E_c,E_f}(e_a,e_b,\Delta x'_\mathrm{s} - e_a x - e_b y, \\ &\qquad \Delta y'_\mathrm{s} + e_b x - e_a y)\,\mathrm{d}e_a\mathrm{d}e_b\,. \end{aligned} \tag{3.34}$$

Considering (3.34) and (3.31) results in:

$$
\begin{aligned}
{}^{\text{simp}}p_{\Delta X'_{\text{s}},\Delta Y'_{\text{s}}}\left(\Delta x'_{\text{s}},\Delta y'_{\text{s}}|x,y\right) &= \frac{1}{(2\pi)^2\sigma_{e_a}\sigma_{e_b}\sigma_{e_c}\sigma_{e_f}} \\
&\cdot \int_{\mathbb{R}^2} \exp\left(-\frac{e_a^2}{2\sigma_{e_a}^2} - \frac{e_b^2}{2\sigma_{e_b}^2} - \frac{\left(\Delta x'_{\text{s}} - e_a x - e_b y\right)^2}{2\sigma_{e_c}^2} \right. \\
&\left. - \frac{\left(\Delta y'_{\text{s}} + e_b x - e_a y\right)^2}{2\sigma_{e_f}^2} \right) \mathrm{d}e_a \mathrm{d}e_b \,. \qquad (3.35)
\end{aligned}
$$

After the two integrations

$$
{}^{\text{simp}}p_{\Delta X'_{\text{s}},\Delta Y'_{\text{s}}}\left(\Delta x'_{\text{s}},\Delta y'_{\text{s}}|x,y\right) = \frac{1}{2\pi\sqrt{N}} \cdot \exp\left(\frac{M}{2N}\right) \qquad (3.36)
$$

$$
\begin{aligned}
\text{with } N = &\left(\left(x^2+y^2\right)^2 \sigma_{e_b}^2 + y^2\sigma_{e_c}^2 + x^2\sigma_{e_f}^2 \right)\sigma_{e_a}^2 \\
&+ \left(x^2\sigma_{e_c}^2 + y^2\sigma_{e_f}^2 \right)\sigma_{e_b}^2 + \sigma_{e_c}^2\sigma_{e_f}^2 \qquad (3.37)
\end{aligned}
$$

$$
\begin{aligned}
\text{and } M = &- \left(x\Delta y'_{\text{s}} - y\Delta x'_{\text{s}}\right)^2 \sigma_{e_a}^2 - \left(x\Delta x'_{\text{s}} + y\Delta y'_{\text{s}}\right)^2 \sigma_{e_b}^2 \\
&- \Delta x'^2\sigma_{e_f}^2 - \Delta y'^2\sigma_{e_c}^2 \qquad (3.38)
\end{aligned}
$$

is obtained (for intermediate steps see Appendix A.2).

Transforming (3.36) into the form of a common bivariate zero-mean normal distribution with ρ being the correlation coefficient between $\Delta X'$ and $\Delta Y'$ leads to the desired final pdf of the displacement estimation error:

$$
\begin{aligned}
{}^{\text{simp}}p_{\Delta X'_{\text{s}},\Delta Y'_{\text{s}}}\left(\Delta x'_{\text{s}},\Delta y'_{\text{s}}|x,y\right) &= \frac{1}{2\pi\,\sigma_{\Delta x'_{\text{s}}}\,\sigma_{\Delta y'_{\text{s}}}\sqrt{1-\rho^2}} \\
&\cdot \exp\left(-\frac{1}{2\left(1-\rho^2\right)} \left[\frac{\Delta x'^2_{\text{s}}}{\sigma^2_{\Delta x'_{\text{s}}}} + \frac{\Delta y'^2_{\text{s}}}{\sigma^2_{\Delta y'_{\text{s}}}} - \frac{2\rho\cdot\Delta x'_{\text{s}}\cdot\Delta y'_{\text{s}}}{\sigma_{\Delta x'_{\text{s}}}\cdot\sigma_{\Delta y'_{\text{s}}}} \right] \right) \qquad (3.39)
\end{aligned}
$$

$$\text{with } \sigma^2_{\Delta x'_s} = N \cdot \left(\left(\sigma^2_{e_a} y^2 + \sigma^2_{e_b} x^2 + \sigma^2_{e_f} \right) \cdot \left(1 - \rho^2 \right) \right)^{-1}, \tag{3.40}$$

$$\sigma^2_{\Delta y'_s} = N \cdot \left(\left(\sigma^2_{e_a} x^2 + \sigma^2_{e_b} y^2 + \sigma^2_{e_c} \right) \cdot \left(1 - \rho^2 \right) \right)^{-1}, \tag{3.41}$$

$$\rho = \frac{\left(\sigma^2_{e_a} xy - \sigma^2_{e_b} xy \right)}{\sqrt{\sigma^2_{e_a} y^2 + \sigma^2_{e_b} x^2 + \sigma^2_{e_f}} \sqrt{\sigma^2_{e_a} x^2 + \sigma^2_{e_b} y^2 + \sigma^2_{e_c}}}. \tag{3.42}$$

Obviously, the variances $\sigma^2_{\Delta x'_s}$ and $\sigma^2_{\Delta y'_s}$ depend on the locations x, y in the frame similarly to the fully affine model (Section 3.1.1). Further on, ${}^{\text{simp}}p_{\Delta X'_s, \Delta Y'_s}(\Delta x'_s, \Delta y'_s | x, y)$ is abbreviated as ${}^{\text{simp}}p_{\Delta X'_s, \Delta Y'_s}(\Delta x'_s, \Delta y'_s)$ for simplicity. Moreover, in contrast to the fully affine case in Section 3.1, the variances of the random processes $\Delta X'_s$ and $\Delta Y'_s$ both depend on the variances of *all* estimated parameters and thus $\Delta X'_s$ and $\Delta Y'_s$ are interdependent.

For equal variances $\sigma^2_{e_a} = \sigma^2_{e_b}$, the correlation coefficient ρ becomes zero, since the influence on $\Delta x'_s$ and $\Delta y'_s$ is pairwise similar. Thus $\Delta x'$ and $\Delta y'$ can be considered as uncorrelated and the pdf of the displacement estimation error of the simplified affine model becomes the solution for the fully affine case.

3.2.2 Rate-distortion analysis of the simplified affine model

To derive the minimum required bit rate for encoding the prediction error employing motion-compensated prediction in video coding using the simplified affine model from Section 3.2.1, the derivations from Section 3.1.4 and Section 3.1.5 are employed again. With the Fourier transform ${}^{\text{simp}}P(\Lambda)$ of ${}^{\text{simp}}p_{\Delta X'_s, \Delta Y'_s}(\Delta x'_s, \Delta y'_s)$ (Appendix A.4) from the last subsection and Equation (3.18), the power spectral density of the prediction error ${}^{\text{simp}}S_{ee}(\Lambda)$ for the simplified affine model is derived. Hereby, the same power spectral density of the signal S_{ss} is assumed as derived in Section 3.1.3. Evaluating the rate-distortion theory by exploiting (3.19) and (3.20) (page 45) yields the distortion D and the minimum required bit rate $R(D)$, which correspond to ${}^{\text{simp}}D$ and ${}^{\text{simp}}R\left({}^{\text{simp}}D\right)$, respectively, for encoding the prediction error by using a simplified affine model as defined in (3.29).

The rate-distortion theory for the simplified affine motion-compensated prediction is evaluated in accordance with the procedure described in Section 3.1.6, where the analysis was carried out for the fully affine model. For evaluation, the same exponential drop rates $\alpha_x = 0.9744$ and $\alpha_y = 0.9677$ of the autocorrelation function (Equation (3.16)) of the signal as measured in Table 3.3 (page 51) were assumed. As

discussed in Section 3.2.2, the only difference in the evaluation is that the Fourier transform of the pdf of the displacement estimation error from the simplified affine model ${}^{\text{simp}}P(\Lambda)$ is inserted in (3.18) instead of the Fourier transform of the pdf of the displacement estimation error from the fully affine model $P(\Lambda)$.

Evaluation of the rate-distortion theory for a distortion of SNR = 30 dB results in minimum required bit rates for different variances $\sigma_{e_i}^2$ of Gaussian displacement estimation error pdfs of the affine transformation parameters using a simplified affine model as shown in Fig. 3.5. For the simulations, the affine parameters were assumed to be in a fixed ratio ($\sigma_{e_b}^2 = 2\sigma_{e_a}^2$) and both translational parameters to be equal ($\sigma_{e_c}^2 = \sigma_{e_f}^2$).

The relationship between $\sigma_{e_a}^2$ and $\sigma_{e_b}^2$ is justified by the fact that small rotation angles ($\theta \leq 5° \approx 0.087$ rad) are more likely to occur. Then, exploiting the small-angle approximation, $s_s \cos\theta$ and $s_s \sin\theta$ from (3.27) and (3.28) approximately become s_s and $s_s \cdot \theta$, respectively. Assuming $s_s = 1$ leads to:

$$\begin{aligned} e_a &= \hat{a} - a = \cos\hat{\theta} - \cos\theta \quad = -2\sin\left(\frac{\hat{\theta}+\theta}{2}\right)\cdot\sin\left(\frac{\hat{\theta}-\theta}{2}\right) \\ &\approx -2\sin(\theta)\cdot\sin\left(\frac{1}{2}\Delta\theta\right); \\ e_b &= \hat{b} - b = \sin\hat{\theta} - \sin\theta \quad = \quad 2\cos\left(\frac{\hat{\theta}+\theta}{2}\right)\cdot\sin\left(\frac{\hat{\theta}-\theta}{2}\right) \\ &\approx \quad 2\cos(\theta)\cdot\sin\left(\frac{1}{2}\Delta\theta\right) \end{aligned} \tag{3.43}$$

for $\hat{\theta} \approx \theta$ and with $\Delta\theta = \hat{\theta} - \theta$. With the assumption of a small rotation angle θ, Equation (3.43) results in $e_a \approx -2\theta k_{\text{ang}}$ and $e_b \approx 2k_{\text{ang}}$, with $k_{\text{ang}} = \sin\left(\frac{1}{2}\Delta\theta\right)$ being constant. Exploiting the definition of the variance $\sigma_{e_i}^2 = \int_{-\infty}^{\infty} p(e_i)\left(e_i - E\{e_i\}\right)^2 \mathrm{d}e_i$ and using e_a and e_b as derived above, for small angles $\sigma_{e_a}^2 < \sigma_{e_b}^2$ applies.

The affinities of the aerial video sequences from the TAVT data set (set 1) were measured similarly to the measures presented in Section 3.1.6.1. The results are given in Table 3.4 and support the ratio. It is obvious that the results for the simplified affine model are almost the same as for the fully affine model (compare Table 3.2 on page 49) since barely no motions are contained in the sequences which cannot be covered by the simplified model. Moreover the smaller number of parameters of the simplified model may be estimated more accurately.

The minimum bit rates as a function of the simplified non-translational (purely) affine (axis from center to left) and translational variances (axis from center to right)

Table 3.4: Measured variances $\sigma^2_{e_a}$, $\sigma^2_{e_b}$ [65] of simplified affine transformation parameters of aerial videos from the TAVT data set (set 1) [46, 81].

	$\sigma^2_{e_a}$	$\sigma^2_{e_b}$
350 m sequence	$1.92 \cdot 10^{-7}$	$6.23 \cdot 10^{-7}$
500 m sequence	$1.86 \cdot 10^{-7}$	$3.74 \cdot 10^{-7}$
1000 m sequence	$1.79 \cdot 10^{-7}$	$4.06 \cdot 10^{-7}$
1500 m sequence	$3.21 \cdot 10^{-7}$	$3.67 \cdot 10^{-7}$
Mean	$\mathbf{2.20 \cdot 10^{-7}}$	$\mathbf{4.43 \cdot 10^{-7}}$

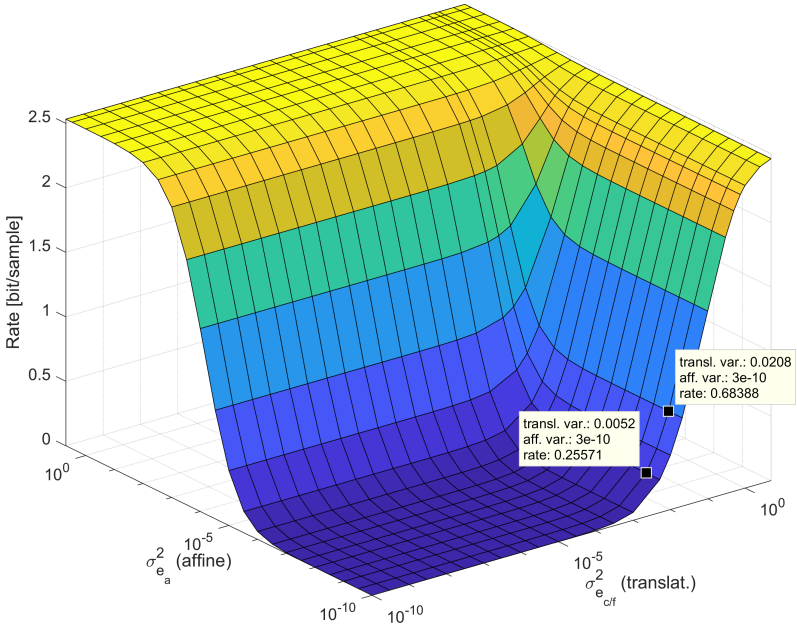


Figure 3.5: Minimum required bit rate versus variances $\sigma^2_{e_i}$, $i = a, b, c, f$ of Gaussian displacement estimation error pdfs for a distortion of SNR = 30 dB assuming $\sigma^2_{e_b} = 2\sigma^2_{e_a}$ and $\sigma^2_{e_c} = \sigma^2_{e_f}$. The surface shows rates for a block size of 64×64 pel and the transform center in the middle of the block [87].

are presented in Fig. 3.5 for a block size of 64×64 pel.

It is noteworthy that the operating point $(\sigma^2_{e_b} = 2\sigma^2_{e_a}, \sigma^2_{e_c} = \sigma^2_{e_f})$ reaches higher bit

rates if the motion contained in the sequence cannot be represented by the motion model. This is the case when a purely translational motion model is used to estimate a sequence containing distinct (non purely translational) affine motion—albeit the resulting bit rate difference decreases for smaller block (or frame) sizes. For the example of a block size of 64×64 pel, the minimum required bit rate for an accurate simplified affine estimation of $\sigma^2_{e_a} = 3 \cdot 10^{-10}$, $\sigma^2_{e_b} = 6 \cdot 10^{-10}$ and $\sigma^2_{e_c} = \sigma^2_{e_f} = 3 \cdot 10^{-5}$ amounts to 0.0020 $^{\text{bit}}/_{\text{sample}}$. For a translational quarter-pel resolution and equal purely affine (non-translational) variances, the bit rate increases to 0.2557 $^{\text{bit}}/_{\text{sample}}$ (lower data tip in Fig. 3.5). On the contrary, for a purely translational motion model with the same translational quarter-pel resolution, the non-translational affine part of the motion contained in the scene cannot be covered at all, leading to high variances $\sigma^2_{e_a} = 2.2 \cdot 10^{-7}$, $\sigma^2_{e_b} = 4.4 \cdot 10^{-7}$ and consequently higher bit rates of 0.2645 $^{\text{bit}}/_{\text{sample}}$ for block sizes of 64×64 pel or 1.5589 $^{\text{bit}}/_{\text{sample}}$ for global motion compensation (both not shown). For the example of translational quarter-pel resolution (which is equal to translational estimation error variances of 0.0052 as already stated), a block size of 64×64 pel and a ratio of the non-translational simplified affine estimation errors of $\sigma^2_{e_b} = 2\sigma^2_{e_a}$, the rate-distortion optimized bit rate for intra encoding of the HD resolution video signal itself amounts to a bit rate of 1.9918 $^{\text{bit}}/_{\text{sample}}$. Considering bit rates and corresponding estimation error variances for encoding the prediction error (Fig. 3.5), it can be concluded that simplified affine motion-compensated prediction achieves improvements for purley affine (non-translational) variances of about $\sigma^2_{e_a} = 3 \cdot 10^{-4}$ or smaller.

The minimum required bit rate for encoding the prediction error of the simplified affine model for different block sizes is shown in Fig. 3.6. The block size represents the number of pels in horizontal and vertical direction each, i. e. the bit rate at "64" at the horizontal axis depicts a block size of 64×64 except for the rightmost two data points "720" and "1080", which correspond to resolutions of 1280×720 and 1920×1080, respectively (as indicated in the plot). The orange line represents the bit rate for using a purely translational motion model with quarter-pel translational accuracy for a sequence with affinities as measured for the TAVT data set (Table 3.2 on page 49). The inherent non-translational variances contained in the sequences are considered as estimation error variances in this case. The green line represents the bit rates for maximum accurate purely affine (non-translational) estimation and translational quarter-pel accuracy whereas the blue line shows results for maximum accurate purely affine estimation and maximum translational accuracy of the implementation [10].

In Fig. 3.7 the bit rates are compared for a fully affine model (six degrees of freedom)

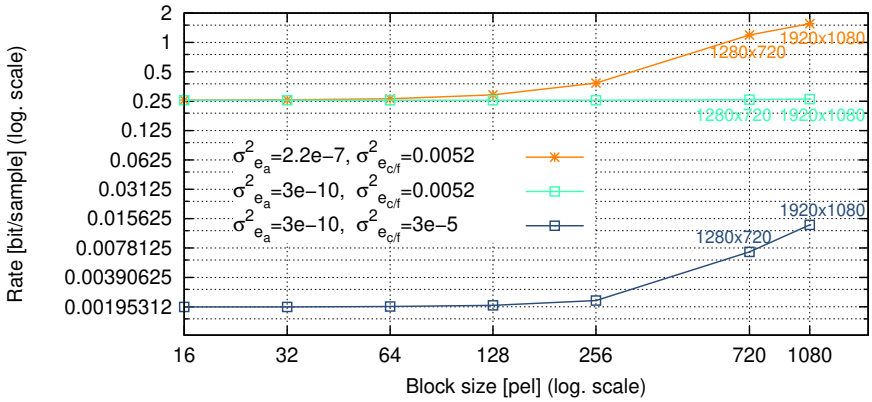
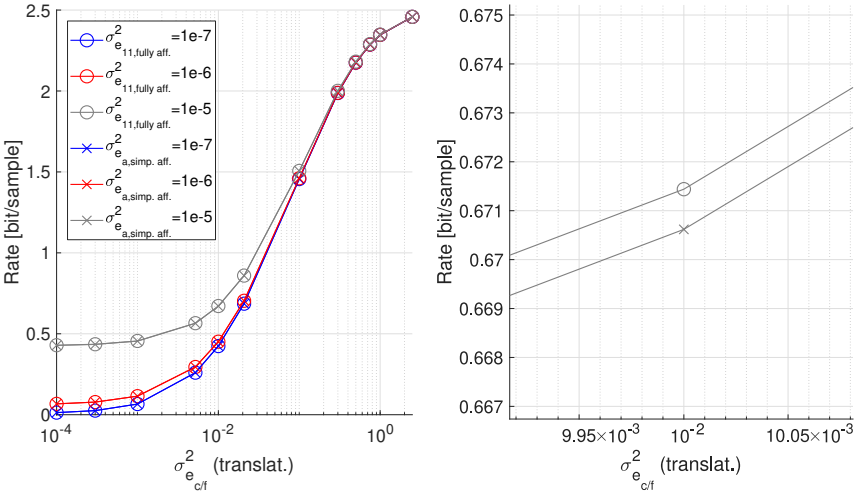


Figure 3.6: Minimum required bit rate for encoding the prediction error of the simplified affine model for different block/frame sizes (SNR = 30 dB, $\sigma^2_{e_b} = 2\sigma^2_{e_a}$, $\sigma^2_{e_c} = \sigma^2_{e_f}$). The block size represents one edge of squared blocks except for the data points 720 and 1080 which represent resolutions of 1280×720 and 1920×1080, respectively (as indicated in the plot). The orange line represents the bit rate resulting from the application of a purely translational motion model with quarter-pel translational accuracy to encode the aerial video sequences from the TAVT data set [46, 81]. Since only translational motion compensation is assumed (for the orange line), the variances of the inherent non-translational motions contained in the sequences are considered as error variances. The green line represents the bit rates for maximum accurate purely affine estimation and translational quarter-pel accuracy whereas the blue line shows results for maximum accurate purely affine estimation and maximum translational accuracy of the implementation [10].

(circles) from Section 3.1 and the simplified, four-parameter model (crosses) discussed in this section for 64×64 pel blocks as used as maximum block size in the current video coding standard HEVC. The plots show that the simplified model requires a smaller amount of bits for encoding the prediction error compared to a fully affine model for equal error variances. This had to be expected since in (3.42) it became obvious that X'_s and Y'_s are correlated for the simplified affine model. On the other hand, the model error may increase for sequences containing motions which cannot be covered by a simplified but by a fully affine model, which occurs, e. g. for shearing.



(a) Simplified vs. fully affine

(b) Magnification

Figure 3.7: Minimum required bit rate and achievable gains of the simplified affine vs. the fully affine motion model for a block size of 64 × 64 pel (SNR = 30 dB, $\sigma^2_{e_b} = 2\sigma^2_{e_a}$, $\sigma^2_{e_c} = \sigma^2_{e_f}$), magnification in (b) [87].

However, the difference between the simplified and the fully affine model is negligible in terms of bit rate saving. Motions which cannot be covered by the simplified affine motion model rarely occur in surveillance video sequences—and presumably also in general videos only to a minor extent. Thus, from a coding point of view it is beneficial to encode as few parameters as possible and consequently, the use of the simplified affine model for encoding purposes is reasonable.

3.3 Summary of Affine Motion-Compensated Prediction in Video Coding

In this chapter, the minimum required bit rate for encoding the prediction error of affine motion-compensated prediction was derived by application of the rate-distortion theory. In particular, in Section 3.1 a fully affine motion model with six degrees of freedom and six independent, zero-mean Gaussian distributed error terms

was assumed. Without loss of generality, simulations were carried out using global motion compensation of full HD resolution video frames as an example. Bit rates for encoding the prediction error were derived. In Section 3.2 similar evaluations were performed for a simplified affine motion model with only four degrees of freedom and the results were compared to those from Section 3.1.

If the motion contained in a scene can be covered by both, the fully affine as well as the simplified affine model, the prediction error is expected to be similar. Since in the case of the simplified affine model the displacement estimation errors in x- and y-direction are correlated (Equations (3.39)–(3.42)), the bit rate for encoding the prediction error is slightly lower for the simplified motion model. On the other hand, the model error might increase for the simplified motion model, if motion is contained in a sequence, which cannot be described by the simplified affine model but by the fully affine model (e. g. shearing). For real-world sequences, however, such motion can be assumed to be rare. Moreover, applied on block-level instead of the entire frame (as performed in global motion compensation), the difference between both models almost disappears.

Compared to a translational motion model, affine motion models may provide significant gains, especially if a high amount of non-translational motion is prevalent in the scenes, as for instance rotation and scaling in aerial sequences. From a coding point of view it is preferable to encode as few parameters as possible. Therefore the application of the simplified affine model for encoding purposes seems reasonable, as in the JEM software for example.

4 ROI-based System for Low Bit Rate Coding of Aerial Videos

In the last chapter, a rate-distortion analysis of affine motion-compensated prediction in video coding was presented. The derivations assumed that a predefined distortion of e. g. SNR = 30 dB is maintained. For aerial video coding, e. g. in surveillance applications, more restrictive constraints may apply. It may become important to reduce the bit rate even more compared to standardized video codecs in order to ensure the transmission of a video signal over very small bandwidth channels.

Taking these additional considerations into account, a region of interest detection and coding system is proposed in this chapter, which is able to further increase the coding efficiency for aerial video sequences beyond the capabilities of common AVC and HEVC video encoders. Parts of this chapter have already been published in [75, 89, 79, 81, 85, 83]. Task-dependent extensions and improvements have been previously published in [80, 84, 78, 81, 77, 82, 86]. These are a stereo en- and decoding of aerial video sequences from a monocular camera system [80, 84] and different improved—but more computationally intensive—moving object detectors [78, 81, 77, 82] as well as an improved long-term mosaicking [86].

The ROI detection and coding system (Fig. 4.1) exploits the characteristic of aerial video sequences of planar landscapes to maintain full resolution and high quality videos over the entire frame at low bit rates. It relies on the encoding and transmission of new emerging areas in each frame (new areas, ROI-NA), which are stitched together in a postprocessing step at the decoder to reconstruct the static parts of the scene (background) by means of global motion compensation (GMC) [75, 79]. In order to retain the motion of moving objects not conforming with the motion of the ground (like moving cars and their previously occluded ground), regions containing such moving objects (ROI-MO) are additionally considered as interesting. Both ROIs are used as input for a video preprocessing. Since non-ROI areas are replaced in the postprocessing, those areas are substituted by content which can be encoded most efficiently, e. g. just black pels. As a consequence, non-ROI areas are encoded at nearly no bit cost and almost the entire available bit rate is assigned to the encoding of ROI areas by common off-the-shelf video encoders like a HEVC encoder. Since only small parts of each frame have to be encoded as ROI, this ROI detection and coding system is capable of providing high video image quality at low bit rates.

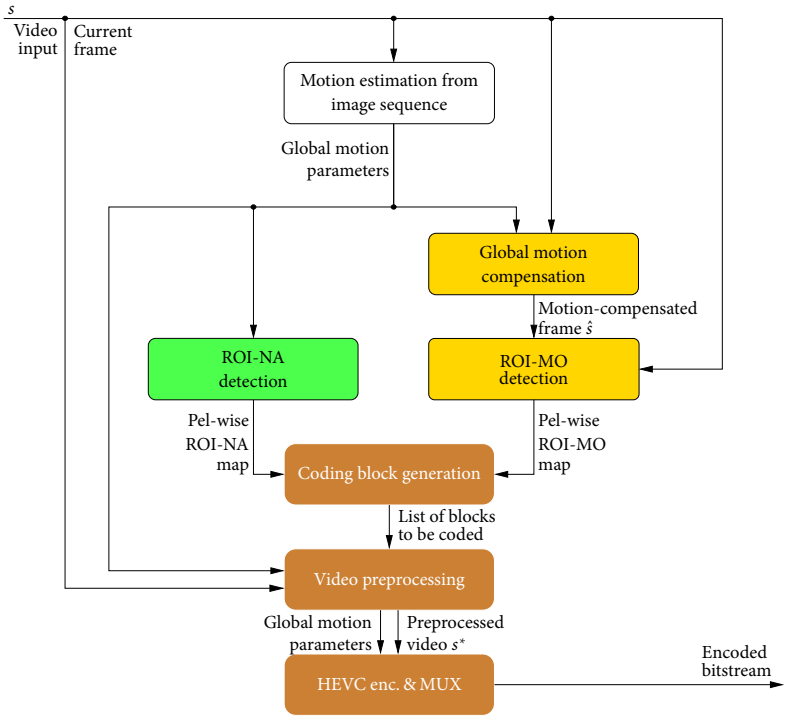


Figure 4.1: Simplified block diagram of ROI detection and coding system. White: motion estimation from image sequence; gold: ROI-MO detection blocks; green: new area detection; brown: block generation, video preprocessing, video encoding ("HEVC enc.") and multiplexing (MUX) (based on [75, 77, 81]).

