A New Algorithm for Accurate Histogram Construction

Zeineb Dhouioui
Computer Science Department
ISG
41, rue de la Liberté, 2000 Le Bardo Tunisia
dhouioui.zeineb@hotmail.fr

Wissem Labbadi
Computer Science Department
ISG
41, rue de la Liberté, 2000 Le Bardo Tunisia
wissem.labbadi@isg.rnu.tn

Jalel Akaichi
Computer Science Department
ISG
41, rue de la Liberté, 2000 Le Bardo Tunisia
jalel.akaichi@isg.rnu.tn

***Abstract*—**Many commercial relational database systems use histograms to summarize data sets and also to determine the frequency distribution of attribute values. Based on this distribution, a database system estimates query result sizes within query optimization useful in effective information retrieval. Moreover, histograms are beneficial for judging whether the quality of the source is reliable or not; therefore, they enable us/ one to decide whether to keep this source in the information retrieval or remove it. Each histogram contains commonly an error which affects the accuracy of the estimation. This work surveys the state of the art on the problem of identifying optimal histograms, studies the effectiveness of these optimal histograms in limiting error propagation in the context of query optimization, and proposes a new algorithm for accurate histogram construction. As a result, we can conclude that theoretical results are confirmed in practice. In fact, the proposed histogram generates a low error.

Keywords- Optimal histograms; query result size estimation; error; query optimization; data summarization.

I. INTRODUCTION

Information retrieval is a science that studies how to respond effectively to a request by finding the appropriate information in a huge stream of data. The need to use information retrieval systems (IRS) has increased with the rapid evolution of information technology and communication and also with the proliferation of computer data and their sources. Diversity and heterogeneity of information sources require the use of IRS that must meet user expectations and needs and provide relevant information among the mass of available information in the shortest time and with reduced cost. However, in front of the fast growth of data databases have witnessed an exponential increase of stored data which make it increasingly difficult to control and effectively manage the potentially flow of information.

A straightforward way to satisfy an information need is to send the query to all sources, and to get results from each one, which are then provided to the user. However, this strategy is not efficient in front of the big number of dispersed sources. This simple method incurs unnecessary cost and an additional waiting time when sending the query to sources not containing the required information [1]. For this reason, more efficient information retrieval techniques that can extract relevant information from large scale distributed sources are needed in order to satisfy users' requirements in the shortest waiting time. The idea suggested for overcoming this problem is to associate to each source a summary which is a compact representation of its content. Then, the interestingness of a given data source with respect to the user requirements, expressed into a query, is assessed by processing this query against the summary. This operation is a simple match and doesn't need to send the query to the considered source and manipulate its big mass of data. Data summary techniques provide concise and complete representation of data and are now considered as accurate tools to handle huge databases, in particular when precise values of data are not needed.

Many commercial DBMSs [3] maintain a variety of types of histograms to summarize the contents of the database relation by approximating the distribution of values in the relation attributes and based on them estimate sizes and value distributions in query results. A histogram approximates the distributions by grouping the data values into buckets. This grouping into buckets loses information. This loss of information engenders errors in estimates based on these histograms. The resulting estimation-errors directly or transitively affect the accuracy of the resulting estimates and hence, degrade the dramatically the performance of the applications using these estimates. This effect may be devastating in the most cases. For multi-join queries that are processed as a sequence of many join operations, the transitive effect of error propagation among the intermediate results on the estimates derived for the complete query may be destructive even if the original errors are small. Motivated by the fact that inaccurate estimations can lead to wrong decisions, we propose in this paper an efficient algorithm, called CM, for accurate histogram constructions. The survey is organized as follows. Both theoretical and effective experiments are done using two datasets.

The remainder of this paper is organized as follows: Section 2 presents the basic definitions on histograms. Section 3 provides an overview of several earlier and some more recent classes of histograms that are close to optimal and effective in many estimation problems. In Section 4, we present the proposed histogram the CM Histogram. In

 ISBN: 978-1-61208-227-1

Section 5, we formulate the main problem. In Section 6, we propose HConst Algorithm for accurate histogram constructions. In Section 7, we present the result of a set of experiments that compare the existing histograms to our algorithm in term of estimation accuracy. Finally, Section 8 concludes and outlines some of the open problems in this area.

