Project acronym: LADIO Project number: 731970 Work package: Management Deliverable number and name: D6.2: Quarterly Management Report 1	Title: Quarterly Management Report 1 Work Package: WP6 Version: 1 Date: February 28, 2017 Author: Carsten Griwodz
Type: [X] Report [ ] Demonstrator, pilot, prototype [ ] Website, patent filings, videos, etc. [ ] Other	Co-Author(s): All partners To: Albert Gauthier, Project Officer
Status: [ ] Draft [ ] To be reviewed [ ] Proposal [X] Final / Released to EC	Confidentiality: [X] PU – Public [ ] CO – Confidential [ ] CL - Classified
Revision: Final	
Contents: Deliverable 6.2: Quarterly Management Report 1	

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.

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.

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 and Jan Krcek, and BSc. students Oleh Rybkin and Viktor Korotynskiy.

Project Progress

LABO Mixed Realities (partner code LAB) announced their intent to withdraw from the project at the LADIO kickoff meeting in Prague, 15-16 December 2016. As a result of LAB´s withdrawal from the Consortium, a first Amendment was prepared. Drafting started on December 22, it was signed and submitted on February 8, and accepted by the Commission on February 20. The termination report for LAB was submitted by the project coordinator on February 23.

The Amendment makes significant changes to the project plan.

- Cambox, Setbox and Minibox (WP1) will not be released as commercial products and instead developed as prototypes and released as open source components.
- The LADIO Application (WP5) is split into two parts: backend and frontend. The backend is released as open source, while the frontend application will remain proprietary to enable MIK servicing. This required the introduction of a new Deliverable D5.6.
- Key personnel of LAB is now represented in the advisory board and hired by SRL, to maintain progress and contact with expected adopters. Manpower has partly been redistributed among partners.
- The business plan of LAB cannot be pursued, but expected development is going to be released as part of LADIO's open core business model.
- Dissemination has been updated, including search for a new partner for productization.

The project is running as planned according the first Amendment, and each partner has key staff in place and an active WP leader. Details on each work package is given below.

WP1 - Onset Data Acquisition

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

- SRL has built a CamBox prototype hardware (Figures 2 and 3). The box is based on an Intel i7 6th generation CPU with a dedicated GPIO card supporting a number of interfaces (USB3, serial and Ethernet ports, etc.), input from SDI/HDMI, WiFi and 4G network interfaces. Work is in progress to add support for SD card readers and better contacts for interfacing to USB3 hardware. (Experience from the POPART project has shown that the standard USB3 plugs are not robust enough for use on a film set.)
- SRL surveyed hardware and other equipment (SDI cards, witness cameras, lenses) to use in the final product. We have found a promising board-level camera (Figure 1) to test, with the aim to build a more cost-effective and robust witness-camera solution than the PointGrey cameras have proven to be in practical use.

Figure 1: USB witness camera

- Finding an SDI capture card that fills all our needs have proven difficult. Particularly getting the entire metadata set in realtime over SDI is more challenging than envisioned.
- CamBox is currently running on CentOS linux 7.2, but we have concluded that SW built in POPART (HAL) may need to be ported to the Windows platform because of wider support by HW manufacturers. S. Vrba has investigated the feasibility of the port, and concluded that the port is feasible: there is more risk in changing the capture HW (SDI cards and witness cameras) than in changing the OS platform.

