<table>
<tr><td>Project acronym: LADIO

Project number: 731970

Work package: Management

Deliverable number and name:

D6.2: Quarterly Management Report 2</td><td>Title: Quarterly Management Report 2

Work Package: WP6

Version: 1
Date: June 30, 2017

Author:
Carsten Griwodz</td></tr>
<tr><td>Type:
[X] Report
[ ] Demonstrator, pilot, prototype
[ ] Website, patent filings, videos, etc.
[ ] Other</td><td>Co-Author(s):
All partners

To:
Albert Gauthier, Project Officer</td></tr>
<tr><td>Status:
[ ] Draft
[ ] To be reviewed
[ ] Proposal
[X] Final / Released to EC</td><td>Confidentiality:
[X] PU – Public
[ ] CO – Confidential
[ ] CL - Classified</td></tr>
<tr><td colspan="2">Revision:
Final</td></tr>
<tr><td colspan="2">Contents:
Deliverable 6.2: Quarterly Management Report 2</td></tr>
</table>

Human Resources

SRL

The project is coordinated by Professor Carsten Griwodz. Konstantin Pogorelov, Jonas Markussen, Gunleik Groven, Stian Zeljko Vrba and Kristian Skarseth all contribute to the innovation and scientific work. Elisabeth Andersen contributes to the administrative work. Gunleik Groven, Stian Zeljko Vrba and Kristian Skarseth were formerly employed by LABO Mixed Realities.

MIK

The project is led at MIK by Benoit Maujean, R&D Manager. Fabien Castan, as senior lead developer is the technical manager of LADIO project, to coordinate the partnership in all development-related project work. Yann Lanthony and Nicolas Rondaud as senior software engineers as well as Jean Melou (PhD student) will contribute to the development of LADIO tools. Thomas Eskénazi, as software architect, will contribute to the data model and API definition. Grégoire De Lillo has joined the team at the end of March, as an R&D intern from IMAC engineering school.

INP

The project at INP is led by Pierre Gurdjos. The innovation and scientific work is supervised by Sylvie Chambon, Vincent Charvillat, Simone Gasparini and Géraldine Morin, all senior researchers. Hatem Rashwan is the full-time post-doc researcher dedicated to the project. Clément Debize is the research engineer.

CTU

The project at CTU is led by Dr. Tomas Pajdla. Main innovative and scientific work is carried out by Ph.D. students Cenek Albl and Michal Polic. Additional contributions come from MSc. students Pavel Trutman, Jan Krcek, and Oleh Rybkin.

Project Progress

WP1 - Onset Data Acquisition

T1.1 - Realtime device data and metadata acquisition (planned duration month 1-6)

During the POPART project, we have created a software, called *appCutie* (Figure 1), to monitor the Live Action camera and the 2 witness cameras. In the 2nd quarter of LADIO, MIK has improved the User Interface to clarify the parameters display, fix some accumulated latency in the video feedback and automatize the calibration process for Previz by automatically recording, downloading and processing the videos.

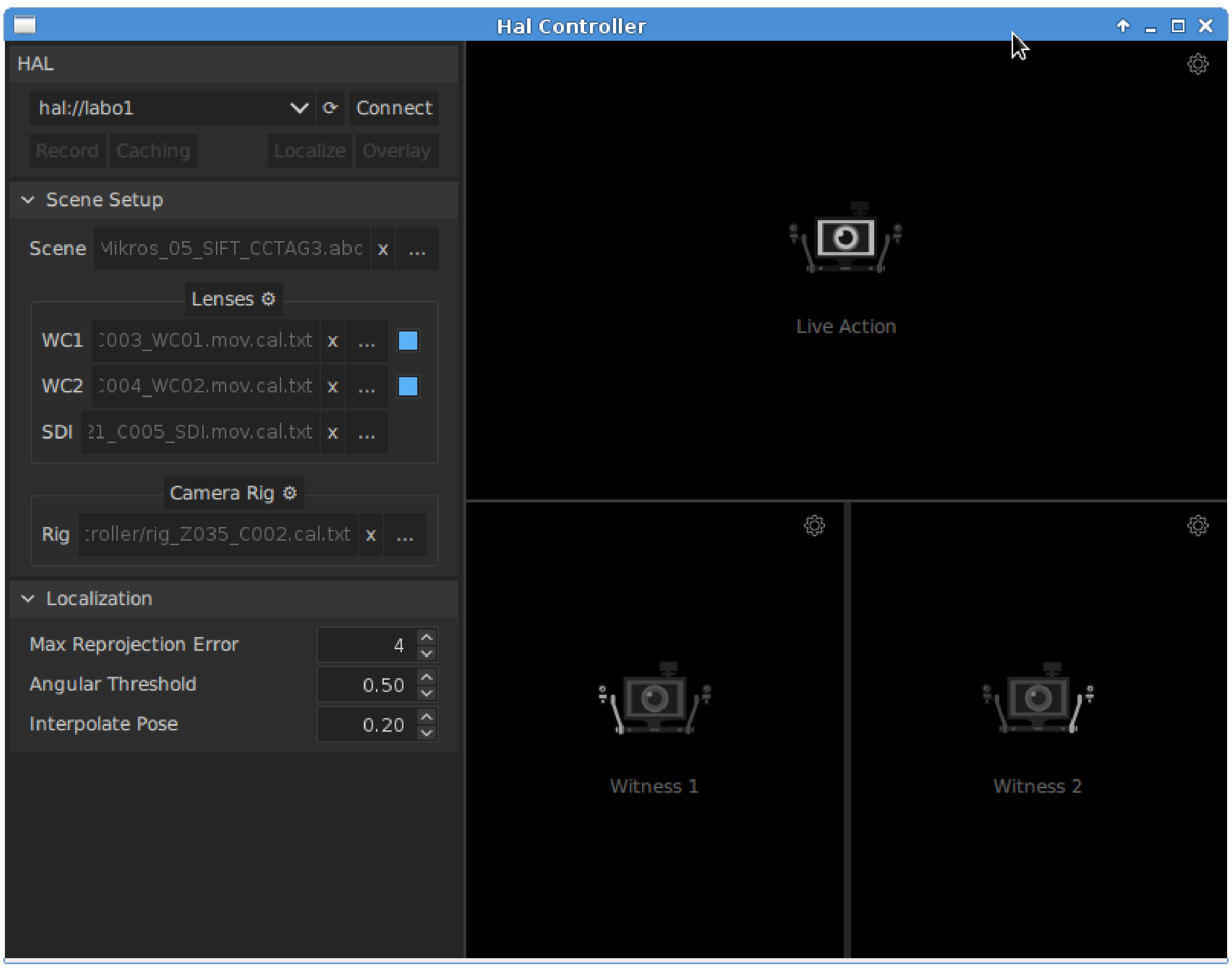


Figure 1 : new appCutie IHM

SRL has been working on the CamBox hardware and software development, as well as the generic software that will be used for both CamBox and MiniBox.

Task 1.1 (real-time data acquisition) has been implemented and delivered, but there remains one main obstacle: the problem of shutter-synchronized recording from multiple cameras (i.e., main camera + witness cameras). We are continuing to address this obstacle in the following ways:

- The manufacturer of witness cameras has indicated that hardware triggering (through a direct electrical pin) will be more reliable than software triggering through the USB bus.
- Following this lead, and after the discussions during the LADIO meetings in Toulouse, we have come up with a viable design that we're going to implement and test as soon as possible.

