SWAV: Semantics-based Workflows for Automatic Video Analysis*

Gayathri Nadarajan, Yun-Heh Chen-Burger, Robert B. Fisher

School of Informatics, University of Edinburgh, U.K.
gaya.n@ed.ac.uk,{jessicac,rbf}@inf.ed.ac.uk

Abstract. This paper outlines the SWAV system – **S**emantics-based **W**orkflows for **A**utomatic **V**ideo Analysis. SWAV utilises ontologies and planning as core technologies to gear the composition and execution of video processing workflows. It is tailored for users without image processing expertise who have specific goals (tasks) and restrictions on these goals but not the ability to choose appropriate video processing software to solve their goals. An evaluation on a set of ecological videos has indicated that SWAV: 1) is more time-efficient at solving video classification tasks than manual processing; 2) is more adaptable in response to changes in user requests (task restrictions and video descriptions) than modifying existing image processing programs; and 3) assists the user in selecting optimal solutions by providing recommended descriptions.

Keywords: semantics based workflows, ontologies, HTN planning, requirements based virtual workflow system, intelligent video processing.

1 Introduction

The field of video analysis is becoming more and more important with the increasing size of real-time data that need to be processed today. The pervasiveness of video data, *e.g.* satellite images, surveillance videos and environmental monitoring recordings, has triggered the need for more efficient means to analyse them than just traditional manual means. At present, analysing them is a tedious task as it requires either a large amount of manual processing time and/or highly specialised computational tools. The use of computational tools would speed up this process considerably, however, users almost always do not have access to such tools nor possess the technical expertise to implement or use them.

To provide a context, consider videos of underwater life available to marine biologists. Among the tasks conducted are video filtering, object detection and counting. The filtering involves removing videos that are unusable, *e.g.* those that are too dark or too bright as they are uninteresting for further analysis. The detection would include distinguishing objects of interest, such as fish, and further, these objects are counted for statistical purposes. Later on, they may also want to classify the fish according to their species type. Hence there is a range of tasks that the user is interested in. Manual analysis involves observing the video clip, pausing the video to take notes and repeating the process

* This research was funded by European Commission FP7 grant 257024, in the Fish4Knowledge project (www.fish4knowledge.eu).

until the task is complete. In order to assist users without image processing expertise to conduct **video and image processing (VIP)** tasks in an efficient manner, a suitable form of *automated* assistance should be provided. For this purpose a combination of computer vision methods, workflow and planning technologies and semantics-based approaches were investigated. First a hybrid workflow composition framework is outlined (Section 3), followed by the core components, the VIP ontology (Section 4), the VIP components (Section 5) and the planner (Section 6). The integrated system (SWAV) is then evaluated for efficiency, adaptability and learnability (Section 7).

2 Related Work

E-Science and Cloud workflow systems have emerged as forerunners in providing a specialised environment to simplify the programming effort required by scientists to orchestrate a computational science experiment. Therefore, Cloud-enabled systems must facilitate the composition of multiple resources, and provide mechanisms for creating and enacting these resources in a distributed manner. This requires means for *composing* and executing complex workflows, which has attracted considerable effort especially within the workflow community.

Major workflow systems include Pegasus, Triana, Taverna and Kepler [5]. Pegasus consists of a mapping and an execution engine. The mapping engine maps abstract workflows to its concrete (executable) form. The abstract workflows may be defined directly by application developers (workflow experts), semi-automatically or constructed with the assistance from a workflow editor. Triana is graphical workflow system that has been used for text, speech and image processing tasks. Workflows are created by drag-and-drop and sent for execution or saved. Workflow manipulation is handled manually by the user.

Taverna facilitates workflows for bioinformaticians who have a deep knowledge of the scientific functionality of the resources they want to link together, but limited expertise in programming and middleware technicalities. Workflow construction is placed in the hands of the user who is a domain expert. Kepler is a Java-based workflow system that can model complex computations. Users compose workflows using its graphical user interface.

