A Computational Model for the Integration of Linked Data in Mobile Augmented Reality Applications

Stefan Zander
Research Group Multimedia Information Systems
University of Vienna
stefan.zander@univie.ac.at

Chris Chiu
Research Group Multimedia Information Systems
University of Vienna
chris.chiu@univie.ac.at

Gerhard Sageder
Research Group Multimedia Information Systems
University of Vienna
gerhard.sageder@univie.ac.at

ABSTRACT

Linked Data, Augmented Reality (AR), and technical advancements in mobile information technology lead to an increasing desire to exploit Linked Data for the integration and visualization in mobile AR applications. However, current approaches are either bound to existing client-server-based infrastructures or use closed data sources and proprietary data formats. Moreover, a number of related approaches are built upon content-based recognition algorithms that are both memory and processing-intensive, require a permanent connection to a host, and thus are inappropriate for a direct deployment onto mobile devices. In this work, we present a computational model that builds on a sensor-based tracking approach and maps proactively replicated Linked Data sets to a virtual representation of the user's vicinity computed by a mathematical model. We demonstrate the applicability of our approach through a proof-of-concept AR application that retrieves and aggregates mountain-specific data from a set of different sources and displays such data in a live-view interface. In consequence, our approach is resource-efficient, does not require a permanent network connection, is independent from existing server-based infrastructures, and allows to process Linked Data directly on a mobile device.

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems—*Artificial, augmented, and virtual realities*; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Virtual reality*

General Terms

Human Factors, Algorithms, Experimentation

Keywords

linked data, mobile information systems, augmented reality, semantic web, visualization

I-SEMANTICS 2012, 8th Int. Conf. on Semantic Systems, Sept. 5-7, 2012, Graz, Austria

1. INTRODUCTION

The concept of Augmented Reality (AR) is well-known since the early 90's; it complements physical objects with computer-generated content in real time and allows users to interact with the real world through computer-generated interfaces [2, 7, 15]. This definition comprises the main aspects an AR system exhibits: (i) it combines real and virtual objects in a real environment; (ii) it runs interactively and in real time; (iii) it registers (aligns) real and virtual objects with each other [3]. The omnipresence of mobile devices together with continuous advancements in mobile information technology make the implementation of mobile AR systems attractive for a broad number of users and use cases – as indicated by the increasing number of conferences and research projects related to that topic [31].

One domain that is well-suited for the exploitation of data to be integrated in AR systems is that of *Linked Data* [5, 13] where the potential of such a synthesis was recently acknowledged (cf. [22]). By fetching and combining Linked Data from different data sets, such data can be complemented with location-based information retrieved from geographic services such as *GeoNames*[1] or *Earthtools*[2] that provide the necessary information to determine the latitude and longitude coordinates of non-information resources [14] as well as their altitude and elevation. Existing approaches host such data in isolated data silos or employ proprietary data formats. Moreover, vision-based or hybrid AR systems, which are discussed in Section 2, exhibit computationally complex matching and classification algorithms and require extensive sets of reference images wherefore their deployment in mobile AR scenarios is limited.

In this work, we present a computational model that builds on a mathematical model and combines a sensor-based tracking approach with a mobile RDF processing and management framework for the provision and complementation of Linked Data with related geographical information to enable an utilization by mobile AR applications. It proactively replicates Linked Data to a mobile device, aggregates and consolidates such data (e.g., filter identical resources as indicated by the `owl:sameAs` property or identify and merge identical resources using declaratively described heuristics), and can be deployed in a wide variety of application scenarios. In consequence, our approach is resource-efficient compared to vision-based tracking approaches, it does not require a permanent network connection as data can be replicated proactively, it operates independently from specific

[1] `http://www.geonames.org/`
[2] `http://www.earthtools.org/`

data sources and host infrastructures, and it is capable to process Linked Data directly on a mobile device.

In the remainder of this work, we introduce different AR tracking methods and discuss relevant works w.r.t. employed tracking methods and data utilization methodologies in Section 2. Details of the proposed computational and mathematical model together with main conceptual constituents are presented on a formal basis in Section 3. We demonstrate the practical applicability of our approach through a prototypical proof-of-concept application that retrieves and federates mountain-specific data from various Linked Data sources and displays them real-time in an AR interface in Section 4. Runtime-specific aspects of the proof-of-concept prototype together with data quality aspects are evaluated in Section 5, and in Section 6 we summarize our findings and discuss possible future directions of Linked Data AR application development.

2. BACKGROUND AND RELATED WORK

In general, AR tracking methods can be categorized into *marker-based* and *marker-less*. Marker-less tracking is further divided into *vision-based tracking*, *sensor-based tracking*, and a *hybrid form* of both methods (cf. [31]). As marker-based tracking requires a controlled environment and is less suited for outdoor mobile AR systems, we exclusively discuss works that employ marker-less tracking techniques.

2.1 Vision-based Tracking

Generally, vision-based tracking systems are classified according to the position of the optical sensor relative to the environment and the objects it tracks [10]. As we specifically focus on tracking methods for mobile AR systems, we only consider works that are built on the assumption of a stationary scene and a flexible optical sensor.

