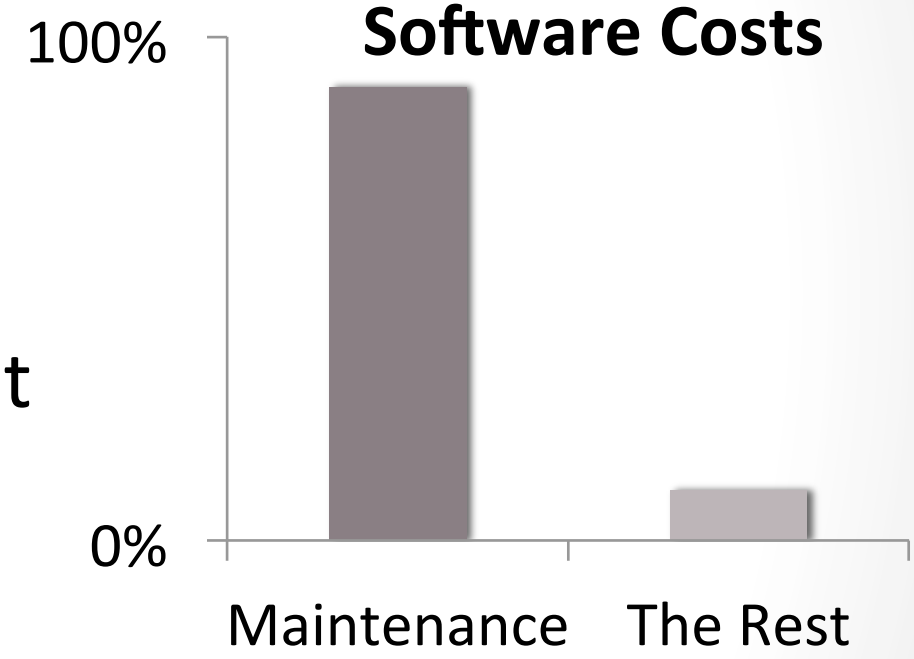
A General Software Readability Model

Jonathan Dorn

December 18, 2012

Software Maintenance Costs

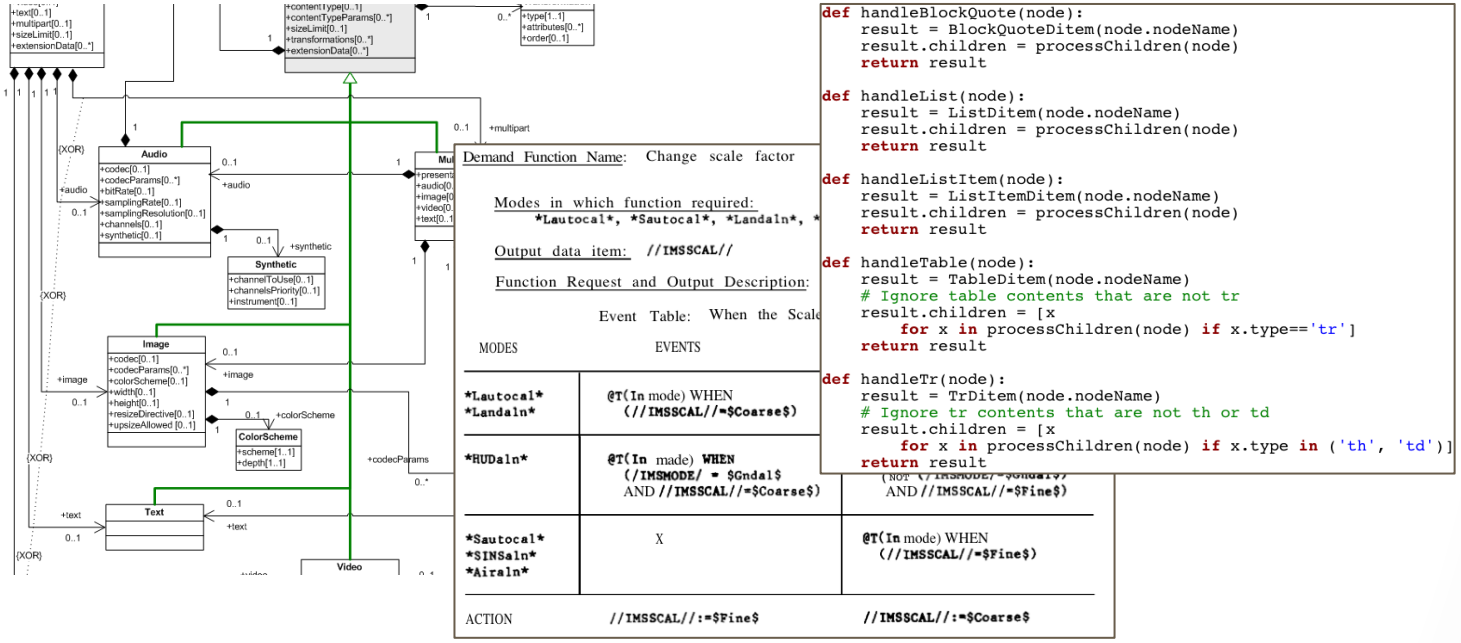
- Maintenance may cost up to **9x** all other development costs.



R.C. Seacord, D. Plakosh, and G.A. Lewis. Modernizing Legacy Systems: Software Technologies, Engineering Process and Business Practices. Addison-Wesley Longman Publishing Co. Inc., Boston, MA, USA, 2003.

Reading and Maintenance

"A central activity in software maintenance is ***reading***."*

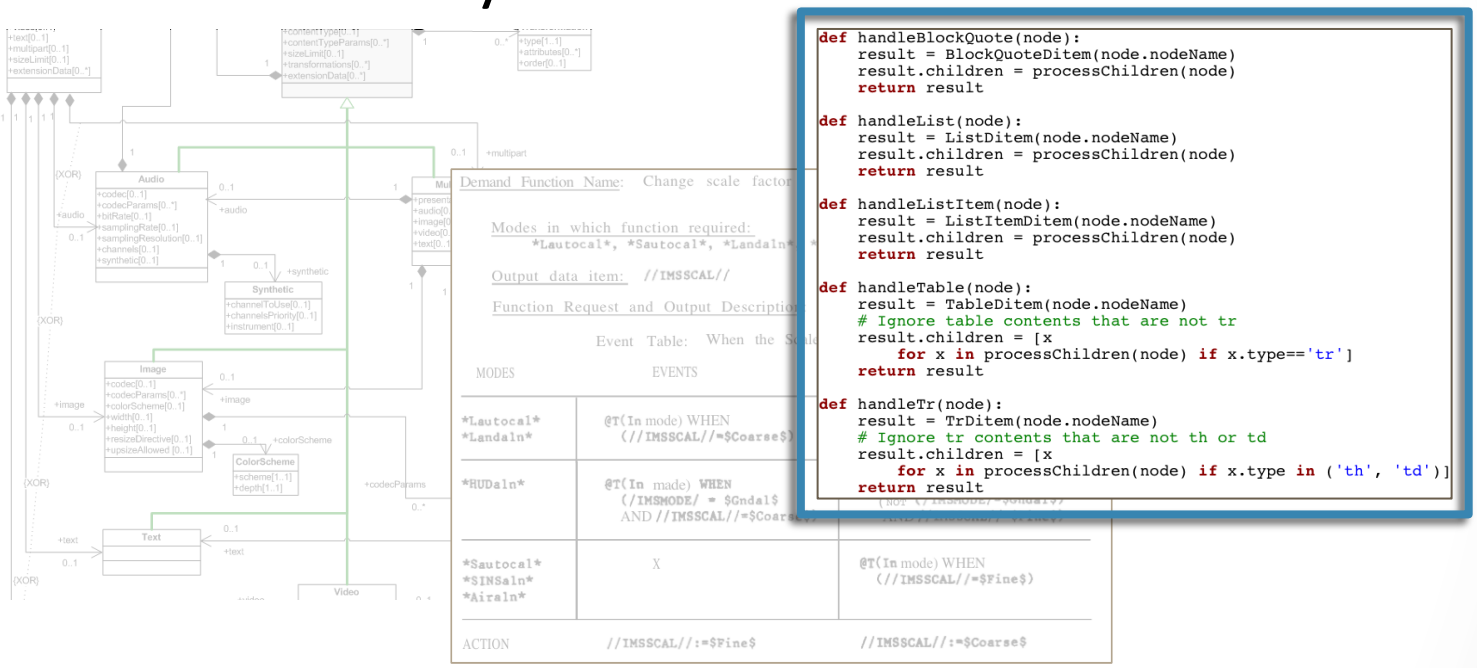



```
def handleBlockQuote(node):
    result = BlockQuoteDitem(node.nodeName)
    result.children = processChildren(node)
    return result

def handleList(node):
    result = ListDitem(node.nodeName)
    result.children = processChildren(node)
    return result

def handleListItem(node):
    result = ListItemDitem(node.nodeName)
    result.children = processChildren(node)
    return result

def handleTable(node):
    result = TableDitem(node.nodeName)
    # Ignore table contents that are not tr
    result.children = [x
        for x in processChildren(node) if x.type=='tr']
    return result

def handleTr(node):
    result = TrDitem(node.nodeName)
    # Ignore tr contents that are not th or td
    result.children = [x
        for x in processChildren(node) if x.type in ('th', 'td')]
    return result
```

* D. R. Raymond, "Reading source code," in *Conference of the Centre for Advanced Studies on Collaborative Research*, 1991.