Pegasus provides assistance to compose workflows using a system that can analyse, verify and correct partial workflows specified by the user. This work, in contrast, aims to automatically or interactively compose workflows from scratch. Triana, Taverna and Kepler provide good graphical interfaces, however, require the user to have domain expertise to compose and manipulate the workflows. To summarise, the limitations of existing workflow initiatives include 1) no provision of automated support in constructing workflows; 2) not tailored to react to changes in user goals and preferences; 3) unable to improve performance autonomously (or with user involvement) in an incremental manner according to specified goals; and 4) no mature integration with ontologies that would allow for more powerful representation and reasoning abilities. This work seeks to address some of these vital research gaps, by designing a framework that incorporates workflow technology with planning, ontologies and computer vision tools.

3 Three-layered Workflow Framework

SWAV was implemented based on a hybrid semantics-based workflow composition method within a three-layered framework (Fig. 1). It distinguishes three different levels of abstraction through the design, workflow and processing layers.

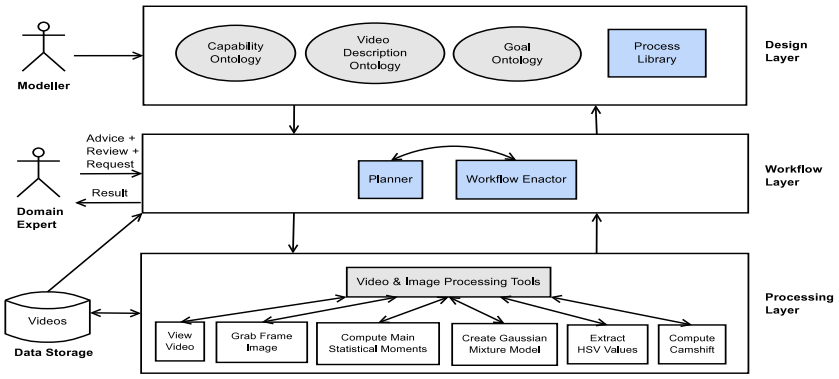


Fig. 1. Hybrid workflow composition framework for video processing with three abstraction levels. The core technologies are ontologies, planning and process modelling.

The design layer contains components that describe the VIP tasks, information about the video, image processing tools and processes to be carried out in the system. These are represented using a modular ontology and a process library. A modeller is able to manipulate the components of this layer, for example populate the process library and modify the ontologies. Typically the modeller is trained in conceptual modelling and has knowledge in the application domain, but not necessarily. Knowledge about VIP tools, user-defined goals and domain descriptions are organised qualitatively and defined declaratively in this layer using ontologies (Section 4), allowing for versatility, rich representation and semantic interpretation. The process library contains the code for the primitive VIP tasks and methods available to the system. These are known as the process models. A primitive task is one that can be directly performed by a VIP tool, while a method is decomposed into primitive and non primitive tasks.

The workflow layer is the main interface between the user and the system. It also acts as an intermediary between the design and processing layers. The workflow enactor ensures the smooth interaction between the components, access to and from various resources such as raw data, VIP toolset, and communication with the user. The main reasoning component is an execution-enhanced planner that is responsible for transforming the high level user requests into low level video processing solutions. More details will be provided in Section 6.

The processing layer consists of a set of VIP tools that can perform various image processing functions. Some examples can be seen in Fig. 1. The functions of these tools are represented in the capability ontology in the design layer. Once a tool has been selected by the planner, it is applied to the video directly. The

final result is passed back to the workflow layer for output and evaluation. The derivation methodology of the VIP components will be described in Section 5.

4 VIP Ontology

A pragmatic ontology was required to model the video and image processing (VIP) field so that it can be used for domain description and understanding, as well as inference. The ontology should describe the domain knowledge and support reasoning tasks, while being reasonably independent from the system. The principles adopted for the ontology construction included simplicity, conciseness and appropriate categorisation. For this reason, several aspects of the VIP field were highlighted. These were identified as *goal, video description* and *capability*. These aspects were motivated by the context of their use within a planning system that requires the goal and initial domain state model (which includes the initial video description) and also a performance-based selection of operators. The VIP ontology was modularised into three independent ontologies[1]. Each ontology holds a vocabulary of classes of things that it represents and the relationships between them. Among the possible domain knowledge representations, ontologies present a number of advantages, the most important being that they provide a formal framework for supporting explicit, machine-processable semantics definition, and they enable the derivation of implicit knowledge through automated inference. A system with full ontological integration has several advantages. It allows for i) cross-checking between ontologies; ii) addition of new concepts into the system; and iii) discovery of new knowledge within the system.