The above-mentioned design is two-fold:

1. Sync output of the main camera will be directly connected to the trigger input of the witness camera. In other words, the triggering of the witness cameras will not be routed through CamBox at all. Due to differing voltage levels at Main Camera and Witness Cameras sides, an active cable may be needed, but is simple to build.
2. Optionally: to robustly detect that a Witness Camera has dropped a frame, CamBox will be equipped with a dedicated GPIO card which will also receive the sync output from the Main Camera. By counting frames and measuring time between frames received from Witness Camera, we can infer whether a frame has been dropped.

We have also discussed the possibility of replacing the USB2 cameras with the USB3 models from the same manufacturer (Do3Think). Although using USB3 cameras would not solve the reliability of triggering via USB, these models have an internal buffer of 64MB, which further minimizes the chance of dropping the frames.

Because too much time has been spent on debugging various HW and driver issues on Linux, we have ported CamBox to the Windows platform. The port is fully functional. Windows is now the main development and testing platform, but we're trying to keep the code cross-platform whenever practical.

We have also added more flexible configuration of inputs (crop, overlays, etc.) and outputs (it is possible to encode each input stream to multiple output files with different characteristics).

We used CamBox to acquire a "difficult" to reconstruct dataset consisting of a Main Camera recording and two Witness Camera recordings. We have also acquired a LiDAR survey corresponding to the camera recordings.

T1.2: Distributed Data Management
(planned duration months 7-9)

SRL has begun work on the design of Task 1.2 (distributed data management). Work towards the distributed data management has started with a decision about interaction patterns. CamBox/MiniBox units require a unique identity. They join the network on set using DHCP, which provides them IP address and time server. They announce their presence using MDNS, which enables other devices including the SetBox to recognize that they are presence and what kind of recording device they represent. The SetBox can subscribe to the data streams from these boxes (automatically or with user intervention).

It is not mandatory that data from all CamBox/MiniBox devices is collected by network transfer. It is desirable because the alternative, using memory cards for data movement, can easily be forgotten. A REST API has been developed to expose the data which is available on the CamBox/MiniBox, and the files will be downloadable over HTTP. When network download is not available, *rsync,* which supports incremental transfer of entire directories, is the primary option to move the data from the CamBox/MiniBox to an external drive. Rsync is desirable because our main cameras record frames in individual files.

Since all metadata, including that needed for distribution, is stored in XML documents (ref WP2), we will need to query an XML database for documents matching various criteria. For this purpose we have decided to use XQuery, an SQL-like language standardized by W3C. Because XQuery is just a specification, we need an engine to run queries. We have looked for an engine that supports C++ and is open-source, and have found the following candidates:

- Zorba (http://zorba.io; Apache License v2).
- BerkeleyDB XML by Oracle (https://www.oracle.com/database/berkeley-db/xml.html; SleepyCat license, a dual license which is free to use in open-source projects with option to buy a commercial license for closed-source projects).
- XQilla (http://xqilla.sourceforge.net/HomePage; Apache License v2). This is the XQuery engine underlying BerkeleyDB XML.

We are currently evaluating these three implementations.

A current discussion point is the logic that is used for local filename generation, which should follow a meaningful, human-readable naming scheme, and their placement in the naming hierarchy that is defined by the data model in Deliverable 2.1.

WP2 - Data Model and API

The work in WP2 was mainly focussed on tasks 2.1 and 2.2 in tight collaboration of MIK and SRL, with support from other partners.

T2.1 - Data Model and API

(planned duration months 1-6)

Task 2.1 resulted in Deliverable 2.1 (“Documentation of extensions for EBU CCDM, EBUCore and a newly defined REST API”). The task was meant to formalize the needs of LADIO for data collection and exchange. It is not concerned with the content encoding itself (addressed in Task 2.2), but with the metadata that is associated with content, the context in which content has been recorded or otherwise created, and relations between content. LADIO wants to retain information about recordings' position on a common timeline, spatial position and orientation of recording devices. This should be done in a standards-compliant manner.

Due to the known wide topical coverage of the EBU Class Conceptual Data Model (CCDM), which is an actively maintained and updated de-facto and de-jure standard forming the basis for a multitude of interchange formats that exist in the broadcasting world, LADIO took it as a starting point. The EBUCore schema derived from the CCDM has formalized a multitude of meta-information covering all kinds of productions, but for aspects of on-set recording and post-processing, its standardization is still at an early stage for several media. For example, while property changes of audio recording devices during a shoot are covered well in the current documents, LADIO found that the equivalent discussion about recording camera metadata that is changing over time during a shoot is still under discussion.

Members of Task 2.1 interacted with the EBUCore working groups on several occasions and the LADIO model derived from the CCDM model and delivered as Deliverable 2.1 has been presented at the EBUCore meeting in Zürich in June 2017. Earlier, LADIO was able to influence the development of an EBUCore extension that was proposed by EBU in collaboration with camera manufacturer Sony and the MediaInfo framework. By analysing the benefits of the proposed models for recording camera parameter updates and providing input to the EBU discussion, we achieved that a compact format suitable for non-real-time applications was selected for standardization alongside a much more expressive, memory-hungry format suitable for real-time application. This discussion is found in the deliverable under "Changing intrinsic camera parameters".

We proposed furthermore the integration of recording devices' relative spatial positions and movements into the camera parameter metadata described above. Due to the requirements of LADIO to maintain relative positions between recording devices and between the individual frames of a single recording devices, we proposed the specify positions of a frame (a)relative to a particular version of the reconstructed static scene, (b) relative to other frames of other recording devices, and (c), relative to different frames of the same recording device in the same recording session. This was extended with functional specifications that allow highly compact storage of positional information, but is probably most suitable for recording devices (we considered here only camera rigs) whose movement path is pre-specified and executed by a robot.

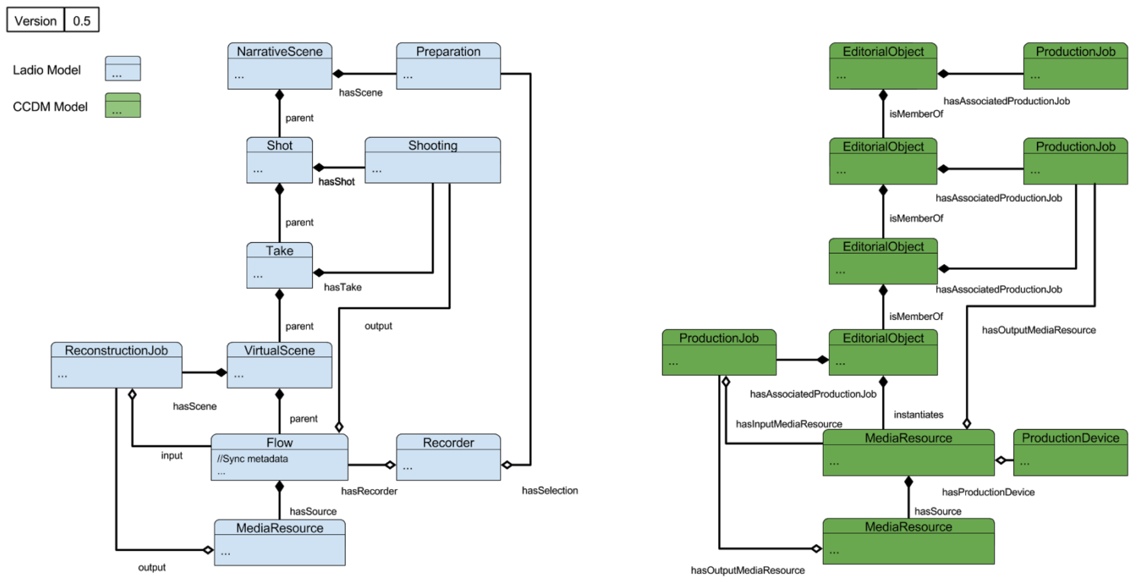


Figure 2: LADIO model as an instance of the EBUCore CCDM

To structure recording data and metadata in a terminology suitable for LADIO, we developed the LADIO model as an implementation of the EBUCore CCDM (Figure 2 [D2.1]). It's components are described in detail in the Deliverable 2.1. It was designed to structure all content that is recorded during the creation of a film by narrative scene. We developed a hierarchy of composite elements that reflects firstly the terminology on a film set, but is extended with our requirements for generating additional data both in advance of the actual live action shooting (in particular the 3D reconstruction of the set precedes the shoot) as well as after live action shooting. The model was developed in several rounds, with the final model including feedback and correction from the chair of the EBU's Metadata Developer Network, Tormod Værvågen.

