

# RAVI

## Image and Table Captions Mapping and Placement in HTML

Supervised by

Prof. M. Balakrishnan & Prof. Volker Sorge

Submitted by

Mini Project COD310

Hire Vikram Umaji (2015CS10227)

&

Minor Project COD891

Amar Agnihotri (2012CS50277 )



**Department Of Computer Science and Engineering  
INDIAN INSTITUTE OF TECHNOLOGY DELHI**

## ACKNOWLEDGEMENTS

We would like to express our deepest gratitude to our advisor, **Dr. M. BalaKrishnan** for his exemplary guidance, monitoring and constant encouragement throughout the course of this project.

A very special thanks to **Volker Sorge**, **Himanshu Garg** and **Neha Jadhav** for their guidance and discussions throughout the project.

We would also like to thank our parents, brother and friends for their moral support and continuous encouragement.

# Contents

<u>Chapter 1</u>		
<u>INTRODUCTION</u>	.....	<u>Page 4</u>
<u>Objective</u>	.....	<u>Page 4</u>
<u>Prior Work</u>	.....	<u>Page 5</u>
<u>Chapter 2</u>		
<u>ASSUMPTIONS</u>	.....	<u>Page 6</u>
<u>Criteria</u>	.....	<u>Page 6</u>
<u>Chapter 3</u>		
<u>ALGORITHM</u>	.....	<u>Page 7</u>
<u>Caption and object mapping</u>	.....	<u>Page 7</u>
<u>Algorithm Input</u>	.....	<u>Page 7</u>
<u>Algorithm Pseudocode</u>	.....	<u>Page 7</u>
<u>Algorithm Output</u>	.....	<u>Page 8</u>
<u>Multimapped glyphs correction</u>	.....	<u>Page 8</u>
<u>Algorithm Input</u>	.....	<u>Page 8</u>
<u>Algorithm pseudocode</u>	.....	<u>Page 8</u>
<u>Algorithm output</u>	.....	<u>Page 8</u>
<u>Placement in HTML Output</u>	.....	<u>Page 9</u>
<u>Algorithm Input</u>	.....	<u>Page 9</u>
<u>Algorithm pseudocode</u>	.....	<u>Page 9</u>
<u>Algorithm output</u>	.....	<u>Page 9</u>
<u>Explanation</u>	.....	<u>Page 10</u>
<u>Chapter 4</u>		
<u>RESULTS</u>	.....	<u>Page 11</u>
<u>Dataset:</u>	.....	<u>Page 11</u>
<u>Thresholds</u>	.....	<u>Page 11</u>
<u>Chapter 5</u>		
<u>FAILURE ANALYSIS</u>	.....	<u>Page 13</u>
<u>Chapter 6</u>		
<u>Integration with RAVI</u>	.....	<u>Page 30</u>
<u>Chapter 7</u>		
<u>Conclusion and Future Work</u>	.....	<u>Page 32</u>
<u>References</u>	.....	<u>Page 32</u>

# Chapter 1

## INTRODUCTION

This document provides an analysis of caption detection and their placement in output HTML which is a mini project of Project RAVI (Reading Assistant for Visually Impaired) which focuses on making digital textbooks accessible to readers by analyzing the pdf data into pixel cluster and classifying into texts as foreground and images as background, finally as html elements to speech for visually impaired.

Our task is to identify captions in foreground text, identify images and tables in background and finally make them as html elements and place them in the correct relative position in the HTML output. The motivation behind this work comes from making images and tables accessible to Visually Impaired Peoples.

## Objective

The objective of our project was to detect the captions inside the document and mapping them to their respective Images/Tables by correctly detecting the Images/Tables. Finally after detecting the position of captions and their respective Objects in the final Html Output needed to be done.

## Prior Work

Prior work has classified text in foreground as lines, headings, list lines, paragraphs. Prior work also clustered together background pixels into background glyphs.

No prior work on Caption detection and Image detection was done. Table Detection was done but it was not able to detect all tables correctly. Mainly Tables with only row lines and no column lines were not detected. Similarly tables with data inside image Glyphs were also not detected.

We have utilised the lines formed using linify() function in the prior work and glyphs formed by backgroundAnalysis() function. Similarly the Caption formation utilised the function getTextBoxes() which groups the lines together to form complete paragraph structure.

# Chapter 2

## ASSUMPTIONS

The Following Assumptions were made in our Algorithm:

- getTextBoxes Function is giving all paragraphs, each paragraph has no excess or lack of lines.
- Images are present at the Left/Right/Top/Bottom position respectively of Figure Caption Bounding Boxes and no other data comes between them except paragraphs starting with Note keys.
- Tables are present at Top/Bottom position respectively of Table Caption Bounding Boxes and no other data comes between them except paragraphs starting with Note keys.
- Every True Caption has a Glyph in Background
- Caption and Note Start with some key Identifiers, in our code. The default value is [figure,fig.,table,tab.] for caption and [note] for notes of images and tables. (Default values are for English documents)

## Criteria

Our project goal is to put the caption element and its image/table element into output HTML at their correct position.

# Chapter 3

## ALGORITHM

### 1. Caption and object mapping

#### Algorithm Input

For Image Analysis:

- Get Image Glyphs from `getbackgroundglyphs()` after `d.BackGroundAnalysis()` is done
- Get line-Glyphs from `lines()` after `d.linify()` is done

For Caption Detection:

- Get Paragraphs from Foreground as `getTextboxes()` after `d.linify()` is done

#### Algorithm Pseudocode

```
--Search for Captions in Paragraphs Using Object Identifier keys as Starting Word  
-- Classify Caption Type into Table/Image  
-- First map image with relative top glyphs and table with relative bottom glyphs as object  
-- if there is atleast a caption with no mapped glyph:  
    -- Find all possible relative 8 position combination glyphs as object of Table and  
Image  
-- Select maximum number of caption mapping in order of priority.  
--Search For Paragraphs starting with Note keys:  
    -- If they are closer to Image/Table than its Caption then include them to the  
Object  
-- Map the Line Glyphs Coming inside/near the Object within line threshold to the Object  
--Remove Captions which have not been mapped to any Object
```

## Algorithm Output

- Image caption and table caption as fig caption element
- Images and table glyphs mapped to their caption as canvas stored in img element

## 2. Multi Mapped glyphs correction

## Algorithm Input

- Get all Caption and its object from getCaptions() after d.Object\_Map() is done

## Algorithm pseudocode

```
--Search for Glyphs which are mapped to more than one object
  -- Assign this glyph to the object which have nearest caption.
  -- Remove this glyph from other caption's mapped object
```

## Algorithm output

- No background glyphs are mapped to more than one object

### 3. Placement in HTML Output

#### Algorithm Input

- All lines of the document as unmatched

#### Algorithm pseudocode

```
--Identify line of unmatched as 1st line of any caption
    -- Add the caption element if all above lines in unmatched are assigned as HTML
elements
    -- Add its mapped object relative to caption as image element in HTML output
    -- Remove this caption lines containing paragraph/list lines
    -- Remove this caption lines from unmatched
```

#### Algorithm output

- All caption and its image/table is added in relative correct position in the output HTML

# Explanation

We have used page pixels as a 2D array and assign different values to denote coloring of page for lines, captions and background glyphs.

