iRep3D: Efficient Semantic 3D Scene Retrieval

Xiaoqi Cao[1] and Matthias Klusch[1]
[1]*German Research Center for Artificial Intelligence, Saarbrücken, Germany*
{*xiaoqi.cao, klusch*}*@dfki.de*

Keywords: Semantic 3D scene retrieval, semantic indexing

Abstract: In this paper, we present a new repository, called iRep3D, for efficient retrieval of semantically annotated 3D scenes in XML3D, X3D or COLLADA. The semantics of a 3D scene can be described by means of its annotations with concepts and services which are defined in appropriate OWL2 ontologies. The iRep3D repository indexes annotated scenes with respect to these annotations and geometric features in three different scene indices. For concept and service-based scene indexing iRep3D utilizes a new approximated logical subsumption-based measure while the geometric feature-based indexing adheres to the standard specifications of XML-based 3D scene graph models. Each query for 3D scenes is processed by iRep3D in these indices in parallel and answered with the top-k relevant scenes of the final aggregation of the resulting rank lists. Results of experimental performance evaluation over a preliminary test collection of more than 600 X3D and XML3D scenes shows that iRep3D can significantly outperform both semantic-driven multimedia retrieval systems FB3D and RIR, as well as the non-semantic-based 3D model repository ADL in terms of precision and with reasonable response time in average.

1 INTRODUCTION

For the success of 3D Web applications in many domains like virtual 3D product engineering both a highly precise and reasonably fast retrieval of relevant 3D scenes modelled in X3D[1], XML3D[2] or COLLADA[3] is of paramount importance. Research on efficient retrieval of semantically annotated 3D scenes gained momentum in the past years. For example, the ISReal platform for intelligent and web-based 3D simulation of realities (Kapahnke et al., 2010) allows users to annotate XML3D scene objects with descriptions of their conceptual meaning and functional behavior with formal concepts, services and hybrid automata, and leverages these hybrid semantic annotations for simulations of virtual 3D worlds with intelligent avatars. Current scene retrieval systems like FB3D (Camossi et al., 2007), RIR (Alvez and Vecchietti, 2011) and the open-source 3D repository ADL[4] leverage in particular advanced methods of matching textual descriptions, geometric features and RDF[5]-based semantic annotations of 3D scenes.

[1]`http://www.web3d.org/x3d/`
[2]`http://www.xml3d.org/`
[3]`https://collada.org/`
[4]`http://3dr.adlnet.gov/Default.aspx`
[5]`http://www.w3.org/RDF/`

However, syntactic-based 3D scene retrieval approaches (Gao et al., 2011; Gong et al., 2011; Leifman et al., 2005; Hou et al., ; Koutsoudis et al., 2011; Qi et al., 2011) offer fairly fast response times in average but almost always suffer from a relatively low average precision due to syntactic mismatches. Alternatively, current RDF(Laborie et al., 2009; Alvez and Vecchietti, 2011) and strict logic-based approaches of 3D scene retrieval(Kalogerakis et al., 2006; Hois et al., 2007a; Hois et al., 2007b; Pittarello and De Faveri, 2006; Yang, 2010) were shown to be capable of alleviating this problem to some extent but at the cost of higher response times and without considering 3D scenes geometric features.

In iRep3D, an annotated 3D scene in XML3D, X3D or COLLADA is indexed not only with respect to its geometric features but referenced concepts and services which formally describe the conceptual and functional semantics of the scene in standard OWL2. The semantic indexing of scenes utilizes, in particular, a new approximated concept similarity measure based on weighted logical abduction, while B+ tree-based scene indices are built for geometric features of scenes. A query for top-k relevant 3D scenes is processed by iRep3D in its three scene indices for concepts, services, and geometric features in parallel. The resulting scene relevance rank lists are then ag-

gregated with Fagin's threshold algorithm (TA) (Fagin, 2002) before the final answer set to the query is returned to the user.
The remainder of the paper is structured as follows: Semantic annotations of 3D scenes and corresponding scene indices are described in section 2 while the hybrid semantic retrieval by iRep3D is explained in section 3. Results of experimental performance evaluation and related work are presented in sections 4 and 5. We conclude the paper in section 6.

2 SEMANTIC SCENE ANNOTATION AND INDICES

The annotation of 3D scene graphs in X3D, XML3D, or COLLADA with concepts, services, and geometric features can be embedded in the respective X-HTML files with standard RDFa. Inspired by (Kapahnke et al., 2010), iRep3D leverages such machine-understandable descriptions of conceptual, functional or behavior-based, and geometric feature-based scene semantics for a more informed retrieval of relevant 3D scenes. In the following, we introduce the different kinds of annotations and corresponding scene indices which are created by iRep3D off line for a given collection of annotated 3D scenes.

2.1 3D Scene Annotation and Query

A simple example of an annotated X3D scene named `Toledo_Car_001` of a special car model is shown in figure 1. The scene annotation includes a scene concept "Toledo", a semantic service "transport", and the color of the car as one of its geometric features. In addition, a free-text description of this 3D scene model is given in its meta-tag. As stated above, the semantic annotations can be embedded with standard RDFa in any XML-based 3D scene description.

```
<html><head>...</head><body>
<X3D id="Toledo_Car_001" ...>...
  <meta content="Toledo Car 001 visual exhibition"
    name="description">...
<Scene> ... <Shape> ... </Shape>..
</Scene>
</X3D>
</body></html>

Scene Concept:
  τ (Toledo) := Vehicle^P ⊓ Car^P ⊓
    =4 hasWeels.Wheel^P ⊓ ∀canCarry.Passenger^P
    ⊓ ¬ ∀ownership.Private^P
Semantic Services: {
  transport := [
     In (Passenger psg, Location lc );
     Out ();
     Prec: ( availableFor ( psg ) ∧ reachable ( lc ) );
     Eff: at ( psg, lc );] }
```

Figure 1: Annotated X3D model *Toledo_Car_001*.

Annotation with scene concepts. For example, the scene concept "Toledo" describes the overall semantics of the 3D scene of the respective car model. In the figure the concept is shown together with its logical expression which is derived by iRep3D from the formal definition of this concept in a referenced ontology in standard OWL2. Such concept expressions contain only logical operators (conjunction $\sqcap$, negation $\neg$) and quantifiers (universal $\forall$, exists $\exists$) over a set of primitive concepts or terms ($\cdot^P$). For example, the logical expression of scene concept "Toledo"' contains the primitive concepts $Vehicle^P$ and Car^P as well as quantified and cardinality restricted (primitive) roles in the clauses $\forall canCarry.Passenger^P$, $\neg\forall onwership.Private^P$ and $= 4hasWheels.Wheel^P$. For the sake of simplicity, in the following we assume ontologies O and O_S in OWL2-DL which are used by 3D scene designers or third-party users for semantic annotation of 3D scenes stored in a given iRep3D repository with concepts and services, and an ontology O_{req} for scene requests. The set of primitive terms is the shared basic minimal vocabulary of these ontologies out of which more complex concepts can be individually defined.

