*Photogrammetric Record,* xx(xxx): 000–000 (Month 20##)

Image Synthetisation, Sensors Variability and Power Consumption In Mobile Device Field Applications

Melanie Kröhnert (melanie.kroehnert@tu-dresden.de)\*

Hans-Gerd Maas (hans-gerd.maas@tu-dresden.de)

*Institute for Photogrammetry & Remote Sensing, TU Dresden, Helmholtzstr. 10, 01069 Dresden, Germany*

Christian Kehl (chke[@d](mailto:remondino@fbk.eu)tu.dk)

*Danmarks Tekniske Universitet, DTU Compute, Richard Petersens Plads, Building 324, 2800 Kongens Lyngby, Denmark*

\* Corresponding author

Abstract

This is an example of an original article in the layout expected by The Photogrammetric Record. The abstract is a concise summary of the essential elements of the paper, and should not contain detailed results. It should be no longer than 150 words. This template indicates how a paper submitted to journal should be structured and formatted, including figures, tables, mathematics and references.

Keywords: photogrammetry, example, word or short phrase, layout, maximum 6 words

I. 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 are digital field instrument. This is illustrated in fig. fig:intro:interpretation, sketching the envisaged use of tablets and smartphones for geological and hydrological purposes.

|  |  |
| --- | --- |
|  |  |

Fig. 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 ~\cite{} for examples). The availability three-dimensional base data and their increasingly-easy acquisition (e.g. via \gls{UAV} applying \gls{SfM} \cite{Wu2013} and multi-view geometry \cite{Goesele2007}, satellite \glspl{DEM}) 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 \gls{VGI} contribute to the geoscience data inventory, being acquired by citizen scientists.

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 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 tablets and smartphones, the array of sensors (e.g. \gls(GNSS) and orientation units with varying degrees of accuracy) available in the field and the high computational qualities compared to the device size as well as it is possible to outsource heavyweight processing tasks to external working stations using largely available mobile networks. 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 original ways.

This article addresses the challenges of mobile sensor variability, their usage in image-to-geometry registration of point cloud base data, and the related energy consumption in comparison to a \gls{DSM} base data mobile application. The technical research is approached via two use cases within the domains of surface hydrology and (petroleum) geology. The content covered in the article is a significant extension of earlier published research \cite{Kroehnert2017b}, 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 (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.

II. 3D base data represenations

Various representation forms for 3D terrain data are available. While early digital systems used gridded \glspl{DEM} for their simplicity and compact storage \cite{Trinks2005,McCaffrey2005}, \glspl{DSM} and \glspl{TIN} are dominating most terrain-based systems for application-specific analysis \cite{Buckley2008a,Caumon2013} \change{at the moment}. A useful example can be seen in \cite{Schwalbe2017b} 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 \glspl{TIN}, organised as \glspl{PLC}, from polygon soup surfaces (fig. \ref{fig:representations:meshDistinction}). While the latter is often employed in early stages of mesh-based software systems due to its simplicity and ease of implementation, valid \glspl{TIN} 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.

|  |  |
| --- | --- |
|  |  |

Fig. 1. Illustrative distinction between valid \glspl{TIN} (a, consisting of one exclusive, smooth, closed surface) and polygonal soups (b). Non-textured model parts are coloured with respect to their actual segment number. Images taken from \cite{Kehl2017\_PhDThesis}.



Fig. 1. Example of a \gls{DOM} as textured triangular surface.

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 \cite{Buckley2008a} and Caumon et. al \cite{Caumon2013}, the surface is supplemented with photographic information via texture projection. The models are referred to as \glspl{DOM} (see fig. \ref{fig:representations:DOM} as reference depiction).

In contrast, other geoscience domains, such as hydrology and free surface flow management, used georeferenced laser scanner point clouds and coloured point data streams provided by terrestrial photogrammetry for small- or \gls{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. gls{SfM}). The \glspl{PSS} support tasks like coastal monitoring \cite{Letortu2017, Medjkane2018}, soil erosion and rain-induced landslide observation, and even monitoring river topography \cite{Watanabe2016} and flood protection management \cite{Leskens2015}. Nevertheless, new approaches for low-cost and on-the-fly river monitoring \cite{Kroehnert2017a} arise due to globally increasing flash flood events after heavy rainfalls \cite{Mueller2011} that are further addressed in section \ref{sec:water\_level\_gauging\_intro}.

Since \gls{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 \gls{TLS}. Regarding 3D annotation, nearest neighbour analysis provides an opportunity whereby surface triangulation can be avoided.

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 \cite{Garcia2015}). The sparse vertex distribution in point clouds can cause problems in the data analysis, which is why \glspl{DEM} have seen a revival in the mobile computing domain. \Glspl{DEM} 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 \glspl{DEM} are advantageous.

III 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 to model presentation as well as for the generation of a synthetic reference image for image-to-geometry registration. Furthermore, it can be used to synthesize an image from available 3D data for interpretation and annotation in 2D.

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 \gls{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. Modern programmable shaders allow in-time vertex decompression (see \cite{Ponchio2016}) as well as texture decompression (see section \ref{sec:technology:graphics}). 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 equals identity matrix and can be neglected.

First, applying a six-parameter transformation transfers three-dimensional object points from world reference frame into a 3D camera system using

(1)

where is a orthonormal rotation matrix and the translation vector to camera's projection center. For simplicity, the usage of the planar Cartesian \gls{UTM} system with pointing to the east and pointing to the north with respect to the prevalent zone number. For -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 and image coordinates involving their depth components.

(1)

For camera's imaging plane, the constant  [mm] defines the physical distance between camera's sensor and its projection centre, which equals focal length  [px] in the image coordinate system. The normalization of the projected points to homogeneous coordinates is key in the further processing.

(1)

(1)

For a final transformation of 2D sensor coordinates into image pixels, the image coordinate system must be shifted to the origin to left upper corner and scale the coordinates from global units in meters per pixel using . Thus, image coordinates for an ideal camera can be described with

(1)

3.3 Calculation of 3D bounding box of interest and image plane

In the mobile rendering scenario, a region of interest regarding 3D point projection has to be defined in order to cull the render content of the virtual camera to the user's field of view (figure \ref{fig:4\_3\_bounding\_box}). 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 \ref{sec:technology:sensors}), 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 and describe the domain of projection centre’s uncertainties parallel to image plane. For errors in depth, the correction is introduced to shift the projection centre along camera axis.