ID	$n = 200$			$n = 1000$		
	SAS (Preprint)	USVT (Chatterjee)	SBA (Airoldi et al., 2013)	SAS (Preprint)	USVT (Chatterjee)	SBA (Airoldi et al., 2013)
1	8.39e-04 ± 6.34e-05	1.96e-03 ± 1.86e-04	2.77e-03 ± 1.66e-04	9.36e-03 ± 5.82e-08	3.86e-03 ± 1.76e-05	9.16e-03 ± 1.76e-05
2	8.92e-04 ± 6.81e-05	2.18e-03 ± 1.95e-04	2.36e-03 ± 1.97e-04	7.12e-03 ± 5.92e-08	4.46e-03 ± 1.54e-05	1.39e-03 ± 3.99e-05
3	8.46e-04 ± 6.81e-05	2.18e-03 ± 1.95e-04	2.36e-03 ± 1.97e-04	7.12e-03 ± 5.92e-08	4.46e-03 ± 1.54e-05	1.39e-03 ± 3.99e-05
4	8.46e-04 ± 5.30e-05	3.51e-03 ± 1.93e-04	2.77e-03 ± 1.69e-04	7.82e-03 ± 5.75e-08	8.83e-03 ± 2.47e-05	1.45e-03 ± 2.63e-05
5	9.74e-05 ± 2.76e-05	3.15e-03 ± 1.76e-05	3.73e-03 ± 3.71e-05	1.09e-05 ± 1.66e-08	8.49e-05 ± 7.33e-06	1.66e-03 ± 3.45e-05
6	4.29e-02 ± 9.27e-05	8.91e-02 ± 1.21e-03	4.77e-02 ± 1.23e-04	4.16e-02 ± 9.59e-08	8.42e-02 ± 6.176e-04	4.22e-02 ± 1.62e-05
7	4.11e-04 ± 7.36e-05	2.46e-03 ± 1.77e-04	2.77e-03 ± 2.06e-04	4.46e-05 ± 7.47e-08	8.76e-04 ± 1.31e-05	1.22e-04 ± 3.68e-05
8	8.36e-04 ± 7.36e-05	2.46e-03 ± 1.77e-04	2.77e-03 ± 2.06e-04	4.46e-05 ± 7.47e-08	8.76e-04 ± 1.31e-05	1.22e-04 ± 3.68e-05
9	8.56e-04 ± 7.36e-05	2.87e-03 ± 2.25e-04	3.96e-03 ± 3.22e-04	1.02e-04 ± 5.15e-08	1.26e-03 ± 3.31e-05	1.11e-04 ± 3.44e-05
10	7.75e-04 ± 1.04e-04	4.74e-03 ± 6.25e-04	1.33e-03 ± 1.23e-04	1.37e-04 ± 1.02e-08	2.24e-03 ± 1.36e-05	7.36e-04 ± 1.67e-05
Average	4.43e-03 ± 7.45e-05	3.79e-02 ± 4.61e-04	8.71e-03 ± 1.99e-04	4.72e-03 ± 5.74e-08	9.38e-03 ± 2.31e-05	9.22e-03 ± 2.06e-05

Table 2. Mean squared error (average ± std. dev.) comparisons between the SAS algorithm, the USVT algorithm (Chatterjee), and the SBA algorithm (Airoldi et al., 2013). MSE is averaged over 50 independent trials.

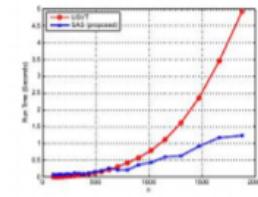


Figure 4: Run time comparison between USVT (Chatterjee) and the SAS algorithm (averaged over 10 graphons listed in Table 1).

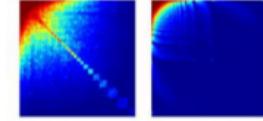


Figure 5: Estimated graphons for real networks. Left: Collaboration network of  $n = 1.8 \times 10^4$  nodes (ca-AstroPh). Right: whandomtom network of  $n = 7.5 \times 10^4$  nodes (Epinions.com).

graphs consisting of  $7.5 \times 10^4$  nodes and  $5.1 \times 10^5$  edges. For both networks, we randomly permute the rows and columns to simulate the raw data scenario where nodes are initially unordered.

Figure 5 shows the results of the SAS algorithm. For the

ca-AstroPh network, the graphon shows close collaborations among a group of people concentrated around the top left corner of the graphon. It also shows a number of small communities along the diagonal. For the soc-Epinions1 network, the graphon indicates that there are some influential nodes which consistently interact among themselves. These can be seen from the repeated patterns of the graphon.

We remark that for the ca-AstroPh network ( $n = 1.8 \times 10^4$ ) and the soc-Epinions1 network ( $n = 7.5 \times 10^4$ ), the estimations are completed in 20 seconds and 170 seconds, respectively, on a PC using an unoptimized MATLAB code. This provides a strong indication of the scalability of the SAS algorithm to larger networks.

## 6. Concluding remarks

The Sorting-And-Smoothing (SAS) algorithm is a consistent and efficient graphon estimation algorithm. The SAS algorithm consists of two steps. In the first step, the observed graph is rearranged so that the degrees are monotonically increasing. In the second step, a histogram estimation and a total variation minimization is applied to estimate a smooth surface that best fits the observed data. The SAS algorithm is evaluated on both simulation data and real network data. Our simulation results indicate that the SAS algorithm outperforms the universal singular value thresholding algorithm and the stochastic blockmodel approximation algorithm. On large-scale real networks, the SAS algorithm returns consistent graphon estimates.

**Code.** Available at: <https://github.com/iroddilab/SAS>

**Acknowledgments.** The authors thank J. J. Yang and C. Q. Han for useful discussions. SHC is partially supported by a Croucher Foundation Postdoctoral Research Fellowship. EMA is partially supported by NSF CAREER award IIS-1149662, AROMURI award W911NF-11-1-0036, and an Alfred P. Sloan Research Fellowship.



In the above figure, captions are shown in blue box, lines pixels are black. So, we are mapping the caption with its possible position of objects up/down/left/right in white colored pixels region.

# Chapter 4

## RESULTS

### Dataset:

About 66 documents of samples folder were utilised in the analysis.

Total Images	Partial Matching	Extra Matching	Correct Matching	Mis/No Matching
440	2	5	418	15
%	0.45%	1.13%	95%	3.40%

### Thresholds

There are two thresholds we have analysed for english documents. One is for background object glyphs and other is for foreground line glyphs. We have observed that there are some background object glyphs outside of the caption's four surrounding areas (as shown in figure as an example). In this example, the object is at the top of the caption. But there are some object glyphs on the top left which are non-overlapping with the resulting object box. So, we need a threshold condition

to include the leftover glyphs. We defined horizontal and vertical threshold as distance between leftover glyphs and object box horizontally and vertically.

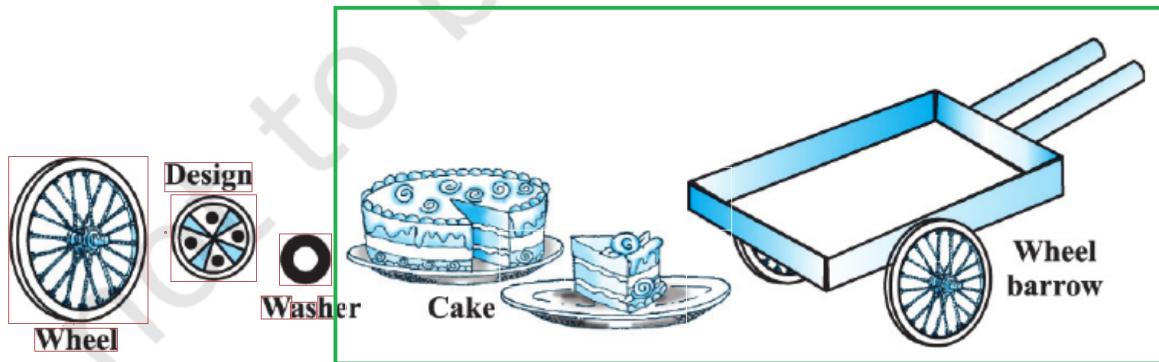


Fig. 12.1

Through objects analysis, we found that  $0.026 * \text{Width of the page}$  and  $0.011 * \text{Height of the page}$  includes all background glyphs of 95% objects.