Definition 1: *Annotated 3D scene*
Let X the set of 3D scenes stored in an iRep3D repository r, O the 3D scene concept ontology. A 3D scene $x \in X$ is defined by the tuple: $x = [id, sd, \tau(C,O), SS, GF, da]$ where id denotes the UUID of x; sd the (syntactic, textual) description of the meaning of x; $\tau(C,O)$ the logical unfolding of the scene concept C of x in the scene concept ontology O in OWL2-DL; SS the set of semantic services in OWL-S provided by x (cf. Def. 2); GF the set of geometric features of x (cf. Def. 3); and da the data of scene x including the XML-based description file and its referenced resources like images, animations, sounds. ■

Annotation with semantic services. The functionality of 3D scene objects like the transport of passengers and goods by a car, or the opening or closing of its doors can be described in terms of appropriate services which semantics are formally defined in OWL-S[6]. A semantic service profile (IOPE) describes the semantics of service signature (I/O) parameter-

[6] http://www.w3.org/Submission/OWL-S/

s in terms of an appropriate conjunctive list of (I/O) concepts defined in OWL2-DL. In addition, the precondition (P) and effect (E) of the service execution is described in terms of logical expressions in standard PDDL. Each semantic service of a 3D scene is grounded in an executable service program such as a 3D animation script (Kapahnke et al., 2010).

For example, the functionality of the car model `Toledo_Car_001` in figure 1 is partly described in the profile of the semantic service "transport" with the concepts *Passenger*, *Location* of the service (program) input variables pg, lc which semantics are defined in a referenced OWL2 ontology. In addition, the service precondition requires that the car should be available for the *Passenger* and the *Location* should be reachable, while the effect $at(psg, lc)$ of executing this transport service means that the *Passenger* eventually will be at the given *Location*. There is a variety of tools for efficient selection of semantic services for a given service request available (Klusch, 2012) like the currently most precise service matchmaker iSeM (Klusch and Kapahnke, 2012).

Definition 2: *Semantic services of a 3D scene*
A semantic service $ss \in x.SS$ of an annotated 3D scene $x \in \mathcal{X}$ is defined by the tuple: $ss = [URI, In, Out, Prec, Eff]$ where URI denotes the URI of the service description file of ss in OWL-S; In (Out) the set of input (output) parameter concepts of ss in OWL2-DL; $Prec$ (Eff) the logical expression of the precondition (effect) of ss in PDDL or SWRL. The concepts in In, Out, $Prec$ and Eff are defined in a service parameter ontology O_{sp}. For sake of simplicity, without loss of generality, we assume one O_{sp} for all semantic services of 3D scenes $x \in \mathcal{X}$ stored in the considered iRep3D repository. Denote A_s the set of predicates that are used in the services of any 3D scene $x \in \mathcal{X}$. Let $ss.In$, $ss.Out$, $ss.Prec$ and $ss.Eff$ denote $ss[i]$, $ss[o]$, $ss[p]$ and $ss[e]$, respectively, and $\mathcal{SS} = \bigcup x \in x.SS$ the set of all semantic services of annotated 3D scenes $x \in \mathcal{X}$. ■

Geometric features of scenes. Any geometric feature gf of a given scene x is an instance of some feature type f which is defined in the specification of X3D, XML3D or COLLADA.

Definition 3: *Geometric features of a 3D scene*
Let $\mathcal{F}$ denote the space of all types of geometric features of 3D scenes in X3D, XML3D and COLLADA. A geometric feature $gf \in \mathcal{GF}$ of a 3D scene $x \in \mathcal{X}$ is defined by the tuple: $gf = [name, f, \{(k, v)\}]$ where *name* denotes the feature name of gf in the context of x; $f \in \mathcal{F}$ the feature type of gf; and $\{(k, v)\}$ the set of attribute-value pairs which assigns values $v(k)$ to each attribute k of the geometric feature type f with a proper data structure according to the X3D, XML3D or COLLADA specifications. Let K_f the set of attributes of feature type f, and $\mathcal{GF}$ the set of geometric features of all scenes $x \in \mathcal{X}$ stored in the considered repository; $x.v(f.k)$ denotes the value of attribute k of feature f in scene x. ■

Semantic query for 3D scenes. The repository allows users to issue semantic queries for relevant 3D scenes, in particular, by means of specifying the desired conceptual, functional and geometric features.

Definition 4: *Semantic 3D scene query*
Let O_{req} denote an ontology used by a requester *req* to formulate a request q for relevant 3D scenes. Such a query q for 3D scenes is defined by the tuple: $q = [sd, \tau(C, O_{req}), SS, GF, A]$ where sd denotes the syntactic (textual) description of the desired scene; $\tau(C, O_{req})$ the logical unfolding of requested scene concept C in O_{req}; SS the set of semantic services that the desired scene should provide; GF the set of geometric feature instances that the desired scene should have; and A the total number of the most relevant scenes requested by *req*. ■

For example, the user query for a 3D scene of a yellow colored car which is capable of carrying passengers and goods to a given destination is transformed by iRep3D into the query tuple $q = \{$"$yellowcar$"; $\tau(YC, O_{req}) = Vehicle^P \sqcap Car^P \sqcap \forall canCarry.\ (Goods^P \sqcap Passenger^P)$; $SS = \{$[URI; haveFun; In(Passenger psg, Goods gds, TargetLocation tl); Out(); Prec(availableFor(psg)); Eff(at(psg,tl)$\wedge$ at(gds,tl))]$\}$ $GF = \{[name : color; f : Material; \{(0.8, 0.9, 0.15)\}]\}\}$.

2.2 Building of Scene Indices

A 3D scene $x \in \mathcal{X}$ is indexed by an iRep3D repository with respect to its different kinds of semantic annotation. In particular, iRep3D is creating three inverted scene indices for (a) scene concepts $C \in O$, (b) semantic services $ss \in \mathcal{SS}$, and (c) geometric features $gf \in \mathcal{GF}$, and stores the indexed scenes in an XML database.[7]

[7]Each annotated 3D object of the XML-based structure of an annotated scene in XML3D, X3D or COLLADA is indexed by iRep3D as an individual 3D scene with a unique (XPATH) scene identifier. We omit the details of subscene identification for reasons of space.

Scene index for concepts. The scene concept index I_{SC} of the repository is a set of ranked lists $R(C')$ of scenes $x \in X$ for all concepts C' in the scene ontology O. Each of these lists $R(C') \in I_{SC}$ contains pairs $(x.id, d(x,C'))$ of scenes x together with their relevance scores $d(x,C')$ for the considered (list) concept $C' \in O$. The scene relevance score is computed as weighted degree $d(x,C')$ of the approximated logical subsumption relation $s_{ab,\sqsubseteq}(x.C,C')$ between the list concept C' and the scene concept of x. Finally, each of these lists $R(C')$ of scenes of the scene concept index I_{SC} is sorted in descending order of the computed scene relevance scores.