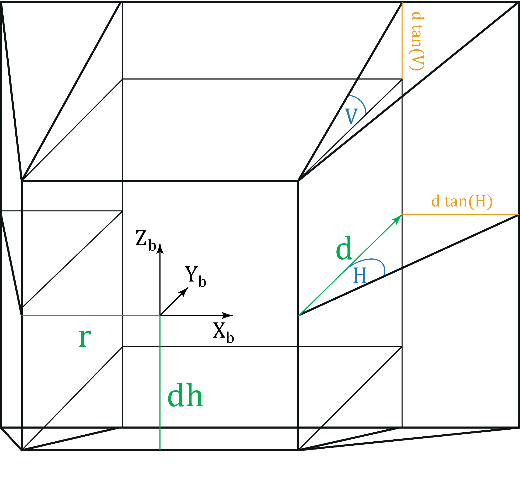


Fig. 1. Bounding box definition.

The box is widened by the horizontal and vertical opening angles with a fixed depth . In order to generate reference data for image-to-geometry 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, the depth is set to . Additional tiling of the 3D base data is advisable for a rapid geometry-in-frustum containment checks.

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 -plane (\ref{fig:4\_3\_bounding\_box}) with eq. 1 for the bounding box’ background plane, described by its upper left and lower right corner.

,

(1)

The height equals the height component in the world reference frame . Because of pyramid frustum, we subsequently eliminate points outside the near- and far clipping plane.

4.3.2 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 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 case of e.g. glass fronts with lacking information (in \gls{TLS} due to deflected lidar or \gls{SfM} when having homogeneous surfaces) or small archs (see figure \ref{fig:4\_3\_dist\_images}). 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 \cite{Sattler2011}.

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 (see eq. \ref{eq:final\_ps}) with , 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 (see fig. {fig:4\_3\_dist\_images} for results).

|  |  |
| --- | --- |
|  |  |

FIG. 1. Obscured but visible 3D points close to arches and windows (a), edge preserving result after filtering (b).

4.3.3 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 registration purposes) there will still be gaps between projected points (see figure \ref{fig:4\_3\_dist\_images}, right). In order to fill these gaps, it is recommended to apply a simple nearest neighbour approach using binary search \cite{Bentley1975} 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 \ref{sec:water\_level\_gauging\_intro}, a before-after comparison of the gap filling is shown in figure \ref{fig:4\_3\_fill\_images\_before\_after}.

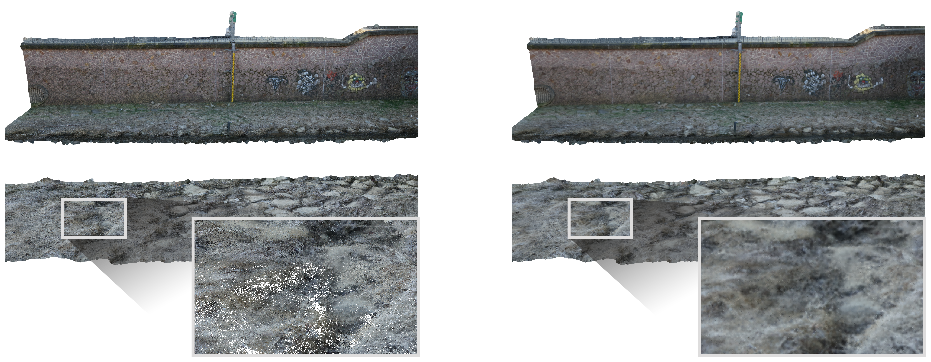


FIG. 1. Fill image gaps using nearest neighbour binary search in 3D domain.

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:

(1)

(1)

(1)

with – rotation matrix; – translation vector

– point of the object; – (projected) point of the image plane

– image plane coordinates; – 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 in the image, such as image-based interpretations, can also be mapped on the surface.

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.

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 \cite{Kehl2016\_ISPRS}). The intersection between the ray and a triangle within the mesh results in the correlated 3D coordinate 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 \cite{Sibbing2013,Sattler2011,Rodriguez2012,Garcia2015}) employs smart rendering techniques that virtually expand the point into an area feature (e.g. blob, disk or sphere), which is subsequently rendered into a depth map. Afterwards, the 3D coordinate of a 2D feature can be inferred directly from the depth map. 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 accuracy errors in the registration process, whereas high-resolution depth maps cost considerable performance in the image generation. This last point is particularly important when employing depth map algorithms on mobile devices.

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 () from the 2D-3D point-based equation system. Each time the camera of a mobile device is started, it has to be re-initialised. Furthermore, the more sensors are activated, the device heats up rapidly affecting the in-built components. Obviously, the stability of inbuilt smartphone cameras is highly critical and thus their calibration (cite Kroehnert2017b). Having enough, well-distributed 2D-3D point pairs, the intrinsic parameters and lens distortion can be refined, too. Non-linear optimisation systems (e.g. Levenberg-Marquardt) are applied to estimate the desired parameter set \cite{Torr2000}. The whole process can easily be executed on mobile devices \cite{Kehl2016\_ISPRS}. One of the prevalent practical challenges when employing feature-based image-to-geometry registration is to achieve a reliable feature correlation, which is often achieved by introducing application-specific constraints (e.g. horizon alignment \cite{Garcia2017), straight-edge enforcement or object outlines).

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 \cite{Gauglitz2014,Sweeney2015} over field geology \cite{Kehl2016\_ISPRS,Kehl2017\_VGC} to surface hydrology \cite{Kroehnert2017a,Boerner2016}. These mobile apps utilize the open-source library *OpenCV4Android*[[1]](#footnote-1), which is also employed in this work[[2]](#footnote-2). 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) \cite{Kehl2017\_PHOR}. A completely alternative technique to feature-based methods is \gls{MI} \cite{Viola1997,Corsini2013}. \Gls{MI} performs a pixel-wise comparison between the photo and the 2D rendering of the 3D scene and aims at minimizing the image discrepancies (i.e. ). 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, \gls{MI} faces challenges in the optimization process: the optimization of a 7 degree-of-freedom equation system (, for 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\cite{Powell2006}) used by Corsini et al. \cite{Corsini2013}. None of the advanced solvers is available in modern- and mobile-device programming languages, thus the use of \gls{MI} on mobile platforms is currently prohibited.