We also observed that there are lines in some objects included in the foreground. So, we have used the `line_threshold` condition to include lines in the object box. For these lines, we have found 0.4 as optimum.

# Chapter 5

## FAILURE ANALYSIS

The algorithm failed in mainly below five cases:

1. The header or footer decorative lines are included inside the image as they are coming close to the object bounding box as shown in Figure 1.1 and Figure 1.2.

Example File:

[/samples/Test\\_bench/RadicalSignDocuments/jemh103/jemh103.html](#)

We observe in Fig. 3.4, that the lines do not intersect anywhere, i.e., they are parallel.

So, we have seen several situations which can be represented by a pair of linear equations. We have seen their algebraic and geometric representations. In the next few sections, we will discuss how these representations can be used to look for solutions of the pair of linear equations.

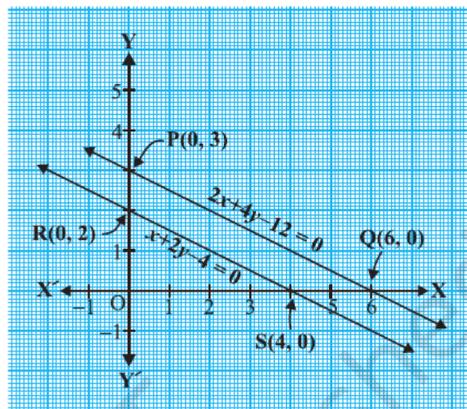


Fig. 3.4

### EXERCISE 3.1

- Aftab tells his daughter, "Seven years ago, I was seven times as old as you were then. Also, three years from now, I shall be three times as old as you will be." (Isn't this interesting?) Represent this situation algebraically and graphically.
- The coach of a cricket team buys 3 bats and 6 balls for ₹ 3900. Later, she buys another bat and 3 more balls of the same kind for ₹ 1300. Represent this situation algebraically and geometrically.
- The cost of 2 kg of apples and 1 kg of grapes on a day was found to be ₹ 160. After a month, the cost of 4 kg of apples and 2 kg of grapes is ₹ 300. Represent the situation algebraically and geometrically.

### 3.3 Graphical Method of Solution of a Pair of Linear Equations

In the previous section, you have seen how we can graphically represent a pair of linear equations as two lines. You have also seen that the lines may intersect, or may be parallel, or may coincide. Can we solve them in each case? And if so, how? We shall try and answer these questions from the geometrical point of view in this section.

Let us look at the earlier examples one by one.

- In the situation of Example 1, find out how many rides on the Giant Wheel Akhila had, and how many times she played Hoopla.

In Fig. 3.2, you noted that the equations representing the situation are geometrically shown by two lines intersecting at the point (4, 2). Therefore, the

We observe in Fig. 3.4, that the lines do not intersect anywhere, i.e., they are parallel.

So, we have seen several situations which can be represented by a pair of linear equations. We have seen their algebraic and geometric representations. In the next few sections, we will discuss how these representations can be used to look for solutions of the pair of linear equations.

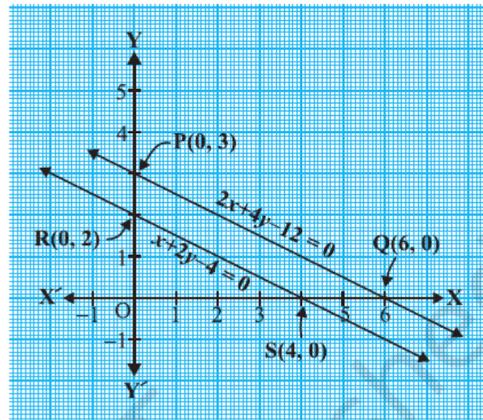


Fig. 3.4

### EXERCISE 3.1

- Aftab tells his daughter, "Seven years ago, I was seven times as old as you were then. Also, three years from now, I shall be three times as old as you will be." (Isn't this interesting?) Represent this situation algebraically and graphically.
- The coach of a cricket team buys 3 bats and 6 balls for ₹ 3900. Later, she buys another bat and 3 more balls of the same kind for ₹ 1300. Represent this situation algebraically and geometrically.
- The cost of 2 kg of apples and 1kg of grapes on a day was found to be ₹ 160. After a month, the cost of 4 kg of apples and 2 kg of grapes is ₹ 300. Represent the situation algebraically and geometrically.

### 3.3 Graphical Method of Solution of a Pair of Linear Equations

In the previous section, you have seen how we can graphically represent a pair of linear equations as two lines. You have also seen that the lines may intersect, or may be parallel, or may coincide. Can we solve them in each case? And if so, how? We shall try and answer these questions from the geometrical point of view in this section.

Let us look at the earlier examples one by one.

- In the situation of Example 1, find out how many rides on the Giant Wheel Akhila had, and how many times she played Hoopla.

In Fig. 3.2, you noted that the equations representing the situation are geometrically shown by two lines intersecting at the point (4, 2). Therefore, the

Figure 1.2 Caption and Image Found

2. Tables were not detected properly due to following reasons:

- a. Table data was coming as Image Glyphs in Background as shown in Figure 2.1 a) and Figure 2.1 b).

Example File:

[/samples/Test\\_bench/RadicalSignDocuments/Energy-Consumption-Based/Energy-Consumption-Based.html](#)

The transceiver “on” duration and the required SNR per bit depend on the modulation scheme used. In this paper, we consider  $M$ -ary phase-shift keying (MPSK),  $M$ -ary quadrature amplitude modulation (MQAM), and  $M$ -ary frequency-shift keying (MFSK) modulation schemes.

For MPSK,  $T_{\text{on}}$  and  $\text{SNR}_{\text{coded}}$  are expressed as in [10]

$$T_{\text{on}} = \frac{L}{bB} \quad (9)$$

$$\text{SNR}_{\text{coded}} = \begin{cases} \frac{N}{K} (\text{erfc}^{-1}(2p_s))^2, & b = 1 \\ \frac{N}{K} \frac{(\text{erfc}^{-1}(p_s))^2}{b(\sin \frac{\pi}{2^b})^2}, & \text{otherwise} \end{cases} \quad (10)$$

where  $p_s$  is the channel symbol error probability corresponding to the desired coded symbol error probability or coded bit error probability and computed while computing the coding gain for ECC.

For MQAM, the expressions are again derived from Proakis [10]

$$T_{\text{on}} = \frac{L}{2bB} \quad (11)$$

$$\text{SNR}_{\text{coded}} = \frac{N}{K} \frac{2(2^b - 1)}{3b} \left( \text{erfc}^{-1} \left( \frac{p_s}{2 \left( 1 - \frac{1}{\sqrt{(2^b)}} \right)} \right) \right)^2. \quad (12)$$

For MFSK,  $T_{\text{on}}$  is [10]

$$T_{\text{on}} = \frac{2^b L}{bB} \quad (13)$$

and  $\text{SNR}_{\text{coded}}$  is upper bounded as

$$\text{SNR}_{\text{coded}} \leq \frac{2N}{bK} \left( \text{erfc}^{-1} \left( \frac{2p_s}{2^b - 1} \right) \right)^2. \quad (14)$$

### B. Computation Energy Model (CompEM)

To estimate the computation energy, we are using Sim-Panalyzer [8], which is a cycle-accurate power simulator for the ARM instruction set architecture. Specifically, it simulates the StrongArm SA-1100 processor.

