ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 4, April 2013

Automatic Template Extraction using Hyper Graph Technique from Heterogeneous Web Pages

D.Kanagalatchumy[1], Dr.S.Pushpa[2]

[1]PG Student, Computer Science and Engineering, Sri Manakula Vinayagar Engineering College, Puducherry

[2]Associate Professor , Department of Computer Science and Engineering, Sri Manakula Vinayagar Engineering College, Puducherry.

***Abstract*- World Wide Web is the most useful source of information. In order to achieve high productivity of publishing, the web pages in many websites are automatically populated by using the common templates with contents. The templates provide readers easy access to the contents guided by consistent structures. However, for machines, the templates are considered harmful since they degrade the accuracy and performance of web applications due to irrelevant terms in templates. Thus, template detection techniques have received a lot of attention recently to improve the performance of search engines, clustering, and classification of web documents. In this paper, we present novel algorithms for extracting templates from a large number of web documents which are generated from heterogeneous templates. We cluster the web documents based on the similarity of underlying template structures in the documents so that the template for each cluster is extracted simultaneously. We develop a novel goodness measure with its fast approximation for clustering and provide comprehensive analysis of our algorithm. Our experimental results with real-life data sets confirm the effectiveness and robustness of our algorithm compared to the state of the art for template detection algorithms.**

Keyword- *Clustering techniques, Template Extraction, DOM, MDL, Minhash, Hypergraph.*

I. INTRODUCTION

Most of the users or clients are based on information from the Worldwide Web. They retrieve the content and using for their day-to-day activities. Information is nothing but a data or knowledge, in its most restricted technical sense, is a sequence of symbols that can be interpreted as a message. Information can be recorded as signs, or transmitted as signals. Information retrieval is the activity of obtaining information resources relevant to an information need from a collection of information resources. Searches can be based on metadata or on full-text indexing. This information is gathered and managed by the organization called information management, which is the collection and management of information from one or more sources and the distribution of that information to one or more audiences. This sometimes involves those who have a stake in, or a right to that information. The management has the full control over planning. Templates form the best source for extraction of information from various fields. There are various fields in which templates are extracted and accepted for further use by which several algorithms and techniques are used.

To overcome the limitation of techniques with the web documents, the method of extracting templates from heterogeneous webpages are carried out here. Mostly, web pages are represented by HTML documents. These web documents are considered as trees for clustering. Due to the assumption of all documents being generated from a single common template, solutions for this problem are applicable only when ball documents are guaranteed to conform to a common template. However, in real applications, it is not trivial to classify massively crawled documents into homogeneous partitions in order to use these techniques. Here the DOM method of constructing trees are used which can be easy to handle lager number of web documents. The methods like Tree-edit distance is considered as very expensive and high rate of time Complexity, hence going for DOM construction.

1.1 Information Extraction And Template Mining

Template defines a specific type of event, with a set of linguistic roles for the typical articles or things involved in such an event. Templates can extract a richer representation which focuses on learning different facts. Information extraction (IE) is the task of automatically extracting structured information from amorphous and/or semi-structured or natural, machine-readable documents. In most of the cases this activity concerns processing human language

texts by means of natural language processing (NLP). Most work on Information Extraction been emerged from various research on NLP. However, taking a detailed view on IE into different groups namely Template filling, Message Understanding conferences (MUC) and other works on information Extraction has been worked out and published by Gaizauskas and Wilks (1998).

Template mining is a particular technique used in IE. Lawson et al. (1996) defined template mining as a natural language processing (NLP) technique used to extract data directly from text if either the data or text surrounding the data forms distinguishable patterns. When text matches a template, the system extracts data according to instructions associated with that template. Although different techniques are used for information extraction and knowledge discovery, Template mining are used from the olden days for the extraction of datas.

Currently, information extraction and retrievel are done mostly from the search engines. Search engines are basically huge databases containing millions and billions of records that include the URL of a particular Web page along with information relating to the content of the Web page supplied in the HTML by the author[8]. A search engine finds this information via a submission from the author or by the search engine by 'robot crawlers'. This technique can also be used in digital libraries.

