# Navigating through logic-based scene models for high-level scene interpretations

Bernd Neumann, Thomas Weiss

FB Informatik, Universität Hamburg, Voigt-Kölln-Str. 30
22527 Hamburg, Germany
{neumann, weiss}@informatik.uni-hamburg.de

**Abstract.** This paper explores high-level scene interpretation with logic-based conceptual models. The main interest is in aggregates which describe interesting co-occurrences of physical objects and their respective views in a scene. Interpretations consist of instantiations of aggregate concepts supported by evidence from a scene. It is shown that flexible interpretation strategies are possible which are important for cognitive vision, e.g. mixed bottom-up and top-down interpretation, exploitation of context, recognition of intentions, task-driven focussing. The knowledge representation language is designed to easily map into a Description Logics (DL), however, current DL systems do not (yet) offer services which match high-level vision interpretation requirements. A table-laying scene is used as a guiding example. The work is part of the EU-project CogVis.

## 1 Introduction

This contribution presents a framework for high-level scene interpretation based on logic-based conceptual models. Model-based scene interpretation in general is a well-known methodology, and various kinds of models - notably relational, frame-based, rule-based, neural and probabilistic models - have been investigated for their utility to capture and apply generic knowledge for Computer Vision systems. In this paper, logic-based models are explored because (i) high-level vision needs an interface to general knowledge and thus to AI-type knowledge representation, (ii) there exist powerful logic-based theories for qualitative spatial and temporal reasoning [1], [2] which may be useful for vision, (iii) little is known about the usefulness of logic-based models for scene interpretation [3], [4], [5], [6], [7] and (iv) even less is known about the use of logic-based models for the particular requirements of a "cognitive vision system" which is understood to exploit context, recognise intentions, apply task-driven focussing, and exploit past experiences.

The paper addresses high-level scene interpretations in the sense that the main interest is in interpretations above the level of single-object recognition. We consider indoor scenes, and a table-setting scene is used as a guiding example. Observed by stationary cameras, a human agent places covers onto a table. An interpretation summarising an evolving scene as "An agent is setting the table" is typical for high-level scene interpretation and exemplifies several characteristics:

- The interpretation describes the scene in qualitative terms, omitting details.

- The interpretation may include inferred facts, unobservable in the scene.
- The scene is composed of several occurrences
- Occurrences are spatially and temporally related.

One of the guiding ideas of this paper is to model constituents of a scene together with their perceptual correlates as "co-occurrence relations" which take the form of aggregates and parts in an object-oriented knowledge representation formalism. In our guiding example, placing a cover is such an aggregate. The approach is inspired by Barwise and Perry [8] who model coherent pieces of 3D scenes and their percepts by relations. The approach also allows to model intentions - mental states of agents in a scene - as parts of an aggregate.

The following section describes the structure of the knowledge base. In Section 3, interpretation strategies are presented for several different cognitive situations. Section 4 explores the usefulness of a DL system for representing the knowledge base and for providing interpretation services. Finally, Section 5 presents the conclusions.

## 2 Conceptual structure

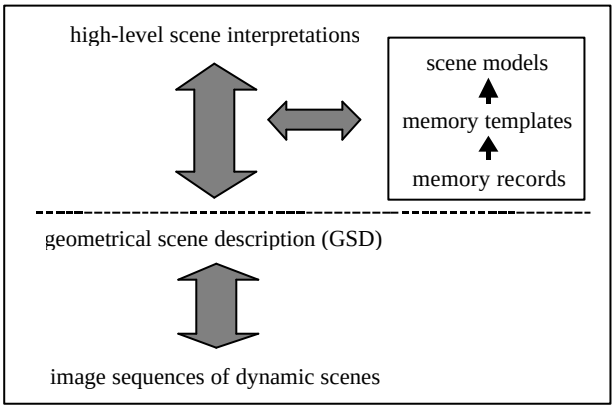


**Fig. 1.** Basic high-level vision architecture

The basic building blocks for high-level scene interpretation are shown in Figure 1. A dynamic scene is captured by several cameras and processed essentially bottom-up to the level of geometric scene descriptions (GSD) [9]. It is assumed that at this level the scene is described by (partial) views of objects ("blobs"). Furthermore it is assumed that moving blobs can be tracked and grouped into blob motions. We are well aware of the problem of providing a perfect GSD. It will be shown further down that high-level processes can cope with degraded information at the GSD level and even support lower-level processes.