II. BACKGROUND

Histograms are widely used as a data summarization approach. They present an efficient and powerful way to capture the data distribution and also estimate the query result. Histograms are composed of a set of buckets where each bucket contains the frequency of occurrence of each attribute value histogram on attribute X of relation R divides the data distribution into buckets.

We assume the following parameters [2]:

- Domain D of X: it is the set of all possible values of X
- Value set V ⊆ D: is the set of values of D present in R
- Frequency f_i of $v_i \in V$ is the number of tuples $t \in R$

The data distribution DD of the attribute X in the relation R is the set of pairs which comprises the attribute value and its frequency: $DD_{i=1...D,\ D \le |X|} = \{(v_i, f_i) \ ... \ (v_D, f_D)\}$

TABLE I. EXAMPLE OF A DATA DISTRIBUTION

V_i	f_i
180	2
250	1
260	1
270	2
320	1
345	1
380	1
410	1
450	3
490	1
550	1

III. STATE OF THE ART

In this section, we present several earlier and other relative recent histograms, listed in the literature and considered as optimal in estimating range query result sizes.

A. Trivial Histogram

This kind of histograms [3] is simple because it is based on uniform distribution assumption.

It is composed of one single bucket where the approximate frequency is identical for all attribute values [4].

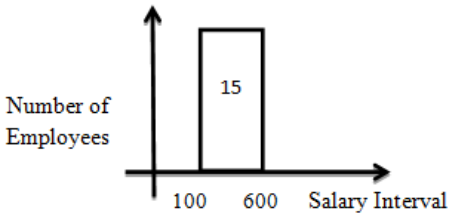

Figure 1. Data Distribution with Trivial Histogram

This type of histogram is based on the principle of uniform distribution. Therefore, the appearance of all values is equally likely; this means that each value appears in the data set a single time.

In this example, we have one single bucket in which the frequency of each value is identical to others.

Trivial histograms have usually a large error rate in query estimation; we will prove this with a selection query.

In our work, we will focus in select queries which allow us to select records according to a specific criterion. Take this example of the selection query and find the number of the employees who have a salary upper than 450.

```
Query:  Select count (*)
From employees
Where salary > 450;
```

According to the histogram, we have at worst 15 values that can be greater than 450 but actually, we have two values greater than 550 and 490.

Subsequently, the corresponding absolute error is: Eabs=|2-15|=13.This is considered a very large error rate.

B. Equi-width Histogram

The idea is to divide the data distribution into buckets. The same width is maintained in all buckets. We apply Equi-depth at the proposed dataset:

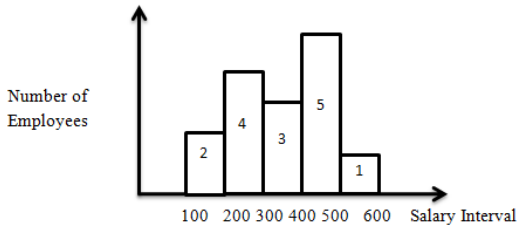

Figure 2. Data Distribution with Equi-width

The height of each bucket presents the total of the frequencies of all attribute values falling in this bucket [5].

The problem in this type of histogram lies in the precision because the error rate is large. This will be proved with the previous query. In the worst case, the maximum error rate of the selection query is half of the height of the bucket.

This case is called unlucky distribution of attribute values where the tallest bucket contains almost 100% of the tuples; then, the error rate is equal to 0.5.

 ISBN: 978-1-61208-227-1

In our example, we have 2 values superior than 450, but according to the histogram, we have 6 values: five values in the interval [400-500] and one value in the interval [500-600].Consequently the absolute error:

Eabs=|2-6|=4

We can conclude that the problem is in the height of buckets, hence the idea of creating a new type of histogram: the equi-height histogram described subsequently.

C. *Equi-depth Histogram*

In equi-depth histogram [4], called also equi-height, the sum of the frequencies in each bucket is the same.

For the construction of this histogram, we must first sort the attribute values in an ascending order to obtain a height balanced histogram. The representation of the attribute values from the previous example consists in a histogram containing seven buckets having equal heights. The threshold per bucket is equal to two as it is shown in the following figure:

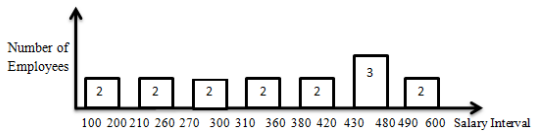

Figure 3. Data distribution with Equi-depth Histogram

The equi-height is more accurate than the equi-width as we will prove with the previous query.