The **goal ontology** contains the high level video processing tasks (goals) and constraints that are communicated by the user to the system. It contains typical goals or classes of VIP tasks such as "Detection", "Classification", "Segmentation" and "Compression". Under each goal umbrella specific instances of goals can be found, such as "classify_fish_green_chromis" and "detect_presence_coral". Constraints are criteria that give additional restrictions to the goal. These include qualifiers to indicate user preferences such as speed of processing, CPU memory used, reliability of result, and accuracy of detection.

The **video description ontology** contains the concepts and relationships that describe the images and videos, such as the lighting conditions, colour information, position, orientation as well as spatial and temporal aspects. Hence, qualitative concepts such as "bright" (high luminosity) and "blur" (low clearness) could be used to describe the input video. The constraints and video description together constitute the domain description. Based on the goal and initial domain information provided by the user, the goal and video description ontologies are used to formulate the input to the planner.

The **capability ontology** contains the classes of VIP tasks and tools that can perform these tasks. Additionally, it organises them hierarchically, links the tasks to the tools and relates the tools with performance measures. Each task

[1] Visual and formal descriptions of the ontology can be found in `http://homepages.inf.ed.ac.uk/gnadaraj/phd/ontologies`.

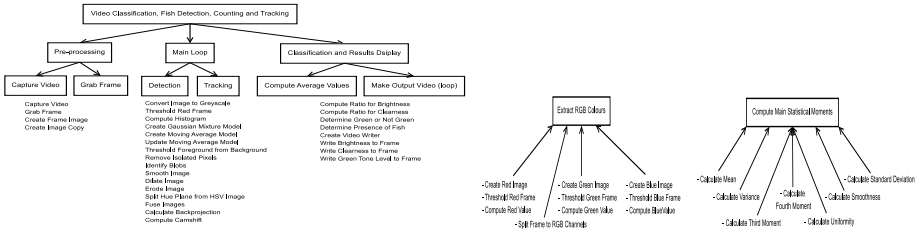
is associated with one or more tools (operators). A tool is a software component that can perform a VIP task independently given some input values, or a technique within an integrated vision library that may be invoked with given parameters. This ontology will be used directly by the planner in order to identify the tools that will be used to solve the task. The performance level of the tools are also tied to applicable criteria, namely these criteria refer to the domain information (video description and/or constraints). For instance, "Create Gaussian Background Model" is the best tool to perform a background model creation when the clearness level and the speed of movement are both high.

The main roles of the ontologies include guiding user and workflow for requirement retrieval, assisting image processing-naive users in decision-making by providing recommended descriptions and checking for consistencies.

5 VIP Components

The VIP components constitute the tools or operators of the workflows composed and executed. They are represented in the process library as primitive processes and the capability ontology as VIP tools. Typically, a VIP task is solved by writing a program that is compiled into an executable which can be run on an input video. However, having just one executable would only work on one task or a small subset of tasks. In order to construct such programs automatically, executables of a lower level of granularity would be required.

Generally image processing experts develop a single precompiled VIP tool, or executable that could work on one or a few similar videos for a VIP task. Often this single executable is modified manually before compiled and executed on a different type of video that requires different algorithms. Using the workflow approach designed for this work, such manual modifications are no longer required, as appropriate algorithms according to user and domain information could be selected automatically. Hence a multiple executable approach was devised for processing a range of VIP tasks using a selection of VIP tools. This multiple executable system is intended to provide the basis for a modular and reusable way to solve VIP tasks.



(a) Top-down approach to identify some VIP operators for video classification, fish detection, counting and tracking task.

(b) Bottom-up refinement to derive the executables 'Extract RGB Colours' and 'Compute Main Statistical Moments'.

Fig. 2. Top-down and bottom-up combination for deriving VIP components.

A combined top-down and bottom-up methodology was applied to derive a multiple executable system for video classification, fish detection, counting and tracking using single executable OpenCV programs designed for a variety of videos [4]. First, the program code was inspected thoroughly and tasks were broken down in a top-down manner (Fig. 2(a)). This involved breaking down the steps used in solving the task into meaningful blocks or components. Each function call and arithmetic operation was regarded as a primitive task. This exercise yielded 85 unique primitive processes in the process library that were encoded as operators in the capability ontology. When run on a one-minute clip containing 300 frames, 69,011 steps or operator invocations were produced. The bottom level tasks or operators were too fine grained and did not provide a manageable level to work with. They were also too technical for an image processing-naive user to comprehend and make decisions upon.