Blobs and blob motions constitute the visual evidence which is used for high-level interpretations. The conceptual framework for interpretations is provided in terms of scene models which range from single object models to complex occurrence models. Scene models are linked to the records of a vision memory and are

considered the result of a learning process. However, this aspect will not be discussed in detail in this paper.

The main conceptual entities are aggregates. An aggregate consists of a set of parts tied together to form a concept and satisfying certain constraints. As an example, consider the conceptual model of a plate in a scene, where the physical plate and two views are combined as an aggregate. Figure 2 shows the concept in a frame-like notation:

| | |
|---|---|
| name: | scene-plate |
| parents: | :is-a scene-object |
| parts: | scpl-body :is-a plate with scpl-body-preds |
| | scpl-view-A :is-a sc-view-A |
| | with scpl-view-A-preds |
| | scpl-view-B :is-a sc-view-B |
| | with scpl-view-B-preds |
| constraints: | (scpl-view-constraints) |

**Fig. 2.** Conceptual model of a plate in a scene

The concept "scene-plate" is a specialisation of the concept "scene-object" and consists of three parts: "scpl-body" describes the physical body, "scpl-view-A" and "scpl-view-B" describe two plate views by camera A and B, respectively. The parts are specialisations of their respective parent concepts and fulfill certain predicates, e.g. shape predicates required for plate views. The constraints section contains constraints which relate parts to each other, e.g. ensuring that the views are compatible with a 3D shape of the physical object (which is, of course, not trivial). Note that the aggregate and its parts are embedded in several specialisation hierarchies: scene-objects, physical bodies, and views. The interpretation process will be guided by these hierarchies.

The next example, shown in Figure 3, specifies an occurrence model of the type "scene-place-cover". This is a crude conceptual description of a scene where a plate, a saucer and a cup are placed onto a table to form a cover. The scene-place-cover aggregate includes a table top, three transport occurrences and a cover configuration as parts. Furthermore, there are time marks which refer to the beginning and ending of the scene-place-cover occurrence. In the constraints section, there are identity constraints, such as pc-tp1.tp-ob = pc-cv.cv-pl, which relate constituents of different parts to each other (the plate of the transport suboccurrence is identical with the plate in the cover) and qualitative constraints on the time marks associated with sub-occurrences. For example, pc-tp3.tp-te = pc-tp2.tp-te denotes that the cup transport should end after the saucer transport. Aggregates involving mobile objects typically require that the objects fulfill certain temporal and spatial constraints. Hence temporal and spatial constraint solving will be an important part of the interpretation process.

The transport occurrences of the scene-place-cover aggregate are examples of conceptual entities embedded in a hierarchy of motion concepts. This hierachy is built on top of primitive occurrences which are generated as parts of the GSD. A primitive occurrence extends over a time interval where a qualitative predicate is fulfilled [10].

| | |
|---|---|
| name: | scene-place-cover |
| parents: | :is-a scene-agent-activity |
| parts: | pc-tt :is-a scene-table-top |
| | pc-tp1 :is-a scene-transport |
| | with (tp-obj :is-a scene-plate) |
| | pc-tp2:is-a scene-transport |
| | with (tp-obj :is-a scene-saucer) |
| | pc-tp3 :is-a scene-transport |
| | with (tp-obj :is-a scene-cup) |
| | pc-cv :is-a scene-cover |
| time marks: | pc-tb, pc-te :is-a timepoint |
| constraints: | pc-tp1.tp-ob = pc-cv.cv-pl |
| | ... |
| | pc-tp3.tp-te = pc-tp2.tp-te |
| | pc-tb = pc-tp3.tb |
| | pc-te = pc-cv.cv-tb |

**Fig. 3.** Conceptual model of a place-cover scene

As stated above, aggregates describe entities which tend to co-occur in a scene, regardless of whether the entities are visible or not. In fact, they provide the means to hypothesise parts without evidence. Hence it is natural to use aggregates which include mental states of agents, in particular intentions, as parts along with occurrences in a scene. The aggregate in Figure 4 is a sketch of an "intended place-cover", specifying an agent along with the place-cover occurrence and an intended cover configuration as the mental state of the agent.