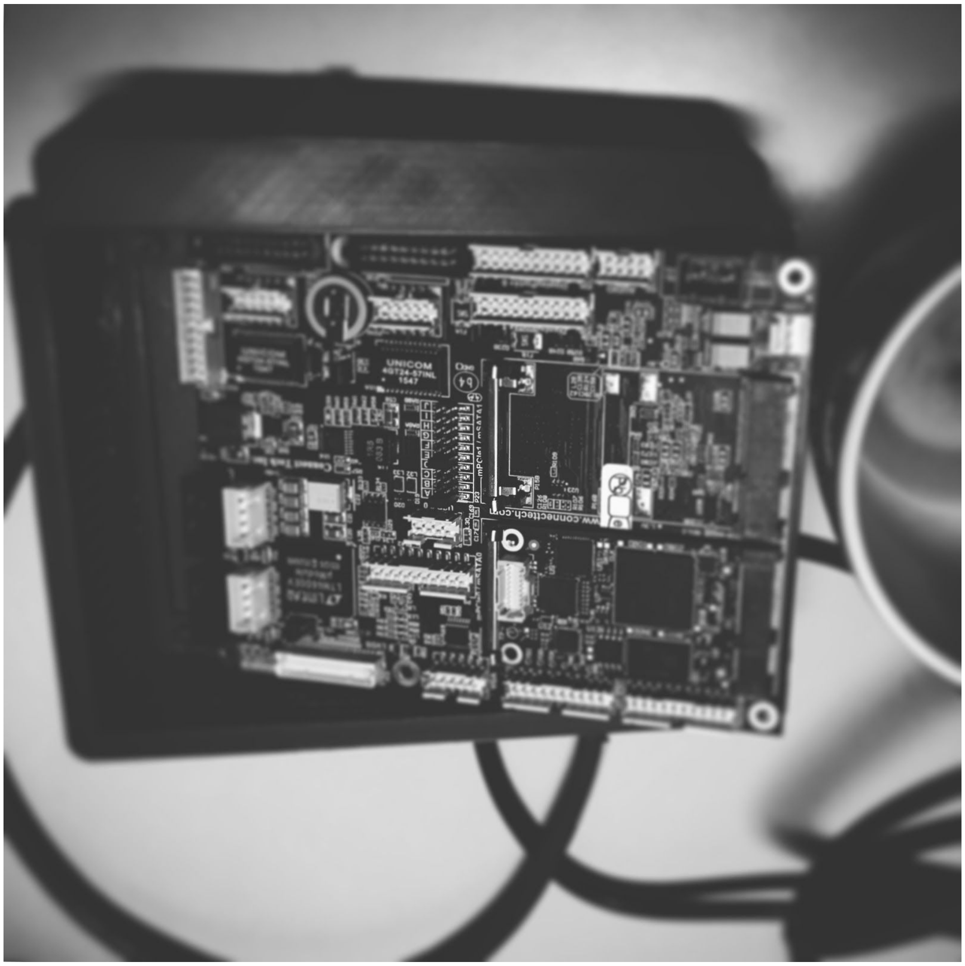


Figure 2: mainboard for the new CamBox

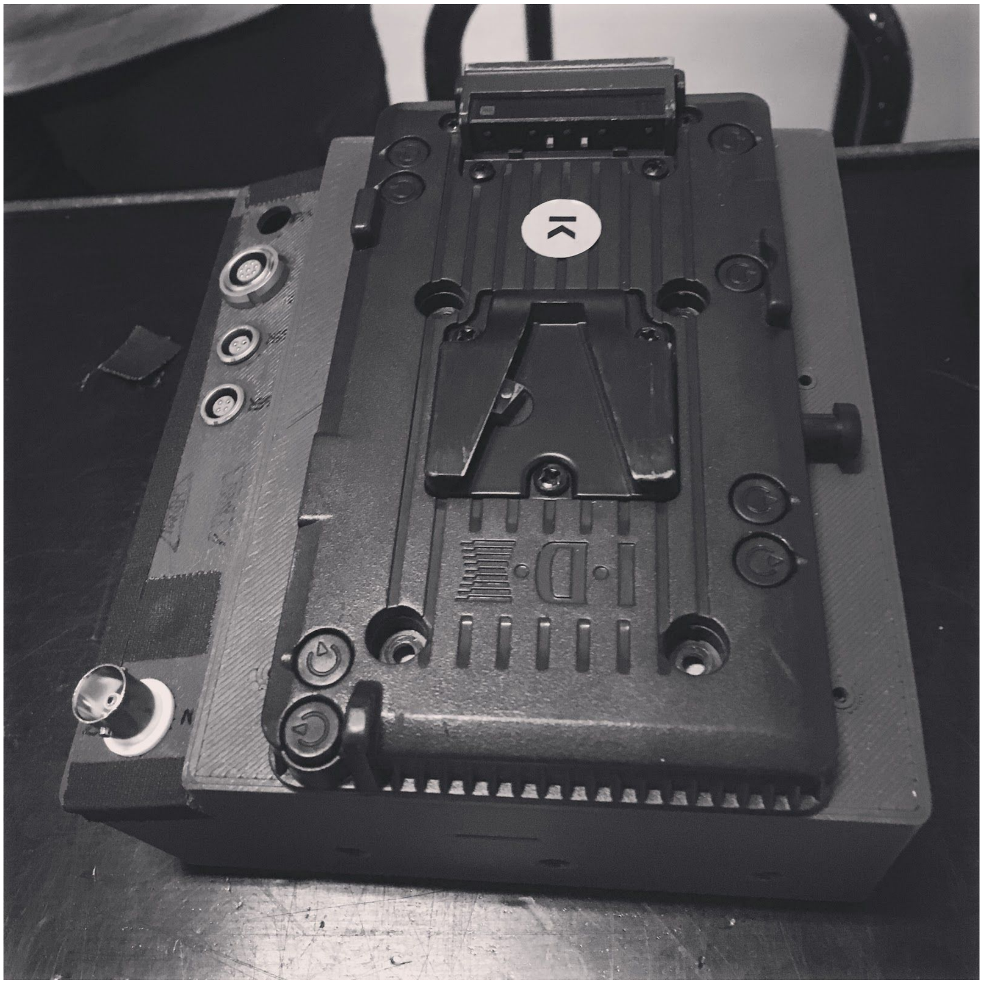


Figure 3: CamBox cabinet with battery mount

WP2 - Data Model and API

T2.1 - Model & API definition
(planned duration months 1-6)