IV 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.

4.1 Localisation

Compared to the years 2008 and 2009, sales volume for navigation systems declined sharply and constantly by approximately 70 percent compared to 2017 in Germany[[3]](#footnote-3).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 smartphones are equipped with \gls{aGPS} receivers 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 \cite{Westhead2011,Masiero2016}, or actually 3D reconstruction \cite{Micheletti2015,Muratov2016,Ishihara2017}.

Based on these facts, many research groups recently discussed the potential of smartphone localisation strategies whereby we want to focus on outdoor use cases based on \glspl{GNSS}. Blum et al. (2013) \cite{Blum2013} 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 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) \cite{Fritsch2011} who determined a overall accuracy for Android smartphone HTC Hero of 15-25 m valid in 95 % of cases which was also estimated by Zhu et al. (2013) \cite{Zhu2013} and Zandbergen et al. (2011) \cite{Zandbergen2011}. Exemplary for open spaces, Meek et al. (2013) \cite{Meek2013} observe an average \gls{GPS} accuracy of 6.8m using a Google Nexus S smartphone. However, height estimation seems to be more critical where \cite{Liu2014} name error margins for altitude determination using smartphone's inbuilt \gls{aGPS} which seem to be 2.5 times more than the horizontal component and recommend the alternative usage of 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

Pre-knowledge about an image's position is a 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 steps of 2.5 m up to a deviation of 25 m for northing, easting and height component and compare the detected water line with ground truth data from an administrative water gauge (see figure \ref{fig:sensor\_sensi:easting}, \ref{fig:sensor\_sensi:northing}) having an enclosing \gls{DEM}.

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

Compared to observed accuracies of smartphone inbuilt \gls{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 \gls{GRIT} enables repositioning based on locally stored \gls{DEM} data for user-guided repositioning. To correct the even more erroneous height measurements, one option is the use of external \glspl{DEM} included in \gls{GRIT} or invoking third party models e.g. via Google Elevation API[[4]](#footnote-4) , as it is implemented in *Open Water Levels*.

FIG. 1. Number of total inliers in [%] registering real and synthetic images (see 4.1) manipulating users’ position with regards to height, northing and easting, respectively. Dashed lines refer to missing values because in case failed matching.

*5.1.3 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 \glspl{IMU}. 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 other’s, 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 \cite{Pacha2015}, he presents two alternative virtual sensors additionally to Android's Kalman filtered *Rotation Vector*[[5]](#footnote-5), 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 \gls{INS} that is commonly used for car and \gls{UAV} navigation. For this, we compare measurements taken at three different times for the devices Google Nexus 5, Samsung Galaxy S8 and the \gls{IMU} Spatial from the Australian company Advanced Navigation v6.1 that serves as ground truth (for sensor specifications refer to tables \ref{table:sensor:specs}[[6]](#footnote-6) and \ref{table:sensor:imu}[[7]](#footnote-7)).

Table I. Orientation sensor specifications for Google Nexus 5 and Samsung Galaxy S8.

|  |  |  |
| --- | --- | --- |
|  | Google Nexus 5 | Samsung Galaxy 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 I. 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 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 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 \gls{IMU} are mounted on an inflexible non-metallic wooden stick at a distance of 1.0 m (to avoid mutual magnetic interferences) with aligned (native) coordinate systems (see figure \ref{fig:technology:sensor:construction}). Only for pitch angle the opposite direction of rotation must be kept in mind.

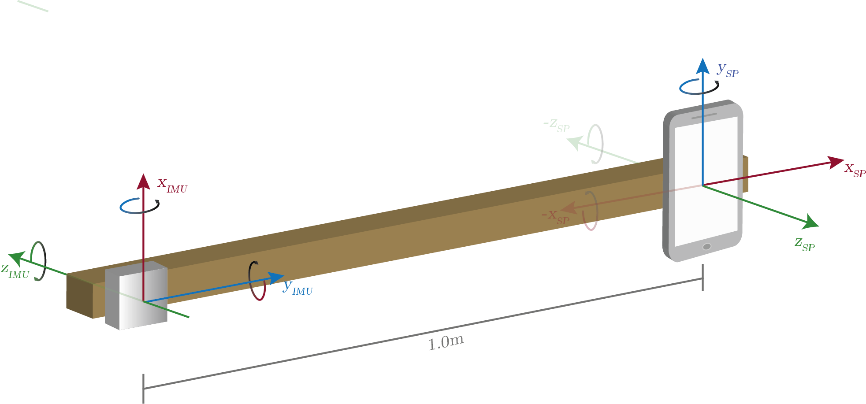


FIG. 1. Measurement setup to observe smartphone sensors accuracies and precisions. Heading around +/- z-axis pointing to/away from the sky, pitch around +/- x-axis pointing out of smartphone's display to the left, roll around y-axis pointing to true north when smartphone is lying on a flat desk.

In the following, orientation tracking for Samsung Galaxy S8 in comparison with the IMU is documented for the first observation period about 60 seconds with a sampling rate of 20 Hz and matched by UTC time every 0.1 s. The recorded values refer to the orientation angles (magnetic) heading, pitch and roll and are given in degrees. The related figures collectively use the same legend, which is given in figure LEGEND.



FIG. 1. Legend.

FIG. 1. Observation of orientation angles measured from Samsung Galaxy S8 and the reference IMU. Both devices are in rest free from (top) and exposed to (bottom) magnetic disturbances.

FIG. 1. Observation of orientation angles measured from Samsung Galaxy S8 and the reference IMU. Both devices are in rest after a short warm up (~ 30 s) free from (top) and exposed to (bottom) magnetic disturbances.

FIG. 1. Observation of orientation angles measured from Samsung Galaxy S8 and the reference IMU. Both devices are in constant movement free from (top) and exposed to (bottom) magnetic disturbances.