The entire basic block diagram of the proposed ROI coding system for aerial surveillance video sequences is depicted in Fig. 4.1 (based on [75, 77, 81]).

This chapter is organized as follows:

The detection of new areas (ROI-NA) is introduced in Section 4.1 (white and green blocks in Fig. 4.1). Section 4.1.2 considers typical distortions during the global motion estimation process as introduced by uncalibrated cameras which are affected by radial distortion. For that the joint estimation of several homographies and one unknown, but piecewise constant radial distortion is proposed and analyzed in Section 4.1.2.1.

Due to its complexity, the latter approach is not real-time capable, especially not for a higher number of jointly estimated homographies. Thus, a real-time capable in-loop radial distortion compensation (RDC) is proposed in Section 4.1.3.

The detection of moving objects (ROI-MO) is described in Section 4.2 (golden blocks in Fig. 4.1). Using the example of a simple, yet effective background subtracting- and difference image-based moving object detector, the detection and processing of moving objects is explained in Section 4.2.1. Two application specific enhancements for the moving object detector are introduced in Section 4.2.1.1. Without structural changes, the proposed more sophisticated moving object detector can be easily integrated into the modular system.

The ROI encoding process itself is described in Section 4.3 (brown blocks in Fig. 4.1). A general ROI coding approach is proposed to become independent of any encoder modification, as typically used for common ROI-based coding approaches.

Afterwards, the ROI decoding by construction of a mosaic—often also referred to as (aerial) panoramic image—to reconstruct static parts of the ROI encoded video from the new areas is explained in Section 4.4.

Finally, in Section 4.5 the video reconstruction from the mosaic and the insertion of moving objects at appropriate positions in the video are explained.

4.1 ROI: New Areas (NAs)

The proposed ROI detection and coding system essentially profits from the encoding of solely new emerging areas of each frame (and moving objects, which will be considered in detail in Section 4.2). This section describes the calculation of the new areas in Section 4.1.1. In Section 4.1.2, the radial distortion as the main typical lens distortion is considered in the context of global motion estimation (and compensation) of aerial video sequences.

4.1.1 Calculation of the new areas

Assuming a planar landscape, the global motion between two video frames can be described by an affine or projective transformation, depending on specific restrictions regarding the camera orientation. Whereas in the previous chapter (Chapter 3), a video camera in nadir view was assumed, small deviation angles are allowed for the proposed system in this chapter to reflect more practical situations were an exact nadir view cannot be guaranteed. This relaxation of the assumptions has two implications: firstly, an affine transformation like considered in Chapter 3 may not be sufficient any

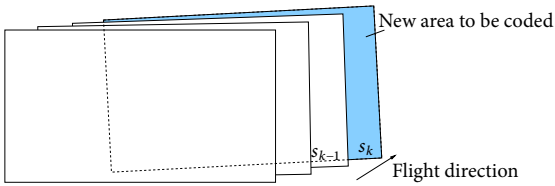


Figure 4.2: New area detection (based on [75]).

more to describe the global motion but a fully projective transformation may have to be employed instead (Section 2.3, page 18). Secondly, although the global motion *estimation* can be performed for arbitrary viewing angles of the camera (as long as the surface of the scene is predominantly planar), the ROI decoding by mosaicking (see Section 4.4 for details) will face implications for larger deviations of the camera from nadir view.

The global motion estimation from image sequences is performed according to Section 2.4 (page 20), using corner features, a KLT feature tracker for the generation of a sparse optical flow and a RANSAC outlier removal. Any pel from the current frame s_k, which is projected outside the previous frame s_{k-1} with the estimated transform parameters is considered as *new area* (ROI-NA, blue parts in Fig. 4.2). The estimated motion parameters as well as the map of the new areas are transmitted to the decoder as side information.

4.1.2 Long-term mosaicking of aerial videos

In the last subsection the estimation of the global motion parameters was shortly reviewed. For undistorted video sequences, such a frame-to-frame-based approach is quite sufficient and the result of several concatenated frame-to-frame projections is approximately the same as the estimation from one frame s_k to a specific frame $s_{k-n_{\mathrm{mos}}}$ with $n_{\mathrm{mos}} \gg 1$. For lens distortion affected video sequences, however, this assumption does not hold, since the distortions of objects, and consequently the feature positions employed for global motion estimation, are typically location-dependent in a frame. This is especially true, e. g. for radial distortion as the most important lens distortion. Thus, for a reliable frame-to-frame-based global motion estimation over a high number of frames, the radial distortion has to be taken into account during the global motion estimation itself. Such an approach is introduced in Section 4.1.2.1. Since it will become apparent that this proposed approach cannot be realized in real-time, it is also inappropriate for the desired aerial surveillance

task with on-board processing on small and medium UAVs with limited energy and computational resources. Hence, in Section 4.1.3 a practical solution is proposed for on-board usage of UAVs, which only introduces a limited delay and is real-time capable in principle.

4.1.2.1 Joint homographies and radial distortion estimation

Since in surveillance scenarios the camera as well as its parameters (e. g. focal length) are constant over several frames, the joint estimation of several homographies and one unknown, but constant radial distortion is proposed. Due to the limited motion vector accuracy in current video coding, it is sufficient to only consider the second order radial distortion according to [8, 124]. Thus, only the first radial distortion parameter κ_1 (Equation (2.2) in Section 2.2.2 on page 14) is optimized in a way that the distortions introduced by a frame-to-frame estimation are kept small for all estimated frames, and thus to enable long-term global motion compensation (Section 4.4).

With the assumption of radially symmetric lenses, the radial distortion parameters in horizontal and vertical direction $\kappa_{1\,x}$ and $\kappa_{1\,y}$, respectively, are equal: $\kappa_1 \coloneqq \kappa_{1\,x} = \kappa_{1\,y}$. With this assumption and Equation (2.3), the undistorted point coordinate (x, y) can be computed in component-wise notation from the distorted coordinate $(x_\mathrm{d}, y_\mathrm{d})$, the first radial distortion parameter κ_1 and the distorted radius $r_\mathrm{d} = \sqrt{{x_\mathrm{d}}^2 + {y_\mathrm{d}}^2}$ as:

$$\begin{aligned} x &= x_\mathrm{d}\left(1 + r_\mathrm{d}^2\kappa_1\right) = x_\mathrm{d} + x_\mathrm{d}^3\kappa_1 + x_\mathrm{d}y_\mathrm{d}^2\kappa_1\,, \\ y &= y_\mathrm{d}\left(1 + r_\mathrm{d}^2\kappa_1\right) = y_\mathrm{d} + y_\mathrm{d}^3\kappa_1 + y_\mathrm{d}x_\mathrm{d}^2\kappa_1\,. \end{aligned} \tag{4.1}$$

In homogeneous coordinates and assuming a common 3×3 projective transformation (homography) matrix H, a point can be mapped from its homogeneous source coordinates $\boldsymbol{x} = (x, y, 1)^\top$ in one frame to its destination coordinates $\boldsymbol{x}' = (x', y', 1)^\top$ in the second frame:

$$\boldsymbol{x}' = \mathrm{H} \cdot \boldsymbol{x}\,. \tag{4.2}$$

A concatenation of several homographies leads to

$$\boldsymbol{x}'^n = \left(\mathrm{H}_n \cdot \ldots \cdot \mathrm{H}_1\right) \cdot \boldsymbol{x}\,. \tag{4.3}$$

To map the undistorted coordinates $\boldsymbol{x}'$ back to the distorted ones $\boldsymbol{x}'_\mathrm{d}$, the inversion of (4.1) with a given κ_1 is employed and (x, y) is replaced by (x', y'), $(x_\mathrm{d}, y_\mathrm{d})$ by $(x'_\mathrm{d}, y'_\mathrm{d})$, and r_d by r'_d, respectively. To numerically solve this non-linear inversion, a Quasi-Newton method is used to determine the corresponding radially distorted

Table 4.1: Run-times and MSE values of Matlab's (Quasi-Newton) numerical solver (Unconstrained Minimization, function `fminunc`). No noise, initialization of solver with movement of distorted point cloud center, optimal value: $\kappa_1 = -3 \cdot 10^{-3}$, optimization criterion: MSE.

# of homographies	MSE $\leq 10^{-3}$		MSE $\leq 10^{-6}$	
	$\kappa_{1\,\mathrm{est}}$	Time in s	$\kappa_{1\,\mathrm{est}}$	Time in s
2	$-2.61 \cdot 10^{-3}$	7.8	$-2.91 \cdot 10^{-3}$	10.9
3	$-2.91 \cdot 10^{-3}$	17.3	$-2.98 \cdot 10^{-3}$	42.0
4	$-2.01 \cdot 10^{-3}$	32.7	$-2.99 \cdot 10^{-3}$	222.2
5	$-3.66 \cdot 10^{-3}$	57.1	$-2.97 \cdot 10^{-3}$	1408.0
10	$-3.48 \cdot 10^{-3}$	257.8	$-2.99 \cdot 10^{-3}$	561436.6

point coordinates $\boldsymbol{x}'_{\mathrm{d}}$ in dependence of the undistorted point coordinates $\boldsymbol{x}'$. The mean squared error (MSE) is optimized, denoting the mean squared Euclidean distance between all estimated feature points $\boldsymbol{x}'_{\mathrm{d}}$ and their corresponding measured feature point positions $\hat{\boldsymbol{x}}'_{\mathrm{d}}$.

4.1.2.2 Simulations for the joint homographies and radial distortion estimation

Simulations using Matlab 2017 [118] by a Quasi-Newton descent prove the approach of a joint estimation of several homographies and one common radial distortion parameter κ_1 to be applicable. For noise-free, artificially generated point clouds with $N = 1000$ points and using RANSAC, approximate solutions according to Table 4.1 are achieved.

Since the run-time increases exponentially, it is obvious that this approach is impractical for the joint estimation of higher numbers of homographies. Thus, a simplified, iterative, adaptive estimation approach is proposed in the next subsection.

4.1.3 In-loop radial distortion compensation

To overcome the computational complexity of the latter radial distortion estimation approach, a real-time capable alternative for the radial distortion estimation and compensation is proposed in this section. In contrast to other approaches, an optimized radial distortion parameter *for the projection* shall be estimated, which does not necessarily match the correct radial distortion. Since this approach works without any prior knowledge, it can adapt to any aerial video sequence which can be

mapped onto a plain.

The KLT- and RANSAC-based global motion estimation framework from Fig. 4.1 is used as a basis. In contrary to the global motion estimation in Subsection 4.1.1, the global motion estimation here is first performed for every pair of subsequent frames of a video sequence within a predefined group of frames (GOF). Further on, based on geometric constraints, the projection of a frame of the GOF is regularized by restricting the change of shape and size of the projection from the camera-plane onto a projection plane. If predefined thresholds are not reached, the feature positions are converted to corresponding feature positions as if they had been affected by another radial distortion. Afterwards, the global motion parameters are estimated and compared with the predefined values again. In an iterative process with a maximum predefined number of iterations the best fitting projection is selected (Fig. 4.3).

The regularization used in this work is based on physical limitations of an UAV: the homography matrix $\mathtt{H}$ can be decomposed into the rotation matrix $\mathtt{R}$ and the translational vector $\boldsymbol{t}$ from one view to the other, the camera parameter matrices $\mathtt{K}$ and $\mathtt{K}'$ of the views, and the surface normal vector $\frac{\boldsymbol{n}_\mathrm{s}}{d_\mathrm{s}}$, with d_s being the distance between the camera center and the surface [42]:

$$\mathtt{H} = \mathtt{K}' \left(\mathtt{R} - \frac{\boldsymbol{t}\boldsymbol{n}_\mathrm{s}^\top}{d_\mathrm{s}} \right) \mathtt{K}^{-1} \,. \tag{4.4}$$

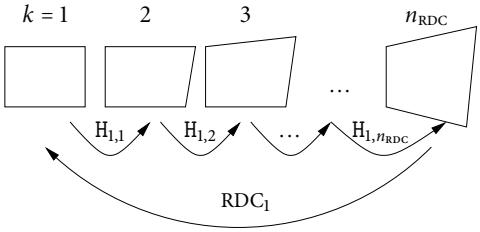
The rotation matrix is represented as

$$\mathtt{R} = \exp\left(\theta_y(t)\,\mathtt{W}_y\right) \cdot \exp\left(\theta_x(t)\,\mathtt{W}_x\right) \cdot \exp\left(\theta_z(t)\,\mathtt{W}_z\right) , \tag{4.5}$$

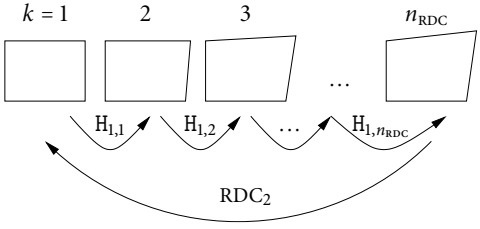
with $\mathtt{W}_x, \mathtt{W}_y, \mathtt{W}_z$ being the skew-symmetric matrices induced by rotation around the X-, Y- and Z-axis, respectively. Assuming the surface geometry to be constant and the Z-axis to be vertical to the surface of the earth, the change in size and shape of the projection of the camera target onto a plane depends on $\theta_x(t)$ and $\theta_y(t)$. For a typical (and physically possible) motion of an aerial vehicle, $\theta_x(t)$ and $\theta_y(t)$ are assumed to only change slowly. Thus, the radial distortion parameter is optimized such that

$$\left| \frac{\mathrm{d}}{\mathrm{d}t}\theta_x(t) \right| < c_x \qquad \text{and} \qquad \left| \frac{\mathrm{d}}{\mathrm{d}t}\theta_y(t) \right| < c_y \,. \tag{4.6}$$

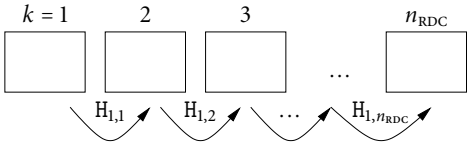
Here c_x and c_y are thresholds which limit the rotation around the x- and y-axis, respectively. For the iterative algorithm, the video is separated into groups of frames (GOFs) first, e. g. containing $n_\mathrm{RDC} = 60$ frames each. Thus, this approach can be used on-the-fly for each GOF and the maximum delay is n_RDC frames. Moreover,



(a) Initial global motion estimation (GME) and first radial distortion compensation (RDC_1)



(b) Second GME improves the projections



(c) Third GME leads to optimal results

Figure 4.3: Iterative in-loop radial distortion compensation (RDC).

the motion estimation for each pair of frames can also be parallelized for speed-up reasons.

The radial distortion parameter for the first GOF $l = 1$ is initialized to $\kappa_{1,l=1} = 0$, but can also be arbitrarily selected. As the radial distortion parameter usually does not change or only changes slowly over time (e. g. for different intrinsic camera parameters such as focal length), $\kappa_{1,l}$ for all subsequent GOFs is initialized with the parameter $\kappa_{1,l-1}$ of the previous GOF. In an iterative loop, first, homographies for each frame of the current GOF are estimated with the fixed, current $\kappa_{1,l_{\text{cur}}}$. With these homographies, the mapping of each frame into the mosaic is evaluated. A plausibility

check is performed, based on geometrical constraints. Assuming a constant flight altitude and fixed camera parameters for the current GOF, geometric properties of the projected frame s_k have to be similar to those of the projected frame s_{k-1}, i. e. the frame should have similar shape and size. The requirements that lengths of opposite sides as well as the size of the projected frames have to change only slowly turned out to be robust measures. Since both of these measures have distinct minimums for the optimum radial distortion parameter, they are well-suited for optimization. Thus, these changes were restricted by a geometric check to be smaller than $c_{\text{shape,max}}$ and $c_{\text{size,max}}$, respectively. If these constraints are not fulfilled, another $\kappa_{1,l}$ is selected using an iterative bisection method (Fig. 4.4). Using values of the current $\kappa_{1,\text{cur}}$ plus or minus $\Delta\kappa_1$ are tested for the first order radial distortion parameter ($\kappa_{1,\text{tmp1}}$ and $\kappa_{1,\text{tmp2}}$ in Fig. 4.4) and the corresponding κ_1 is selected which minimizes the distortion of the projection for the next iteration. The loop is repeated until convergence or until a maximum number of iterations i_{RDC} ("$\text{counter}_{\text{max}}$" in Fig. 4.4) is reached. Since the approach aims at the minimization of the distortion of the projected frames, the estimated radial distortion will not necessarily be the real camera lens distortion but can also conceal (limited) model violations.

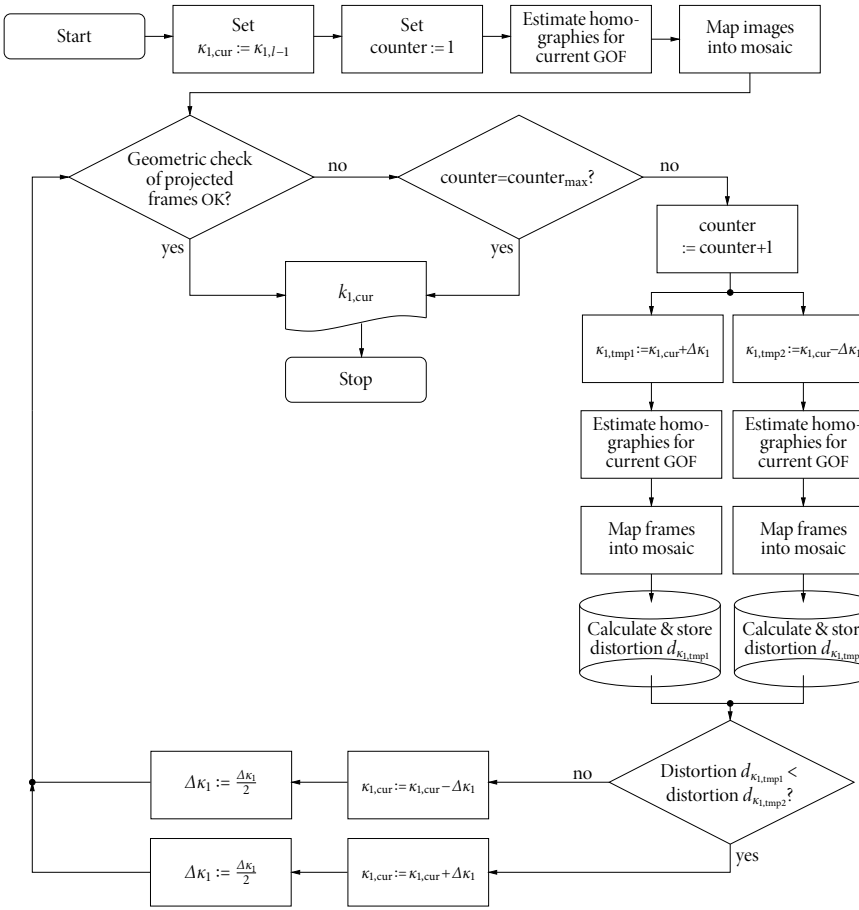


Figure 4.4: Flowchart of the selection of $\kappa_{1,l}$ for the group of frames (GOF) l by the in-loop radial distortion compensation (RDC) algorithm.

4.2 ROI: Moving Objects (MOs)

In the last subsections, an efficient ROI-based coding system was introduced, which solely relies on the encoding of new emerging areas (ROI-NAs) in each frame. In such a system, moving objects are displayed statically at the position of their first occurrence. Since no additional information about their movement is available, they are also considered to be static background.

For aerial surveillance, however, moving objects are typically of high interest. Thus, in order to retain the motion information also for ROI encoded and decoded video sequences, it is important to additionally define moving objects as regions of interest (ROI-MO). Consequently, the block diagram (Fig. 4.1 on page 66) of the ROI detection and coding system consists of a dedicated ROI-MO detector (yellow block).

4.2.1 Highly performant difference image-based moving object detection

The detection of moving objects is performed by background subtraction as explained in Section 2.8.1.1: the pel-wise difference in the luminance channel (Y) between the current frame s_k and the global motion-compensated prediction $\hat{s}_k$ is computed (further referred to as *difference image*). Spots of high energy which are not removed as noise are marked as MOs in an *activation mask*, which will be used later on for the encoding (Fig. 4.7 on page 78).

Such a background subtraction-based approach is a simple, yet effective moving object detection method, especially for aerial video surveillance with (predominantly) planar landscape as assumed in this work. However, for specific applications with different conditions, more specifically adapted ROI-MO detectors may deliver better results. Previously published work on such application-dependent solutions is summarized in the following subsection. Due to the modular concept, any arbitrary ROI-MO detector may be used in the proposed framework.

4.2.1.1 Application-dependent ROI-MO detector improvements

A block diagram of an enhanced ROI detection and coding system may be organized as shown in Fig. 4.5 (based on [81]). The basic structure of the system including the global motion estimation based on sparse optical flow calculation (white block "Optical flow (KLT)" and green block "RANSAC planar landscape model" in Fig. 4.5), ROI-NA detection (lower green block in Fig. 4.5), and the video encoding itself (brown blocks in Fig. 4.5) remains similar to that in Fig. 4.1 (page 66), although there are

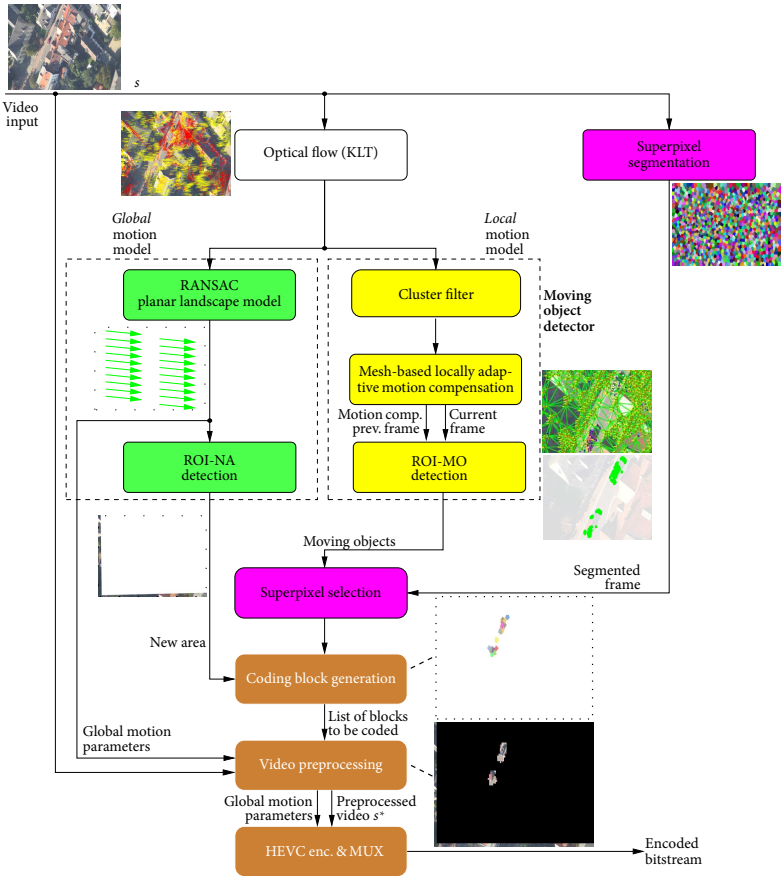


Figure 4.5: Block diagram of an enhanced ROI detection and coding system. White: optical flow using a Kanade-Lucas-Tomasi (KLT) feature tracker; yellow: cluster filter to eliminate *false positive* MO detections and mesh-based motion estimation/compensation incl. ROI detector; magenta: superpixel segmentation and selection; green: global motion estimation and new area detector (including RANSAC outlier detection); brown: block generation, video preprocessing, video coder ("HEVC enc.") and multiplexing (MUX) (based on [81]).

(a) Original (excerpt) (b) MO detection mask (c) Decoded frame

Figure 4.6: Original excerpt (a) and reconstructed image after ROI encoding and decoding (c) with inaccurate MO detection due to homogeneous, unstructured regions on the car roof. Missing detections (b) of the rear part of the red car as ROI-MO lead to reconstruction errors since the front part of the car, which was detected as ROI-MO, does not match the reconstructed background due to the ego-motion of the car [78].

additional blocks for a sophisticated—and much more computationally complex—moving object detection. The additional blocks are summarized in the following paragraphs. A more thorough description of this enhanced system can be found in [81].