Approximated concept subsumption. The above mentioned degree $s_{ab,\sqsubseteq}(C,C') \in [0,1]$ of approximated logical concept subsumption between concepts C, C', where C is approximately subsumed by C', bases on the process of structured logical concept (contraction and) abduction. That is, the incompatible part G, the compatible and the missed parts K and M of the logical definition of concept C compared with the one of concept C' are first identified ($C = G \sqcap K$) by means of concept contraction (Di Noia et al., 2009). These identified parts are then used by the process of logical abduction to rewrite the original concept definition of C such that the resulting approximated concept C_{app} is logically subsumed by the target concept C'.

Definition 5: *Approximated logical concept subsumption*
Let C^P a primitive term in C, which conflicts with a primitive term $\bar{C}^P$ (named as the counter-part of C^P) in C'; $|C|$ the number of conjunctive primitive terms in C; $PC(C)$ the set of primitive concepts of C; $PR(C)$ the set of primitive roles of C; $PRE(C)$ the set of primitive numeric restrictions of C. The approximated concept subsumption score $s_{ab,\sqsubseteq}(C,C')$ is computed as follows:

$$\begin{aligned}
s_{ab,\sqsubseteq}(C,C') &= \tfrac{|K|}{|C'|} \cdot (1 - s_{acf}(C,C')),\\
s_{acf}(C,C') &= \tfrac{\sum_{C^P \text{ in } G \text{ or } M}(s_{cf}(C^P,C') \cdot w(C^P,C'))}{|C|},\\
s_{cf}(C^P,C') &= 1, \text{ if } C^P \text{ in } M \text{ or } C^P \text{ in } PC(C) \cup PR(C)\\
s_{cf}(C^P,C') &= \tfrac{rg(C^P) \setminus rg(\bar{C}^P)}{rg(C^P)}, \text{ else } (C^P \in PRE(C)) .\\
w(C^P,C') &= \tfrac{1}{\sum_{C'^P \text{ in } G \text{ or } M} impt(\bar{C}^P,C')} \cdot impt(\bar{C}^P,C').
\end{aligned}$$

where $\frac{|K|}{|C'|}$ denotes the proportion of the compatible part K of C w.r.t. C'; s_{acf} the averaged strength of logical conflicts between C and C'; $s_{cf}(C^P,C')$ the strength of an (atomic) logical conflict on C^P in C w.r.t. C'. The latter is computed as follows: If C^P is in M then C^P will surely appear in the abduced (new) concept C_{app} of C w.r.t. C'; while in case of C^P being a primitive concept or role in G, any logical conflict on C^P will cause the full rewriting of C^P during concept abduction. If C^P is a primitive numeric restriction in G, the conflict strength is the fraction of uncovered range of C^P w.r.t. its counter-part $\bar{C}^P$ in C'. The function $rg(C^P)$ computes the restricted numeric range of $C^P \in PRE(C)$.
Each atomic conflict strength $s_{cf}(C^P,C')$ is further weighted with a weight $w(C^P,C')$ $((\sum_{C^P \text{ in } G \text{ or } M} w(C^P,C') = 1,\ w(C^P,C') > 0)$ which estimates the importance of this conflict on C^P w.r.t. C' for the corresponding approximated logical subsumption relation. Let C'_{lp} the direct parent concept of C' in O (O_s); C'' the rewritten (abduced) concept of C' which is generated by replacing $\bar{C}^P$ with C^P if C^P is in G, or removing C^P from C' if C^P is in M. The binary function $impt(\bar{C}^P,C') \in \{a,b\}$ $(0 < a < b \leq 1)$ determines the importance of $\bar{C}^P$ in terms of keeping the hierarchy of C' in O (O_s): It returns b if $C'' \sqsubseteq C'_{lp}$ is false; a otherwise. In other words, if the replacement of $\bar{C}^P$ (in C') with C^P or the removal of C^P makes C'' no longer a subsumee of C'_{lp}, the conflict on C^P between C and C' then has a greater negative impact on C being subsumed by C'. ■

Example 1: Consider the example of an annotated 3D scene $x \in X$ in figure 1. The indexing of x in the scene index for scene concepts starts with computing the similarity score $s_{ab,\sqsubseteq}(Toledo,C')$ between scene concept $Toledo$ and each concept C' in the given ontology O. Let the logical unfolding of the defined concept $FamilyCar \in O$ (abbr. FC) $\tau(FC) := Vehicle^P \sqcap Car^P \sqcap \leq 4 hasWheels.Wheel^P \sqcap \forall canCarry.Goods^P \sqcap \forall canCarry.Passenger^P \sqcap \forall ownership.Private^P \sqcap \forall hasNickname.Name^P$. Further, let $PrivateCar \in O$ (abbr. PC) the direct parent concept of FC with $\tau(PC) = Vehicle^P \sqcap Car^P \sqcap \forall ownership.Private^P \sqcap \forall hasNickname.Name^P$. For indexing x in $R(FC)$, the relevance score $d(x,R(FC)) = s_{ab,\sqsubseteq}(Toledo, FC)$ of x is computed based on approximated logical subsumption as follows: The determined incompatible part $G = \neg\forall onwership.Private^P$ and the missed part $M = \forall canCarry.Goods^P \sqcap \forall hasNickname.Name^P$ of scene concept $Toledo$ w.r.t. list concept FC lead to the respective conflict strengths: $s_{cf}(\neg\forall onwership.Private^P, FC) = 1$, $s_{cf}(\forall canCarry.Goods^P, FC) = 1$ and $s_{cf}(\forall hasNickname.Name^P, FC) = 1$. Then, in very brief, the abduction of FC' from FC based on these conflicts of $Toledo$ w.r.t. FC is done as follows. Let $impt(\cdot,\cdot) \in \{0.1, 0.9\}$. If we replace $\forall onwership.Private^P$ in $\tau(FC)$ with

$\neg\forall onwership.Private^P$, then the abduced concept FC' is no longer subsumed by PC which implies $impt(\forall onwership.\ Private^P,\ FC) = 0.9$, therefore: $w(\forall onwership.Private^P,\ FC) = w(\forall hasNickname.Name^P,\ FC) = \frac{0.9}{0.9+0.1+0.9} = 0.47$, $w(\forall canCarry.Goods^P,\ FC) = 0.06$. Subsequently, the averaged conflict strength is $s_{acf} = \frac{1}{|Poledo|} \cdot \sum_{C^P \text{ in } G\sqcap M}(s_{cf}(C^P,FC) \cdot w(C^P,FC)) = \frac{1}{5}(1 \cdot 0.9 + 1 \cdot 0.1 + 1 \cdot 0.9) = 0.38$. Finally, $s_{ab,\sqsubseteq}(Toledo,\ FC) = \frac{|K(Poledo,FC)|}{|FC|} \cdot (1 - s_{acf}(Poledo,FC)) = \frac{3}{7} \cdot (1 - 0.38) = 0.26$. A pair $(x, 0.26)$ is inserted into $R(FC) \in I_{SC}$. $\Diamond$