The power model of the simulator comprises several components, which models distinct parts of a processor: cache power models, datapath and execution unit power models, clock tree power models, and I/O power models. Sim-Panalyzer computes the energy dissipation in a program, based on counting the number of transitions in these parts of the processor. To make CompEM closer to a real sensor node, we have configured the simulator to include the external memory for data and instruction. The simulator takes the encoding (or decoding) function, written in C, and after compiling it for the target architecture, it estimates the overall energy spent by the processor in executing these functions. The total computation energy per bit is

$$E_{\text{comp}} = \frac{E_{\text{enc}} + E_{\text{dec}}}{L} \quad (15)$$

where  $E_{\text{enc}}$  and  $E_{\text{dec}}$  are the encoder and decoder computation energies, respectively.

TABLE I  
PARAMETERS FOR SIMULATIONS

$P_{FS}$	13.7 mW	$\alpha$	1.9
$P_{INA}$	0.55 mW	$G_r, G_t$	1
$P_{BWF}$	6.12 mW	$n$	4
$P_{IFA}$	0.2 mW	$N_0$	$4 \times 10^{-21}$ W/Hz
$P_{LPF}$	0.29 mW	$N$	10 dB
$P_{ADC}$	4.1 mW	$P_c$	$10^{-5}$
$P_{DAC}$	55 mW	$B$	1MHz
$T$	100 ms	$f_c$	2.4 GHz

### C. Node Energy

For the coded system, the node energy per bit  $E_{\text{node\_coded}}$  is the summation of radio energy and computation energy. Using (8) and (15)

$$E_{\text{node\_coded}} = \frac{(1 + \alpha)P_{\text{sig}}T_{\text{on}} + P_{\text{ckt}}T_{\text{on}}\frac{N}{K} + LE_{\text{comp}}\frac{N}{K}}{L}. \quad (16)$$

### IV. EXPERIMENTAL SETUP

Using our framework, we have explored and analyzed the energy consumption of a sensor node with and without using ECC and with various modulation schemes. In this paper, we have considered various configurations of Hamming code, Reed-Solomon (RS) code, and CC.

Decoding of coded data may be done in two ways in a sensor network. First, the coded data are transmitted from the node to the base station, and they are decoded at the base station. In such a type of network, the nodes are simply collecting data, and data analysis and decision making are done at the base station. As the base station does not have any energy constraint, at the node, we consider only encoding energy overheads. Second, the encoded data are decoded at another node. In such cases, the nodes are not only collecting data but also taking local decisions. In this situation, we have to consider encoding, as well as decoding, energy overheads. Therefore, we have considered the following two types of nodes:

- Type 1) in which decoding is not done at the node but at the non-energy-constrained end;
- Type 2) in which decoding is also done at the battery-powered sensor node.

Radio energy  $E_{\text{radio}}$  is calculated using (8), with parameters being shown in Table I. Sim-Panalyzer's energy model and the circuit component power [11]–[14] listed in Table I are both based on 0.18- $\mu\text{m}$  technology. Sim-Panalyzer is configured for a node with a StrongArm SA-1100 processor operating at 200 MHz, 8-kB data cache, 16-kB instruction cache, and 1-MB RAM. MPSK and MQAM modulations require a linear PA, so a class-AB PA is used [15]. For MFSK, linearity is not a constraint, so a class-C amplifier with drain efficiency of 75% is used. The coding gains for Hamming and RS codes are calculated for hard-decision decoding, whereas for CCs, the coding gain is taken for soft-decision Viterbi decoding [16].

### V. RESULTS

In this section, we present the exploration results for the node energy for a variety of ECCs and modulation schemes. In a

Figure 2.1 a) Original Document Page with table

$\bar{P}_{FS}$	13.7 mW	$\alpha$	1.9
$P_{LNA}$	0.55 mW	$G_r, G_t$	1
$P_{BPF}$	6.12 mW	$n$	4
$P_{IFA}$	0.2 mW	$N_0$	$4 \times 10^{-21} \text{ W/Hz}$
$P_{LPF}$	0.29 mW	$NF$	10 dB
$P_{ADC}$	4.1 mW	$P_e$	$10^{-5}$
$P_{DAC}$	55 mW	$B$	1MHz
$T$	100 ms	$f_c$	2.4 GHz

Figure 2.1 b) Background Document Page with table Data as Image glyphs

b. Tables with only row lines in background were not detected as shown in Figure 2.2 a) and Figure 2.2 b).

Example File:

[/samples/Test\\_bench/RadicalSignDocuments/RP34\\_ResearchPaper/RP34\\_ResearchPaper.html](#)

**Table 1. Summary of Notation**

Symbol	Meaning
$k$	The number of loci that influence a quantitative trait
$\ell$	The ploidy of the individuals being considered
$M$	An individual's population membership; takes values A and B
$L_{ij}$	An individual's allelic type at the $j$ th allele at the $i$ th locus; takes values 0 and 1
$p_i$	The frequency of the 1 allele at locus $i$ in population A (Eq. 1)
$q_i$	The frequency of the 1 allele at locus $i$ in population B (Eq. 1)
$\bar{p}$	The mean frequency of the 1 allele across loci in population A (Eq. 2)
$\bar{q}$	The mean frequency of the 1 allele across loci in population B (Eq. 2)
$s_p^2$	The variance across loci in the frequency of the 1 allele in population A (Eq. 3)
$s_q^2$	The variance across loci in the frequency of the 1 allele in population B (Eq. 3)
$V_j$	An indicator for whether an individual's $j$ th allele at the $i$ th locus is a + allele (Eq. 4)
$T$	An individual's value for a quantitative trait (Eq. 4)
$X_i$	An indicator for whether the 0 or the 1 allele is also the + allele at locus $i$ (Eq. 5)
$S$	The number of 1 alleles an individual carries (Eq. 6)
$\delta_i$	The difference between populations in the frequency of the 1 allele at locus $i$ (Eq. 8)
$\bar{\delta}$	The mean allele-frequency difference between populations, or $\bar{q} - \bar{p}$ (Eq. 9)
$\bar{\delta}^2$	The mean squared difference between populations in the frequency of the 1 allele (Eq. 10)
$F_{ST}^k$	The ratio of the mean (across $k$ loci) within-population variance in an allelic indicator variable to the mean (across $k$ loci) total variance in an allelic indicator variable (Eq. 14)
$D_L^2$	A function of $F_{ST}^k$ that bears the same relationship to $F_{ST}^k$ as the square of Cohen's $d$ does to $r^2$ (Eq. 15)
$F_{ST(\ell)}$	A generalization of $F_{ST}^k$ for the sum of $\ell$ independent allelic indicator variables (Eq. 16)
$D_{L(\ell)}^2$	A function of $F_{ST(\ell)}^k$ that bears the same relationship to $F_{ST(\ell)}^k$ as the square of Cohen's $d$ does to $r^2$ (Eq. 17)
$U_i$	A transformation of the $X_i$ ; if $X_i = 1$ , then $U_i = 1$ ; if $X_i = 0$ , then $U_i = -1$ (Eq. 18)
$D_t$	The standardized difference between populations A and B on the trait (Eqs. 25, 34)
$\rho_t^2$	The proportion of the total variance in the trait attributable to between-population difference on the trait (Eqs. 26, 27, 41)
$Q_{ST}$	A quantitative-trait analogue of $F_{ST}^k$ ; if $\ell = 1$ (haploid organisms), then $Q_{ST} = \rho_t^2$ (Eq. 28)
$W_S$	A quantity that equals 1 if an individual is classified into the wrong population on the basis of its value of $S$ , and that equals 0 otherwise (Eq. 55)
$W_T$	A quantity that equals 1 if an individual is classified into the wrong population on the basis of its value of $T$ , and that equals 0 otherwise (Eq. 56)

haploids. Here, we extend our earlier model to allow arbitrary allele-frequency distributions and arbitrary ploidy. Our results provide another way of establishing the result that between-group differentiation on a neutral trait mirrors between-group genetic differentiation at a neutral locus, one that makes minimal evolutionary assumptions. In Section 2, we describe our extended model. In Section 3, we define several measurements of between-group genetic differentiation. In Sections 4 and 5, we describe properties of two statistics that summarize the degree of difference between two populations on a quantitative trait. In Section 6, we introduce two simplifying assumptions that allow us to analyze the problem of inferring

an individual's population of origin using either genetic or phenotypic information. Finally, we discuss the results with respect to the interpretation of population differences in disease phenotypes. Figure 1 provides a conceptual map of the structure of the article.