Improved shape retrieval of moving objects by integration of temporally consistent superpixels in the moving object detection

Difference image-based moving object detectors lack accuracy when it comes to unstructured, homogeneous regions within the MOs—e. g. car roofs—as for those areas the pel-wise differences between the current and the motion-compensated frame are relatively small [6]. Fig. 4.6 illustrates occurring problems: if parts of a MO (original in Fig. 4.6a) are detected as ROI whereas other parts of the same MO are not recognized (MO detection mask in Fig. 4.6b) and the particular MO moves too fast, reconstruction errors might occur since the motion-compensated background and the moving object (foreground) might not match exactly, leading to errors in the reconstructed video (Fig. 4.6c, cf. Sections 4.4 and 4.5 on pages 84 and 85, respectively, for details). To overcome such problems especially in environments where the assumption of planarity is not thoroughly fulfilled any more, moving object

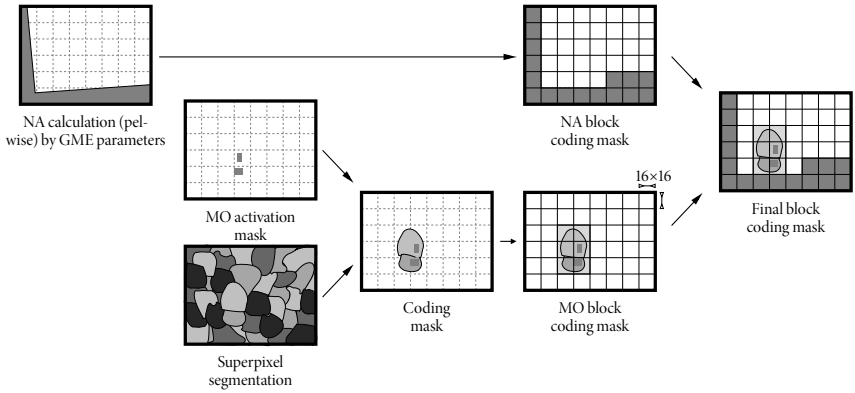


Figure 4.7: Coding mask generation for new area (top row) and moving objects. The *MO activation mask* from the difference image calculation is overlaid with an independent *superpixel segmentation* in order to obtain accurate shape information of the moving objects. The *coding mask* is adapted to a coding block pattern (*MO block coding mask*) and combined with the *NA block coding mask* to the *final block coding mask*. Any block marked in the latter mask will be encoded as ROI (based on [81]).

areas may be identified more accurately by combining an independently calculated superpixel segmentation (upper right magenta block in Fig. 4.5) with the difference image-based detector result (Fig. 4.7, middle and bottom row). The results from the difference image-based detector are used as seeds to automatically activate only those superpixels containing moving objects (lower magenta block "Superpixel selection" in Fig. 4.5). Since the superpixels exploit not only the gray-values of neighboring pels but also consider weighted color and spatial-distance information, they are an ideal complement for the combination with the difference image-based detector. Additionally, by using a *temporally consistent* superpixel segmentation [105] the system is able to bridge temporal detection gaps, thus reducing the amount of missed detections per frame (see Fig. 4.8 for illustration). As shown in [78], the temporally consistent superpixels are able to outperform other state-of-the-art segmentation methods like an efficient *GraphCut*-based *SlimCut* implementation [106].

Reduction of false positive detections of MOs by integrating a mesh-based moving object detection

Given the use of the projective transformation, a planar ground is assumed which is

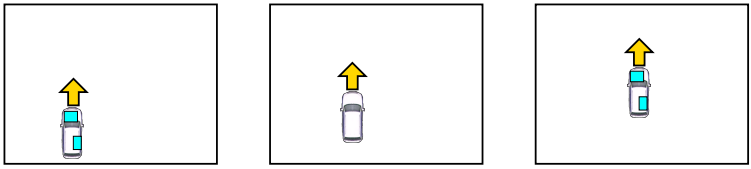


Figure 4.8: *Temporally consistent superpixels* (TCS) are used to bridge *false negative* detections of the ROI-MO. In the figures *true positive* detections within the MO (white car) are marked as cyan blocks. If no moving object is detected by the MO detector in frame s_{k-1}, the MO in that frame would not be selected for coding. Due to the temporal consistency of the superpixels, the position of the car can also be predicted in frame s_{k-1} and thus correct processing and transmission of the car in all frames can be achieved [81].

(prevalently) true for sequences recorded at high flight altitudes. This assumption is not suitable for non-planar ground structures like buildings or trees. These lead to image regions falsely detected as MO (*false positive* detections, FP) resulting in an increased ROI area. Consequently an increased number of superpixels is selected for encoding. For the moving object detection the replacement of the planar GMC by a mesh-based motion estimation and compensation is proposed [90, 91]. Instead of one global plane for the entire frame, multiple smaller planes are used to enable the motion-compensated image to adapt to non-planar scene geometry (yellow block "Mesh-based locally adaptive motion compensation" in Fig. 4.5) [81]. The mesh-based local motion estimation and compensation as well as a locally adaptive outlier detector (yellow block "Cluster filter" in Fig. 4.5) for the moving object detection (yellow block "ROI-MO detection " in Fig. 4.5) in non-planar areas has been shown to achieve superior detection rates in [90] and the integration in a GMC-based ROI detection and coding system as described here has also been proven to be successful [81].

4.3 ROI Coding of Aerial Video Sequences

The drawback of common ROI coding approaches is the degradation of non-ROI areas that cannot be reconstructed at full quality by the decoder [59, 28, 19, 79, 16]. This problem is overcome by the proposed system via reconstruction of non-ROI areas by means of global motion compensation (Section 4.4). As a consequence, a subjectively

(a) Original (b) Reconstructed from new areas

Figure 4.9: Perspective view change (highlighted by red ellipse) introduced by ROI coding and decoding and GMC (*350 m sequence* from the TAVT data set [46, 81]).

high quality over the entire frame is provided at the expense of perspectively incorrect views at static parts of the scene. Since no update information is provided for ROI-NAs, the angle of view at 3D structures like houses or trees remains the same as it was when they emerged the first time, as long as these objects are visible in the video. The effect is illustrated in Fig. 4.9: while Fig. 4.9a shows the original frame as recorded by the camera, a slightly different view is visible in Fig. 4.9b (red ellipse) due to global motion compensation. It is noteworthy that such perspective distortions only occur at non-planar structures and increase for higher 3D structures not matching the ground plane. Some luminance changes are also visible at the upper and lower lines of Fig. 4.9b due to an altered global illumination. Moreover, vignetting effects may occur all over a ROI encoded and decoded video (not visible in Fig. 4.9), since the entire background is reconstructed based on new areas originally located at the frame borders—which are most affected by vignetting effects.

To determine the final ROI coding mask, it is assumed that any ROI detector provides pel-wise information of the detected ROIs. This ensures that arbitrary ROI detectors and any number of ROI detectors can be used as long as they provide a pel-wise classification of ROI and non-ROI, e. g. indicated by a binary "1" or "0", respectively, in a pel-wise map. As depicted in Fig. 4.7, the pel-wise information of ROI-NA as well as ROI-MO is extended to a fixed block grid. If at least one ROI-NA or ROI-MO pel is located in a block of a predefined size (16×16 is used as grid block size in

Figure 4.10: Example ROI areas for one frame of the suburban test sequence *350 m sequence* from the TAVT data set [46, 81].

this work), this block is marked for encoding as ROI in a *final block coding mask* as shown in Fig. 4.7 for the enhanced system. For the basic system without superpixels, the coding mask generation works accordingly and the *MO activation mask* is used directly without superpixel-enhancement.

Although a standard compliant bit stream is produced by the encoder, an additional postprocessing is necessary to reconstruct non-ROI areas (static background) of the scene [75, 79], e. g. in a mosaic (Section 4.4) or in a reconstructed video sequence (Section 4.5). For this postprocessing the entropy-encoded block coding mask (map) as well as the global motion parameters are encapsulated as *supplemental enhancement information* (SEI) in the bit stream.

The coding gain of the system, compared to the common encoding of entire frames without any ROI processing or coding, depends on the amount of ROIs to be encoded. As an upper limit the entire frame has to be encoded (e. g. if moving objects are distributed all over the frame). The additional ROI map and transform parameters have to be encoded in either case in this ROI coding system. However, since the bit rate of both additional elements is negligible compared to that of the video stream, the resulting coding efficiency approximately resembles that of the unmodified video coder. As a lower limit no ROI areas have to be encoded of the entire frame, for instance if no UAV motion is prevalent and no moving objects are detected within the frame. This results in the encoding of control and syntax elements, the final block coding mask and the global motion parameters only. The latter cause a small bit rate in relation to the overall bit rate also for low bit rate scenarios. For typical scenarios only a few percent of each frame have to be encoded and transmitted. An example of all ROI areas to be encoded for one frame of a suburban video sequence is shown for the *350 m sequence* from the TAVT data set [46, 81] in Fig. 4.10 (non-ROI areas represented by white).

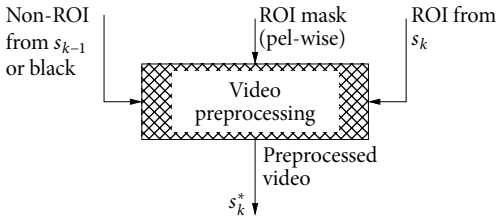


Figure 4.11: General ROI preprocessing. Image composition by preprocessing as introduced by the proposed general ROI coding. The pel-wise ROI information is expanded to blocks within the *video preprocessing* (e. g. of size 16×16 pel), based on [85].

The actual video encoding of the proposed ROI coding system is performed using a common, unmodified video encoder. In contrast to typical other ROI coding approaches, absolutely no encoder modifications are necessary, which may be challenging, time-consuming and thus cost-intensive to implement. Moreover, to exploit later encoder optimizations or even the coding efficiency of new or other video coding standards, the video encoder itself can be replaced by any off-the-shelf video encoder without loss of functionality of the ROI system.

The idea of the proposed general ROI coding system is to dedicate *any* coding decision to the highly optimized encoder internal rate distortion optimization (RDO). Accordingly, it has to be ensured that the input video stream contains only relevant information needed at the decoder for reconstruction. Areas of each frame, which are irrelevant for the reconstruction of a frame (non-ROI) are replaced in order to encode the entire image as efficiently as possible (Fig. 4.11) [85]. This means that in contrast to common video coding, where the output is optimized to be as similar to the input as possible, modifications of the input signal by the general ROI preprocessing are deliberately accepted.

In Fig. 4.12 the encoding scheme from the camera to the encoded bit stream is depicted. As suggested above, two operation modes are distinguished, where each non-ROI block in the current frame s_k is replaced by:

mode 1: the corresponding block from the (temporally) preceding frame s_{k-1}. This mode aims at utilizing coding tools for unchanged content (e. g. skip mode).

mode 2: a black block. This mode aims at utilizing coding tools for monochrome areas (skip, DC intra prediction).

Both modes are based on the fact that non-ROI areas in the preprocessed video frame s_k^* are discarded in the postprocessing step in any case. However, subsequent

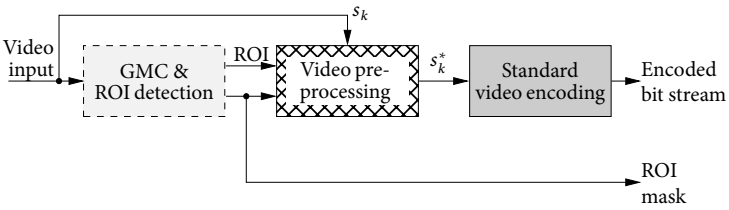


Figure 4.12: Entire encoding scheme from the camera to the encoded bit stream using the proposed general ROI coding system (based on [85]).

image reconstructions have to be applied as required by the specific ROI detection and coding system (see Sections 4.4 and 4.5). The GMC parameters as well as the ROI maps are encapsulated in the SEI message (or any similar private data field as offered by that particular video codec), as they are needed by the postprocessing for reconstruction of the frame. It is stressed again that both, pre- and postprocessing are *independent* of the employed video codec and that no image degradation is caused by general ROI coding—except for errors introduced by the video encoding itself, i. e. quantization errors for lossy coding modes. Consequently, no restrictions for special video codecs are triggered by the preprocessing nor by the postprocessing of the general ROI coding framework.

4.3.1 Inherent noise removal of the proposed general ROI coding

Modern hybrid video encoders like HEVC already consist of very efficient coding modes also for encoding static parts of a frame (e. g. *direct mode* in AVC, *merge/skip mode* in HEVC). These modes are most efficient for noise free signals s. However, camera captured signals contain additive white Gaussian noise (AWGN) n_G with the noise power $N_\mathrm{P}(n_\mathrm{G})$. Thus, the superimposed signal $s+n_\mathrm{G}$ is the input of the video encoder. Assuming perfect motion compensation, the noise of the reference frame used for motion compensation $n_{\mathrm{G},k-1}$ has to be removed and the noise $n_{\mathrm{G},k}$ of the current frame has to be added to the prediction error signal. The resulting noise power accumulates to $2{\cdot}N_\mathrm{P}(n_\mathrm{G})$ in the prediction error signal, leading to higher bit rates than required for encoding s only.

Using ROI and general ROI coding, the noise is encoded only once at the first occurrence of each coding block, since in all subsequent frames a copy is inserted from the temporally preceding frame (mode 1) or the block is entirely replaced by a

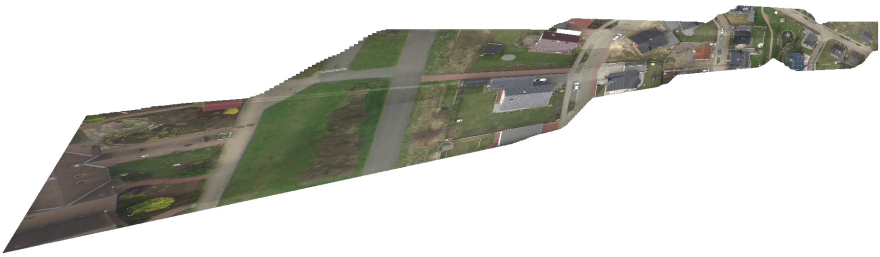
black block (mode 2). Consequently, the resulting bit rate will decrease.

4.4 Mosaicking of ROI-Encoded Videos

One common procedure of visualizing aerial surveillance videos is the generation of a mosaic. Such an approach is also applicable for ROI encoded aerial videos. As will be explained in the next subsection, it is also possible to reconstruct a video out of ROI encoded aerial videos. For that, however, a mosaic also has to be generated. In this context, one could say that during the reconstruction of a video from ROI encoded data, a mosaic can be obtained without extra effort and functions as supportive overview image.

Depending on the goal of the mosaicking itself, two different approaches have to be distinguished for the mosaicking of ROI encoded aerial videos. If the mosaic is only needed for reconstruction of the video frames (see Section 4.5), a short-time mosaic is sufficient. The principle of a possible memory management can be found in [44]. There, object memories were used for the original data, e. g. one object memory could store the ROI information of one frame. Based on the transformation parameters a motion vector is derived for each pel position in the current frame. This motion vector points to the corresponding object memory, which contains the appropriate ROI information. Such an approach prevents multiple filtering since any required information is saved in the original resolution of the ROI without any filtering. Older ROI information from previous frames, which is not referenced any more by the current frame, can be discarded.

In contrast to a short-time mosaic approach, the mosaic may be additionally used for visualization of the overflown area. In that case, a long-term mosaic has to be generated. This may be especially beneficial for surveillance tasks and is possible without additional bit rate expense for the encoded ROI data. Thus, a long-term mosaic approach is used in this work. To generate a long-term mosaic, all ROI blocks as signaled by the final block coding mask (Fig. 4.10 on page 81) of the current frame are registered into a common coordinate system by using the global transformation parameters. To prevent unnecessary multiple filtering also for such an approach, the coordinate system and resolution of the first frame are used as reference and all subsequent frames are projected into that coordinate system. For different flight altitudes or zoom changes, a new reference frame should be defined and a new mosaic should be initiated to prevent information loss due to scaling. Since different types of ROI like ROI-NA or ROI-MO are not distinguished, ROI-MOs appear static in the mosaic at the position of their last occurrence.



(a) Mosaic generated from the camera-captured video sequence directly without any radial distortion compensation

(b) Mosaic generated from the video sequence with in-loop radial distortion compensation

Figure 4.13: Mosaic consisting of 330 frames of the *350 m sequence* from TAVT [46, 81]. In (a) the mosaic was directly generated from the radial distortion affected camera-captured video, in (b) the in-loop radial distortion compensation has been applied, leading to much better results ((a) scaled to fit line width, (b) scale manually adjusted) [83].

However, for real-world sequences from an uncalibrated camera, especially the radial lens distortion accumulates to severe distortions resulting in entirely distorted mosaics after several ten or a few hundred frames (Fig. 4.13a). Moreover, the effect is amplified, since the new areas are captured from the borders of each frame where the predominant effect of radial lens distortion is most distinct.

To prevent stitching errors induced by radially distorted video frames, the in-loop radial distortion compensation from Section 4.1.2 can be employed, resulting in a fully automatically processed panoramic image like in Fig. 4.13b.

4.5 Video Reconstruction from ROI-Encoded Videos

The entire chapter so far dealt with the efficient *encoding* of an aerial video sequence, the video decoding itself by a standard compliant decoder and the construction of a mosaic out of the decoded ROI areas. The process of video reconstruction of ROI encoded videos using the proposed ROI coding system is described in this section.

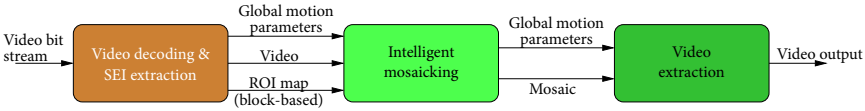


Figure 4.14: ROI decoder for the reconstruction of a video from ROI encoded data (based on [75, 79]).

A simplified block diagram of a ROI decoder is shown in Fig. 4.14. As already described, the video bit stream is decoded by an arbitrary standard compliant video decoder and the block-wise ROI map as well as the global motion parameters are regained from the SEI message. A global motion compensation of ROI blocks of the current frame according to the ROI map is performed. All ROI blocks (ROI-NA and ROI-MO) are treated identically and are projected into the mosaic as described in Section 4.4 (block *intelligent mosaicking* in Fig. 4.14). In the *video extraction* (Fig. 4.14) the current frame from the mosaic is extracted. The correct cutting position is calculated based on the global motion parameters. Since ROI-MO are inserted into the mosaic for each frame and extracted from it directly afterwards, ROI-MO blocks appear at the correct positions in the reconstructed video and their motion is retained.

5 Experiments

This chapter is divided into two main parts: in Section 5.1 the rate-distortion theory for affine motion-compensated prediction from Chapter 3 is investigated. The unquantized prediction error is quantized so that a predefined distortion, e. g. of 30 dB signal-to-noise ratio (SNR), is introduced between the original signal and the quantization error. In this manner, the prediction error caused by inaccurate motion estimation as modeled in Chapter 3 is simulated and the prediction error bit rate model is validated.

Operational rate-distortion diagrams for common video test sequences containing distinct non-translational affine motions are presented and compared to those of real-world aerial and non-aerial video sequences in Section 5.1.2.

In Section 5.2 the ROI coding system from Chapter 4 is employed to efficiently encode aerial video sequences. It is shown that the coding efficiency outperforms the most recent and most efficient video coding standard HEVC in objective means measured as peak signal-to-noise ratio (PSNR) in the regions of interest (ROI-PSNR) as well as in subjective tests for aerial sequences. The latter confirm that the subjectively perceived overall image quality of the ROI coding system is preferred compared to that of common HEVC coding for low and very low bit rates while concurrently more relevant details for surveillance tasks are preserved.

In Section 5.2.3 finally results of the in-loop radial distortion compensation applied for long-term mosaicking of aerial sequences as introduced in Section 4.1.3 are presented.

5.1 Affine Motion Compensation in Video Coding

In this section the model for calculating the minimum required bit rate for encoding the prediction error using affine motion-compensated prediction from Chapter 3 is verified. In Section 5.1.1 the bit rate model from Section 3.1 is validated based on measurements. In Section 5.1.2 operational rate-distortion diagrams from video sequences encoded with the former exploration model JEM of the upcoming video coding standard VVC with and without affine motion-compensated prediction are

presented and the results are discussed. Moreover, video sequences with and without distinct affine—non purely translational—motion are compared.

5.1.1 Efficiency measurements for fully affine motion-compensated prediction in video coding

In Chapter 3 the rate-distortion function for affine motion-compensated prediction was derived and the bit rate for encoding the prediction error was calculated as a function of the motion estimation accuracy. The latter was characterized by the variances of errors of the affine mapping parameters. For visualization of the results in Fig. 3.4 (see page 53) in a 3D plot, the variances of the errors of the translational affine parameters $\sigma^2_{e_{13}}$, $\sigma^2_{e_{23}}$ and the non-translational affine parameters $\sigma^2_{e_{11}}$, $\sigma^2_{e_{12}}$, $\sigma^2_{e_{21}}$, $\sigma^2_{e_{22}}$, respectively, were assumed to be equal. Since the Gaussian distribution has the highest entropy among all distributions with given mean and variance—and Gaussian distributions have been assumed for the distributions of the motion estimation errors—the resulting model bit rate is the supremum of the minimum required bit rate for encoding the prediction error. In other words, for any non-Gaussian distribution, the rate-distortion optimized minimum required bit rate for encoding the prediction error is expected to be smaller than predicted by the model.

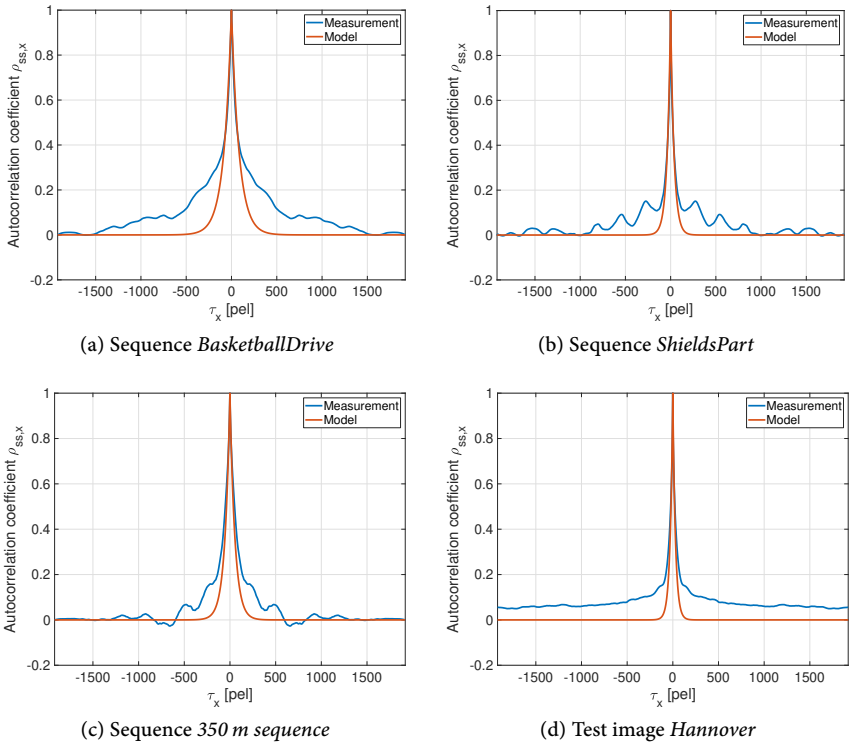
To validate the model introduced in Section 3.1, the 10000×10000 pel aerial image of Hannover (test image *Hannover*, Fig. 5.1, cf. Section 3.1.6.1 on page 46, [63]) was used. The image provides a similar signal characteristic as the other HD test sequences in terms of its autocorrelation coefficients. In Fig. 5.2 the autocorrelation coefficient of *Hannover* is compared to those of a HD resolution JCT-VC test sequence [13], a test sequence which contains high amounts of non-translational affine motions proposed by Li [65], and an aerial test sequence from the TAVT data set [46, 81]. The plots show that the autocorrelation coefficients almost perfectly match the model assumed in Section 3.1.3 for small and medium pel shifts of $|\tau_x| \leq 50$.

A virtual camera has been used to extract several full HD resolution (1920×1080 pel) frames from the large aerial image of Hannover which were concatenated to a video sequence. The frame-to-frame motions comply with an affine motion model (Section 3.1.1). To generate the affine motion for the virtual camera path, (pseudo-) random numbers were drawn from a Gaussian distribution with given means and variances. The means of the affine parameters are selected such that the mean frame-to-frame motion is zero, i. e. the mean value is 1 for the parameters a_{11} and a_{22}, and zero for the remaining parameters a_{12}, a_{13}, a_{21}, a_{23}. The resulting video sequence signal s now has well-known frame-to-frame mappings. Thus, the sequence can be used to measure the bit rates needed for encoding the prediction error. Assuming

(a) Full image, 10000×10000 pel

(b) Cropped image in full HD resolution of 1920×1080 pel

Figure 5.1: Test image *Hannover* [63].



(a) Sequence *BasketballDrive*

(b) Sequence *ShieldsPart*

(c) Sequence *350 m sequence*

(d) Test image *Hannover*

Figure 5.2: Measured autocorrelation coefficients $\rho_{ss,x}$ in horizontal direction of the natural video test sequences *BasketballDrive* (Fig. 5.6a on page 97) [13] and *ShieldsPart* (Figs. 5.5c and 5.5d on page 94) [65] as well as of the aerial test sequence *350m sequence* (Fig. 5.7a on page 97) from the TAVT data set [46, 81] (averaged over 50 frames each) and of the test image *Hannover* (10000×10000 pel) [63] (cropped to same width, Fig. 5.1b). For the model evaluation in the plots (red lines), the correlation coefficient of each specific test sequence or image, respectively, was used. It can be seen that the exponentially decaying model perfectly fits for small to medium shifts up to ±50 pel and that the measured correlations are small (≤ 0.4) for larger shifts.

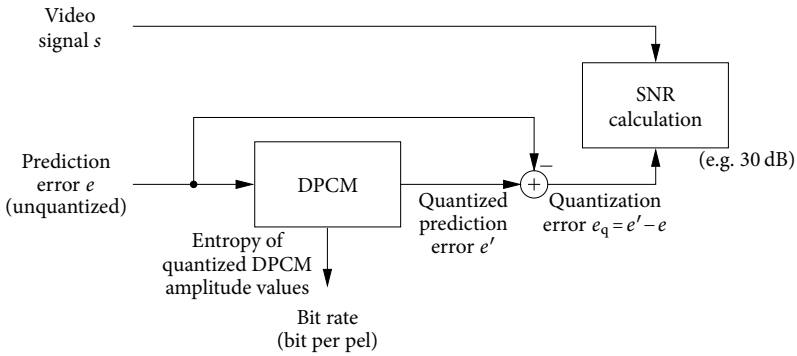


Figure 5.3: Setup for measuring the bit rate for encoding the prediction error. The quantization is adjusted so that a predefined distortion, e. g. 30 dB SNR, is met and the corresponding entropy of the quantized DPCM amplitude values is determined.

the most trivial motion estimation system which always predicts "no motion", the artificially introduced motion becomes exactly the prediction error e, which can be calculated just as the difference between two consecutive frames in the video sequence.

The setup for the measurement is shown in Fig. 5.3. The unquantized prediction error e is decorrelated using a differential pulse-code modulation (DPCM), where only the correlations between horizontally and vertically neighboring pels are exploited for the prediction of the current pel at position (x, y), whereas $\hat{s}_{k_{x,y}} = s_{k_{x,y}} - 0.5 s_{k_{x-1,y}} - 0.5 s_{k_{x,y-1}}$. An uniform quantization is applied afterwards so that the signal-to-noise ratio between the video signal s and the quantization error $e_q = e' - e$ is equal to a predefined value, e. g. of 30 dB as assumed for the model in Section 3.1.6.3. The bit rate is calculated as the entropy of a memoryless source, which corresponds to the bit rate needed for encoding the quantized prediction error, assuming perfectly decorrelated symbols after the DPCM.

