

# Interactive interpretation of 3D surfaces in field-based geosciences using mobile devices - concepts, challenges and applications

Melanie Kröhnert<sup>a,\*</sup>, Christian Kehl<sup>b</sup>, Sophie Viseur<sup>c</sup>, Simon J. Buckley<sup>d,e</sup>

<sup>a</sup>*Institute for Photogrammetry & Remote Sensing, TU Dresden, Helmholtzstr. 10, 01069 Dresden, Germany*

<sup>b</sup>*anonymous*

<sup>c</sup>*Aix-Marseille Université, CNRS, IRD, Centre Européen de Recherche et d'Enseignement des Géosciences de l'Environnement (CEREGE) UM 34, Dept. Sedimentary and Reservoir Systems, 13001 Marseille, France*

<sup>d</sup>*Uni Research AS – CIPR, Nygårdsgaten 112, 5008 Bergen, Norway*

<sup>e</sup>*Department of Earth Science, University of Bergen, Allégaten 41, 5007 Bergen, Norway*

---

## Abstract

### STILL MISSING

*Keywords:* discrete geometry, surface reconstruction, volume reconstruction, surface parameterization, digital outcrops

*2010 MSC:* 00-01, 99-00

---

## 1. Introduction

Mobile devices are ubiquitously available in modern society. Apps for a diverse range of purposes emerged over the past decade. Such mobile devices are also increasingly applied for professional use and scientific purposes to solve computational tasks in outdoor- and field study environments. Geosciences such as hydrology, geology or glaciology rely on the documentation of field observations. In order to improve field study, these domains now attempt employing mobile devices as digital field instrument. This is illustrated in fig. 1, sketching the envisaged use of tablets and smartphones for geological and hydrological

---

\*Corresponding author

Email addresses: [melanie.kroehnert@tu-dresden.de](mailto:melanie.kroehnert@tu-dresden.de) (Melanie Kröhnert), [anonymous@anonymous](mailto:anonymous@anonymous) (Christian Kehl), [viseur@cerege.fr](mailto:viseur@cerege.fr) (Sophie Viseur), [Simon.Buckley@uni.no](mailto:Simon.Buckley@uni.no) (Simon J. Buckley)

10 purposes.

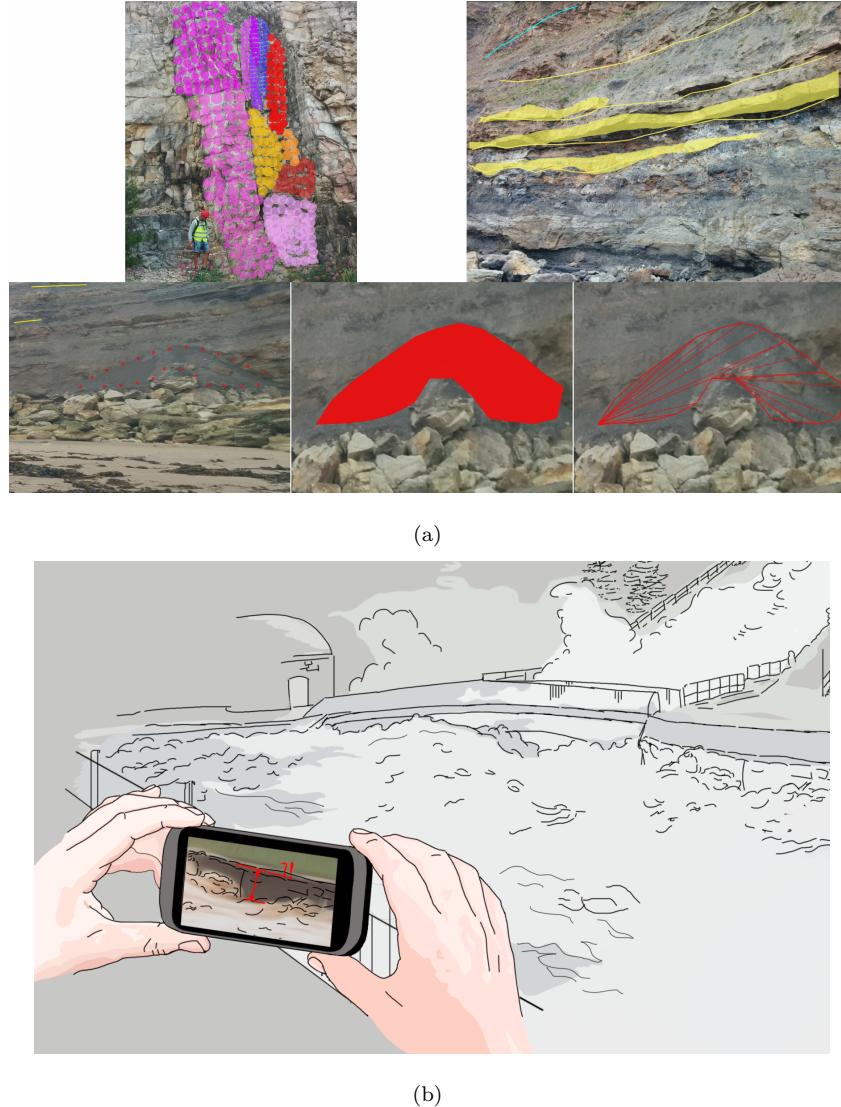


Figure 1: Illustrative examples for geological interpretation (a) and hydrological annotation (b).

Geoscience apps for assessing two-dimensional data have been available for several years now (see [] for examples). The availability three-dimensional base data and their increasingly-easy acquisition (e.g. via drones, structure from



Figure 2: Target application of field-based interpretation and annotation on mobile devices.

motion (SfM) [? ] and multi-view geometry [? ], satellite digital elevation  
models (DEMs)) makes it possible to assess 3D data in various scenarios. The  
application domains benefit from the development in acquisition technology  
as well as fundamental 3D data processing by being able to analyse the data  
on small-scale devices right in the field. Furthermore, crowdsourced data and  
volunteered geographic information (VGI) contribute to the geoscience data  
inventory, being acquired by citizen scientists.  
20

Domain-specific mobile software is required to realise the various domain  
expert requirements on the target devices, as off-the-shelf software insufficiently  
addresses the envisaged usage scenario or the domain-specific data types. This  
is not novel as domain-specific software is often needed to address specific needs  
and usage scenarios also on common workstations. What makes mobile software  
25 distinct is the modality of the device: the only commonplace interaction is via  
touch screen, data memory is a scarce resource when compared to workstations,  
and a prime concern of mobile usage is energy consumption while performing  
certain tasks. The advantage, on the other hand, is the mobility permitted by

30 tablets and smartphones, the array of sensors (with varying degrees of accuracy) available in the field and the high computational qualities compared to the device size. Developing geoscience, domain-specific mobile software requires to address these challenges (interaction, data handling, energy efficiency) while highlighting how the device advantages are used to support fieldwork tasks in  
35 original ways.

This article addresses the challenges of mobile sensor variability, their usage in image-to-geometry registration of pointcloud base data, and the related energy consumption in comparison to a digital surface model (DSM) base data mobile application. The technical research is approached via two use cases  
40 within the domains of surface hydrology and (petroleum) geology. The content covered in the article is a significant extension of earlier published research [?], focussing on extensive measurements to verify the reasoning and statements of previous studies.

The sections within this article adhere to the following structure: First, different 3D surface data representations are briefly discussed which are employed in hydrology and geology. Second, algorithmic baseline concepts that are key for 3D base data interaction on mobile devices are introduced, summarising project-internal development by the authors as well as referencing key literature on the subject. Third, the challenge of mobile sensor positioning and orientation is addressed with an in-depth study measuring mobile sensors and comparing their accuracy and variability to professional IMU reference data. Fourth, power consumption of such 3D surface data mobile applications is addressed via measurements and analysis of energy efficient control parameters. Subsequently, a section discussed how available mobile systems are used in surface hydrology  
50 (i.e. water level gauging) and petroleum geology (i.e. field interpretation) to improve data analysis and integrate outdoor measurements in digital workflows. Then, the article is finalized with some concluding remarks and discussions for future developments in this research trajectory.  
55

## 2. 3D base data representations

Various representation forms for 3D terrain data are available. While early digital systems used gridded DEMs for their simplicity and compact storage [? ? ], DSMs and triangulated irregular networks (TINs) are dominating most terrain-based systems for application-specific analysis [? ? ] at the moment. A useful example can be seen in [? ] for glaciology, where the authors use a triangulated digital surface model to represent a Patagonian glacier front. For triangular surfaces, it is important to distinguish geometrically valid TINs, organised as piecewise-linear complexes (PLCs), from polygon soup surfaces (fig. 3). While the latter is often employed in early stages of mesh-based software systems due to its simplicity and ease of implementation, valid TINs are employed in mature project stages as automated analysis methods (e.g. auto-interpretation, volume derivation) require clean surfaces with coherently outward-oriented surface normals.

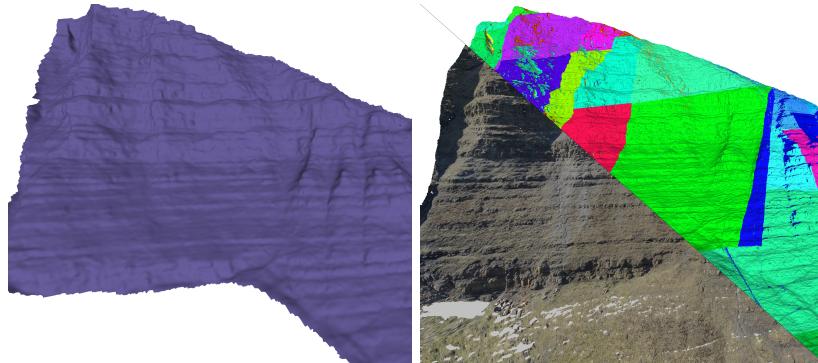


Figure 3: Illustrative distinction between valid TINs (consisting of one exclusive, smooth, closed surface) and polygonal soups. Non-textured model parts are coloured with respect to their actual segment number. Images taken from [? ].

In geoscience domains such as petroleum geology, texture- and color information are vital for interpretation- and analysis tasks. In these cases, as demonstrated by Buckley et. al [? ] and Caumon et. al [? ], the surface is

supplemented with photographic information via texture projection. The models are referred to as digital outcrop models (DOMs) (see fig. 4 as reference depiction).



Figure 4: Example of a DOM as textured triangular surface.

In contrast, other geoscience domains, such as hydrology and free surface flow  
80 management, used georeferenced laser scanner point clouds and coloured point  
data streams provided by terrestrial photogrammetry for small- or unmanned  
aerial vehicle (UAV) for large-scale study cases. The colour component of the  
base data is either provided by auxilliary photographs or embedded as part  
of the point cloud reconstruction (e.g. glsSfM). The point set surfaces (PSSs)  
85 support tasks like coastal monitoring [? ? ], soil erosion and rain-induced  
landslide observation, and even monitoring river topography [? ] and flood  
protection management [? ]. Nevertheless, new approaches for low-cost and on-  
the-fly river monitoring [? ] arise due to globally increasing flash flood events  
after heavy rainfalls [? ] that are further addressed in section 6.1.