## 2. Preliminaries

### 2.1 Model

Our extended model is parallel to our previously reported model (Edge and Rosenberg 2015) and is similar to models used by Risch et al. (2002), Edwards (2003), and especially Tal (2012) to

This content downloaded from  
 103.27.10.102 on Wed, 26 Jun 2019 04:55:54 UTC  
 All use subject to https://about.jstor.org/terms

Figure 2.2 a) Original Document Page with table

**Table 1. Summary of Notation**

Symbol	Meaning
$k$	The number of loci that influence a quantitative trait
$\ell$	The ploidy of the individuals being considered
$M$	An individual's population membership; takes values A and B
$L_{ij}$	An individual's allelic type at the $j$ th allele at the $i$ th locus; takes values 0 and 1
$p_i$	The frequency of the 1 allele at locus $i$ in population A (Eq. 1)
$q_i$	The frequency of the 1 allele at locus $i$ in population B (Eq. 1)
$\bar{p}$	The mean frequency of the 1 allele across loci in population A (Eq. 2)
$\bar{q}$	The mean frequency of the 1 allele across loci in population B (Eq. 2)
$s_p^2$	The variance across loci in the frequency of the 1 allele in population A (Eq. 3)
$s_q^2$	The variance across loci in the frequency of the 1 allele in population B (Eq. 3)
$V_j$	An indicator for whether an individual's $j$ th allele at the $i$ th locus is a + allele (Eq. 4)
$T$	An individual's value for a quantitative trait (Eq. 4)
$X_i$	An indicator for whether the 0 or the 1 allele is also the + allele at locus $i$ (Eq. 5)
$S$	The number of 1 alleles an individual carries (Eq. 6)
$\delta_i$	The difference between populations in the frequency of the 1 allele at locus $i$ (Eq. 8)
$\bar{\delta}$	The mean allele-frequency difference between populations, or $\bar{q} - \bar{p}$ (Eq. 9)
$\bar{\delta}^2$	The mean squared difference between populations in the frequency of the 1 allele (Eq. 10)
$F_{ST}^k$	The ratio of the mean (across $k$ loci) within-population variance in an allelic indicator variable to the mean (across $k$ loci) total variance in an allelic indicator variable (Eq. 14)
$D_L^2$	A function of $F_{ST}^k$ that bears the same relationship to $F_{ST}^k$ as the square of Cohen's $d$ does to $r^2$ (Eq. 15)
$F_{ST(\ell)}$	A generalization of $F_{ST}^k$ for the sum of $\ell$ independent allelic indicator variables (Eq. 16)
$D_{L(\ell)}^2$	A function of $F_{ST(\ell)}^k$ that bears the same relationship to $F_{ST(\ell)}^k$ as the square of Cohen's $d$ does to $r^2$ (Eq. 17)
$U_i$	A transformation of the $X_i$ ; if $X_i = 1$ , then $U_i = 1$ ; if $X_i = 0$ , then $U_i = -1$ (Eq. 18)
$D_t$	The standardized difference between populations A and B on the trait (Eqs. 25, 34)
$\rho_t^2$	The proportion of the total variance in the trait attributable to between-population difference on the trait (Eqs. 26, 27, 41)
$Q_{ST}$	A quantitative-trait analogue of $F_{ST}^k$ ; if $\ell = 1$ (haploid organisms), then $Q_{ST} = \rho_t^2$ (Eq. 28)
$W_S$	A quantity that equals 1 if an individual is classified into the wrong population on the basis of its value of $S$ , and that equals 0 otherwise (Eq. 55)
$W_T$	A quantity that equals 1 if an individual is classified into the wrong population on the basis of its value of $T$ , and that equals 0 otherwise (Eq. 56)

haploids. Here, we extend our earlier model to allow arbitrary allele-frequency distributions and arbitrary ploidy. Our results provide another way of establishing the result that between-group differentiation on a neutral trait mirrors between-group genetic differentiation at a neutral locus, one that makes minimal evolutionary assumptions. In Section 2, we describe our extended model. In Section 3, we define several measurements of between-group genetic differentiation. In Sections 4 and 5, we describe properties of two statistics that summarize the degree of difference between two populations on a quantitative trait. In Section 6, we introduce two simplifying assumptions that allow us to analyze the problem of inferring

an individual's population of origin using either genetic or phenotypic information. Finally, we discuss the results with respect to the interpretation of population differences in disease phenotypes. Figure 1 provides a conceptual map of the structure of the article.

## 2. Preliminaries

### 2.1 Model

Our extended model is parallel to our previously reported model (Edge and Rosenberg 2015) and is similar to models used by Risch et al. (2002), Edwards (2003), and especially Tal (2012) to

3. Some other Symbol Glyphs nearby Image are also include in the image causing the image to become much bigger and including irrelevant data as shown in Figure 3 a) and Figure 3 b).

Example File:

[/samples/Test\\_bench/RadicalSignDocuments/RP34\\_ResearchPaper/RP34\\_ResearchPaper.html](#)

To define  $F_{ST(\ell)}^k$  we sum these terms across loci to construct a ratio of the between-group variance to the total variance:

$$\begin{aligned} F_{ST(\ell)}^k &= \frac{\sum_{i=1}^k \text{Var}_M \left[ E \left( \sum_{j=1}^{\ell} L_{ij} \mid M \right) \right]}{\sum_{i=1}^k \text{Var} \left( \sum_{j=1}^{\ell} L_{ij} \right)} \\ &= \frac{\ell \bar{\delta}^2}{2[\bar{p}(1-\bar{p}) - s_p^2 + \bar{q}(1-\bar{q}) - s_q^2] + \ell \bar{\delta}^2}. \end{aligned} \quad (16)$$

We show in Appendix 1 that  $F_{ST(\ell)}^k \in [F_{ST}^k, \ell F_{ST}^k]$  with  $F_{ST(\ell)}^k = F_{ST}^k$  if and only if  $F_{ST}^k = 0$  or  $F_{ST}^k = 1$ . Figure 3 shows the relationship between  $F_{ST}^k$  and  $F_{ST(\ell)}^k$  for several values of  $\ell$ , illustrating the relative increase in  $F_{ST(\ell)}^k$  compared with  $F_{ST}^k$  as  $\ell$  increases. Figure 3 also illustrates, as shown in Appendix 1, that  $F_{ST(\ell)}^k$  is comparable to  $\ell F_{ST}^k$  for  $F_{ST}^k$  close to 0 and comparable to  $F_{ST}^k$  for  $F_{ST}^k$  close to 1.