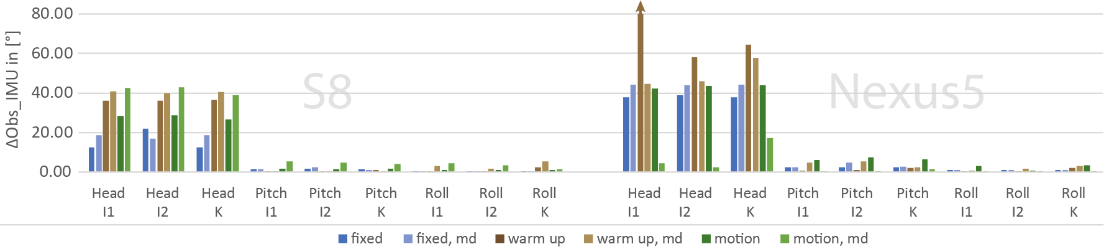


FIG. 1. Mean deviations between Samsung Galaxy S8 and Google Nexus 5 for heading, pitch and roll, respectively, use of *Improved Orientation Sensor 1, 2* (I1, I2) *and Kalman* (K) filtering.

Ein Bild, das Text, Anzeigetafel, Wand enthält.

Mit sehr hoher Zuverlässigkeit generierte Beschreibung

FIG. 1. Correlation between observed orientation and reference IMU, respectively, use of *Improved Orientation Sensor 1, 2* (I1, I2) and *Kalman* (K) filtering.

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 fig:fixed, fig:warmup, fig:motion. Especially *Improved Orientation Sensor 1* and the factory Kalman-filtered *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 \gls{IMU} even if the correlations in figure \ref{fig:sensor\_sensi:imu\_correlation\_s8\_nex5} shows little similarity. This correlation issue is caused by slight synchronisation errors visible e.g. in the graphs of figures \ref{fig:sensor\_obs:s8:1st:runin:4} and \ref{fig:sensor\_obs:s8:1st:motion:5} which may be caused by averaging sensor values to establish comparability between both smartphones and \gls{IMU} respectively by UTC time.

Considering \glspl{RMSE} in figure \ref{fig:sensor\_sensi:imu\_mean\_s8\_nex5\_tab}, good agreement with related studies can be resolved. Blum et al. (2013) \cite{Blum2013} determined orientation errors up to 30 degrees for heading with significant drifts accelerating over while walking in the streets with an iPhone 4 for several minutes. Similary, Kok et al. (2017) \cite{Kok2017} 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 \gls{IMU}.

5.1.4 Parameter stability

Focussing on both rigid measurements, visualised in figures \ref{fig:sensor\_obs:s8:1st:fixed:1} and \ref{fig:sensor\_obs:s8:1st:fixed:2}, sensor stability can be assessed by comparing standard deviations summarised over full observation period of 2min 30sec (see figure \ref{fig:sensor\_sensi:imu\_sensor\_stabi\_s8\_nex5}). Except heading angle, both devices show very stable measurements for all virtual sensors with standard deviations less than 0.1 degrees. Surprisingly, angles belonging to sensor \textit{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) \cite{Fritsch2011}, who measured standard deviations over 20 samples for heading, pitch and roll of 1.0, 0.2 and 0.1 degrees respectively (using smartphone HTC Hero). Similar results are presented by Meek et al. (2013) \cite{Meek2013} for compass (heading) and tilt (pitch) angles referring to standard deviations of 5.1 ° and 0.75 °.

Including gyroscopes, heading is vulnerable for large drifts over time \cite{Kok2017} which is especially noticeable for our reference \gls{IMU}. Also, the \gls{IMU} seems to be rather susceptible to magnetic disturbances (see figures \ref{fig:sensor\_obs:s8:1st:fixed:1} \& \ref{fig:sensor\_obs:s8:1st:fixed:2} on the right hand side). As mentioned in the beginning, sensor fusion can compensate negative impacts regarding sensor hardware. We have no information for filter algorithms used in Advanced Navigation Spatial v6.1, but for smartphone orientation determination the applied virtual sensors try to compensate this deficiency.

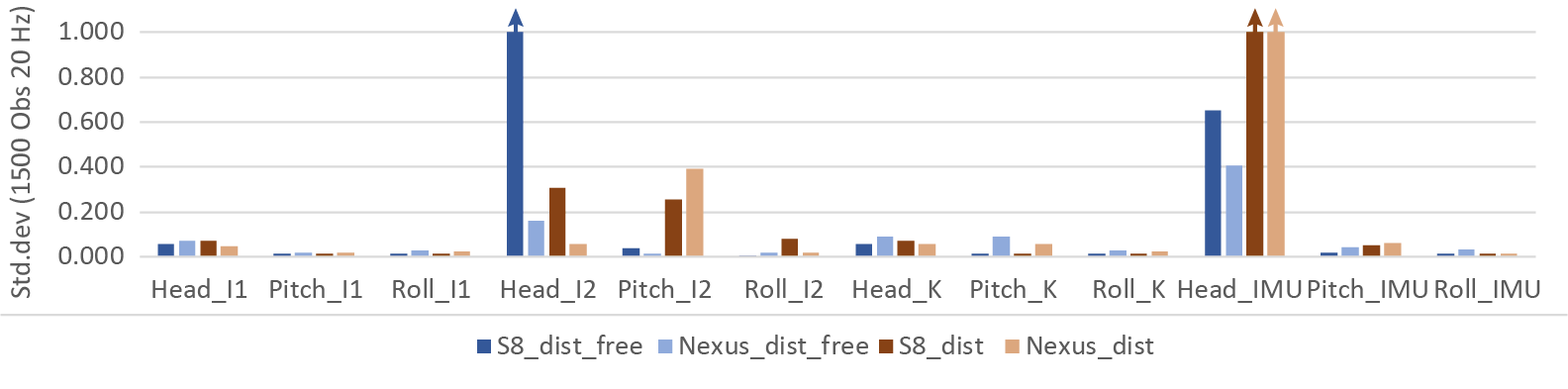


FIG. 1. Research of the orientation angles heading, pitch and roll when the devices Samsung Galaxy S8 and Google Nexus 5 are in rest, having an environment, free and full of magnetic disturbances (refer to sensor observation 1 and 2). Apply *Improved Orientation Sensor 1* and *2* (I1, I2) and *Kalman* (K) to filter the values.