| | |
|---|---|
| name: | scene-intended-place-cover |
| parents: | :is-a scene-intended-action |
| parts: | sipc-pc :is-a scene-place-cover |
| | sipc-ag :is-a scene-agent |
| | sipc-cv :is-a scene-cover |
| constraints: | sipc-ag.desire = sipc-cv |
| | (and other constraints) |

**Fig. 4.** Conceptual model for an intended action

As a summary of this section, Figure 5 gives on overview of the conceptual structure of the high-level vision system, restricted to static concepts for the sake of simplicity. The arrows denote is-a relationships. Dotted arrows indicate is-a relationships over several specialisation stages. Aggregates are shown as boxes with their parts as interior boxes. In general, parts represent concepts restricted by constraints specified by the enclosing aggregate (see the examples above). Hence parts are specialisations of the corresponding unconstrained concepts and linked accordingly.

As stated eralier, there are different hierarchies for physical objects and scene objects. The former are concepts independent of a scene, whereas the latter are described by aggregates containing physical objects together with the views

provided by sensors. Separate hierarchies are also provided for important descriptive entities such as 3D bodies and trajectories and their 2D counterparts, regions and 2D trajectories. Parts which constitute a region description, e.g. colour, shape, texture, are not shown explicitly. Concepts of these hierarchies provide indices for the interpretation process. As an example, "oval-region" is linked to "oval-view-B" which in turn is linked to the scene objects "scene-plate" and "scene-saucer" which may have an oval view (among others). Trajectories describe consecutive locations of a physical bodies, including constant locations for static bodies, and also form a separate hierarchy.

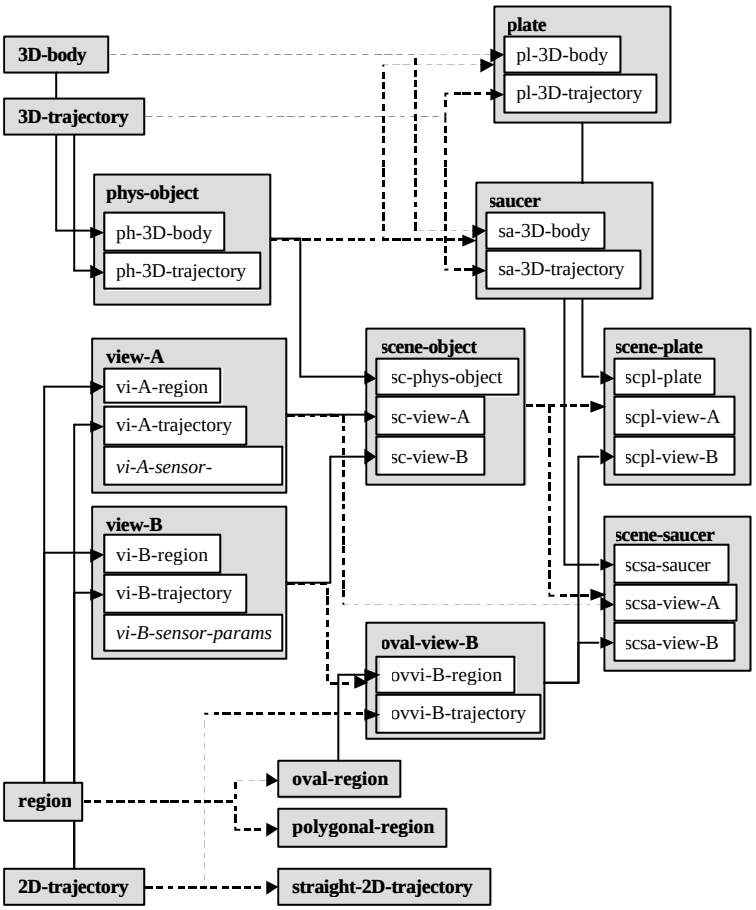


**Fig. 5.** Is-a hierarchies of concepts

Note that higher-level aggregates may be expanded until they contain only scene objects by resursively replacing aggregate parts by their conceptual descriptions. The expanded form of an aggregate includes all view entities which may support the aggregate based on visual evidence.