In transferring content and metadata from a film set into a post-production process, a reduction will take place. A particular example is that it is usual that a single Take of a Shot is selected before a post-production team is requested to work on it (which in the LADIO model, is implemented as an attribute of Take). This leads to a divergence of everyday terminology between live action shooting and post-production teams that the LADIO tools must address. The proposed REST API will therefore mainly rely on business resources to simplify the tool-specific access.

The task has further worked on the problem of diverging clocks on a set. With the multitude of recording devices that LADIO will support, we face devices that are time-synchronized to a global clock, which on set is usually provided by the Audio department, and such that are either not time-synchronized (like the Ricoh Theta 360 camera) or synchronized to a different time sources (like the GPS clock of the Velodyne Puck LiDAR). Whereas the CamBox and MiniBox devices have been pointed as the software components in the LADIO system that observe device timecodes as well as on-set "wallclock" timecodes and as responsible for estimating clock drift and computing clock skew, Deliverable 2.1 specifies also where these clocks should be placed in the LADIO model. This is here illustrated in Figure 3 [D2.1].

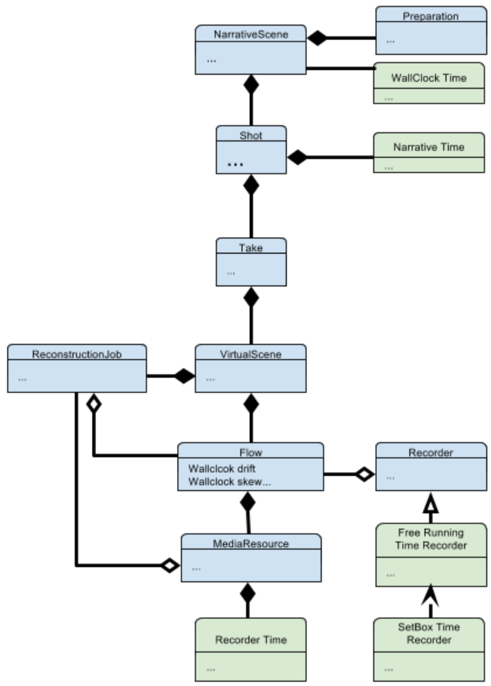


Figure 3: Clocks in the LADIO model

T2.2 - File and database formats for data storage (planned duration months 3-6)

Task 2.2 resulted in Deliverable 2.2 ("List of file formats to use unchanged and recommendation for filling gaps"). The task is complementary to task 2.1, focussing on the encoding formats for recorded content as well as generated content, not the metadata.

The task was bound to be influenced by the experiences of the consortium during initial stages of other work packages. Selecting a list of file formats that make sense in for integration into the LADIO system required an understanding of a new production pipeline with the open source version of CMPMVS at its core. The needs and abilities of CMPMVS, and the steps required to bring this research system into productive use, however, were not clearly understood by partners during the writing of the proposal. Also the specifications of hardware components that would be used in upgrading the results of the earlier POPART project were not known. Consequently, task 2.2 was relatively open-ended with some basic assumptions about LiDAR hardware and 360 cameras in particular, and under the assumption a final hardware selection would be made early in the project. However, so far, affordable LiDAR devices are not delivering acceptable quality and LADIO partners are still relying on third-party services. The Velodyne Puck at 8000 USD is the most promising right now, an example scan provided by a third party service is included in the dataset released in collaboration with Quine (http://www.quine.no/datasets/qb-rt-set-01/). The 360 camera that has so far been used by several LADIO partners, the Ricoh Theta, has several features that are desirable for LADIO, but its low resolution means that LADIO will remain explore further 360 cameras.

This led to decisions for Task 2.2 that are somewhat weaker than originally intended.

Video formats follow the expected recommendations, where main cameras and witness cameras store individual frames in a directory that is created for a Take (semantics of Take following Deliverable 2.1) in their original uncompressed format, which is specified in the metadata of the camera. To reduce the storage requirements of frames but remain compatible with the most relevant software tools, LADIO has selected Apple's ProRes format over Avid's DNxHD format. For consumer devices, LADIO supports H.264 (using the modern profiles for 4K video compression) and HEVC.

The recording format for LiDAR data that LADIO has chosen is specified by the American Society for Photogrammetry and Remote Sensing (ASPRS) and is named LASer (LAS). It is a lossless binary format with an origin in aerial scanning that is very well suited in terms of the information that it records, because it has been designed from the ground up to support moving LiDAR devices and recording samples with accuracy information. It has the ability to classify points following a user-defined mapping, record multiple returns, storing a GPS timestamp per point and store RGB data. These features make it more suited for storing original recording data than more recent point cloud formats discussed in conjunction with MPEG 4 part 23. However, we discuss regularly with the MPEG secretary and with contributors to MPEG 4 part 23, and may still revise this decision.

All consumer 360 cameras deliver as default streaming output an H.264 stream or HEVC stream that has been projected into a rectangular frame using an equirectangular mapping. As discussed in deliverable 2.2, this is very impractical mapping both in terms of data quality and in terms of performance. Based on our understanding of static camera rigs, LADIO would greatly prefer to record the data streams from the separate cameras that comprise the 360 camera and maintain their unchanging spatial relationship via the metadata as it is specified in deliverable 2.1. In this way, all recorded pixels would keep the maximum amount of information that can be derived from the input stream, and transfer this information unmodified to post-processing stages. However, for scenarios where 360 cameras should play a role in real-time processing on set (not a primary goal of LADIO, but still important for LADIO partners), simple hardware-assisted processing would benefit from the CubeMap projection, which is efficiently processed by all GPUs that support OpenGL (including OpenGL ES). A CubeMap projection created from original separate input streams per camera is a compromise yielding fast processing with moderate distortion and moderate efficiency. However, when a 360 camera provides only equirectangularly mapped frames, a further mapping into a CubeMap projection reduces quality further due to the additional interpolation step and is therefore only interesting for real-time approximations, and it should probably be avoided.

A requirement that was not understood during the writing of the proposal is the need to store accuracy information for 3D points for more than intermediate processing steps. In discussion involving all partners, we understood that accuracy information should be generated in 3D reconstruction for use in independent steps such as image-based camera localization. Consequently, Task 2.2 specified a file format for this. The results of the discussion called for an adaptive storage format for point clouds that could support several per-point representations for accuracy information, since Task 3.3 (“SfM: Accuracy and precision evaluation and improvement”) has not explored the best accuracy representation for LADIO yet. The point cloud will be stored in Alembic format, which is supported by a variety of tools already in use by LADIO partners. Alembic’s support for the transparent extension of point attributes allows these tools to function even in extended formats, even when the accuracy information is present.

T2.3: File and database representation for models and metadata (planned duration months 7-9)

The implementation arises from requirements and decisions made in Tasks 2.1 and 2.2. It has just been started.