Scene index for services. The scene index I_{SS} for semantic services consists of two (sub-)indices: the scene index I_{IO} for semantic service I/O concepts, and the scene index I_{PE} for semantic service preconditions and effects. Similar to the scene index for scene concepts, we create the first index I_{IO} as a set of two ranked lists $R(C_s)[i] = \{(x.id, d_s(x,C_s)[i])\}$ and $R(C_s)[o] = \{(x.id, d_s(x,C_s)[o])\}$ of scenes $x \in \mathcal{X}$ for each concept C_s in the given service ontology O_s of the repository. Each entry of the list $R(C_s)[i]$ ($R(C_s)[o]$) states that some scene x is annotated with a semantic service $ss \in x.SS$ which has an input (output) parameter concept $C'_s \in O_s$ that is sufficiently and maximally similar with the list concept $C_s \in O_s$: $d_s(x,C_s)[l] = max_{C'_s \in ss[l], ss \in x.SS}\ d_c(C'_s,C_s)[l], l \in \{i,o\}$ where $d_c(C'_s,C_s)[l] = fr(C'_s)[l] \cdot s_{ab,\sqsubseteq}(C'_s,C_s)$ denotes the approximated concept similarity subject to $s_{ab,\sqsubseteq}(C'_s,C_s) \geq \theta \in [0,1]$. The weight $fr(C'_s)[l] = \frac{|x.SS_{C'_s}[l]|}{|x.SS|} \cdot max_{ss \in x.SS_{C'_s}} \frac{n(C'_s,ss[l])}{|ss|}$ is the frequency of occurrence of concept C'_s in $x.SS$ with $n(C'_s,ss[l])$ the number of occurrences of C'_s in the input ($l = i$) or output ($l = o$) parameter set and $|ss|$ the total number of parameters of service $x.ss$. Each list $R(C_s)[l], l \in \{i,o\}$ of scenes is sorted in descending order of their relevance scores $d_s(x,C_s)[l]$.

The second index I_{PE} consists of ranked lists $R(\alpha)[p]$ and $R(\alpha)[e]$ of scenes $x \in \mathcal{X}$ for each defined predicate $\alpha \in A_s$ which appears in the logical precondition or effect of annotated services of these scenes. Each scene x is ranked in the lists $R(\alpha)[l], l \in \{p,e\}$ of pairs $(x.id, d_a(x,\alpha[l]))$ according to its relevance score $d_a(x,\alpha)[l] = pl(\alpha,x)[l]$ which denotes the plausibility of α over the preconditions (effects) of all services of x. In particular, let $l' \in \{p,e\}$; $A_s(x)[l']$ the set of non-negative predicates that appear in the preconditions or effects of services provided by x; and $\mathcal{H} = 2^{A_s(x)[l']}$:

$$
\begin{aligned}
pl(\alpha,x)[l'] &= 1 - Bel_{A_s(x)[l']\setminus\alpha}(x),\\
Bel_H(x)[l'] &= \textstyle\sum_{h\subseteq H} \nu(h),\\
\nu_H(x)[l'] &= \tfrac{n_H(x)[l']}{n_{\mathcal{H}}(x)[l']}, \text{ subject to:}\\
& \quad \nu(\emptyset) = 0, \textstyle\sum_{H\subseteq\mathcal{H}} \nu(H) = 1,\\
n_{\mathcal{H}}(x)[l'] &= \textstyle\sum_{H\subseteq\mathcal{H}} n_H(x)[l'],\\
n_H(x)[l'] &= \textstyle\sum_{\alpha\in H} n_\alpha(x)[l'],\\
n_\alpha(x)[l'] &= \textstyle\sum_{ss\in x.SS} P_\alpha(x.ss[l']|\alpha).
\end{aligned}
$$

where $P_\alpha(x.ss[l']|\alpha)$ is the probability that the logical precondition or effect $x.ss[l']$ is evaluated to true given that α is true according to the truth table of $x.ss[l']$.

Example 2: Consider the scene x in Example 1. Let the semantic service *transport* (abbr. *tr*) the only one provided by scene x, and $\theta = 0.25$. The process of indexing x in the scene index I_{SS} starts with the subindex I_{IO}. Assume that the degree $s_{ab,\sqsubseteq}(Passenger,\ People)$ of approximated subsumption relation between service input concept *Passenger* and requested concept *People* in the service ontology O_s of the repository is 0.5. The frequency of occurrence of *Passenger* in $x.SS$ is $fr(Passenger)[i] = \frac{|1|}{|1|} \cdot max_{ss \in x.SS_{Passenger}[i]} \{\frac{1}{2}\} = 0.5$. The weighted and maximal approximated similarity between *Passenger* and $People \in O_s$ then is $d_c(Passenger,\ People)[i] = fr(Passenger)[i] \cdot s_{ab,\sqsubseteq}(x,\ People) = 0.5 \cdot 0.5 = 0.25$. Note that $d_c(Location, People)[i]$ is ignored since their similarity score $s_{ab,\sqsubseteq}(Location,\ People) = 0.1$ is smaller than θ. Finally, $d_s(x, People)[i] = d_c(Passenger,\ People)[i] = 0.25$. Assume that $d_s(x, Place)[i] = 0.63$. The pair $(x, 0.25)$ ($(x, 0.63)$) is inserted into the rank list of scenes $R(People)[i]$ ($R(Place)[i]$). For indexing x in the second subindex I_{PE}, the plausibilities of the predicates *availableFor*, *reachable* and *at* (denoted as *ava*, *rea* and *at*, respectively) are computed. For this purpose, we consider the truth tables for the service precondition $tr[p]$ and effect $tr[e]$, respectively:

$ava(psq)$	T	T	F	F
$rea(lc)$	T	F	T	F
$tr[p]$	T	F	F	F

,

$at(psg,lc)$	T	F
$tr[e]$	T	F

Based on these truth tables the indexing process estimates the probabilities $Pa(tr[p]|ava(psg)) = 0.5$, $Pa(tr[p]|rea(lc)) = 0.5$ and $Pa(tr[e]|at(psg,at)) = 1.0$. Regarding the power set $\mathcal{H} = 2^{\{ava,rea\}}$ of the predicates, we obtain the plausibility values $pl(ava,\ x)[p] = pl(rea,\ x)[p] = 0.9$ and $pl(at,x)[e] = 1$. As a result, the pairs $(x, 0.9)$, $(x, 0.9)$ and $(x, 1)$ are inserted into the rank lists $R(ava)[p]$, $R(rea)[p]$ and $R(at)[e]$ of the subindex I_{PE}. $\Diamond$

Scene index for geometric features. In contrast to the scene indices for concepts and services, the scene index I_{GF} is concerned with the geometric features of a scene x. Each such feature $gf \in \mathcal{GF}$ of type $f \in \mathcal{GF}$ (cf. Def. 3) consists of a set K_f of attributes k with numeric or string data type. The scene index I_{GF} is the set of B+ trees $bt(f,k)$ of scenes which is built for every attribute $f.k$ of each feature $f \in \mathcal{F}$. The scenes $x \in \mathcal{X}$ are maintained in these trees according to the feature attribute values $v(f.k)$, if x has such values: Each leaf node of $bt(f,k)$ points to the address of a ranked list $R_j(f,k)$ of pairs $(x.id, x.v(f.k))$ in the descending order of $x.v(f.k)$.