- SRL/MIK worked towards an integration of dynamic metadata for the main camera into the EBUCore CCDM. 4 proposals for the integration of metadata were reviewed and discussed, and LADIO will report its considerations and conclusion to the relevant EBUCore bodies. Specifically, we intend to support 2 proposals, where one formulation is suited for a compact representation that is generated in offline form, whereas the other is a compact representation for an online, streaming form. The streaming form may not become highly relevant for the communication of the CamBox/SetBox with the SetBox in LADIO, because the bandwidth waste that is inherent in streamed XML-encoded metadata is remarkably large compared to a binary format (usually more than 1000%). Consequently, XML-violating and CPU-wasting application-specific compression is usually used in environments with low or unreliable network resources. Consequently, we chose de-jure and de-facto standard binary encodings for the on-set streaming, and considered ASN.1, XDR and Google Protocol Buffers. The latter was selected for LADIO because it is available as a standalone package, and has a wider support for modern platforms and languages.However, we hope to influence standardization, such that future recording devices that can make use of more advanced wireless networks may generated metadata according to this standard. To work for a feasible standard, we have initiated discussions with active contributors to EBUCore.

- SRL has started work on defining the proprietary protocol for metadata transmission from CamBox/MiniBox to SetBox. Data and metadata will be encoded using Google Protocol Buffers. The translation into standardized formats adherent to the EBUCore CCDM will be performed on the SetBox. The reason for this is low computing power on the CamBox/MiniBox and reduce bandwidth consumption for the wireless network on set.
- The CamBox/MiniBox provide a download API for data and metadata that is based on HTTP, a control API based using REST, and a push mode for data and metadata. Download and control will be implemented first as part of WP1.
- SRL studied the issue of time synchronization on set and decided to implement a basic common time base for all devices on set in the form of an NTP server (IETF RFC 5905). All instances of CamBox/MiniBox/SetBox retrieve their wallclock time reference from this NTP server (which is probably running on the SetBox) and compute their relative drift. All recorded data and metadata is annotated with a basic wallclock time and drift information relative to the data/metadata's own time base. The reason for this is that a MiniBox clock is independent of the recording device's clock, and a CamBox clock may be independent. The clock translation must be stored in EBUCore, unless we want to transcode data flows on the SetBox.
- SRL Studied the issue of device detection on set and decided to use MDNS (multicast DNS, IETF RFC 6762) to proactively announce devices names and addresses for instances of CamBox/MiniBox. To minimize bandwidth consumption, the translation of device to feature is based on configuration data that is loaded into the SetBox.

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

- SRL/MIK Work towards a representation for spatial and temporal dependencies. LADIO needs to keep track of spatial and temporal dependencies between recording data from several data sources (cameras, 360 cams, LIDAR, DSLR cameras, microphones, …), their metadata, content that is generated from these sources (3D reconstructions), and virtual elements that are added to the content. Beyond the state-of-the-art, LADIO will keep detailed (e.g. per-pixel) accuracy information for generated content, which must be stored along with the data. Exploration of the state-of-the-art in Scene description formats has not yielded an obvious basis for such a representation. Work is ongoing.