According to the previous query, we can estimate the absolute error:

E_{abs}=|2-4|=2

D. *V-optimal Histogram*

V-optimal Histograms are also called Variance-optimal [4].The basic idea of this histogram is to minimize the weighted variance inside each bucket [6].

The weight here is the number of attribute values in the j^{th} bucket.

$$\sum_{j=1}^{\beta} nj\ vj \quad (1)$$

Where:

- j is the number of buckets.
- n_j is the number of entities in the j^{th} bucket.
- v_j is the variance between the values of the entities in bucket.
- β is the maximum number of buckets

We apply V-optimal histogram to our example; the corresponding absolute error is equal to:

E_{abs} =|2-5|=3

E. *MaxDiff Histogram (Maximum Difference)*

In a MaxDiff histogram, there is a bucket boundary between the adjacent values which have the maximum difference [3]. We compute [7] the difference between $f(v_{i+1})*S_{i+1}$ and $f(v_i)*S_i$.

Where:

- S_i is the spread of attribute value v_i

  $S_i = v_{i+1}-v_i$ (2)
- $f(v_i)* S_i$ is the area of v
- $f(v_i)$:frequency of v_i

We apply max-diff histogram to our example; we separate the adjacent values with a large change in the area.

TABLE II. COMPUTING THE SPREAD, AREA AND Δ AREA

Value	180	250	260	270	320	345	380	410	450	490	550
Frequency	2	1	1	2	1	1	1	1	3	1	1
Spread	70	10	10	50	25	35	35	40	40	60	-
Area	140	10	10	100	25	35	35	40	120	60	-
Δ Area	130	0	90	75	10	0	5	80	60	-	-

TABLE III. Max-diff Histogram

Bucket	Frequency
[100-180]	2
[200-260]	2
[270-300]	2
[320-400]	3
[410-460]	4
[480-600]	2

We compute the absolute error corresponding to the previous query:

E_{abs}=|2-4|=2

F. *Compressed Histogram*

In this type of histogram, we assign the n highest source values in n individual bucket and we apply the equi-heigth histogram on the rest [6].

n is the number of values that exceeds the sum of all the frequencies, Sum_F, that is divided by the number of buckets B:

$$n > \frac{SumF}{B} \quad (3)$$

$n > \frac{15}{7}$=2.14

Hence, we affect the frequencies which are upper than 2.14 to an individual bucket.

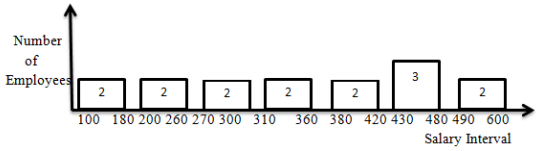

Figure 4. Data Distribution with Compressed Histogram

It looks like end-biased histogram since it distinguishes the highest value of the others, but it differs in the organization of the remaining values.It is an improvement of equi-depth.

The most frequent value is 450 which belong to the interval [430-480].The corresponding absolute error is equal to:

E_{abs} =|2-5|=3

G. Error Metrics

The authors in [8] present the error metrics; we define some concepts useful to compute the error:

- $q_{i:}$ is a query
- A_i: the real size
- A_i': the estimated size found using the histogram
- N: the number of queries.

There are many types of error metrics; the first one is the absolute error computed by the formula:

$$E_{abs} = |A_i - A_i'| \quad (4)$$

The average absolute error is the ratio between the absolute error and the number of [9].

$$AE_{abs} = \sum_{i=1}^{N} Eabs / N \quad (5)$$

And the second type is the relative error:

$$E_{rel} = E_{abs} / A_i = |A_i - A_i'| / A_i \quad (6)$$

The average relative error is the ratio between the relative error and the number of queries [9].

$$AE_{rel} = \sum_{i=1}^{N} Erel/N \quad (7)$$

We can also use average standard deviation which can be computed directly from the histogram.

We find similarly the SSE Sum of Squared Error [8].