90 Since SfM became state of the art in geosciences, the acquisition of (true- ) coloured "point cloud" models is not that difficult and commonly employed because of its rapid processing, compared to conventional approaches like terrestrial laser scanning (TLS). Regarding 3D annotation, nearest neighbour analysis provides an opportunity whereby surface triangulation can be avoided.

95 The above representation forms are also valid for mobile device software. Because of the limited processing speed of mobile chipsets, the usage of point clouds appears most common within the graphics literature (e.g. Garcia et. al

[? ]). The sparse vertex distribution in point clouds can cause problems in the data analysis, which is why DEMs have seen a revival in the mobile computing domain. DEMs provide dense, closed geometric models that can be rendered and processed efficiently. Furthermore, because of the smaller device memory, the possible compression options for point clouds and DEMs are advantageous.

### 3. Algorithms

This section demonstrates novel- as well as existing algorithms and methods on mobile devices that are needed for case-specific field-based analysis within the geosciences. The effectiveness of each algorithm depends on the applied model representation and the target usage.

#### 3.1. Mesh-based rendering

Rendering a surface model in this context refers to the image generation of the 3D base data by projective rasterization to the 2D image plane of a virtual camera. This process is performed on mobile devices for the purpose of model presentation as well as for the generation of a synthetic reference image for image-to-geometry registration.

Algorithms for rendering textured triangulated surfaces are well-known amongst practitioners. In the common rendering pipeline, the textured mesh is transferred as a set of (attributed) vertices and primitive sets (e.g. triangles, polygons) to the graphics processing unit (GPU). The virtual camera is set up using the pre-defined view projection matrix while the graphics primitives are repositioned using the model-related transformation matrix. The rasterizer projects the available 3D information into the camera plane and performs hidden-surface removal. The result is a discrete-space pixel representation. Available textures are mapped as images on the surface using the texture coordinate vertex attributes. The mesh-based rendering algorithms employed on desktop computers are analogous to mobile devices, whereas the technological details are posing the actual challenges.

### 3.2. A novel approach to mobile point-based rendering

In comparison to mesh-based rendering, simple point projection seems to be a nice alternative, saving computational resources and efforts for post-processing regarding outlier removal. Thus, we simply project object points onto an image plane using perspective projection, assuming a distortion-free ideal camera with centred principle point. Thus, the camera matrix  $\mathbf{K}$  equals identity matrix  $\mathbf{I}$  and can be neglected.

First, applying a six-parameter transformation transfers three-dimensional object points from world reference frame  $\vec{X}_W$  into a 3D camera system  $\vec{X}_c$  using

$$\vec{X}_c = \mathbf{R} (\vec{X}_W - \vec{X}_0) \quad (1)$$

where  $\mathbf{R}$  is a  $3 \times 3$  orthonormal rotation matrix and  $\vec{X}_0$  the translation vector to camera's projection center. For simplicity, the usage of the planar Cartesian UTM system with  $x$  pointing to the east and  $y$  pointing to the north with respect to the prevalent zone number. For  $z$  component, the height over the Earth Gravitational Model 1996 (EGM96) is advisable to use.

Counting for homogeneous coordinates, we can describe the relation between camera  $\vec{X}_c$  and image coordinates  $\tilde{x}$  involving their depth components.

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \\ c_c \end{pmatrix} = \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} \quad (2)$$

For camera's imaging plane, we introduce the constant  $c_c$  that defines the distance between camera's sensor and its projection center in [mm], which equals focal length  $f$ . To separate camera sensor system and image system, we use the term  $c_c$  when referring to the sensor [mm], and  $f$  for digital image coordinates [px]. For conversion,  $c_c$  must be divided by the sensor's pixel pitch. The normalization of the projected points to homogeneous coordinates is key in the further processing. This is analogous to the image-to-geometry projection in eq.

<sup>145</sup> 6, where the projection variable  $w$  is replaced with the camera constant  $c_c$ .

$$\vec{X}_{Cam} = \begin{pmatrix} \frac{\tilde{u}}{c_c} \\ \frac{\tilde{v}}{c_c} \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{x_c}{z_c} \\ \frac{y_c}{z_c} \\ 1 \end{pmatrix} \quad (3)$$

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} = \begin{pmatrix} \frac{x_c}{z_c} \cdot c_c \\ \frac{y_c}{z_c} \cdot c_c \end{pmatrix} \quad (4)$$

For a final transformation of 2D sensor coordinates into image pixels, we need to shift the image coordinate system to the origin to left upper corner and scale the coordinates from global units in meters per pixel using  $p_s$ . Thus, we derive image coordinates  $(u, v)$  for an ideal camera using

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{p_s} \begin{pmatrix} \frac{x_c}{z_c} \cdot c_c - u_0 \\ \frac{y_c}{z_c} \cdot c_c - v_0 \end{pmatrix} \quad (5)$$

### <sup>150</sup> 3.2.1. Calculation of 3D bounding box of interest and image plane

In the mobile rendering scenario, we need to define a region of interest regarding 3D point projection in order to cull the render content of the virtual camera to the user's field of view (figure 5). The view frustum's bounding box corner points are calculated using the position and orientation from fused smartphone sensors. Thereby it must be noted that only the heading is used for estimating viewing direction; tilt and roll are excluded. Because of uncertainties regarding exterior information (section 4), the bounding box must be expanded to cover more object space than described by the sensors as well as the camera's field of view. Because of possible noise due to positioning, constants  $r$  and  $dh$  describe the domain of projection center's uncertainties parallel to image plane. For errors in depth, we define the correction  $c = \frac{r}{\tan(H)}$  for shifting the projection center along camera axis.

The box is widened by the horizontal  $H$  and vertical  $V$  opening angles with a fixed depth  $d$ . In order to generate reference data for image-to-geometry

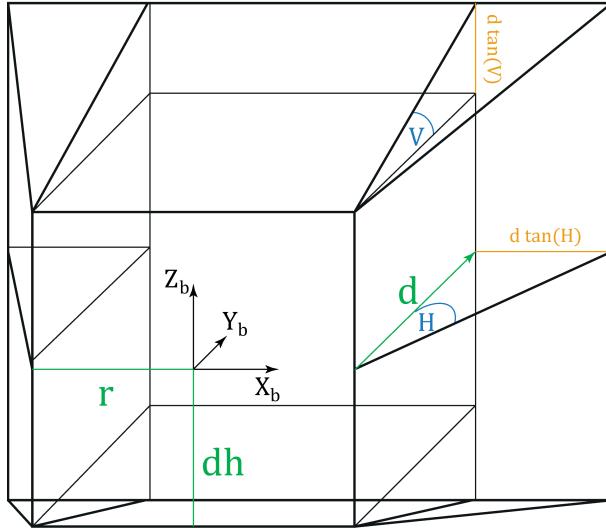


Figure 5: Bounding box definition.

<sup>165</sup> registration to annotate 3D data by mobile imagery, the lateral accuracy given by the mobile positioning system as well as the prevalent camera characteristics solve for the mentioned parameters. For camera based gauging, we set  $d = 200[m]$ . Additional tiling of the 3D base data is advisable for a rapid geometry-in-frustum containment checks.

<sup>170</sup> Using the defined frustum of a pyramid as region of interest with a local reference system, the image plane for 3D point rendering can be defined by perspective projection of the remote  $xz$  plane (5) with

$$\vec{X}_b = \begin{pmatrix} -r - d \tan H \\ d \\ dh + d \tan V \end{pmatrix}$$

for the bounding box' background plane upper left and

$$\vec{X}_b = \begin{pmatrix} r + d \tan H \\ d \\ -dh - d \tan V \end{pmatrix}$$

for the lower right corner. Its height equals the height component in the  
 175 world reference frame  $z_w$ . Because of pyramid frustum, we subsequently eliminate points outside the near- and far clipping plane.

### 3.2.2. Pyramid approach for depth filtering

Because of a limited range of pixels with defined size inside a image plane it seems to be obvious that, in most cases, more than one 3D object points  
 180 corresponds to the same image pixel. Due to inhomogeneous coordinates it is not possible to figure out afterwards which points are in foreground compared to the camera distances and which ones are behind and thus not visible. This problem can easily be solved during point cloud projection described above by a simple camera-to-object distance check. However, one problem still remains in  
 185 case of e.g. glass fronts with lacking information (in TLS due to deflected lidar or SfM when having homogeneous surfaces) or small archs (see figure 7). Then, points might be visible pointing away from camera projection center. On the one hand, point normals may solve the problem but due to the data acquisition technique and the model's complexity, they are more or less easy to derive [? ].

190 Scale-space image pyramids are a nice alternative approach to overcome the issue. Our scale space is constructed from multiple synthetic images via step-by-step adjustment of  $k \cot p_s$  (see eq. 5) with  $k = 2$ , resulting in halve the number of image rows and columns per layer. Then, the algorithm verifies if two pixels corresponds in two subsequent layers, preserving edges (figure 6,7).

### 195 3.2.3. Filling gaps due to missing points

Because of pixel size and image plane definition with a specific resolution (i.e. depending on smartphone full-scale camera's resolution for image regisra-

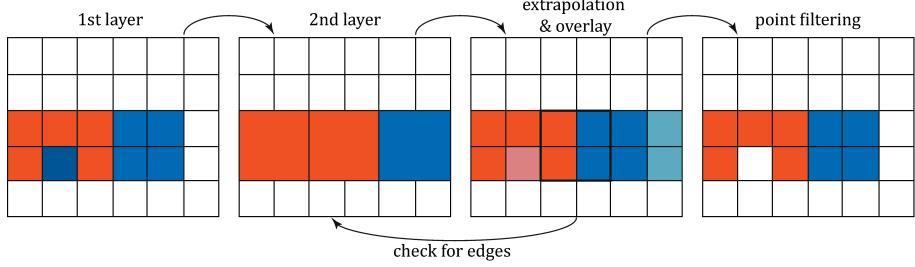


Figure 6: Visualisation of hierarchical depth filtering to handle point occlusions.

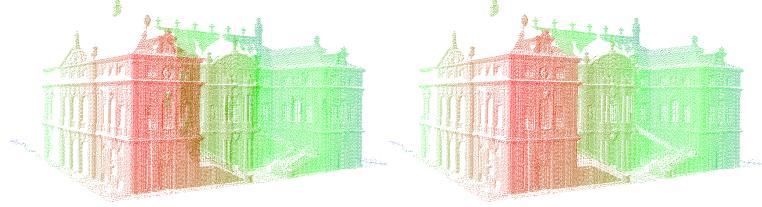


Figure 7: Left, actually obscured visible 3D points close to archs and windows. Right, edge preserving result after filtering.

tion purposes) there will still be gaps between projected points (see figure 7, right). In order to fill these gaps, we recommend to use a simple nearest neighbour approach using binary search [? ] in the 3D domain to fill these gaps, applying weights to average 3D points color attributes depending on their euclidean distances. For this, thresholds for maximum distances between 3D points must be applied to avoid unreasonable gap-filling. Exemplary for use case in section 6.1, a before-after comparison of the gap filling is shown in figure 8.