WP3 - Advanced 3D reconstruction

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

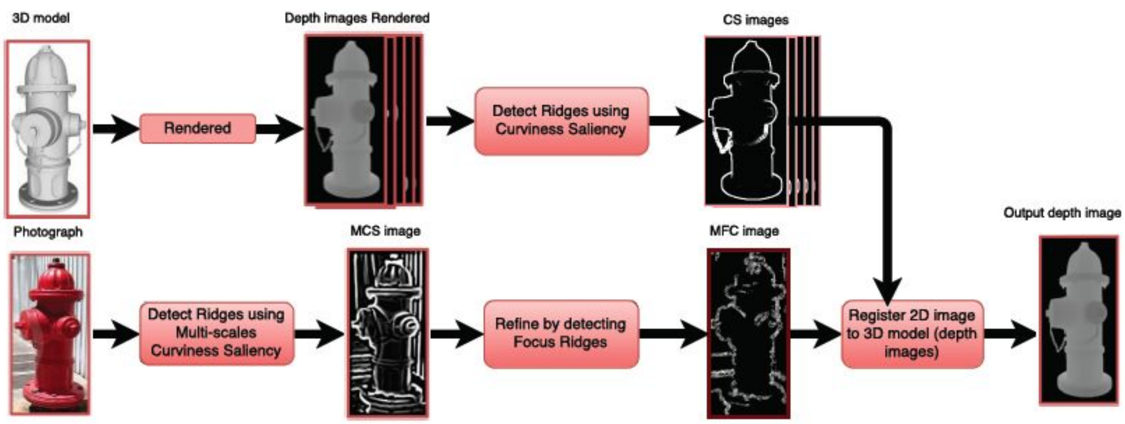
INP started this task in advance because of the availability of a relevant post-doc for this topic. So INP has started on T3.1 instead of T.3.3 to avoid wasting efforts on SfM improvements, which may have to be reconsidered based on results of the work started by CTU on T3.3.

- Within the LADIO project, this task registers optical images of a scene to existing (textureless) 3D dense range models of the scene captured by LiDAR scanning systems. Different approaches can be applied to solve such that problem either by directly registering a set of 2D optical images to a 3D geometric model or by reconstructing 3D point clouds of this stream of images using one of common Structure-from-Motion methods. In such case, 3D-3D registration can be solved.
- INP has explored some state-of-the-art works in 3D/3D geometric registration. By 3D/3D registration, we refer to the problem of finding the relative pose and scale of a "3D model" with respect to a "3D scene" (both either in mesh or point cloud representation) by only using 3D geometric information.
- SRL investigated some 3D-3D matching and aligning algorithms. The goal of this action is to develop a two-stage approach that consists of fast coarse alignment of two (potentially very different) mesh representations of a scene (our current approach follows the paper "*Matching 3D Models with Shape Distributions*"), and a slower iterative, exact alignment in a subsequent step that minimizes Hausdorff distances. The goal is develop for a situation where the observer is located inside the scene rather than the classical approach often found in the literature, where the observed object is completely visible from the outside.
- INP has experimented with three approaches, all with publicly available software or codes. The first one is *SurfaceMatching* (code at github.com/tolgabirdal/ppf_matching) as proposed in "*Model globally, match locally: Efficient and robust 3d object recognition*" (*B. Drost et al., CVPR 2010*). The *SurfaceMatching* pipeline consists in randomly extracting Pair Points Features (PPF) from depth images or generic point clouds (by uniformly sampling 3D meshes), indexing them and then real-time querying them efficiently by means of an adequate hashing. The "best" alignment is obtained from matched points using an Iterative Closest Point (ICP) algorithm. This method is able to detect and to align an object in a complete scene. In addition, it is simple and its implementation on OpenCV is efficient . However, it depends on selecting arbitrary reference points from the scene and assumes that it lies on the object needed to be detected. So, the alignment can be erroneous, especially when the object has repetitive surface structures, when the initial pose is incorrect or when the number of sample points of the surfaces is too low. Moreover, after sampling, if the surface variation is missing, and end up with dominantly sampling of a large plane, then theoretically you

have poor chances of registering them. The second one is a method based on the 4-Point Congruent Set (4PCS) that is considered as one of the most robust algorithms against noise or the presence of outliers. 4PCS combines a non-local descriptor that is simple and fast in order to detect four coplanar points in the point cloud and then to use a RANSAC scheme to choose matched descriptors. A fast variant of the 4PCS method, named *Super4PCS* (code at github.com/nmellado/Super4PCS) presented in "*Fast Global Pointcloud Registration via Smart Indexing*" (*N. Mellado et. al., EUROGRAPHICS 2014*) produces the same solutions as 4PCS, in linear time. However, it should be noted that the execution time and effectiveness of *Super4PCS* depends on a set of parameters for the data at hand: estimation of vertex noise, surface overlap, and sampling quality. These parameters constitute input for the method and directly affect the final matching time. In addition, We experimentally evaluated Super4PCS by aligning an object to a scene as *SurfaceMatching*, however the parameters of *Super4PCS* was difficult to adapted.