*5.1.5 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 corresponding captured photos are of pivot importance. Key drivers for this are matching extrinsics due to the projection provisions. As shown above, positioning 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 of camera-based water gauging using the application *Open Water Levels* on 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 \gls{DEM} of the scene captured at lower water (see figure \ref{fig:sensor\_sensi}). 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.

As shown in figure \ref{fig:sensor\_sensi:heading} and \ref{fig:sensor\_sensi:pitch}, 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. \ref{sec:technology:sensors:orientation}) can vary in range of [-40, 40] degrees. A similar picture emerges when assessing the angles for pitch, which can change in range of [-25, 25] degrees. 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 \gls{SIFT} descriptor \cite{Lowe2004} and thus, changing the roll angle does not have major influence on the outcome as shown in \ref{fig:sensor\_sensi:roll}.

Compared to sensor accuracy measurements in \ref{sec:technology:sensors:orientation}, 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 \ref{sec:technology:sensors:orientation} up to 120.6 °.

FIG. 1. Number of total inliers in [%] registering real and synthetic images (see 4.1) manipulating the orientation with regards to heading, pitch and roll, respectively. Dashed lines refer to missing values because in case failed matching.

5.3 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 for specific studies. In application domains such as field geology, the target operation time is in the 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 \gls{*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 \cite{Carroll2010}. This study utilised the Trepn Profiler[[8]](#footnote-8), 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. \gls{GPU}- and \gls{CPU} load, memory consumption, \gls{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 (4-core ARM \gls{CPU}, Qualcomm Adreno \gls{GPU}). Additional measurements have been obtained with a Samsung S8 (8-core ARM \gls{CPU}, ARM Mali \gls{GPU}), which can be located in the supplementary data of this article.

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 image presentation and image-related operations (e.g. waterline and geological boundary delineation). In 3D, the GPU is responsible for 3D base data rendering and on-device image-to geometry registration (see section ??). 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 ?? and ??.

In both apps, a clear dependency with \gls{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 \gls{GRIT} displays 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 \gls{CPU}-related and \gls{GPU}-related states, we conclude that while the \gls{CPU} drives the average power consumption, the GPU (being used for rendering images and annotations within them) drives the peak power consumption.

Figure \ref{fig:power:Power\_CPU\_GPU\_OWL:2D} visualises the relationship of power consumption, \gls{CPU}, as well as \gls{GPU} for 2D data processing in \textit{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 \gls{CPU} behaviour, \gls{GPU} load is steadily increasing while storing each co-registered image. After image processing, both \gls{CPU} as well as \gls{GPU} load are released whereas app modifications via the user interface leads, as expected, once more to higher loads.

When comparing fig. \ref{fig:power:Power\_CPU\_GPU:2D} and fig. \ref{fig:power:Power\_CPU\_GPU:3D}, 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 \gls{CPU} load also increases in a 3D data processing setting because the main processors deliver the geometric- and texture data to the \gls{GPU}. Additionally, for the Google Nexus 5 smartphone, the \gls{CPU} needs to decompress the image textures, resulting in a higher processing load.