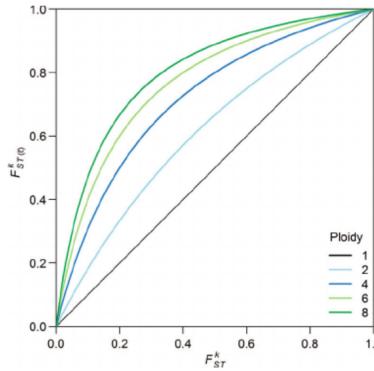
Similarly, we can define an analogue of  $D_L^2$  for an  $\ell$ -ploid locus:

$$\begin{aligned} D_{L(\ell)}^2 &= \frac{4F_{ST(\ell)}^k}{1-F_{ST(\ell)}^k} \\ &= \frac{\ell \bar{\delta}^2}{[\bar{p}(1-\bar{p}) - s_p^2 + \bar{q}(1-\bar{q}) - s_q^2]/2} \\ &= \ell D_L^2. \end{aligned} \quad (17)$$

Whereas  $F_{ST}^k$  and  $D_L^2$  can be viewed as indices of the amount of information about population membership available in a single copy of a typical locus,  $F_{ST(\ell)}^k$  and  $D_{L(\ell)}^2$  assess the total amount of population membership information at a typical locus, considering all  $\ell$  copies.

### 3.2 Simulation-Based Allele Frequency Differences

Because some of our results depend on specific characteristics of the  $p_i$  and  $q_j$ , we simulated allele frequencies under a model similar to that of Nicholson et al. (2002) to obtain suitable example distributions for the  $p_i$  and  $q_j$  (see Figure 4 for a schematic). Specifically, we generated allele frequencies for derived alleles in an ancestral population according to the neutral site frequency spectrum with  $2N = 20,000$ , choosing each allele frequency  $\pi_i$  according to  $P(\pi_i = j/(2N)) \propto 1/j$  (Charlesworth and Charlesworth 2010, Eq. B6.6.1). To simulate



**FIGURE 3.** The relationship between  $F_{ST}^k$  (Eq. 14), which partitions the variance of allelic indicator variables representing a single copy of each locus, and  $F_{ST(\ell)}^k$  (Eq. 16), which partitions the variance of sums of  $\ell$  allelic indicator variables at each locus. Thus, for haploids ( $\ell = 1$ ),  $F_{ST(\ell)}^k = F_{ST}^k$ . For higher ploidy,  $F_{ST(\ell)}^k \in [F_{ST}^k, \ell F_{ST}^k]$ , with  $F_{ST(\ell)}^k = F_{ST}^k$  if and only if  $F_{ST}^k = 0$  or  $F_{ST}^k = 1$  (see Appendix 1). The plot is obtained from Eq. A1.4.

drift after divergence, we produced postdivergence allele frequencies by adding to each “ancestral” allele frequency  $\pi_i$  an independently drawn Normal( $0, 0.3\pi_i(1-\pi_i)$ ) random number, where 0.3 is chosen so that  $F_{ST}^k$  approximates worldwide human  $F_{ST}$  estimates. Any postdivergence allele frequencies less than 0 or greater than 1 were set to 0 or 1, respectively. After simulating postdivergence frequencies of the derived allele independently in two populations, we assigned the frequencies of either the ancestral or the derived allele in each population to be  $p_i$  and  $q_j$ , requiring  $q_j \geq p_i$ . We generated  $10^6$  pairs of allele frequencies  $(p_i, q_j)$  after removing loci at which the same allele fixed in both populations. (Such loci do not contribute to  $F_{ST}^k$  or  $D_L^2$ .) For our simulated allele frequencies,  $\bar{p} \approx 0.457$ ,  $\bar{q} \approx 0.542$ ,  $\bar{s}_p^2 \approx s_q^2 \approx 0.191$ ,  $\bar{\delta} \approx 0.086$ ,  $\bar{\delta}^2 \approx 0.025$ ,  $\bar{\delta}^4 \approx 0.006$ ,  $F_{ST}^k \approx 0.099$ , and  $D_L^2 \approx 0.440$ . The  $F_{ST}^k$  value of 0.099 is similar to estimates of  $F_{ST}$  for human populations.

### 4. Properties of the Trait Value $T$ Conditional on the Labeling $X_i$

We next consider the distribution and properties of the trait value  $T$  in each population. In this section, we condition on the labeling of the alleles at each locus  $X_1, X_2, \dots, X_k$ . These labels determine, for each locus, whether the 1 or the 0 allele increases an individual’s trait value. In Section 5, we remove this condition and consider the expected behavior of the trait value under random assignment of the labels.

**Figure 3 a) Original Document Page with Figure**

To define  $F_{ST(\ell)}^k$  we sum these terms across loci to construct a ratio of the between-group variance to the total variance:

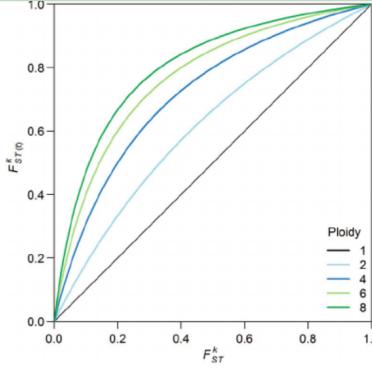
$$\begin{aligned} F_{ST(\ell)}^k &= \frac{\sum_{i=1}^k \text{Var}_M \left[ E \left( \sum_{j=1}^{\ell} L_{ij} \mid M \right) \right]}{\sum_{i=1}^k \text{Var} \left( \sum_{j=1}^{\ell} L_{ij} \right)} \\ &= \frac{\ell \bar{\delta}^2}{2[\bar{p}(1-\bar{p}) - s_p^2 + \bar{q}(1-\bar{q}) - s_q^2] + \ell \bar{\delta}^2}. \end{aligned} \quad (16)$$

We show in Appendix 1 that  $F_{ST(\ell)}^k \in [F_{ST}^k, \ell F_{ST}^k]$  with  $F_{ST(\ell)}^k = F_{ST}^k$  if and only if  $F_{ST}^k = 0$  or  $F_{ST}^k = 1$ . Figure 3 shows the relationship between  $F_{ST}^k$  and  $F_{ST(\ell)}^k$  for several values of  $\ell$ , illustrating the relative increase in  $F_{ST(\ell)}^k$  compared with  $F_{ST}^k$  as  $\ell$  increases. Figure 3 also illustrates, as shown in Appendix 1, that  $F_{ST(\ell)}^k$  is comparable to  $\ell F_{ST}^k$  for  $F_{ST}^k$  close to 0 and comparable to  $F_{ST}^k$  for  $F_{ST}^k$  close to 1.

Similarly, we can define an analogue of  $D_L^2$  for an  $\ell$ -ploid locus:

$$\begin{aligned} D_{L(\ell)}^2 &= \frac{4F_{ST(\ell)}^k}{1-F_{ST(\ell)}^k} \\ &= \frac{\ell \bar{\delta}^2}{[\bar{p}(1-\bar{p}) - s_p^2 + \bar{q}(1-\bar{q}) - s_q^2]/2} \\ &= \ell D_L^2. \end{aligned} \quad (17)$$

Whereas  $F_{ST}^k$  and  $D_L^2$  can be viewed as indices of the amount of information about population membership available in a single copy of a typical locus,  $F_{ST(\ell)}^k$  and  $D_{L(\ell)}^2$  assess the total amount of population membership information at a typical locus, considering all  $\ell$  copies.