Vision-based systems generally provide accurate results but require manual initialization and a permanent connection to a host infrastructure for computing object recognition and feature classification algorithms [11, 21]. Such an architectural approach is particularly suited for mobile AR systems as handheld devices usually lack processing power and memory capacity, or—in case of entry-level devices—the deployment of powerful floating point units (FPU), which requires a transformation into fixed point integer-based data structures [11, 18]. Moreover, vision-based tracking depends on the quantity and quality of reference images and might be influenced by lighting conditions [23]. Another problem is *scalability* as the computational effort increases with the amount of objects to be recognized and tracked as well as with the quantity of reference images [9].

Vision-based tracking methods are applied for both indoor and outdoor localization and are complemented with different technologies for classification and recognition such as Fourier transformation-based matching techniques [23], machine-learning techniques to improve tracking approximations (e.g. [8]), scale invariant feature transformation and classification [29], annotation features [27], or adaptive landmark-based feature extraction and clustering to increase efficiency and robustness [24]. Other works (e.g. [18, 19]) utilize vision-based tracking methods for indoor localization where the virtual representation of physical objects is complemented with computer-generated 3D models that are projected over a recognized object.

2.2 Sensor-based Tracking

Sensor-based tracking techniques make use of data streams acquired from built-in sensors such as accelerometer, gyroscope, GPS sensor, or digital compass to determine the physical location and the orientation of a device. The accurateness of sensor-based tracking systems depends on the quality and precision of acquired sensor data as well as on the user's current location. It requires an augmentation of objects with geographic information to determine their position relative to the orientation and field-of-view of a camera. A computational model calculates the projections of the augmented content displayed in an AR interface in relation to the real world coordinates of the pertaining physical objects and keeps track of camera movements.

Existing works can be classified into AR browsers and domain-specific AR applications; *Mixare*[3], *Wikitude*[4], and the *Layar Reality Browser*[5] are recently published AR browsers that employ sensor-based tracking techniques. While Wikitude is built upon the Augmented Reality Markup Language (ARML)[6] for mapping and visualizing geo-referenced objects such as POIs and their meta data, Mixare and Layar use hidden and proprietary data structures encapsulated in developer APIs. All works rely on a server-based infrastructure and enable the integration of user-generated content through specific infrastructure-dependent Web services.

A sensor-based AR application for the visualization of user-generated content in an AR interface is introduced by [4] that analyzes the EXIF metadata annotations of multimedia objects and places pictures over the corresponding POIs. *Sekai Camera*[7] and *LibreGeoSocial*[8] introduce social-sharing features and allow users to post and discover user-generated multimedia content in a physical space. While *Sekai Camera* exhibits a closed architecture, *LibreGeoSocial* and [4] operate independently from specific content providers and can be adapted to specific application needs.

Two works that make use of Linked Data are the *SmartReality*[9] project and the AR browser presented in [25] in which Linked Data is combined with domain-specific content for the exploration and visualization of cultural heritage content. A backend server handles aggregation, consolidation, and complementation tasks and fetches Linked Data resources from DBpedia, LinkedGeoData, and Geonames. The SmartReality framework combines AR technology with Web services and semantic technologies for the exploration of information about Things of Interest (TOIs) located in the user's immediate vicinity. Reasoning, aggregation, classification, and filtering tasks are handled by a server-side platform that preprocesses acquired data for client-side visualization.

2.3 Hybrid Tracking

Hybrid tracking approaches combine vision- and sensor-based tracking techniques; while sensor-based tracking is fast, computationally less expensive, and does not require a host infrastructure, vision-based tracking methods provide more accurate recognition results at the cost of complex host

[3] http://www.mixare.org/
[4] http://www.wikitude.com/
[5] http://www.layar.com/
[6] http://www.openarml.org/wikitude4.html
[7] http://www.tonchidot.com/en/
[8] http://www.libregeosocial.org/
[9] http://www.smartreality.at/

infrastructures. In hybrid approaches, sensor-based tracking methods usually serve as initialization means for vision-based recognition and matching methods and allow for a more precise and robust estimation of the camera pose in uncontrolled environments [17].

Hybrid tracking techniques are mainly deployed in outdoor localization system for urban environments and combine vision-based tracking methods with GPS data to minimize the necessity of re-initialization inputs and reduce GPS signal inaccuracies, which are caused by atmospheric signal delays and multi-path signals [21]. Other works (e.g. [9]) employ sophisticated patch-retrieval and image clustering algorithms for real-time camera pose recovery or employ multi-layered tracking methods (e.g. [17]) where sensor-based tracking is complemented with edge detection and feature point tracking techniques. Some hybrid approaches use statistical models or *Kalman filters* (e.g. [1, 16, 20]) for the fusion of sensor and vision-based mensurations.

2.4 Summary

In summary, most works exhibit client/server-based infrastructures for hosting location-based information and are developed for specific domains or are centered around specific data sources (e.g. Wikipedia). They rely on proprietary data formats and specific APIs for the integration of user-generated content. Only two projects make use of Linked Data repositories and discuss aspects related to the identification, integration, consolidation, and complementation of Linked Data with domain-specific content. Nevertheless, standardization efforts towards a common markup format for AR-specific content are currently ongoing.

3. APPROACH

Our approach builds on sensor-based tracking technology and maps Linked Data to a virtual representation of the user's vicinity computed by a mathematical model that is described in detail in Section 3.1. The computational model underlying our approach is capable to process RDF data directly on a mobile device, performs consolidation and reasoning heuristics, and displays relevant information of both information and non-information resources in AR interfaces at run-time. We presuppose the existence of a local RDF processing and management infrastructure which can be realized with a mobile RDF framework such as *Androjena*[10]. In contrast to other mobile RDF frameworks, Androjena revealed to be most mature and is able to process comparatively large data sets containing around 100,000 triples on current state-of-the-art devices in reasonable time [30].