### 205 3.3. Image-to-geometry registration

Image-to-geometry algorithms aim at registering 2D images to a given 3D surface, providing a transformation from the 2D image coordinate system to 3D model coordinate system as follows:



Figure 8: Fill image gaps using nearest neighbour binary search in 3D domain.

$$P' \in \mathbb{R}^3 = \begin{pmatrix} u \\ v \\ w \end{pmatrix} = [R_{3,3}|T_{1,3}] \cdot P \quad (6)$$

$$P \in \mathbb{R}^3 = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (7)$$

$$P' \in \mathbb{R}^2 = \frac{P' \in \mathbb{R}^3}{w} \quad (8)$$

$R_{3,3}$  – rotation matrix;  $T_{1,3}$  – translation vector

210  $P$  – point of the object;  $P'$  – (projected) point in image plane

$(u, v, w)$  – image plane coordinates;  $(x, y, z)$  – world coordinates

Using this coordinate system transformation in combination with a known interior camera orientation, it is possible to project each image on the surface. Specific objects outlined within the image can also be mapped on the surface.

215 Amongst the published literature, feature-based registration algorithms are most common. Here, salient points (e.g. SIFT, SURF, Harris corners, see Mikolajczyk and Schmid [? ] for details) or edges within the photograph and rendered image of the target 3D model are used to establish an image-to-image correlation.

<sup>220</sup> In order to establish a 2D–3D correlation, there are two prevalent approaches available: for triangle mesh models, the 2D feature locations within the rendered image are raycasted using the virtual camera’s vanishing point, the imaging plane, and the 3D surface model (see fig. 2 in [? ]). The intersection between the ray and a triangle within the mesh results in the correlated 3D coordinate  
<sup>225</sup> of the 2D feature. An alternative approach is needed for point-based models because raycasting does not apply to point representations (i.e. points cannot be intersected directly due to their zero-extent). The alternative approach often applied (see [? ? ? ? ]) employs smart rendering techniques that virtually expand points into an area features (e.g. blobs, disks or spheres), being subsequently  
<sup>230</sup> rendered into a depth map. Afterwards, the 3D coordinate of a 2D feature can be inferred directly from the depth map. The novel algorithm introduced in section 3.2 also relates to this category of 2D–3D correspondence. Though cleverly utilising graphics technology, this approach is limited by an accuracy-to-speed trade-off: low-resolution and low-quantisation depth maps introduce artificial  
<sup>235</sup> accuracy errors in the registration process, whereas high-resolution depth maps cost considerable performance in the image generation.

<sup>240</sup> When 2D–3D point pairs are established, the coordinates are normalized and put into a least-squares optimization system, where the target is to determine the exterior camera parameters ( $t_x, t_y, t_z, \psi, \varphi, \theta$ ) from the 2D–3D point-based equation system. Non-linear optimisation systems (e.g. Levenberg-Marquardt) are applied to estimate the desired parameter set [? ]. The whole process can be executed on mobile devices [? ]. One of the prevalent practical challenges when employing feature-based image-to-geometry registration is to achieve a reliable feature correlation, which is achieved by introducing application-specific  
<sup>245</sup> constraints (e.g. horizon alignment, straight-edge enforcement or object outlines).

<sup>250</sup> Feature-based registration is the most common approach for establishing image-to-geometry correlation on mobile devices due to its implementation simplicity, its rapid execution speed, its option for application-specific constraints and the wealth of available code that can be used. Application examples are

ample within the literature, ranging from augmented reality [? ? ] over field geology [? ? ] to surface hydrology [? ? ]. These mobile apps utilize the open-source library *OpenCV4Android*<sup>1</sup>, which is also employed in this work<sup>2</sup>. Problems in real-world cases are caused by imaging variances, resulting in reduced reliability (i.e. failing to determine any camera parameters) and stability (i.e. determining different parameters for the same sets of images) [? ]. A completely alternative technique to feature-based methods is Mutual Information (MI) [? ? ]. MI performs a pixel-wise comparison between the photo  $I_{2D}$  and the 2D rendering of the 3D scene  $I'_{3D}$  and aims at minimizing the image discrepancies (i.e.  $\text{argmin}\Delta(I_{2D}, I'_{3D})$ ). The technique uses information theory quantities such as self-information and entropy in order to compare the similarity of both image. In contrast to feature-based techniques, MI faces challenges in the optimization process: the optimization of a 7 degree-of-freedom equation system ( $t_x, t_y, t_z, \psi, \varphi, \theta, f$ , for  $f$  being the focal length) is unstable and prone to rest in local function minima. Only few optimisation solvers are known that can solve such equation systems reliably and provide stable results - most notably NEWUOA (i.e. Powell's method[? ]) used by Corsini et al. [? ]. None of the advanced solvers is available in modern- and mobile-device programming languages, thus the use of MI on mobile platforms is currently prohibited.

#### 270 4. Sensors

What is the great difference between former mobiles and today's smartphones? Smartphones have many inbuilt sensors such as acceleration measurement units, compasses or gyroscopes, playing increasing rolls not only to have control over display or camera rotation. In the following we assess orientation accuracy and precision by applying different sensor fusion methods, which in their turn influence image-to-geometry registration. Furthermore, we give a short review over smartphones' positioning quality in relation to 3D annotation.

---

<sup>1</sup>OpenCV4Android 2.4.10 - <https://opencv.org/platforms/android/>

<sup>2</sup>OpenCV4Android extensions at [https://github.com/CKehl/opencv4Android\\_extension](https://github.com/CKehl/opencv4Android_extension)

#### *4.1. Localization*

Compared to the years 2008 and 2009, sales volume for navigation systems  
280 declined sharply and constantly by approximately 70 percent compared to 2017  
in Germany<sup>3</sup>. One of the most important factors behind this may lie in the  
distribution of smartphones with inbuilt positioning systems, providing quite  
interesting alternatives to former navigation systems. For this, most of today's  
285 smartphones are equipped with absolute global positioning system (aGPS) re-  
ceivers that are able to receive data from American GPS, Russian GLONASS  
and increasingly European GALILEO as well as Chinese BAIDOU. Even within  
the geosciences, smartphones gain more and more popularity e.g. for mobile  
mapping [? ? ], or actually 3D reconstruction [? ? ? ].

Based on these facts, many research groups recently discussed the potential  
290 of smartphone localisation strategies whereby we want to focus on outdoor use  
cases based on global navigation satellite systems (GNSSs). Blum et al. (2013)  
[? ] observe the positioning for Android smartphone Samsung Galaxy Nexus  
and Apple Iphone 4 with different environmental conditions. Walking through  
the city they get lateral accuracies of about 10-15m close to buildings no taller  
295 than three stories. Near skyscrapers, errors of about 30m should be expected  
with local extremas up to 60m. Similar things are published by Fritsch et al.  
(2011) [? ] who determined a overall accuracy for Android smartphone HTC  
Hero of 15-25m valid in 95 percent of cases which was also estimated by Zhu  
et al. (2013) [? ] and Zandbergen et al. (2011) [? ]. Exemplary for open  
300 spaces, Meek et al. (2013) [? ] observe an average global positioning system  
(GPS) accuracy of 6.8m using a Google Nexus S smartphone. However, height  
estimation seems to be more critical where [? ] name error margins for altitude  
determination using smartphone's inbuilt aGPS which seem to be 2.5 times  
more than the horizontal component and recommend the alternative usage of

---

<sup>3</sup>see statista.com Survey: *Sales development in Germany for navigation systems since 2005* <https://de.statista.com/statistik/daten/studie/3902/umfrage/entwicklung-der-verkaufszahlen-von-navigationsgeraeten-seit-2005/>

<sup>305</sup> barometric approaches, providing height accuracies up to 3m. Unfortunately, only a few of common smartphones have inbuilt barometers and reference data, necessary for barometric altitudes, is quite difficult to obtain.

#### 4.2. Location sensitivity

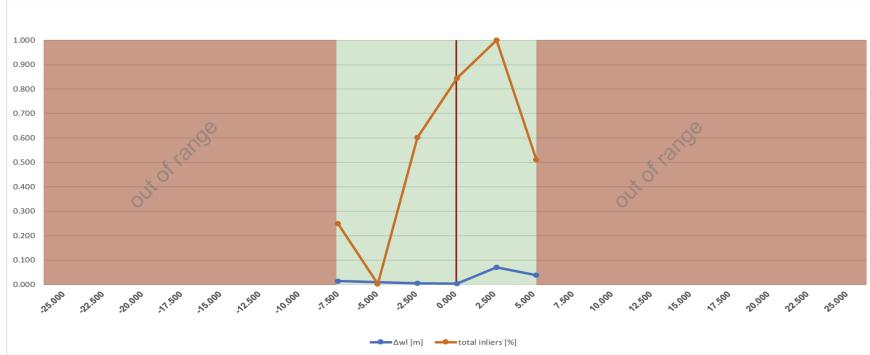
<sup>310</sup> Pre-knowledge about an image's position is a necessary prerequisite for image-to-geometry registration. Thus, we are asking for how do uncertainties in positioning affect feature detection and furthermore the matching results. We observe the behaviour on the example of *Open Water Levels* using manually registered reference data to derive the prevalent water level. Afterwards, we change the image's real position, defined in UTM33 WGS84 reference frame, in <sup>315</sup> steps of 2.5 m up to a deviation of 25m for northing, easting and height component and compare the detected water line with ground truth data from an administrative water gauge (see figure 9(a), 9(b)) having an enclosing DEM.

<sup>320</sup> Surprisingly, all components are rather equally affected by erroneous locations which rapidly leads to infeasible matchings when location differs more than 2.5m/5m (northing/easting). For height component, the results are quite unstable regarding inlier occurrences and results for water levels.

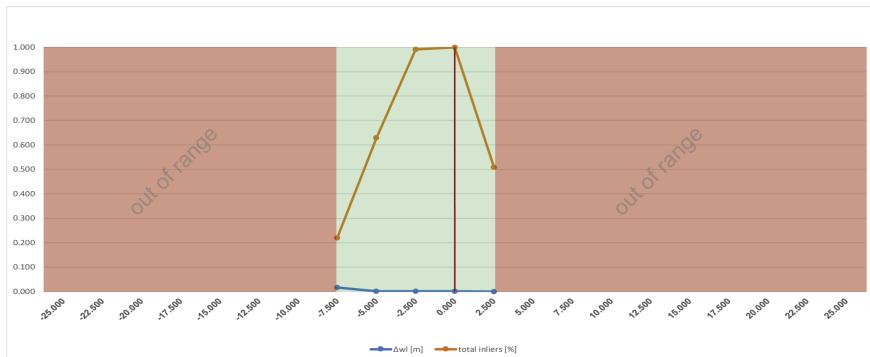
<sup>325</sup> Compared to observed accuracies of smartphone inbuilt GNSS, the results refer to be non-negligible issues. Thus, in *Open Water Levels*, the user can call Google Maps (if access to the internet is permitted) for manual position refinement, whereas Geological Registration and Interpretation Toolset (GRIT) enables repositioning based on locally stored DEM data for user-guided repositioning. To correct the even more erroneous height measurements, one option is the use of external DEMs included in GRIT or invoking third party models e.g. via Google Elevation API<sup>4</sup>, as it is implemented in *Open Water Levels*.