WP3 - Advanced 3D reconstruction

T3.1 - SfM : LIDAR integration (planned duration months 7-15)

3D/3D LiDaR to SfM registration:
For matching SfM and LiDAR 3D clouds, as we mentioned in QMR1, the two main problems when registering LiDAR to SfM point clouds are how to deal with the differences in the point density and scaling. Regarding to density, INP used voxel grid to downsample the two point clouds with different voxel sizes. Regarding to scaling, our algorithm uses Principal Component Analysis (PCA) to generate two sets of cumulative contribution rate. The scale ratio is then defined as the scale factor that is used to rescale one point cloud to the other. In turn, we found out that the registration based on ISS+FPFH (ISS=Intrinsic Shape Signatures, FPFH=Fast Point Feature Histograms) descriptors yields the best registration. ISS features are local keypoints, which create view-independent signatures of the local surface patch. The algorithm scans the surface and chooses only points with large variations in the principal direction (the shape of the surface), which are ideal for keypoints. Then, each keypoint is described with a FPFH descriptor that considers only the direct connections between the current keypoint and its neighbors, removing additional links between neighbors. This dramatically decreases the algorithm complexity. Using matching between the extracted keypoints of both clouds, we refine the correspondence using using RANSAC. An ICP (Iterative Closest Points) technique is used finally to refine the final pose.

Furthermore, other descriptors are used for describing ISS key-points instead of FPFH, such as SHOT (Signature of Histograms of Orientations). It encodes information about the topology (surface) within a spherical support structure. However, the registration based on ISS+FPFH yields the best registration. In turn, SIFT descriptors for 3D points have been tested, however it can be used only for textured clouds. Figure 4 shows an overview of the proposed algorithm, in addition an example is shown in Figure 5.
The code is uploaded in the private repository:
https://github.com/alicevision/pointCloudAlignment

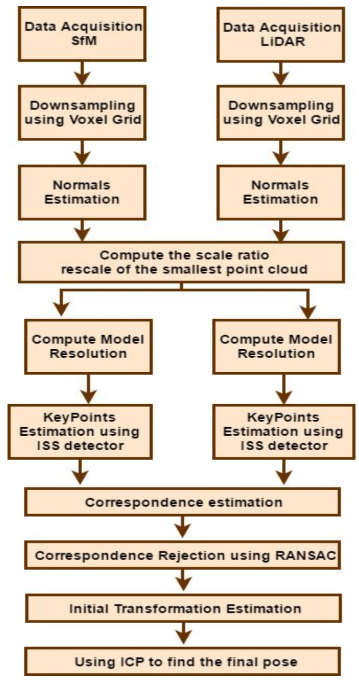
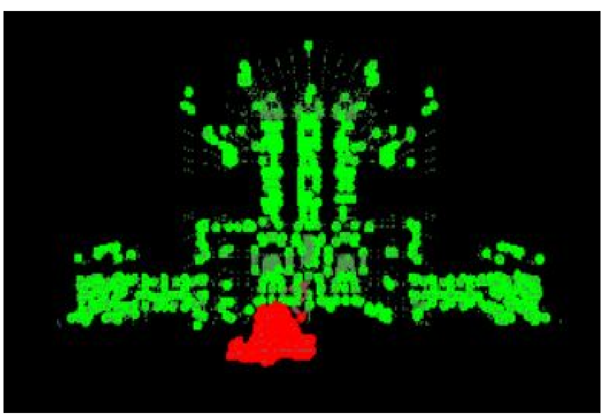
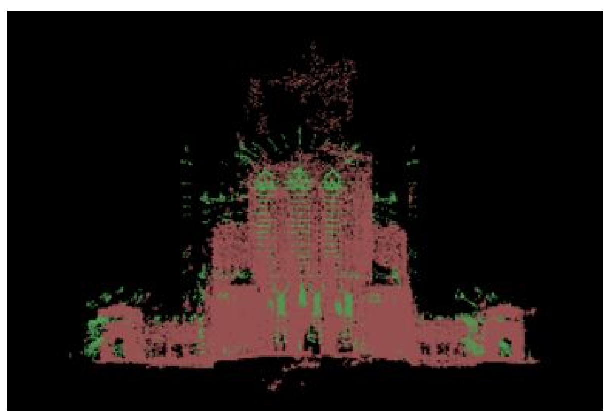


Figure 4: An overview of the proposed algorithm of SfM to LiDAR registration.



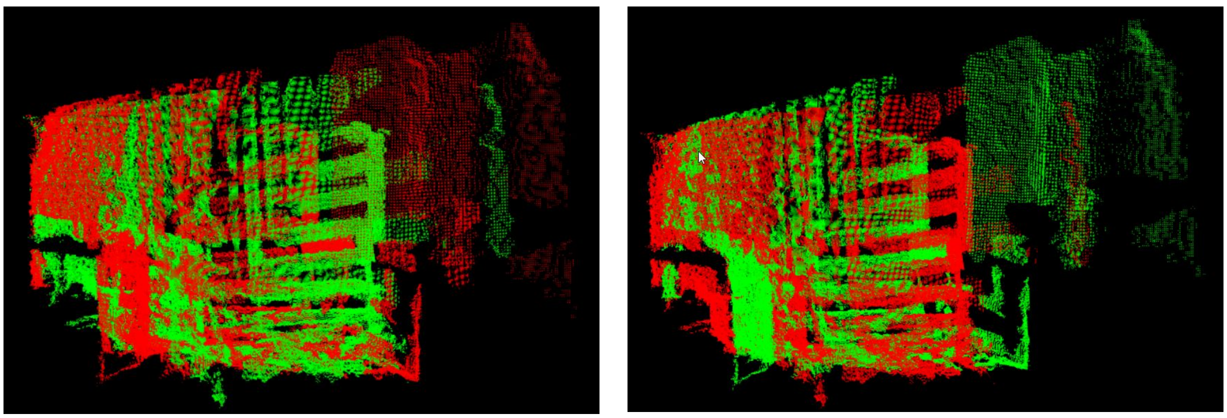
(Red) SfM point cloud (Green) LiDAR point cloud



The final registration between the two clouds.

Figure 5: SfM to LiDAR registration using the proposed algorithm (see text).

3D-3D LiDaR to LiDaR registration: we used a C++ implementation of PCL (**Point** Cloud Library) to incrementally register a series of point clouds two by two in order to to transform all the clouds in the first cloud's frame. This code is suitable for constructing the complete scene for LiDar scanning. First, we optionally downsampled our clouds that is useful in the case of large datasets. Curvature are then computed (for visualization purpose). An ICP (Iterative Closest Point) technique is finally used to align a pair of point clouds. The C++ implementation of PCL library proposes a formulation of the ICP algorithm that avoids the problems of outliers and and missing data often observed in 3D scans by formulating the registration optimization using sparsity inducing norms. This method yields good results with the evaluated database. Figure 6 shows an illustration.
The code is uploaded in the private repository:
https://github.com/alicevision/pointCloudAlignment



Two frames captured from LiDAR scanning.

(Green) the final aligned frame. (Red) the previous frame.

Figure 6: Registering two LiDAR scanning frames in one complete frame.

2D/3D image to model registration: We tackled the 2D/3D registration problem of one object in a cluttered scene in the case when only *one view* is available. In the proposed method, we represent each 3D model with a set of depth images rendered from a sphere around the object. Our 2D/3D registration problem consists of three subproblems:

- Detect features in a depth image of the 3D model generated using a randomly chosen 3D pose;
- Detect features in 2D intensity image;
- Match 2D features between the intensity and depth images from the 3D model and decide whether the 3D pose is the good one.