The computational model presented as approach in this work consists of six phases and is depicted in Figure 1. It relies on the assumption that both position and orientation sensors are deployed on a mobile device. Current mobile operating systems provide well-defined APIs and interfaces for requesting data from locally deployed sensors and use such data for further processing as indicated by the *Acquisition of Sensor Data* phase. Position data serve as initialization parameters for Linked Data replication and the mathematical model, whereas orientation data, which are continuously acquired, are used as runtime parameters for the real-time calculation of the corresponding screen coordinates updates.

In the *Linked Data Replication* phase, acquired position data are analyzed and used for initiating requests to Linked Data sources; depending on offered interfaces, such data can be integrated in *GeoSPARQL queries*[11] or incorporated in dedicated API requests (e.g., the current location coordinates plus a particular radius). Retrieved data sets are stored in a local triple store on the mobile device as *named graphs* and are prepared for further processing steps.

The *Consolidation* phase comprises the identification and removal of identical resources as well as the aggregation of different descriptions pertaining to a resource. In order to discover links between resources retrieved from different LOD sources and to perform data-level consolidations on aggregated and federated data, our approach builds on the inclusion of a link discovery framework such as *Silk* [6] that allows for a declarative specification of linking type semantics, conditions, and (aggregation) operations for establishing RDF links between federated resources. Moreover, Silk considers sub graph topologies of RDF resources and allows to define reverse/inverse properties on the basis of extended graph-based edge navigation and filter-based node selection [6]. These aspects allow for a declarative definition of complex consolidation heuristics that serve as means for computing similarities between resources hosted in different domains and identify `owl:sameAs` relationships. For instance, the similarity of descriptions can be computed on different levels and assigned different weights to infer whether they describe the same resource. We demonstrate the consolidation of two such descriptions as part of the proof-of-concept by means of a simple example in Section 4.

In order to display Linked Data in AR interfaces, the descriptions of replicated resources need to be complemented with geographical data that allow for determining the real-world position (3D coordinates) of an object's representation; for instance, the latitude and longitude property values of a resource are complemented with altitude information acquired from secondary data sources in case a resource's description does not exhibit this information originally. In the *Complementation* phase, all resources hosted in the local tripe store are therefore analyzed and become augmented in case such information is missing. Missing information can either be acquired from the description of another resource identified as identical or by invoking requests to geo-location services such as GeoNames or Earthtools. However, depending on the amount of replicated resources hosted in the local triple store, the augmentation may cause multiple concurrent asynchronous requests.

The consolidated and augmented resource descriptions are then processed by the *Mathematical Model* to calculate the corresponding screen coordinates on the basis of the acquired sensor data. The mathematical model is a central aspect of our approach and described in detail in Section 3.1. It allows for creating a virtual representation of resources located in the surrounding environment on the basis of their real-world coordinates. For performance reasons and to differentiate run-time-specific aspects from static resource descriptions, we host the calculated screen coordinates of the pertaining resources in a separate internal data structure. However, to make calculated data available to external applications, we suggest to represent them as RDF and exhibit interfaces for their utilization by external applications such as mobile AR browsers (cf. Section 2) or to map them to AR

[10] `http://code.google.com/p/androjena/`

[11] `http://geosparql.org/`

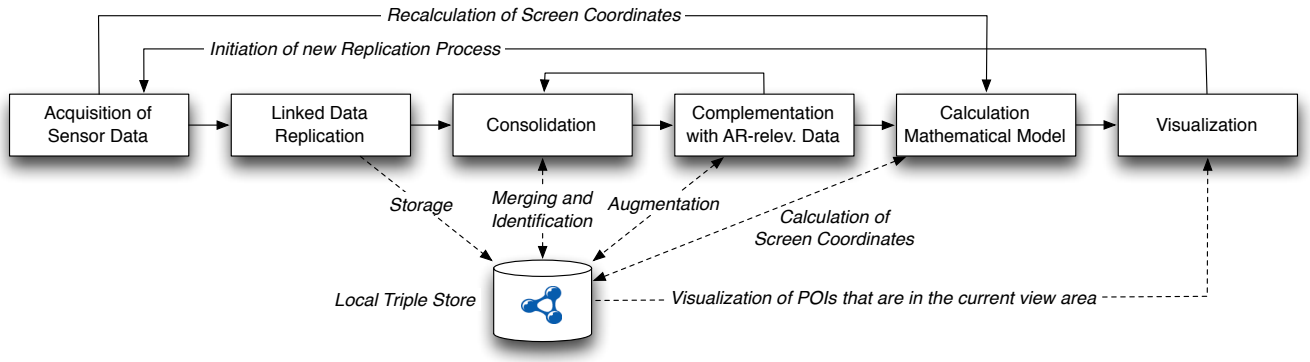

Figure 1: Conceptual workflow of the computational model

markup languages such as ARML 2.0[12]. We consider that future work as it requires the existence of efficient RDF retrieval, persistence, and query implementations for mobile platforms.

After transforming real-world 3D coordinates of replicated resources to 2D screen coordinates, they are displayed in an AR user interface where the mathematical model calculates the corresponding screen coordinates relative to the user's current position and the orientation of a device in realtime.