---

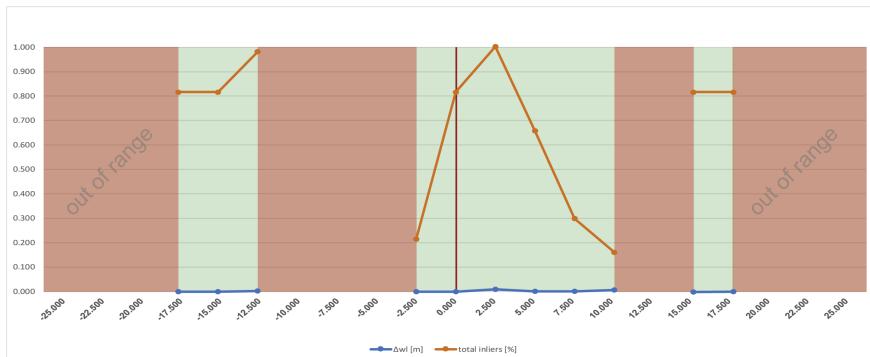
<sup>4</sup>Developer Guide - Google Elevation Api, <https://developers.google.com/maps/documentation/elevation/intro?hl=de>



(a) Easting observation



(b) Northing observation



(c) Height observation

Figure 9: Behaviour of image-to-geometry intersection depending on location uncertainties. Tuning the easting, northing and height component independently from each other whereas the others remained unchanged.  $\Delta w_{wl}$ .. absolute error for water level in meters, total inliers.. number of image features in feature-based matching displayed in percent depending on most occurrence, red/yellow/green area.. range of accuracy.

330 4.3. Orientation

Nothing to say that low-cost sensor systems for orientation determination, as they are integrated in smartphones, may not have precision and stability compared to professional initial measurement units (IMUs). Thus, we put forward the hypotheses that noise in smartphone sensor stability as well as their accuracies may not be in ranges comparable to navigation systems in autonomous navigation applications.

To give some basis, a smartphone orientation unit never depends on only one single sensor. It commonly consists of several components like accelerometers, magnetometers, gravity sensors or gyroscopes that measure in all three axes of the device. Due to single characteristics, sensors may complement each others, e.g. both, gyroscopes and accelerometers, measure the rotation of the smartphones in device-specific coordinate systems where gyroscopes are quite precise but suffer from drift effects. On the contrary, accelerometers are less sensitive for drifts but have poor signal-to-noise ratios. Applying sensor fusion helps to compensate for these negative characteristics in an ideal way.

Thus, Android divides sensors in two categories where, on the one hand, hardware sensors are true inbuilt components and on the other, virtual or soft sensors stand for fused hardware to generate a new synthetic sensor. That's why, when using Kalman filter approaches with different weights, more stability or accuracy can be given to smartphone's orientation.

Referring to Pacha [?], he presents two alternative virtual sensors additionally to Android's Kalman filtered *Rotation Vector*<sup>5</sup>, where the *Improved Orientation Sensor 1* should be more precise than Android's Rotation Vector but less stable whereas *Improved Orientation Sensor 2* seems to be less accurate but more robust. In the following, we check the three sensor types *Android Rotation Vector*, *Improved Orientation Sensor 1* and *Improved Orientation Sensor 2* for their stability and accuracy compared to a initial navigation system (INS)

---

5see Android Developers guidance for sensor event values at <https://developer.android.com/reference/android/hardware/SensorEvent.html#values>

that is commonly used for car and UAV navigation. For this, we compare measurements taken at three different times for the devices Google Nexus 5, Samsung Galaxy S8 and the IMU Spatial from the Australian company Advanced Navigation v6.1 (for sensor specifications refer to tables 1<sup>6</sup> & 2<sup>7</sup>).  
<sup>360</sup>

Table 1: orientation sensor specifications for Google Nexus 5 and Samsung Galaxy S8

	Google Nexus 5	Samsung S8
accelerometer/ gyroscope	InvenSense MPU- 6515 (6-axes)	ST Microelectronics LSM6DSL (6-axes)
magnetic compass	Asahi Kasei AK8963	Asahi Kasei AK09916C
pricing	(-)	6.50 USD

Table 2: IMU specifications for Advanced Navigation Spatial v6.1

roll & pitch accuracy (static)	0.1
heading accuracy (static)	0.5
roll & pitch accuracy (dynamic)	0.2
heading accuracy (dynamic with GNSS)	0.2
heading accuracy (dynamic, magnetic only)	0.8
pricing	3.500 USD

Each measuring epoch comprises six parts. We assume that the sensors  
<sup>365</sup> will show different behaviour when they are rigid or in motion. Furthermore, magnetic disturbances may influence the heading angle which mainly depends on the magnetic compass sensor. Additionally, we assume that results of sensors

---

<sup>6</sup>see <https://technology.ihs.com/api/binary/592077> for Samsung Galaxy S8 and <https://de.ifixit.com/Teardown/Nexus+5+Teardown/19016> for Google Nexus 5

<sup>7</sup>see <http://www.advancednavigation.com.au/product/spatial#specifications>

in rest may be slightly better when they are able to calibrate themselves after a short running time in motion. All observations are independent from each other, measured over time periods of more or less 2min 30sec.

For comparison, smartphone and IMU are mounted on an inflexible non-metallic wooden stick at a distance of 1.0m (to avoid mutual magnetic interferences) with aligned (native) coordinate systems (see figure 10). Only for pitch angle the opposite direction of rotation must be kept in mind.

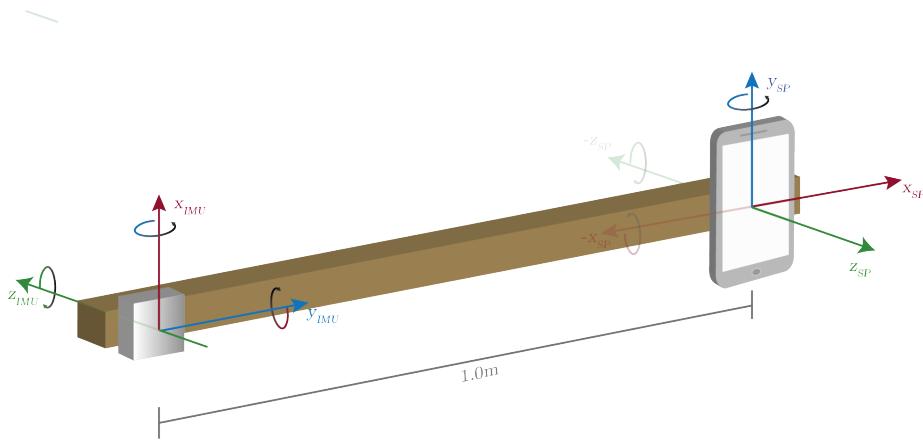


Figure 10: Measurement setup to observe smartphone sensors accuracies and precisions. Heading (green circle).. rotation around positive/ negative z-axis referring to IMU's/ smartphone's native coordinate systems pointing away from sky. Pitch (red circle).. rotation around positive/ negative x-axis in IMU's/ smartphone's reference frame pointing out of smartphone's display to the left. Roll (blue circle).. rotation around positive x-axes for both IMU and smartphone pointing to (true) north when smartphone is lying on a flat desk. Distance between devices on wooden stick is 1.0m.

In the following figures 12, 13, 14, orientation tracking for Samsung Galaxy S8 in comparison with INS (1st run) is documented whereas the others can be found in the supplementary material. The related figures collectively use the same legend, which is given in fig. 11.

Results for the indoor measurements of the smartphones (i.e. Samsung Galaxy S8 and Google Nexus 5) show very good agreement as can be seen in the following statistics (fig. 15) for both first runs respectively.

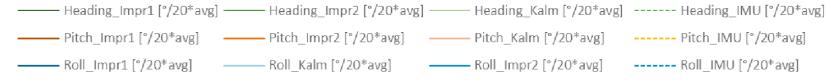
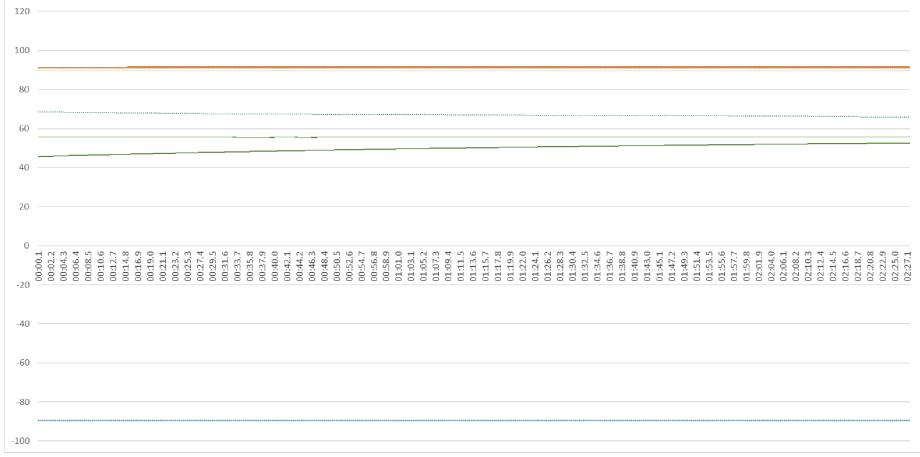


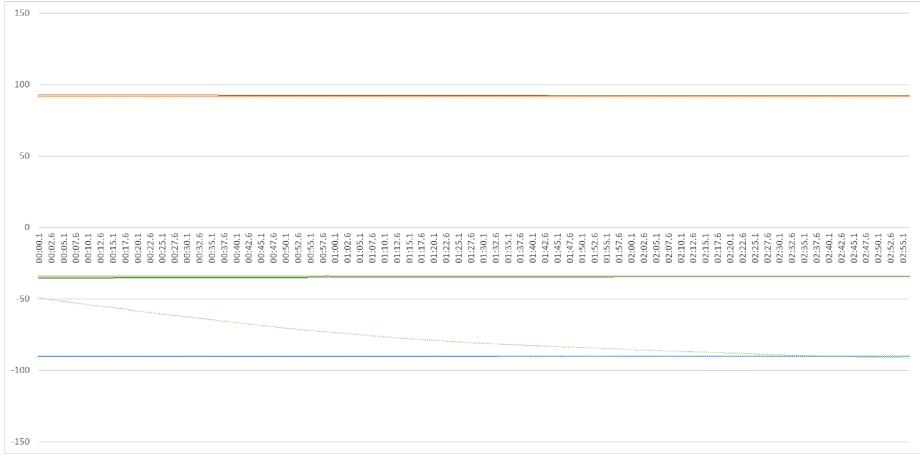
Figure 11: Legend applicable for figure 12 to 14, contrasting `heading`, `pitch` and `roll` for different sensor fusion methods.

Surprisingly, the results of all three virtual sensors of both smartphones show independently from each other almost the same behaviour as visualised in the figures 12, 13, 14. Especially *Improved Orientation Sensor 1* and *Android Rotation Vector* are very similar whereas *Improved Orientation Sensor 2* seems to be slightly more stable as expected. Beside this, note that pitch and especially roll angles of both smartphones are close to the orientation of the reference IMU even if the correlations in figure 17 shows little similarity. This correlation issue is caused by slight synchronisation errors visible e.g. in the graphs of figures 13(b) and 14(a) which may be caused by averaging sensor values to establish comparability between both smartphones and IMU respectively by UTC time.