Subsequently, the bottom level tasks were grouped by procedure to provide a coarser level of granularity that was more manageable. This involved grouping the bottom level processes (primitive tasks) by procedure. For the most part, the primitive tasks were grouped to represent the subtask one level immediately above them. This exercise yielded 30 operators, termed as *independent executables*, that were much more manageable to work with. This methodology has been used effectively to accomplish the derivation of the VIP components.

The advantage of this bottom-up refinement approach has led to the identification of modules that could be reused for most video processing tasks. In addition, the executables provided a more intuitive representation of the video/image processing tasks than their primitive level counterparts. For instance, in Fig. 2(b), the independent executable "Compute Main Statistical Moments" which was derived by merging seven primitive tasks is a more compact and concise concept to represent a subtask to compute the mean, standard deviation and other statistical moments of an image. With this reduction of almost threefold in the number of operators from 85 to 30, a sample run on the same one-minute clip of 300 frames tested on the operators from the top-down approach now yielded 8706 execution steps, a reduction of almost eightfold in the number of steps [3].

6 Workflow Enactor and Planner

The workflow enactor plays the important role of orchestrating the flow of processing within the system (see Fig. 3). First it reads in the user request in textual form (use selects from a list of options). Next it consults the goal and video description ontologies to formulate the input that is then fed to the planner. When the planner, with the assistance of the process library and capability ontology, returns the final solution plan, the enactor prompts the user for further action. The user has access to the final result of the video processing task textually and visually (step 2), has the choice to rerun the same task on the same video but with modifications to the domain information (step 3), rate the quality of the result or perform another task. The composed workflow is saved in a script file that can be invoked easily off-line. By being able to view the result of each solution with changes to the domain information, the user can assess the quality

of the solution produced. This feedback mechanism could be used as a basis for improving the overall performance of the system as verifying the quality of the video processing solutions automatically is not a trivial task.

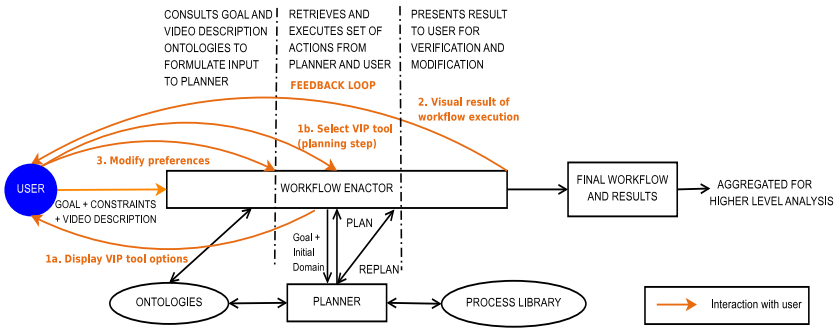


Fig. 3. Overview of interaction between the user, workflow and other system components when planning in semi-automatic (interactive) mode.

The planner acts as the "brain" of the system, which translates the high level user request into low level VIP steps. Adopting the principle that VIP tasks can be solved using a hierarchical decomposition approach, a Hierarchical Task Network (HTN) planner was implemented to realise this. This principle states that a task (goal) may be achieved by performing a set of primitive or non primitive subtasks, each non primitive subtask is further decomposed recursively until primitive tasks are reached. A primitive task could be performed directly by a VIP tool (operator). The role of the planning algorithm is to select the optimal set of VIP tools to achieve a given task. In HTN planning, the goal, initial state, and a set of methods are provided to the system. The methods encode the decomposition of known tasks. For example, video classification according to brightness, clearness and green tone levels may be achieved by first preprocessing the video, followed by computing the average values for the attributes to be classified. Computing the average values for the attributes involves computing the brightness, clearness and green tone levels in each frame image accumulatively. These best known practices adopted by image processing experts or *heuristics* are included as methods in the process library. In an HTN planner, the search space is reduced greatly because only the subtasks that are applicable to solve a current task are considered as nodes for further expansion. The set of options are reduced as planning progresses as only those options that match the preconditions for a subtask (either primitive processes or methods) are selected as valid choices. HTN planners are very efficient as a result of this.