Reading and Maintenance

"A central activity in software maintenance is ***reading***."*



```
def handleBlockQuote(node):
    result = BlockQuoteDitem(node.nodeName)
    result.children = processChildren(node)
    return result

def handleList(node):
    result = ListDitem(node.nodeName)
    result.children = processChildren(node)
    return result

def handleListItem(node):
    result = ListItemDitem(node.nodeName)
    result.children = processChildren(node)
    return result

def handleTable(node):
    result = TableDitem(node.nodeName)
    # Ignore table contents that are not tr
    result.children = [x
        for x in processChildren(node) if x.type=='tr']
    return result

def handleTr(node):
    result = TrDitem(node.nodeName)
    # Ignore tr contents that are not th or td
    result.children = [x
        for x in processChildren(node) if x.type in ('th', 'td')]
    return result
```

* D. R. Raymond, "Reading source code," in *Conference of the Centre for Advanced Studies on Collaborative Research*, 1991.

readability, *n.*

The **ease** with which a text may be **scanned** or **read**; the quality in a book, etc., of being easy to understand and enjoyable to read.

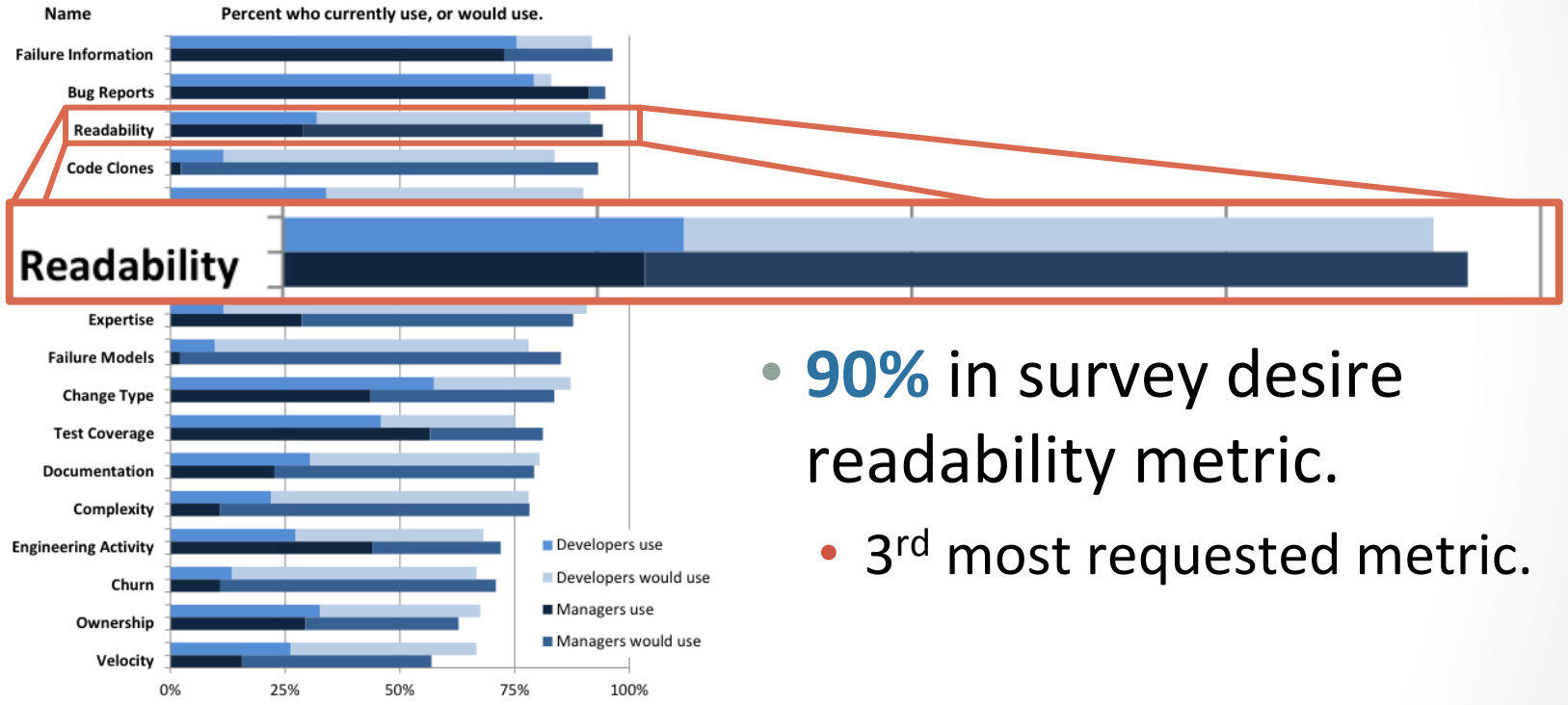
www.oed.com

Making Code More Readable

- **Programming languages**
  - Literate Programming (e.g. CWEB) [Knuth 1984]
  - Python [Van Rossum 1996]

- **Development Process**
  - Readability development phase [Elshoff & Marcotty 1982]
  - Readability review phase [Knight & Myers 1993]
  - Readability team [Haneef 1998]

Is It Working?



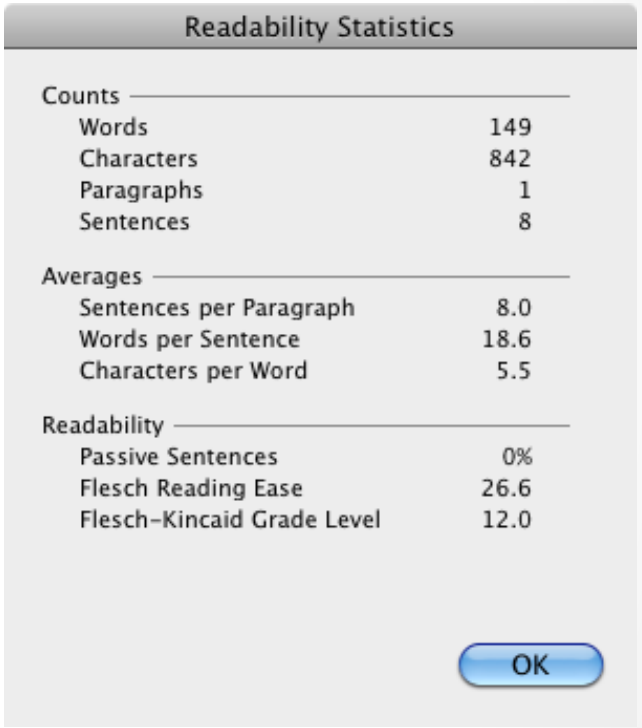
- **90%** in survey desire readability metric.
  - 3rd most requested metric.

R.P.L. Buse and T. Zimmermann, "Information needs for software development analytics," in International Conference on Software Engineering, 2012.

Parallels: English Readability

- Flesch-Kincaid Grade Level

- Government mandated
  - Military manuals: **9th grade**
    DOD MIL-M-28784B
  - Insurance policies: **10th grade**
    C.R.S 10-16-107.3 (1)(a)



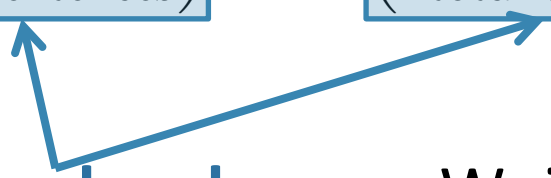
Flesch-Kincaid Grade Level

$$0.39\left(\frac{\text{total words}}{\text{total sentences}}\right) + 11.8\left(\frac{\text{total syllables}}{\text{total words}}\right) - 15.59$$

- Simple surface-level features (syllables, words, sentences).
- Weights calculated using regression analysis.