Let M_l the maximum number of scenes that a ranking can accommodate; M_n the maximum fanout (the number of child nodes) of each node; $\mathcal{X}_f \subseteq \mathcal{X}$ the subset of scenes containing a value of $k \in K_f$: The construction of $bt(f,k)$ is performed in the following steps: (i) sort scenes in $\mathcal{X}_f$ in the descending order of $v(f.k)$; (ii) compute the number n_l of needed ranked lists: $n_l = \lceil \frac{|\mathcal{X}_f|}{M_l} \rceil$; (iii) create $\lceil \frac{n_l}{M_n} \rceil$ leaf-nodes; (iv) initialize the pointers from leaf-node to rankings and label each pointer with the attribute value of the first entry in the corresponding ranking; (v) compute the number of needed non-leaf-nodes in each level from bottom to top and create their pointers and labels. If a scene x contains multiple instances of the same feature type attribute, then x has multiple entries in $bt(f,k)$. Each of these entries are additionally labeled with the specific name $gf.name$ of the geometric feature f of x.

Example 3: Consider the scene x in Example 1. For indexing x in the geometric index I_{GF}, the B+ tree $bt(Material,\ diffuseColor)$ is created since x has a geometric feature of type $f = Material$ and x has a value $x.v(Material.diffuseColor) = (1.0, 0.9, 0.0)$ for the feature attribute $k = diffuseColor$. Thus, $(x, (1.0, 0.9, 0.0))$ is inserted into a ranked list $R_j(f,k)$ of scenes refered to by the tree $bt(Material, diffuseColor)$. $\Diamond$

3 HYBRID SEMANTIC SCENE RETRIEVAL

Once the semantic indices have been created off line for a given collection of 3D scenes by the repository, the user can make requests q (cf. Def. 4) for relevant 3D scenes based on concepts, services, and geometric features. Key idea of answering a scene request with high precision and fairly fast is (a) to process the respective subqueries in the corresponding indices I_{SC}, I_{SS} and I_{GF} in parallel, and (b) to aggregate the resulting rank lists $\mathcal{R}_{sc}$, $\mathcal{R}_{ss}$ and $\mathcal{R}_{gf}$ of scenes that are relevant for q with Fagin's threshold algorithm. Finally, the iRep3D repository returns and displays the top-k relevant scenes to the user. As mentioned above, iRep3D's preprocessing of annotated 3D scenes allows the indexing of annotated scenes that are part of others. If indexed scenes are relevant but part of non-relevant scenes, only the first will be displayed together with meta-information on the latter. For example, if a scene model of a yellow car is requested and such an indexed scene is found to be part of another indexed scene of a parking garage with tens of different cars, only the first scene is returned to the user with a link to the overall scene.

Scene concept-based query processing. If the query q includes a request for scenes about some concepts C' defined by the logical expression $\tau(C', O_{req})$ then iRep3D classifies this concept into the current scene ontology O and returns the corresponding rank list $\mathcal{R}_{SC}$ of scenes which are relevant to q with respect to the approximated logical similarity of their scene concepts with the requested one C'.

Scene service-based query processing. If the query q contains the description of desired scene services ($ss \in q.SS$) then iRep3D processes the respective subqueries in the scene index for services. Firstly, for each $ss \in q.SS$, a rank list $R(ss)$ of scenes that are relevant to ss is computed. For this purpose, the indices I_{IO} and I_{PE} are searched in parallel. The resulting ranked lists $R(ss)[io]$ and $R(ss)[pe]$ are further merged into the list $R(ss)$ of scenes which are relevant to ss. Finally, all lists $R(ss)$ of $ss \in q.SS$ are merged, which leads to the ranked list $\mathcal{R}_{SS}$ of scenes which are relevant to q in terms of the requested semantic services.

Searching index I_{IO} for scenes with service ss: For each $ss \in q.SS$, iRep3D first retrieves in parallel a set $\{R(C_s')[l], l \in \{i,o\}\}$ of ranked lists of scenes each of which relevant to a distinct service signature parameter concept $C_s'[l]$ in $ss[l]$. In particular, the logical expression of each concept C_s' in $ss[i]$ ($ss[o]$) gets classified into the ontology O_s, and the corresponding ranked list with suffix $[i]$ ($[o]$) is eventually retrieved. Subsequently, the aggregation with TA(Fagin, 2002) is performed on $\{R(C_s')[l]\}$ to compose a ranked list $R(ss)[io]$ of scenes relevant to q with respect to the I/O parameters of the requested service ss.

In particular, the TA performs a sorted scan of all its input rank lists in $\{R(C_s')[l]\}$ from top to bottom in parallel. The i-th scan fetches the score values at the i-th positions of all lists in $\{R(C_s')[l]\}$, and then employs a m-ary (m the cardinality of $\{R(C_s')[l]\}$ for ss)

function t that computes the aggregated relevance score and threshold. The general form of t is given in (Fagin, 2002) and can be further customized for any application. In our context, we define t as the weighted average of the vector of scores $\vec{s}$ fetched from each rank list in $\{R(C'_s)[l]\}$ per scan. The weight v_j of the j-th list in $\{R(C'_s)[l]\}$ refers to the number of occurrences of C'_s in either $ss[i]$ or $ss[o]$:

$$t(\vec{s}) = \frac{\sum_{j=1}^{m} v_j \cdot s_j}{\sum_{j=1}^{m} v_j}$$

Each scan performed by the TA may find a new scene x_n that does not exist in the current $R(ss)[io]$. To insert x_n into $R(ss)[io]$, iRep3D computes the aggregated relevance score $s(x_n, ss)[io]$ of scene x_n to q w.r.t. the I/O concepts of $ss \in q.SS$: From each ranked list in $\{R(C'_s)[l]\}$, TA collects (possibly by random access) the so far missed $d_s(x_n.id, C'_s)$ of x_n; and further applies the t function on all $d_s(x_n.id, C'_s)$ in order to compute $s(x_n, ss)[io]$. Then TA maintains a threshold value T for determining its termination, which is updated with the t function value over the latest scanned values after each scan. TA terminates, if $T \leq s(x, ss)[io]$ for all the ranked objects x in $R(ss)[io]$.