- INP has explored in-depth state-of-the-art works in 2D/3D registration. By 2D/3D registration, we refer to the problem of finding, given one or multiple views, the relative pose of the camera(s) with respect to a 3D (untextured) model (e.g., captured by LidAR) by using the geometric information of the 3D model and the 2D view photometric information. As in any 2D/3D registration task, the subproblems are those of: 1) feature detection in the 2D photography 2) feature detection in the 3D model and 3) 2D-3D matching to simultaneously align the features and estimate the 3D pose. Hatem Rashwan has experimented with publicly available software (code at meshlab.sourceforge.net) for the approach described in "*Fully automatic registration of image sets on approximate geometry*" (*M. Corsini et al., IJCV 2013*). It is a statistical-based method that is typically used for registration using Mutual Information (MI), which catches the non-linear correlations between the image and the geometric properties of the target surface. This approach registers the image by maximizing the mutual information of the surface normals with the corresponding gradient variations of the image. However, this approach is not fast, and requires a manual camera pose initialization to converge to the right solution, and the algorithm is difficult to parametrize.
- INP has tackled the 2D/3D registration problem of one object in a cluttered scene in the case when only *one view* is available. INP focused on two approaches where a 3D object model is described by a set of synthetic *depth images* taken from different viewpoints placed on a sphere around the object using sampled elevation/azimuth angles and distances between the camera and the object. This problem has at least three crucial issues in the photograph: how to separate the foreground (i.e., the object) from the background, how to be robust to lighting changes and how to get rid of textures (on the object). The first approach is based on the pipeline described in "*Registering Images to Untextured Geometry using Average Shading Gradients*" (*T. Plötz et al., ICCV 2015*). In a photography, the detected features are the gradient magnitudes. In the depth map, they are the "average shading gradient" (ASG) magnitudes over all lighting directions that can be computed, under simplified Lambertian shading hypotheses, independently of any lighting direction information. ASG is matched with the image

gradients to estimate the coarse pose, and subsequently refined using SIFT flow, and finally verified using RANSAC. INP has proposed an innovative approach based on the “Curviness Saliency” (CS) feature which is a scalar curvature index that measures the curvature at a local patch on a surface. In the (view-dependent) depth map, the CS features describe the “elongated” discontinuities of the 3D model, in particular the apparent (e.g., occulting) contours ; they are, by nature, robust to texture and light changes. Regarding the background elimination, the novel idea is to compute some infocus curves in the photograph describing the sharp parts of the imaged scene i.e., focused ridges, assuming that the background is blurred, with defocused points. INP has described how to extract these infocus curves from a blurring amount computed from the CS features. Regarding the texture elimination, INP proposed a multi-scale approach, similar to that in SIFT, assuming that a “curviness saliency” only exists at one scale.



“Multiscale Curviness Saliency and Infocus Curves” 2D-3D registration pipeline

- INP has implemented this 2D-3D registration approach based on “Multiscale Curviness Saliency and Infocus Curves” in Matlab (and will prepare a prototype for C++ implementation in OpenMVG). He evaluated the the two techniques of intensity image representation --i.e., the “Average Shading Gradient” (ASG) approach *vs.* the proposed one-- and showed that the proposed pipeline outperforms the ASG one.

2D-3D registration approach based on Multiscale Curviness Saliency and Infocus Curves: (1) Real 2D images (2) Views of 3D models in arbitrary poses (3) 2D-3D registration

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

- MIK has started a survey of available 360 cameras, with various parameters like price, number of cameras, coverage, image size, other sensors, recording support, video frame rate, sensitivity, …
  See 360 cameras Google sheet
- Some external contributors have started working on 360 cameras and RIG integration in openMVG.

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

- CTU released Structure-from-Motion pipeline YASFM-Yet Another SfM (github.com/fsrajer/yasfm), 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.

- CTU implemented the first uncertainty propagation for SfM working with thousand cameras and million of points in reasonable time. The work "*M. Polic and T. Pajdla.*

Uncertainty Computation in Large 3D Reconstruction" has been presented at CVWW 2017 (cvww2017.prip.tuwien.ac.at) and submitted to SCIA 2017 (scia2017.org).

- SRL explored feature matching options. Finding the two nearest neighbours in a 128-dimensional space is a large bottleneck in Structure-from-Motion pipeline and in camera localization because a huge number of distance computations between pairs of descriptors have to be performed. We have explored different approaches to speed up this process.
  We found that brute force matching of descriptors is extremely fast with cuBLAS when expressed as matrix multiplication. However, we need a different subset of the database for each input image, and since cuBLAS does not support sparse matrices, the overhead of moving data from CPU to GPU makes the approach useless.