## 3 Model-based interpretation of a scene

This section describes how scene interpretation can be guided by the conceptual structure presented above. In particular we want to demonstrate that different cognitive situations can be treated with interpretation strategies based on the same conceptual basis. This is an important feature which distinguishes our approach from rule-based or deduction-based approaches where interpretation strategies are much narrower defined. The following cognitive situations will be considered:

- Context-free interpretation
- Exploiting spatial context
- Exploiting temporal context
- Exploiting domain context
- Exploiting focus of attention
- Intention-guided interpretation

Before dealing with these tasks, we present the framework of the incremental interpretation process.

### 3.1 Framework of interpretation process

An interpretation of a scene is a partial description in terms of instances of concepts of the knowledge base. It is partial because, in general, only parts of the scene and a subset of the concepts are interesting, depending on the cognitive situation. The interpretation process can be viewed as an incremental information gathering process with the goal to verify interesting instances. The increments are based on the internal structure of the concepts and the is-a structure in which they are embedded. Let I be an instance of a concept C, $PC_1$ ... $PC_N$ the parent concepts of C, $IP_1$ .. $IP_K$ instances of its parts, and $CE_1$ ... $CE_K$ the concept expressions associated with the parts. Then a verification of I with respect to C has the following logical structure:

$$\begin{aligned} Ver(I, C) = \quad & Ver(I, PC_1) \,\&\, ... \,\&\, Ver(I, PC_N) \,\& \\ & Ver(IP_1, CE_1) \,\&\, ... \,\&\, Ver(IP_K, CE_K) \,\& \\ & Ver(I, constraints(C)) \end{aligned}$$

Note that the verification of I w.r.t. C is recursively defined in terms of the verification of I w.r.t. the parents of C, and in terms of the verification of its parts. A recursion terminates successfully either at the root of a taxonomy (which by definition contains the instance) or when instances of the concept are already known and can be merged with the instance in question. The latter case includes the important step when an expected view instance is merged with one of the views generated from the GSD. As the GSD will not be perfect, parts may be occluded, and models may be too crude etc., it is mandatory that verification provides graded results. The operator "&" will combine graded partial results. Details of the grading scheme are outside the scope of this paper.

The actual interpretation procedure differs from the recursive structure in that (i) the execution order of the verification subtasks is subject to an independent control,

and (ii) constraints are partially evaluated and used to restrict the selection of missing parts. The control will be based on a probabilistic rating scheme currently under development. The interpretation procedure is composed of 3 types of interpretation steps. The first is *aggregate instantiation*. This step transforms an interpretation as shown in Figure 6.

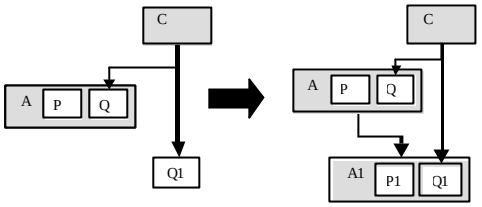


**Fig. 6.** Aggregate instantiation

In the figure, thin arrows denote is-a links, bold arrows instance links. Similarly, thin boxes denote concepts and bold boxes instances. Aggregate instantiation can be carried out when an instance Q1 of a concept C exists which may be part of an aggregate A. Instantiation of an aggregate causes the aggregate with all additional parts to be instantiated. This step corresponds to part-whole reasoning where a part gives rise to a hypothetical larger structure.

It may seem somewhat arbitrary that by this step aggregates are only instantiated if one part is already instantiated. However, this does not prevent aggregates to be instantiated as a whole via instance refinement (with no parts already instantiated) or as a part of a higher-level aggregate.

A second type of interpretation step is *instance refinement*. With this step one tries to find a more special concept for some instance. In general, instances are parts of some aggregate, hence instance refinement can be illustrated as in Figure 7.

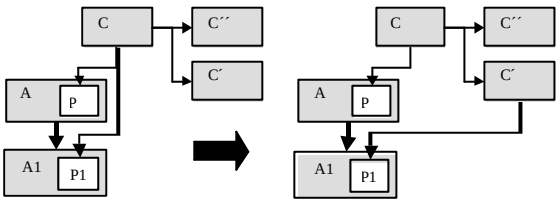


**Fig. 7.** Instance refinement

The figure shows an instance P1 which is reclassified from C to C´. As indicated with the alternative concept C´´, there may be many possible refinements. A control scheme will be required to avoid arbitrary guesses.