3.1 Mathematical Model

Both the camera's position as well as the positions of replicated resources[13] are provided in geographic coordinates λ, φ, ψ (cf. Table 1). The camera's geographic position is obtained by using the values provided by the mobile device's GPS sensor. If the mobile device's sensors do not provide an accurate sea level altitude value, we make use of geo information services to obtain elevation data for given longitude/latitude values.

In order to convert geographic coordinates into camera view space, the distance of each POI relative to the camera's position is calculated using the *Vincenty's inverse formula* [26]. It approximates the earth's shape as an ellipsoid with a shorter polar axis than equatorial diameter (oblate spheroid) as opposed to other calculation methods such as the *haversine formula*, which assumes the earth's shape to be perfectly spherical (cf. [12]) causing less accurate results near the polar caps. The formula is iterative and loops until the required degree of accuracy is reached. Thus, the higher computational costs of the Vincenty formula compared to less accurate methods such as the haversine formula can be partially mitigated by setting conservative accuracy requirements. As a result, we obtain the final bearing angle β from the camera's position to the POI's position and the distance z. We also calculate the vertical inclination angle θ between the camera eye point and and the specific POI (Figure 2) by considering the altitude of the camera and the POI using basic trigonometry.

We introduce definitions for each part of our mathematical model: *Definition 1* introduces bearing, distance, and inclination as functions. *Definition 2* introduces camera angle-of-view borders, and *Definition 3* shows the combined mathematical model our approach uses.

[12] `http://www.opengeospatial.org/projects/groups/arml2.0swg`
[13] We refer to resources as points of interest (POIs)

Table 1: Mathematical symbols

Symbol	Description
$\lambda_{poi}, \lambda_{cam}$	Longitude of POI/camera
$\varphi_{poi}, \varphi_{cam}$	Latitude of POI/camera
ψ_{poi}, ψ_{cam}	Altitude of POI/camera
r	Range radius within which POIs are queried
α	Recorded azimuth of device
ρ	Recorded pitch of device
β	Bearing from device to POI
z	Ellipsoidal distance between device and POI
θ	Inclination angle from camera to POI
H_{aov}	Horizontal angle of view
V_{aov}	Vertical angle of view
x_{deg}, y_{deg}	Horizontal/vertical angle mapped to camera's horizontal/vertical angle of view
x_{screen}, y_{screen}	2D screen space position in pixels
w, h	Device screen dimensions
P	Set of POI geo-locations
C	Set of camera parameters
$\mathbf{S}$	Vector field of angle-based screen positions

Definition 1. Let P be a set of POI geographic locations $p = (\lambda_{poi}, \varphi_{poi}, \psi_{poi})$, and C be a set of camera parameters $c = (\alpha, \rho, H_{aov}, V_{aov}, \lambda_{cam}, \varphi_{cam}, \psi_{cam})$. We define a function

$$v_{bearing} \colon P \times C \to [0, 2\pi)$$

that calculates the bearing angle β from a camera point $c \in C$ to a geographic location $p \in P$ using the Vincenty Inverse Formula. Furthermore, we define a function

$$v_{distance} \colon P \times C \to [0, r]$$

that calculates the distance from c to p, also using Vincenty's Inverse Formula. Lastly, we define an inclination function $f_{incl} \colon P \times C \to [-\pi, +\pi]$ as

$$f_{incl}(p, c) = \arctan\left(\frac{\psi_{poi} - \psi_{cam}}{v_{distance}(p, c)}\right) \text{ with } p \in P, c \in C$$

that calculates the inclination angle θ from a camera point c to a POI p.

The camera's azimuth α and pitch ρ can be obtained from the mobile device's orientation sensors, and we also retrieve the focal length/field of view of the camera lens through the device API. We can calculate the horizontal and vertical angles of view from the camera's field of view. In combination

with the camera's location $(\lambda_{cam}, \varphi_{cam}, \psi_{cam})$ obtained by the GPS sensor we then have a complete definition of the device's position and rotation in space.

Definition 2. Let $c = (\alpha, \rho, H_{aov}, V_{aov}, \lambda_{cam}, \varphi_{cam}, \psi_{cam}) \in C$ be a tuple of camera parameters. We define two horizontal *angle-of-view border functions*

$$l_{border}(c) = (\alpha - H_{aov}/2) \bmod 2\pi$$
$$r_{border}(c) = (\alpha + H_{aov}/2) \bmod 2\pi$$

yielding the bearing angles of the left and right border of the camera's horizontal field of view respectively. Analogous we define two vertical angle-of-view border functions

$$t_{border}(c) = (\rho + V_{aov}/2)$$
$$b_{border}(c) = (\rho - V_{aov}/2)$$

for defining the angles of the top and bottom border of the camera's vertical field of view.

The bearing β can now be mapped to the horizontal angle of view, and the inclination angle θ to the vertical angle of view by considering both the device's azimuth α, pitch ρ and horizontal/vertical angles of view. We only consider POIs inside the camera's view frustum, and calculate the horizontal and vertical angles (x_{deg}, y_{deg}) inside the camera's horizontal and vertical angles of view (Figure 2).