Searching index I_{PE} for scenes with service ss: For each $ss \in q.SS$, the searching of I_{PE} for ss results in two sets of ranked lists $\{R(\alpha)[l']\}$ ($l' \in \{p, e\}$) for every non-negative predicate α in $ss[l']$. In addition, it merges the ranked lists in each set into a list $R(ss)[l']$ of scenes that are relevant to ss in terms of $ss[l']$. For this purpose, multiple pairs of the same scene x in different lists are merged; pairs in different lists are merged if they have the same scene id. The score value $s(x, ss[l'])$ of x in $R(ss)[l']$ of each result pair is computed by applying the Gödel minimum t-norm and maximum t-conorm functions according to the conjunctive, respectively disjunctive relations between the predicates in $ss[l']$:

$$\begin{aligned} s(x, ss[l']) &= min_{cla \in ss[l']}(s(x, cla[l'])), \\ s(x, cla[l']) &= max_{\alpha \in cla}(d_a(x, \alpha)[l']). \end{aligned}$$

where $cla[l']$ denotes a clause of disjunctive predicates. Finally, the search process merges $R(ss)[p]$ and $R(ss)[e]$ in order to compute $R(ss)[pe]$ of scenes which are relevant to ss in terms of the precondition and effect. The completion of the parallel computations of $R(ss)[io]$ and $R(ss)[pe]$ triggers their merging and yields the ranked list $R(ss)$ of scenes relevant to q in terms of $ss \in q.SS$. The relevance score $s(x, ss)$ of x in $R(ss)$ is the convex combination of the corresponding scores in $R(ss)[io]$ and $R(ss)[pe]$:

$$s(x, ss) = \phi s(x, ss[io]) + \psi s(x, ss[pe]),$$

where the real positive values ϕ and ψ ($\phi + \psi = 1$) are the weights of IO and PE matching respectively. They can vary in specific systems with different concerns.

Merging of scene rank lists $R(ss)$ for all $ss \in q.SS$: In a next step, the resulting ranked lists $R(ss)$ for all $ss \in q.SS$ are merged, if (some of) their entries in different lists share the same id. The relevance score $s(x, q.SS)$ for x with respect to $q.SS$ is the average of the scores $s(x, ss)$ of x in $R(ss)$ for each service ss:

$$s(x, q.SS) \quad = \frac{1}{|q.SS|} \sum_{ss \in q.SS} s(x, ss).$$

Finally, the merged list are resorted in descending order of $s(x, q.SS)$ yielding the ranked list $\mathcal{R}_{SS}$ of scenes partially relevant to q with respect to $q.SS$.

Geometric feature-based query processing. If the query q contains the description of desired geometric features $gf \in q.GF$ of a scene then iRep3D processes the respective subqueries in the scene index I_{GF} as follows. Firstly, for each $gf \in q.GF$, a parallel search is performed in the B+ trees $bt(gf.f, k)$ where each search results in a ranked list $R(gf.f.k)$ of scenes relevant to q in terms of $gf.f.k$. Please note that $R(gf.f.k)$ does not have similarity scores but the feature attribute values. Secondly, for each entry $(x.id, x.v(f.k)) \in R(gf.f.k)$, iRep3D computes the degree of geometric feature attribute similarity $s_k(q.v(f.k), x.v(f.k))$ between the requested and existing feature attributes based on its values $q.v(f.k)$ and $x.v(f.k)$. This results in a new rank list $R(q, gf.f.k)$ of scenes that are relevant to q for the requested value of $gf.f.k$. Thirdly, all lists $R(q, gf.f.k)$ of attributes which belong to the same feature type $gf.f$ are further merged (by scene id) into a ranking $R(q, gf)$ of scenes that are relevant to q with respect to the gf. Finally, all feature-level rankings $R(q, gf)$ for all $gf \in q.GF$ are merged into one which yields the overall ranking of scenes relevant to q.

The data types of geometric feature attributes defined in the X3D, XML3D and COLLADA specifications include the following primitive data types: (i) single number, string or boolean (e.g. `SFDouble`, `SFString`); (ii) 2-, 3- or 4-ary tuple of numbers or strings (e.g. `SFVec2d`, `SFVec3f`, `float4_type`); (iii) vector of values of the types in (i) and (ii) (e.g. `MFDouble`, `MFVec3d`). Let $tp(k)$ denote the primitive data type of feature attribute k. iRep3D computes the geometric feature attribute similarity score as follows:

$s_k(v_1, v_2) =$

- $xor(v_1, v_2)$, if $tp(k)$ is single boolean;
- $EDS(v_1, v_2) = 1 - \frac{ED(v_1, v_2)}{max(|v_1|, |v_2|)}$, if $tp(k)$ is single string, where $|v_1|$ denotes the length of v_1;
- $min(\frac{v_1}{v_2}, \frac{v_2}{v_1})$, if $tp(k)$ is single number;
- $\frac{1}{|v_1|} \sum_{i=1}^{|v_1|} xor(v_{1i}, v_{2i})$, if $tp(k)$ is a boolean vector, where $|v_1|$ denotes cardinality of v_1;
- $cos\_sim(v_1, v_2)$, if $tp(k)$ is a pair, triple or a vector

of numbers;
• $VEDS(v_1,v_2) = \frac{1}{|v_1|}\sum_{i=1}^{|v_1|} EDS(v_{1i},v_{2i})$, if $tp(k)$ is a pair, triple or a vector of strings;
• $\frac{1}{|v_1|}\sum_{i=1}^{|v_1|} cos\_sim(v_{1i},v_{2i})$, if $tp(k)$ is a vector of pairs or triples of numbers;
• $\frac{1}{|v_1|}\sum_{i=1}^{|v_1|} VEDS(v_{1i},v_{2i}$, if $tp(k)$ is a vector of pairs or triples of strings;
where $nor(v_1,v_2)$ is the exclusive OR of v_1 and v_2; $EDS(v_1,v_2)$ the Levenstein edit distance of v_1 and v_2; $cos\_sim(v_1,v_2)$ the cosine distance of v_1 and v_2. We omit the data types `SFImage, MFImage, SFTime` and `MFTime` of the X3D specification since they are not considered as geometric data types.
The geometric feature-based retrieval of relevant scenes computes the rank lists $R(gf.f.k)$ each of which entries contain the identifiers of scenes and their values $v(f.k)$ for the requested feature attribute k. Instead of directly retrieving a pointed ranking by a leaf node of the B+ tree, $R(gf.f.k)$ is computed by applying a window tolerant strategy which retrieves at most N entries from both parts of the entry $(x.id, x.v(gf.f.k))$ whose feature attribute value has a minimum distance to $q.v(gf.f.k)$ (N is called half-window width value).