- SRL also implemented AVX2 optimized CPU distance calculation. While not as fast as the cuBLAS solution, it allowed us to perform about 60 million L2-distance computations per second. This result will be integrated into OpenMVG to speed up the current Approximate Nearest Neighbor matching implementation based on FLANN (fast library for approximate nearest neighbors). We will also investigate the usage of this solution to speed-up the voctree implementation.
  SRL then made a CPU-based implementation of a randomized KD-tree based on the description by M. Muja and D.G. Lowe in "Scalable Nearest Neighbor Algorithms for High Dimensional Data". Our implementation is optimized for space as it uses indexes instead of pointers (a node uses only 8 bytes), and is therefore also suitable for use with CUDA. It can also be built efficiently on-demand: building 10 randomized KD-trees for 1M descriptors takes less than 20 seconds when using 8 threads.
  In discussion with MIK and INP, SRL learned that nearest neighbour searches in a large database return too many false positives and lose a lot of true positives as the distance ratio test is only valid with a limited number of descriptors (as in image matching). To avoid this, we need to filter the results returned by a KD-tree query such that the descriptors belong to the most similar images from the database. The next step is to design and implement such an algorithm.

- SRL explored options to increase SIFT extraction speed. SIFT feature points are the basis for several steps in LADIO, such as SfM and camera tracking. Since classical CPU-based implementations are rather time-consuming, LADIO makes use of the GPU SIFT library: PopSift. However, the speed of PopSift can be further improved by varying algorithms throughout the stages of the SIFT pipeline, and by making decision concerning the extraction of feature points. In the pipeline, we have explored (1) direct downscaling from the input image, (2) several Gaussian filter width including fixed-width filters, (3) the use of layered textures to extract descriptors for located features at the most appropriate level, (4) rotated extraction of descriptors to trade reduced cache hit ratio for reduced computation time. We have reached the understanding that different applications have a preference for many or few descriptors, a complete set or a uniform distribution of descriptors, and a preference for descriptors with a large or small scale factors. Consequently, making the appropriate choice before the actual extraction of the descriptors is the next step.

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

- J. Krcek, MSc. student supervised by T. Pajdla at CTU, has started extending of multibody structure from motion within YASFM framework (The work has started earlier since J. Krcek had to submit his thesis by the end of may 2017.). Existing approach, which is a part of YASFM, has been studied. A more detailed formal description of the algorithm has been written down and several experiments testing the limits of the existing algorithm proposed. This is a preliminary consolidation effort that is reviewing previous work done at the CTU in order to avoid a cold start on T3.4. This effort has not been counted to CTU PMs.

Figure 11: Experiment 4 result. Side view.

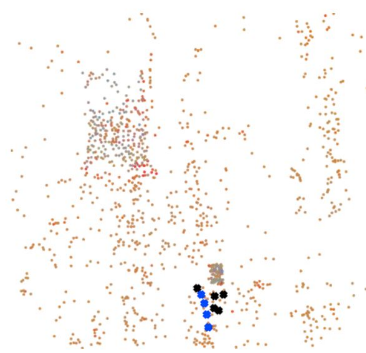


Figure 10: Experiment 4 result. Top view.

Two models (sarse background and dense foreground and the corresponding two sets of cameras - black and blue) recovered.

Avery preliminary reconstruction of a moving object (a small planar patch), which has been correctly segmented from the background (large cloud of points). Blue cameras are attached to the moving object. The black cameras are attached to the background. The scale of the moving object is much smaller than the scale of the background because it was reconstructed completely independently from the background.

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

- CTU investigated the existing approach to lens calibration (a.k.a. absolute camera pose computation in computer vision) and suggested its extension to robust lens calibration including unknown focal lens and camera radial distortion. He carried out first experiments in Matlab to prepare a prototype for following C++ implementation in

OpenMVG. O. Rybkin was working on a robust method for focal length extraction from fundamental matrix computed uncalibrated or partially calibrated settings.

WP4 - Detailed 3D representation

T4.1 MVS Baseline

(planned duration months 3-9)

MIK started this task earlier instead of T.3.3 to avoid wasting efforts on SfM improvements, which may have to be reconsidered based on results of the work started by CTU on T3.3. Meshing and Texturing is also an important part of the 3D reconstruction process to be able to evaluate the visual quality of the results. As a matter of fact, the first results of this task have already been integrated into production at MIK.

- CTU developed the CMPMVS code in a period of 5 years preceding LADIO solely on Windows. It was chosen for LADIO after a series of experience because it outperforms the other options in terms of visual quality of the resulting 3D reconstructions.
- CTU: T. Pajdla and C. Albl worked on re-engineering of the CMPMVS pipeline by explaining principles, existing software and supporting MIK and SRL teams in their re-engineering effort, in particular during the Prague workshops and T. Pajdla's visit to MIK in Feb 2017.
- MIK's first action was to rewrite the build system with cmake to enable multi platform compiling. Some of the code was relying on DirectX methods (Windows only library) for mesh simplification. We removed all related code as none of these parts was required for the standard meshing workflow. We have fixed problems due to case sensitivity. We have replaced platform specific methods and types by the equivalent ones in the c++ standard library. We have replaced platform specific code for file system management by the multi-platform implementation from the boost filesystem library. We also removed all unused and untested code.