Flesch-Kincaid Grade Level

$$0.39\left(\frac{\text{total words}}{\text{total sentences}}\right) + 11.8\left(\frac{\text{total syllables}}{\text{total words}}\right) - 15.59$$



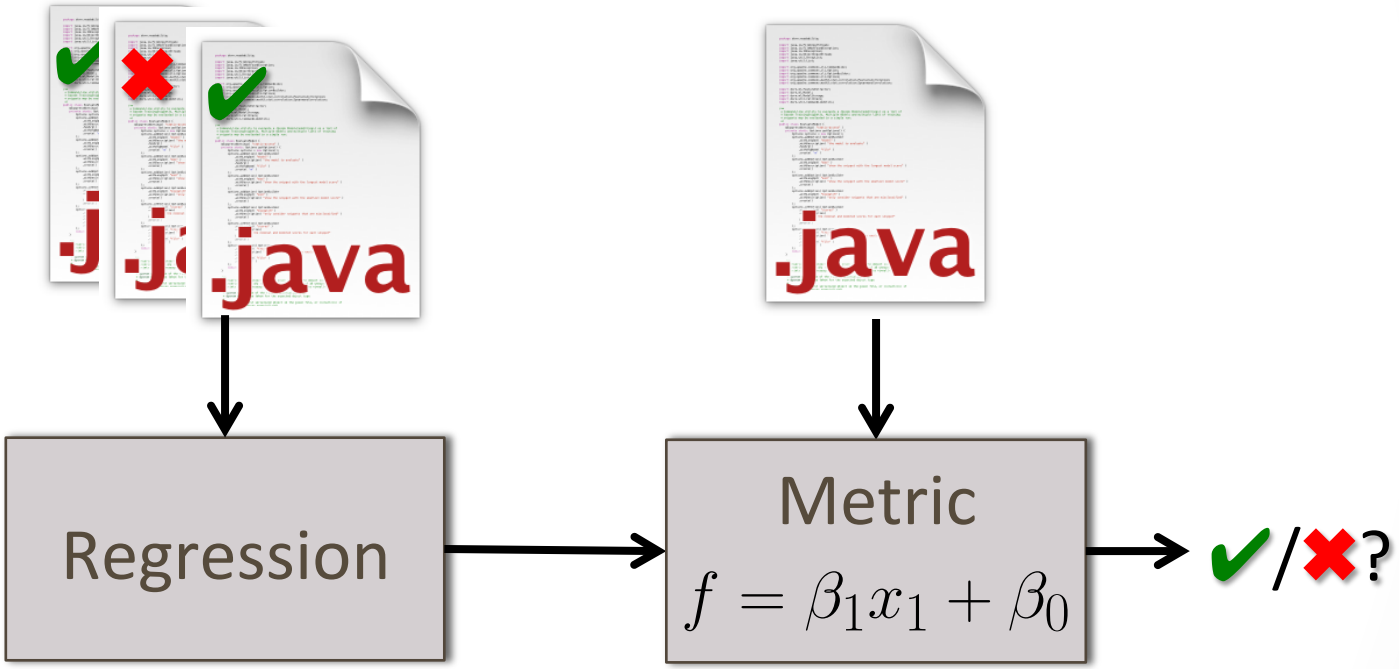
- Simple **surface-level features** (syllables, words, sentences).
- Weights calculated using regression analysis.

Flesch-Kincaid Grade Level

$$0.39\left(\frac{\text{total words}}{\text{total sentences}}\right) + 11.8\left(\frac{\text{total syllables}}{\text{total words}}\right) - 15.59$$

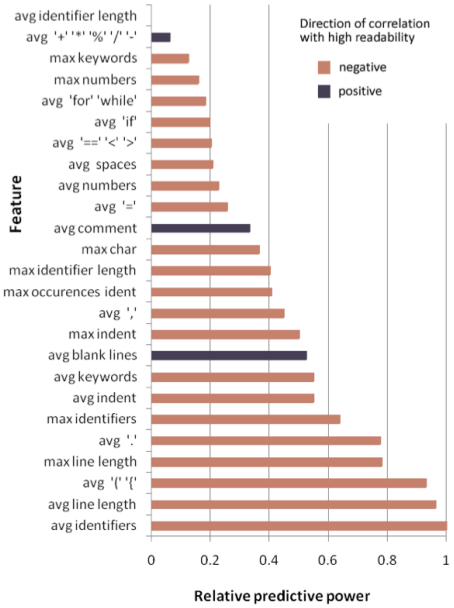
- Simple surface-level features (syllables, words, sentences).
- **Weights** calculated using regression analysis.

Learning a Readability Metric



Source Code Readability

- Buse & Weimer 2008
  - 25 surface features (max line length, average whitespace, etc.)



- Posnett, et al. 2011

$$8.87 - 0.033(\text{Halstead volume}) + 0.4(\text{total lines}) - 1.5(\text{token entropy})$$

Problem solved?

Code Examples

```
def handleBlockQuote(node):
    result = BlockQuoteDitem(node.nodeName)
    result.children = processChildren(node)
    return result

def handleList(node):
    result = ListDitem(node.nodeName)
    result.children = processChildren(node)
    return result

def handleListItem(node):
    result = ListItemDitem(node.nodeName)
    result.children = processChildren(node)
    return result

def handleTable(node):
    result = TableDitem(node.nodeName)
    # Ignore table contents that are not tr
    result.children = [x
        for x in processChildren(node) if x.type=='tr']
    return result

def handleTr(node):
    result = TrDitem(node.nodeName)
    # Ignore tr contents that are not th or td
    result.children = [x
        for x in processChildren(node) if x.type in ('th', 'td')]
    return result
```

Code Examples

```
        //float *attenuationIntegralPlaneArray_d;    //stores partial integral on planes parallel to the camera
        //CUDA_SAFE_CALL(cudaMalloc((void **)&attenuationIntegralPlaneArray_d, img->dim[1]*img->dim[3]*sizeof(float)));

        et_line_integral_attenuated_gpu_kernel <<<G1,B1>>> (*d_activity, *d_attenuation, currentCamPointer);

        CUDA_SAFE_CALL(cudaThreadSynchronize());
}
```

Example Readability

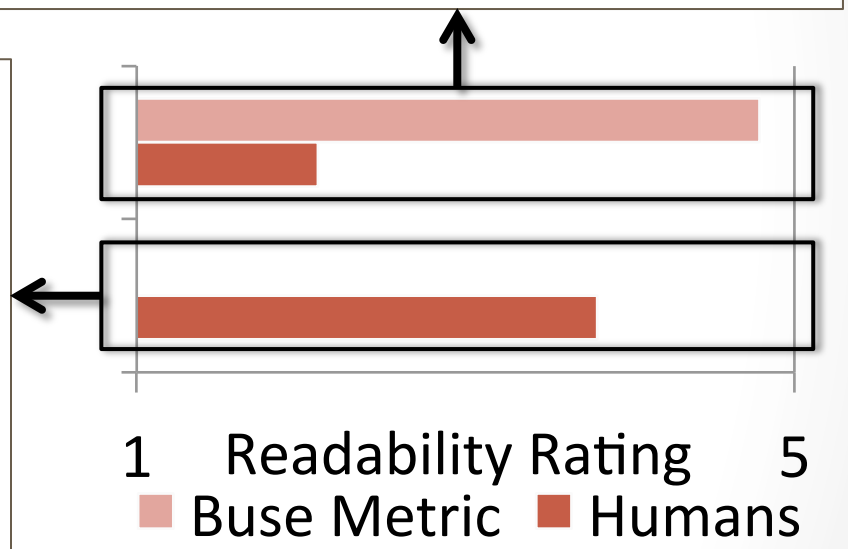
```
        //float *attenuationIntegralPlaneArray_d;    //stores partial integral on planes parallel to the camera
        //CUDA_SAFE_CALL(cudaMalloc((void **)&attenuationIntegralPlaneArray_d, img->dim[1]*img->dim[3]*sizeof(float)));

        et_line_integral_attenuated_gpu_kernel <<<G1,B1>>> (*d_activity, *d_attenuation, currentCamPointer);

        CUDA_SAFE_CALL(cudaThreadSynchronize());
}
```

```
def handleBlockQuote(node):
    result = BlockQuoteDitem(node.nodeName)
    result.children = processChildren(node)
    return result

def handleList(node):
    result = ListDitem(node.nodeName)
    result.children = processChildren(node)
    return result

def handleListItem(node):
    result = ListItemDitem(node.nodeName)
    result.children = processChildren(node)
    return result

def handleTable(node):
    result = TableDitem(node.nodeName)
    # Ignore table contents that are not tr
    result.children = [x
        for x in processChildren(node) if x.type=='tr']
    return result

def handleTr(node):
    result = TrDitem(node.nodeName)
    # Ignore tr contents that are not th or td
    result.children = [x
        for x in processChildren(node) if x.type in ('th', 'td')]
    return result
```