The results are shown in Fig. 5.4 for a SNR of 30 dB (like also assumed in Section 3.1) and using 30 frames with different motions for each data point. It is obvious that the measurement qualitatively perfectly matches the theory, but that the measured bit rates are smaller than those of the model (Fig. 3.4 in Section 3.1.6.3 on page 53). Selected rate points are summarized in Table 5.1. For instance, the measured maximum bit rate is 2.507 $^{\text{bit}}/_{\text{sample}}$, for 10^{-5} for the translational variances $\sigma^2_{\dot{e}_{13}}$,

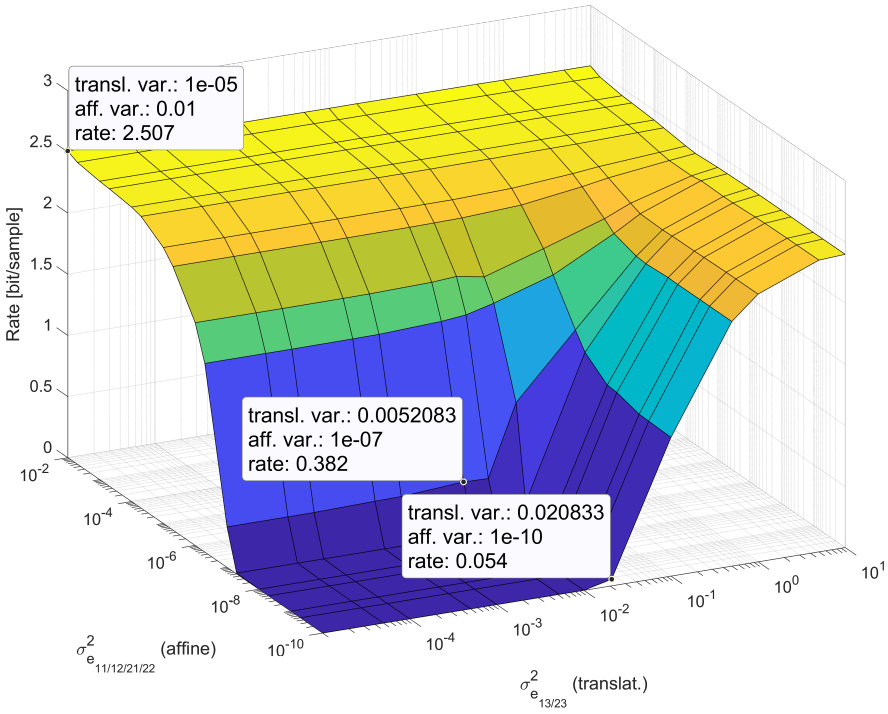


Figure 5.4: Bit rate for encoding the prediction error as a function of the motion estimation error variances for a frame in full HD resolution (1920×1080) using DPCM for signal decorrelation and uniform quantization.

$\sigma^2_{e_{23}}$ and 10^{-2} for the non-translational affine variances $\sigma^2_{e_{11}}$, $\sigma^2_{e_{12}}$, $\sigma^2_{e_{21}}$, $\sigma^2_{e_{22}}$ (upper left data tip in Fig. 5.4), instead of 2.53 $^{\text{bit}}/_{\text{sample}}$ as predicted by the model (upper left data tip in Fig. 3.4a). Accordingly for the translational variances of 0.0052 and the non-translational affine variances of 10^{-7}, the measured bit rate of 0.382 $^{\text{bit}}/_{\text{sample}}$ (central data tip in Fig. 5.4) is lower than the model bit rate of 1.034 $^{\text{bit}}/_{\text{sample}}$ (central data tip in Fig. 3.4a). For very small variances below 0.02 (translational) and 10^{-8} (non-translational), the measured bit rates faster decrease to zero than predicted by the model (dark blue plateau in Fig. 5.4). These differences may mainly be caused by the low-pass filtering character of the Lanczos interpolation filter used during the

Table 5.1: Selected rate points of the fully affine motion model (cf. Fig. 3.4 on page 53) and the measurement (cf. Fig. 5.4).

Translational variances	Affine non-translational variances	Modeled bit rate in $^{\text{bit}}/_{\text{sample}}$	Measured bit rate in $^{\text{bit}}/_{\text{sample}}$
10^{-5}	10^{-2}	2.533	2.507
0.0052	10^{-7}	1.034	0.382

generation of the test sequence *Hannover*. By low-pass filtering, higher frequencies are flattened or entirely removed from the signal, which finally leads to smaller bit rates needed for encoding the prediction error in the measurement compared to the model. The pronounced lower plateau mainly occurs since for very small affine motions introduced during the generation of the test sequence, the affine distorted image perfectly matches the original after interpolation filtering. Thus, the prediction error image as introduced above is nil and consequently no bit rate is needed for encoding it.

Since the model bit rates represent the supremum of the minimum required bit rate for encoding the prediction error, the measurements empirically prove the correctness of the model.

To reveal the operating range of the model for real-world sequences, the two sequences *TractorPart* (1920 × 1080, 25 fps, 8 $^{\text{bit}}/_{\text{sample}}$, chroma subsampling 4:2:0, 100 frames) and *ShieldsPart* (1920 × 1080, 50 fps, 8 $^{\text{bit}}/_{\text{sample}}$, chroma subsampling 4:2:0, 100 frames) (Fig. 5.5) [65] were encoded using JEM 7.1 (SVN revision 603) [55] with the *random access* (RA) profile and the *low-delay p* (LDP) profile [111]. Both sequences have been proposed to demonstrate the efficiency of affine motion-compensated prediction by Li et al. [65] since they contain distinct non-translational affine motion. Taking the average luminance values of both sequences of 94.7 for *TractorPart* and 49.5 for *ShieldsPart* into account, a SNR of 30 dB corresponds to a PSNR of 38.6 dB and 44.2 dB, respectively, on an 8-bit scale. The averaged bit rates over both profiles, only using non-intra coded frames for encoding the sequences at the given PSNR for the luminance component, are 424 $^{\text{kbit}}/_{\text{s}}$ for *TractorPart* and 68599 $^{\text{kbit}}/_{\text{s}}$ for *ShieldsPart*. These bit rates correspond to a mean bit rate of 0.0082 $^{\text{bit}}/_{\text{sample}}$ for the *TractorPart* and 0.66 $^{\text{bit}}/_{\text{sample}}$ for the *ShieldsPart* sequence, respectively. Both bit rates also include signaling, which is neither covered by the model nor considered in the measurement. Extrapolating the findings for AVC from Klomp [60] to HEVC or, more specific, to the

(a) *TractorPart*, frame 1 (b) *TractorPart*, frame 100 (last frame)

(c) *ShieldsPart*, frame 1 (d) *ShieldsPart*, frame 100 (last frame)

Figure 5.5: Test sequences *TractorPart* (25 fps) and *ShieldsPart* (50 fps) containing distinct (non-translational) affine motion [65].

HEVC reference HM-based JEM software, the signaling data may account for half of the bit rate for medium quantization (*TractorPart*, QP 31 for RA and QP 29 for LDP) and for less than 10 percent in the case of fine quantization (*ShieldsPart*, QP 15 for RA and QP 16 for LDP). The high difference in the coding bit rate is caused by the very high amount of blur contained in the *TractorPart* sequence compared to the *ShieldsPart* sequence.

As can be seen in Fig. 5.4, the operating points for the sequences are located in the right area of the lower plateau for *TractorPart* and in the middle of the mid-blue area above the marked point at "transl. var.: 0.0052; aff. var.: 1e-7" (central data tip in Fig. 5.4) for *ShieldsPart*. Hereby, translational quarter-pel resolution like used in the JEM software [64] is assumed, which corresponds to the isoline at the translational variance 0.0052. Using the derivations of Section 3.1, the model bit rate for both sequences is approximately 2.2 $^{\text{bit}}/_{\text{sample}}$ (not shown) for a block size of 128×128 pel as used in JEM with translational quarter-pel resolution and non-translational affine motion vector accuracies of $\frac{1}{16}$ pel, which corresponds to the internal luma resolution

of JEM. Compared to the modeled bit rate, the measured bit rate is magnitudes smaller for the *TractorPart* sequence and still less than half as high for *ShieldsPart*, which had to be expected for the following reasons. First, the assumption of a stationary signal was made in the model, which may not entirely be true for natural (non-aerial) videos, e. g. if the upper part of a frame shows the sky whereas the lower part shows a field like in *TractorPart*. Second, in the example calculations the translational and the non-translational parameter error variances were each assumed to be identical, i. e. $\sigma^2_{e_{13}} = \sigma^2_{e_{23}}$ and $\sigma^2_{e_{11}} = \sigma^2_{e_{12}} = \sigma^2_{e_{21}} = \sigma^2_{e_{22}}$, which may not be fulfilled in case that the error variances are not predominantly caused by artificially quantization of the (simplified) affine parameters. Third, the autocorrelation function of the signal was assumed to be isotropic and exponentially decreasing, which is a good approximation of a video signal although it is not entirely reflecting reality. Especially for larger shifts it was demonstrated in Fig. 5.2 that the exponentially decreasing autocorrelation model tends to overestimate the high frequency components contained in the signal, which increases the bit rate in the model. Moreover, in the model calculations averaged correlations between adjacent pels were assumed to generalize the model. However, for specific sequences these correlations may highly vary, leading to different modeled bit rates (cf. Table 3.3 on page 51). Finally, the displacement estimation error pdf was derived to be Gaussian distributed, induced by the assumed Gaussian distributed affine estimation errors (Section 3.1.2), which leads to the highest entropy compared with other pdfs of the same variance. Thus, it will overestimate the minimum required bit rate for finite real-world signals.

In conclusion, it has been proven that the model provides valuable indications of the prediction error bit rate as a function of the affine motion estimation accuracy. It was verified by measurements that the model qualitatively perfectly predicts the behavior of the prediction error bit rate. Due to several assumptions made in the model which approximate real-world signals, the result obtained by the model can be considered as a supremum for the minimum required bit rate for encoding the prediction error.

5.1.2 Operational rate-distortion diagrams using JEM without and with affine motion-compensated prediction

To evaluate the performance of (simplified) affine motion-compensated prediction in video coding, video sequences with different characteristics are encoded using JEM 7.1 (SVN revision 603) [55] with and without affine motion compensation. From the JCT-VC test set [13], the full HD resolution sequences *BasketballDrive* and *Cactus*, both recorded at 50 fps, were arbitrarily selected to represent natural video content

(Fig. 5.6).

Predominantly planar, high quality, full HD resolution aerial sequences with a preferentially translational global motion, which were recorded at 30 fps, are represented by the *TNT Aerial Video Testset* (TAVT) sequences (set 1) named *350 m sequence*, *500 m sequence*, *1000 m sequence* and *1500 m sequence* (Fig. 5.7) [46, 81]. The names represent the approximate recording height and indicate that for higher flight altitudes the ground resolution decreases, since the camera settings have not been changed for all sequences. Test sequences containing distinct non-translational affine motion are the two sequences *TractorPart* and *ShieldsPart* (Fig. 5.5, see above) [65].

Operational rate-distortion (RD) curves for 500 frames of the test sequences each, except for *TractorPart* and *ShieldsPart* which only consist of 100 frames each, are shown in Fig. 5.8 for the cases of enabled (squares) and disabled (stars) affine motion-compensated prediction. Identically colored curves belong to the same sequences. Two different profiles were tested, *low-delay p* (LDP, Fig. 5.8a) and *random access* (RA, Fig. 5.8b). For reasons of clarity, the operational RD curves for the TAVT sequences are represented by only the *350 m sequence*, since the other sequences behave similarly albeit at other bit rate levels. It is obvious that for the sequence *BasketballDrive* from the JCT-VC test set as well as for the *350 m sequence* (and accordingly the other three TAVT sequences used here but not shown in the graphs) only small gains can be achieved by using affine motion compensation. For the evaluation, Bjøntegaard delta (BD) rates were calculated, which measure the average bit rate difference between two rate-distortion curves [11, 12]. For the TAVT sequences BD rate gains of only 0.88 % (LDP) and 0.54 % (RA) for *BasketballDrive*, and 0.33 % (LDP) and 0.41 % (RA) for the *350 m sequence* were achieved. However, the BD rate gains for the *Cactus* sequence including rotating elements are 6.32 % for LDP and 5.48 % for RA.

For the sequences *TractorPart* and *ShieldsPart* containing a considerable amount of non-translational affine zoom motion, the observed gains even increase to 30.96 % (LDP) and 20.59 % (RA) and to 24.75 % (LDP) and 13.29 % (RA), respectively.

For sequences containing distinct non-translational affine motion, affine motion-compensated prediction may highly increase the coding efficiency of upcoming video coding standards. Especially aerial sequences captured from a drone with high amounts of rotation and zoom may highly benefit from affine motion-compensated prediction.

(a) *BasketballDrive*

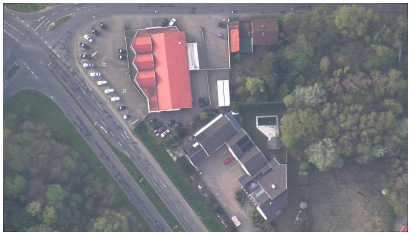
(b) *Cactus*

Figure 5.6: JCT-VC test sequences *BasketballDrive* and *Cactus* (both full HD resolution, 50 fps) [13].

(a) *350 m sequence*, ground resolution: 43 $\frac{\text{pel}}{\text{m}}$

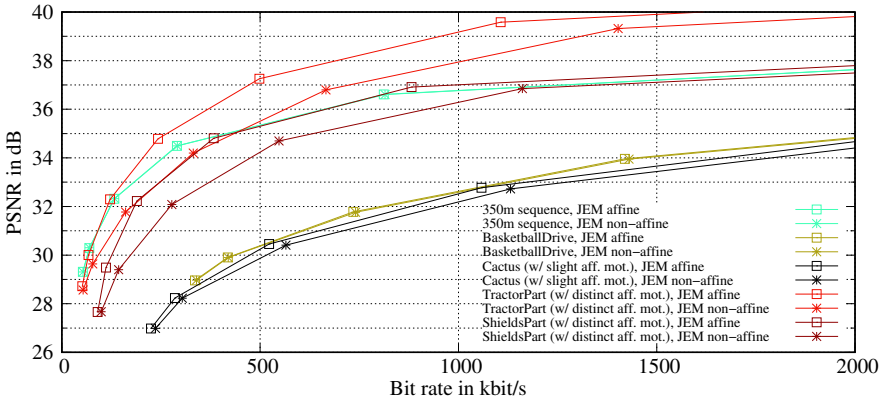
(b) *500 m sequence*, ground resolution: 30 $\frac{\text{pel}}{\text{m}}$

(c) *1000 m sequence*, ground resolution: 15 $\frac{\text{pel}}{\text{m}}$

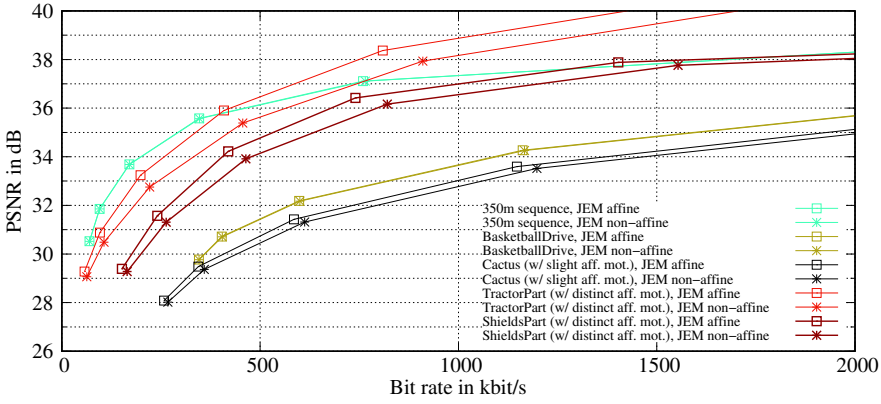


(d) *1500 m sequence*, ground resolution: 10 $\frac{\text{pel}}{\text{m}}$

Figure 5.7: Test sequences from the *TNT Aerial Video Testset* (TAVT) [46, 81].



(a) LDP



(b) RA

Figure 5.8: Operational rate-distortion curves for different sequences encoded by JEM 7.1 (SVN revision 603) [55] with (squares) and without (stars) simplified affine motion compensation. In (a) results for the *low delay p* (LDP) profile are shown, in (b) results for the *random access* (RA) profile are displayed. Sequences containing high amounts of non-translational motion (*TractorParts*, *ShieldsPart*) clearly profit from simplified affine motion compensation whereas sequences without such motions (*350 m sequence*, *BasketballDrive*) do not benefit from the simplified affine motion model.

5.2 Evaluation of the ROI-based System for Low Bit Rate Aerial Video Coding

The ROI coding system introduced in Chapter 4 is evaluated in this section with the focus on its coding efficiency. First, in Section 5.2.1, the PSNR is analyzed for the luminance component in ROI areas (ROI-PSNR), assuming that the global motion compensation introduces only small interpolation artifacts which are negligible compared to the quality impairment caused by the video encoding itself. This holds true especially for high quantization levels, i. e. using very coarse quantization. As encoding framework the general ROI coding system (Section 4.3) is used.

In the second part of this section (Section 5.2.2), the overall image quality is evaluated in subjective tests based on ITU-T Recommendation P.913 [54]. In top-bottom comparisons between a commonly encoded and a ROI encoded sequence at similar bit rates the test subjects were asked to 1. analyze, which video provides more details suitable for surveillance tasks, 2. evaluate, which shows their favorite perceived overall quality and 3. state whether they believe that a better image quality could reveal more helpful details in the context of aerial surveillance.

5.2.1 Objective evaluation of the general ROI-coding system compared to a modified HEVC-encoder and common HEVC coding

Aiming at aerial surveillance tasks, the ROI coding system introduced in Chapter 4 exploits the characteristic that a landscape appears to be planar for medium to high flight altitudes. It achieves a high subjective quality also at extremely low bit rates below 1 $\mathrm{Mbit/s}$, where common video encoders are not able to provide useful image qualities any more. To provide such high subjective qualities at these very low bit rates, global motion compensation is employed to reconstruct any content in the frame which already appeared in one of the preceeding frames. Only new emerging areas in each frame (new areas, ROI-NA) and areas showing locally moving objects like cars (moving objects, ROI-MO) not conforming with the global motion are encoded.

To objectively assess the coding gain of the ROI coding system for aerial sequences of the TAVT data set (Fig. 5.7) [46, 81] the PSNR of the luminance component is measured in ROI areas (ROI-PSNR). It is hereby assumed that errors introduced by global motion compensation and interpolation are small compared to those of the actual video encoder internal quantization at these small bit rates. As a common video encoder without any modifications, the HEVC reference encoder *HEVC Test*

Model (HM) 16.2 [73] was used. To evaluate the performance of the general ROI coding system compared to a specifically adopted encoder, an externally controlled HM video encoder was applied (HEVC-skip) [81]. The latter encoder enforces all non-ROI areas to be encoded in the special HEVC mode "merge/skip" [89]. Furthermore, the video was preprocessed and encoded according to the proposed general ROI coding framework (Section 4.3). An unmodified HM encoder was employed, firstly using mode 1 (copy of the temporally preceding corresponding block in the preprocessing (PP)), called "HEVC-PP (mode 1)", and secondly using mode 2 (replacement of non-ROI blocks by black areas), called "HEVC-PP-black (mode 2)".

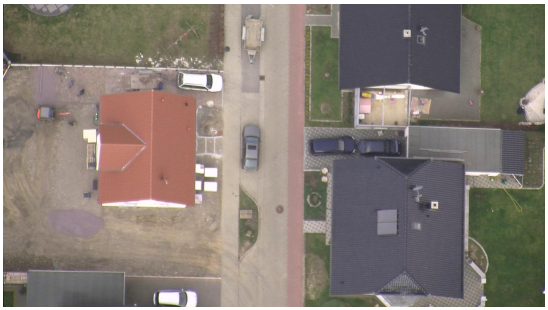
The test setup and evaluation as well as the following description are based on the previous publication [85]. For evaluation, all coding systems were adjusted to match the same PSNR of the luminance channel as given in the results table which represents a subjectively "good quality" for each specific sequence. Only the luminance values within ROI areas (ROI-PSNR) are considered, similar to [38, 81]. Hereby it is assumed that errors in non-ROI areas are irrelevant because non-ROI is reconstructed by external means as part of the postprocessing at the decoder. If it was not possible to exactly match the ROI-PSNR, it was interpolated linearly in between the adjacent rate points, which is justified by a relatively linear curve between neighboring rate points in an operational rate-distortion plot.

Major parts of the frames of the aerial sequences were selected to be non-ROI by the ROI-detectors, depending on the camera movement relative to the ground and the small amount of areas containing moving objects.

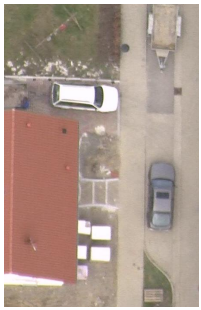
Table 5.2 shows the coding gains (negative numbers) relative to the HEVC reference (indicated by *Ref.*) as marked in the columns. The resulting image quality using the proposed general ROI coding is similar to the one achieved with the specifically adapted encoder HEVC-skip. Compared to common video coding, the subjective quality remains relatively high over the entire frame for all ROI-based systems even at low bit rates, as more bits can be allocated to ROI areas (Fig. 5.9) due to the bit savings in non-ROI regions.

With the HEVC-skip video encoder the bit rate is decreased for the HD resolution sequences by up to 94.8 % and by approximately 90.4 % on average, and to about 600 $^{\text{kbit}}/_{\text{s}}$ for a perceptional "good quality" of about 38 dB. These bit rates are very low compared to the bit rates of the unmodified HEVC codec between 5500–11900 $^{\text{kbit}}/_{\text{s}}$.

Both proposed coding modes of the general ROI coding system provide coding performances similar to the specifically adapted HEVC-skip encoder. The slight drop of coding performance can be compensated by the simplicity of the general ROI approach compared to the complex, time consuming and thus expensive encoder internal modifications for implementing an external mode control. Finally, from the

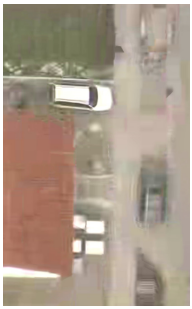


(a) Original frame (whole frame)

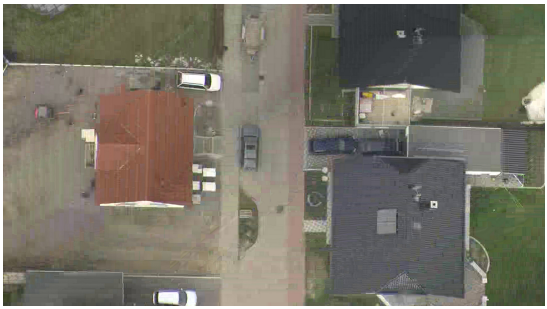


(b) Original, magnif.

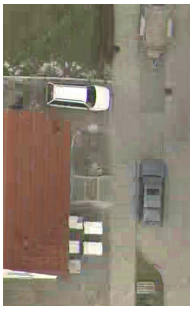
(c) HEVC en- and decoded (whole frame, 126 kbit/s)



(d) HEVC, magnif.



(e) General ROI en- and decoded (whole frame, 117 kbit/s)



(f) General ROI, magnif.

Figure 5.9: Subjective image quality comparison for extremely low bit rates (*350 m sequence* [46, 81]), showing whole frames (left) and corresponding magnifications ("magnif.", right). (a), (b): original frame as recorded; (c), (d): commonly HEVC coded; (e), (f): general ROI coded as proposed.

Table 5.2: Bit rate comparison for similar PSNRs in ROI areas (ROI-PSNRs) as indicated in the columns between *general ROI preprocessing* (PP) with block insertion from previous frames (mode 1) or insertion of black areas (PP-black, mode 2) for non-ROI areas compared to a modified video encoder (HEVC-skip) and an unmodified HEVC encoder. HEVC encoder in every system: HM 16.2, LD profile. Test sequences from the TAVT data set (set 1) [46, 81]. Negative numbers are gains compared to the reference (indicated by *Ref.* in the table, reference ROI-PSNR values corresponding to results of skip-implementation at default settings). General ROI coding provides similar coding performance as the modified video encoder (HEVC-skip) [85].

	350 m sequence, 43 pel/m, 821 frames, ROI-PSNR ≈ 38.9 dB		*500 m sequence*, 30 pel/m, 1121 frames, ROI-PSNR ≈ 37.2 dB		*1000 m sequence*, 15 pel/m, 1166 frames, ROI-PSNR ≈ 37.7 dB		*1500 m sequence*, 10 pel/m, 1571 frames, ROI-PSNR ≈ 37.6 dB	
	Bit rate in kbit/s	Diff. in %	Bit rate in kbit/s	Diff. in %	Bit rate in kbit/s	Diff. in %	Bit rate in kbit/s	Diff. in %
HEVC	5568	*Ref.*	7947	*Ref.*	5849	*Ref.*	11901	*Ref.*
HEVC-skip	558	−90.0	851	−89.3	739	−87.4	614	−94.8
HEVC-PP (mode 1)	644	−88.4	939	−88.2	815	−86.1	686	−94.2
HEVC-PP-black (mode 2)	562	−89.9	875	−89.0	751	−87.2	618	−94.8

results it can be stated that the general ROI coding mode 2 slightly outperforms mode 1 in all cases in the test set, which may be caused by an imperfect sub-partitioning of the coding blocks (called *coding tree units*, CTUs, in HEVC).

In conclusion, the general ROI coding system combines high coding efficiency which outperforms common state-of-the-art video coding standards by far in terms of ROI-PSNR with a simple implementation of the necessary pre- and postprocessing. In the next subsection it will be shown that the ROI-PSNR—although it only considers the PSNR in ROI areas—is a valid indicator for the perceived overall image quality.