Definition 3. Let $\mathbf{S}$ be a vector field of POI positions within the camera's field of view $s = (x_{deg}, y_{deg}, z)$ with $x_{deg} \in [0, H_{aov}]$, $y_{deg} \in [0, V_{aov}]$ (see Figure 2) and the camera-POI-distance $z \in [0, r]$. We define a function $f_{aov} : P \times C \to \mathbf{S}$ that transforms POI locations to angle-of-view coordinates as

$$f_{aov}(p, c) = \begin{pmatrix} t_x(v_{bearing}(p, c), c) \\ t_y(f_{incl}(p, c), c) \\ v_{distance}(p, c) \end{pmatrix}$$

where t_x is a horizontal transformation function $t_x : [0, 2\pi) \times C \to [0, 1]$ defined as

$$t_x(\beta, c) = \begin{cases} \beta - l_{border}(c) & \text{if } \beta \geq l_{border}(c), \\ \beta - (2\pi - l_{border}(c)) & \text{otherwise} \end{cases}$$

and t_y is a vertical transformation function $t_y : [-\pi, \pi] \times C \to [0, 1]$ defined as

$$t_y(\theta, c) = t_{border}(c) - \theta$$

t_x and t_y result in a coordinate $x_{deg} \in [0, H_{aov}]$ and $y_{deg} \in [0, V_{aov}]$ defining the position within the camera's field of view. To get final device screen coordinates, we use the screen dimensions w and h to calculate

$$x_{screen} = \frac{x_{deg}}{H_{aov}} \, w \tag{1}$$
$$y_{screen} = \frac{y_{deg}}{V_{aov}} \, h \tag{2}$$

We obtain the coordinates $(x_{screen}, y_{screen}, z)$ for each POI which can be used as center point for drawing each POI information overlay on screen. The distance value z can be used for visualizing how far each POI is situated.

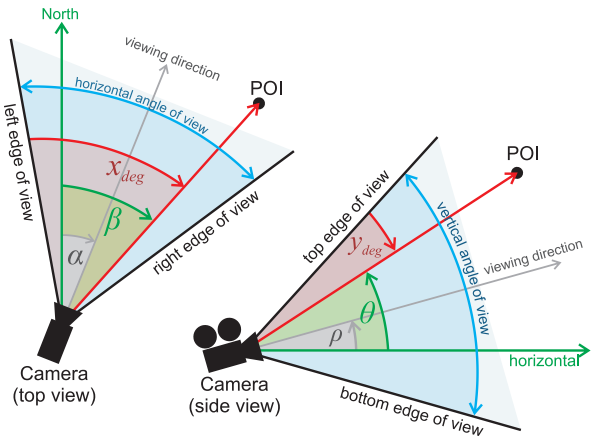

Figure 2: Mapping the POI's bearing β and inclination θ to (x_{deg}, y_{deg})

4. PROOF OF CONCEPT

We demonstrate the practical applicability of our approach in form of a prototype running on the Android mobile operating system that proactively replicates mountain-specific Linked Data to the mobile device and visualizes such data real-time in an AR interface depending on the user's current position and the orientation of the device.

We assume the following scenario to illustrate the steps described in Figure 1: the GPS sensor acquires device location information and provides them to the Linked Data replication, which in turn retrieves mountain-specific data located in the vicinity of the device determined by a specific range radius. For example, a camera location might be $\lambda_{cam} = 47.689254°$, $\varphi_{cam} = 15.688808°$, $\psi_{cam} = 1,969$ m, and we look for POIs within the range radius around the camera using Linked Data.

4.1 Data Integration

We use multiple Linked Data sources that have to be replicated, aggregated, and consolidated as well. Since there are real-world entities described by different schemes and ontologies, there is a necessity to define rules how to integrate these data sets respectively to identify identical or highly similar resources. Exemplary responses from GeoNames and LinkedGeoData are shown in Listing 1 and 2. These two descriptions represent a well-known peak in the Styrian Rax Alps - the *Dreimarkstein*. We will show how these two representations can be mapped using both textual and geographic information, even if the retrieved data do not coincide exactly. Thus, we define a composite metric constructed of a geospatial and text criterion using the Silk Link Specification Language [6] in order to determine objects considered as equivalent. Listing 3 illustrates a basic `LinkageRule` and therewith how resources of GeoNames (x) and LinkedGeoData (y) are matched using a measure of length as threshold for the geographic vertical and horizontal distance. In addition, `gn:name` respectively `rdfs:label` must correspond according to the Jaro distance metric for string similarity. If the weighted and aggregated similarities fall below a certain threshold (`<Filter threshold="0.85" limit="1"/>`), a link is being created using the `owl:sameAs` property.

Listing 1: Response from GeoNames

```
<http://sws.geonames.org/6940124/> gn:Feature [
  rdfs:isDefinedBy
    <http://sws.geonames.org/6940124/about.rdf> ;
  gn:name "Dreimarkstein" ;
  gn:featureClass gn:T ;
  gn:featureCode gn:T.PK ;
  gn:countryCode "AT" ;
  wgs84_pos:lat "47.70861" ;
  wgs84_pos:long "15.71583" ;
  wgs84_pos:alt "1948" ;
  gn:parentCountry <http://sws.geonames.org/2782113/> .
] .
```

Listing 2: Response from LinkedGeoData

```
<http://linkedgeodata.org/triplify/node428866475>
  = <http://sws.geonames.org/6940124/> ;
  lgdo:ele "1960"^^xsd:float ;
  rdfs:label "Dreimarkstein" ;
  a <http://linkedgeodata.org/ontology/Peak> ;
  georss:point "47.708731 15.7154637" ;
  geo:lat "47.708731"^^xsd:decimal ;
  geo:long "15.7154637"^^xsd:decimal .
```