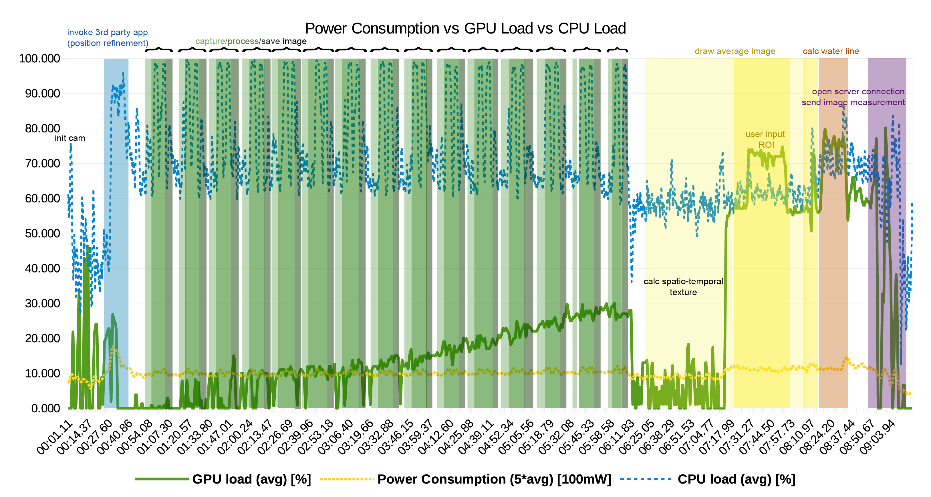
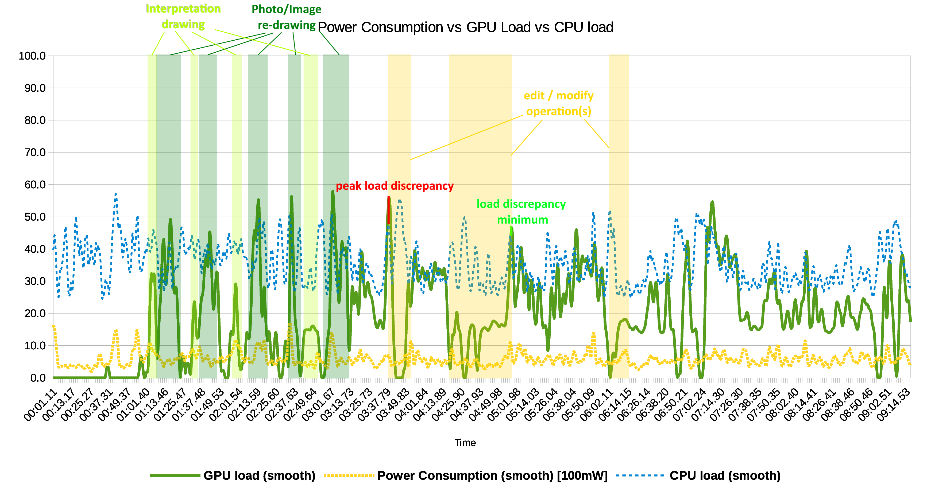
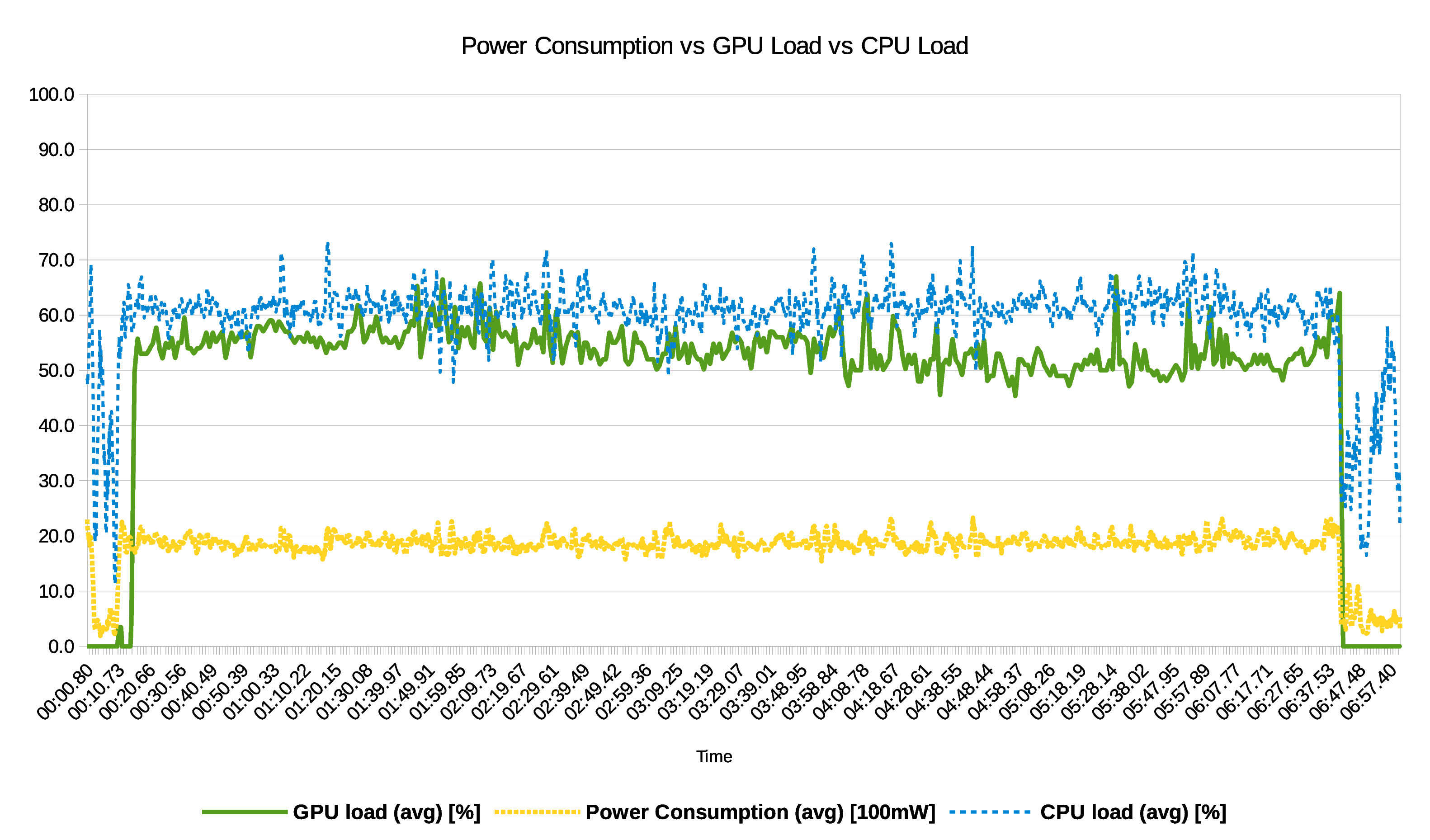


FIG 1. Integrated diagram of power consumption, CPU- \& GPU load of *Open Water Levels* in 2D mode.

The conclusions of this power consumption study for field apps is manifold. We obtained benchmark measurements for specific target apps in hydrology (*Open Water Levels)* and geology (\gls{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 of 1090.41 milliampere per hour (natively measured in milliampere), allowing a theoretical operability of 2.11 hours on the Google Nexus 5. For \gls{GRIT}, we have to distinguish between the mode in which it is operated: when conducting 2D operations, the app consumes 568.50 mW per hour, which results in an operation time of 14.56 h at an average current of 3.6 V. When making full use of the 3D capabilities of \gls{GRIT} all the time, the average power consumption rises to 1788.80 mW per hour, which results in an operation time of only 4.63 h at an average current of 3.6 V. 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 being the \textit{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. \cite{Carroll2010}, the app-specific consumption (in particular with ''visual apps`` and the sensor applications) also depends on the screen brightness and the sensor usage. Key measures on power consumption, and related metrics of processor temperature and memory usage, are given in table \ref{table:power:GRIT} for \gls{GRIT} and table \ref{table:power:OWL} for *Open Water Levels*.



(a) 2D mode.



(b) 3D mode.

FIG 1. 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 \gls{CPU}--\gls{GPU} behaviour.

Table I. Average measurements of GRIT.

|  |  |  |
| --- | --- | --- |
|  | 2D ops. | 3D ops. |
| power consumption [] | 568.59 | 1788.80 |
| power consumption [] | 157.94 | 498.89 |
| memory usage (avg.) [GB] | 1.75 | 1.72 |
| temperature [°C] | 49.91 | 52.05 |

Table I. Average measurements of *Open Water Levels*.

|  |  |  |
| --- | --- | --- |
|  | Google Nexus 5 | Samsung Galaxy S8 |
| power consumption [] | 1090.41 | n.a. |
| memory usage (avg.) [GB] | 1.54 | n.a. |
| temperature [°C] | 58.55 | n.a |

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 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 subsequent study locations. Insufficient planning and an overuse of 3D field app features can reduce the effective “digital fieldwork” time using \gls{GRIT} to 9.26 h at best when carrying one external battery pack. Also, with this measure we want 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*.