For an interval I [i , j] the SSE is calculated by the following formula:

$$SSE([i,j]) = \sum_{k=i}^{k=j} (F[k] - AVG([i,j]))^2 \quad (8)$$

$$AVG(i,j) = \sum_{k=i}^{k} \frac{F[k]}{j-i+1} \quad (9)$$

Where:

F[k]: frequency of the element k

IV. CM HISTOGRAM

The problem of histogram construction is primordial in many tasks in databases. Therefore, many researches have been developed extensively in the past since histograms are characterized by their popularity, accuracy and simplicity in representing the data distribution efficiently.

The idea behind all histograms is to reduce the error and to reasonably consume a small space.

Many algorithms were proposed in the past; they differ in how the values are assigned to buckets.

In this work, we propose a new approach which is an algorithmic solution: Hconst (Histogram construction) Algorithm which tends to find and construct the optimal histogram: CM histogram.

This naming comes from the idea to ameliorate the existing version of compressed histogram by the principle of Maximum-difference histogram.

This approach reconciles the benefits of Max-diff histogram. We developed an experimental evaluation to underline the effectiveness and the accuracy of our algorithm and to prove that the error is lower than existing techniques.

One of the drawbacks of previous techniques in their accuracy is that the error rate is large, so we attempt to overcome this problem by introducing Hconst algorithm to construct CM histogram.

Our algorithm is applicable to minimize the error rate in query optimization.

We propose an improved version of compressed histogram; we will demonstrate that the concept of Max-diff extends to optimize the compressed histogram.

We realize the effectiveness and the advantages of Max-diff histogram to develop an approach and find a compromise between efficiency, accuracy and applicability.

Our idea is based on the principle of compressed histogram that affects the most frequent value in an individual bucket and applies equi-depth histogram on the remaining values.

The idea of CM histogram is assigning the highest frequency, i.e., the more occurring attribute value in an individual bucket. We apply Max-diff histogram to the remaining values.

According to the literature review, a lot of researches have shown with the experimental studies that Max-diff histogram is more accurate, and that is why we have the idea of applying Max-diff.

Remember that Max-diff histogram minimizes the maximum difference of adjacent source values [10].

Nigel Srikanth [11] has shown that max-diff histogram uses efficiently the Central Processing Unit CPU and memory. Kyung [12] stated that this histogram allows grouping the closest frequency since it inserts a boundary between values that have a maximum difference; so the estimation of the query size is more correct and precise.

V. PROBLEM FORMULATION

Consider [13], a relational table R which comprises n attributes $X_1 ... X_{n.}$

D: domains of attributes $X_1 ... X_{n.}$

Given a data set, find the histogram H associated with the attribute X with the smallest error:

Min Error (H)

So, our need is to find an efficient algorithm for constructing an accurate histogram.

The accuracy of the histogram relates to the accuracy of each bucket.

VI. HCONST ALGORITHM

We observe that any histogram contains an error; this error is due to the loss of information in their summary. There remains a need to find the optimal histogram; motivated by this, we propose a new technique to construct a histogram: CM histogram with a smaller error. Hconst algorithm fails to respond to this challenge.

Definition

Consider two histograms Hi and Hj which represent the distribution of a given dataset; we say that Hi is more accurate than Hj if and only if:

Error (Hi) < Error (Hj)

 ISBN: 978-1-61208-227-1

Theorem

A Max-difference bucket with a height h1 provides estimation more accurate than an Equi-depth bucket with a height h2 for all h1≤ h2.

Proof

From a set V of values with their corresponding set F of frequencies, we construct a maximum difference histogram M and an equi-depth histogram E, supposedly composed of a same number N of buckets.

Let (H_i^M)i=1 to N and (H_i^E)i=1 to N be the respective heights of the buckets (B_i^M)i=1 to N and (B_i^E)i=1 to N that compose the two histograms M and E.

Suppose that:

$H_i^E \geq H_i^M$ for a given $1 \leq i \leq N$.

To prove that estimation of the Histogram M is better than that of histogram E, it is sufficient to prove that:

Error $(B_i^E) \leq$ Error (B_i^M).

This inequality is verified using SSE metric since M, the max-diff histogram is already constructed by minimizing the variance, as this kind of histogram tends to group closer values, whereas equi-depth controls just the sum of the frequencies by bucket.

Hence $E_{abs}(M) \leq E_{abs}(E)$

We can conclude that for each value A, the estimation determined using Max-diff histogram is more accurate than equi-depth which justifies our choice of applying max-diff histogram instead of equi-depth.