Listing 3: Silk LSL LinkageRule

```
<LinkageRule>
  <Aggregate type="average">
    <Compare metric="jaro" weight="2">
      <Input path="?x/gn:name" />
      <Input path="?y/rdfs:label" />
    </Compare>
    <Compare metric="numSimilarity" weight="1">
      <Input path="?x/wgs84_pos:lat" />
      <Input path="?y/geo:lat" />
    </Compare>
    <Compare metric="numSimilarity" weight="1">
      <Input path="?x/wgs84_pos:long" />
      <Input path="?y/geo:long" />
    </Compare>
    <Compare metric="numSimilarity" weight="1">
      <Input path="?x/wgs84_pos:alt" />
      <Input path="?y/lgdo:ele" />
    </Compare>
  </Aggregate>
</LinkageRule>
```

4.2 Calculation of Screen Coordinates

Continuing our example scenario, the *Dreimarkstein* POI has the geographic location $\lambda_{poi} = 47.70861°$, $\varphi_{poi} = 15.71583°$ and the altitude $\psi_{poi} = 1,948$ m. As orientation and field-of-view data[14], we assume a camera azimuth of $\alpha = 45°$, a camera pitch of $\rho = 5°$, a horizontal angle of view of $H_{aov} = 45°$, and a vertical angle of view of $V_{aov} = 30°$.

Using Definition 1, we calculate a Vincenty distance of $z = 2,957.228$ m, a final bearing angle of $\beta = 43.3129°$ (rounded), and an inclination angle of $\theta = -0.4069°$ between camera and POI. We use Definition 2 to determine $l_{border} = 22.5°$, $t_{border} = 20°$. Subsequently, Definition 3 yields $x_{deg} = 20.8129°$ and $y_{deg} = 20.4069°$ (both rounded). Finally, using Equation 1 and 2 and assuming a screen resolution of 800x480, we obtain the 2D coordinate $x_{screen} = 370$, $y_{screen} = 327$ in pixels (rounded to integers) which is used as the center point for the POI information overlay on the screen for visualization.

Our interface implements the *magic lens metaphor* [28] which utilizes the mobile device as a see-through interface into the AR view, combining footage from the real world via the device's camera with virtual information overlays correlating to the current real-world view. Compared to other AR interface approaches such as wearable glasses, this approach reduces both the amount and the cost of hardware the user needs to carry along and makes it suitable for mobile use in countryside and mountain environments. The mathematical model serves as connecting link between the real-time camera footage and the virtual information overlays.

[14]We use degrees instead of radians in this example

5. EVALUATION

We have conducted several experiments in oder to evaluate the proposed approach in terms of performance and data quality of the used LOD data sources. Since GPS-related performance measures are tremendously varying from device to device, we spare location gathering issues for this evaluation and realize our performance measuring by determining the amount of time needed for addressing, querying and aggregating the utilized data graphs of 3 LOD sources, the computational costs of the calculation of screen coordinates from raw LOD triples as well as the expense for visualizing the pre-processed POIs in the field of view represented by the Augmented Reality application. Moreover we analyzed 3 LOD data sources and evaluated the availability of resources in general as well as the respective ontology related type definitions. We provide semantically identical concepts/classes as a proposition for reasonable consolidation of LOD triples that are potentially useful for being utilized as POIs.

5.1 Data Sources

Even though LOD sources have a non-proprietary and heterogeneous nature and in general cannot be deemed to comply with certain quality aspects such as correctness or consistency we however examined 3 LOD sources, the data sets as well as the underlying ontologies, with the intention to determine similar semantic concepts that describe a *summit* entity. We define a *summit* as a *topologically highest elevation or ridge atop a mountain in the country* and try to identify the most appropriate equivalence as shown in Table 2.

The LinkedGeoData ontology (prefix: `lgdo`) covers 1,293 concepts that describe various spatial aspects which have been derived from OpenStreetMap. There is only one class `lgdo:Peak` that matches our definition of a summit and fits perfectly. DBpedia (prefix: `dbpo`) offers 27,895 instances of the class `dbpo:Mountain` which is considered to be equivalent to a `summit` as defined, thus `lgdo:Peak` matches `dbpo:Mountain` in our scenario. GeoNames (prefix: `gn`) does not offer a public SPARQL endpoint that can be used to retrieve POI descriptions. However, there is a variety of webservices that fulfill our requirements even if the data is served as XML and needs further processing steps for consolidation with LinkedGeoData and DBpedia. The nodes in GeoNames are identified by the type `gn:Feature` having `gn:featureClass` and `gn:featureCode` as properties. According to the GeoNames taxonomy, the feature class `gn:T` covers mountain, hill, and rock concepts. We use the feature codes `gn:T.PK` and `gn:T.PKS` having the SKOS definition "pointed elevation(s) atop a mountain, ridge, or other hypsographic features".

5.2 Performance

In order to evaluate the performance of our approach,

Table 2: Equivalent LOD Concepts

Data Source	Concept	Instances
LinkedGeoData	`lgdo:Peak`	271,990
DBpedia	`dbpo:Mountain`	27,895
GeoNames	`gn:T.PK` ∪ `gn:T.PKS`	40,205

we defined 4 time deltas Δt_q, Δt_m, Δt_c, and Δt_q that are obtained directly through our implemented prototype. We analyzed the runtime behavior of the main constituents while covering the main aspects of the data flow processes as shown in Table 3 depending on the number of POIs N_{POI} respectively the range radius r which defines the maximum POI distance to be considered.