Metric Mismatch

```
        //float *attenuationIntegralPlaneArray_d;    //stores partial integral on planes parallel to the camera
        //CUDA_SAFE_CALL(cudaMalloc((void **)&attenuationIntegralPlaneArray_d, img->dim[1]*img->dim[3]*sizeof(float)));

        et_line_integral_attenuated_gpu_kernel <<<G1,B1>>> (*d_activity, *d_attenuation, currentCamPointer);

        CUDA_SAFE_CALL(cudaThreadSynchronize());
}
```

```
def handleBlockQuote(node):
    result = BlockQuoteDitem(node.nodeName)
    result.children = processChildren(node)
    return result

def handleList(node):
    result = ListDitem(node.nodeName)
    result.children = processChildren(node)
    return result

def handleListItem(node):
    result = ListItemDitem(node.nodeName)
    result.children = processChildren(node)
    return result

def handleTable(node):
    result = TableDitem(node.nodeName)
    # Ignore table contents that are not tr
    result.children = [x
        for x in processChildren(node) if x.type=='tr']
    return result

def handleTr(node):
    result = TrDitem(node.nodeName)
    # Ignore tr contents that are not th or td
    result.children = [x
        for x in processChildren(node) if x.type in ('th', 'td')]
    return result
```



What happened?

What Happened?

Model

- Character features only.
- Missing:
  - Structural patterns.
  - Line-to-line variation.
  - Spatial layout.
  - Syntax highlighting.

Ground Truth

- Small survey
  - 120 participants.
- Similar backgrounds
  - All UVa students.
- One programming language
  - Java.
- Short code samples
  - 4 – 13 lines.

General Readability Metric

1. New model.
   - Buse baseline features
   - Additional visual features
2. Ground truth from a large human study.
3. Combine and evaluate.

General Readability Metric

1. **New model.**
   - Buse baseline features
   - **Additional visual features**
2. Ground truth from a large human study.
3. Combine and evaluate

Visual Structural Features

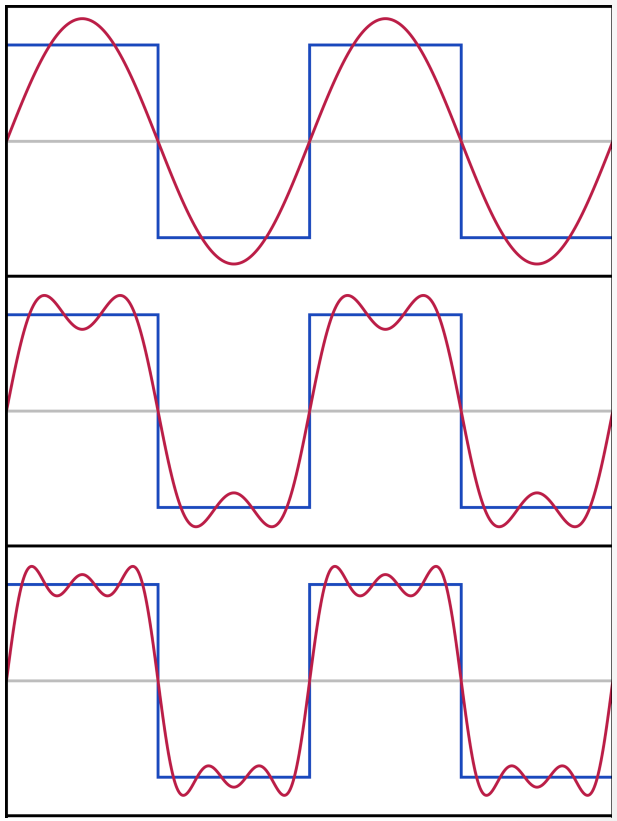
Visual Structural Features

- Line-to-line periodic structure
  - E.g. indentation.
- How can we measure periodicity?

Fourier Series

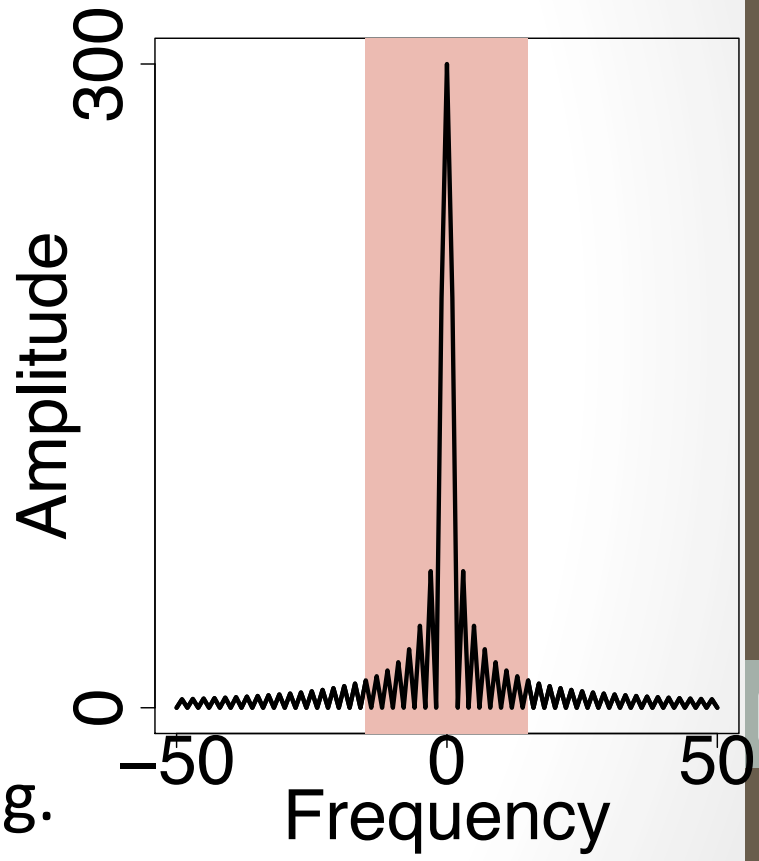
- **Idea**: periodic functions can be written as the sum of a series of sines.

$$\sum_{n=-\infty}^{\infty} c_n \left(\cos(nx) + i\sin(nx)\right)$$



Discrete Fourier Transforms

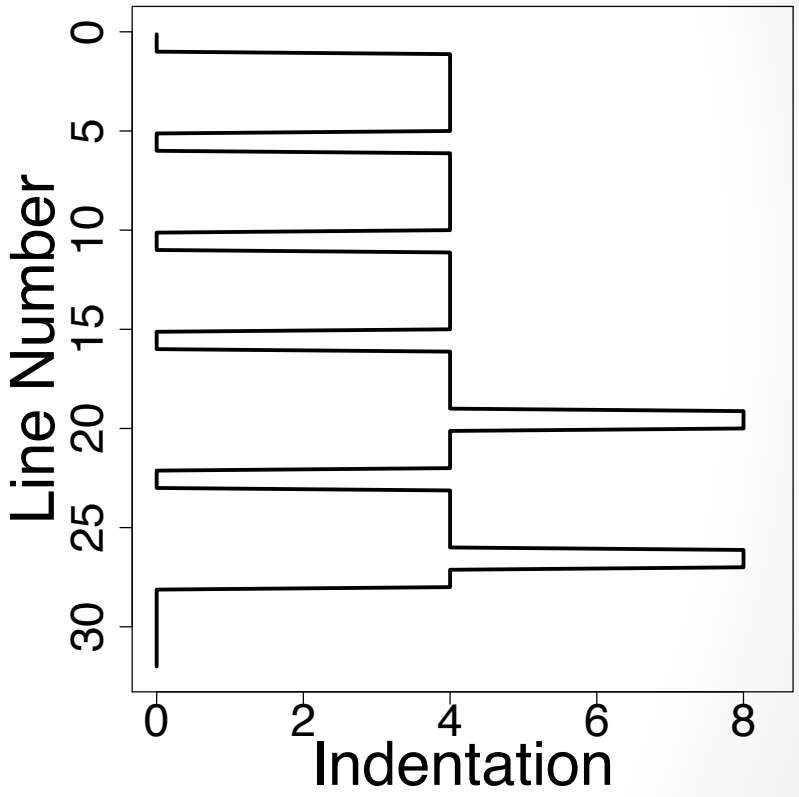
- The **Discrete Fourier Transform** (DFT) computes the coefficients.
- **Bandwidth**: the range of important coefficients.
- Common in signal processing.