5.2.2 Subjective tests

In the previous subsection it has already been shown that the proposed general ROI coding system outperforms the common video coding standard HEVC in terms of ROI-PSNR. Thereby it was assumed that the global motion compensation employed by the ROI coding system does not impair the perceived image quality. This assertion shall be verified in this subsection by subjective tests. The test subjects were asked to

Table 5.3: Bit rates used for subjective tests. Test sequences from the TAVT data set [46, 81]. "Ref.": commonly HEVC encoded, "ROI": general ROI encoded (proposed). The video encoder software in either case was *x265* (as part of the framework *ffmpeg Lavc57.107.100 libx265*) [126, 34].

Test #	*350 m sequence*		*500 m sequence*		*1000 m sequence*		*1500 m sequence*	
	Ref. in kbit/s	ROI in kbit/s	Ref. in kbit/s	ROI in kbit/s	Ref. in kbit/s	ROI in kbit/s	Ref. in kbit/s	ROI in kbit/s
1	126	117	138	126	122	122	105	104
2	210	209	198	189	222	202	205	194
3	333	335	285	278	313	322	293	301
4	395	379	414	399	449	453	481	481
5	548	554	502	505	540	510	663	653
6	754	711	739	718	649	648	957	868
7	1052	1051	1085	1025	940	919	2022	1914
8	1511	1542	1594	1624	2622	2539	2835	2662
9	5282	5131	4471	4433	5243	5156	5296	5359

judge the quality of a commonly HEVC encoded video (reference) versus the quality of a general ROI encoded sequence (proposed; only new areas (ROI-NA) and moving objects (ROI-MO) are encoded in each frame, whereas non-ROI areas are set to black, see Section 4.3) in a top-bottom comparison. Hereby, it was randomly selected whether the reference or the ROI coded sequence was displayed at the top. The other sequence was displayed at the bottom, accordingly. The four aerial sequences from the TAVT data set (Fig. 5.7 on page 97) [46, 81] were used as test sequences.

Nine different bit rate levels from extremely low bit rates of about 100 kbit/s to common moderate operational bit rates for HEVC encoded content of 5000 kbit/s according to Table 5.3 were presented to the test subjects (Fig. 5.10, visually adjusted at the test monitor).

At the lowest bit rate levels it may be hard to recognize, *if* a car is parked near the street or not (Fig. 5.10c, red car at the bottom right, especially in the image of the reference system). For the noisy *1500 m sequence* it may even be hard to recognize houses in the reference (Fig. 5.10g, upper panel), whereas those are recognizable in the ROI coded image (Fig. 5.10g, lower panel). As video encoder for the subjective tests the software *x265* (as part of the framework *ffmpeg Lavc57.107.100 libx265*) [126, 34] was used at medium preset. It provides fast and high performance HEVC encoding potentially suitable for on-board encoding of small and medium drones with limited

(a) *350 m sequence*, worst quality (top: ROI coding, 117 kbit/s; bottom: reference, 126 kbit/s)

(b) *350 m sequence*, best quality (top: reference, 5282 kbit/s; bottom: ROI coding, 5131 kbit/s)

Figure 5.10: Subjective test images [46, 81].

(c) *500 m sequence*, worst quality (top: reference, 138 kbit/s; bottom: ROI coding, 126 kbit/s)

(d) *500 m sequence*, best quality (top: reference, 4471 kbit/s; bottom: ROI coding, 4433 kbit/s)

Figure 5.10: Subjective test images (continued).

(e) *1000 m sequence*, worst quality (top: ROI coding, 122 kbit/s; bottom: reference, 122 kbit/s)

(f) *1000 m sequence*, best quality (top: ROI coding, 5156 kbit/s; bottom: reference, 5243 kbit/s)

Figure 5.10: Subjective test images (continued).

(g) *1500 m sequence*, worst quality (top: reference, 105 kbit/s; bottom: ROI coding, 104 kbit/s)

(h) *1500 m sequence*, best quality (top: ROI coding, 5359 kbit/s; bottom: reference, 5296 kbit/s)

Figure 5.10: Subjective test images (continued).

Figure 5.11: Subjective test setup.

energy and computational power.

Since the details of a complex suburban scene like shown in the test sequences may easily overwhelm a test subject, randomly drawn still images from each sequence were presented instead of the videos. Each subject was given the same frames and was allowed to view the images for an arbitrary time to account for different grades of observation skills.

The subjective tests were performed as *comparison category rating* (CCR) at lab conditions (Fig. 5.11) according to ITU-T Rec. P.913 [54]. Comparison category ratings were favored over absolute category ratings since for low and very low bit rates the absolute ratings may not indicate significant differences between the reference and the proposed system as the image quality is impaired in any case. The test frames were cropped to exactly fit the test display, a Dell U2713HM 27" LED-LCD with a resolution of 2560×1440 pel, to avoid scaling and interpolation artifacts. The test subjects were requested to adjust their chairs to provide a centered, perpendicular view of the screen to minimize viewing angle dependent effects.

In the course of the test, the subjects were asked whether the top or bottom image provides more sharp details, with a special focus on aerial surveillance tasks. For example it may be of interest if cars are recognizable, if persons are recognizable, if details from cars (presence or absence of the side-mirror, dents in cars etc.) or persons (e. g. color of jacket) can be distinguished. Moreover, the subjects should state which overall subjective impression they preferred. Finally, they were asked to judge whether they expect to recognize more relevant details if the quality was improved for both coding systems separately. The answers had to be entered in a *graphical user interface* (GUI, as illustrated in Fig. 5.12) designed with Matlab 2018b

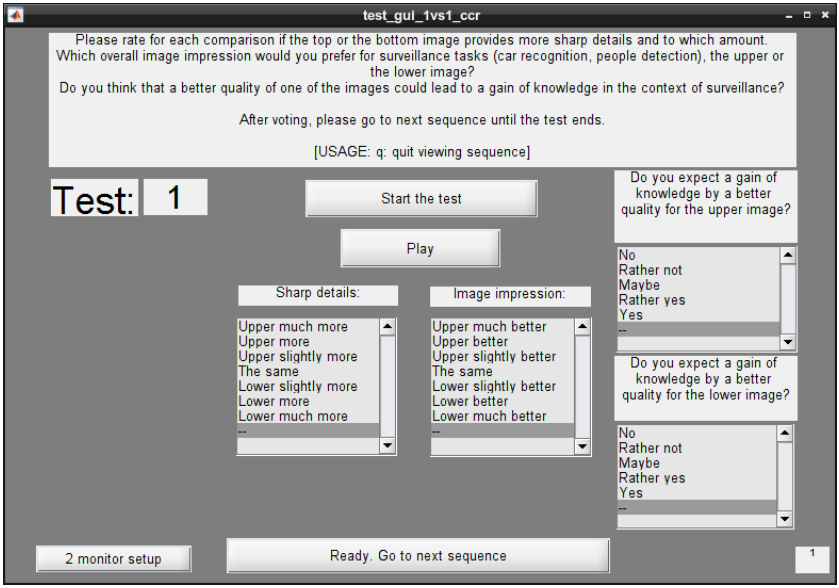


Figure 5.12: Graphical user interface (GUI) for subjective tests using *comparison category rating* (CCR). Experimental setup and quality criterion according to ITU-T Rec. P.913 [54]. In each test a top-bottom comparison was presented to the test subjects. The reference image or the ROI coded image (proposed) was presented at the top by random choice. The other sequence was presented at the bottom, respectively. Similar bit rates were used in each comparison and the sequences were displayed with increasing bit rates for each sequence (HEVC encoder: *x265* as part of the framework *ffmpeg Lavc57.107.100 libx265* [126, 34]).

[119]. The subjects had to answer the questions for the sharpness and overall image impression based on a seven-grade Likert scale ("much better", "better", "slightly better", "same", "slightly worse", "worse", "much worse") according to ITU-T Rec. P.913 [54]. Moreover, the subjects were requested to state whether they expect to gain any more knowledge relevant for surveillance tasks on an adapted five-grade scale ("no", "rather not", "maybe", "rather yes", "yes") for the reference and the ROI coding system separately. The answers to the latter questions will indicate, which minimum bit rate is required to successfully fulfill surveillance tasks.

The test was performed with 27 test subjects and the average duration was 50 min-

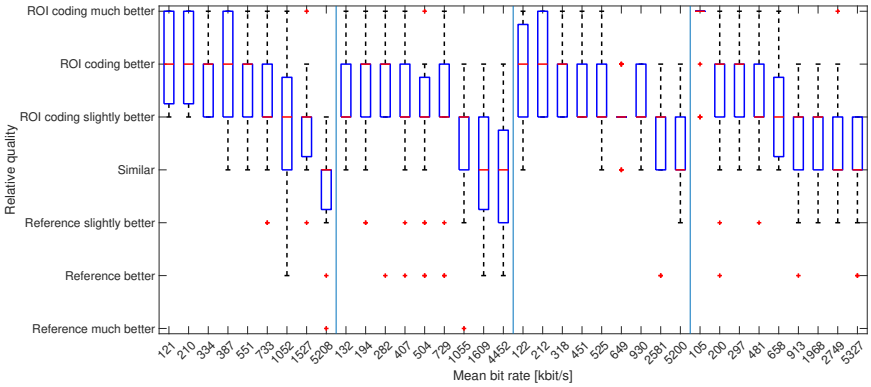
utes, which equals about 1.5 minutes per test image. This indicates that the subjects intensely observed the sequences with regard to details.

The results of the subjective tests are summarized in Figures 5.13 and 5.14 in box plot-like graphs. Technically, the answer items of the Likert scale may not be assigned to numerical values since they only represent relative differences and no absolute quality gradings like in absolute category rating scales (e. g. *mean opinion score* or MOS scale). However, the box plot-like graphs provide a robust estimate of the distribution of the answers for the purpose of comparison.

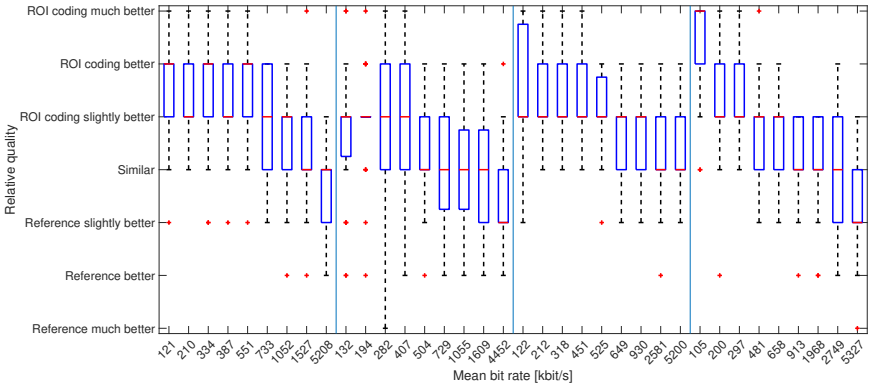
The Mathworks online description for box plots and particularly the Matlab documentation of the `boxplot` command [72] state: "On each box, the central mark indicates the median, and the bottom and top edges of the box indicate the 25^{th} and 75^{th} percentiles, respectively. The whiskers extend to the most extreme data points not considered outliers, and the outliers are plotted individually using the '+' symbol", and "observations beyond the whisker length are marked as outliers. By default, an outlier is a value that is more than 1.5 times the interquartile range away from the top or bottom of the box." Two similar bit rates, given in Table 5.3 were averaged and displayed on the abscissa of Figures 5.13 and 5.14. The four compartments in the box plot-like graphs show results for the four test sequences (Fig. 5.7) in ascending numerical order, starting with the *350 m sequence* on the far left to the *500 m sequence*, the *1000 m sequence* and finally the *1500 m sequence* right-lateral. The order of presentation is equal to the order in the result graphs. Before each test, the worst and the best quality was shortly presented to the test subjects using an example not included in the test images.

From the results in Fig. 5.13a it becomes obvious that for very low to medium low bit rates between 105 and 1609 kbit/s the proposed ROI system is clearly able to retain more sharpness compared to the reference system. For higher bit rates, both systems provide sufficient sharpness of the images resulting in similar median values reflecting comparable subjective ratings. In contrast to the other sequences, for the lowest bit rate of the *500 m sequence* (132 kbit/s) the image quality is comparably low for both systems. Therefore, the test subjects rated the ROI system as "slightly better" only (Fig. 5.10d). The lower quality compared to the other sequences at similar bit rates may be caused by the highest amount of details in this scene. For the next higher bit rate of 194 kbit/s, however, the ROI system can provide an obviously better image quality than the reference system based on the test results.

A similar trend for the overall personal impression (Fig. 5.13b) can be seen in terms of sharpness. At very low and low bit rates the proposed ROI coding system clearly outperforms the reference. At higher bit rates, ROI coding induced artifacts—for instance visible in Fig. 5.10d at the house roof on the right—are more disturbing



(a) Sharpness of details



(b) Overall personal impression

Figure 5.13: Sharpness of details and overall personal impression. The red central marks represent the medians, the bottom and top edges of the blue boxes indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points not considered outliers, and the outliers are plotted individually using the red '+' symbol [72]. Images from the reference and the ROI system were randomly shown either in the upper or the lower panel for testing.

than a slightly smoothed overall visual impression, exceptionally pronounced for example in the noisy sequence *1500 m sequence* (Fig. 5.10h). Since the *350 m sequence*

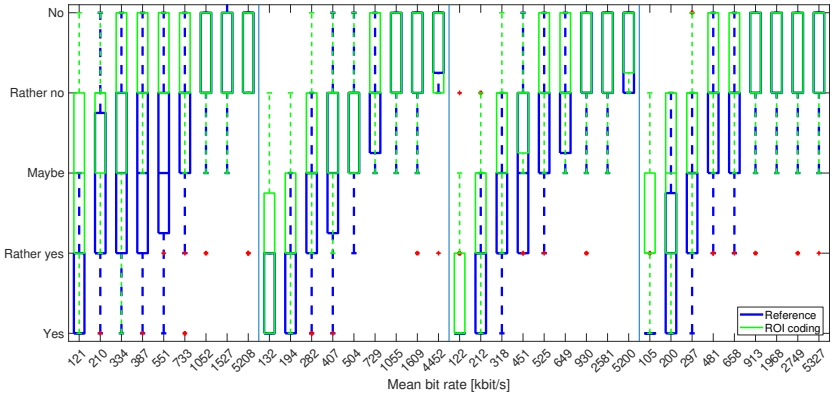


Figure 5.14: Expectations of the test subjects whether an improved image quality provides more information relevant for surveillance tasks using the reference (blue) or the ROI system (green). The central marks represent the medians, the bottom and top edges of the boxes indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points not considered outliers, and the outliers are plotted individually using the red '+' symbol [72]. Images from the reference and the ROI system were randomly shown either in the upper or the lower panel for testing.

was presented to the test subjects first, the results tend to be more volatile due to the adjustment of the subjects to the different quality levels. Moreover, the randomly drawn test image of the *350 m sequence* contains a relatively low amount of details. Thus, relatively high quality can be achieved by both, the reference as well as the ROI coding system. For the sequences *350 m sequence* and *1000 m sequence* not containing severe ROI coding artifacts, the overall impression was assessed equally for higher bit rates. For the *500 m sequence* and the *1500 m sequence*, ROI coding artifacts impair the overall image impression for the highest bit rates, which is the reason why the reference system was preferred. The most disturbing artifacts can be seen in Fig. 5.10d at the house roof in the middle right and the parked cars in Fig. 5.10h.

The answers of the subjects whether they expect to gain any further knowledge useful for surveillance tasks by an improved image quality, are visualized in Fig. 5.14. Herein, blue markers and lines represent the answers for the reference system while the answers for the ROI coding system are shown in green. It is obvious again that the ROI coding system outperforms the reference system especially for very low bit rates, which are always associated with a preference of the ROI system. The maximum

achievable quality for managing surveillance tasks like object recognition and detail detection in the monitored scene is reached at considerably lower bit rates compared to the reference system. Already at bit rates starting from 210 kbit/s for the *350 m sequence* (*500 m sequence*: 407 kbit/s; *1000 m sequence*: 318 kbit/s; *1500 m sequence*: 297 kbit/s) most test subjects stated they rather do not expect to gain any more relevant details for surveillance tasks by an increased image quality using the ROI coding system. In contrast to that, for the same statement considering the reference system, bit rates of 733, 504, 649 and 481 kbit/s were required for the four test sequences in ascending order.

5.2.3 Long-term mosaicking

As introduced in Section 4.1.2, one underlying assumption of a frame-to-frame-based motion estimation in the proposed ROI coding system is that the video frames are only marginally affected by radial lens distortion. In order to enable the ROI coding system to also deal with videos containing non-negligible radial distortion, an efficient in-loop radial distortion compensation (RDC) was proposed in Section 4.1.3. It is based on the idea that the radial distortion does not or only slowly change over time. Thus, one constant but unknown radial distortion parameter κ_1 is iteratively estimated within the motion estimation process for each group of frames (GOF, cf. Figs. 4.3 and 4.4 on pages 72 and 74, respectively). The following description and results have been previously published in [83].

To demonstrate the effectiveness of the proposed algorithm, the unprocessed, non radial distortion compensated, test sequences from the TAVT data set [46, 81] were employed again. The empirically optimized number of iterations is $i_{\mathrm{RDC}} = 14$ and the size of one GOF $n_{\mathrm{RDC}} = 60$. The summarized changes of shape and size between the first and the last frame of a GOF were limited to $c_{\mathrm{shape,max}} = 10\,\%$ and $c_{\mathrm{size,max}} = 20\,\%$, respectively. The proposed radial distortion compensation leads to an increased PSNR of the global motion compensation of 0.27 dB on average (Table 5.4). The automatically generated mosaic of the *350 m sequence* was manually matched with Google Earth [37]. A drift of only 4 pel per 909 pel was achieved, corresponding to 0.0044 pel/frame or about 1 m per 230 m, respectively.

Whereas it is impossible to generate a panorama image from uncorrected content as recorded by a camera, the in-loop radial distortion compensation is able to mosaic the same sequence fully automatically (Fig. 4.13 on page 85). On an Intel Xeon CPU E5-2670 with 2.60 GHz, the unoptimized C/C++ algorithm runs for the first GOF ($l = 1$) of a sequence with about 1000 ms/frame due to the initial estimation of a radial distortion parameter $\kappa_{1,l=1}$. This time can be decreased by providing a good initialization for $\kappa_{1,l=1}$, by reducing the numbers of frames n_{RDC} in one GOF, or by

Table 5.4: Gains of global motion-compensated prediction by the proposed radial distortion compensation (RDC) in dB PSNR.

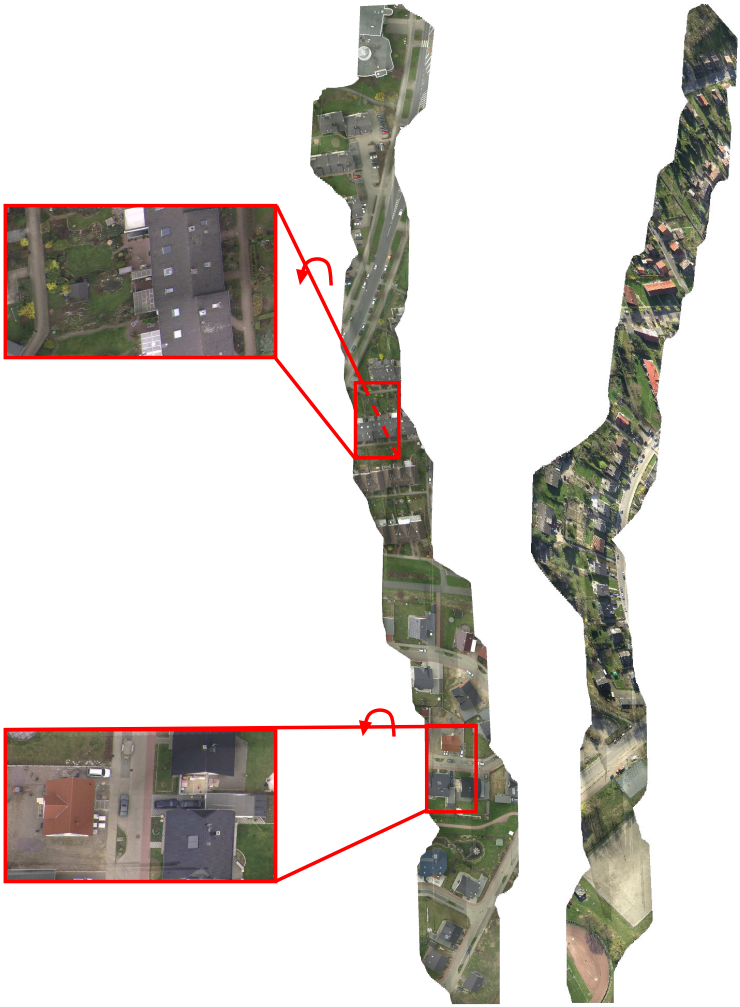
Sequence name (with reference to the recording flight altitude)	GMC without RDC in dB	GMC with RDC in dB	Gain of RDC in dB
350 m sequence	35.68	35.94	0.26
500 m sequence	31.96	32.21	0.25
1000 m sequence	34.17	34.48	0.31
1500 m sequence	32.17	32.43	0.26
Average	**33.50**	**33.77**	**0.27**

limiting the maximum allowed number of iterations i_{RDC}; albeit the latter may reduce the accuracy of the final κ_1. For any other GOF, the average run-time per frame is about 200 $^{\mathrm{ms}}/_{\mathrm{frame}}$, depending on the number of iterations necessary for fulfilling the optimization criteria.

Entire panoramic images for the test sequences are presented in Fig. 5.15.

In this subsection, the proposed iterative in-loop radial distortion compensation has been applied and its effectiveness empirically proven for real-world sequences. It is capable of compensating radial lens distortion without any prior knowledge of the latter. The main advantage of the proposed approach is that it does neither need orthorectified, nor georeferenced video sequences as input. It only relies on the quasi-planar ground of the video sequence recorded at sufficiently high flight altitudes.

In conclusion, in this section the capabilities of the ROI coding system from Chapter 4 have been investigated with respect to a common HEVC coding system as reference. In an objective evaluation, the PSNR measured in ROI areas indicated that the ROI system can provide a comparable image quality at only 10 % of the HEVC bit rate. This implies that the global motion compensation of the ROI coding system does introduce no or only marginal artifacts into non-ROI regions. To confirm this assumption, subjective tests were performed. They successfully verified that especially for very low and low bit rates of less than 1 $^{\mathrm{Mbit}}/_{\mathrm{s}}$ for full HD resolution sequences, recorded e. g. at 30 fps, the ROI coding system outperforms the reference system in terms of detail sharpness and overall image impression. Moreover, the ROI coding system reaches its maximum performance for surveillance tasks at considerably lower bit rates than the common HEVC reference system. Finally, the automatic radial distortion compensation was employed to generate long-term aerial mosaics out of more than 1500 frames of real-world aerial sequences recorded at different flight altitudes.



(a) *350 m sequence*, 821 frames with magnifications (b) *500 m sequence*, 1121 frames

Figure 5.15: Panoramic images of non-preprocessed, full HD resolution sequences from TAVT [46, 81], rotated and scaled to fit page height. The full resolution as captured by the camera is retained as can be seen in the magnifications.

(c) *1000 m sequence*, 1166 frames (d) *1500 m sequence*, 1571 frames

Figure 5.15: Panoramic images of non-preprocessed, full HD resolution sequences from TAVT [46, 81], rotated and scaled to fit page height (continued). The full resolution as captured by the camera is retained.

6 Summary and Conclusions

Motion-compensated prediction is one key element in modern hybrid video coding. With upcoming new application scenarios like the encoding of aerial surveillance videos captured from *unmanned aerial vechicles* (UAVs), new challenges occur. Whereas in common natural videos a translational motion model is sufficient to describe most of the motion contained in the videos, in aerial surveillance videos a high amount of non-translational motion like scaling (zoom) and rotation is present. Such motion types can be described by an affine motion model. Thus, as a first contribution of this work, affine motion-compensated prediction in video coding was analyzed. The minimum bit rate required for encoding the prediction error was derived for a fully affine motion model with six degrees of freedom as well as for a simplified affine motion model with only four degrees of freedom. The latter is currently investigated by JVET within the scope of the standardization activities of a video coding standard succeeding HEVC. In the context of aerial surveillance, the derived minimum bit rates for providing reasonable image qualities for surveillance tasks with limited bandwidth may still be too high. Thus, as a second contribution, a codec-independent *region of interest*- (ROI-) based video coding system for aerial surveillance videos was proposed, which exploits the characteristic of predominant planarity of the observed areas in aerial sequences. Already known background is reconstructed by means of affine global motion compensation. In order to also retain local motion on the ground, moving objects are detected and additionally encoded. As a third contribution, a long-term mosaicking approach was proposed. It is capable of compensating radial lens distortion without any prior knowledge about the camera. Within the global motion estimation, one radial distortion compensation parameter for a given group of frames is jointly estimated so that the projections of the video frames in the mosaic fulfill predefined geometric restrictions.

As first contribution, a model for affine motion-compensated prediction in video coding has been derived in Chapter 3. The derivations were performed with a special focus on a simplified affine motion model with four degrees of freedom, especially because such a model is investigated in the recent standardization activities from JVET in the context of a video coding standard succeeding HEVC. It is capable to describe

scaling, rotation and translation and thus covers typical motion types contained in real-world aerial video sequences. By using the rate-distortion theory, the minimum required bit rate for encoding the prediction error as a function of the motion estimation accuracy has been modeled (cf. Fig. 3.1 on page 39 for a comprehensive flowchart of the model derivation). To achieve this, the four parameters of the simplified affine motion model were assumed to be affected by statistically independent estimation errors with probability density functions (pdfs) following zero-mean Gaussian distributions. Due to the Gaussian assumption, the affine transform parameter estimation errors are entirely characterized by their variances. From the joint probability density function of these parameter estimation errors, the pdf of the displacement estimation error in the image was derived by applying the transform theorem for pdfs. In contrast to previously existing models, e. g. for purely translational motion-compensated prediction, the displacement estimation error is location-dependent in the case of any affine motion-compensated prediction. The pdf of the displacement estimation error is Gaussian distributed as well and is a function of the affine parameter estimation errors. By combining the Fourier transform of the pdf of the displacement estimation error and the modeled power spectral density (PSD) of aerial videos, the PSD of the prediction error is derived. Applying the rate-distortion theory results in the minimum required bit rate for encoding the prediction error for a given signal-to-noise ratio (SNR). Due to the Gaussian distribution assumptions and since the Gaussian distribution has the highest entropy among all distributions with same mean and variance, the supremum of the minimum bit rate required for encoding the prediction error was finally obtained.