Regarding 1), we used features based on differential geometry and theoretically explain and then showed why measurements based on the difference of principal curvatures yields the best results.
Regarding 2), we theoretically explained why features based on the image gradient magnitude should be used to be matched with 1).

Regarding 3), we tried to reduce the number of matching by clustering the features of the depth images to a set of groups. We first estimate a coarse pose for the intensity image: the features extracted from an image is matched with each group. Then, we estimate a finer pose by selecting the first closest consistent views and then find the minimum and maximum elevation and azimuth angles and the distance between these views, and searching about the finer pose in this region by small changing in the angles to render new depth images. We are using HOG (histogram of oriented gradients) descriptors to represent color and depth images. We construct a set codebooks by clustering the descriptors related to the depth images; each codebook containing similar descriptors (views). The centers of the corresponding codebook are used to compute the coarse pose (initial pose). The matching will be performed with all views belonged the corresponding codebook to find the finer pose. We tested K-means, mean-shift to clustering the descriptors of depth images to N codebooks. However, the accuracy of the previously tested algorithm is better than the new one (!). Thus, we need to use different descriptors, maybe shape descriptors instead of HOG and find a robust clustering technique.

We still work on extending the method to work with a complete scene instead of a 3D model of object. In addition, We are working on the C++ implementation of the proposed method of 2D/3D registration based on the curviness saliency.

T3.2 - SfM: 360° cameras and camera rigs integration (planned duration months 3-9)

Our 3D reconstruction pipeline is based on still images. To use a video camera or a rig of synchronized cameras as inputs of the 3D reconstruction pipeline, MIK has developed a keyframes extraction software component. We first select the frames that are not too blurry (to avoid useless processing), then we ensure that the new candidate image is not too close to a previously selected one and finally select the image with the lowest blur level in its neighborhood.
This component has been released in AliceVision's openMVG library.

T3.3 - Improve SfM accuracy and precision (planned duration months 1-9)

As explained in “Towards Linear-time Incremental Structure from Motion”, Changchang Wu and in “Multistage SFM: Revisiting Incremental Structure from Motion”, Rajvi Shah, Aditya Deshpande, PJ Narayanan, the inlier/outlier ratio is better on the high-level features than on the low level features. So if we add more and more features, the inlier/outlier ratio will decrease and make the reconstruction more difficult.
So to improve the robustness on challenging datasets, MIK has added an option to combine multiple feature types, like SIFT, AKAZE, CCTAG, etc. This idea has been proposed in several papers, as [Ferber:2016] and [Abeles:2013].

We have also added a weighting strategy in the validation of the resectioning, so if you have only the minimal number of CCTag markers for the resectioning the result is considered valid, but you need more features to validate the pose estimation if you only have SIFT features for instance.
We have done an important refactoring to manage features and descriptors more easily and add the notion of feature types in the whole pipeline. We have integrated the SIFT GPU implementation (PopSift) released in POPART project into the SfM pipeline (previously it was only used in the camera localization part).

Improvements on Bundle Adjustment (BA). It is well-known that BA is a primary bottleneck in the Structure from Motion (SfM) task which becomes critical in large-scale reconstructions (e.g., in OpenMVG, if more than 500 views).

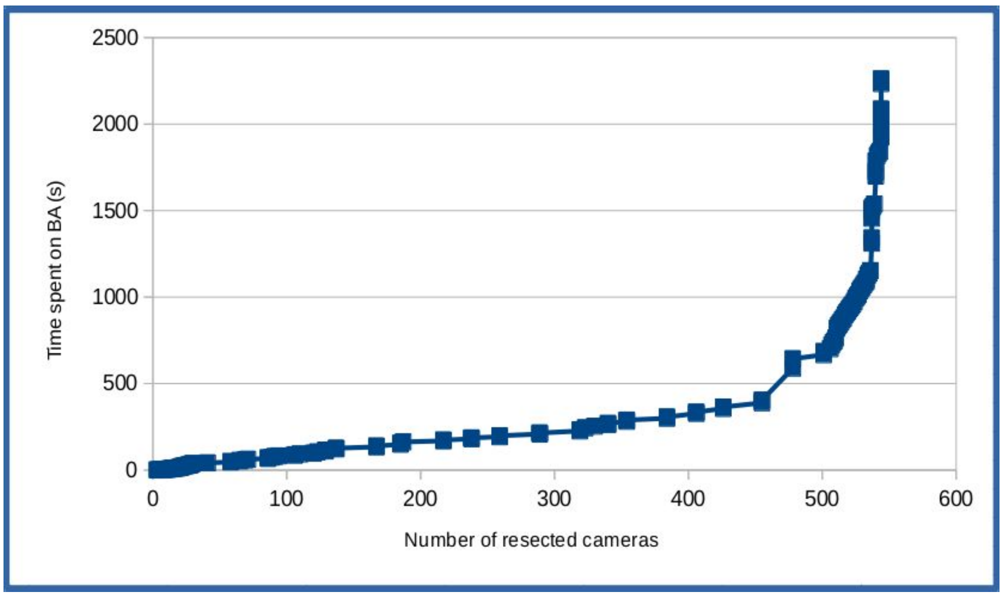


Figure 7: Performance of Bundle Adjustment on Levallois dataset

As LADIO uses an incremental SfM pipeline, particular attention must be given how to efficiently apply BA (see Figure 7). Existing approaches improving incremental SfM can be divided into two groups, (1) focusing on mathematical improvements to make the BA algorithm as efficient as possible or (2) reducing the numbers of unknowns i.e., of the considered camera and landmarks to be estimated in BA. INP and MIK recently investigated three solutions for (2) and are currently implementing them in OpenMVG. They fall into the class of approaches often referred to as "**Local Bundle Adjustment**" (LBA). The idea behind *Local Bundle Adjustment* (LBA) is that of introducing "active regions" and "passive regions" in the camera graph, i.e., in the graph where camera poses are vertices and edges are matched measurements of 3D landmarks between two poses. More precisely, INP and MIK defined

- the *active region* of a camera C w.r.t. a property P as the subgraph of vertices connected to camera C for which the property P holds ;

- the *passive region* of camera C as the subgraph of vertices not in the active region but which have edges connected to vertices of the active region i.e., of non-active cameras that have matched measurements of landmarks visible from active cameras.

The important point is that only cameras of the active region and landmarks visible from the active cameras are optimized in the LBA. INP and MIK studied the state-of-the art and selected three LBA solutions. The first LBA solution that INP and MIK has implemented is that of defining the property P as “the camera is within a given distance D from camera C in the camera graph”, where the distance refers to the number of edges in a shortest path from from camera C to the camera. It is basically the solutions suggested in the SLAM pipeline of [Mei:2011]. It is also similar to [Mouragnon:2006] where views are acquired one at a time from camera poses C_{t-D}, C_{t-D+1},..., C_t and a camera is at distance D from camera C_t if it is one of the last D-th cameras.

The second LBA solution that INP and MIK will implemented next is inspired by the work of [Sibley:2010] where the property P is “the camera has average reprojection error changes by more than a given threshold”. In other words, only views that change the “state” of the previous 3D reconstruction have to be considered.

There is the third solution, called “propagation of innovative information” [Steedly:2001], which takes over this idea of only optimizing in BA the new camera poses and landmarks that bring innovative information to the 3D reconstruction. The property P is “the camera is at distance D from camera C and the stopping criteria in the non-linear minimization for BA meets (P=0) or not (P=1) for the camera and all landmarks visible from it”.