II. RELATED WORK

The approach wrapper induction is used as supervised learning to learn data extraction rules from a set of manually labeled positive and negative examples. Manual labeling of data is, however, labor intensive and time consuming. Additionally, for different sites or even pages in the same site, the manual labeling process needs to be repeated because they follow different templates/patterns. The second approach is automatic extraction. A study is made to automatically identify data record boundaries. The method is based on a set of heuristic rules, e.g., highest-count tags, repeating-tags and ontology-matching.

The template extraction problem can be categorized into two broad areas. The first area is the site-level template detection[2] where the template is decided based on several pages from the same site. In site-level template, each template is decided based on several pages from same site. One page is considered as initial template and other pages are compared and updated when there are mismatches. First only tags are considered to find templates. Considering documents as trees in each document are usually too expensive.

In page-level template[2], each template is computed within a single document. Identify the records in document and extracts data from them. Clusters cannot be grouped document by URL. Some URL of different pages looks identical but having different contents. This gives cluster formation difficult. In partial tree alignment [1], the approach aligns multiple tag trees by progressively growing a seed (tag) tree. The seed tree, denoted by *T*s, is initially picked to be the tree with the maximum number of data fields.

Note that the seed tree is similar to the center tree but without the $O(k2)$ pair-wise tree matching to choose it. In the vertex system[3], A single website may contain pages compliant to multiple different templates. Different groups of template based pages can be classified by clustering the pages within the sites. A single XSLT rule is learnt for each cluster containing pages with similar structure. The clustering component starts by collecting sample pages P from the Web site for which rules is to be learnt.

III. CONSTRUCTING PATHS AND TOKENS

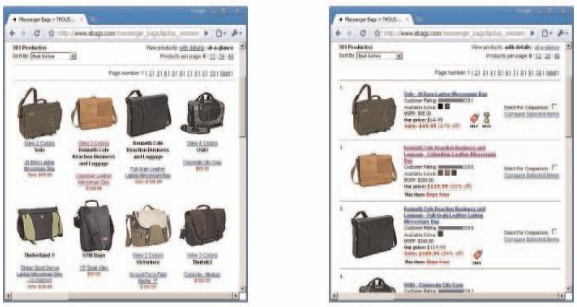

Figure 1: Different templates of same URL

In figure 1, the URL of the two pages looks different, whereas the contents of the pages are same. If we use only URLs to group pages, these pages from the different templates will be included in the same cluster. To overcome the limitation of the techniques with the assumption that the web documents are from a single template, the problem of extracting the templates from a collection of heterogeneous web documents, which are generated from multiple templates, was also studied. The paths of the web pages can be noted as tree and represented as in figure 2. For example, let us consider simple HTML documents and paths in Figure 2 and Table 1.

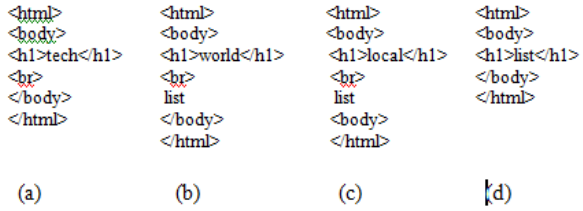

Figure. 2. Simple web documents.
(a) Document d1. (b) Document d2. (c) Document d3.
(d) Document d4.

Table1: Paths of tokens and their supports

ID	PATH	SUPPORT
P1	Document\<html>	4
P2	Document\<html>\<body>	4
P3	Document\<html>\<body>\<h1>	3
P4	Document\<html>\<body>\ 	3
P5	Document\<html>\<body>List	3
P6	Document\<html>\<body>\<h1>Tech	1
P7	Document\<html>\<body>\<h1>world	1
P8	Document\<html>\<body>\<h1>Local	1

Document d1 is represented as a set of paths $\{p_1; p_2; p_3; p_4; p_6\}$ and the template of both d_1 and d_2 is another set of paths $\{p_1; p_2; p_3; p_4\}$. Our goal is to manage an unknown number of templates and to improve the efficiency and scalability of template detection and extraction algorithms. To deal with the unknown number of templates and select good partitioning from all possible partitions of web documents, we employ Rissanen's Minimum Description Length (MDL) principle. The support value is determined for every path by considering the document.