The planner is able to plan in automatic and semi-automatic (interactive) modes. In the semi-automatic mode, it presents to the user all the available VIP tools that can perform a specific primitive task along with their recommended descriptions whenever more than one tool is available to solve the task. The user can make an informed decision based on these descriptions, making it an informative and interactive tool. In this fashion, the user is given some level of control during the planning phase.

7 Evaluation

30 videos originating from an ecological source via the Ecogrid project, Taiwan[1] were used for evaluating the overall approach. Interesting characteristics of marine life such as fish and coral can be extracted from the videos by performing analysis such as classification, detection, counting and tracking. The videos were taken in an uncontrolled open sea environment where the degree of luminosity and water flow may vary depending upon the weather and the time of the day. The water may also have varying degrees of clearness and cleanness. In addition, the lighting conditions change very slowly, the camera and the background are fixed and images are degraded by a blocking effect due to the compression. Taking into consideration factors such as diversity in user requirements, variety in the quality of the videos (*e.g.* lighting conditions, object movement) and vastness of the data made available, three hypotheses were formulated:

1. Automated support could be provided for users without image processing expertise to perform VIP tasks in a *time-efficient* manner using SWAV without loss of accuracy in the quality of the solutions produced.
2. Constructing VIP solutions using multiple VIP executables employed by SWAV is more *flexible* and *adaptable* towards changing users needs than modifying single executable programs.
3. The SWAV's mechanism to compose and execute workflows for VIP tasks helps the user *learn* the processes involved in constructing optimal solutions.

To test the first hypothesis (efficiency), the task completion time of performing video classification according to brightness, clearness and green tone levels using SWAV (automatic tool) was compared to the task completion time of the classification task conducted manually. Eight participants from a variety of backgrounds, none of whom possessed image processing expertise were selected as subjects, including an ecologist and a marine biologist.

Table 1. Time and accuracy of automatic (SWAV) versus manual processing, and their differences for video classification according to brightness, clearness and green tone levels using 30 Ecogrid videos.

Subject	Automatic (SWAV)		Manual		Difference	
	Time (s)	Accuracy (%)	Time (s)	Accuracy (%)	Time d_e	Accuracy d_a
1	2.12	61.11	47.90	76.19	-45.78	-15.08
2	2.13	61.11	39.65	53.33	-37.52	7.78
3	2.09	61.11	40.12	25.00	-38.03	36.11
4	2.06	61.11	45.33	87.50	-43.28	-26.39
5	2.13	61.11	35.02	52.38	-32.89	8.73
6	2.14	61.11	48.25	80.00	-46.11	-18.89
7	2.06	61.11	37.20	66.67	-35.14	-5.56
8	2.02	61.11	17.95	52.78	-15.93	8.33
Average	2.09	61.11	38.93	61.73	-36.83	-0.62

Statistical hypothesis testing using the t-distribution [2] was conducted to measure the dependencies between the results obtained for the times taken to conduct automatic and manual processing. For this sample set, the two sample dependent t-test was performed to determine the t value and its corresponding p value in order to accept or reject the null hypothesis. A significance level of $p < 0.05$ was taken as an acceptable condition to reject the null hypothesis. Using the values from Table 1, t was computed using Equation 1 below:

$$\bar{d_e} = -36.83 \qquad \sigma_{de} = 9.12 \qquad n = 8$$

$$t = \frac{\bar{d_e}}{\sqrt{\sigma_{de}^2/n}} = \frac{36.86}{\sqrt{9.12^2/8}} = -11.43 \tag{1}$$

where n is the sample size, $\bar{d_e}$ is the mean of the differences between the manual and automatic times and σ_{de} is the standard deviation of this mean. Based on the values of t and n, a significance level was computed. The degree of freedom was set to 7 $(n-1)$. A value of $t(7) = -11.43$ corresponds to a significance level of $p \ll 0.0001$. Therefore the efficiency of automatic processing using SWAV is significantly higher than the efficiency of manual processing. A similar statistical testing was conducted for accuracy, where there was no significant difference in the accuracies of the two methods. Hence the efficiency of automatic processing is significantly higher than manual processing without loss of accuracy.

