**Context-aware Multimedia Computer Vision applications**

**AMEBA Project Demonstrator**

**WORK PLAN**

**Frank Wickström, Natalia Díaz**

**Description**

A semantic framework for context-aware Smart Space applications applied to detection, recognition and learning of humans, object and human positioning. For this purpose, existing computer vision detection and recognition algorithms will be used, as well as location-aware technologies in order to semantically annotate humans and locations in real time, given a GPS, video or other multimedia data stream.

**Methodology** [Poveda-Villalón, NeOn Methodology for building ontology networks 2010]

NeOn Ontology Engineering methodology was chosen for improving aspects of previous ones (Methontology, On-To-Knowledge and Diligent). Among other benefits, it targets software developers and ontology practitioners, it has dynamic guidelines for ontology evolution, it treats context dimension and distributed collaboration. [Sure, Y., C. Tempich and D. Vrandecic]

1. Environment and feasibility study by the ontology developers, ontology engineers, software developers, domain experts and final users if possible. Decide whether ontologies should be developed or not for the specific problem.
   * For the ontology searching task, with the aim of reusing existing work, there are several tools that help us finding present ontologies:

|  |  |
| --- | --- |
|  |  |
| * + OMV: Ontology Metadata Vocabulary (<http://omv.ontoware.org/>)   + Oyster   + Swoogle, Semantic Web Search   + Watson.   + [Scarlet](http://scarlet.open.ac.uk/) - Semantic relations between concepts play an important role in several Semantic Web tasks, such as search, ontology matching, ontology enrichment. Scarlet discovers such relations by exploring the entire Semantic Web as a source of background knowledge. Relying on the functionality of Semantic Web gateways such as Watson or Swoogle, Scarlet automatically finds and combines knowledge provided by multiple online ontologies.   + Other Ontology Registries   When a candidate set of ontologies is found, a criteria for comparing them needs to be identified. |  |

1. Knowledge acquisition activities (during whole development). Requirements that the ontology should fulfill.
2. Ontology requirements specification. The main objective is to produce an ontology requirements specification document (ORSD) that includes the purpose, scope and implementation language of the ontology network, target group and the intended uses of the ontology network and its requirements to fulfill.
   * Ontology Evaluation. Quality Criteria:
     1. Accuracy
     2. Adaptability
     3. Clarity
     4. Completeness
     5. Computational efficiency
     6. Conciseness
     7. Consistency
     8. Organizational fitness
   * OntoClean (formal evaluation of taxonomic relationships, based on philosophical principles and hard to understand for experts)
   * AEON approach tries to (automatically) approximate the tagging with OntoClean
3. Scheduling. Search for knowledge resources and use it, together with the ORSD, to carry out the scheduling. Establish the ontology network life cycle and the human resources needed for the ontology project.
   * Develop required scenarios. E.g., reusing and reengineering non-ontological resources to merging ontological resources.
   * Conceptualization, formalization and implementation of the ontology.

**Functional Requirements (To be completed)**

* The technologies/tools used must be compatible with the W3C standards RDF and semantic query language SPARQL in order to support formal semantics, validation and data and device interoperability.
* The system can recognize faces and possibly objects as well. Objects that help to infer context are very useful, e.g. holding a coffee mug when leaving the Embedded System Lab. will determine that the user is not actually leaving work, but just having a coffee break in the kitchen of the same building.

**Non Functional Requirements:**

* Computer Vision techniques and algorithms should consider and compromise the degree of human invasiveness when recording and storing video data.
* Support for reasoning with a rule engine for context-aware inference. If any rules are exposed to the end-user, these must be in as much readable format as possible.

**Definition of Context Providers for Knowledge Acquisition**

* Location awareness:
  + Foursquare REST/PUSH API: Location-based social networking service for mobile devices.
  + Walkbase
  + Google Calendar Events (through Push alerts or mails containing location information)
  + GPS
  + Google Indoor Maps
  + Google Location API (uses IP and W-LAN location techniques)
  + Image’s inferred context. E.g., identifying people or familiar objects from the user common environment or using automatically generated maps by technologies such as Martin Werner’s indoor location or 13th Lab (<http://pointcloud.io/>). Inconvenient of the latter: available only for iOS so far.
* Visual awareness:
  + Smartphone or device camera (such as Looxie wearable mini-camera).
  + Static cameras, e.g., security cameras installed in the ICT-building.