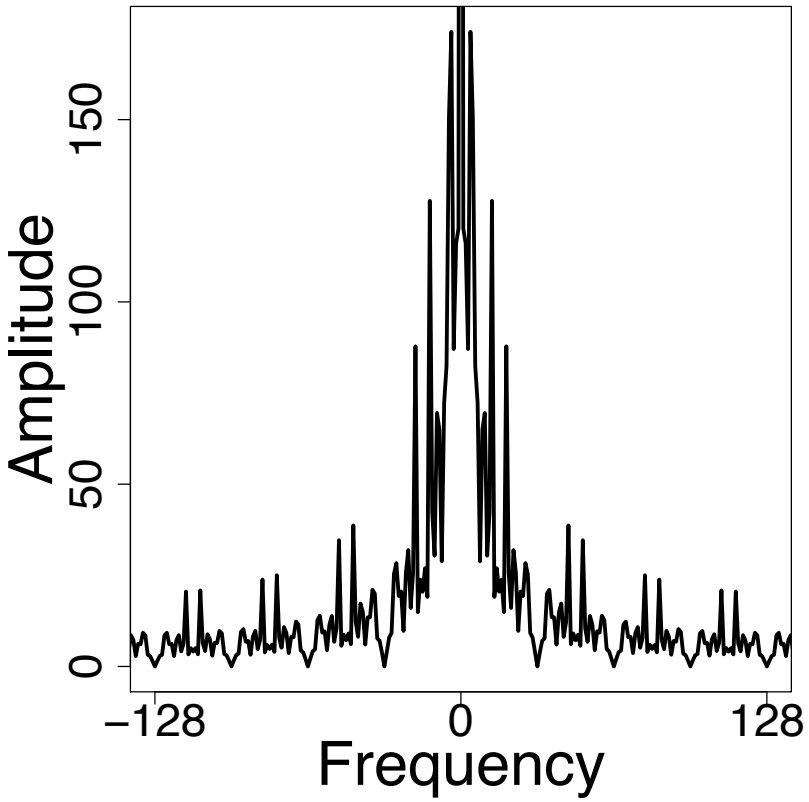
Visual Structural Features

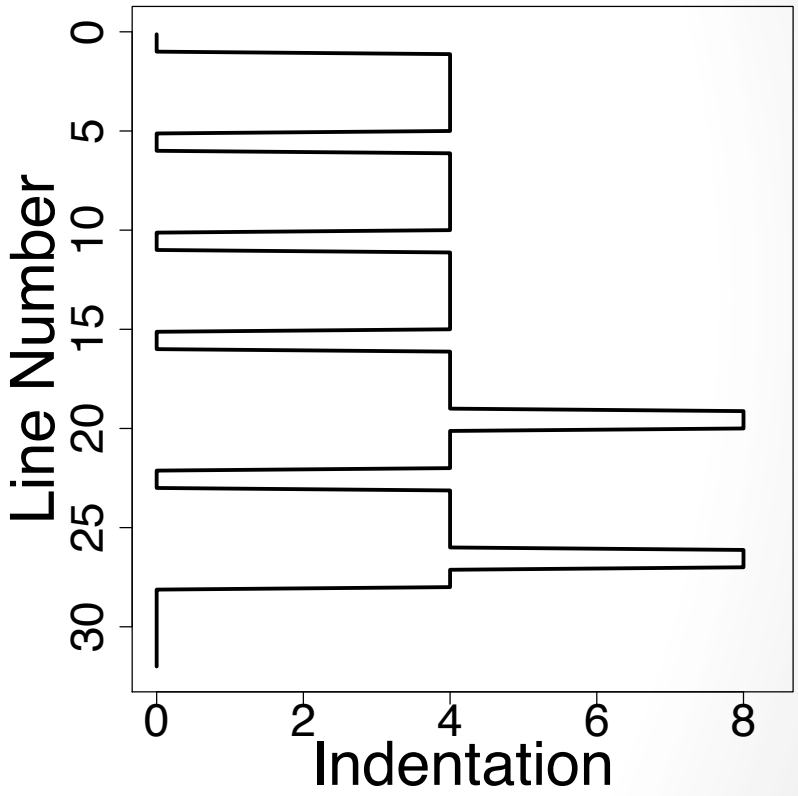
- Sample at each line.
- Take DFT of samples.
- Record bandwidth.

DFT Example (indentation)

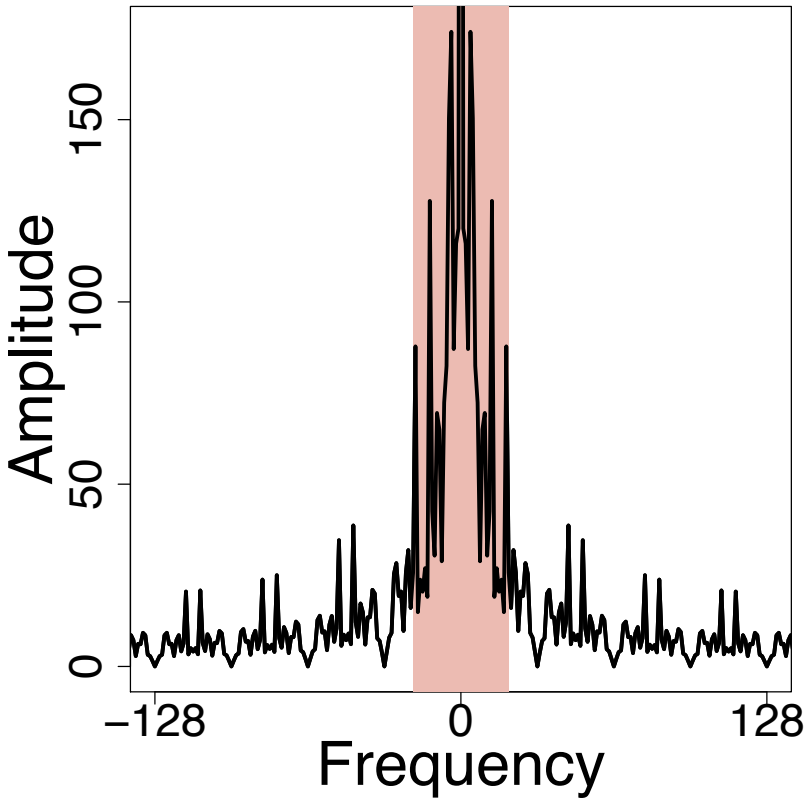


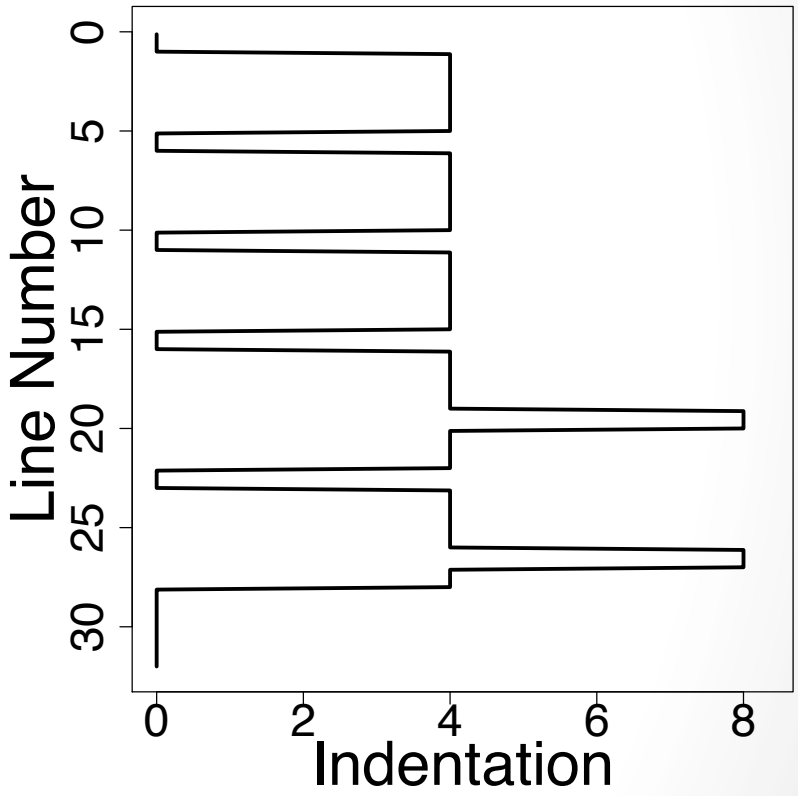
DFT Example (indentation)