Considering root mean square errors (RMSEs) in figure 16, good agreement with related studies can be resolved. Blum et al. (2013) [?] determined orientation errors up to  $30^\circ$  for heading with significant drifts accelerating over  $4^\circ/\text{s}$  while walking in the streets with an iPhone 4 for several minutes. Furthermore, Kok et al. (2017) [?] show how magnetic disturbances affect all three orientation angles referring to errors of more than 30 degrees (especially for heading) which is recognizable in our studies too. They also figured out that the heading's accuracy is relatively low compared to roll and pitch, which are considered accurate. They justify the results with a worse signal-to-noise ratio of the magnetometer compared to that of the accelerometer and the local magnetic field vector, being commonly used for the compass direction that points to magnetic instead of true north. In our case, we use the magnetic field to compute true north by location-dependent declination adding to sensor's heading and thus correct heading pointing to true north as it appears for our reference IMU.



(a) Construction fixed



(b) Construction fixed + magnetic disturbances

Figure 12: Deviation between orientation angles [deg] from Samsung Galaxy S8 (dotted line) and Advanced Navigation Spatial v6.1 IMU (straight line) captured with devices at rest. **Heading**, **pitch** and **roll** captured with frequencies of 20Hz and matched by UTC time every 0.1sec.

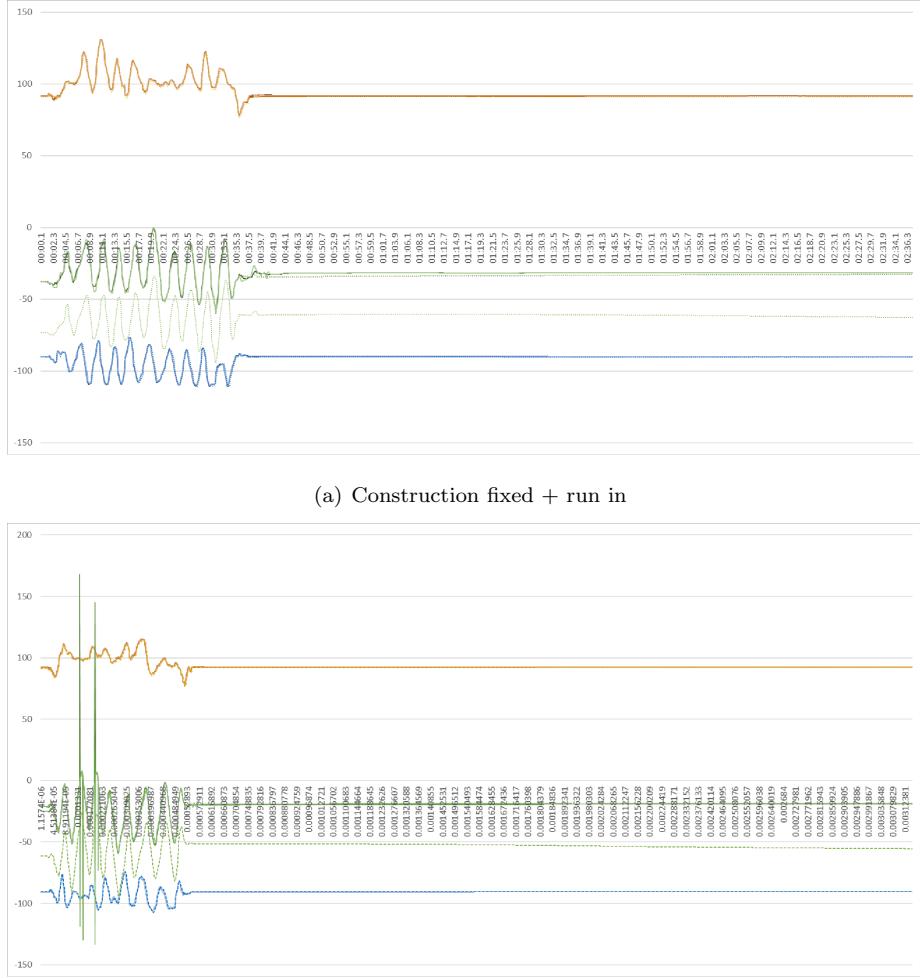
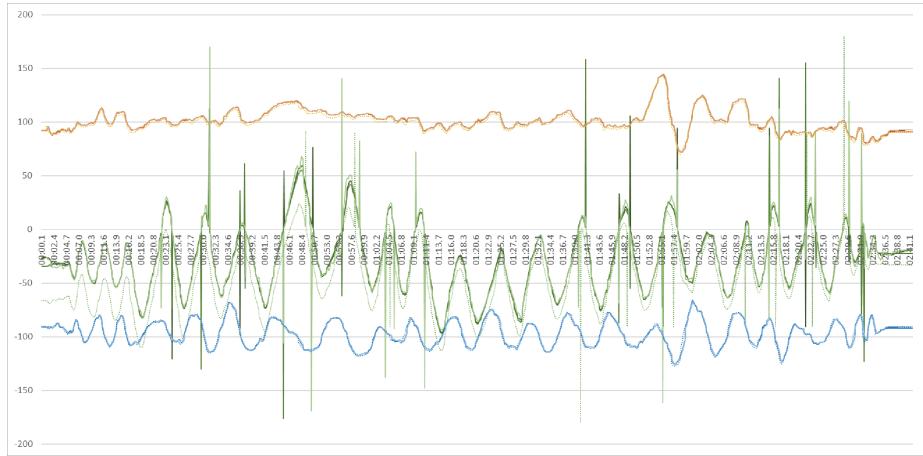
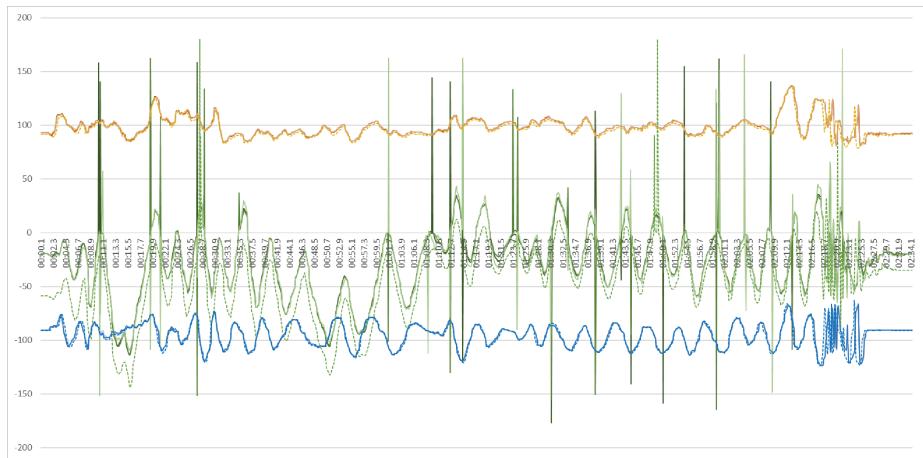


Figure 13: Deviation between orientation angles [deg] from Samsung Galaxy S8 (dashed line) and Advanced Navigation Spatial v6.1 IMU (straight line) captured with devices at rest after a short run-in period (device rotation around all three axes). **Heading**, **pitch** and **roll** captured with frequencies of  $20Hz$  and matched by UTC time every  $0.1sec$ .



(a) Construction in motion



(b) Construction in motion + magnetic disturbances

Figure 14: Deviation between orientation angles [deg] in-motion from Samsung Galaxy S8 and Advanced Navigation Spatial v6.1 IMU. **Heading**, **pitch** and **roll** captured with frequencies of  $20Hz$  and matched by UTC time every 0.1sec.

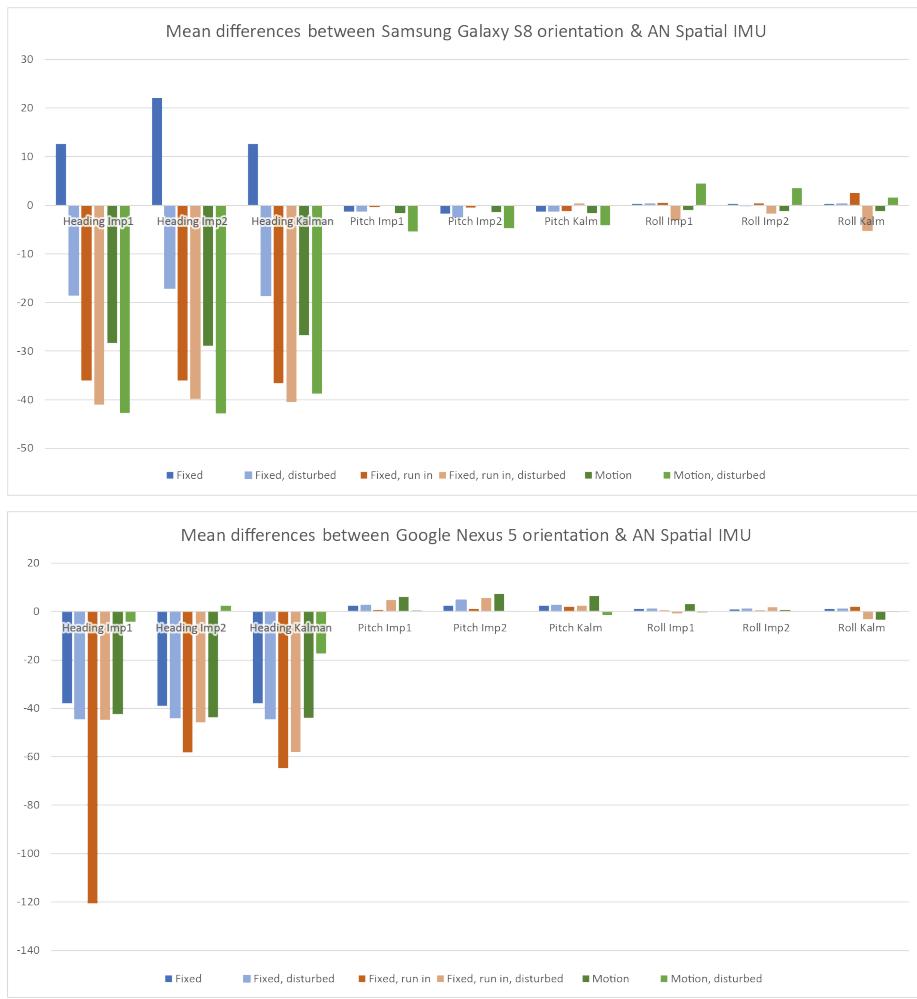


Figure 15: Mean deviations between the smartphones Samsung Galaxy S8 and Google Nexus 5 for each orientation angle respectively captured by the three virtual sensors *Android Rotation Vector*, *Improved Orientation Sensor 1* and *2*.