  CUDA code was crashing on Linux due to a problem of memory alignment (called pitch in cuda library) that was not used correctly. Surprisingly the problem was not visible on windows (maybe also depends on the graphic card used). It was challenging as all access to cuda arrays were wrong. That has required a lot of code changes and a lot of debugging to get a stable multiplatform version.

  We have started to add doxygen documentation into the code, add comments and rename unclear variables and method names.

  We have also worked on the integration of CMPMVS steps into the workflow, especially on renderfarm. We created a simple command line containing only the steps required for the standard meshing pipeline without all the unused and untested functionalities. We also added an option to compute a subset of depth maps. This allows to compute depth maps in parallel on renderfarm, each computer is responsible to compute a small set of depth maps. The pipeline steps are: 1) export from openMVG to CMPMVS, 2) Compute depth maps, 3) Refine depth maps, 4) Filter depth maps, 5) Meshing, 5) Texturing. To

maximize parallelization on renderfarm by avoiding unnecessary synchronization wait between steps we have decided to fuse steps 3 and 4. We also added a step between 1 and 2 to precompute the camera connectivities once, before running depth maps computation in parallel.
We simplified the generation of intermediate files to limit the time spent in useless disk access and to limit the required storage.
We made some code optimization mainly on the texturing step by simplifying the code and, for instance, sharing the image cache across steps.
We discovered a large regression in term of quality introduced during the last year of CMPMVS developments at CTU. We analyzed one year of code history to found the 6 changes that have a dramatic impact on most of the datasets that are more complex than a small object with regular camera rotations.

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 is 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 initiated CMPMVS code exploration and performance analysis. We checked the code compatibility with the two major versions of GPU computation support library CUDA 7.5 and 8.0 and checked the code compatibility with the Ubuntu 16.04 x64 operating system. As result, we applied CUDA-related patches to the build system.

  Next we did a research on the methods of performance profiling for mixed applications containing both CPU- and GPU-intensive computations. The final conclusion made: there is no suitable out-of-box solution to do such kind of profiling for both simultaneously. As suitable workaround we decided to use a two-stage iterative code profiling. First, we do the CPU-performance profiling on the application and system function levels. Than we use the results to find the most compute-intensive functions of the application. If these functions do not contain GPU-related code we perform the CPU code optimization and repeat performance profiling from the beginning. In case, if the found functions contain GPU-related code we do performance profiling for all GPU-related code. Then use the results in conjunction with the CPU performance results to figure out what part of the function requires more time: CPU or GPU. Next we apply CPU or GPU code optimization and do the performance evaluation from the beginning. For all the performance evaluations we use the real datasets and run all the CMPMVS computation pipeline.

  As first step for the optimization of the most time consuming parts, we evaluated the performance of plane sweeping code. We did the optimization of the GPU code using

CUDA intrinsic and the CPU through deep cycles optimization. But further evaluation showed, that the performance improvement was platform-specific (only on MacBook Pro) and we rolled back the code changes. It means on most of the GPUs and CPUs (not mobile chipset versions) compiler together with CPU run-time optimizer doing the great job, and deep manual optimization can degrades the performance of the code.

Further performance evaluation showed the problem point in the getAtlasTexture (now called writeObjAtlasTexture) function. It was eating more than 50% of whole pipeline execution time due to intensive using of classes within cycles, intensive repeatable computations of the same values within cycles, and 2D arrays access strategy, that was making CPU caching useless. The most time critical parts have been rewritten. The performance increases by 2 times, run time decreased from 2 minutes to 55 seconds on the small buddha dataset.

Next we did the analysis of the CPU code and found that we have a lot of overhead in connection to intensively used primitive operations on matrices and points. They all implemented as objects and structures with number of corresponding member and friend functions. Many of that functions (e.g. multiplication, dot production, normalization, etc.) called inside the cycles with multiple levels of nesting. This causing a significant overhead due to huge number of calls (up to 6.5E9 times per small dataset, and much more for bigger ones). We found, that there is no way of unwrapping these structures - they object-oriented properties used widely. The best option for the moment is to make all often called methods inline. This will give us noticeable performance improvement without altering the main code. We made all the corresponding changes to source code (various matrix and point classes). The improvements were tested on the full buddha dataset. Overall computation time before this speedup was 59 minutes 2 seconds. After this changes - 53 minutes 47 seconds.