The second hypothesis (adaptability) was tested using two subjects, an image processing expert and a workflow modeller, to make changes to the system available to them to perform fish detection and counting task on a video when domain descriptions (user preferences) change. Both have access to the same set of VIP tools; the former has an OpenCV program with available image processing algorithms written as functions and the latter in the form of multiple executables within a planning and ontology-enhanced workflow context, as defined in the SWAV tool. The time taken to make the appropriate modifications for six types of changes are contained in Table 2.

Table 2. Comparisons of number of new lines of code written, processing times and accuracies of solutions between single-executable image processing program and multiple-executable workflow system (SWAV) to adapt to changing domain descriptions.

Domain Descriptions (User Preference)	Image Processing Expert			Workflow Modeller		
	New Lines of Code	Time (min.)	Accuracy %	New Lines of Code	Time (min.)	Accuracy %
Prefer false alarm than miss	43	16	58.25	3	3	59.30
Prefer miss than false alarm	56	23	62.55	2	2	64.80
Clear, no background movement	43	16	58.46	3	3	60.71
Clear, background movement	61	27	60.42	2	2	60.10
Blur, no background movement	43	16	60.88	3	3	62.09
Blur, background movement	57	32	63.80	2	2	61.22
Average	50.50	21.67	60.73	2.50	2.50	61.37

Statistical hypothesis testing using the t-distribution was conducted to measure the dependencies between the results obtained for the times taken to make

changes to the OpenCV program and the SWAV tool. The significant level of $p \ll 0.05$ was obtained, proving that the workflow tool is faster to adapt to changes in domain descriptions than the image processing program.

The third hypothesis (user learnability) was conducted using the same eight subjects from the first experiment. Each was given 14 pairs of videos to work with, to perform fish detection and counting task using SWAV. Each pair was either similar or dissimilar; similar pairs have the same video descriptions associated with them (*e.g.* brightness, speed of movement) while dissimilar videos have different descriptions. Similar videos will require the same detection algorithm for the most optimal result while dissimilar pairs do not require the same detection algorithms for optimal result. The aim was to test if subjects were able to determine the most optimal tool for the detection algorithm based on the recommended descriptions provided by SWAV. If they were, then they should select the same tool as the most optimal one if the second video is similar. They should also not conclude to select the same tool as the most optimal one if the second video is dissimilar. At each run, the workflow tool will display the video to the subject before proceeding to solve the task that would enable them to recognise the video descriptions. When the user has to select the detection algorithm, the workflow tool provides a set of recommended descriptions for each exiting tool (via the semi-automatic planning mode). Using these recommendations and their knowledge of the video descriptions, the user should be able to make an informed decision. Each subject on average selected the correct optimal tool for the second video 5 out of 7 times within the similar videos, and only 2.25 times out of 7 times within the dissimilar videos. Statistical testing using the t-distribution has yielded a significance level of $p = 0.0004$, proving that the workflow tool has helped the user learn and manage the processes involved in selecting the optimal steps when solving a VIP task.

8 Conclusions

This paper has outlined SWAV, an efficient, adaptable and user sensitive workflow system for solving VIP tasks. Its novelties include (semi-)automatic workflow composition, new flexible way of solving VIP tasks and enabling naive users to learn optimal VIP solutions. Efforts to incorporate SWAV onto distributed infrastructures such as the Cloud for processing large-scale videos is underway.

References

1. Ecogrid: National Center for High Performance Computing, Taiwan (2006), `http://ecogrid.nchc.org.tw`
2. Howell, D.C.: Statistical Methods for Psychology. Belmont, CA, 6th edn. (2007)
3. Nadarajan, G., Chen-Burger, Y.H., Fisher, R.B.: A Knowledge-Based Planner for Processing Unconstrained Underwater Videos. In: STRUCK'09 (2009)
4. Spampinato, C., Chen-Burger, Y.H., Nadarajan, G., Fisher, R.B.: Detecting, Tracking and Counting Fish in Low Quality Unconstrained Underwater Videos. In: VISAPP'08. pp. 514–519 (2008)
5. Taylor, I., Deelman, E., Gannon, D., Shields, M.: Workflows for e-Science. Springer, New York (2007)