IV. PRELIMINARIES

4.1 Tree Construction:

Trees are constructed from the paths of each document. The XML DOM views an XML document as a tree-structure. The tree structure is called a node-tree. All nodes can be accessed through the tree. Their contents can be modified or deleted, and new elements can be created. The node tree shows the set of nodes, and the connections between them. The tree starts at the root node and branches out to the text nodes at the lowest level of the tree. Figure 3 shows the nodes and the branches of the tree. Each root element has its attribute and an element, it continues till the branches of the tree gets over.

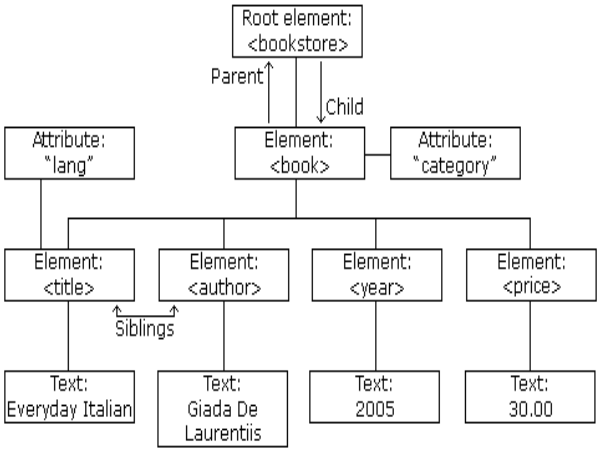

Figure 3: Sample DOM tree for books

4.2 *Essential Paths and Templates*:

From the collection of documents taken $D= \{d_1, d_2, \ldots d_n\}$, we assume the path as the set of all the path in the document. The path is taken for all the documents and its support value is calculated as the number of documents, which contains the particular path. As such, for each document we provide the minimum support threshold. The threshold for two same documents may be different.

The documents are taken as d_i and d_j and its threshold as td_i and td_j. If any path is contained in document di and support is atleast minimum support threshold, then the path is called essential path of di and it is denoted as $E(d_i)$. Corresponding with the path Pd and document, the matrix is formed with the values 0's and 1's. If the path pi is an essential path of document d_j, then the value of cell(i,j) in matrix is 1, otherwise it is 0.

4.3 *Clustering and calculating cost:*

Many clustering algorithms are experimented with the data sets TEXT-MDL is an agglomerative hierarchical clustering algorithm which starts with each input document as an individual cluster. When a pair of clusters is merged, the MDL cost of the clustering model can be reduced or increased. The procedure [7] GetBestPair finds a pair of clusters

whose reduction of the MDL cost is maximal in each step of merging and the pair is repeatedly merged until any reduction is not possible. For the optimal template paths of clusters, independence, sparsity, Candidacy of template paths, Decision of optimal template paths are calculated.

V. LITERATURE SURVEY

5.1 The RTDM Algorithm:

The main goal of the RTDM algorithm is to determine the cost of the minimal mapping, i.e., the tree-edit distance between two input trees, to perform template detection. There is a need to find the actual sequence of operations that lead to this minimal mapping. Here, we introduce RTDM TD, an extension of RTDM in which the mapping result is built while the mapping is determined[5]. There are several differences between the two algorithms

. RTDM requires as input a threshold, since processing can be halted in cases in which the partial cost being calculated is already above this threshold. As RTDM-TD needs to construct the complete sequence of operations, regardless of the cost, it does not need such a threshold. Another distinction is that RTDMTD treats certain nodes in a special way. Because nodes that are identical in two trees are regarded as being part of a template, RTDM-TD keeps track of cases in which no insertion, removal or update operations were applied to a given node.

5.2 The Partial Tree Alignment:

The approach [1] aligns multiple tag trees by progressively growing a seed (tag) tree. The seed tree, denoted by *T*s, is initially picked to be the tree with the maximum number of data fields. Note that the seed tree is similar to the center tree but without the $O(k2)$ pair-wise tree matching to choose it. The reason for choosing this seed tree is clear as it is more likely for this tree to have a good alignment with data fields in other data records.

This technique follows following steps:

- Input Given a webpage page**.**
- Building the Dom Trees Based on it Visual Information.
- DEPTA (Data extraction based on partial tree alignment)
- This method consists of two steps:

1) Identifying individual records in a page (Mining Data Regions).

2) Aligning and extracting data items from the Identified records (Identifying Data Records).