A third interpretation step is *instance merging*. As is evident from the aggregate instantiation step, new instances are generated as parts of a new aggregate irrespective of existing instances which could be used to build the aggregate bottom-up. Hence merging may be necessary. Roughly, two instances P1 and Q1 may be merged, if they have the same class and their expansions are grounded in the same

views. We indicate instance merging graphically by assigning a common name, see Figure 8.

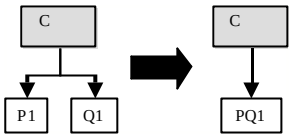


**Fig. 8.** Instance merging

### 3.2 Cognitive situations

We now address the cognitive situations listed earlier and describe how the tasks can be realised by the repertoire of interpretation steps.

By *context-free interpretation* we mean interpretation based initially solely on visual evidence and without other restricting information. Hence this situation essentially tests bottom-up interpretation facilities. We assume that blobs and blob motions are available from the GSD and automatically mapped into instances of views of the respective sensors, constituting the initial state of the interpretation.

The next step may be to carry out an aggregate instantiation step and create a "scene-object" with views as parts. Alternatively, a region could be specialised - say to an "oval-region" - by an instance refinement step. An "oval-region" is known to be a part of an "oval-view", hence another aggregate instantiation step can be carried out. The same result can be achieved by specialising the corresponding view with an instance refinement step. An "oval-view" may be part of several scene-objects, including a plate and a saucer. Aggregate instantiation steps will generate the corresponding instances. Again,there may be alternate paths leading to the same instantiations, for example successive instance refinement steps in the scene-object hierarchy. In summary, context-free interpretation is achieved by classifying low-level evidence via successive refinements and by instantiating scene-objects and aggregates based on the classified evidence.

As a second cognitive situation we consider exploitation of *spatial context*. In our framework, context is understood as an instantiated aggregate which specifies constraints between entities, including scene objects. Hence, if a spatial context is given, this is equivalent to an instantiated aggregate which specifies spatial constraints. As an example, consider a given context in terms of a kitchen bordering the living room scene. In our framework, this context will be modelled by an instantiated aggregate specifying the spatial relationship of the two rooms and including typical occurrences. Thus aggregates such as "bring-plate-from-kitchen" may become possible as the kitchen context supports the corresponding part-whole-reasoning. Note that within an aggregate, spatial constraints between parts may provide a dynamic spatial context provided by one part for another.

Exploitation of *temporal context* is very similar to the exploitation of spatial context. Temporal constraints in aggregates relate parts, e.g. occurrences, temporally to each other. Temporal properties of instantiated parts, e.g. begin and end, can be

propagated to restrict the temporal window of expected parts. An efficient temporal constraint mechanism for temporally related occurrences has been presented in [9].

By *domain context* we mean thematic knowledge restricting the possible contents of a scene, for example, knowledge that the scene will show a living-room or a dinner-table. Domain context is brought to bear by instantiating a corresponding aggregate. In this case, aggregates typically express co-occurrence relationships at a high abstraction level with potentially many alternative choices for parts.

We examine now how the interpretation process can be controlled by a *focus of attention*. One obvious way to express a thematic focus of attention is by instantiating concepts of interest and using the interpretation steps to elaborate these instances. This is similar to providing a domain context. To provide a temporal or spatial focus, a concept for temporally or spatially restricted interpretations may be instantiated.

Finally, we consider the role of recognised *intentions* for predicting the development of a scene. As shown in Section 2, the intention of an agent is modelled as a mental state which may be ascribed to an agent, given certain occurrences. Our interpretation process can instantiate intentions by part-whole reasoning (aggregate instantiation) and thus provide information about the goal state intended by the agent.

It is outside the scope of this paper to deal with other aspects of the interpretation process, in particular uncertainty management and preference ranking of hypotheses. We are developing a ranking system based on the statistics of recorded experiences which guides the possible choices for interpretation steps.

## 4 Translating into a Description Logic (DL)

In this section we sketch how the frame-like modelling formalism introduced above and the interpretation steps can be translated into the formal language of the highly expressive description logic *SHIQ* [11] implemented by the system RACER [12]. The purpose is to investigate to which extent inference mechanisms available in DL systems may be used to support the interpretation process. *SHIQ* is the basic logic *ALC* augmented with qualifying number restrictions, role hierarchies, inverse roles, and transitive roles. In addition to these basic features, RACER also provides concrete domains for dealing with min/max restrictions over the integers and linear polynomial (in-)equalities over the reals.