WP5 - LADIO Application

The developments of the LADIO Application was planned to start progressively in February 2017. With the withdrawal of LAB who was planned to be the main contributor, partners have reorganized the repartition between partners MIK and SRL. The LADIO Application will be split between a backend autonomous service and a user interface to visualize and interact with the data. SRL first starts by creating the main functionalities of the LADIO Application backend in adequacy with the data model developed in WP2.

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

- SRL has started the implementation of the backend API in subtask 5.1.1. This implementation is concerned with the interaction patterns, API to control

CamBoxes/MiniBoxes, and networking code. Implementation decisions are based on protocol decisions taken in Task 2.1.

WP6 - Management

- The Kick off meeting was held at CTU in Prague December 15-16th 2016, and both physical meetings and video conferences were planned according to the Description of Work.

Kick-off meeting in Prague, Dec 2016

- The Consortium agreement was signed by all partners on February 8th 2017 including LAB in the beginning of January.
- The pre-financing according to the original project plan was made to all partners except LAB on January 23rd 2017.

T6.1 - Platform for internal communication

- The technical mechanisms for LADIO were decided before or during than the Kickoff meeting in December. These support the consortium's approach to communication and cooperation, which requires continuous tracking of progress and frequent bilateral interaction. Tool support for tele-conferencing, asynchronous communication, document sharing and code co-development was selected and implemented by December 31th 2016.
- Two tools are considered "experimental". Of these, the #diary channel on asynchronous communication platform Slack has been well-received.

- The code repository for LADIO is hosted on GitHub in a combination of public and private repositories. The consortium decided to organize the repositories under a project-independent Git *organization* named AliceVision. This was done to demonstrate our ambition and intent to maintain LADIO developments beyond the duration of the project and with the involvement of third parties.
- The Deliverable D6.2 “Documentation of the deployed technical mechanisms” was submitted on December 31th 2016.

T6.3 - Platform for dissemination of public project information

- A dedicated website for the project has been developed and published on github http://ladioproject.eu

Workshops

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

- 2017-01-23 - 27: One week workshop with SRL, MIK and CTU to work on the analysis of CMPMVS and discuss about possible improvements in term of quality and performances.
- 2017-02-16: Tomas Pajdla (CTU) came at MIK to see datasets and discuss about the evolution of the algorithms.

Plenary meetings

Project meetings are held at least every 6 months. This is the meeting held in the reporting period in question:

- December 15-16: Kick-off meeting at Czech Technical University, Prague.

Plenary audio / video conferences

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

- January 11 - video conference
- February 8 - 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).

Status of deliverables

Deadlines for internal review and final submission of this quarters deliverables have all been met.

- Technical mechanisms for Management. Deliverable submitted on December 31. 2016.
- D6.2 - Quarterly management report #1. Completed QMR1 on February 28. 2017 and will be submitted as part of D6.2 in month 7.

Advisory board

LADIO starts with the outset of integrating industrial, open source and research developments that have been going on for several years. The first meeting with the advisory board is planned for month 7, when we expect to have prototypical integration of existing components ready, and present them in a consistent way to the board. This will provide them with a basis for comments about feasibility of and directions for LADIO.

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	*3,00*	5,17	*2,25*	0,06	*1,33*	0,42	*0,00*	0,00	*0,50*	0,00	*0,83*	0,68	***7,92***	**6,33**
MIK	Benoit Maujean Fabien Castan Yann Lanthony Nicolas Rondaud Jean Mélou Thomas Eskénazi	*0,00*	0,39	*2.00*	0,17	*2,67*	0,00	*0,71*	6,35	*0,50*	0,00	*0,5*	0,49	***6,38***	**7,40**
INP	Vincent Charvillat Sylvie Chambon Simone Gasparini Pierre Gurdjos Geraldine Morin Hatem Rashwan	*0,00*	0,00	*0,00*	0,00	*2,75*	3,50	*0,00*	0,00	*0,00*	0,00	*0,17*	0,17	***2,92***	**3,67**
CTU	Tomas Pajdla, Cenek Albl, Michal Polic	*0,00*	0,00	*0,00*	0,00	*2,58*	2.90	*0,57*	2.02	*0,00*	0,00	*0,17*	0,00	***3,32***	**4,92**
Total		***3,00***	**5,56**	***4,25***	**0,23**	***9,33***	**6,82**	***1,29***	**8,37**	***1,00***	**0,00**	***1,67***	**1,34**	***20,54***	**22,32**

Figure 1: The cumulative budgeted and contributed person-months since project start for each partner sorted by Work Package.