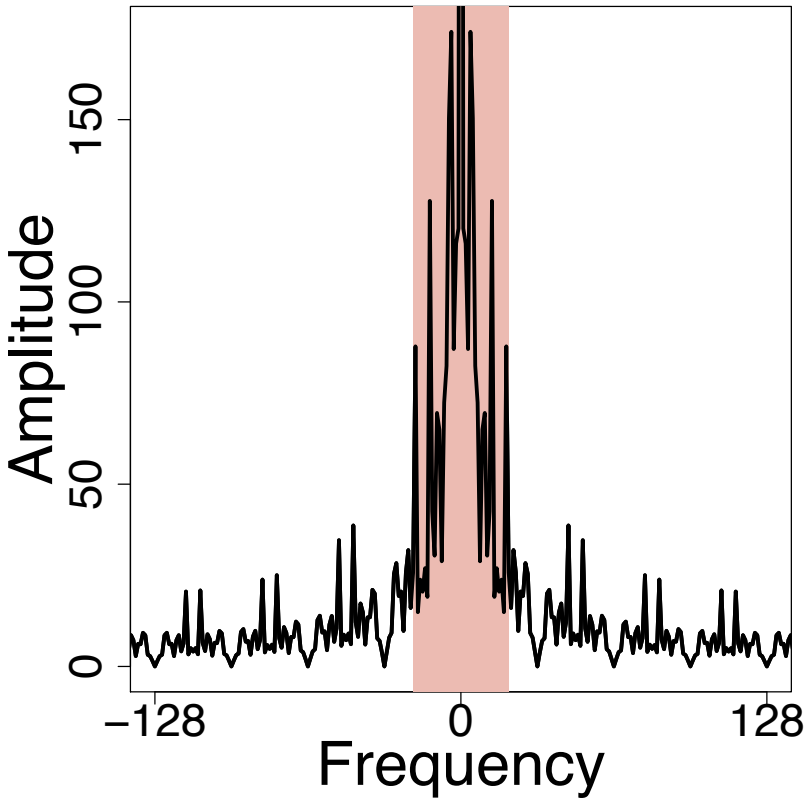


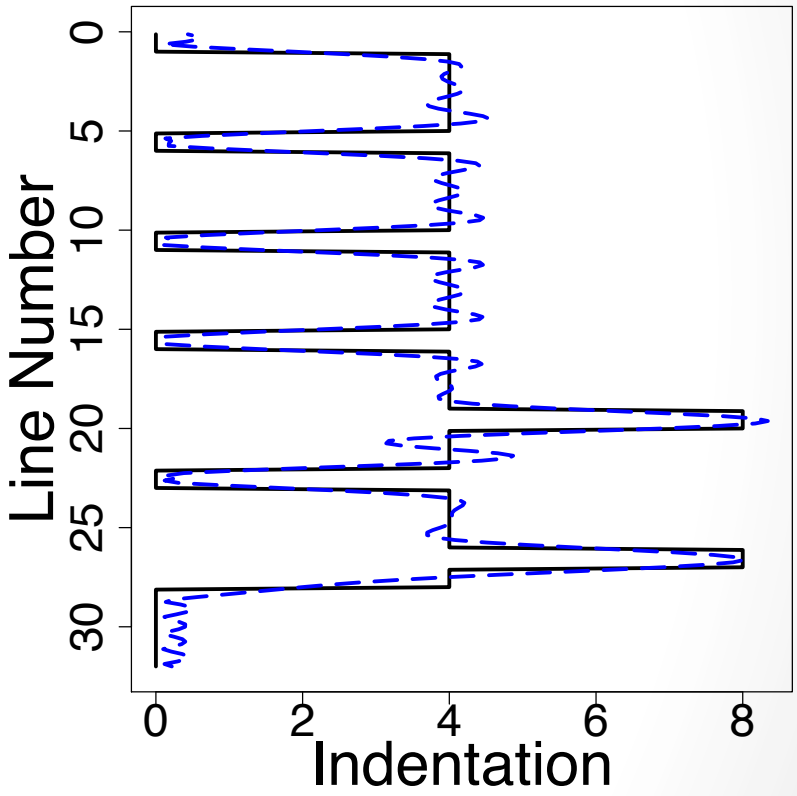
DFT Example (indentation)





DFT Example (indentation)





Spatial Layout Features

```
            deltaW += vd[threadIdx.x] * hd[threadIdx.y] - vr[threadIdx.x] * hr[threadIdx.y];
    }

    // update weights
    if (i < I && j < J) {
            deltaW /= samples;

            int w = j * I + i;

            cudafloat learningRate = UpdateLearningRate(learningRateW, lastDeltaWithoutLearningMomentumW, deltaW, w, u, d);
            UpdateWeight(learningRate, momentum, deltaW, lastDeltaW, lastDeltaWithoutLearningMomentumW, weights, w);
    }

    if(i < I && threadIdx.y == 0) {
            errors[i] = error;

            // Update a
            if (j == 0) {
                    deltaA /= samples;

                    cudafloat learningRate = UpdateLearningRate(learningRateA, lastDeltaWithoutLearningMomentumA, deltaA, i, u, d);
                    UpdateWeight(learningRate, momentum, deltaA, lastDeltaA, lastDeltaWithoutLearningMomentumA, a, i);
            }
    }

    // Update b
    if (i == 0 && j < J) {
            deltaB /= samples;
```

```
class class_attribute(PythonStructural, Element): pass
class expression_value(PythonStructural, Element): pass
class attribute(PythonStructural, Element): pass

# Structural Support Elements
# ---------------------------

class parameter_list(PythonStructural, Element): pass
class parameter_tuple(PythonStructural, Element): pass
class parameter_default(PythonStructural, TextElement): pass
class import_group(PythonStructural, TextElement): pass
class import_from(PythonStructural, TextElement): pass
class import_name(PythonStructural, TextElement): pass
class import_alias(PythonStructural, TextElement): pass
class docstring(PythonStructural, Element): pass

# ================
#  Inline Elements
# ================

# These elements cannot become references until the second
# pass.  Initially, we'll use "reference" or "name".

class object_name(PythonStructural, TextElement): pass
class parameter_list(PythonStructural, TextElement): pass
class parameter(PythonStructural, TextElement): pass
class parameter_default(PythonStructural, TextElement): pass
class class_attribute(PythonStructural, TextElement): pass
class attribute_tuple(PythonStructural, TextElement): pass
```

Spatial Layout Features

- Fraction of screen occupied by each color.
  - Count area highlighted with each color.
  - Record ratios between colors.