Table 3: Time Deltas

Δt_q	Query process
Δt_m	Triple merge, aggregation, and consolidation
Δt_c	Geometrical transformation of POI coordinates
Δt_v	POI Visualization

We conducted test runs for six different locations, which have been chosen representatively for topographic characteristics in Austria by classifying them according to the terms *urban*, *rural*, and *alpine*. For these places, the range radius has been varied from 1 km to 75 km in order to evaluate the scalability and the temporal costs. While Δt_q and Δt_m strongly depend on the camera location, the local area characteristics and thus the chosen range radius, Δt_c as well as Δt_v are evaluated merely in relation to the number of POIs by performing 100 test runs for each POI number. All tests have been performed on the current reference device Samsung Galaxy Nexus running Android OS version 4.0.2.

Table 4: Test Locations & Characteristics

Location	urban	rural	alpine
48.231077, 16.294781	✓		
47.391435, 13.048575		✓	✓
47.359699, 13.639694			✓
47.073542, 15.437586	✓	✓	
47.688706, 15.799706		✓	✓
47.272154, 11.396935	✓		✓

Figure 3 illustrates the coherence between the range radius and the number of retrieved POIs as well as the amount of time needed for querying LOD sources. Δt_m as well as Δt_v are shown in Figure 4. The consolidation of resource descriptions takes a maximum of 772.46 ms for a range radius of 75 km and the mean of 802 POIs, which is acceptable for a practical case study. Furthermore, it illustrates that there exists an almost linear correlation between the number of POIs and the respective visualization time. Whereas the number of concurrently visualized POI markers is technically limited by display resolution and camera characteristics, the visualization process scales rather assertive.

The calculation respectively the transformation from the gathered POI location information into the mathematical model is given by Δt_c, which amounts to a maximum calculation time of 75.23 ms, a mean of 0.69 ms, and a minimum of 0.03 ms.

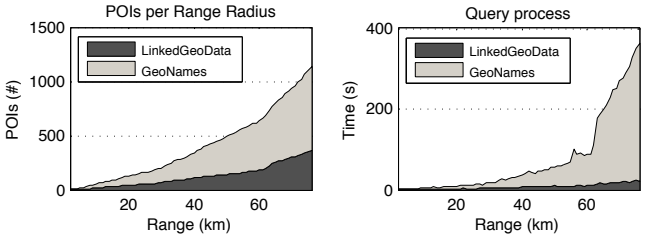

Figure 3: Number of POIs N_{POI} and Query time Δt_q

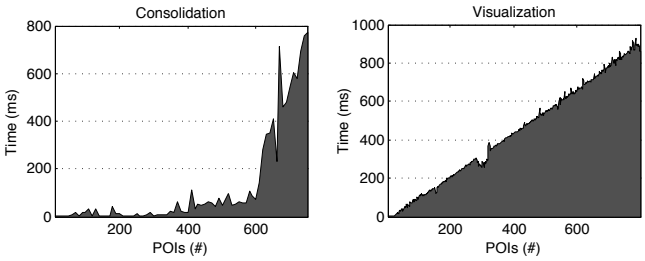

Figure 4: Consolidation Δt_m and Visualization Δt_v

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a computational model that exploits locally gathered sensor data in combination with geographic information provided by multiple Linked Data sources in order to enable the visualization of points of interests in Augmented Reality interfaces. Linked Data graphs are replicated, aggregated, and consolidated directly on mobile devices, whereby it simplifies the geometric transformation from POI locations in a 3-dimensional space to normalized 2D screen coordinates, which can be easily used for the visualization. Our experiments demonstrate the applicability for real-world use cases and the effectiveness of our approach.

As future work, we plan to extend and generalize the computational model to consider and interpolate on movement and rotation patterns, thus supporting camera roll and predictive POI fetching. On a technical basis, the graphics processing units (GPUs) of mobile devices can be utilized to accelerate calculation and transformation processes in order to increase the amount of POIs that can be simultaneously processed and visualized on the screen. We further plan to implement intelligent caching and replication strategies that reduce replication and consolidation overheads. Moreover, we intend to explore possibilities of enhancing the accuracy of tracking techniques by using content-based features.

Acknowledgements

We thank Silvia Fürst for implementing parts of the proof-of-concept prototype.

7. REFERENCES

[1] F. Ababsa. Advanced 3d localization by fusing measurements from GPS, inertial and vision sensors. In *IEEE int. conf. on Systems, Man and Cybernetics*, SMC'09, pages 871–875, Piscataway, NJ, USA, 2009.

[2] R. Azuma. A survey of augmented reality. *Presence*, 6(4):355–385, 1997.

[3] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *Computer Graphics and Applications, IEEE*, 21(6):34 –47, nov/dec 2001.

[4] P. Belimpasakis, P. Selonen, and Y. You. Bringing user-generated content from internet services to mobile augmented reality clients. In *Virtual Reality Workshop (CMCVR)*, pages 14 –17, 2010.

[5] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data: Principles and State of the Art. In *17th World Wide Web Conference (WWW2008)*, April 2008.

[6] C. Bizer, J. Volz, G. Kobilarov, and M. Gaedke. Silk - a link discovery framework for the web of data. In *18th International World Wide Web Conference*, 2009.