Furthermore, the model error was determined which occurs if a translational motion model is used for a sequence containing motions, which can only be described by an affine or higher-order motion model, i. e. rotation, scaling and shearing ("affinities") in addition to translation. The results show that both, the affine parameter estimation errors as well as the affinities inherently contained in a sequence, can be mathematically modeled in the same way. From this finding, two different conclusions can be drawn: first, if the parameter estimation errors are considered, the derived bit rate depends on the estimation process only, i. e. is a characteristic of the specific affine estimator. In case of any additional quantization of the affine parameters, e. g. due to an efficient parameter encoding, these additionally introduced quantization errors may be modeled as estimation errors as well. For affine motion-compensated prediction to be more efficient than simple intra coding of a HD resolution video signal itself, a bit rate of less than 2.0 $^{\text{bit}}/_{\text{sample}}$ must be provided for a SNR of 30 dB. Using the example of a simplified affine motion model with a

block size of 64×64 pel and a translational quarter-pel accuracy, this can only be achieved for an affine motion estimation accuracy of $\sigma^2_{\dot{e}_a} = 3 \cdot 10^{-4}$ or smaller. Such estimation accuracies can easily be achieved with real-world implementations. For instance, the implementation used in this work provides estimation error variances of about $5 \cdot 10^{-10}$ for non-translational affine parameters. Second, in the case that non-translational (simplified) affine motion types are contained in a sequence and nevertheless a purely translational motion model is used (e. g. as in the current video coding standard HEVC), the minimum bit rate for encoding the prediction error is mainly induced by the model violations. As an example, for the full HD resolution test sequences *TractorPart* and *ShieldsPart* containing large zoom drives (scaling), Bjøntegaard delta rate gains of up to 31 % and 25 %, respectively, can be achieved by using the JEM software for non-intra coded frames and a simplified affine motion model instead of a purely translational one.

Similar to the simplified affine motion model a fully affine model with six degrees of freedom was investigated. Compared to the simplified affine motion model which already covers scaling (zoom), rotation, as well as translation—and thus most of the motion types contained in aerial and presumable also general video sequences—it can additionally describe shearing. The derivations for the fully affine motion model result in a different probability density function of the displacement estimation error and consequently in a different PSD of the prediction error. Comparing the results from the fully affine motion model with those from the simplified one, it can be found that for typical affine transformation parameter estimation error variances the bit rate difference is negligible. Otherwise, assuming that in a video sequence affine motion is contained which cannot be described by a simplified affine model, i. e. shearing, the fully affine motion model may provide significant gains in terms of coding efficiency. However, since the vast majority of motions contained in real-world sequences can already be described by the simplified affine motion model, no additional gain can be expected from a motion model additionally covering motions of very rare occurrence. Moreover, from a coding point of view, it is typically beneficial to encode as few parameters as possible.

The derived model of the minimum required bit rate for encoding the prediction error as a function of the motion estimation accuracy has been experimentally verified in Section 5.1. Due to several assumptions in order to approximate the real world and since the supremum of the minimum prediction error bit rate is modeled, the absolute measured bit rates are below the modeled bit rates. However, in conclusion, the model provides valuable information about the minimum required motion estimation accuracy to enable a predefined bit rate for encoding the prediction

error and to design upcoming video coding standards in terms of minimum required affine motion parameter accuracy. Moreover, the comparison between the fully and the simplified affine motion model justifies the selection of the latter for the upcoming video coding standard succeeding HEVC, presumably to be named *Versatile Video Coding* (VVC).

Taking even higher camera resolutions (4K and above) and multiple camera setups on-board an UAV for aerial surveillance tasks into account, the demand for a coding system which provides a coding efficiency exceeding existing and upcoming video coding standards becomes obvious. Consequently, the second contribution of this thesis is a codec-independent ROI-based coding system for the efficient encoding of aerial video sequences as proposed in Chapter 4. This coding system especially takes into account the non-translational, e. g. affine, motion contained in aerial sequences. For the ROI detection and encoding system, it is assumed that the surface of the earth appears mainly planar in the video frames. This is approximately true for medium and high flight altitudes and a camera facing perpendicularly downwards. Hereby the prevalent motion in the scene is induced by the global motion of the camera. For such sequences it is sufficient to only encode new emerging areas and to reconstruct the remaining areas of each frame by means of affine or even higher-order global motion compensation. This results in coding efficiency gains compared to standardized video codecs since the noise contained in the sequence has to be encoded only once in the new areas. Perspective changes in the video caused by 3D objects not matching the assumption of planarity are neglected in favor of a reduced bit rate. The reconstruction of such a ROI encoded video is realized by registering any new area in a mosaic and extracting video frames at appropriate positions to recover the aerial video sequence.

Locally moving objects like moving cars or persons are detected and encoded as additional regions of interest (ROI-MO) since moving objects are typically of high interest for surveillance tasks. Two different moving object detectors are proposed in this work. The primary one aims at a high computational efficiency. It relies on a simple yet effective pel-wise difference detection between the motion-compensated predicted and the current video frame. Spots of high energy are considered as moving objects and consequently treated as ROI-MO. The second proposed moving object detector combines an independent superpixel segmentation of the input video frames with the difference image-based moving object detector described above in order to obtain more accurate moving object boundaries. To avoid mis-detections, e. g. due to non-planar ground structures like trees or houses violating the assumption of

planarity, a mesh-based local motion compensation is incorporated into the system. The latter moving object detector provides highly increased pel-wise detection rates. However, the computational complexity may exceed the computational resources of small and medium UAVs. Since the focus of this work is on video encoding rather than accurate object detection, the first moving object detector was used in the coding system. It should be noted that without any structural changes of the coding system any arbitrary moving object detector may be used instead.

For the video encoding itself, a *general ROI coding* approach is proposed to enable the application of any off-the-shelf video encoder. It is based on the idea that non-ROI areas are compensated at the decoder anyway, and almost no bits should be spent for the encoding of these areas. In a preprocessing step prior to the actual video encoding, every non-ROI block in each frame is replaced by a black area. As a result, the video encoder itself determines the application of one of its most efficient coding modes, e. g. DC mode or skip mode in the case of HEVC.

In the experimental results in Section 5.2, the performance of the proposed general ROI coding system was evaluated. Four HD resolution aerial test sequences from the TAVT data set have been used which were recorded at 30 fps. In objective measurements, the bit rate was reduced by up to 95 % at maximum and 90 % on average compared to the HEVC reference encoder HM. Total bit rates of 562–875 $^{\text{kbit}}/_{\text{s}}$ were achieved for PSNR values in ROI areas between 37.2 and 38.9 dB. To verify that the overall image quality remains subjectively high, subjective tests have been performed. The test subjects clearly stated that for very low bit rates between 100 $^{\text{kbit}}/_{\text{s}}$ and at least 1600 $^{\text{kbit}}/_{\text{s}}$ (full HD resolution, 30 fps) the ROI coding system preserves more sharp details while concurrently a more pleasant overall image impression is provided. For most test sequences, the ROI system was preferred also for higher bit rates up to about 4500 $^{\text{kbit}}/_{\text{s}}$. Finally, the test subjects decided for all test sequences that the ROI coding system already provided all details relevant for surveillance tasks at considerably lower bit rates compared to the HEVC reference system.

As mentioned above, the proposed ROI coding system inherently relies on the generation of a mosaic out of new areas for the reconstruction of video frames at the decoder-side. For the reconstructed frame to be consistent, it is highly important that the processed video is not affected by significant radial lens distortion. Otherwise the global motion estimation becomes inaccurate, which results in discontinuities within the reconstructed frames. To enable ROI processing also for aerial videos, captured from a non-calibrated, unknown camera, an in-loop radial distortion compensation was proposed as a third contribution of this thesis. In contrast to other approaches, the

proposed algorithm does not aim at a highly accurate radial distortion compensation. Instead, it aims at minimizing the distortions of the projected frames in the mosaic. This means, for every group of frames (GOF), a suitable radial distortion compensation parameter is iteratively estimated jointly with the frame-to-frame homographies, in order to preserve similar geometric properties like shape and size of the frames within one GOF. Using this approach, the quality of the global motion compensation was increased by 0.27 dB PSNR on average. Since lens distortion induced inaccuracies are reduced, the fully automatic generation of long-term mosaics consisting of more than 1500 frames is enabled for videos captured by an unknown camera and without any manual preprocessing or adjustment. Those overview mosaics may provide additional contextual information for surveillance tasks without any extra efforts and expenses.

Outlook

In this work, a model to determine the minimum required bit rate for encoding the prediction error of affine motion-compensated prediction in video coding has been presented. Such a model may especially be valuable for the design of upcoming video coding standards or systems employing affine motion-compensated prediction.

Taking cameras not pointing vertically downwards into account, the affine transformation model might be extended towards a projective one with eight degrees of freedom. This would also enable to manage for instance trapezoid distortions.

A Appendix

A.1 Derivation of the Probability Density Function of the Displacement Estimation Error for Fully Affine Motion-Compensated Prediction

The probability density function of the displacement estimation error for fully affine motion compensation in video coding can be expressed as given in (3.11) (page 43) by

$$
\begin{aligned}
p_{\Delta x'}\left(\Delta x' | x, y\right) &= \int_{\mathbb{R}^2} p_{E_{11},E_{12},E_{13}}\left(e_{11}, e_{12}, \Delta x' - x e_{11} - y' e_{12}\right) \mathrm{d}e_{11} \mathrm{d}e_{12} \\
&= \underbrace{\frac{1}{\sqrt{2\pi\sigma_{e_{11}}^2}} \cdot \frac{1}{\sqrt{2\pi\sigma_{e_{12}}^2}} \cdot \frac{1}{\sqrt{2\pi\sigma_{e_{13}}^2}}}_{A} \\
&\quad \cdot \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp\left(-\frac{e_{11}^2}{2\sigma_{e_{11}}^2}\right) \cdot \exp\left(-\frac{e_{12}^2}{2\sigma_{e_{12}}^2}\right) \\
&\quad \cdot \exp\left(-\frac{\left(\Delta x' - x e_{11} - y e_{12}\right)^2}{2\sigma_{e_{13}}^2}\right) \mathrm{d}e_{11} \mathrm{d}e_{12} \\
&= A \cdot \iint_{-\infty}^{\infty} \exp\Bigg(-\frac{1}{2\sigma_{e_{11}}^2 \sigma_{e_{12}}^2 \sigma_{e_{13}}^2} \\
&\quad \cdot \Bigg[\underbrace{\sigma_{e_{12}}^2 \sigma_{e_{13}}^2}_{C} e_{11}^2 + \underbrace{\sigma_{e_{11}}^2 \sigma_{e_{13}}^2}_{D} e_{12}^2 \\
&\quad + \underbrace{\sigma_{e_{11}}^2 \sigma_{e_{12}}^2}_{E} \left(\Delta x' - x e_{11} - y e_{12}\right)^2 \Bigg] \Bigg) \mathrm{d}e_{11} \mathrm{d}e_{12} \,,
\end{aligned}
\tag{A.1}
$$

with $A = \frac{1}{\sqrt{2\pi\sigma_{e_{11}}^2}} \cdot \frac{1}{\sqrt{2\pi\sigma_{e_{12}}^2}} \cdot \frac{1}{\sqrt{2\pi\sigma_{e_{13}}^2}}$ (see (3.11) in Section 3.1.2 on page 43).

Note that capital letters in these derivations are only shorthand symbols whereby their

meaning is restricted to the current section. With $B = 2(\sigma_{e_{11}}^2 \sigma_{e_{12}}^2 \sigma_{e_{13}}^2)$ (A.1) yields

$$p_{\Delta X'}\left(\Delta x' | x, y\right) = A \cdot \iint_{-\infty}^{\infty} \exp\left(\underbrace{-\frac{1}{B} \cdot \left(C e_{11}^2 + D e_{12}^2 + E\left(\Delta x' - x e_{11} - y e_{12}\right)^2 \right)}_{M} \right) \mathrm{d}e_{11} \mathrm{d}e_{12} \,. \tag{A.2}$$

Resorting the coefficients of M leads to

$$\begin{aligned}
M &= -\frac{1}{B}\Big[C e_{11}^2 + D e_{12}^2 + E\Delta x'^2 + E x^2 e_{11}^2 + E y^2 e_{12}^2 - 2E\Delta x' x e_{11} - 2E\Delta x' y e_{12} + 2E x y e_{11} e_{12} \Big] \\
&= -\frac{1}{B}\Big[e_{11}^2 (C + E x^2) + e_{11}(2xy e_{12} E - 2E\Delta x' x) + D e_{12}^2 + E\Delta x'^2 + E y^2 e_{12}^2 - 2E\Delta x' y e_{12} \Big] \\
&= -\frac{1}{B}\Big[e_{11}^2 (C + E x^2) + 2 e_{11} E (xy e_{12} - \Delta x' x) \underbrace{+ \frac{E^2 \left(xy e_{12} - \Delta x' x\right)^2}{C + E x^2} - \frac{E^2 \left(xy e_{12} - \Delta x' x\right)^2}{C + E x^2}}_{\text{completing the square}} \\
&\quad + D e_{12}^2 + E\Delta x'^2 + E y^2 e_{12}^2 - 2E\Delta x' y e_{12} \Big] \\
&= -\frac{1}{B}(C + E x^2)\left[e_{11} + \underbrace{\frac{E(xy\, e_{12} - \Delta x' x)}{C + E x^2}}_{K} \right]^2 \\
&\quad + \underbrace{\frac{E^2 (xy\, e_{12} - \Delta x' x)^2}{B(C + E x^2)} - \frac{D}{B} e_{12}^2 - \frac{E}{B}\Delta x'^2 - \frac{E}{B} y^2 e_{12}^2 + \frac{2E}{B}\Delta x' y\, e_{12}}_{N} \,.
\end{aligned} \tag{A.3}$$

Inserting the transformed M from (A.3) into (A.2) yields

$$p_{\Delta X'}\left(\Delta x' | x, y\right) = A \cdot \iint_{-\infty}^{\infty} \exp\left(-\frac{C + E x^2}{B} \cdot \left(e_{11} + K \right)^2 \right) \mathrm{d}e_{11} \cdot \exp\left(N\right) \mathrm{d}e_{12} \,. \tag{A.4}$$

Using the common integral formula

$$\int_{-\infty}^{\infty} \exp\left(-a x^2\right) \mathrm{d}x = \sqrt{\frac{\pi}{a}}\,, \quad \text{for } a > 0 \tag{A.5}$$

(with a as placeholder, only referring in this section) leads to the intermediate solution of

(A.4):

$$p_{\Delta x'}\left(\Delta x'|x,y\right) = A \cdot \sqrt{\frac{\pi \cdot B}{C+Ex^2}} \cdot \int_{-\infty}^{\infty} \exp\left(N\right) \mathrm{d}e_{12} \,. \tag{A.6}$$

The auxiliary variable N from (A.3) can be rewritten by

$$\begin{aligned}
N &= -\frac{1}{B}\Bigg[\frac{-E^2}{C+Ex^2}\left(x^2y^2e_{12}^2 - 2\Delta x' x^2 y e_{12} + \Delta x'^2 x^2\right) + \left(D+Ey^2\right)e_{12}^2 \\
&\quad - 2E\Delta x' y e_{12} + E\Delta x^2\Bigg] \\
&= -\frac{1}{B}\left[e_{12}^2\left(D+Ey^2 - \frac{E^2}{C+Ex^2}x^2y^2\right) + 2E\Delta x'\Big(-y + \frac{Ex^2y}{C+Ex^2}\Big)e_{12}\right. \\
&\quad \left. + E\Delta x'^2\left(1-\frac{Ex^2}{C+Ex^2}\Big)\right] \\
&= -\frac{1}{B}\left[e_{12}^2\left(\frac{CD+E(Cy^2+Dx^2)}{C+Ex^2}\right) + e_{12}\left(-\frac{2E\Delta x' yC}{C+Ex^2}\right) + \frac{CE\Delta x'^2}{C+Ex^2}\right] \\
&= -\frac{1}{B}\Bigg[e_{12}^2\left(\frac{CD+E\left(Cy^2+Dx^2\right)}{C+Ex^2}\right) + e_{12}\left(-\frac{2E\Delta x' yC}{C+Ex^2}\right) \\
&\quad \underbrace{+\frac{\left(CE\Delta x' y\right)^2}{\left[CD+E(Cy^2+Dx^2)\right](C+Ex^2)} - \frac{\left(CE\Delta x' y\right)^2}{\left[CD+E(Cy^2+Dx^2)\right](C+Ex^2)}}_{\text{completing the square}} \\
&\quad + \frac{CE\Delta x'^2}{C+Ex^2}\Bigg] \\
&= -\frac{1}{B}\Bigg[\left(\frac{CD+E(Cy^2+Dx^2}{C+Ex^2}\right)\cdot\left(e_{12}^2 - e_{12}\frac{2E\Delta x' yC}{CD+E\left(Cy^2+Dx^2\right)}\right. \\
&\quad \left. + \frac{\left(CE\Delta x' y\right)^2 \cancel{\left(C+Ex^2\right)}}{\left[CD+E\left(Cy^2+Dx^2\right)\right]\cancel{\left(C+Ex^2\right)}} - \frac{\left(CE\Delta x' y\right)^2 \cancel{\left(C+Ex^2\right)}}{\left[CD+E\left(Cy^2+Dx^2\right)\right]\cancel{\left(C+Ex^2\right)}}\right) \\
&\quad + \frac{CE\Delta x'^2}{C+Ex^2}\Bigg]
\end{aligned}$$

$$
\begin{aligned}
&= \overbrace{-\frac{1}{B}\left[\left(\frac{CD + E\left(Cy^2 + Dx^2\right)}{C + Ex^2}\right) \cdot \underbrace{\left(e_{12} - \frac{CE\Delta x' y}{CD + E\left(Cy^2 + Dx^2\right)}\right)^2}_{\text{binomial formula}}\right]}^{I} \\
&\quad - \frac{1}{B} \cdot \frac{-\left(CE\Delta x' y\right)^2 + CE\Delta x'^2\left(CD + E\left(Cy^2 + Dx^2\right)\right)}{\left[CD + E\left(Cy^2 + Dx^2\right)\right](C + Ex^2)} \\
&= I - \frac{1}{B} \cdot \frac{\cancel{-C^2E^2\Delta x'^2 y^2} + C^2DE\Delta x'^2 + \cancel{C^2E^2\Delta x'^2 y^2} + Dx^2E^2C\Delta x'^2}{\left[CD + E\left(Cy^2 + Dx^2\right)\right](C + Ex^2)} \\
&= I - \frac{1}{B} \cdot \frac{CDE\Delta x'^2 \cdot \cancel{\left(C + Ex^2\right)}}{\left[CD + E\left(Cy^2 + Dx^2\right)\right]\cancel{\left(C + Ex^2\right)}} \\
&= \underbrace{I - \frac{1}{B} \cdot \frac{CDE\Delta x'^2}{\left[CD + E\left(Cy^2 + Dx^2\right)\right]}}_{H} \quad .
\end{aligned} \tag{A.7}
$$

By combining (A.6) and (A.7), integration and insertion of A, the pdf of the displacement estimation error in x-direction is obtained:

$$
\begin{aligned}
p_{\Delta X'}\left(\Delta x' | x, y\right) \; = & \frac{1}{\sqrt{2\pi\sigma_{e_{11}}^2}} \cdot \frac{1}{\sqrt{2\pi\sigma_{e_{12}}^2}} \cdot \frac{1}{\sqrt{2\pi\sigma_{e_{13}}^2}} \\
& \cdot \underbrace{\sqrt{\frac{\pi \cdot B}{C + Ex^2}}}_{\text{see (A.6)}} \cdot \sqrt{\frac{\pi \cdot B\left(C + Ex^2\right))}{CD + E\left(Cy^2 + Dx^2\right)}} \cdot \exp\left(H\right) .
\end{aligned} \tag{A.8}
$$

Resubstitution of the auxiliary variables finally leads to (3.12) (page 43):

$$
\begin{aligned}
p_{\Delta X'}\left(\Delta x' | x, y\right) \; = & \frac{1}{\sqrt{2\pi\left(\sigma_{e_{11}}^2 x^2 + \sigma_{e_{12}}^2 y^2 + \sigma_{e_{13}}^2\right)}} \\
& \cdot \exp\left(-\frac{\Delta x'^2}{2 \cdot \left(\sigma_{e_{11}}^2 x^2 + \sigma_{e_{12}}^2 y^2 + \sigma_{e_{13}}^2\right)}\right) .
\end{aligned} \tag{A.9}
$$

A.2 Derivation of the Probability Density Function of the Displacement Estimation Error for Simplified Affine Motion-Compensated Prediction

The probability density function of the displacement estimation error for simplified affine motion compensation in video coding can be expressed as given in (3.35) (page 57) by

$$
\begin{aligned}
{}^{\text{simp}}p_{\Delta X'_{\text{s}},\Delta Y'_{\text{s}}}\left(\Delta x'_{\text{s}},\Delta y'_{\text{s}}|x,y\right) &= \underbrace{\frac{1}{(2\pi)^2\,\sigma_{e_a}\,\sigma_{e_b}\,\sigma_{e_c}\,\sigma_{e_f}}}_{A} \\
&\cdot \int_{\mathbb{R}^2} \exp\Bigg(-\frac{e_a^2}{2\sigma_{e_a}^2} - \frac{e_b^2}{2\sigma_{e_b}^2} - \frac{\left(\Delta x'_{\text{s}} - e_a x - e_b y\right)^2}{2\sigma_{e_c}^2} \\
&- \frac{\left(\Delta y'_{\text{s}} + e_b x - e_a y\right)^2}{2\sigma_{e_f}^2}\Bigg)\,\mathrm{d}e_a\mathrm{d}e_b\,. \qquad \text{(A.10)}
\end{aligned}
$$

Substituting $\frac{1}{2\sigma_{e_a}^2} = c_a$, $\frac{1}{2\sigma_{e_b}^2} = c_b$, $\frac{1}{2\sigma_{e_c}^2} = c_c$, and $\frac{1}{2\sigma_{e_f}^2} = c_f$ and expanding (A.10) leads to

$$
\begin{aligned}
{}^{\text{simp}}p_{\Delta X'_{\text{s}},\Delta Y'_{\text{s}}}\left(\Delta x'_{\text{s}},\Delta y'_{\text{s}}|x,y\right) &= A\cdot\int_{\mathbb{R}^2}\exp\Big[-c_a e_a^2 - c_b e_b^2 \\
&- c_c\left(\Delta x'^2_{\text{s}} + e_a^2 x^2 + e_b^2 y^2 - 2e_a x\Delta x'_{\text{s}} - 2e_b y\Delta x'_{\text{s}} + 2e_a e_b xy\right) \\
&- c_f\left(\Delta y'^2_{\text{s}} + e_b^2 x^2 + e_a y^2 + 2e_b x\Delta y'_{\text{s}} - 2e_a y\Delta y'_{\text{s}} - 2e_b e_a xy\right)\Big]\,\mathrm{d}e_a\mathrm{d}e_b\,. \qquad \text{(A.11)}
\end{aligned}
$$

Note that capital letters in these derivations as well as the substitutions c_a, c_b, c_c, and c_f are again only shorthand symbols whereby their meaning is restricted to the current section.

The exponent of (A.11), further on abbreviated as N, is sorted by prefactors of e_a^2, e_a, and e_b^2:

$$
\begin{aligned}
N = &-e_a^2\left(c_a + x^2 c_c + y^2 c_f\right) + e_a\left(2x\Delta x'_{\text{s}}c_c - 2e_b xyc_c + 2y\Delta y'_{\text{s}}c_f + 2e_b xyc_f\right) \\
&- e_b^2\left(c_b + y^2 c_c + x^2 c_f\right) - c_c\Delta x'^2_{\text{s}} + 2c_c e_b y\Delta x'_{\text{s}} - c_f\Delta y'^2_{\text{s}} - 2e_b x\Delta y'_{\text{s}}c_f\,. \qquad \text{(A.12)}
\end{aligned}
$$

Using the common integral formula

$$
\int_{-\infty}^{\infty}\exp\left(-ax^2 + bx + c\right)\mathrm{d}x = \sqrt{\frac{\pi}{a}}\,\exp\left(\frac{b^2}{4a} + c\right), \quad \text{for } \mathrm{Re}\{a\} > 0 \qquad \text{(A.13)}
$$

with Re{a} denoting the real part of a yields

$$
\begin{aligned}
&{}^{\text{simp}}p_{\Delta X'_{\text{s}},\Delta Y'_{\text{s}}}\left(\Delta x'_{\text{s}},\Delta y'_{\text{s}}|x,y\right)\\
&\quad= A\cdot\sqrt{\frac{\pi}{c_a+c_c x^2+c_f y^2}}\cdot\int_{-\infty}^{\infty}\exp\Bigg[\underbrace{\frac{\left(2c_c\Delta x'_{\text{s}}x-2xe_b yc_c+2c_f\Delta y'_{\text{s}}y+2c_f e_b xy\right)^2}{4\left(c_a+c_c x^2+c_f y^2\right)}}_{B}\\
&\qquad-e_b^2\left(c_b+c_c y^2+c_f x^2\right)+e_b\left(2\Delta x'_{\text{s}}yc_c-2c_f\Delta y'_{\text{s}}x\right)+\left(-c_c\Delta x_{\text{s}}'^2-c_f\Delta y_{\text{s}}'^2\right)\Bigg]\,\mathrm{d}e_b
\end{aligned}
\tag{A.14}
$$

after the first integration.