Final aggregation of relevance rank lists of scenes. In the end, the computed three different relevance rank lists $\mathcal{R}_{sc}$, $\mathcal{R}_{ss}$ and $\mathcal{R}_{gf}$ of 3D scenes for q are merged by, again, leveraging Fagin's TA algorithm as described above. If the score of a scene x is missing in some of these rank lists, the lowest score in the respective list is used by default. The TA terminates if the threshold is not larger than the least score of the A-th (cf. Def.4) entry in the total ranking, or all three lists above are scanned over.

4 EXPERIMENTAL EVALUATION

The repository iRep3D has been fully implemented in Java and stores its 3D scenes in the XML database BaseX. We conducted an experimental evaluation of the performance of iRep3D in comparison with three other representative open-source repositories for 3D scenes. For this purpose, we selected (a) the FB3D system for functional and behavioral ontology-based semantic retrieval of 3D scenes (Camossi et al., 2007), (b) the RIR system for RDF index-based scene retrieval approach (RIR) (Alvez and Vecchietti, 2011), and (c) the syntactic-based 3D model repository ADL.
Experimental settings. Since there is, to the best of our knowledge, no 3D scene retrieval test collection publicly available yet, we built a first version of it, called 3DS-TC, which consists of 616 manually annotated scene graphs (591 in X3D[8], 25 in XML3D). The respective scene ontology O in OWL2 contains 260 concepts, 48 roles and 7 role restrictions, and the scenes in 3DS-TC are also annotated with references to 33 services in OWL-S in total. The precondition and effect of services are encoded in RDF plain literals. As mentioned above, all annotations are embedded into the scene graphs with standard RDFa. Further, the test collections consists of a set Q of 20 scene queries together with relevance sets each of which containing 10 relevant scene graphs with relevance scores $rel \in \{1.0, 0.9, \ldots, 0.1\}$), while non-relevant scenes were assigned a score of 0 by default. Further, we set $A = 10$ for all $q \in Q$; $\theta = 0.5$; $\phi = \psi = 0.5$; $a = 0.1$, $b = 0.9$ for the importance function; and the half-tolerance window width $N = 10$.
In order to enable FB3D reasoning on functional descriptions of scenes, we added 12 concepts and 4 roles extracted from the annotated scene services to our scene ontology. Besides, we let FB3D pre-load the scene concepts before its query processing in order to eliminate the loading and parsing time of 3D scenes. For the RIR system, we (i) created the required RDF triples for the scene concepts and service parameter concepts of annotated 3D scenes with the Jena OWL analyzer[9], (ii) employ the indexing facilities of MySQL database to index the generated RDF triples in terms of their subject, predicate and object, and (iii) constructed one SPARQL query for each query $q \in Q$. For the ADL system, we store the syntactic descriptions of scene semantics provided in the meta-tags in a MySQL database.

	iRep3D	FB3D	RIR	ADL
AP	**0.721**	0.490	0.633	0.408
DCG_{10}	**2.133**	0.952	1.370	0.767
AQRT (sec)	0.166	1.887	0.059	**0.042**

Table 1: AP, DCG_{10} and AQRT of iRep3D and competitors

Performance evaluation measures. We use the following standard retrieval performance evaluation metrics for our comparative experimental evaluation of scene retrieval by the 3D scene repositories iRep3D, FB3D, RIR and ADL: Macro-average precision (MAP_λ) at 11 recall levels (RE_λ) (MAP@RE) with equidistant steps of 0.1; average precision (AP); Averaged discounted cumulative gain (DCG_{10}) at rank position 10; and average query response time (AQRT) in seconds.

[8] `http://www.web3d.org/x3d/content/examples`
[9] `http://jena.apache.org/`

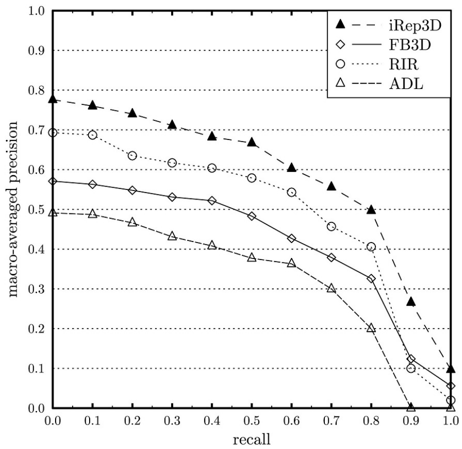

Figure 2: MAP@recall of iRep3D, FB3D, RIR, and ADL.

Evaluation results. The experimental results reveal, among other, that for the given collection 3DS-TC the iRep3D repository significantly outperforms its competitors in terms of retrieval precision (MAP@RE, AP and DCG_{10}): In particular, its average precision is 34%, 13%, and 55% higher than that of FB3D, RIR, and ADL, respectively. Compared with FB3D, the main reason of this improvement in precision is that iRep3D avoids misclassifications caused by strict logic-based matching of scene concepts and due to its hybrid semantic matching of scenes tolerates more parameter mismatches than the one-shot functional concept matching performed by FB3D. The RIR system alleviates the problem of text similarity-based classification failures of ADL by exploiting RDF-based scene descriptions but due to its exact SPARQL query pattern matching it still remains much less accurate than iRep3D. Given some conjunctive keyword query, ADL directly queries its underlying database by wildcard SQL and limits its search for relevant scenes by ignoring text segmentation.

On the other hand, the high precision of hybrid semantic retrieval of scenes by iRep3D is not achieved at the cost of extremely high response times. In fact, the average query response of iRep3D appears reasonably fast (0.166 secs) compared to those of FB3D (1.887 secs), RIR (0.059 secs) and ADL (0.042 secs). However, iRep3D is slower than RIR and ADL since it requires more time for logical classification of requested scene and service parameter concepts into its scene (and service concept) ontology than the SPARQL query processing by RIR and keyword matching by ADL.

5 RELATED WORK

Many content and geometric feature-based approaches to 3D model retrieval have been proposed in the past decade such as (Tangelder and Veltkamp, 2004; Bustos et al., 2007; Paquet et al., 2000) but their mutually incompatible geometric feature definitions and formalisms limit their usage. The majority of 3D scene retrieval systems still relies on merely syntactic-based classification of scenes based on their geometric or non-geometric descriptive properties. For example, (Gao et al., 2011) proposes a probabilistic classification of 3D objects based on a Gaussian process while (Leifman et al., 2005) refines geometric-topological feature matching with unsupervised off-line learning and subsequent on-line supervised feature extraction from scenes. The approaches presented in (Gong et al., 2011) and (Hou et al.,) perform SVM-based (off line) learning of 3D object classification based on their non-geometric features and label each grounded object with the category in a predefined universe of discourse. Similarly, (Akguel et al., 2010) proposes SVM-based learning of a geometric feature-based classifier of 3D object descriptions off-line, and then estimates a probabilistic similarity between a given query and candidate objects on line. In contrast to iRep3D, the average precision of these adaptive approaches to 3D scene retrieval essentially depends on the chosen type of kernel function and the training set used by the SVM for learning the binary relevane classifier of 3D scenes.