**FIGURE 3.** The relationship between  $F_{ST}^k$  (Eq. 14), which partitions the variance of allelic indicator variables representing a single copy of each locus, and  $F_{ST(\ell)}^k$  (Eq. 16), which partitions the variance of sums of  $\ell$  allelic indicator variables at each locus. Thus, for haploids ( $\ell = 1$ ),  $F_{ST(\ell)}^k = F_{ST}^k$ . For higher ploidy,  $F_{ST(\ell)}^k \in [F_{ST}^k, \ell F_{ST}^k]$ , with  $F_{ST(\ell)}^k = F_{ST}^k$  if and only if  $F_{ST}^k = 0$  or  $F_{ST}^k = 1$  (see Appendix 1). The plot is obtained from Eq. A1.4.

drift after divergence, we produced postdivergence allele frequencies by adding to each “ancestral” allele frequency  $\pi_i$  an independently drawn Normal( $0, 0.3\pi_i(1-\pi_i)$ ) random number, where 0.3 is chosen so that  $F_{ST}$  approximates worldwide human  $F_{ST}$  estimates. Any postdivergence allele frequencies less than 0 or greater than 1 were set to 0 or 1, respectively. After simulating postdivergence frequencies of the derived allele independently in two populations, we assigned the frequencies of either the ancestral or the derived allele in each population to be  $p_i$  and  $q_i$ , requiring  $q_i \geq p_i$ . We generated  $10^6$  pairs of allele frequencies ( $p_i, q_i$ ) after removing loci at which the same allele fixed in both populations. (Such loci do not contribute to  $F_{ST}$  or  $D_L^2$ .) For our simulated allele frequencies,  $\bar{p} \approx 0.457$ ,  $\bar{q} \approx 0.542$ ,  $\bar{s}_p^2 \approx \bar{s}_q^2 \approx 0.191$ ,  $\bar{\delta} \approx 0.086$ ,  $\bar{\delta}^2 \approx 0.025$ ,  $\bar{\delta}^4 \approx 0.006$ ,  $F_{ST}^k \approx 0.099$ , and  $D_L^2 \approx 0.440$ . The  $F_{ST}^k$  value of 0.099 is similar to estimates of  $F_{ST}$  for human populations.

### 3.2 Simulation-Based Allele Frequency Differences

Because some of our results depend on specific characteristics of the  $p_i$  and  $q_i$ , we simulated allele frequencies under a model similar to that of Nicholson et al. (2002) to obtain suitable example distributions for the  $p_i$  and  $q_i$  (see Figure 4 for a schematic). Specifically, we generated allele frequencies for derived alleles in an ancestral population according to the neutral site frequency spectrum with  $2N = 20,000$ , choosing each allele frequency  $\pi_i$  according to  $P(\pi_i = j/(2N)) \propto 1/j$  (Charlesworth and Charlesworth 2010, Eq. B6.6.1). To simulate

### 4. Properties of the Trait Value $T$ Conditional on the Labeling $X_i$

We next consider the distribution and properties of the trait value  $T$  in each population. In this section, we condition on the labeling of the alleles at each locus  $X_1, X_2, \dots, X_k$ . These labels determine, for each locus, whether the 1 or the 0 allele increases an individual’s trait value. In Section 5, we remove this condition and consider the expected behavior of the trait value under random assignment of the labels.

4. Images with only line type data are coming in the foreground which cannot be detected by our algorithm as it searches for glyphs in background first as shown in Figure 4.

Example:

[/samples/Test\\_bench/TwoColumns/3315002.3317567/3315002.3317567.html](#)

```

<mjx-container class="MathJax" jax="CHTML" display="true" hasspeech="true" tabindex="1">
  <mjx-math type="relseq" role="equality" id="16" children="15,10" content="9"
    aria-label="a\u20d7x\u00b2+bx\u00b7c\u00b3=0"
    speech="a\u20d7x\u00b2+bx\u00b7c\u00b3=0">
    <mjx-mrow type="infixop" role="addition" id="15" children="12,14,8" content="4,7" parent="16"
      speech="a\u20d7x\u00b2+bx\u00b7c\u00b3=0">
      <mjx-mrow type="infixop" role="implicit" id="12" children="0,3" content="11" parent="15"
        speech="a\u20d7x\u00b2">
        <mjx-mi class="mjx-i" type="identifier" role="latinletter" font="italic" id="0" parent="12" speech="a">
          <mjx-c c="a" />
        </mjx-mi>
        <mjx-mo class="mjx-n" type="operator" role="multiplication" id="11" parent="12" speech="times">
          <mjx-c c="2062" />
        </mjx-mo>
        <mjx-msup type="superscript" role="latinletter" id="3" children="1,2" parent="12" speech="x\u00b2">
          <mjx-mi class="mjx-i" type="identifier" role="latinletter" font="italic" id="1" parent="3" speech="x">
            <mjx-c c="x" />
          </mjx-mi>
          ...
        </mjx-msup>
      </mjx-mrow>
    </mjx-mrow>
  </mjx-math>
</mjx-container>

```

Figure 2: Rendered quadratic equation in MathJax v3 with embedded semantic tree and speech.

MathML, with L<sup>A</sup>T<sub>E</sub>X being clearly prevalent on the majority of web pages. And unlike most historical systems [4] MathJax has no control over type or quality of input content. Similarly, MathJax was designed for displaying formulas, offering a number of different rendering solutions, such as HTML with CSS or SVG output. Figure 1 presents two standard ways of representing the quadratic equation  $ax^2 + bx + c = 0$ , in L<sup>A</sup>T<sub>E</sub>X on the top and in MathML on the left. Both are given in a flat structure, that is sufficient for a linear representation but in general is not enough to create good mathematical explanations and interaction support.

Consequently, the first step towards accessibility support in MathJax is by imposing a semantic interpretation on a given math expression and generating a tree representation that can be embedded into rendered MathJax expressions to ensure a similar user experience across browsers. The idea of the semantic interpretation is an extension of the heuristics implemented in the screen reader ChromeVox [13] and further developed in the context of MathJax [1], which effectively rewrites a flat expression into a term tree structure by first interpreting the basic nature of symbols, and propagating this through the expression to determines the scope of operators, relations, etc. Figure 1 presents this transformation for the example of the quadratic equation  $ax^2 + bx + c = 0$ , which is rewritten from either its MathML or L<sup>A</sup>T<sub>E</sub>X form into its semantic interpretation on the right.

The resulting semantic tree can be understood as an orthogonal view of the mathematical expression. To exploit it, we embed it using **data** attributes into the DOM elements that represent MathJax's rendering of the equation regardless of the choice of a particular rendering solution. MathJax offers several different output formats, where expressions in the DOM are collections of **div** and **span** elements, HTML5 custom elements, or SVG graphics elements. These collections often don't correspond well to the mathematical structures they represent. Figure 2 contains the MathJax's DOM output using HTML5 custom elements. **data** attributes **type**, **role**, **id**, **parent**, **children**, **content**, and **speech** are injected (here the **data-semantic-** prefix is omitted to preserve space and improve readability). The actual semantic tree structure is represented via the **id**, **parent**, **children**, and **content** attributes. Note that **data** attributes provide a

fast and standardized means of retrieving information from the DOM that is fully consistent with HTML5 practices.

### 3. ACCESSIBILITY FEATURES

MathJax's assistive technology extension is mainly aimed at supporting users with reading disorders, such as dyslexia, and visual impairments. However, some aspects of it could also be used as a general aid for readers unfamiliar with the content or for learners at different levels. We summarize the main features in this section.

#### 3.1 Speech Output

One main emphasis of the assistive technology extension is to support screen-reader users, but make them independent of their particular screen-reader's Math capabilities. MathJax exploits SRE's feature to provide aural rendering for mathematical expressions. Speech string computation is based on the embedded semantic tree. Speech strings for a formula are either generated on the fly when running MathJax in a browser client, or pre-computed by a page author, when formulas are pre-rendered and inserted into the DOM.