Abbreviating the exponent from (A.14) by M and sorting by prefactors of e_b^2 and e_b yields

$$
\begin{aligned}
M=&\underbrace{\frac{\left(2c_c\Delta x'_{\text{s}}x-2xe_b yc_c+c_f 2\Delta y'_{\text{s}}y+2c_f e_b xy\right)^2}{4\left(c_a+c_c x^2+c_f y^2\right)}}_{B}\\
&-e_b^2\left(c_b+c_c y^2+c_f x^2\right)+e_b\left(2\Delta x'_{\text{s}}yc_c-2c_f\Delta y'_{\text{s}}x\right)+\left(-c_c\Delta x_{\text{s}}'^2-c_f\Delta y_{\text{s}}'^2\right)\\
=&\frac{\left(\left(x\Delta x'_{\text{s}}c_c+y\Delta y'_{\text{s}}c_f\right)+e_b xy\left(c_f-c_c\right)\right)^2}{B}\\
&-e_b^2\left(c_b+c_c y^2+c_f x^2\right)+e_b\left(2\Delta x'_{\text{s}}yc_c-2c_f\Delta y'_{\text{s}}x\right)+\left(-c_c\Delta x_{\text{s}}'^2-c_f\Delta y_{\text{s}}'^2\right).
\end{aligned}
\tag{A.15}
$$

Solving the integral of (A.14) with (A.15) using (A.13) again results in

$$
\begin{aligned}
&{}^{\text{simp}}p_{\Delta X'_{\text{s}},\Delta Y'_{\text{s}}}\left(\Delta x'_{\text{s}},\Delta y'_{\text{s}}|x,y\right)\;=\\
&\quad A\cdot\frac{\sqrt{\pi}}{\sqrt{c_a+c_c x^2+c_f y^2}}\cdot\frac{\sqrt{\pi}}{\sqrt{c_b+y^2c_c+x^2c_f-\frac{\left(xy\left(c_f-c_c\right)\right)^2}{B}}}\\
&\quad\cdot\exp\Bigg[\frac{2^2\left(c_c y\Delta x'_{\text{s}}-x\Delta y'_{\text{s}}c_f+\frac{\left(xy\left(c_f-c_c\right)\right)\cdot\left(x\Delta x'_{\text{s}}c_c+y\Delta y'_{\text{s}}c_f\right)}{B}\right)^2}{4\cdot c_b+y^2c_c+x^2c_f-\frac{\left(xy\left(c_f-c_c\right)\right)^2}{B}}\\
&\quad-c_c\Delta x_{\text{s}}'^2-c_f\Delta y_{\text{s}}'^2+\frac{\left(x\Delta x'_{\text{s}}c_c+y\Delta y'_{\text{s}}c_f\right)^2}{B}\Bigg],
\end{aligned}
\tag{A.16}
$$

which can be converted into (3.36) and finally into (3.39).

A.3 2D Fourier Transform of the Displacement Estimation Error for Fully Affine Motion-Compensated Prediction

The 2D Fourier transform of the displacement estimation error (Equation (3.13) on page 44) of the fully affine motion model as used in Equation (3.18) (page 45) is derived as

$$P(\omega_x, \omega_y) = \frac{1}{2\pi\sigma_{\Delta x'}\sigma_{\Delta y'}} \cdot \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\left(-\frac{\Delta x'^2}{2\sigma_{\Delta x'}^2}\right) \cdot \exp\left(-\frac{\Delta y'^2}{2\sigma_{\Delta y'}^2}\right) \cdot \exp\left(-j\omega_x\Delta x'\right) \cdot \exp\left(-j\omega_y\Delta y'\right) \mathrm{d}\Delta x' \,\mathrm{d}\Delta y' , \tag{A.17}$$

where j denotes the imaginary unit.

Separating the integrands and only considering terms of $\Delta x'$ leads to

$$\begin{aligned}
&\int_{-\infty}^{\infty} \exp\left(-\frac{\Delta x'^2}{2\sigma_{\Delta x'}^2} - j\omega_x\Delta x'\right) \mathrm{d}\Delta x' \\
&= \int_{-\infty}^{\infty} \exp\left(-\frac{\Delta x'^2}{2\sigma_{\Delta x'}^2} - j\omega_x\Delta x' \underbrace{+\frac{j^2\cdot\omega_x^2\cdot\sigma_{\Delta x'}^2}{2} - \frac{j^2\cdot\omega_x^2\cdot\sigma_{\Delta x'}^2}{2}}_{\text{completing the square}}\right) \mathrm{d}\Delta x' \\
&= \exp\left(-\frac{\omega_x^2\cdot\sigma_{\Delta x'}^2}{2}\right) \cdot \int_{-\infty}^{\infty} \exp\left(-\underbrace{\left(\frac{\Delta x'}{\sqrt{2}\cdot\sigma_{\Delta x'}} + j\cdot\frac{\omega_x\cdot\sigma_{\Delta x'}}{\sqrt{2}}\right)^2}_{\text{binomial formula}}\right) \mathrm{d}\Delta x' \\
&= \exp\left(-\frac{\omega_x^2\cdot\sigma_{\Delta x'}^2}{2}\right) \cdot \int_{-\infty}^{\infty} \exp\left(-\frac{\left(\Delta x' + j\cdot\omega_x\cdot\sigma_{\Delta x'}^2\right)^2}{2\sigma_{\Delta x'}^2}\right) \mathrm{d}\Delta x' \\
&= \exp\left(-\frac{\omega_x^2\cdot\sigma_{\Delta x'}^2}{2}\right) \cdot \sqrt{2\pi}\cdot\sigma_{\Delta x'} .
\end{aligned} \tag{A.18}$$

Calculating the $\Delta y'$ terms accordingly and inserting both integration results in (A.17) yields

the final Fourier transform of the displacement estimation error

$$P(\omega_x, \omega_y) = \frac{1}{2\pi\sigma_{\Delta x'}\sigma_{\Delta y'}} \cdot \sqrt{2\pi} \cdot \sigma_{\Delta x'} \cdot \exp\left(-\frac{\omega_x^2 \cdot \sigma_{\Delta x'}^2}{2}\right) \cdot \sqrt{2\pi} \cdot \sigma_{\Delta y'} \cdot \exp\left(-\frac{\omega_y^2 \cdot \sigma_{\Delta y'}^2}{2}\right)$$
$$= \exp\left(-\frac{1}{2}\left(\omega_x^2\,\sigma_{\Delta x'}^2 + \omega_y^2\,\sigma_{\Delta y'}^2\right)\right). \tag{A.19}$$

A.4 2D Fourier Transform of the Displacement Estimation Error for Simplified Affine Motion-Compensated Prediction

The 2D Fourier transform of the displacement estimation error (Equation (3.39) on page 57) of the simplified affine motion model as used in Equation (3.18) (page 45) with j being the imaginary unit is derived as

$$\begin{aligned}
&{}^{\text{simp}}p_{\Delta X'_s,\Delta Y'_s}\left(\Delta x'_s, \Delta y'_s | x, y\right) \\
&= \frac{1}{2\pi\,\sigma_{\Delta x'_s}\,\sigma_{\Delta y'_s}\sqrt{1-\rho^2}} \\
&\quad \cdot \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\left(-\frac{1}{2\left(1-\rho^2\right)}\left[\frac{\Delta x_s'^2}{\sigma_{\Delta x'_s}^2} + \frac{\Delta y_s'^2}{\sigma_{\Delta y'_s}^2} - \frac{2\rho\cdot\Delta x'_s\cdot\Delta y'_s}{\sigma_{\Delta x'_s}\cdot\sigma_{\Delta y'_s}}\right]\right) \\
&\quad \cdot \exp\left(-j\omega_x\Delta x'_s - j\omega_y\Delta y'_s\right)\,\mathrm{d}\Delta x'_s\,\mathrm{d}\Delta y'_s \\
&= \frac{1}{2\pi\,\sigma_{\Delta x'_s}\,\sigma_{\Delta y'_s}\sqrt{1-\rho^2}} \\
&\quad \cdot \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\Bigg(\underbrace{-\Delta x_s'^2\,\frac{1}{2\sigma_{\Delta x'_s}^2\left(1-\rho^2\right)}}_{a} + \underbrace{\Delta x'_s\left(\frac{\rho\,\Delta y'_s}{\sigma_{\Delta x'_s}\sigma_{\Delta y'_s}\left(1-\rho^2\right)} - j\omega_x\right)}_{b} \\
&\quad \underbrace{-\frac{\Delta y_s'^2}{2\sigma_{\Delta y'_s}^2\left(1-\rho^2\right)} - j\omega_y\Delta y'_s}_{c}\Bigg)\,\mathrm{d}\Delta x'_s\,\mathrm{d}\Delta y'_s\,.
\end{aligned} \tag{A.20}$$

Note that substitutions in this section like a, b, c are shorthand symbols only referring to the current section.

Using two times the common integral formula

$$\int_{-\infty}^{\infty} \exp\left(-ax^2 + bx + c\right) \mathrm{d}x = \sqrt{\frac{\pi}{a}} \exp\left(\frac{b^2}{4a} + c\right), \quad \text{for } \mathrm{Re}\{a\} > 0 \tag{A.21}$$

with Re{a} denoting the real part of a yields

$$\begin{aligned}
&{}^{\text{simp}}p_{\Delta X'_\mathrm{s},\Delta Y'_\mathrm{s}}\left(\Delta x'_\mathrm{s}, \Delta y'_\mathrm{s} | x, y\right) \\
&= \frac{1}{2\pi\, \sigma_{\Delta x'_\mathrm{s}}\, \sigma_{\Delta y'_\mathrm{s}} \sqrt{1-\rho^2}} \\
&\quad \cdot \int_{-\infty}^{\infty} \sqrt{\frac{\pi}{\left(\frac{1}{2\sigma^2_{\Delta x'_\mathrm{s}}(1-\rho^2)}\right)}} \cdot \exp\left(\frac{\left(\frac{\rho\, \Delta y'_\mathrm{s}}{\sigma_{\Delta x'_\mathrm{s}}\, \sigma_{\Delta y'_\mathrm{s}}(1-\rho^2)} - j\omega_x\right)^2}{\cancel{4} \cdot \frac{1}{\cancel{2}\sigma^2_{\Delta x'_\mathrm{s}}(1-\rho^2)}} - \frac{\Delta y'^2_\mathrm{s}}{2\sigma^2_{\Delta y'_\mathrm{s}}(1-\rho^2)} - j\omega_y \Delta y'_\mathrm{s} \right) \mathrm{d}\Delta y'_\mathrm{s} \\
&= \frac{\cancel{\sqrt{2\pi} \cdot \sigma_{\Delta x'_\mathrm{s}} \cdot \sqrt{1-\rho^2}}}{2\pi \cdot \cancel{\sqrt{1-\rho^2}} \cdot \cancel{\sigma_{\Delta x'_\mathrm{s}}} \cdot \sigma_{\Delta y'_\mathrm{s}}} \\
&\quad \cdot \int_{-\infty}^{\infty} \exp\left(\frac{\left(\frac{\rho\, \Delta y'_\mathrm{s}}{\sigma_{\Delta x'_\mathrm{s}}\, \sigma_{\Delta y'_\mathrm{s}}(1-\rho^2)}\right)^2 - \frac{2j\omega_x\, \rho\, \Delta y'_\mathrm{s}}{\sigma_{\Delta x'_\mathrm{s}}\, \sigma_{\Delta y'_\mathrm{s}}(1-\rho^2)} - \omega_x^2}{\frac{2}{\sigma^2_{\Delta x'_\mathrm{s}}(1-\rho^2)}} - \frac{\Delta y'^2_\mathrm{s}}{2\sigma^2_{\Delta y'_\mathrm{s}}(1-\rho^2)} - j\omega_y \Delta y'_\mathrm{s} \right) \mathrm{d}\Delta y'_\mathrm{s} \\
&= \frac{1}{\sqrt{2\pi} \cdot \sigma_{\Delta y'_\mathrm{s}}} \cdot \int_{-\infty}^{\infty} \exp\left(-\Delta y'^2_\mathrm{s} \cdot \left[\frac{-\rho^2}{2\sigma^2_{\Delta y'_\mathrm{s}}(1-\rho^2)} + \frac{1}{2\sigma^2_{\Delta y'_\mathrm{s}}(1-\rho^2)} \right] \right. \\
&\quad \left. + \Delta y'_\mathrm{s} \left(-j\omega_y - \frac{j\omega_x\, \rho\, \sigma_{\Delta x'_\mathrm{s}}}{\sigma_{\Delta y'_\mathrm{s}}} \right) + \left(-\frac{\omega_x^2 \cdot \sigma^2_{\Delta x'_\mathrm{s}} \cdot \left(1-\rho^2\right)}{2} \right) \right) \mathrm{d}\Delta y'_\mathrm{s} \\
&= \frac{1}{\sqrt{2\pi} \cdot \sigma_{\Delta y'_\mathrm{s}}} \cdot \int_{-\infty}^{\infty} \exp\left(-\Delta y'^2_\mathrm{s} \cdot \underbrace{\left[\frac{1}{2\sigma^2_{\Delta y'_\mathrm{s}}} \right]}_{a^*} \right. \\
&\quad \left. + \Delta y'_\mathrm{s} \underbrace{\left(-j\omega_y - \frac{j\omega_x\, \rho\, \sigma_{\Delta x'_\mathrm{s}}}{\sigma_{\Delta y'_\mathrm{s}}} \right)}_{b^*} + \underbrace{\left(-\frac{\omega_x^2 \cdot \sigma^2_{\Delta x'_\mathrm{s}} \cdot \left(1-\rho^2\right)}{2} \right)}_{c^*} \right) \mathrm{d}\Delta y'_\mathrm{s} \,.
\end{aligned} \tag{A.22}$$

Using (A.21) again with a^* for a, b^* for b, and c^* for c results in

$$
\begin{aligned}
&{}^{\text{simp}}p_{\Delta X'_{\text{s}},\Delta Y'_{\text{s}}}\left(\Delta x'_{\text{s}},\Delta y'_{\text{s}}|x,y\right) \\
&= \frac{1}{\sqrt{2\pi}\cdot\sigma_{\Delta y'_{\text{s}}}}\cdot\sqrt{\frac{\pi}{\frac{1}{2\sigma^2_{\Delta y'_{\text{s}}}}}}\cdot\exp\left(\frac{\left(-j\omega_y-\frac{j\omega_x\,\rho\,\sigma_{\Delta x'_{\text{s}}}}{\sigma_{\Delta y'_{\text{s}}}}\right)^2}{4\cdot\frac{1}{2\cdot\sigma^2_{\Delta y'_{\text{s}}}}}-\frac{\omega_x^2\,\sigma^2_{\Delta x'_{\text{s}}}\left(1-\rho^2\right)}{2}\right) \\
&= \exp\left(-\frac{\omega_y^2\,\sigma^2_{\Delta y'_{\text{s}}}}{2}+\frac{2j\omega_y\,j\omega_x\,\rho\,\sigma_{\Delta x'_{\text{s}}}}{\sigma_{\Delta y'_{\text{s}}}}\frac{\sigma^2_{\Delta y'_{\text{s}}}}{2}-\frac{\omega_x^2\,\rho^2\,\sigma^2_{\Delta x'_{\text{s}}}}{\cancel{\sigma^2_{\Delta y'_{\text{s}}}}}\frac{\cancel{\sigma^2_{\Delta y'_{\text{s}}}}}{2}-\frac{\omega_x^2\,\sigma^2_{\Delta x'_{\text{s}}}\left(1-\rho^2\right)}{2}\right). \qquad \text{(A.23)}
\end{aligned}
$$

Simplification finally results in the 2D Fourier transform of the displacement estimation error

$$
\begin{aligned}
P(\omega_x,\omega_y) &= \exp\left(-\frac{\omega_x^2\,\sigma^2_{\Delta x_{\text{s}}'}}{2}-\frac{\omega_y^2\,\sigma^2_{\Delta y_{\text{s}}'}}{2}-2\,\omega_x\,\omega_y\,\rho\,\sigma_{\Delta'_{x_{\text{s}}}}\sigma_{\Delta'_{y_{\text{s}}}}\right) \\
&= \exp\left(-\frac{1}{2}\left(\omega_x^2\,\sigma^2_{\Delta x_{\text{s}}'}+\omega_y^2\,\sigma^2_{\Delta y_{\text{s}}'}\right)-2\,\omega_x\,\omega_y\,\rho\,\sigma_{\Delta'_{x_{\text{s}}}}\sigma_{\Delta'_{y_{\text{s}}}}\right). \qquad \text{(A.24)}
\end{aligned}
$$

Bibliography

[1] T. Aach and A. Kaup. *Bayesian Algorithms for Adaptive Change Detection in Image Sequences Using Markov Random Fields*. Signal Processing–Image Communication, 7:147–160, 1995.

[2] T. Aach, A. Kaup, and R. Mester. *Statistical Model-Based Change Detection in Moving Video*. Signal Processing, 31(2):165–180, 1993. doi: 10.1016/0165-1684(93)90063-G.

[3] E. Alshina, A. Alshin, K. Choi, and M. Park. *Performance of JEM 1 Tools Analysis*. Joint Video Exploration Team (JVET) of ITU-T VCEG and ISO/IEC MPEG, 2nd Meeting: San Diego, CA, USA, Document: JVET-B0022, Feb. 2016. URL `http://phenix.it-sudparis.eu/jvet/`.

[4] C. H. Anderson, P. J. Burt, and G. S. Van der Wal. *Change Detection and Tracking Using Pyramid Transform Techniques*. SPIE Intelligent Robotics and Computer Vision IV, 579:72–78, 1985.

[5] AOMedia Video 1 (AV1). *AOM – AV1: How does it work?* July 2017. URL `https://parisvideotech.com/wp-content/uploads/2017/07/AOM-AV1-Video-Tech-meet-up.pdf`.

[6] J. Bang, D. Kim, and H. Eom. *Motion Object and Regional Detection Method Using Block-based Background Difference Video Frames*. In Proceedings of the 18th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), pages 350 –357, Seoul, Korea (South), Aug. 2012. doi: 10.1109/RTCSA.2012.58.

[7] T. Berger. *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Prentice-Hall electrical engineering series. Prentice-Hall, Inc., Englewood Cliffs, NJ, USA, 1971. ISBN 9780137531035.

[8] H. Beyer. *Accurate Calibration of CCD-Cameras*. In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pages 96 –101, Champaign, IL, USA, June 1992. doi: 10.1109/CVPR.1992.223221.

[9] M. Bierling. *Hierarchische Displacementschätzung zur Bewegungskompensation in digitalen Fernsehbildsequenzen*. Number 179 in Fortschritt-Berichte VDI: Reihe 10, Informatik/Kommunikation. VDI Verlag GmbH, Fachbereich Elektrotechnik und Informationstechnik der Universität Hannover, 1991. ISBN 978-3-18-1479100. PhD thesis.

[10] S. Birchfield. *KLT: An Implementation of the Kanade-Lucas-Tomasi Feature Tracker*. 2007. URL `https://cecas.clemson.edu/~stb/klt/`.

[11] G. Bjøntegaard. *VCEG-M33: Calculation of Average PSNR Differences between RD Curves*. In ITU-T SG 16/Q6. 13th Meeting, Austin, TX, USA, Apr. 2001.

[12] G. Bjøntegaard. *VCEG-AI11: Improvements of the BD-PSNR model*. In ITU-T SG 16/Q6. 35th Meeting, Berlin, Germany, 2008.

[13] F. Bossen. *L1100: Common HM Test Conditions and Software Reference Configurations*. Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 12th Meeting, Geneva, Switzerland, 2013.

[14] B. Bross. *Versatile Video Coding (Draft 1)*. Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 10th Meeting: San Diego, CA, USA, Document: JVET-J1001, July 2018. URL `http://phenix.it-sudparis.eu/jvet/`.

[15] H. Broszio. *Schätzung von Brennweite und Rotation einer Kamera mit Radialverzerrung und Rotationsachsenversatz aus Bildsequenzen*. Fortschritt-Berichte VDI: Reihe 10, Informatik/Kommunikation. VDI Verlag GmbH, Fachbereich Elektrotechnik und Informationstechnik der Universität Hannover, 2007. ISBN 978-3-18-378310-6. PhD thesis.

[16] C. Bulla, C. Feldmann, and M. Schink. *Region of Interest Encoding in Video Conference Systems*. In Proceedings of the 5th International Conferences on Advances in Multimedia, pages 119–124, Venice, Italy, Apr. 2013.

[17] X. Cao, J. Lan, P. Yan, and X. Li. *KLT Feature Based Vehicle Detection and Tracking in Airborne Videos*. In Proceedings of the 6th International Conference on Image and Graphics (ICIG), pages 673–678, Hefei, China, Aug. 2011. doi: 10.1109/ICIG.2011.92.

[18] J. Chen, E. Alshina, G.-J. Sullivan, J.-R. Ohm, and J. Boyce. *Algorithm Description of Joint Exploration Test Model (JEM) 1.* Joint Video Exploration Team (JVET) of ITU-T VCEG and ISO/IEC MPEG, 1st Meeting: Geneva, Switzerland, Document: JVET-A1001, Oct. 2015. URL `http://phenix.it-sudparis.eu/jvet/`.

[19] M.-J. Chen, M.-C. Chi, C.-T. Hsu, and J.-W. Chen. *ROI Video Coding based on H.263+ with Robust Skin-Color Detection Technique*. IEEE Transactions on Consumer Electronics, 49(3):724–730, Aug. 2003. ISSN 0098-3063. doi: 10.1109/TCE.2003.1233810.

[20] S.-C. S. Cheung and C. Kamath. *Robust Techniques for Background Subtraction in Urban Traffic Video*. In S. Panchanathan and B. Vasudev, editors, Proceedings of the Visual Communications and Image Processing (VCIP), volume 5308, pages 881–892, San Jose, CA, USA, 2004. SPIE.

[21] W. Chimiak and B. V. Smith. *The Xfig Buildings Library*. URL: `http://xfig.org`.

[22] K. Cordes. *Occlusion Handling in Scene Reconstruction from Video*. Number 834 in Fortschritt-Berichte VDI: Informatik/Kommunikation. VDI Verlag GmbH, 2014. ISBN 978-3-18-383410-5. PhD thesis.

[23] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2012. ISBN 9781118585771.

[24] N. Dalal and B. Triggs. *Histograms of Oriented Gradients for Human Detection*. In C. Schmid, S. Soatto, and C. Tomasi, editors, Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST, Montbonnot, France, June 2005.

[25] G. Dane and T. Q. Nguyen. *The Effect of Global Motion Parameter Accuracies on the Efficiency of Video Coding*. In Proceedings of the IEEE International Conference on Image Processing (ICIP), volume 5, pages 3359–3362 Vol. 5, Singapore, Oct. 2004. ISBN 0-7803-8554-3. doi: 10.1109/ICIP.2004.1421834.

[26] F. Devernay and O. Faugeras. *Straight Lines Have to Be Straight: Automatic Calibration and Removal of Distortion from Scenes of Structured Environments*. In Proceedings of SPIE, volume 2567, pages 14–24, Secaucus, NJ, USA, Aug. 2001. Springer-Verlag New York, Inc. doi: 10.1007/PL00013269.

[27] L. Ding, Y. Tian, H. Fan, Y. Wang, and T. Huang. *High-Efficiency Coding for Shaking Surveillance Videos based on Global Motion Compensation*. In Proceedings of the 2nd IEEE International Conference on Multimedia Big Data (BigMM), pages 259–265, Taipei, Taiwan, Apr. 2016. doi: 10.1109/BigMM.2016.42.

[28] N. Doulamis, A. Doulamis, D. Kalogeras, and S. Kollias. *Low Bit-Rate Coding of Image Sequences using Adaptive Regions of Interest*. IEEE Transactions on Circuits and Systems for Video Technology, 8(8):928 –934, Dec. 1998. ISSN 1051-8215. doi: 10.1109/76.736718.

[29] C. E. Duchon. *Lanczos Filtering in One and Two Dimensions*. Journal of Applied Meteorology, 18(8):1016–1022, 08 1979. doi: 10.1175/1520-0450(1979)018<1016:LFIOAT>2.0.CO;2.

[30] A. Eden. *H.264 - Systemanalyse und Evaluation der Leistungsfähigkeit des Codecs*. August 2004. Diploma thesis, Institut für Nachrichtentechnik, Technische Universität Braunschweig, Braunschweig, Germany.

[31] T. Elbrandt and J. Ostermann. *Enabling Accurate Measurement of Camera Distortions using Dynamic Continuous-Tone Patterns*, volume 18. IOS Press, 2011.

[32] L. Falkenhagen. *Blockbasierte Disparitätsschätzung unter Berücksichtigung statistischer Abhängigkeiten der Disparitäten*. Number 657. VDI Verlag GmbH, Fachbereich Elektrotechnik und Informationstechnik der Universität Hannover, 2001. ISBN 978-3-18-365710-0. PhD thesis.

[33] M. K. Fard, M. Yazdi, and M. MasnadiShirazi. *A Block Matching Based Method for Moving Object Detection in Active Camera*. In Proceedings of the 5th Conference on Information and Knowledge Technology (IKT), pages 443–446, Shiraz, Iran, May 2013. doi: 10.1109/IKT.2013.6620108.

[34] FFmpeg Developers. *ffmpeg Lavc57.107.100 libx265*, 2018. URL: `http://ffmpeg.org/`.

[35] M. A. Fischler and R. C. Bolles. *Random Sample Consensus: a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography*. Communications of the ACM, 24(6):381–395, June 1981. ISSN 0001-0782. doi: 10.1145/358669.358692.

[36] B. Girod. *The Efficiency of Motion-Compensating Prediction for Hybrid Coding of Video Sequences*. IEEE Journal on Selected Areas in Communications, 5(7): 1140 – 1154, Aug. 1987. ISSN 0733-8716. doi: 10.1109/JSAC.1987.1146632.

[37] Google LLC. *Google Earth*, 1998. URL: `https://www.google.com/earth/`, accessed Dec. 2015.

[38] P. Gorur and B. Amrutur. *Skip Decision and Reference Frame Selection for Low-Complexity H.264/AVC Surveillance Video Coding*. IEEE Transactions on Circuits and Systems for Video Technology, 24(7):1156–1169, July 2014. ISSN 1051-8215. doi: 10.1109/TCSVT.2014.2319611.

[39] D. Grois and O. Hadar. *Complexity-Aware Adaptive Spatial Pre-Processing for ROI Scalable Video Coding with Dynamic Transition Region*. In Proceedings of the 18th IEEE International Conference on Image Processing (ICIP), pages 741–744, Brussels, Belgium, Sept. 2011. doi: 10.1109/ICIP.2011.6116661.

[40] M. J. Hannah. *Digital Stereo Image Matching Techniques*. In International Archives of Photogrammetry and Remote Sensing, number Bd. 27.B3, pages 280–293, 1988.

[41] C. Harris and M. Stephens. *A Combined Corner and Edge Detection*. In Proceedings of the 4th Alvey Vision Conference, pages 147–151, Manchester, UK, 1988.

[42] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, 2nd edition, 2004.

[43] B. K. P. Horn and B. G. Schunck. *Determining Optical Flow*. Artificial Intelligence, 17:185–203, 1981. doi: http://dx.doi.org/10.1016/0004-3702(81)90024-2.

[44] M. Hötter. *Objektorientierte Analyse-Synthese-Codierung basierend auf dem Modell bewegter, zweidimensionaler Objekte*. Fortschritt-Berichte VDI: Reihe 10, Informatik/Kommunikation. VDI Verlag GmbH, Düsseldorf, Fachbereich Elektrotechnik und Informationstechnik der Universität Hannover, 1992. ISBN 3-18-141710-6. PhD thesis.