For the case h1 > h2, we improve the accuracy of Max-diff by using the following algorithm:

Hconst Algorithm

Input: frequencies of each the attribute value

Output: the accurate histogram

1. Begin
2. Find (n, freqV,B)
3. maxDiff (remaining values, maxDiff histogram)

Optimization phase

4. For each Maxdiff bucket
5. If (exceptional bucket=True)
6. If H(BucketI-1) < heightMax
   BucketI-1← minVal
7. Else
8. If H(BucketI+1) < heightMax
   BucketI+1← maxVal
9. Return CM histogram
10. End;

This algorithm takes as input the different frequencies of each attribute value; later, there will be a call to the procedure Find to determine the highest frequencies; and then, there will be a call to the procedure max-Diff.

In the optimization phase, we reduce the height of the exceptional buckets.

We mean with Exceptional bucket whether the height bucket of max-diff histogram is greater than the height of equi-depth.

This phase proceeds as follows:

If the height of the previous bucket is lower than the maximal height; we migrate the minimum value in the bucket; else we migrate the maximum value to the next bucket.

We can change the location more than one value as we have not exceeded the maximal height.

As output, the result of the proposed algorithm will be an efficient and accurate histogram.

TABLE IV Time Complexity

Algorithm	Time Complexity
Procedure Find	O(N)
Procedure maxDiff	O(M)
Function Boundry	O(M)
Algorithm Hconst	O(N)

Where:

- N attribute value
- M remaining values

VII. EXPERIMENTAL RESULT

We attempt to prove that the theoretical results are confirmed in practice.

We investigated the effectiveness of the different histogram types cited above for estimating range query result sizes. The absolute errors due to the different histograms, as a function of the number of the bucket, are omputed based on a selectivity query applied on two data distribtions on the attribute salary from real database: National League Baseball Salaries for the years 2003 and 2005 to compare the performance of existing histograms; we assign the same number of buckets to different histograms.

The typical behavior of the histogram errors for the selection query applied respectively on the dataset for the year 2005 and 2003 are illustrated respectively in figure 5 and 6, with the number of bucket indicated on the x-axis and the absolute error indicated on the y- axis

As a visualization tool, we will use MATLAB, an example of a selection query on the National League Baseball Salaries dataset of the year 2005:

Select count (*)
Where salary =1000000;

The real frequency of the value 1000000 = 5

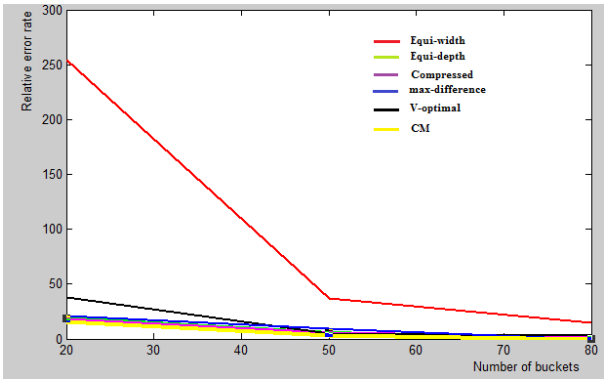

Figure 5. The Absolute Error of the first Dataset

We apply the same query on the second dataset corresponding to the year 2003 to better show the accuracy of our technique:

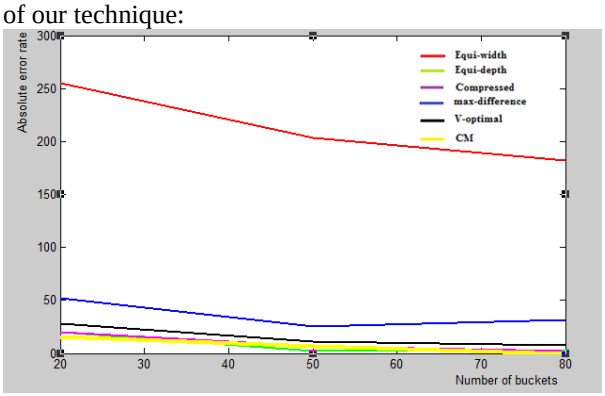

Figure 6. The Absolute Error of the second Dataset

In those plots, the number of buckets is varied. The error generated is proportional to the number of buckets. As shown in the two figures, the accuracy can be reached when increaing the number of buckets for all histogram types and the compressed, max-diff and v-optimal histograms are significantly better than the others that they show the least errir for different number of buckets. Moreover, the equi-width histogram exhibits the worst accuracy.