5.3 EXALAG Algorithm:

EXALG makes several assumptions regarding the unknown template and values used to generate its input pages[2]. The important assumptions:

A1: A large number of tokens occurring in template have unique roles, to bootstrap the formation of equivalence classes and subsequent differentiation.

A2: A large number of tokens are associated with each type constructor. Further, each type constructor is instantiated a large number of times in the input pages. This assumption is ensures that the equivalence class derived from a type constructor is recognized as an LFEQ.

A3: There is no "regularity" in encoded data that leads to the formation of invalid equivalence classes.

A4: There are "separators" around data values. In the model, this translates to the assumption that the strings associated with type constructors are non-empty.

VI. SYSTEM ARCHITECTURE

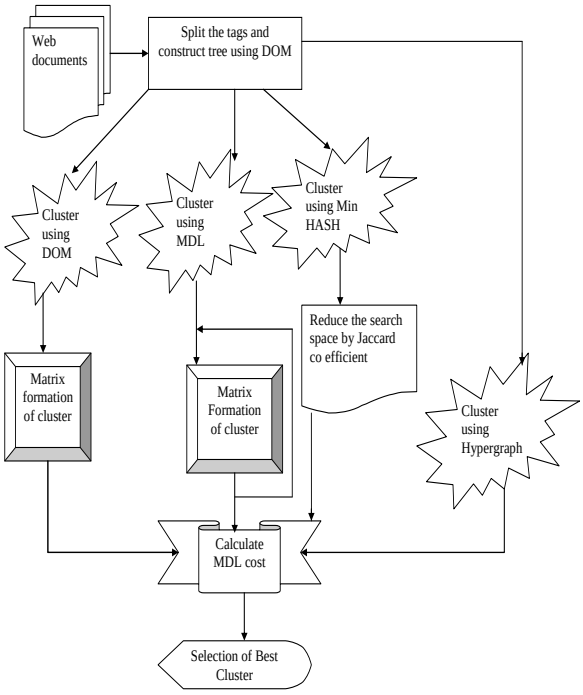

Figure 4: Overall Architecture

This is the architecture of the project in which it describes the entire process. Here various heterogeneous web documents are collected and by splitting a tag, a tree is constructed from the paths specified. Based on the similarities of the documents and the paths, clustering process is done. Various clustering techniques are used and cost is calculated. In the MDL clustering technique, taking each document as individual cluster, pair of clusters are merged in order to reduce the final cost. In Minhash algorithm, the duplications in the webpages are found and Jaccard co-efficient is used to reduce the search space. Hypergraph is used for reducing the cost and finding the minimum weight edge and so as to cut depends on similarity. Finally, the best algorithm selection was done for the efficient extraction.

VII. PROPOSED WORK

In the proposed work we used for heterogeneous extraction in which templates are extracted from multiple websites. Information extraction is attempting to find some of the structure and meaning in the hopefully template driven web pages. This provides a faster and efficient way of clustering. Clustering on web documents is used practically to handle large number of web documents. We cluster only documents not paths, and moreover, the numbers of clusters of columns and rows are dynamic. Here the efficient clustering technique Hyper graph is used for reducing the cost and extracting the template without sacrificing the efficiency. Thus reducing the steps for extraction of templates. Here the novel based algorithms are used which provide fast approximation for clustering with very high precision.

- *Pseudo code for Hyper graph*

```
MLCLUSTER (V, N)
   Level-no = 1
   While termination condition not occurred do
         C = Initialize-Clusters(v)
         C = SLCLUSTER (V, N, C)
         C = Refine (V, N, C)
         (V, N)  = Create-Next-Level-Hypergraph(V,
                                        N, C)
         Level-no = level-no + 1
     Return C
```

VIII. MODULES

8.1 Construction of DOM tree:

The Document Object Model (DOM) is a cross-platform and language -independent convention for representing and interacting with objects in HTML, XHTML and XML documents.
Objects in the DOM tree may be addressed and manipulated by using methods on the objects. The DOM presents an XML document as a tree-structure. Input HTML document are extracted from different WebPages which is taken for preprocessing. In the html document the text information and html tags are splitted separately. The separated html tags are been constructed into html DOM tree and have been investigated for clustering. Then the path is discovered by the DOM model and also it is used to calculate the number of support values in the individual tags.