The aggregate structure as shown in Figures 2 - 4 maps into the RACER concept language roughly as follows.

? the name of an aggregate is a RACER concept name
? the parents of an aggregate are concept names defining unary predicates
? part names are roles defining binary predicates
? with-expressions are role qualifications
? constraints map into concrete domain predicates

Currently, only inequality constraints can be handled by RACER´s concrete-domain facilities. This is sufficient, for example, to implement a qualitative temporal constraint system for a time point algebra. Other constraint schemes, e.g. for qualitative spatial contraints, would require extensions.

Assuming that these extensions can be provided, we will examine the interpretation process now. It should be clear that automatic instance classification cannot be employed since concrete views do not provide logically sufficient conditions for hgher-level classification. As shown by [3] and further elaborated in [5], image interpretation can be formally described as partial model construction ("model" in the logical sense). In fact, RACER offers an ABox consistency check which amounts to model construction. Given an ABox with concrete views as individuals, model construction generates an interpretation including all additional individuals which are required to satisfy the conceptual framework.

Unfortunately, model construction in RACER (and other reasoning systems) is conceived as an open-world consistency check where any model means success and additional individuals are hypothesised liberally without consideration of missing visual evidence. Hence this process cannot be employed without severe changes. For example, partial evidence for a cover in terms of a plate should only be extended to a full cover if the possible views of missing objects are compatible with the actual scene. Furthermore, as in general many models are possible, a ranking is required so that "preferred interpretations" can be delivered.

RACER can be used, however, in support of one of the more modest interpretation steps outlined in Section 3: Instance refinement is available in RACER as individual classification. Also general services such as a TBox consistency checking may be used.

## 5 Conclusions

A conceptual framework for high-level vision has been presented using a representational formalism which easily maps into an expressive description logic. The main conceptual units are aggregates which represent co-occurring physical bodies and their percepts. Guided by the need to deal with various cognitive situations, interpretation steps have been proposed which support flexible interpretation strategies. As it turns out, current DL reasoning systems do not (yet) provide the services which would optimally support high-level vision. In particular, a ranking scheme should guide possible choices.

## References


1. Cohn, A.G., Hazarika, S.M.: Qualitative Spatial Representation and Reasoning: An Overview, Fundamenta Informaticae,46 (1-2) (2001) 1-29
2. Vila, L.: A Survey on Temporal Reasoning in Artificial Intelligence", AI Communications, Vol. 7, (1994) 4-28
3. Reiter, R., Mackworth, A.: The Logic of Depiction, TR 87-23, Dept. Computer Science, Univ. of British Columbia, Vancouver, Canada (1987)
4. Matsuyama, T., Hwang, V.S.: SIGMA - A Knowledge-Based Aerial Image Understanding System, Advances in Computer Vision and Machine Intelligence, Plenum (1990)
5. Schröder, C.: Bildinterpretation durch Modellkonstruktion: Eine Theorie zur rechnergestützten Analyse von Bildern, Dissertation, DISKI 196, infix (1999)

6. Nagel, H.-H.: From Video to Language - a Detour via Logic vs. Jumping to Conclusions, Proc. Integration of Speech and Image Understanding, IEEE Computer Society (1999) 79-99
7. Möller, R., Neumann, B., Wessel, M.: Towards Computer Vision with Description Logics: Some Recent Progress, Proc. Integration of Speech and Image Understanding, IEEE Computer Society (1999) 101-116
8. Barwise, J., Perry, J.: Situations and Attitudes, Bradford (1983)
9. Neumann, B.: Description of Time-Varying Scenes, Semantic Structures, Lawrence Erlbaum (1989) 167-206
10. Neumann, B.: Conceptual Framework for High-Level Vision, FBI-HH-B-241/02, FB Informatik, Universität Hamburg (2002)
11. Horrocks, I., Sattler, U., Tobies, S.: Reasoning with Individuals for the Description Logic SHIQ, Proc. 17th Int. Conf. on Automated Deduction (CADE-17). LNCS Springer (2002)
12. Haarslev, V., Möller, R.: RACER User´s Guide and Reference Manual Version 1.7, http://www.fh-wedel.de/?mo/3214/racer-manual-1-7.pdf