**Use Cases**

* See Use Case Document and Sequence Diagrams.

**Functionality Modules available –Updates-**

* IT Department Staff Facial Database: An image crawler for a starting face database. A crawler for IT department personal was done, gathering all pictures and personal data.
* RDF Store. Sesame and Tomcat were installed, no support for connectivity with OO languages. Rdflib was considered as one of the few Python tools that are currently maintained, with support for JSON mapping for easy integration with common third party applications that use REST APIs.
* Viability studied for use of Foursquare (supporting REST and PUSH API –requires HTTPS-), to enable intelligent updates of the offices status traffic lights.
* The IT Department offices LCD status traffic lights are able to be controlled and updated with dynamic text.

**Issues**

* Using EMF independently of the Ontology or using EODM (EMF Ontology Definition Metamodel)? Also: [EMFText OWL2 Manchester Editor](http://www.emftext.org/index.php/EMFText_Concrete_Syntax_Zoo_OWL2_Manchester), Eclipse-based, open-source, Pellet integration.
* Should we restrict ourselves to smartphone cameras (even if low power and short battery life, low resolution)? Looxies mobile application is only available for Android and iOS.
* Applications Delimitation:
  + **Facial and object recognition**: Algorithms such as TLD, (and better OpenCV simplifications) have show enough feasibility. Also different AI implementations, highlighting:

**Face Recognition with Local Binary Patterns** Timo Ahonen, Abdenour Hadid, and Matti Pietik¨ainen, Machine Vision Group, Infotech Oulu PO Box 4500, FIN-90014 University of Oulu, Finland,*{*tahonen,hadid,mkp*}*@ee.oulu.fi, <http://www.ee.oulu.fi/mvg/>

* + **Activity recognition**: A potential future work, after basic image tracking and recognition is implemented in a lightweight form for smartphones. Activity recognition requires extra dept sensor (i.e. available in Kinect).
* Create an ontology that suits our specific needs or create an ontology network from existing ontologies? Reusing, e.g., Time W3C Ontology.
* Viability study: Execution of facial detection in the phone, sending detected faces to the system. The system realizes the Facial Recognition algorithm. Is it viable?

**To Do Next:**

-Formalize Ontology Requirements. Specify (possibly) using ORSD document. Includes Functional and non Functional requirements. See template for ORSD in *NeOn Methodology for Building Ontology*

*Networks: Ontology Specification (or more details in* paper *How to write and use the Ontology requirements Specification Document*, and <http://semanticweb.org/wiki/Ontology_Requirements_Specification_Document>).

-Motivate our ontology: We want to annotate what a camera sees, from the human point of view; thus, we need to know what sensations the images evoke in us, and for that, annotations must be modeled as features we perceive through our 5 senses and their relations among them. This type of annotation will allow us reasoning about what the camera sees, querying about happenings in our life (if we carry a wearable device camera), and for example, among others, querying our memories, helping us finding lost objects or retrieve memories for aiding Alzheimer.

-Create a hierarchy of priorities for determining current location coming from different context provider sources.

**Semantic language extensions for data-flow applications**

A multimedia data-flow semantic annotation model can be approached in several ways:

1. Taking Canals model as a starting point. The description of the Canals language's semantic extension can be done in two ways:
   1. Extend Canals metamodel with Java code. Since a metamodel must be done for our semantic annotation architecture, in e.g. EMF, the corresponding generated Java classes can be interfaced with Canals.
   2. Make Canals metamodel EMF-compatible (since Canals metamodel was designed ad-hoc, manually) and extend Canals EMF metamodel with our semantic annotation metamodel. Since Canals metamodel is much more complex and large in size, manual adaptations of the generated code might easily become a hassle.
2. Creating an independent semantic annotation metamodel. For this purpose, using UML or OWL modeling languages is a design question. However, a rule engine, production system (i.e. expert system) must be able to become integrated into the system to manage both data-flow and semantic annotation algorithms. Besides, a reasoner will validate the metamodel with, e.g., rules (business rules, etc.).