VI Applications and Requirements

Due to the increasing usability of mobile devices for field study annotations, several use cases concerning geosciences has become apparent. In the following, two key applications are subsequently presented: water level gauging through field observations for small and medium-sized catchments, geological interpretation of sedimentary features in field geology, and the use of mobile devices in virtual field trips.

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 \cite{Siedschlag2015}. Averaged over defined time intervals, it is advisable to remain cautious regarding these accuracies possibly being too optimistic \cite{Horner2018}. 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[[9]](#footnote-9) \cite{Buettner2015}. 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 \cite{CrowdWaterApp2017a,Kisters2014} 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.

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 *Open Water Levels* is developed, which uses the freely available open source camera framework Open Camera[[10]](#footnote-10). *Open Water Levels* allows for free stationing water level detection using short, handheld time-lapse image sequences \cite{Kroehnert2017a}.

As figured out in section \ref{sec:technology:sensors}, 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 \ref{sec:technology:sensors:sensitivity}) 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 being done for advanced car navigation. Unfortunately, the magnetic strengths attaching the phone may change substantially in short time especially in natural or urban environments. 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 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 \ref{sec:technology:sensors:sensitivity}, image to geometry registration is very error-prone for inaccurate exterior parameters except roll angle. The reason for this lies in rendering a synthetic image from coloured 3D reference point clouds using a person's location and orientation (see \ref{sec:algorithms:I2G}). 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. Consequently, the water level detection will fail or give false results caused by adverse inlier distribution in image matching that impedes a correct positioning (e.g. when images have too little overlap). Described in section \ref{sec:technology:sensors:location\_sensitivity}, inbuilt \gls{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 \gls{GNSS} modules are rolled out, solving lateral accuracies of 50 centimetres \cite{Moore2017}. 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 availability of free available 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[[11]](#footnote-11). 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 \cite{Sardemann2018}. Thus, 3D point sets can be acquired very fast (e.g. using \gls{LiDAR} or \gls{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).

6.2 Field geology

The goal of geological fieldtrips is to gather insight in the rock record and 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 resolutions and physical limitations of the surveying technique. Therefore, surface outcrops are used for the study. Outcrops can be scanned with modern equipment (e.g. \gls{LiDAR} \cite{Buckley2008a,Buckley2010}, \cite{Dewez2015} and \gls{SfM} \cite{Chandler2016}, inter alia, with UAVs \cite{Dewez2015}) to generate digital surface representations. The most common representations of digital outcrops are coloured point clouds and textured \glspl{TIN}.

The geological aspect is introduced by interpreting the outcrop models. In this case, interpretations refer to (i) line marks for separating stratigraphic 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 process: sketches are drawn by hand in a dedicated fieldbook to document the geologist's observation of the architecture. 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 \cite{Kehl2018\_AGU} for further details).

Geological interpretations can be documented on various scales, but from observations of the author most interpretations are conducted on medium-range. This results in an average observation distance for architectural interpretations of between 100 m to 500 m to document individual depositional elements, and further distances of around 400 m to 1400 m to document the overall stratigraphic framework of an outcrop. These distances can vary to some degree depending on the physical accessibility of an outcrop. Therefore, as a result of perspective observations, the required lateral localisation accuracy is in the range of  2.5 m for the individual element setting and ≤ 8 m for the wide-angle stratigraphic setting. While achieving the former resolution can still be challenging with mobile sensors (see section \ref{sec:technology:sensors:localization}), the latter resolution is almost guaranteed for \gls{GPS} localisation. The more important problem is in the vertical resolution: the vertical position has, especially in close-distance observations, a drastic impact on the view perspective. Even more important, a vertical localisation error of 1.5 m may result in positioning the mobile device ''under ground'', making any image-based registration impossible. It is this vertical accuracy that is crucial for mobile device interpretation systems to work. Several improvements, such as \glspl{DEM} and barometric altitude \cite{Kehl2017\_VGC}, have been proposed to reduce the vertical positioning error (see section \ref{sec:technology:sensors:localization}). There is still room for novel research proposals to provide more accurate vertical positioning or ground-based constraints on the altitude estimation.

One of the dominant challenges for digital field geology is the free availability of 3D surface models. Currently, research groups in the domain (e.g. from the University of Manchester \cite{Hodgetts2013}, Durham University \cite{McCaffrey2005}, University of Aberdeen \cite{Howell2014}, University of Bergen and UniResearch CIPR \cite{Dreyer1993}) are building their own digital outcrop databases. Due to the strong industry involvement, these and other databases (see SAFARI \cite{Dreyer1993} and FAKTS \cite{Colombera2012a}) are excluded from public access. Recent developments aim at providing digital outcrops in an open-access manner \cite{Cawood2018} to resolve the issue. Furthermore, due to the vertical positioning problem above, easy access to high- and medium resolution \glspl{DEM} is important. As demonstrated by recent measurements, the usage of \glspl{DEM} has a significant influence on the projection accuracy of image-based interpretation on mobile device towards 3D surface models \cite{Kehl2017\_VGC}.

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 \cite{Kehl2017\_PHOR}, but drastic changes in terms of fog and moisture are still problematic to treat in an automatic, computational manner. 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 field trips.

Currently available systems that provide digital outcrop interpretation capabilities on mobile devices in 3D include \gls{GRIT} \cite{Kehl2016\_VGCabstract} and Outcrop \cite{Viseur2014\_VGCabstract}, though earlier prototypes have been demonstrated \cite{Hama2013}. Outcrop, developed by \gls{CEREGE} at Aix-Marseille Université, is a mobile device app for Android devices that is able to load and process various forms of numerical outcrops. Its major focus 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 extended note annotations to the model. \gls{GRIT}, developed as a collaboration between UniResearch AS CIPR, University of Bergen, University of Aberdeen and \gls{CEREGE}, is a mobile app for Android devices that can handle 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.

|  |  |
| --- | --- |
| (a) GRIT | (b) Outcrop |

Fig. 1. Visual comparison between two 3D mobile apps for \gls{DOM} interpretation, namely \gls{GRIT} (a) and Outcrop (b), with a model of the Calvisson quarry (Calvisson, département Gard, région Occitanie, France). Images taken from \cite{Kehl2017\_PhDThesis}.