We show the efficiency of different traditional histograms, namely equi-width, equi-depth, compressed, V-opt, Max-diff and CM histogram obtained using our algorithm.

The results from the experiments show that the absolute error generated by our method is lower than the absolute error from existing histograms. This is a consequence of the fact that attribute values in our histogram are closer.

Those results do not only confirm our theoretical results presented in the previous chapter, but also confirm that the accuracy of our method is superior to that of previous histograms.

VIII. CONCLUSION

The use of histograms is widespread especially in approximating frequency distributions in data bases thanks to their simplicity and accuracy.

In our work, we studied and discussed various kinds of existing histograms; in addition to that, we have introduced a new technique for histogram constructing using an algorithm called Hconst.

We have also proposed a theorem to justify our technique.

Furthermore, we can deduce from the experimental comparisons that the histogram reduces the error; accordingly, we can confirm, and based on those experiments on a real database, that the quality of the histogram improves.

The identification of the optimal histogram remains an open field. As several new research opportunities appear, we will try to identify optimal histograms for different types of queries such as joins and non-equality joins, to limit not only the absolute error but also other metrics of error, to determine the appropriate number of buckets to build the optimal histogram and to find the histogram that can handle uncertain data.

And finally, we want to treat the problem of data stream which is the transmission of the flow of data that changes over time. Existing database systems do not process data streams efficiently; and this makes this area a popular search field [13].

REFERENCES

[1] C. Yu, G. Philip, and W. Meng, “Distributed top-N query processing with possibly uncooperative local systems,” In Proc, 29th VLDB Conference, 2003, pp 117-128.

[2] K. Chakrabarti, G. Minos, R. Rajeev, and S. Kyuseok, “Approximate Query processing using wavelets,” The VLDB Journal Vol. 10 Issue 2-3, 2001.

[3] V. Poosala, and Y. Ioannidis, “Improved Histograms for selectivity estimation of range predicates,” In Proc, 23rd VLDB Conference, 1997.

[4] Y. Ioannidis, “Query optimization,” ACM Computing Surveys, symposium issue on the 50th Anniversary of ACM, Vol. 28, 1996, pp. 121-123.

[5] S. Joseph, “Adaptive histogram algorithms for approximating frequency queries in dynamic data streams, “ world comp, 2011.

[6] V. Poosala, Y. Ioannidis, P. Haas, and E. Shekita, “Improved histograms for selectivity estimation of range predicates,” International ACM SIGMOD Conference, pp. 294-305, 1996.

[7] Y. Liu, “Data preprocessing,” Department of Biomedical, Industrial and Human Factors Engineering Wright State University, 2010.

[8] V. Jagadish, J. Hui, C. Beng, and T. Kian-Lee, “Global optimization of histograms,” ACM SIGMOD Vol. 30 Issue 2, 2001.

[9] X. Lin, and Q. Zhang, “Error minimization for approximate computation of range aggregates,” Proceedings of the Eighth International Conference on Database Systems for Advanced Applications IEEE Computer Society, 2003.

[10] B. Zina, J. Liu, B. Omran , L. Huian , M. Jesse, B. Chavali , and O. Robert, “Use and Maintenance of Histograms for Large Scientific Database Access Planning: A Case Study of a Pharmaceutical Data,” Repository Journal of Intelligent Information Systems, 2004.

 ISBN: 978-1-61208-227-1

[11] E. Nigel, and A. Srikanth, "Query planning using a maxdiff histogram, " Microsoft Corporation, 2000.

[12] H. Kyoung, "Query Size Estimation through Sampling," phd thesis, North Carolina State University, 2005.
(http://repository.lib.ncsu.edu/ir/handle/1840.16/1274)

[13] V. Jagadish, V. Poosala, K. Nick, S. Ken, S. Muthukrishnan, and S. Torsten, "Optimal Histograms with Quality Guarantees," VLDB Proceedings, 1998.

[14] P. Pawluk, "Stream Databases," PhD thesis, York University Department of Computer Science and Engineering, 2006.
(http://www.bth.se/fou/cuppsats.nsf/all/e1571a10dc340e51c12571bc005ddc43/$file/prpa05-master_thesis.pdf).

 ISBN: 978-1-61208-227-1