Taking a bottom-up approach, and focusing on context-aware Smart Space applications, we decide to take approach 2, creating first an independent metamodel in EMF for our semantic framework and providing later an interface with Canals as in option 1.b). This approach leaves flexibility to the semantic annotation metamodel, which could also be run with e.g., another data-flow or parallel language that adapts to certain domain applications, keeping independency in this case of Canals and the semantic annotation framework.

The idea is to start modeling a concrete application model for validation of the semantic metamodel, and then, a schedule could be written in Canals for parallelizing and scheduling a data intensive application.

**Some References**

*Computer Vision*

Moving object trajectory estimation for action recognition: <http://www.linkedin.com/groups/Need-2-segment-moving-obj-2642596.S.98382416?view=&srchtype=discussedNews&gid=2642596&item=98382416&type=member&trk=eml-anet_dig-b_pd-ttl-cn&ut=1ecuGRnvUoxB81>

Match moving. Existing software AfterEffect or Autodesk (former Realviz MatchMover). [http://en.wikipedia.org/wiki/Match\_moving](http://www.linkedin.com/redirect?url=http%3A%2F%2Fen%2Ewikipedia%2Eorg%2Fwiki%2FMatch_moving&urlhash=fcK_&_t=tracking_disc)

*Ontologies*

Semantic Image Annotation and Retrieval <http://www.svcl.ucsd.edu/projects/imgnote/>

**Appendix**

1. Image Annotation W3C <http://www.w3.org/2005/Incubator/mmsem/XGR-image-annotation/>

Different types of metadata

While various classifications of metadata have been described in the literature, every annotator should at least be aware of the difference between annotations describing properties of the image itself, and those describing the subject matter of the image, that is, the properties of the objects, persons or concepts depicted by the image. In the first category, (with typical annotations such as title, creator, resolution, image format, image size, copyright, year of publication, etc), common vocabularies such as [Dublin Core](http://www.w3.org/2005/Incubator/mmsem/XGR-image-annotation/#DublinCore) and [VRA Core](http://www.w3.org/2005/Incubator/mmsem/XGR-image-annotation/#VraCore) are used. The second category describes what is depicted by the image,

A combination of various multimedia related ontologies, including [FOAF](http://xmlns.com/foaf/0.1/) and the [MINDSWAP digital-media ontology](http://www.mindswap.org/2005/owl/digital-media). More specifically, image depictions can be asserted via a *depiction* property (a sub-property of foaf:depiction) defined in the MINDSWAP Digital Media ontology. Thus, images can be semantically linked to instances defined on the Web. Image regions can be defined via an *ImagePart* concept (also defined in the MINDSWAP Digital Media ontology). Additionally, regions can be given a bounding box by using a property named *svgOutline*, allowing localizing of image parts. Essentially SVG outlines (SVG XML literals) of the regions can be specified using this property. Using the [Dublin Core](http://dublincore.org/schemas/rdfs/) and the [EXIF Schema](http://www.w3.org/2003/12/exif/) more general annotations about the image can be asserted including its creator, size, etc. A subset of these sample annotations are shown in the [RDF graph below](http://www.w3.org/2005/Incubator/mmsem/XGR-image-annotation/#nasaRDFGraph).

1. Ontology Engineering Methodologies:
   1. UPON, (Nicole et al., 2005)
   2. HCONE (Kotis et al., 2004)
   3. OTK Methodology (Sure, 2003)
   4. OntoWeb project, (Leger et al., 2002)
   5. CommonKADS, (Schreiber et al., 1999)
   6. DOGMA, (Jarrar and Meersman , 2002)
   7. The Enterprise Ontology, (Uschod and King, 1995)
   8. The KACTUS, (Bernaras et al., 1996)
   9. METHODOLOGY, (Fernandez-Lopez et al., 1999)
   10. Etc.
2. Ontologies become Ontology Networks if there is a requirement or it is advisable to express meta relationship, for example
3. priorVersionOf
4. useImports
5. extendingBy
6. composedByModules
7. haveMapping

Ontologies permits a fluent knowledge sharing and an easy enrichment of the network.