To expose the speech strings to a screen reader, MathJax uses two main techniques. Firstly, a description of an entire formula is given in the ARIA label at the top level of the DOM structure representing the expression. Figure 2 demonstrates the ARIA label embedding as well as the use of a custom **data** attribute for speech. The latter allows us to aurally render not only the entire formula but each of its sub-expressions, which is exploited during user interaction with the formula, as discussed below. For this purpose, MathJax introduces a dedicated assertive ARIA live region into the DOM and updates it with the desired speech output. Content changes in the live region are picked up by screen readers and spoken. This is a feature that is particularly in line with MathJax's mission to support mathematics rendering in all browsers and on all platforms, working with any modern screen reader that supports live regions.

#### 3.2 Tactile Output

In addition to speech output, SRE now provides generation of Braille output using Unicode Braille symbols. It currently supports translation of expressions into Nemeth

Figure 4 Figure is Present Completely in the Document Page Foreground

5. As we are using the getTextBoxes() function inside the previous code, if this code fails in correct paragraph formation or identifier formation then our code fails in detecting the captions correctly as shown in Figure 5 a) and Figure 5 b).

Example:

[/samples/Test\\_bench/SingleColumn/paper2/paper2.html](#)

**Table 1.** Two scenes used for captioning evaluation

	scene A	scene B
duration	3m 30s	3m 8s
no. of sentences to be written	15	15

**Table 2.** Grouping for captioning evaluation

group	1st experiment	2nd experiment
A	scene A without easy insertion	scene B with easy insertion
B	scene A with easy insertion	scene B without easy insertion
C	scene B without easy insertion	scene A with easy insertion
D	scene B with easy insertion	scene A without easy insertion

We grouped the participants into four groups. Each group viewed the scenes in a different order and with or without the easy insertion. Table 2 lists the details of each group. We gave the participants a practice lecture scene and allowed them about two minutes to study it before each captioning experiment. Figure 4 shows the arrangement of the experiment. The left monitor displayed the screen of the proposed system and right monitor displayed the lecture clip. This situation simulated remote captioning. The participants watched the lecture through the monitor from a distant location and typed a summary text.

We selected the important sentences in the lecture that should be written as captions in advance. The number of sentences to be written is shown in Table 1. We counted the number of important sentences that appeared in the sentences the participants input. The number of input sentences by each participant is given in Table 3.

After a participant finished captioning, we asked about the usefulness of the system. We asked two questions. One was about the easiness of captioning, and the other was about the usefulness of the easy insertion. Figures 5 and 6 show the histograms of each answer, respectively.

**Table 3.** Number of input sentences

group	A					B					
participant	1	2	3	4	5	6	7	8	9	10	11
with easy insertion	8	10	9	9	10	8	14	8	10	10	12
without easy insertion	4	5	7	9	12	8	8	0	4	3	3
group	C					D					
participant	12	13	14	15	16	17	18	19	20	21	22
with easy insertion	7	2	3	5	4	4	3	4	4	9	6
without easy insertion	2	5	4	5	4	3	5	3	2	6	2

Figure 5 a) Original Document Page with table

The screenshot shows the Chrome DevTools interface with the 'Console' tab selected. The console output is a list of log messages, each consisting of a text entry and a file path. The messages are as follows:

- it on the page.ts:352
- line115 page.ts:352
- text: screen for the participants. We showed them two scenes. The details of the page.ts:367
- length and page.ts:367
- line116 page.ts:352
- text: scenes are summarized in Table 1. The scenes were almost the same page.ts:367
- line117 page.ts:352
- text: same difficulty level in captioning. We recruited 22 graduate and undergraduate page.ts:367
- line118 page.ts:352
- text: students who could type very quickly and who would understand the content of page.ts:367
- line119 page.ts:352
- text: the lecture. page.ts:367
- line120 page.ts:352
- text: Captioning System with Function of Inserting Mathematical Formula Images 37 page.ts:367
- line121 page.ts:352
- text: Tab 1 e 1. Two scenes used for captioning evaluation page.ts:367
- line122 page.ts:352
- text: scene A scene B page.ts:367
- line123 page.ts:352
- text: duration 3m 30s 3m 8s page.ts:367
- line124 page.ts:352
- text: no. of sentences to be written 15 15 page.ts:367
- line125 page.ts:352
- text: Tab 1 e 2. Grouping for captioning evaluation page.ts:367
- line126 page.ts:352
- text: group 1st experiment 2nd experiment page.ts:367
- line127 page.ts:352
- text: A scene A without easy insertion scene B with easy insertion page.ts:367
- line128 page.ts:352
- text: B scene A with easy insertion scene B without easy insertion page.ts:367
- line129 page.ts:352
- text: C scene B without easy insertion scene A with easy insertion page.ts:367
- line130 page.ts:352
- text: D scene B with easy insertion scene A without easy insertion page.ts:367
- line131 page.ts:352
- text: We grouped the participants into four groups. Each group viewed page.ts:367

Figure 5 b) Table 1 & 2 are not detected as their Key identifiers start as 'Table'

# Chapter 6

## Integration with RAVI

### Workflow:

- let d = new pdf2charinfo.document();
- d.linify();
- d.backgroundAnalysis();
- d.lines\_textify();

We have observed that line's html attribute innerText is not set but its spans contain html attribute innerText, so, here we are concating all spans' html attribute innerText and assigning as line's html attribute innerText.

- d.object\_map();

This is our main algorithm. It can take Arguments inside as Caption Identifiers, Note Identifiers for different languages and Glyphs Threshold Fractions, Line Threshold Fractions for inclusion of nearby glyphs, lines to form a complete image . Our default Identifiers are set in English Language and default Thresholds are saved inside 'documentThresholds.ts' file. This Algorithm finds the captions using the identifiers and then searches at all four sides of caption position for images. After searching for all directions for all captions it takes the mappings which are giving the maximum number of captions to Object maps.

- `d.find_multimap_glyph();`

In the above caption mappings if by chance 2 or more captions are mapped to the same object this function separates them based on their relative distance and position from the captions creating two different Objects for both of the captions.

- `d.table_presence();`

As we are placing only the Figure Captions in the final html output, this function stores the table type captions and their mapped table image inside the file ‘documentThresholds.ts ’ for future work on table detection and placement.

- `d.showPage();`

This function refreshes the pages with both the foreground and background data.

- `d.draw_object_box();`

The above function page is cropped to get the Complete images of the captions for placement in the final html.

- `d.match();`

Matches the lines inside the document with their respective element tag.

- `d.toHtml();`

Creates the final Html output and places the lines inside with their respective tags

- `d.ImagestoHtml();`

Places the Image formed by the captions mapping in the final Html

# Chapter 7

## Conclusion and Future Work

We have tested 66 documents with a total 490 objects using our algorithm. It is resulting in 95% accuracy. Our caption detection and object mapping takes an average of 667 milliseconds/Page.

Future work are as follows:

- Column lines ordering correction in output HTML.
- Enhancement of getTextBoxes() Function Paragraph Detection.
- Page layout/Decorative background glyphs identification and its removal from the caption mapped object in the document.
- Mathematical symbol background glyphs identification and its removal from the caption mapped object in the document.
- Tables are placed as images and some tables were not detected correctly. So, placement and detection of the correct table is also one of future work.

## References

1. Clark, C. A., & Divvala, S. K. (2015). Looking Beyond Text: Extracting Figures, Tables and Captions from Computer Science Papers. (AAAI Workshop: Scholarly Big Data).