[45] A. W. N. Ibrahim, P. W. Ching, G. G. Seet, W. M. Lau, and W. Czajewski. *Moving Objects Detection and Tracking Framework for UAV-based Surveillance*. In Proceedings of the 4th Pacific-Rim Symposium on Image and Video Technology (PSIVT), pages 456–461, Singapore, Nov. 2010. doi: 10.1109/PSIVT.2010.83.

[46] Institut für Informationsverarbeitung (TNT), Leibniz Universität Hannover. *TNT Aerial Video Testset (TAVT)*, 2010–2014. URL `https://www.tnt.uni-hannover.de/project/TNT_Aerial_Video_Testset/`.

[47] ISO/IEC. *ISO/IEC 11172-2 (MPEG-1 Part 2): Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s – Part 2: Video*. Aug. 1993.

[48] ISO/IEC. *ISO/IEC 14496:2000-2: Information technology - Coding of Audio-Visual Objects – Part 2: Visual*. Dec. 2000.

[49] ISO/IEC and ITU-T. *Recommendation ITU-T H.264 and ISO/IEC 14496-10 (MPEG-4 Part 10): Advanced Video Coding (AVC)*. Geneva, Switzerland, 3rd edition, July 2004.

[50] ISO/IEC and ITU-T. *Recommendation ITU-T H.262 and ISO/IEC 13818-2 (MPEG-2 Part 2): Information technology - Generic coding of moving pictures and associated audio information: Video*. Mar. 1995.

[51] ISO/IEC and ITU-T. *ITU-T Recommendation H.265/ ISO/IEC JTC 1/SC 29 23008-2:2015-05-01 MPEG-H Part 2/: High Efficiency Video Coding (HEVC)*. 2nd edition, Apr. 2015.

[52] ITU-T. *Recommendation ITU-T H.261: Video codec for audiovisual services at p×64 kbit/s*. Geneva, Switzerland, Nov. 1988.

[53] ITU-T. *Recommendation ITU-T H.263: Video Coding for Low Bit Rate Communication*. Geneva, Switzerland, 1998. version 1, Nov. 1995; version 2, Jan. 1998; version 3, Nov. 2000.

[54] ITU-T Study Group 12. *Recommendation P.913: Methods for the Subjective Assessment of Video Quality, Audio Quality and Audiovisual Quality of Internet Video and Distribution Quality Television in any Environment*. Mar. 2016. URL `http://handle.itu.int/11.1002/1000/12775`.

[55] Joint Video Exploration Team (JVET). *Joint Exploration Model (JEM)*, Oct. 2017. URL `https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/branches/HM-16.6-JEM-7.1-dev/`. accessed on Oct. 2018.

[56] R. Jones, B. Ristic, N. Redding, and D. Booth. *Moving Target Indication and Tracking from Moving Sensors*. In Proceedings of Digital Image Computing:

Techniques and Applications (DICTA), Cairns, Australia, Dec. 2005. doi: 10.1109/DICTA.2005.57.

[57] P. Kaewtrakulpong and R. Bowden. *An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection*. In Proceedings of the 2nd European Workshop on Advanced Video-Based Surveillance Systems, (AVBS01), Video-Based Surveillance Systems: Computer Vision and Distributed Processing, London, UK, Sept. 2001. Kluwer Academic Publishers.

[58] J. Kang, I. Cohen, G. Medioni, and C. Yuan. *Detection and Tracking of Moving Objects from a Moving Platform in Presence of Strong Parallax*. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), volume 1, pages 10–17, Beijing, China, Oct. 2005. doi: 10.1109/ICCV.2005.72.

[59] L. Karlsson, M. Sjöström, and R. Olsson. *Spatio-Temporal Filter for ROI Video Coding*. In Proceedings of the 14th European Signal Processing Conference (EUSIPCO), Florence, Italy, Sept. 2006.

[60] S. Klomp. *Decoderseitige Bewegungsschätzung in der Videocodierung*. Fortschritt-Berichte VDI: Reihe 10, Informatik/Kommunikation. 2012. ISBN 978-3-18-382010-8. PhD thesis.

[61] Z. Kukelova, J. Heller, M. Bujnak, and T. Pajdla. *Radial Distortion Homography*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 639–647, Boston, MA, USA, June 2015. doi: 10.1109/CVPR.2015.7298663.

[62] R. Kumar, H. Sawhney, S. Samarasekera, S. Hsu, H. Tao, Y. Guo, K. Hanna, A. Pope, R. Wildes, D. Hirvonen, M. Hansen, and P. Burt. *Aerial Video Surveillance and Exploitation*. Proceedings of the IEEE, 89(10):1518–1539, Oct. 2001. ISSN 0018-9219. doi: 10.1109/5.959344.

[63] Landesamt für Geoinformation und Landesvermessung Niedersachsen (LGN). Test image: *Hannover*, July 2013. Name: dop20_32550000_5802000_col.tif, Ground resolution: 0.2 m x 0.2 m, Format: TIFF image, File size: 301 MB, Format: Raw, Width: 10000 pel, Height: 10000 pel, Color space: RGB, Bit depth: 8 bit, Compression mode: Lossless.

[64] T. Laude, Y. G. Adhisantoso, J. Voges, M. Munderloh, and J. Ostermann. *A Comprehensive Video Codec Comparison*. In APSIPA Transactions on Signal and Information Processing, volume 8, Oct. 2019.

[65] L. Li, H. Li, D. Liu, Z. Li, H. Yang, S. Lin, H. Chen, and F. Wu. *An Efficient Four-Parameter Affine Motion Model for Video Coding*. IEEE Transactions on Circuits and Systems for Video Technology, PP(99):1–1, 2017. ISSN 1051-8215. doi: 10.1109/TCSVT.2017.2699919.

[66] X. Li, J. Boyce, P. Onno, and Y. Ye. *L1009: Common Test Conditions and Software Reference Configurations for the Scalable Test Model*. Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11. 12th Meeting, Geneva, Switzerland, 2013.

[67] Y. Liu, Z. Li, Y. Soh, and M. Loke. *Conversational Video Communication of H.264/AVC with Region-of-Interest Concern*. In Proceedings of the IEEE International Conference on Image Processing (ICIP), pages 3129–3132, Atlanta, GA, USA, Oct. 2006. doi: 10.1109/ICIP.2006.312936.

[68] Y. Liu, Z. G. Li, and Y. C. Soh. *Region-of-Interest Based Resource Allocation for Conversational Video Communication of H.264/AVC*. IEEE Transactions on Circuits and Systems for Video Technology, 18(1):134–139, Jan. 2008. ISSN 1051-8215. doi: 10.1109/TCSVT.2007.913754.

[69] D. G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision, 60(2):91–110, 2004.

[70] B. D. Lucas. *Generalized Image Matching by the Method of Differences*. July 1984. Technical report and PhD Thesis, Robotics Institute, Carnegie Mellon University, PA, USA.

[71] B. D. Lucas and T. Kanade. *An Iterative Image Registration Technique with an Application to Stereo Vision*. In Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI), volume 2, pages 674–679, Vancouver, Canada, Aug. 1981.

[72] Mathworks. General documentation of box plots at `https://de.mathworks.com/help/stats/box-plots.html` and `boxplot` command documentation: `https://de.mathworks.com/help/stats/boxplot.html`, accessed on April 23, 2019.

[73] K. McCann, C. Rosewarne, B. Bross, M. Naccari, K. Sharman, and G. J. Sullivan. *High Efficency Video Coding (HEVC) Test Model 16 (HM16) Encoder Description*. Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3

and ISO/IEC JTC1/SC29/WG11, 18th Meeting: Sapporo, Japan, Document JCTVC-R1002, July 2014.

[74] M. Meddeb, M. Cagnazzo, and B. Pesquet-Popescu. *Region-of-Interest Based Rate Control Scheme for High Efficiency Video Coding*. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 7338–7342, Florence, Italy, May 2014. doi: 10.1109/ICASSP.2014.6855025.

[75] H. Meuel, M. Munderloh, and J. Ostermann. *Low Bit Rate ROI Based Video Coding for HDTV Aerial Surveillance Video Sequences*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition – Workshops (CVPRW), pages 13 –20, Colorado Springs, CO, USA, June 2011. doi: 10.1109/CVPRW.2011.5981687.

[76] H. Meuel, M. Munderloh, and J. Ostermann. *Radial Distortion in Hybrid Video Coding*. Technical report, Institut für Informationsverarbeitung, Leibniz Universität Hannover, Hannover, Germany, June 2012. URL `http://www.tnt.uni-hannover.de/papers/data/976/Radial_Distortion_Compensation_in_Hybrid_Video_Coding_-_Technical_Report_-_Meuel_Munderloh_Ostermann.pdf`.

[77] H. Meuel, M. Munderloh, M. Reso, and J. Ostermann. *Optical Flow Cluster Filtering for ROI Coding*. In Proceedings of the Picture Coding Symposium (PCS), pages 129–132, San Jose, CA, USA, Dec. 2013. doi: 10.1109/PCS.2013.6737700.

[78] H. Meuel, M. Reso, J. Jachalsky, and J. Ostermann. *Superpixel-based Segmentation of Moving Objects for Low-Complexity Surveillance Systems*. In Proceedings of the 10th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 395–400, Kraków, Poland, Aug. 2013. doi: 10.1109/AVSS.2013.6636672.

[79] H. Meuel, J. Schmidt, M. Munderloh, and J. Ostermann. *Advanced Video Coding for Next-Generation Multimedia Services – Chapter 3: Region of Interest Coding for Aerial Video Sequences Using Landscape Models*. Intech, Jan. 2013. URL `http://www.intechopen.com/books/advanced-video-coding-for-next-generation-multimedia-services/region-of-interest-coding-for-aerial-video-sequences-using-landscape-models`.

[80] H. Meuel, M. Munderloh, and J. Ostermann. *Stereo Mosaicking and 3D-Video for Singleview HDTV Aerial Sequences using a Low Bit Rate ROI Coding Framework*. In Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 1–6, Karlsruhe, Germany, Aug. 2015. doi: 10.1109/AVSS.2015.7301789.

[81] H. Meuel, M. Munderloh, M. Reso, and J. Ostermann. *Mesh-based Piecewise Planar Motion Compensation and Optical Flow Clustering for ROI Coding*. In APSIPA Transactions on Signal and Information Processing, volume 4, 2015. doi: 10.1017/ATSIP.2015.12. URL `http://journals.cambridge.org/article_S2048770315000128`.

[82] H. Meuel, L. Angerstein, R. Henschel, B. Rosenhahn, and J. Ostermann. *Moving Object Tracking for Aerial Video Coding using Linear Motion Prediction and Block Matching*. In Proceedings of the 32nd IEEE Picture Coding Symposium (PCS), pages 1–5, Nuremberg, Germany, Dec. 2016. doi: 10.1109/PCS.2016.7906333.

[83] H. Meuel, S. Ferenz, M. Munderloh, H. Ackermann, and J. Ostermann. *In-Loop Radial Distortion Compensation for Long-Term Mosaicking of Aerial Videos*. In Proceedings of the 23rd IEEE International Conference on Image Processing (ICIP), pages 2961–2965, Phoenix, AZ, USA, Sept. 2016. ISBN 978-1-4673-9961-6/16. doi: 10.1109/ICIP.2016.7532902.

[84] H. Meuel, F. Kluger, and J. Ostermann. *Illumination Change Robust, Codec Independent Low Bit Rate Coding of Stereo from Singleview Aerial Video*. In Proceedings of the 10th IEEE International 3DTV Conference, pages 1–4, Hamburg, Germany, July 2016. doi: 10.1109/3DTV.2016.7548961.

[85] H. Meuel, M. Munderloh, F. Kluger, and J. Ostermann. *Codec Independent Region of Interest Video Coding using a Joint Pre- and Postprocessing Framework*. In Proceedings of the 9th IEEE International Conference on Multimedia and Expo (ICME), pages 1–6, Seattle, WA, USA, July 2016. doi: 10.1109/ICME.2016.7552876.

[86] H. Meuel, S. Ferenz, F. Kluger, and J. Ostermann. *Robust Long-Term Aerial Video Mosaicking by Weighted Feature-based Global Motion Estimation*. In Proceedings of the 17th International Conference on Computer Analysis of Images and Patterns (CAIP), Ystad, Sweden, Aug. 2017.

[87] H. Meuel, S. Ferenz, Y. Liu, and J. Ostermann. *Rate-Distortion Theory for Affine Global Motion Compensation in Video Coding*. In Proceedings of the 25th IEEE International Conference on Image Processing (ICIP), pages 3593–3597, Athens, Greece, Oct. 2018. doi: 10.1109/ICIP.2018.8451136.

[88] H. Meuel, S. Ferenz, Y. Liu, and J. Ostermann. *Rate-Distortion Theory for Simplified Affine Motion Compensation Used in Video Coding*. In IEEE Visual Communications and Image Processing (VCIP), pages 1–4, Taichung, Taiwan, Dec. 2018. doi: 10.1109/VCIP.2018.8698702.

[89] H. Meuel, F. Kluger, and J. Ostermann. *Region of Interest (ROI) Coding for Aerial Surveillance Video using AVC & HEVC*. ArXiv e-prints, Jan. 2018.

[90] M. Munderloh. *Detection of Moving Objects for Aerial Surveillance of Arbitrary Terrain*. Number 847 in Fortschritt-Berichte VDI: Reihe 10, Informatik/Kommunikation. VDI Verlag GmbH, Düsseldorf, 2016. ISBN 978-3-18-384710-5. PhD thesis.

[91] M. Munderloh, H. Meuel, and J. Ostermann. *Mesh-based Global Motion Compensation for Robust Mosaicking and Detection of Moving Objects in Aerial Surveillance*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 1–6, Colorado Springs, CO, USA, June 2011. doi: 10.1109/CVPRW.2011.5981678.

[92] H.-G. Musmann. *Quellencodierung (Lecture Notes)*. Institut für Informationsverarbeitung, Leibniz Universität Hannover, Hannover, Germany, Reprint: Autumn 2002.

[93] H.-G. Musmann. *Statistische Methoden der Nachrichtentechnik (Lecture Notes)*. Institut für Informationsverarbeitung, Leibniz Universität Hannover, Hannover, Germany, Reprint: May 2017.

[94] S. Negahdaripour and P. Firoozfam. *Positioning and Photo-Mosaicking with Long Image Sequences; Comparison of Selected Methods*. In Proceedings of the OCEANS, MTS/IEEE Conference and Exhibition, volume 4, pages 2584–2592 vol.4, Honolulu, HI, USA, 2001. doi: 10.1109/OCEANS.2001.968407.

[95] J. O'Neal and T. Natarajan. *Coding Isotropic Images*. IEEE Transactions on Information Theory, 23(6):697–707, Nov. 1977. ISSN 0018-9448. doi: 10.1109/TIT.1977.1055796.

[96] S. Parker, Y. Chen, D. Barker, P. de Rivaz, and D. Mukherjee. *Global and Locally Adaptive Warped Motion Compensation in Video Compression*. In Proceedings of the IEEE International Conference on Image Processing (ICIP), pages 275–279, Beijing, China, Sept. 2017. doi: 10.1109/ICIP.2017.8296286.

[97] D. Patel, T. Lad, and D. Shah. *Review on Intra-prediction in High Efficiency Video Coding (HEVC) Standard*. International Journal of Computer Applications, 132:26–29, 12 2015.

[98] K. Pearson. *Notes on Regression and Inheritance in the Case of Two Parents*, volume 58. Proceedings of the Royal Society of London (Great Britain), editors Taylor & Francis, June 1895.

[99] P. Z. Peebles, Jr. *Probability, Random Variables and Random Signal Principles*. McGraw-Hill Education, New York, 1993. ISBN 9780070492738.

[100] F. C. Pereira and T. Ebrahimi. *The MPEG-4 Book*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2002. ISBN 0130616214.

[101] P. W. Power and J. A. Schoonees. *Understanding Background Mixture Models for Foreground Segmentation*. In Proceedings of the Image and Vision Computing, page 267, Auckland, New Zealand, Nov. 2002.

[102] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 2nd edition, 1992. ISBN 0-521-43108-5.

[103] K. Quast and A. Kaup. *Spatial Scalable Region of Interest Transcoding of JPEG2000 for Video Surveillance*. In IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 203–210, Santa Fe, NM, USA,„ Sept. 2008. doi: 10.1109/AVSS.2008.17.

[104] U. Reimers. *DVB: The Family of International Standards for Digital Video Broadcasting*. Springer, New York, 2nd edition, 2004. ISBN 978-3540435457.

[105] M. Reso, J. Jachalsky, B. Rosenhahn, and J. Ostermann. *Temporally Consistent Superpixels*. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 385–392, Dec. 2013. doi: 10.1109/ICCV.2013.55.

[106] B. Scheuermann and B. Rosenhahn. *SlimCuts: GraphCuts for High Resolution Images Using Graph Reduction*. In Y. Boykov, F. Kahl, V. Lempitsky, and F. R.

Schmidt, editors, Energy Minimization Methods in Computer Vision and Pattern Recognition, volume 6819 of *Lecture Notes in Computer Science*, pages 219–232. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-23093-6. doi: 10.1007/978-3-642-23094-3_16.

[107] A. Shastry and R. Schowengerdt. *Airborne Video Registration and Traffic-Flow Parameter Estimation*. IEEE Transactions on Intelligent Transportation Systems, 6(4):391–405, Dec. 2005. ISSN 1524-9050. doi: 10.1109/TITS.2005.858621.

[108] J. Shi and C. Tomasi. *Good Features to Track*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, June 1994.

[109] C. Stauffer and W. E. L. Grimson. *Adaptive Background Mixture Models for Real-Time Tracking*. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, volume 2, pages 246–252, Los Alamitos, CA, USA, Aug. 1999. IEEE Computer Society.

[110] K. Suehring and X. Li. *JVET Common Test Conditions and Software Reference Configurations*. Joint Video Exploration Team (JVET) of ITU-T VCEG and ISO/IEC MPEG, 2nd Meeting: San Diego, CA, USA, Document JVET-B1010, Feb. 2016. URL `http://phenix.it-sudparis.eu/jvet/`.

[111] K. Suehring and X. Li. *JVET Common Test Conditions and Software Reference Configurations*. Joint Video Exploration Team (JVET) of ITU-T VCEG and ISO/IEC MPEG, 8th Meeting: Macau, Document: JVET-H1010, Oct. 2017. URL `http://phenix.it-sudparis.eu/jvet/`.

[112] G. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand. *Overview of the High Efficiency Video Coding (HEVC) Standard*. IEEE Transactions on Circuits and Systems for Video Technology, 22(12):1649–1668, Dec. 2012. ISSN 1051-8215. doi: 10.1109/TCSVT.2012.2221191.

[113] R. Szeliski. *Image Alignment and Stitching: A Tutorial*. Foundations and Trends in Computer Graphics and Vision, 2(1):1–104, Jan. 2006. ISSN 1572-2740. doi: 10.1561/0600000009. URL `http://dx.doi.org/10.1561/0600000009`.

[114] R. Szeliski. *Computer Vision: Algorithms and Applications*. Texts in Computer Science. Springer, 2010. ISBN 9781848829343. URL `http://szeliski.org/Book/`.

[115] J.-C. Terrillon, M. David, and S. Akamatsu. *Automatic Detection of Human Faces in Natural Scene Images by use of a Skin Color Model and of Invariant Moments*. In Proceedings of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition, pages 112 –117, Nara, Japan, Apr. 1998. doi: 10.1109/AFGR.1998.670934.

[116] M. Teutsch. *Moving Object Detection and Segmentation for Remote Aerial Video Surveillance*. KIT Scientific Publishing, 2014. ISBN 978-3-7315-0320-0. doi: http://dx.doi.org/10.5445/KSP/1000044922. PhD thesis.

[117] M. Teutsch and W. Kruger. *Detection, Segmentation, and Tracking of Moving Objects in UAV Videos*. In Proceedings of the 9th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS), pages 313–318, Beijing, China, Sept. 2012. doi: 10.1109/AVSS.2012.36.

[118] The MathWorks Inc. *Matlab version 2017a*, 2017.

[119] The MathWorks Inc. *Matlab version 2018b*, 2018.

[120] T. Thormählen. *Zuverlässige Schätzung der Kamerabewegung aus einer Bildfolge*. Fortschritt-Berichte VDI: Informatik/Kommunikation. VDI-Verlag, Feb. 2006. ISBN 3-18-376510-1. PhD thesis.

[121] T. Thormählen and H. Broszio. *Automatic Line-based Estimation of Radial Lens Distortion*. Integrated Computer-Aided Engineering, 12(2):177–190, Apr. 2005.

[122] T. Thormählen, H. Broszio, and I. Wassermann. *Robust Line-Based Calibration of Lens Distortion from a Single View*. Proceedings of Mirage 2003 (Computer Vision/Computer Graphics Collaboration for Model-based Imaging, Rendering, Image Analysis and Graphical Special Effects), INRIA Rocquencourt, France, pages 105–112, Mar. 2003.

[123] P. H. Torr and D. W. Murray. *Outlier Detection and Motion Segmentation*. In P. S. Schenker, editor, Proceedings of the SPIE Sensor Fusion VI, volume 2059, pages 432–444. International Society for Optics and Photonics, Aug. 1993. doi: 10.1117/12.150246.

[124] R. Tsai. *A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology using Off-the-Shelf TV Cameras and Lenses*. IEEE Journal of Robotics and Automation, 3(4):323 –344, Aug. 1987. ISSN 0882-4967. doi: 10.1109/JRA.1987.1087109.

[125] M. Unger, M. Asbach, and P. Hosten. *Enhanced Background Subtraction using Global Motion Compensation and Mosaicing*. In Proceedings of the 15th IEEE International Conference on Image Processing (ICIP), pages 2708 –2711, San Diego, CA, USA, Oct. 2008. doi: 10.1109/ICIP.2008.4712353.

[126] VideoLAN Organization. *x265*, 2014–2016. URL `http://www.videolan.org/developers/x265.html`. v1.4–v1.9.

[127] C.-Y. Wu and P.-C. Su. *A Region of Interest Rate-Control Scheme for Encoding Traffic Surveillance Videos*. In Proceedings of the 5th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), pages 194–197, Kyoto, Japan, Sept. 2009. doi: 10.1109/IIH-MSP.2009.114.

[128] C.-Y. Wu, P.-C. Su, C.-H. Yeh, and H.-C. Hsu. *A Joint Content Adaptive Rate-Quantization Model and Region of Interest Intra Coding of H.264/AVC*. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), pages 1–6, Chengdu, China, July 2014. doi: 10.1109/ICME.2014.6890247.

[129] P. Xing, Y. Tian, T. Huang, and W. Gao. *Surveillance Video Coding with Quadtree Partition based ROI Extraction*. In Proceedings of the IEEE Picture Coding Symposium (PCS), pages 157–160, San Jose, CA, USA, Dec. 2013. doi: 10.1109/PCS.2013.6737707.

[130] S. Yahyanejad, M. Quaritsch, and B. Rinner. *Incremental, Orthorectified and Loop-Independent Mosaicking of Aerial Images Taken by Micro UAVs*. In Proceedings of the IEEE International Symposium on Robotic and Sensors Environments (ROSE), pages 137–142, Montréal, Canada, Sept. 2011. doi: 10.1109/ROSE.2011.6058531.

[131] H. Yalcin, M. Hebert, R. Collins, and M. Black. *A Flow-Based Approach to Vehicle Detection and Background Mosaicking in Airborne Video*. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), volume 2, page 1202, San Diego, CA, USA, June 2005. doi: 10.1109/CVPR.2005.29.

[132] H. Yang, H. Chen, Y. Zhao, and J. Chen. *Draft Text for Affine Motion Compensation*. Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 11th Meeting: Ljubljana, Slovenia, Document: JVET-K0565, July 2018. URL `http://phenix.it-sudparis.eu/jvet/`.

[133] B. Yao, X. Cai, and B. Wei. *Long-Term Background Reconstruction with Camera in Motion*. In Proceedings of the 2nd International Congress on Image and Signal Processing (CISP), pages 1–5, Tianjin, China, Oct. 2009. doi: 10.1109/CISP.2009.5301556.

[134] H. Zhang, H. Chen, X. Ma, and H. Yang. *Performance analysis of affine inter prediction in JEM1.0*. Joint Video Exploration Team (JVET) of ITU-T VCEG and ISO/IEC MPEG, 2nd Meeting: San Diego, CA, USA, Document: JVET-B0037, Feb. 2016. URL `http://phenix.it-sudparis.eu/jvet/`.

[135] X. Zheng, Z. Cao, and F. Wolf. *Aerial Photography Sequences for Video Coding Standard Development*. Joint Video Exploration Team (JVET) of ITU-T VCEG and ISO/IEC MPEG, 4th Meeting: Chengdu, China, Document: JVET-D0060, Oct. 2016. URL `http://phenix.it-sudparis.eu/jvet/`.

[136] X. Zheng, W. Li, Z. Cao, W. Su, C. Zhao, Y. Li, Z. Lorenz, H. Wu, Z. Du, and D. A. Hoang. *New Aerial Photography Sequences for Video Coding Standard Development*. Joint Video Exploration Team (JVET) of ITU-T VCEG and ISO/IEC MPEG, 6th Meeting: Hobart, Australia, Document: JVET-F0062, Mar. 2017. URL `http://phenix.it-sudparis.eu/jvet/`.

Curriculum Vitae of Holger Meuel

February 27, 1983	Born in Lübeck, Germany

Work

06/2010 – 12/2019	Scientific employee at *Institut für Informationsverarbeitung* (TNT) at *Gottfried Wilhelm Leibniz Universität Hannover*, Hannover, Germany

University education

10/2002 – 03/2010	Studies of electrical engineering, specialization communications engineering at *Technische Universität Braunschweig* (TU BS), Braunschweig, Germany, Graduation: Diplom-Ingenieur (Dipl.-Ing.)

School education

08/1995 – 06/2002	Secondary education at *Gymnasium Munster*, Munster, Germany, Graduation: Abitur
03/1994 – 06/1995	Secondary education at *Orientierungsstufe Munster*, Munster, Germany
08/1993 – 02/1994	Secondary education at *Orientierungsstufe Bergen*, Bergen, Germany
08/1989 – 07/1993	Elementary school *Eugen-Naumann Grundschule*, Bergen, Germany