- *Pseudo code for DOM*

```
Global:int nodeId ← 0; int chunkId ← 0; intArray[] f
irstNodeId; stack P; stackArray[] Pc;
Procedure ChunkCreate(dataIn, size)
Begin ← dataIn;
End ← begin+size+δ; /* avoid splitting XML tags
and going beyond EOF */
For each (e, type) ∈ [begin, end] do
Switch type do
Case START:
Node Id++; /* Next preorder number */
If first START tag in chunk then
Pc[chunk Id] ← P; /* Copy stack P */
First Node Id[chunk Id] ← nodeId;
End
P.push(node Id, e);
Break;
Case END: P.pop(); break;
Otherwise do nothing;
End
End
Chunk Id++;
Data In ← end+ 1;
```

8.2 Clustering and matrix formation using DOM (Document Object Model):

Here we use DOM based clustering mechanism to cluster the html tags that are extracted. In this clustering mechanism we are providing a support threshold value and this threshold value depends upon the document minimum path support value (D_i). For the formation of the matrix value we take the considerations as web document set D with its path set P_D, we use a $|P_D|$ x $|D_i|$ matrix M_E with 0/1 values to represent the documents with their essential paths. The value in the matrix M_E is 1 if a path is an essential path of a document d_i. Otherwise, it is 0.The MDL cost is identified in order to find the efficiency of the individual clustering algorithm and is given as

Cost(M,D) = Cost(D|M) + Cost(M) where Cost(M) - cost of the path and Cost(D|M) - cost of the data D if path M is given. Thus we do not need any additional template extraction process after clustering.

8.3 Applying Cluster using MDL (Minimum Description Length):

In order to manage the unknown number of clusters and to select a good partitioning of cluster from all possible partitions of HTML documents, we employ MDL principle. TEXT-MDL is an agglomerative hierarchical clustering algorithm which starts with each input document as an individual cluster. When a pair of clusters is merged, the MDL cost of the clustering model can be reduced or increased. The procedure GetBestPair finds a pair of clusters whose reduction of the MDL cost is maximal in each step of merging and the pair is repeatedly merged until any reduction is not possible.

- *Pseudo code for MDL*

```
Subdue( graph G, int Beam, int Limit )
queue Q = { v | vertex v has a unique label in G }
bestSub = first substructure in Q
repeat
newQ = {}
for each substructure S in Q
newSubs = ExtendSubstructure(S)
newQ = newQ U newSubs
Limit = Limit - 1
Q = substructures in newQ with top Beam MDL scores
if best substructure in Q better than bestSub
then bestSub = best substructure in Q
until Q is empty or Limit <= 0
return bestSub.

ExtendSubstructure(substructure S)
for each remaining attribute Ai
newSubs ¬ SelectAttributeValues(S, Ai)
return newSubs
SelectAttributeValues(substructure P, attribute A)
values V = {v | value v is unique in A}
compute frequency of each v in V, and determine if A is continuous or discrete
if A is continuous, compute mean
add attribute A to substructure P with values V
evaluate substructure P using MDL
bestP ¬ P
repeat
if A is continuous remove v from V that is farthest from mean
if A is discrete remove v from V that occurs least frequently
evaluate substructure P using MDL
if P is better than bestP
bestP ¬ P
until |V| = 1
return bestP
```

8.4 Cluster using Min HASH:

A way to consistently sample words from bags and which is a technique for quickly estimating how similar two sets are. This Clustering algorithm uses hash intersections to probabilistically cluster similar user data. In order to find the duplications in the web page we utilize the jaccard coefficient for similarity measurement.

```
C={c1,c2…..cn}
(ci,cj,ck)=GetBestPair(C)
Let ci and cj be the best pair of merging
Let ck be a new cluster made by merging ci and cj
While(ci,cj,ck) is not empty do
{
C=C-{ci,cj}U{ck}
(ci,cj,ck)=GetBestPair( C)
}
return C
end
```

8.5 Hyper graph:

A hypergraph is a generalization of a graph where in edges can connect more than two vertices and are called hyperedges. Hypergraph is considered as safe and high quality clustering algorithm[9]. Hypergraph (H) = (V, E) V::a set of vertices; E::a set of hyperedges. The clustering problem is then formulated as of finding the minimum-cut of a hypergraph. A minimum-cut is the removal of the set of hyperedges (with minimum edge weight) that separates the hypergraph into k unconnected components.