T3.4 - SfM : Multi-body
(planned duration months 9-14)

Preliminary research on multi-body SfM has been continuing at the CTU. We have added another strategy for dealing with multiple moving objects to YASFM[1]. IT updates the existing YASFM code also provides a possibility to solve for the relative scale of different reconstructions if they can be reconstructed altogether in some subset of images. The work has been described in [Krcek:2017].

Development of multi-body SfM requires a new geometric verification for image matching. The standard approach that verifies tentative matches by finding the largest consistent model w.r.t. a single epipolar geometry is not applicable since more relative motions a represent. Using multiple epipolar geometries is also problematic since moving objects are often relatively flat and hence lead to a degeneracies in epipolar geometry computations. We therefore plan to test

[1] YASFM-Yet Another SfM (github.com/fsrajer/yasfm) is a Structure-from-Motion pipeline which was developed by students F. Srajer and C. Albl supervised by T. Pajdla, under Mozilla Public License Version 2.0. The YASM will be used to benchmark OpenMVG and to develop prototype of the multibody SfM.

and integrate the approach based on growing multiple homographies thaat was previously developed by the CTU in [Srajer:2016] and implemented in YASFM.

T3.5 - Lens calibration from 3D Reconstruction
(planned duration months 1-4)

We have finalized and verified our Matalb implementation of the minimal solvers for (i) computing a solution to camera absolute pose with unknown focal length (algorithm P4Pf) and for (ii) computing a solution to camera absolute pose with unknown focal length and unknown radial distortion (algorithm P4Pfr). A full pipeline including RANSAC was implemented in Matlab and all functionality was verified on test data from production. Unit test were constructed to allow verifiable reimplementation in C++. The algorithms were reimplemented in C++ including the unit tests and verified. First integration has been done into AliceVision's openMVG. We have obtained stable and efficient implementations of camera resectioning for partially caclibrated situation, Figure 8. See Deliverable D3.5 for more technical details.

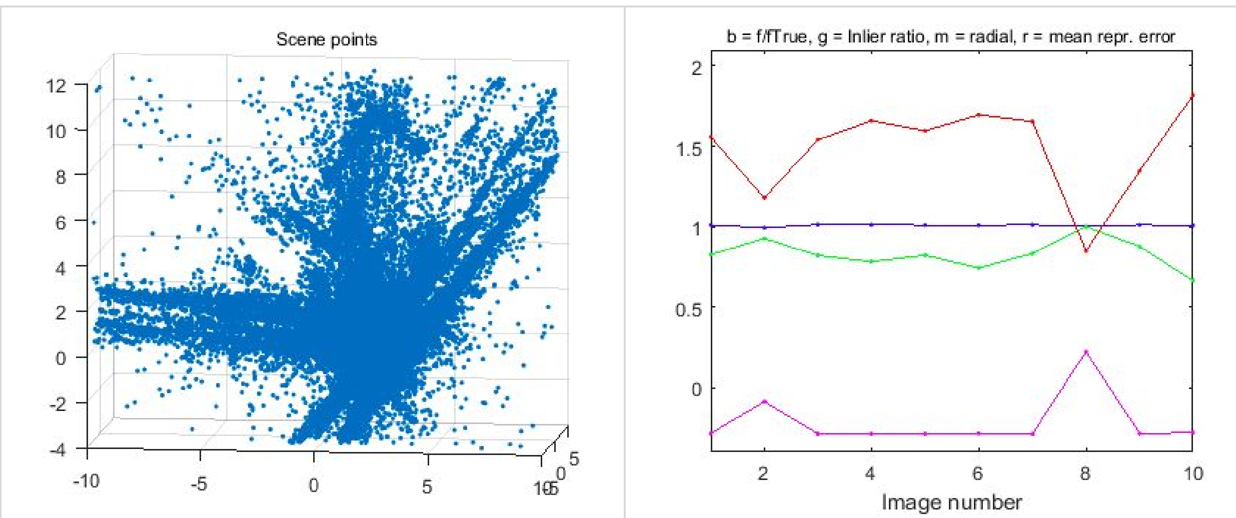


Figure 8. "Street" scene and (b) the comparison of the estimated focal length to the "ground truth" focal length (blue), inlier ratio (green), mean reprojection error (red) and estimated single-parameter division radial distortion coefficient (magenta).

We still plan to continue in this activity by implementing relative camera pose computation with single unknown focal length and/or radial distortion as well as investigate the case of computing the focal lengths of a camera system with any different and unknown focal lengths.

WP4 - Detailed 3D representation

T4.1 MVS Baseline
(planned duration months 3-9)

After the major refactoring of CMPMVS done in the first quarter, we have tested it on many datasets and fixed many small issues to improve the robustness of the library. We have completed the development work by addressing the texturing step to further improve the performances. We have also added the support to use multiple image resolutions at the same time. Previously, CMPMVS was only able to process images of the same size and that was a major limitation of the pipeline. This has required to change the way we preallocate memory buffers and access them across the whole MVS pipeline.

The support library PopSift, a SIFT implementation relying on NVidia's CUDA that was mainly developed during the POPART project, was improved for LADIO, where it can improve performance in early stages of SfM-MVS. A relevant bug has been removed, its speed and usability has been improved. Based on a renewed evaluation using the public Mikolajczyk-Schmid dataset[2], we have confirmed that PopSift still performs better in terms of correspondences, correct matches, scale change and compression resistance than OpenCV's SIFT implementation and nearly identical to VLFeat. At the time of testing, the extremely fast CudaSIFT had entirely different results. The latest version of CudaSIFT has not been tested. We confirmed that it can act as a faster drop-in replacement for VLFeat. We explored also a series of CPU-sided and GPU-sided descriptor matching algorithms.

T4.2 MVS GPU Optimization
(planned duration months 10-18)

SRL contributed early to MIK's and CTU's efforts on T4.1 because SRL requires a tight collaboration to learn about and understand the algorithms and code of CMPMVS. This early effort led to some insights about possible improvements, whose delayed implementation would also have penalized efforts in T4.1.

SRL continued CMPMVS code exploration and performance analysis. After CPU profiling of the current CMPMVS code we discovered a computation-intensive fragment of code in the plane sweeping implementation. After improvement of the plane sweeping code, we reduced the overall system execution time on the mini6 buddha dataset from 9 minutes 44 seconds downto 8 minutes 57 seconds. Observed reduction of CPU utilization from 257% to 214% means that OMP *parallel for* is not efficient enough in the particular code.

Next, the whole computational pipeline was profiled and no more too-CPU-intensive code that should be should be rewritten immediately were found.

To increase efficiency of joined CPU and GU profiling we tried one more method of automated NVTX wrapping with the function call injections. We found that this method is producing non-working and/or crashing code for complex projects like CMPMVS, and that it is suitable only for educational purposes. Instead we started a process of manual wrapping of pipeline substeps

[2] Mikolajczyk K. and Schmid C. A performance evaluation of local descriptors, Pattern Analysis and Machine Intelligence, 2005, Vol. 27, Number 10, pp. 1615-1630

and the most compute-intensive functions with NVTX to be able to perform joined CPU+GPU profiling later.

As the first result of performed memory usage analysis we have found unnecessary transfer operations between host and GPU memory in CUDA implementation of plane sweeping code.

Intensive code analysis and performance profiling of the whole CMPMVS pipeline allowed us to perform code and data optimization that will be described in detail in Deliverable 4.2.