VII Conclusions and Discussion

This article assessed the possibility of interactive interpretation and annotation of 3D surfaces (pre-acquired by \gls{TLS}, drones or \gls{SfM}) on mobile devices in multiple geoscientific domains. 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 data annotation and interpretation. 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 \cite{McCaffrey2005}. 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 3D data \cite{Kroehnert2017b}, algorithmic proposals for image-to-geometry registration (see \cite{Gauglitz2014,Kehl2017\_VGC}) and on-device 3D rendering (as presented in \cite{Agus2017} and in this article for point-based surfaces) specifically designed for mobile devices, make the actual use for mobile apps in the field possible. The utilisation of crowdsourced \gls{VGI} and the introduction of mobile devices as low-cost measuring devices for real-world problems \cite{Eltner2017} contribute to the acceptance of this mobile device technological development 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 for improvement and 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 \cite{Powell2006}, out-of-core rendering \cite{Borgeat2005}, \gls{MI} \cite{Viola1997}) are ported to mobile platforms (e.g. Android). This allows realising the most state-of-the-art techniques on mobile devices that require the additional precision and performance, instead of being limited by the small function collections currently available.

The measurements presented in this article as well as its related studies suggest that localisation 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 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 \gls{SLAM} techniques to be used for sensor data augmentation.

This article also compared two specific apps, namely *Open Water Levels* and \\gls{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 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 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 \cite{Kehl2017\_VGC}, but there is considerable room 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 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 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 synthetic image generation using perspective 3D-to-2D image projection with respect to object space definition and to solve for point occlusions (see section \ref{sec:algorithms:pbr}). Furthermore, we want to graduate the European Social Fund and the Free State of Saxony for their financial support (funding no. 100235479).

REFERENCES

Ackermann, F., 1984. Digital image correlation: performance and potential application in photogrammetry. *Photogrammetric Record*, 11(64): 429–439.

Granshaw, S. I., 2016. Photogrammetric terminology: third edition. *Photogrammetric Record*, 31(154): 210-251.

Gruen, A., 1996a. Digital photogrammetric stations revisited. *International Archives of Photogrammetry and Remote Sensing,* 31(B2): 127–134.

Gruen, A., 1996b. Least squares matching: a fundamental measurement algorithm. Chapter 8 in *Close Range Photogrammetry and Machine Vision* (Ed. K. B. Atkinson). Whittles, Caithness, Scotland. 371 pages: 217–255.

Hosseininaveh Ahmadabadabian, A., Robson, S., Boehm, J. and Shortis, M., 2014. Stereo-imaging network design for precise and dense 3D reconstruction. *Photogrammetric Record*, 29(147): 317-336.

ISPRS, 2014. www.isprs.org/publications (Accessed 23rd November 2014).

Pierrot-Deseilligny, M. and Paparoditis, N., 2006. A multiresolution and optimization-based image matching approach: an application from SPOT5-HRS stereo imagery. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(1/W41): 5 pages (on CD-ROM).

Toldo, R., Fantini, F., Giona, L., Fantoni, S. and Fusiello, A., 2013. Accurate multiview stereo reconstruction with fast visibility integration and tight disparity bounding*. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(5/W1): 243-249.

Vosselman, G. and Maas, H.-G. (Eds.), 2010. *Airborne and Terrestrial Laser Scanning*. Whittles, Caithness, Scotland.. 318 pages.

Wiley, 2016. http://onlinelibrary.wiley.com/journal/10.1111/(ISSN)1477-9730 (Accessed 23rd November 2014).

Zhang, L., 2005. *Automatic Digital Surface Model (DSM) Generation from Linear Array Images*. Doctoral thesis. Mitteilungen 88. Dissertation No. 16078. Institute of Geodesy and Photogrammetry, ETH Zurich, Switzerland. 218 pages. <http://e-collection.library.ethz.ch/view/eth:28126>

*Résumé*

L’histoire de l’appariement d’images remonte à plus de cinquante ans, lorsque les premières …

Zusammenfassung

Die digitale Bildzuordnung hat seit den ersten analogen Ansätzen für die automatisierte ...

Resumen

La correspondencia de imágenes tiene una historia de más de 50 años, desde los primeros …

摘要

影像匹配技术在模拟摄影测量中首次应用开始，已经有50年的发展 …

1. OpenCV4Android 2.4.10 - https://opencv.org/platforms/android [↑](#footnote-ref-1)
2. OpenCV4Android extensions - https://github.com/CKehl/opencv4Android\_extension [↑](#footnote-ref-2)
3. see Statista (accessed 2018-04-25) *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/ [↑](#footnote-ref-3)
4. see Android Developer Guide (accessed 2018-04-25) *Google Elevation API* https://developers.google.com/maps/documentation/elevation/intro?hl=de [↑](#footnote-ref-4)
5. See Android developers guidance (accessed 2018-04-25) *Sensor event values* https://developer.android.com/reference/android/hardware/SensorEvent.html\#values [↑](#footnote-ref-5)
6. See specifications for Samsung Galaxy S8 and Google Nexus 5 (both accessed 2018-04-25) https://technology.ihs.com/api/binary/592077, https://de.ifixit.com/Teardown/Nexus+5+Teardown/19016 [↑](#footnote-ref-6)
7. See specifications for Advanced Navigation Spatial v6.1 (accessed 2018-04-25) http://www.advancednavigation.com.au/product/spatial\#specifications [↑](#footnote-ref-7)
8. see Trepn Profiler (accessed 2018-04-29) https://developer.qualcomm.com/software/trepn-power-profiler [↑](#footnote-ref-8)
9. see Saxon Flood Centre, Water Levels & Flow Rates (accessed 2018-04-29) https://www.umwelt.sachsen.de/umwelt/infosysteme/hwims/portal/web/wasserstand-uebersicht [↑](#footnote-ref-9)
10. see Open Camera, version 1.3.8 (accessed 2018-04-29) https://sourceforge.net/projects/opencamera/ [↑](#footnote-ref-10)
11. see USA Today: Google Maps launches 'river view' of Grand Canyon (accessed 2018-04-29)   
    https://www.usatoday.com/story/tech/personal/2014/03/13/  
    google-maps-grand-canyon-colorado-river/6339489/ [↑](#footnote-ref-11)