	Heading Imp1	Heading Imp2	Heading Kalm	Pitch Imp1	Pitch Imp2	Pitch Kalm	Roll Imp1	Roll Imp2	Roll Kalm
Fixed	12.63	22.10	12.63	-1.36	-1.71	-1.32	0.30	0.25	0.30
Fixed, magnetic disturbed	-18.50	-17.07	-18.63	-1.25	-2.41	-1.22	0.26	-0.12	0.26
Fixed, run in	-36.10	-36.03	-36.62	-0.37	-0.44	-1.18	0.51	0.42	2.53
Fixed, run in, magnetic disturbance	-41.04	-39.83	-40.52	-0.09	-0.09	0.34	-3.03	-1.81	-5.34
Motion	-28.40	-28.85	-26.80	-1.69	-1.43	-1.68	-1.01	-1.18	-1.19
Motion, magnetic disturbed	-42.69	-42.88	-38.76	-5.37	-4.76	-4.16	4.49	3.46	1.55
RMSE	31.93	32.49	30.85	2.42	2.37	2.03	2.27	1.68	2.55

	Heading Imp1	Heading Imp2	Heading Kalm	Pitch Imp1	Pitch Imp2	Pitch Kalm	Roll Imp1	Roll Imp2	Roll Kalm
Fixed	-37.88	-39.00	-37.95	2.33	2.45	2.38	1.15	0.94	1.15
Fixed, magnetic disturbed	-44.25	-43.82	-44.22	2.58	4.81	2.64	1.06	1.13	1.05
Fixed, run in	-120.61	-58.19	-64.70	0.69	1.10	1.94	0.44	0.39	2.01
Fixed, run in, magnetic disturbance	-44.66	-45.86	-57.94	4.70	5.57	2.46	-0.76	1.85	-3.10
Motion	-42.42	-43.71	-43.94	6.09	7.41	6.45	2.98	0.67	-3.35
Motion, magnetic disturbed	-4.31	2.48	-17.35	0.37	0.27	-1.42	-0.44	0.13	-0.02
RMSE	60.21	42.52	46.85	3.46	4.41	3.32	1.43	1.02	2.13

Figure 16: Tabular mean deviations between the smartphones Samsung Galaxy S8 and Google Nexus 5 for each orientation angle respectively captured by the three virtual sensors Android rotation vector, Improved Orientation Sensor 1 and 2.



Figure 17: Correlation between the orientation angles obtained by the smartphones Samsung Galaxy S8, Google Nexus 5 and Advanced Navigation Spatial v6.1 IMU with respect to the mentioned virtual sensors.

#### 4.4. Parameter stability

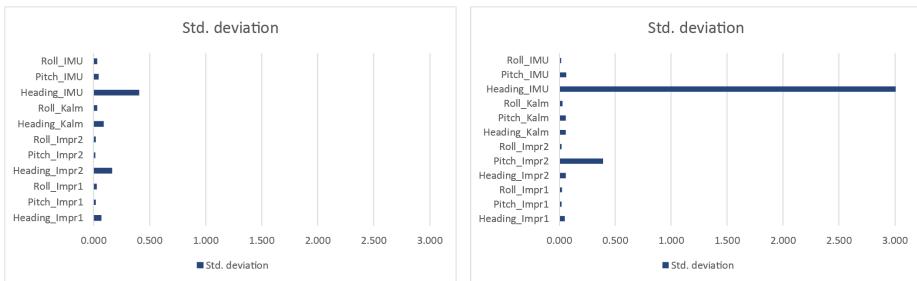
Focussing on both rigid measurements, visualised in figures 12(a) and 12(b), sensor stability can be assessed by comparing standard deviations summarised over full observation period of 2min 30sec (see figure 18). Except heading angle,  
410 both devices show very stable measurements for all virtual sensors with stand-  
ard deviations less than 0.1 degrees. Surprisingly, angles belonging to sensor  
*Improved Orientation Sensor 2* show higher discrepancies although they should  
be less accurate but more stable. Nevertheless, for all virtual sensors the results  
are comparable to others studies like Fritsch et al (2011) [? ], who measured  
415 standard deviations over 20 samples for heading, pitch and roll of  $1.0^\circ$ ,  $0.2^\circ$  and  
 $0.1^\circ$  respectively (using smartphone HTC Hero). Similar results are presented  
by Meek et al. (2013) [? ] for compass (heading) and tilt (pitch) angles referring  
to standard deviations of  $5.1^\circ$  and  $0.75^\circ$ .

Including gyroscopes, heading is vulnerable for large drifts over time [? ]  
420 which is especially noticeable for our reference IMU. Also, the IMU seems to  
be rather susceptible to magnetic disturbances (see figures 12(a) & 12(b) on  
the right hand side). As mentioned in the beginning, sensor fusion can com-  
pensate negative impacts regarding sensor hardware. We have no information  
for filter algorithms used in Advanced Navigation Spatial v6.1, but for smart-  
425 phone orientation determination the applied virtual sensors try to compensate  
this deficiency.

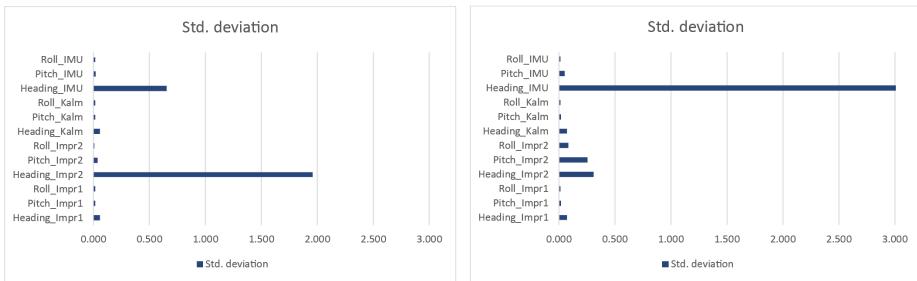
#### 4.5. Parameter sensitivity

When we talk about the registration of 3D objects and 2D image data, it  
becomes obvious that similarities between virtual representations and the cor-  
430 responding captured photos are of pivot importance. Key drivers for this are  
matching extrinsics due to the projection provisions. As shown above, posi-  
tioning and orientation using smartphones may be a serious problem caused by  
incorrect exterior orientation leading to different image contents.

We observe the behaviour of image-to-geometry intersection by the example  
435 of camera-based water gauging using the application *Open Water Levels* on



(a) Standard derivation for Samsung Galaxy S8, 1st run with fixed device (left) and fixed device with magnetic disturbances (right)



(b) Standard derivation for Google Nexus 5, 1st run with fixed device (left) and fixed device with magnetic disturbances (right)

Figure 18: Assessment of sensor stability.

Samsung Galaxy S8. Thus, we compare the results of a manually selected image point of the water line that is transferred into object space using a DEM of the scene captured at lower water (see figure 19). For this, we changed the true angles of heading, pitch and roll (marked with red line) independently from each other in increments of five degrees turning clock- and counter-clockwise.  
440 As shown in figure 19(a) and 19(b), changing the heading and pitch angle is correlated with the number of matching feature points essential for camera estimation. Surprisingly, the result for the determined water level seems to be almost unaffected if only 10% of inliers, compared to the total amount of features being found inside the measurements series, remain due to changing view direction. Thus, the most critic angle (refer to fig. 4.3) can vary in range of  
445  $[-40, 40]$  degrees. A similar picture emerges when assessing the angles for pitch, which can change in range of  $[-25; 25]$ . If these limits are exceeded, the camera only sees the sky or the ground (for pitch) or looks in a completely wrong (compass) direction (regarding heading). Compared to these two angles, the roll angle show a different behaviour. For image matching we chose the rotation-invariant Scale-Invariant Feature Transform (SIFT) descriptor [? ] and thus, changing the roll angle does not have major influence on the outcome as shown in 19(c).

455 Compared to sensor accuracy measurements in 4.3, the results give a comfortable feeling using smartphone sensor fusion for the determination of approximate orientation where pitch and roll show maximum errors up to 7.4 and 5.3 degrees. However, for heading there could be massive problems ahead showing errors in section 4.3 up to 120.6 degrees.

460 **5. Power consumption**

Power consumption is an important metric for mobile field applications, which is at the same time also distinct to the mobile device platform. This metric governs the operation time of an app in an outdoor field setting. In application domains such as field geology, the target operation time is in the



Figure 19: Behaviour of image-to-geometry intersection depending on orientation changes. Tuning the angles heading, pitch and roll independently from each other whereas the others remained unchanged.  $\Delta\text{wl}.$  absolute error for water level in meters, total inliers.. number of image features in feature-based matching displayed in percent depending on most occurrence, red/yellow/green area.. range of accuracy.

<sup>465</sup> range of four to eight hours without device recharging. The original operation time can be extended with external battery packs, although there is a limit of how many battery packs can be taken into the field before their total weight renders the mobile device impractical as a field tool.

We measured the energy consumption of *Open Water Levels* and *GRIT* in realistic settings for case studies in waterline detection and field interpretation. Measuring the power consumption on an app-specific level is not supported by default on mobile devices. Formerly, the power consumption has only been assessed on a hardware component level [? ]. This study utilised the Trepn Profiler<sup>8</sup>, which is currently the only known app on Android devices that facilitate app-specific measurements. Trepn Profiler also allows for the simultaneous logging of technical indicators (e.g. GPU- and central processing unit (CPU) load, memory consumption, CPU temperature), which is used in this study to draw higher-level conclusions on the utilisation of the apps. The presented measurements were obtained on a Google Nexus 5 smartphone. Additional measurements have been obtained with a Samsung S8, which can be located in the supplementary data of this article.  
<sup>475</sup>  
<sup>480</sup>

Our tests involve the quantification of energy consumption contribution from application-specific tasks that relate to CPU and GPU usage. GPU usage is mostly related to image-space operations in 2D, such as the the image presentation and image-related operations (e.g. waterline delineation, geological boundary delineation). In 3D, the GPU is responsible for 3D base data rendering and on-device image-to geometry registration (see section 3). The CPU is responsible for the all non-graphical tasks as well as data loading, photo capturing and mobile sensor management. The dependency of power consumption, CPU load and GPU load is shown in figures 20 and 21.  
<sup>485</sup>  
<sup>490</sup>

In both apps, a clear dependency with CPU load and power consumption is observable. In Open Water Levels, one can observe the reoccurring "double-hump" series within CPU process and power consumption, whereas GRIT dis-

---

<sup>8</sup>Trepn Profiler - <https://developer.qualcomm.com/software/trepn-power-profiler>

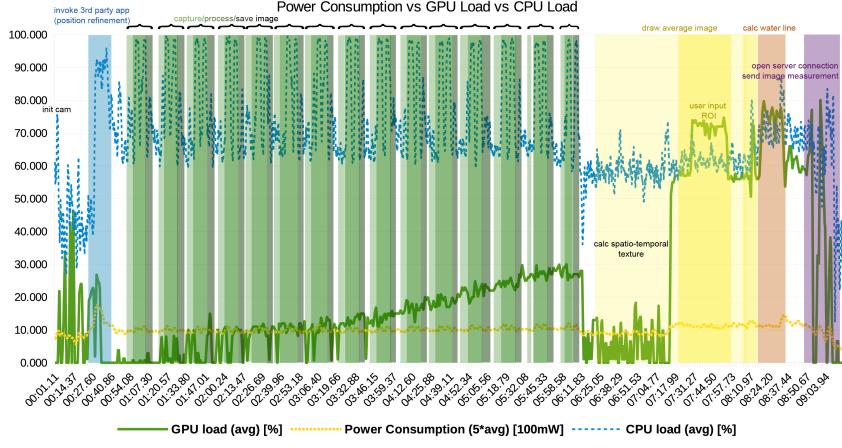
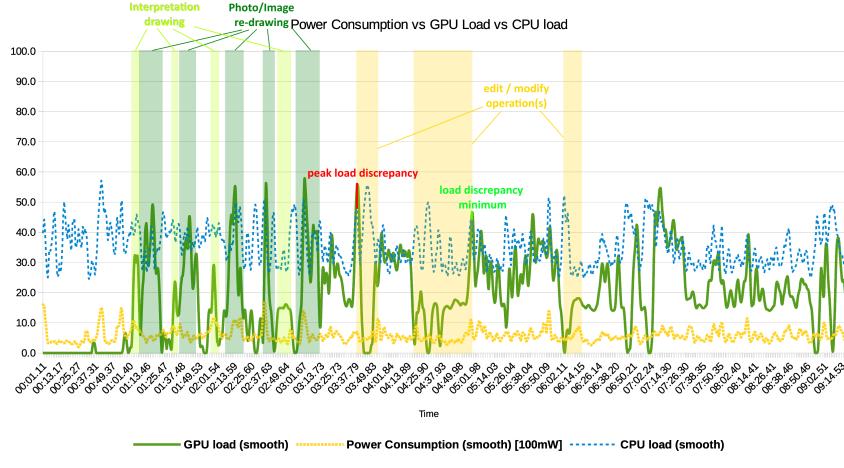


Figure 20: Integrated diagram of power consumption, CPU- & GPU load of *Open Water Levels* in 2D mode.