We performed a performance evaluation of the original and optimized versions of the CMPMVS code using the 67 images *dataset_buddha*. Performance evaluation has been performed on the two following hardware configurations:

- development PC
    - Intel Core i7-6700K CPU @ 4.00GHz × 8
    - 16 GB DDR4 RAM
    - nVidia GeForce GTX TITAN X/PCIe/SSE2
    - Samsung SSD 256 Gb
    - Ubuntu 16.04 x64
- development/demonstration laptop
    - Intel Core i7-7820HK CPU @ 2.90GHz × 8
    - 32 GB DDR4 RAM
    - nVidia GeForce GTX 1080/PCIe/SSE2
    - Samsung SSD 512 Gb
    - Ubuntu 16.04 x64

Evaluation showed the following results:

test environment	The whole pipeline run time for different CMPMVS versions		Performance gain
	Original	CPU and data optimized	
Development PC	2:13:26 (8006 seconds)	1:36:49 (5809 seconds)	27.4%
Development/ demonstration laptop	2:28:09 (8889 seconds)	1:46:33 (6393 seconds)	28.1%
		Average performance gain:	**27.8%**

Finally, SRL started data usage and flow analysis. Initial experiments showed that even relatively small the 67 images *dataset_buddha* resulting in huge amount of data written to disk as intermediate pipeline steps' results. The amount of data written to disk is approximately 9

GB. Creation of such amount of data can affect the performance significantly, even while using modern SSDs, which can provide outstanding disk operations performance. To check the effects of this intensive disk workload, we performed experiment using in-memory-only RAM-disk as a target medium for the intermediate pipeline results. Experimental results showed that using the RAM-disk did not noticeably affect the measured pipeline execution time. Absence of performance changes in this case was caused by the limited amount of physical memory. Further investigation is required using higher amount of RAM.

WP5 - LADIO Application

T5.1 - LADIO Application Timeline (planned duration months 3-12)

As mentioned in T1. we have improved appCutie application, in order to monitor the Live Action camera and the 2 witness cameras. These improvements concern the layout of User Interface, and the lens and rig calibration integration (T5.1, T5.2).

Mik has also developed a few mockups (T5.1), regarding the global features of LADIO Application. Figures 9 and 10 show images of the mockups.

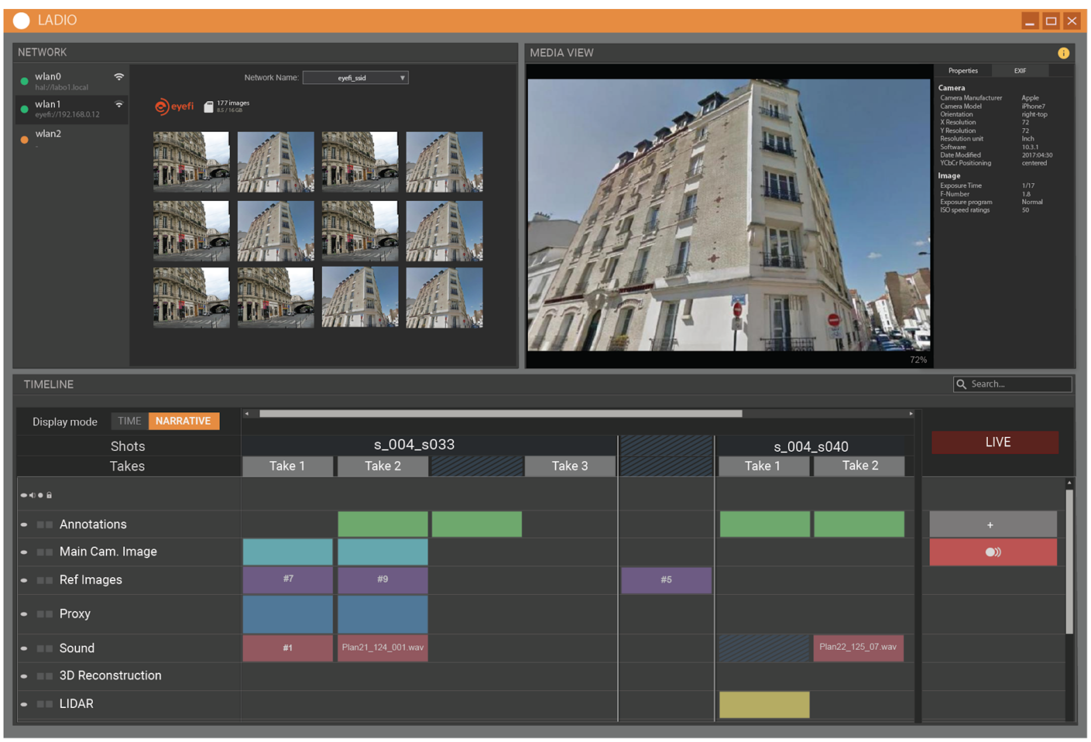


Figure 9: LADIO Application mockup (1)

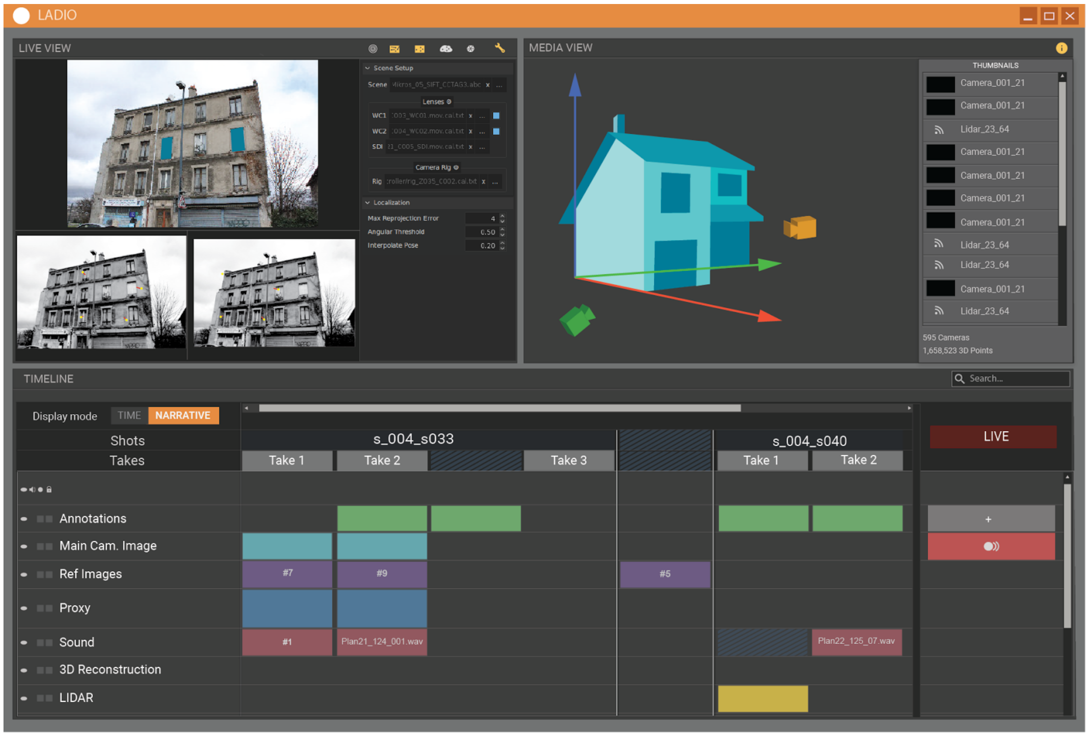


Figure 10: LADIO Application mockup (2)

We expect to invite an SME to the LADIO project who would contribute to work packages 1 and 5 in particular, and have a strong interest in the choices concerning the LADIO application. We are therefore considering the work on T5.1, which is developed as extensions and improvements to the *appCutie* application, rather as mockups meant for discussion about the expected functionality of the LADIO application.