IX. CONCLUSION

Various clustering techniques like Document object model, Minimum Description Length, Minhash algorithm are observed and explained briefly. Each clustering algorithm forms a cluster based on the similarities between the templates from heterogeneous webpage's. MDL cost is calculated and the best algorithm is found. Novel based algorithms are used and thus it Provide fast approximation for clustering. Clustering on sampled

web documents is used to practically handle a large number of web documents. Clustering techniques is used in the web documents for cluster the similar template structure web pages. So, that the template for each cluster is extracted simultaneously.

Using MDL clustering, the cost can be reduced by pairing the clusters until any reduction is possible. In Minhash algorithm, iteration is done for each document while clustering, which forms more empty space. The duplications in the web page can be found and search space can be reduced, by utilizing Jaccard co-efficient which is used for comparing the similarity of sample sets. Hypergraph is used in text information retrieval with reduced cost. Thus it increase the throughput of the system and this technique can also be applied to new document collection. It provides fast and efficient way of clustering with high precision

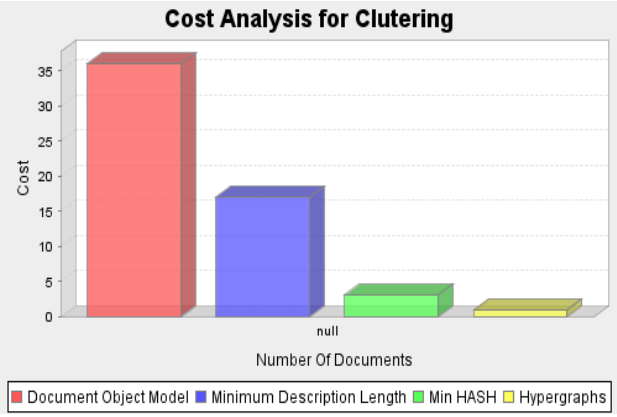

Figure 5: Cost analysis for clustering

From the figure 5, it is proved that the cost for clustering the web documents using hypergraph reduces the total cost. The graph is drawn by taking the cost as x-axis and number of documents as y-axis, respectively.

X. FUTURE WORK

From the results of our approach, we can predict that hyper graph technique is much helpful for extracting templates from different web pages. This work can be extended to derive multiple templates and the extracted templates can be referred for the uniqueness and the templates with the same details can be deleted. This part of the work can be considered as future work. An interesting compliment in this work is it can be applied to any search engines. The simplicity and efficiency forms the best solution, which reduces the cost without degrading the performances.

XI. REFERENCES

[1] Yanhong Zhai, "Web Data Extraction Based on Partial Tree Alignment",ACM 2005

[2] A. Arasu and H. Garcia-Molina, "Extracting Structured Data from Web Pages," Proc. ACM SIGMOD, 2003.

[3] Development of an efficient vertex based template extraction technique for web pages (n.p.k. ganesh kumar, dr.n.k. sakthivel)2011

[4]] M. de Castro Reis, P.B. Golgher, A.S. da Silva, and A.H.F. Laender, "Automatic Web News Extraction Using Tree Edit Distance," Proc. 13th Int'l Conf. World Wide Web (WWW), 2008.

[5] H. Zhao, W. Meng, and C. Yu, "Automatic Extraction of Dynamic Record Sections from Search Engine Result Pages," Proc. 32nd Int'l Conf. Very Large Data Bases (VLDB), 2006.

[6] V. Crescenzi, P. Merialdo, and P. Missier, "Clustering Web Pages Based on Their Structure," Data and Knowledge Eng., vol. 54, pp. 279- 299, 2005.

[7] V. Crescenzi, G. Mecca, and P. Merialdo, "Roadrunner: Towards Automatic Data Extraction from Large Web Sites," Proc. 27th Int'l Conf. Very Large Data Bases (VLDB), 2001.

[8] S. Zheng, D. Wu, R. Song, and J.-R. Wen, "Joint Optimization of Wrapper Generation and Template Detection," Proc. ACM SIGKDD, 2007.

[9] Jackey Z. Yan, Chris Chu and Wai-Kei Mak "SafeChoice: A Novel Approach to Hypergraph Clustering for Wirelength-Driven Placement", IEEE Transactions On Computer-Aided Design, 2011.