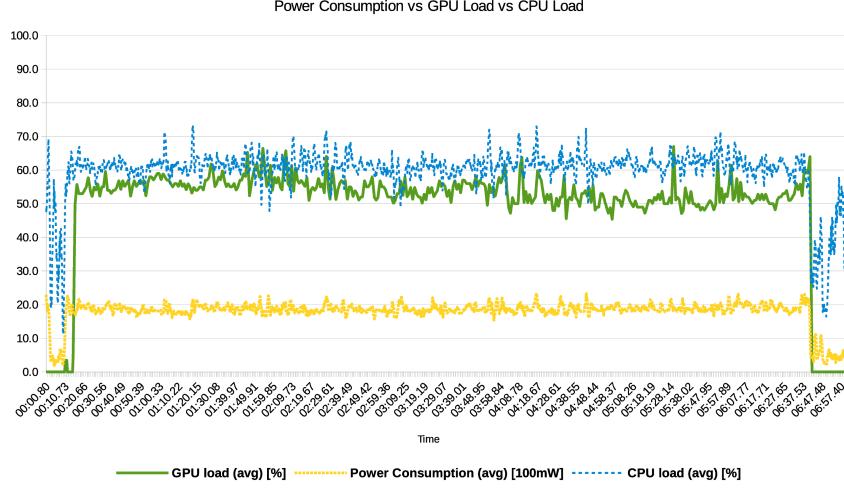
plays a more irregular peak distribution with direct correlations. We can therefore conclude that the mobile processors adapt their clock frequency when less operations are performed, which leads to a reduced power consumption. When comparing CPU-related and GPU-related states, we conclude that while the CPU drives the average power consumption, the GPU (being used for rendering images and annotations within them) drives the peak power consumption.

Figure 20 visualises the relationship of power consumption, CPU, as well as GPU for 2D data processing in *Open Water Levels*. For water line detection, a spatio-temporal texture must be calculated using time lapse images. Thus, the CPU load locally exceeds and falls significantly for each single frame processing (here 15 peaks for 15 images). Unlike CPU behaviour, GPU load is steadily increasing while storing each co-registered image. After image processing, both CPU as well as GPU load are released whereas app modifications via the user interface leads, as expected, once more to higher loads.

When comparing fig. 21(a) and fig. 21(b), the 3D operations result in a drastic energy cost, raising the average power consumption by around 1220.21 mW. In contrast to novice expectation, the CPU load also increases in a 3D data processing setting because the main processors delivers the geometric- and



(a) 2D mode



(b) 3D mode

Figure 21: Integrated diagram of power consumption, CPU- & GPU load of GRIT in 2D- & 3D mode. Particular operations, such as image rendering and interpretation editing, are interpreted within the bands as they result in a distinct CPU-GPU behaviour.

texture data to the GPU. Additionally, the CPU needs to decompress the [image textures](#), resulting in a higher processing load.

The conclusions of this power consumption study for field apps is manifold.  
 515 We obtained benchmark measurements for specific target apps in hydrology (*Open Water Levels*) and geology (GRIT), and explained how to replicate the study on Android devices with other field apps in the future. For *Open Water Levels*, the app can be operated on an average of 1090.41 milliampere per hour (natively measured in milliampere), allowing a theoretical operability of  
 520 2.11 hours on the Google Nexus 5. For GRIT, we have to distinguish between the mode in which it is operated: when conducting 2D operations, the app consumes 568.50 milliwatt per hour, which results in an operation time of 14.56 hours at an average current of 3.6V. When making full use of the 3D capabilities of GRIT all the time, the average power consumption rises to 1788.80  
 525 milliwatt per hour, which results in an operation time of only 4.63 hours at an average current of 3.6V. The applied current for the GRIT measurements is of theoretical nature, applied because the measurements were taken in watt exclusively while the battery capacity of mobile devices is commonly given in milliampere hours (mAh). Furthermore, we highlight these measurements as  
 530 being the *theoretical* operation time because most users have other apps and background services open on their mobile device that simultaneously consume power, further reducing the operation time. Lastly, as stated by Carroll et al. [?], the app-specific consumption (in particular with "visual apps" and the sensor applications) also depends on the screen brightness and the sensor usage. Key  
 535 measures on power consumption, and related metrics of processor temperature and memory usage, are given in table 3 for GRIT and table 4 for *Open Water Levels*.

Table 3: Average measurements of GRIT

metric	2D ops.	3D ops.
power consumption [mW/h]	568.59	1788.80

Table 3: Average measurements of GRIT

power consumption [mA/h 3.6V]	157.94	496.89
memory usage (avg.) [GB]	1.746	1.721
temperature [°C]	49.91	52.05

Table 4: Average measurements of *Open Water Levels*

metric	Google Nexus 5	Samsung S8
power consumption [mA/h]	1090.41	?
memory usage (avg.) [GB]	1.543	?
temperature [°C]	58.55	?

540 In more general terms, the study shows that **application domain practitioners** need to be aware of what data they are dealing with in order to get the maximum operation time and most efficient workload done during the field study. This will have implications for fieldwork planning for expert users and practitioners, as they can modify their study plan to first collect **photos and observations** from  
 545 several viewpoints of their study objective and then use 3D operation features "in burst" for visual checks and data interrogation before moving on to **the next** study locations. Insufficient planning and an overuse of 3D field app features can reduce the effective "digital fieldwork" time using GRIT to 9.26 hours at best when carrying one external battery pack. Also, with this measure we want  
 550 to highlight that the operation time error in the measurements is significant because we need to assume an average current of 3.6V, which may be far off when comparing the measurements to *Open Water Level*.

[SHORTENED AND REARRANGED UNTIL HERE]

## 6. Applications and Requirements

<sup>555</sup> Use cases and application scenarios within the geosciences emerged recently for mobile technology due to the increasing usability of mobile devices for field studies. In the following, two key applications are presented: water level gauging through field observations for small and medium-sized catchments and geological interpretation of sedimentary features in field geology.

<sup>560</sup> *6.1. Derivation of hydrological parameters: Water level gauging*

The past decade is characterized by a continued increase of globally devastating flash floods after heavy rainfalls. Even smallest creeks turned into hazardous streams resulting in floodings and landslides. Conventional gauging stations provide precise information about water levels measured over short time periods. State of the art techniques for administrative water gauging comprise water pressure sensors, floating gauges and conventional tide gauges. They are characterised by long-term stability and outdoor robustness providing accuracies of several millimetres up to one centimetre [? ]. Averaged over defined time intervals, it is advisable to remain cautious regarding these accuracies possibly being too optimistic [? ]. Because of high costs in purchase and maintenance, gauging stations with their complex sensing devices must be sparsely installed. A prime example here is the hydrological network in Saxony, Germany where 184 gauging stations are installed for permanent observation on 154 of 259 rivers rising from small, medium and large catchments<sup>9</sup> [? ]. Thus, around a third is not monitored neither during flood events when the most protection is required. Recently, commercial smartphone applications arose to provide tools for crowd sourcing-based water level estimation (see [? ? ] for details). All of them have one thing in common: the water level is entered manually by engaged citizens who photograph tide gauges close to rivers that presents potential danger to themselves. Beside this, the technique is still limited to open and visible gauges.

<sup>565</sup>

<sup>570</sup>

<sup>575</sup>

<sup>580</sup>

---

<sup>9</sup>see Saxon Flood Centre, Water Levels & Flow Rates <https://www.umwelt.sachsen.de/umwelt/infosysteme/hwims/portal/web/wasserstand-uebersicht>

Improvements can be achieved by the registration of situation-dependent images to 3D point surfaces for automatic water level determination on running waters without requiring reference gauges. For this, the Android application  
585 *Open Water Levels* is developed, which uses the freely available open source camera framework *Open Camera*<sup>10</sup>. *Open Water Levels* allows for free stationing water level detection using short, handheld time-lapse image sequences [? ].

As figured out in section 4, a good approximation of extrinsic parameters is a basic prerequisite for successful 3D annotation whereby precision and stability is strongly correlated with measuring environment. Especially magnetic perturbations affecting user's orientation (see section 4.5) can represent a special problem (e.g. due to metallic railings close to rivers). The issue can be circumvented for stationary perturbation sources by re-calibration of magnetic sensors just before the measurement, as it is often been done for advanced car navigation. Unfortunately, the magnetic strengths attaching the phone may change substantially in short time especially in natural or urban environments.  
590 A typical scenario might look like this: a citizen scientist walks along street, carrying his phone in a baggage close to a bunch of keys. He passes several street lamps, signs, etc. Finally, he arrives at a bridge over a urban river, takes out the phone, looks down to the river and records the time lapse image sequence  
595 a few centimetres above a metallic railing. Meanwhile, several cars passing the same bridge. In this simple use case, the magnetic field around the smartphone changes countless times due to several unpredictable disturbances. Described in section 4.5, image to geometry registration is very error-prone for inaccurate exterior parameters except roll angle. The reason for this lies in rendering  
600 a synthetic image from coloured 3D reference point clouds using a person's location and orientation (see 3.3). Thereby, heading and pitch mainly define the depth direction, incorrect angles provide a false viewing direction resulting in a synthetic image that has little-to-no similarity with the time lapse sequence.  
605 Consequently, the water level detection will fail or give false results caused by  
610

---

<sup>10</sup>see Open Camera, <https://sourceforge.net/projects/opencamera/>, used version 1.3.8

adverse inlier distribution in image matching that impedes a correct positioning (e.g. when images have too little overlap). Described in section 4.2, inbuilt GNSS receivers should be considered as another major source of error for 3D annotation. In urban scenes with several shadow effects due to high-rise buildings, as well as in situations of heavy cloud coverage, errors of several metres in latitude and up to more than 30 metres in altitude are highly possible. However, assistance is provided by external data sources and it is very likely that, in the near future, smartphone GNSS modules are rolled out, solving lateral accuracies of 50 centimetres [? ]. Thus, having internet connection is indispensable but not a problem in urban environments. It is worth mentioning that UMTS/LTE are furthermore needed to enable online water line processing after transmitting a compressed package containing the master image, the derived water line in 2D space and some meta data to describe the prevalent object scene.