On the other hand, the leveraging of semantic technologies for 3D scene annotation and retrieval has gained some momentum recently. For example, the work presented in (Alvez and Vecchietti, 2011; Laborie et al., 2009) utilizes RDF stores with efficient SPARQL query processing for indexing and retrieving RDF-annotated 3D scenes. In these cases, however, the query answering requires exact matches of scene graph patterns and attribute labels. In (Hois et al., 2007b) an approach for 3D image recognition is proposed based on a logic-based scene ontology for object recognition during the planning of robot actions; and (Camossi et al., 2007) presents a knowledge-based system for a semantic annotation and retrieval of 3D models based on an a specific ontology in OWL-DL about scene formation, functionality and behavior. In contrast to these approaches, iRep3D leverages approximated logical reasoning on ontology-based conceptual semantics of annotated scenes which shows to be less prone to be affected by syntactic and strict pattern mismatches, and may avoid strict logic-based misclassifications of scene annotations. (Yang, 2010) proposes to use high-level content signatures and linguistic extensions of multimedia contents for being able to handle imprecise queries for 3D scenes but at the cost of potential loss of information about the original scene semantics. Möller et. al. (Peraldi et al., 2009) apply rule-based abduction on the extracted low-level semantic descriptions of multimedia objects for answering grounded conjunctive queries in the fact base of a given scene ontology. Unlike iRep3D, these retrieval approaches do not rely on efficient scene indexing, hence might not as well scale to very large and distributed settings of scene retrieval.

6 CONCLUSION

We presented a new approach, called iRep3D, for efficient semantic indexing and retrieval of XML-based annotated 3D scenes. Results of experimental performance evaluation over a given preliminary test collection of X3D and XML3D scenes shows that iRep3D can significantly outperform representative, open-source and state of the art multimedia retrieval systems in terms of average precision and with reasonable response time.

REFERENCES

Akguel, C., Sankur, B., Yemez, Y., and Schmitt, F. (2010). Similarity learning for 3d object retrieval using relevance feedback and risk minimization. *International Journal of Computer Vision*, 89(2):392–407.

Alvez, C. and Vecchietti, A. (2011). Efficiency analysis in content based image retrieval using rdf annotations. *Advances in Soft Computing*, pages 285–296.

Bustos, B., Keim, D., Saupe, D., and Schreck, T. (2007). Content-based 3d object retrieval. *Computer Graphics and Applications, IEEE*, 27(4):22–27.

Camossi, E., Giannini, F., and Monti, M. (2007). Deriving functionality from 3d shapes: Ontology driven annotation and retrieval. *Computer-Aided Design & Applications*, 4(6):773–782.

Di Noia, T., Di Sciascio, E., and Donini, F. (2009). A tableaux-based calculus for abduction in expressive description logics: Preliminary results. In *Proc. of DL09 Workshop*.

Fagin, R. (2002). Combining fuzzy information: an overview. *ACM SIGMOD Record*, 31(2):109–118.

Gao, B., Zhang, S., and Pan, X. (2011). Semantic-oriented 3d model classification and retrieval using gaussian processes. *Journal of Computational Information Systems*, 7(4):1029–1037.

Gong, B., Liu, J., Wang, X., and Tang, X. (2011). 3d object retrieval with semantic attributes. In *Proc. of the 19th ACM international conference on Multimedia*, pages 757–758. ACM.

Hois, J., Schill, K., and Bateman, J. (2007a). Integrating uncertain knowledge in a domain ontology for room concept classifications. *Research and Development in Intelligent Systems XXIII*, pages 245–258.

Hois, J., Wünstel, M., Bateman, J., and Röfer, T. (2007b). Dialog-based 3d-image recognition using a domain ontology. *Spatial Cognition V Reasoning, Action, Interaction*, pages 107–126.

Hou, S., Lou, K., and Ramani, K. Svm-based semantic clustering and retrieval of a 3d model database. *Computer Aided Design and Application*, 2(2).

Kalogerakis, E., Moumoutzis, N., and Christodoulakis, S. (2006). Coupling ontologies with graphics content for knowledge driven visualization. In *Proc. of the IEEE Virtual Reality Conference 2006*.

Kapahnke, P., Liedtke, P., Nesbigall, S., Warwas, S., and Klusch, M. (2010). Isreal: an open platform for semantic-based 3d simulations in the 3d internet. *The Semantic Web–ISWC 2010*, pages 161–176.

Klusch, M. (2012). The s3 contest: Performance evaluation of semantic service matchmakers. In *Semantic Web Services Advancement through Evaluation*. Springer.

Klusch, M. and Kapahnke, P. (2012). The isem matchmaker: A flexible approach for adaptive hybrid semantic service selection. *Web Semantics: Science, Services and Agents on the World Wide Web*.

Koutsoudis, A., Stavroglou, K., Pavlidis, G., and Chamzas, C. (2011). 3dsse–a 3d scene search engine: Exploring 3d scenes using keywords. *Cultural Heritage*.

Laborie, S., Manzat, A., and Sedes, F. (2009). Managing and querying efficiently distributed semantic multimedia metadata collections. *Multimedia, IEEE*, (99):1–1.

Leifman, G., Meir, R., and Tal, A. (2005). Semantic-oriented 3d shape retrieval using relevance feedback. *The Visual Computer*, 21(8):865–875.

Paquet, E., Rioux, M., Murching, A., Naveen, T., and Tabatabai, A. (2000). Description of shape information for 2-d and 3-d objects. *Signal Processing: Image Communication*, 16(1):103–122.

Peraldi, S., Kaya, A., and Möller, R. (2009). Formalizing multimedia interpretation based on abduction over description logic aboxes. In *Proc. of DL09 Workshop, Oxford, UK*.

Pittarello, F. and De Faveri, A. (2006). Semantic description of 3d environments: a proposal based on web standards. In *Proc. of the 11th international conference on 3D web technology*, pages 85–95. ACM.

Qi, G., Aggarwal, C., Tian, Q., Ji, H., and Huang, T. (2011). Exploring context and content links in social media: A latent space method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (99):1–1.

Tangelder, J. and Veltkamp, R. (2004). A survey of content based 3d shape retrieval methods. In *Shape Modeling Applications, 2004*, pages 145–156. IEEE.

Yang, B. (2010). Dsi: A model for distributed multimedia semantic indexing and content integration. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 6(1):3.