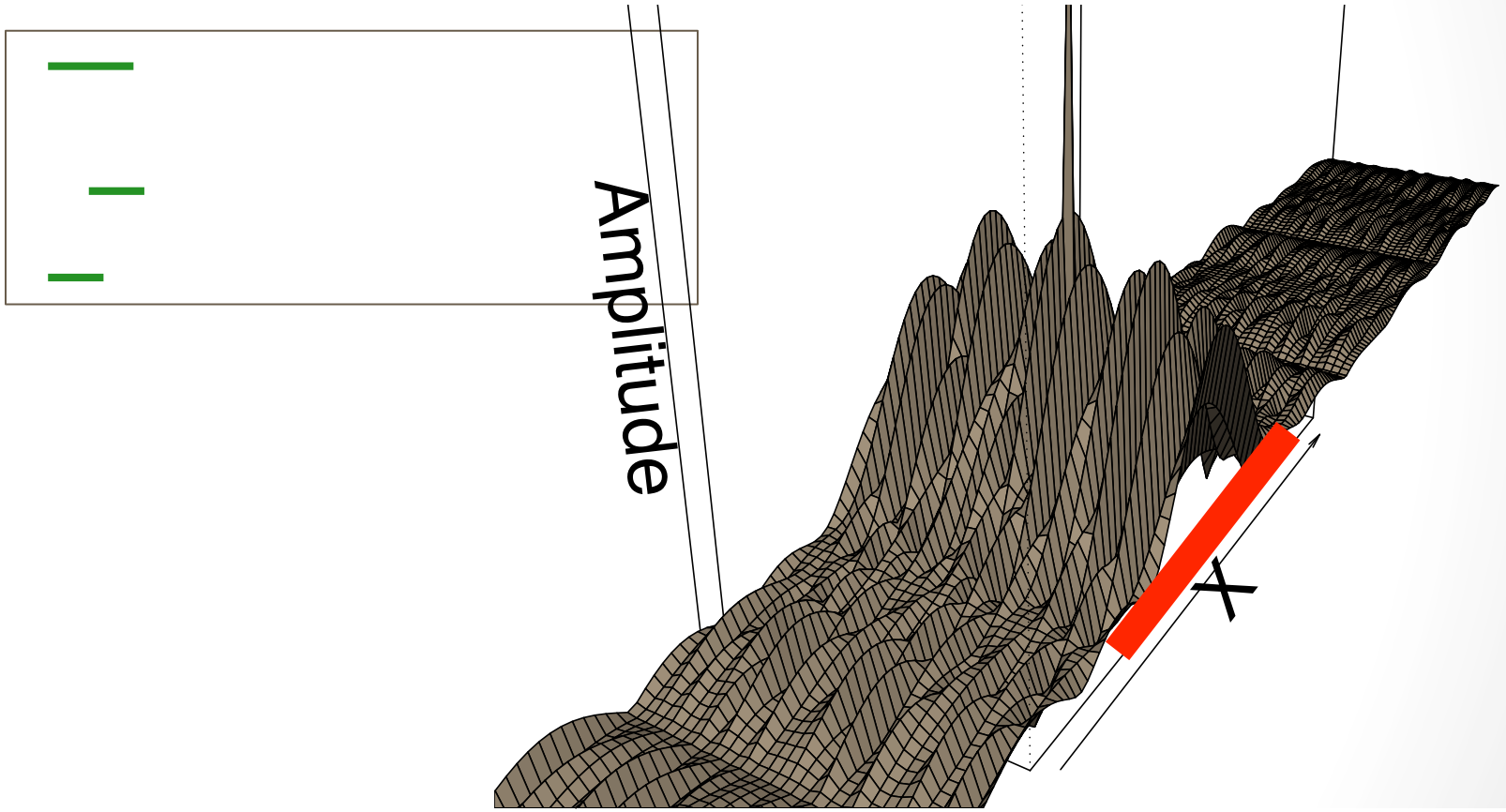
- Patterns of color.
  - Construct matrix of 0s (whitespace) and 1s (highlighted text).
  - Compute 2D DFT of matrix.
  - Record average bandwidth in X and Y dimensions.

DFT Example (comments)

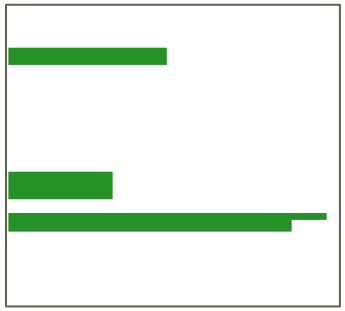
```
        deltaW += vd[threadIdx.x] * hd[threadIdx.y] - vr[threadIdx.x] * hr[threadIdx.y];
    }

    // update weights
    if (i < I && j < J) {
        deltaW /= samples;

        int w = j * I + i;

        cudafloat learningRate = UpdateLearningRate(learningRateW, lastDeltaWithoutLearningMomentumW, deltaW, w, u, d);
        UpdateWeight(learningRate, momentum, deltaW, lastDeltaW, lastDeltaWithoutLearningMomentumW, weights, w);
    }

    if(i < I && threadIdx.y == 0) {
        errors[i] = error;

        // Update a
        if (j == 0) {
            deltaA /= samples;

            cudafloat learningRate = UpdateLearningRate(learningRateA, lastDeltaWithoutLearningMomentumA, deltaA, i, u, d);
            UpdateWeight(learningRate, momentum, deltaA, lastDeltaA, lastDeltaWithoutLearningMomentumA, a, i);
        }
    }

    // Update b
    if (i == 0 && j < J) {
        deltaB /= samples;
```

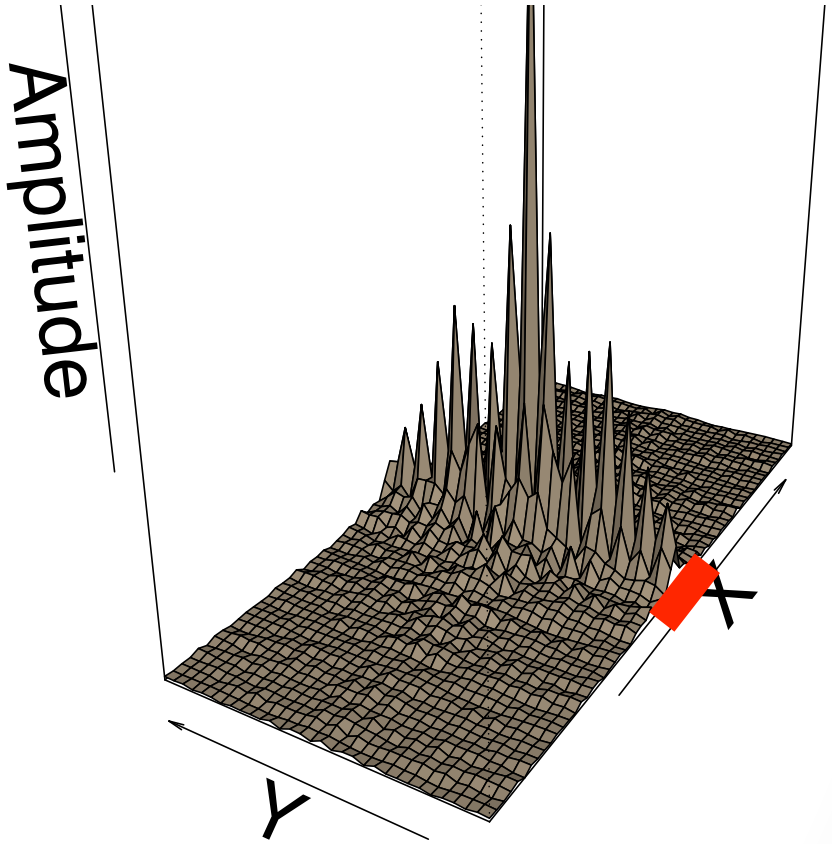
```
class class_attribute(PythonStructural, Element): pass
class expression_value(PythonStructural, Element): pass
class attribute(PythonStructural, Element): pass

# Structural Support Elements
# ---------------------------

class parameter_list(PythonStructural, Element): pass
class parameter_tuple(PythonStructural, Element): pass
class parameter_default(PythonStructural, TextElement): pass
class import_group(PythonStructural, TextElement): pass
class import_from(PythonStructural, TextElement): pass
class import_name(PythonStructural, TextElement): pass
class import_alias(PythonStructural, TextElement): pass
class docstring(PythonStructural, Element): pass

# ================
#  Inline Elements
# ================

# These elements cannot become references until the second
# pass.  Initially, we'll use "reference" or "name".

class object_name(PythonStructural, TextElement): pass
class parameter_list(PythonStructural, TextElement): pass
class parameter(PythonStructural, TextElement): pass
class parameter_default(PythonStructural, TextElement): pass
class class_attribute(PythonStructural, TextElement): pass
class attribute_tuple(PythonStructural, TextElement): pass
```

DFT Example (comments)



DFT Example (comments)





Alignment Features

```
wxSCHEDULER_DAILY      = 1
wxSCHEDULER_WEEKLY     = 2
wxSCHEDULER_MONTHLY    = 3
wxSCHEDULER_TODAY      = 4
wxSCHEDULER_TO_DAY     = 5
wxSCHEDULER_PREV       = 6
wxSCHEDULER_NEXT       = 7
wxSCHEDULER_PREVIEW    = 8
```

- Identify 3+ lines with same token/token or token/whitespace transitions.
- Record number and length of matches.

Linguistic Features

- Average dictionary words in identifiers
  - Underscore-separated words
  - CamelCase
  - Prefix and suffix

General Readability Metric

1. New model.
   - Buse baseline features
   - Additional visual features
2. **Ground truth from a large human study.**
3. Combine and evaluate.

Ground-Truth Survey

- Similar backgrounds (all UVa students).
- Single programming language (Java).
- Short code samples (4 – 13 lines).

Ground-Truth Survey

- ~~Similar backgrounds (all UVa students).~~
- Single programming language (Java).
- Short code samples (4 – 13 lines).

Ground-Truth Survey

- ~~Similar backgrounds (all UVa students).~~
- Diverse backgrounds:
  - Udacity students: beginners, professionals learning Python
  - reddit users: forum on programming

- Single programming language (Java).

- Short code samples (4 – 13 lines).