[7] S. Feiner, B. Macintyre, and D. Seligmann. Knowledge-based augmented reality. *Commun. ACM*, 36(7):53–62, July 1993.

[8] Y. Genc, S. Riedel, F. Souvannavong, C. Akinlar, and N. Navab. Marker-less tracking for AR: a learning-based approach. In *Mixed and Augmented Reality, 2002. ISMAR 2002. Proceedings. International Symposium on*, pages 295 – 304, 2002.

[9] W. Guan, S. You, and U. Neumann. Gps-aided recognition-based user tracking system with augmented reality in extreme large-scale areas. In *2nd ACM conference on Multimedia systems*, pages 1–10, New York, NY, USA, 2011. ACM.

[10] P. Gupta, N. da Vitoria Lobo, and J. Laviola. Markerless tracking using polar correlation of camera optical flow. In *Virtual Reality Conference (VR), 2010 IEEE*, pages 223 –226, march 2010.

[11] J. Ha, K. Cho, and H. S. Yang. Scalable recognition and tracking for mobile augmented reality. In *9th ACM SIGGRAPH Conf. on Virtual-Reality Continuum and its Applications in Industry*, VRCAI '10, pages 155–160, New York, NY, USA, 2010. ACM.

[12] G. W. Hart, S. Levy, and R. McLenaghan. Spherical geometry & trigonometry. In D. Zwillinger, editor, *CRC standard mathematical tables and formulae*, chapter 4.19. CRC Press, 31st edition, 2003.

[13] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 1st edition, 2011.

[14] I. Jacobs and N. Walsh. *Architecture of the World Wide Web, Volume One - W3C Recommendation 15 December 2004*. World Wide Web Consortium.

[15] W. E. Mackay. Augmented Reality: linking real and virtual worlds: a new paradigm for interacting with computers. In *Conf. on Advanced Visual Interfaces*, pages 13–21, New York, NY, USA, 1998. ACM.

[16] T. Miyashita, P. Meier, T. Tachikawa, S. Orlic, T. Eble, V. Scholz, A. Gapel, O. Gerl, S. Arnaudov, and S. Lieberknecht. An augmented reality museum guide. *Mixed and Augmented Reality, IEEE / ACM International Symposium on*, 0:103–106, 2008.

[17] J. Oh, M.-H. Lee, H. Park, J.-I. Park, J.-S. Kim, and W. Son. Efficient mobile museum guidance system using augmented reality. In *IEEE Symposium on Consumer Electronics*, pages 1 –4, 2008.

[18] R. Paucher and M. Turk. Location-based augmented reality on mobile phones. In *Vision and Pattern Recognition Workshops*, pages 9 –16, 2010.

[19] N. Ravi, P. Shankar, A. Frankel, A. Elgammal, and L. Iftode. Indoor localization using camera phones. In *7th IEEE Workshop on Mobile Computing Systems and Applications*, pages 1 –7, 2006.

[20] G. Reitmayr and T. Drummond. Going out: robust model-based tracking for outdoor augmented reality. In *5th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '06, pages 109–118, Washington, DC, USA, 2006. IEEE.

[21] G. Reitmayr and T. W. Drummond. Initialisation for visual tracking in urban environments. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '07, pages 1–9, Washington, DC, USA, 2007. IEEE.

[22] V. Reynolds, M. Hausenblas, A. Polleres, M. Hauswirth, and V. Hegde. Exploiting linked open data for mobile augmented reality. In *W3C Workshop: Augmented Reality on the Web*, June 2010.

[23] D. Stricker. Tracking with reference images: a real-time and markerless tracking solution for outdoor augmented reality applications. In *Virtual Reality, Archeology, and Cultural Heritage*, VAST '01, pages 77–82, New York, NY, USA, 2001. ACM.

[24] G. Takacs, V. Chandrasekhar, N. Gelfand, Y. Xiong, W.-C. Chen, T. Bismpigiannis, R. Grzeszczuk, K. Pulli, and B. Girod. Outdoors augmented reality on mobile phone using loxel-based visual feature organization. In *1st int. conf. on Multimedia Inf. Ret.*, pages 427–434, New York, NY, USA, 2008. ACM.

[25] C. Van Aart, B. Wielinga, and W. R. Van Hage. Mobile cultural heritage guide: location-aware semantic search. In *17th int. conf. on Knowledge engineering and management by the masses*, EKAW'10, pages 257–271. Springer, 2010.

[26] T. Vincenty. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey Review*, 23(176):88–93, 1975-04-01T00:00:00.

[27] D. Wagner, A. Mulloni, T. Langlotz, and D. Schmalstieg. Real-time panoramic mapping and tracking on mobile phones. In *Virtual Reality Conference (VR), 2010 IEEE*, pages 211 –218, 2010.

[28] D. Wagner, T. Pintaric, F. Ledermann, and D. Schmalstieg. Towards massively multi-user augmented reality on handheld devices. In *3rd Int. Conference on Pervasive Computing*, 2005.

[29] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. Real-time detection and tracking for augmented reality on mobile phones. *Visualization and Computer Graphics, IEEE*, 16(3):355 –368, 2010.

[30] S. Zander and B. Schandl. Context-driven rdf data replication on mobile devices. *Semantic Web Journal Special Issue on Real-time and Ubiquitous Social Semantics*, 1(1), 2011.

[31] F. Zhou, H. B.-L. Duh, and M. Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. In *7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '08, pages 193–202, Washington, DC, USA, 2008. IEEE Computer Society.