T5.2 - LADIO Application Media Presentation (planned duration months 10-15)

During the POPART project, we have started a software called Meshroom to perform the 3D reconstruction with openMVG. We have started to extend this software with the integration of the MVS steps of the pipeline (from WP4). We have added the display of the final textured mesh using Qt3D.

The ambition of LADIO is to use this software on-set and not only in-house with renderfarm. So we have improved the nodal workflow with a caching mechanism.

WP6 - Management

T6.1 - Platform for internal communication

Most tools in our platform work as expected.

The #diary channel of Slack works as a technology watch and discussion, and has also proven to be a valuable tool for all aspects of reporting, because project members base other reports on the continuously tracked progress of the diary channel.

T6.2 - Reporting requirements

The revised work plan that has been in force since the accepted of the first Amendment clarified that LADIO is “looking for an European SME who would partner with us and take advantage of the LADIO open source releases, and integrate them into a comprehensive offering” (section 2.2.1 of Annex B). We believe that we have found this partner in the Norwegian SME Quine, and work on a new Amendment has been initiated. We see a particular challenge in the necessary inclusion of Quine IP as Background into LADIO.

T6.3 - Platform for dissemination of public project information

A new website alicevision.github.io has been started to explain the concepts behind the global photogrammetric computing vision pipeline enabled by LADIO project.
The project website ladioproject.eu has also been updated.

T6.4 - Data management planning

The LADIO project participates to “Open Access to research data” initiative. In this task, we have formulate a data management plan to make available data sets that can benefit the academic community and other users. The consortium will provide at least two of the three following data sets :

- LADIO_DATASET_Live_Action
- LADIO_DATASET_Advanced_3D_Reconstruction
- LADIO_DATASET_Multi_Body

LADIO’s released data sets will be permanently available on http://ladioproject.eu, the project's Github page https://github.com/alicevision and from Zenodo https://zenodo.org/collection/user-ladio (to be created) or similar data repository.
The data sets will be released to the general public under the license of Creative Commons Attribution-ShareAlike 4.0 International, allowing researchers and other interested parties to exploit the data sets.

The LADIO_DATASET_Live_Action is already accessible through the LADIO web page at https://griwodz.github.io/ladioproject/#datasets , but a few files will still be added. The dataset comprises challenging scenes for camera tracking, as well as still images, 360 recordings and LiDAR scans. The dataset will be uploaded to its final location when it is complete.

The academic partners of the consortium are also committed to “Open access publishing” (aka “Gold open access”) whenever this option is provided by the venues where we must publish to reach the highest impact of our results.

Workshops

In the second quarter, there have been workshops where the partners met to make progress on the development:

- March 24th, 2017 - Meeting between Tormod Værvågen, NRK (EBU’s Metadata Developer Network), SRL and MIK concerning the LADIO model; LADIO was invited to present at the EBUCore meeting June 27th-29th, 2017.
- March 20-24, 2017 - Fabien Castan (MIK) went to Simula to work on WP1 about sync issues and data model.
- March 29-31, 2017 - Tomas Pajdla (CTU) went to Mikros Image to discuss about MVS and finalize Lens calibration task (T3.4)

Plenary meetings

Project meetings are held at least every 6 months. The next meeting will be held in Toulouse at INP on June 28-29th, preceded by the first Advisory Board of the project.

Plenary audio / video conferences

In addition to these face to face meetings, the LADIO consortium holds a video conference every 4th week to keep track of the project progress:

- May 10th 2017 - Video conference
- April 12th 2017 - Video conference
- March 8th 2017 - Video conference

More spontaneous teleconferences are also held supporting the different work packages and the partners involved (e.g., a weekly teleconference between INP and MIK, bi-weekly conferences concerning Task 2.1 exclusively).

Status of deliverables

The following deliverables were submitted in this quarter. As agreed with the project officer, the deadline for D3.5 was extended with two weeks.

- D1.1 - CamBox and MiniBox. Deliverable submitted on May 31 2017.
- D2.1 - Model & API definition. Deliverable submitted on May 31 2017.
- D2.2 - File and database formats for data storage. Deliverable submitted on May 31 2017.
- D3.5 - Lens calibration from 3D Reconstruction. Deliverable submitted on April 10 2017.
- D6.5 - Data Management Plan. Deliverable submitted on May 31 2017.
- D6.2 - Quarterly management report #2. Will be submitted together with QMR1, in month 7.

Person Months Contributed to the Project

Participant	Names of staff	WP1		WP2		WP3		WP4		WP5		WP6		Total	
		Planned	Actual	*Planned*	Actual	*Planned*	Actual	*Planned*	Actual	*Planned*	Actual	*Planned*	Actual	*Planned*	Actual
SRL	Carsten Griwodz Gunleik Groven Stian Z. Vrba Kristian Skarseth Elisabeth Andersen	6,00	10,08	6,00	0,98	2,67	0,91	0,00	0,00	2,00	3,15	1,67	1,00	**18,33**	**16,12**
MIK	Benoit Maujean Fabien Castan Yann Lanthony Nicolas Rondaud Jean Mélou Thomas Eskénazi	0,00	1,20	5,00	0,54	5,33	1,05	2,86	7,25	2,00	2,02	1,00	0,61	**16,19**	**12,67**
INP	Vincent Charvillat Sylvie Chambon Simone Gasparini Pierre Gurdjos Geraldine Morin Hatem Rashwan	0,00	0,00	0,00	0,00	7,00	5,10	0,00	0,00	0,00	0,00	0,33	0,33	**7,33**	**5,43**
CTU	Tomas Pajdla, Cenek Albl, Michal Polic Oleh Rybkin Pavle Trutman	0,00	0,00	0,00	0,00	3,67	6.76	2,29	4.70	0,00	0,00	0,33	0,00	**6,29**	**11,46**
Total		**6,00**	**11,28**	**11,00**	**1,52**	**18,67**	**13,82**	**5,14**	**11,95**	**4,00**	**5,17**	**3,33**	**1,94**	**48,14**	

Figure 10: The cumulative budgeted and contributed person-months since project start for each partner sorted by Work Package.

References

[Abeles:2013] Peter Abeles, "Examination of Hybrid Image Feature Trackers"

[Ferber:2016] Marvin Ferber, Mark Sastuba, Steve Grehl, Bernhard Jung, "Combining SURF and SIFT for Challenging Indoor Localization using a Feature Cloud",

[Mei:2006] C. Mei *et al.* "RSLAM: A System for Large-Scale Mapping in Constant Time Using Stereo." IJCV 2011.

[Mouragnon:2006] É. Mouragnon *et al.* "Real TIme Localization and 3D Reconstruction." CVPR 2006.

[Sibley:2010] D. Sibley *et al.* "Vast-scale Outdoor Navigation Using Adaptive Relative Bundle Adjustment", IJRS 2010.

[Steedly:2001] D. Steedly *et al.* "Propagation of Innovative Information in Non-Linear Least-Squares Structure-from-Motion", CVPR 2001.

[Srajer:2016] F. Srajer. Image Matching for Dynamic Scenes. Master Thesis. CTU in Prague 2016.

	(https://dspace.cvut.cz/bitstream/handle/10467/64771/F3-DP-2016-Srajer-Filip-filip-srajer-diploma-thesis.pdf?sequence=-1&isAllowed=y)
[Krcek:2017]	J. Krcek. Multi-Body Structure from Motion. Master Thesis. CTU in Prague 2017. (https://dspace.cvut.cz/bitstream/handle/10467/68618/F3-DP-2017-Krcek-Jan-Master_s_thesis.pdf?sequence=-1&isAllowed=y)