For now, an issue is the free-access availability of 3D representations captured close to rivers with focus on shore environment. However, first attempts from Google Street View to cover near shore environments by river cruises are published<sup>11</sup>. In the future, this option is expected to expand to other rivers on a global scale. Furthermore, some research projects deal with autonomous river crossing to acquire hydrological parameters as well as shore land information in short timespans [? ]. Thus, 3D point sets can be acquired very fast (e.g. using light detection and range (lidar) or SfM) covering the same place at different times to deal with multiple representations caused by season-dependent vegetation, snow coverage or changed illumination due to the ambient conditions at specific times of the day. Referring to this, issues in image-to-geometry registration still remain when the visual appearances of both, the photo and the synthetic image, vary widely (e.g. strong back lighting or shadows that let appear everything totally black).

---

<sup>11</sup>USA Today - Google Maps launches 'river view' of Grand Canyon, <https://www.usatoday.com/story/tech/personal/2014/03/13/google-maps-grand-canyon-colorado-river/6339489/>

## 6.2. Field Geology

The goal of geological fieldtrips is to gather insight in the rock record and  
640 the structural- and sedimentary rock architecture of a given location. Rock  
architecture can be studied within subsurface seismic records, but this approach  
suffers from inferior imaging (**seismic**) resolution and physical limitations of  
the surveying technique. Therefore, surface outcrops are used for the study.  
Outcrops can be scanned with modern equipment (e.g. lidar [? ? ], drones [?  
645 ] and SfM [? ]) to generate digital surface representations. The most common  
representations of digital outcrops are coloured point clouds and textured TINs.

The geological aspect is introduced by interpreting the outcrop models. In  
this case, interpretations refer to (i) line marks for separating stratigraphic  
650 layers, (ii) surface-projected polygons to highlight structural- and sedimentary  
facies or specific architectural elements and (iii) minor ticks (e.g. lines, points,  
patterns) to indicate supplementary attributes such as deposition orientation or  
grain geometry. The interpretations was, until recently, performed in a two-step  
655 process: sketches are drawn by hand in a dedicated fieldbook to document the  
geologist's **observations**. After the fieldtrip, the observations are digitised in the  
office by transferring the sketched architecture on the available digital outcrop.  
From there on, further study goals (e.g. geomodelling) are pursued. As recently  
published, this workflow is currently being transformed into an integrated digital  
workflow in the field using mobile devices (see [? ] for further details).

Geological interpretations can be documented on various scales **while most**  
660 **observations** are conducted on medium-range. This results in an average ob-  
servation distance for architectural interpretations of between 100m to 500m  
to document individual depositional elements, and further distances of around  
400m to 1400m to document the overall stratigraphic framework of an outcrop.  
These distances can vary **depending** on the physical accessibility of an outcrop.  
665 Therefore, **and** as a result of perspective observations, the required lateral local-  
isation accuracy is in the range of  $\leq 2.5m$  for the individual element setting and  
 $\leq 8m$  for the wide-angle stratigraphic setting. While achieving the former res-  
olution can still be challenging with mobile sensors (see section 4.1), the latter

resolution is almost guaranteed for GPS localisation. The more important problem  
670 is in the vertical resolution: the vertical position has, especially **on close distance**, a drastic impact on the view perspective. Even more important, a vertical localisation error of  $\geq 1.5m$  may result in positioning the mobile device ”under ground”, making any image-based registration impossible. **Several** improvements, such as DEMs and barometric altitude [? ], have been proposed  
675 to reduce the vertical positioning error (see section 4.1). **Despite the proposed improvements, there** is still room for novel research proposals to provide more accurate vertical positioning or ground-based constraints on the altitude estimation.

One particular challenge in digital field geology is the treatment of environmental changes. Digital outcrops are infrequently collected and the textured models are used for field study all across the year. Therefore, in image registration terms, there is a drastic difference in local illumination, moisture content as well as fog and snow between acquired 3D surface models and the outcrop images collected during field trips. The issue has been previously discussed in terms of illumination differences [? ], but drastic changes in terms of fog and moisture **remain challenging for auto-registration algorithms**. Therefore, it is advisable to collect digital outcrop models for prominent locations in different seasonal conditions to allow for variety in model selection when planning **actual** field trips.  
680  
685

Currently available systems that provide digital outcrop interpretation capabilities on mobile devices in 3D include GRIT [? ] and Outcrop [? ], though earlier prototypes have been demonstrated [? ]. Outcrop, developed by CEREGE at Aix-Marseille Université, is **an** app for Android devices that is able to load and process various forms of numerical outcrops. Its major focus  
690 is the documentation of structural features (e.g. fault areas, fractures and rock deformations) on outcrops using line interpretations. **Furthermore**, it is possible to pin **notes to points within** the model. GRIT, developed as a collaboration between UniResearch AS CIPR, University of Bergen, University of Aberdeen and CEREGE, is a mobile app for Android devices that can handle  
695

<sup>700</sup> large-area digital outcrops of tens of kilometres in surface length in 3D. Its major focus is the documentation of the sedimentary- and stratigraphic architecture (e.g. strata boundaries, depositional object envelopes, facies areas) on outcrops via lines, polygons and brushes. The interpretations are mapped in a 2D–3D interplay between outcrop surface and field photograph.

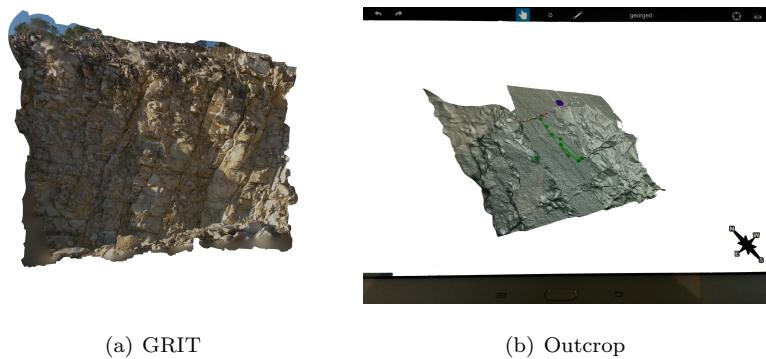


Figure 22: Visual comparison between two 3D mobile apps for DOM interpretation, namely GRIT (a) and Outcrop (b), with a model of the Calvisson quarry (Calvisson, département Gard, région Occitanie, France). Images taken from [? ].

### <sup>705</sup> 6.3. Virtual Field Trips

## 7. Conclusions and Discussion

This article addressed challenges for employing mobile devices and digital tools for outdoor field studies within the geosciences with a special focus on 3D surface utilisation. Due to the research effort in recent years, novel mobile applications such as *Open Water Levels* for surface hydrology and *GRIT* for field geology were introduced to the community to bridge the gap between lab assessment and outdoor field work for documenting observations digitally. This article also showed further application areas that build upon mobile device technology and the interactive annotation of 3D surface data for geoscientific problem solving.

McCaffery et al. proposed the use of mobile devices for field interpretation in geology in 2005 [? ]. The technological specifics of mobile device app development hampered the progress on this goal for years – for geology as well as other branches of the geosciences. Only recent advancements in efficient treatment of 720 3D data [? ], algorithmic proposals for image-to-geometry registration (see [? ? ]) and on-device 3D rendering (as presented in [? ] and in this article for point-based surfaces) specifically designed for mobile devices, **enable professional field use of mobile apps**. The utilisation of crowdsourced VGI and the introduction of mobile devices as low-cost measuring devices for real-world problems 725 [? ] contribute to the acceptance of this mobile device **technology** within the geoscientific community. Computer Vision challenges such as image registration under changing illumination conditions and with reduced image resolution can be viewed as "sufficiently solved" to make photogrammetric- and vision-based algorithms applicable to real-world outdoor settings, while still leaving space 730 for improvement **in** quality and performance. Potentially significant improvement will be achieved in the future when an increasing number of advanced algorithms in numerics, graphics and vision (e.g. NEWUOA [? ], out-of-core rendering [? ], MI [? ]) are ported to mobile platforms (e.g. Android). This allows realising the most state-of-the-art techniques on mobile devices that require 735 the additional precision and performance, instead of being limited by the small function collections currently available.

Here, possible reformulation from MKroehnert, please. This was just a quick-text – enough for CAGEO, not enough for PHOR The measurements presented in this article as well as its related studies suggest that localisation 740 and orientation of mobile device sensors with respect to the application-specific accuracy requirements is a persisting challenge. The sensors employed by low-cost devices have accuracy limitations. Sensor filtering- and fusion techniques are required to even moderately consider the use of such sensor data. Environmental effects such as device-internal heating processes and the system-internal handling of sensor initialisation further complicate the calibration of such sensors without user involvement.

Furthermore, this study gives a representative overview about the energy consumption of mobile apps employing 3D surfaces, computer vision and computer graphics procedures. It was shown that the distinction between 2D- and  
750 3D data used by mobile apps significantly drives the power consumption, and therefore the operation time of the mobile field apps during a study. Means of reducing the power consumption in the future have, next to extended periods of app use by domain experts, beneficial secondary effects: power-reduced main functions of the mobile app allow energy-expensive simultaneous localisation  
755 and mapping (SLAM) techniques to be used for sensor data augmentation.

This paragraph can possibly go if space is needed This article also compared two specific apps, namely *Open Water Levels* and *GRIT*. Both software applications are working on different data structures but, in the end, utilise the same process – namely image-to-geometry registration and user-selected corner point  
760 surface intersection – to generate surface-based annotations and interpretations. A persisting challenge with respect to the relation between power consumption and sensor accuracy is the user feedback: it is currently rarely possible to guarantee the user a correct pose estimation for his base photo, be it individual image or time lapse, upon which annotations and interpretations are done. On  
765 mobile devices, it is important to provide the software user early (visual) feedback about the prospective success and quality, so that potential image capture repetitions can be decided early. This is also in the interest of power conservation on mobile devices by only expending computing power where necessary. Early steps in this direction have been taken [? ], but there is considerable room  
770 for improvement.

Lastly, the treatment of vegetation within scanned- and photographed data during mobile field studies remains a challenge in the context of interactive interpretation. 3D reference data are obtained less frequent than they are used in a given outdoor setting. Vegetation itself is visually dynamic content that  
775 complicates image registration to existing 3D data, which complicates interpretations in common outdoor settings. While current procedures of data processing try to segment and remove vegetation data from scans, it leaves the

mobile device app with less information to work with when registering photos.  
Therefore, proposing means of 3D topographic data processing that homogenizes vegetation in 3D scans and photos without removing the related data will  
<sup>780</sup> have an impact on accurate outdoor photo registration on 3D base data.

### Acknowledgements

First, we would like to thank M.Sc. Richard Boerner from TU Munich, Germany for his assistance with the development of an alternative approach for  
<sup>785</sup> synthetic image generation using perspective 3D-to-2D image projection with respect to object space definition and to solve for point occlusions (see section 3.2). Furthermore, we want graduate the European Social Fund and the Free State of Saxony for their financial support (funding no. 100235479).

### References