Ground-Truth Survey

- Diverse backgrounds: Udacity students, reddit users.
- ~~Single programming language (Java).~~
- Short code samples (4 – 13 lines).

Ground-Truth Survey

- Diverse backgrounds: Udacity students, reddit users.

- ~~Single programming language (Java).~~
- Multiple languages: Java, Python, CUDA.

- Short code samples (4 – 13 lines).

Ground-Truth Survey

- Diverse backgrounds: Udacity students, reddit users.
- Multiple languages: Java, Python, CUDA.
- ~~Short code samples (4 – 13 lines).~~

Ground-Truth Survey

- Diverse backgrounds: Udacity students, reddit users.
- Multiple languages: Java, Python, CUDA.
- ~~Short code samples (4 – 13 lines).~~
- Three code sample lengths: 10, 30, and 50 lines.

Code Samples

- Top-ten most recently updated projects in SourceForge.
- 360 total code samples.
  - 120 samples from each language.
  - 120 samples of each length.
- Survey takers rated 20 randomly selected samples.
  - Syntax pre-highlighted on server.

Survey Summary

- Over **76,000** individual ratings (**6x larger**).
- Over **2,600** completed surveys (**21x larger**).

Category	Median (yrs)	> 1 year	> 5 years	> 10 years
Overall	8	2598	1972	1242
Java	2	1896	646	247
Python	1	1655	253	59
CUDA	0	181	8	2
School	3	2118	522	28
Industry	3	1808	1091	655

Survey Summary

- Over 76,000 individual ratings (6x larger).
- Over 2,600 completed surveys (21x larger).

Category	Median (yrs)	> 1 year	> 5 years	> 10 years
Overall	8	2598	1972	1242
Java	2	1896	646	247
Python	1	1655	253	59
CUDA	0	181	8	2
School	3	2118	522	28
Industry	3	1808	1091	655

Survey Summary

- Over 76,000 individual ratings (6x larger).
- Over 2,600 completed surveys (21x larger).

Category	Median (yrs)	> 1 year	> 5 years	> 10 years
Overall	8	2598	1972	1242
Java	2	1896	646	247
Python	1	1655	253	59
CUDA	0	181	8	2
School	3	2118	522	28
Industry	3	1808	1091	655

Survey Summary

- Over 76,000 individual ratings (6x larger).
- Over 2,600 completed surveys (21x larger).

Category	Median (yrs)	> 1 year	> 5 years	> 10 years
Overall	8	2598	1972	1242
Java	2	1896	646	247
Python	1	1655	253	59
CUDA	0	181	8	2
School	3	2118	522	28
Industry	3	1808	1091	655

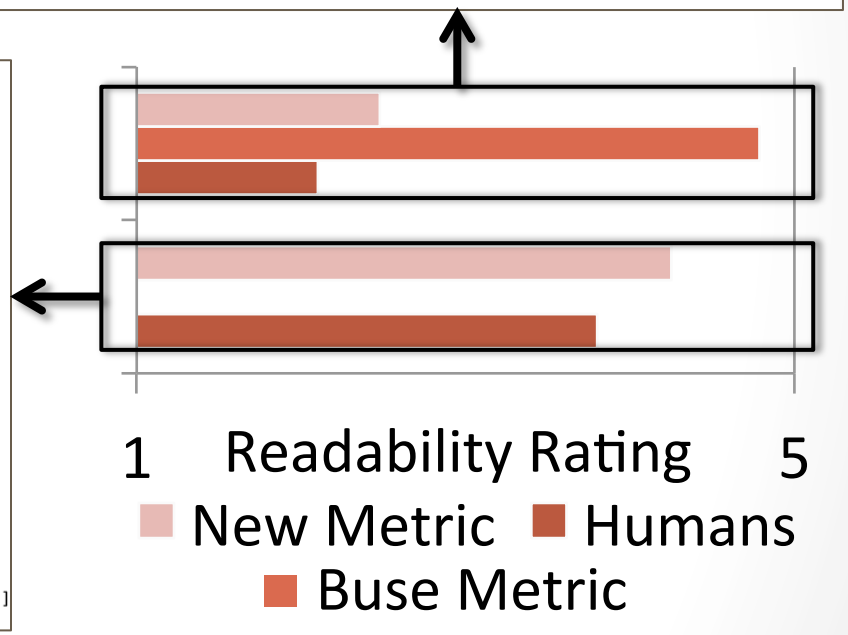
General Readability Metric

1. New model.
   - Buse baseline features
   - Additional visual features
2. Ground truth from a large human study.
3. **Combine and evaluate.**

Example Readability

```
        //float *attenuationIntegralPlaneArray_d;    //stores partial integral on planes parallel to the camera
        //CUDA_SAFE_CALL(cudaMalloc((void **)&attenuationIntegralPlaneArray_d, img->dim[1]*img->dim[3]*sizeof(float)));

        et_line_integral_attenuated_gpu_kernel <<<G1,B1>>> (*d_activity, *d_attenuation, currentCamPointer);

        CUDA_SAFE_CALL(cudaThreadSynchronize());
}
```

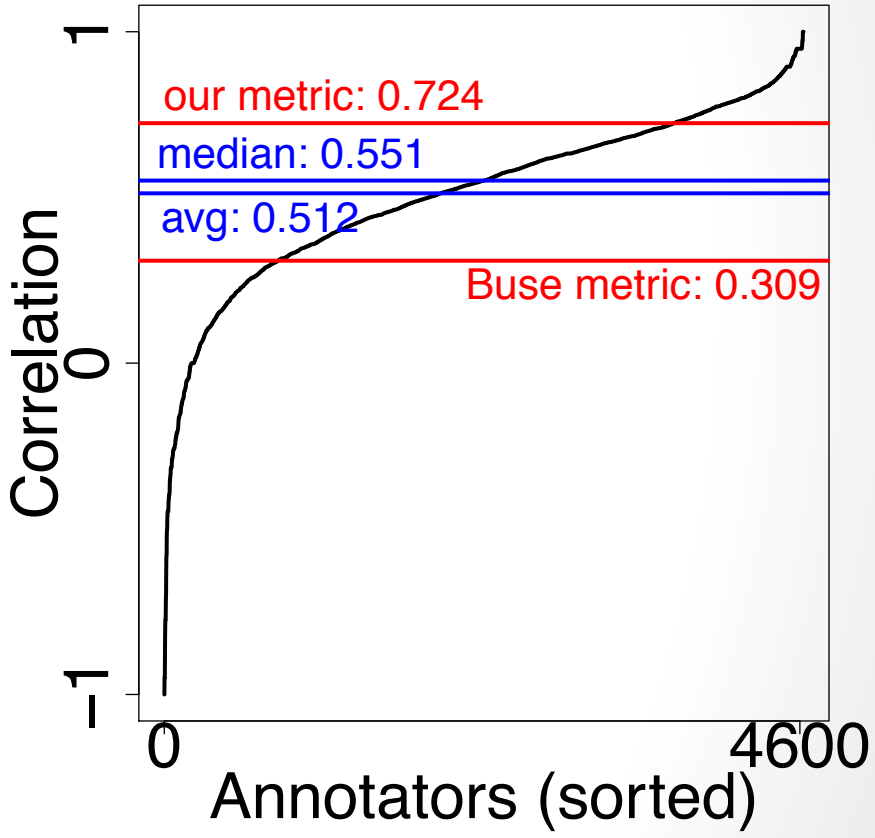
```
def handleBlockQuote(node):
    result = BlockQuoteDitem(node.nodeName)
    result.children = processChildren(node)
    return result

def handleList(node):
    result = ListDitem(node.nodeName)
    result.children = processChildren(node)
    return result

def handleListItem(node):
    result = ListItemDitem(node.nodeName)
    result.children = processChildren(node)
    return result

def handleTable(node):
    result = TableDitem(node.nodeName)
    # Ignore table contents that are not tr
    result.children = [x
        for x in processChildren(node) if x.type=='tr']
    return result

def handleTr(node):
    result = TrDitem(node.nodeName)
    # Ignore tr contents that are not th or td
    result.children = [x
        for x in processChildren(node) if x.type in ('th', 'td')]
    return result
```



Annotator Agreement

- Spearman correlation: Agreement on ordering

Score	Meaning
+1	Perfect agreement
0	No relationship
-1	Perfect disagreement



Impact of New Features

- How much improvement is due to our new features?
- **Re-train** Buse metric with our survey results.
- Compare our metric (**old + new features**) to Buse metric (**old features only**)

Impact of New Features

- Compute **f-measure**:

$$f = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

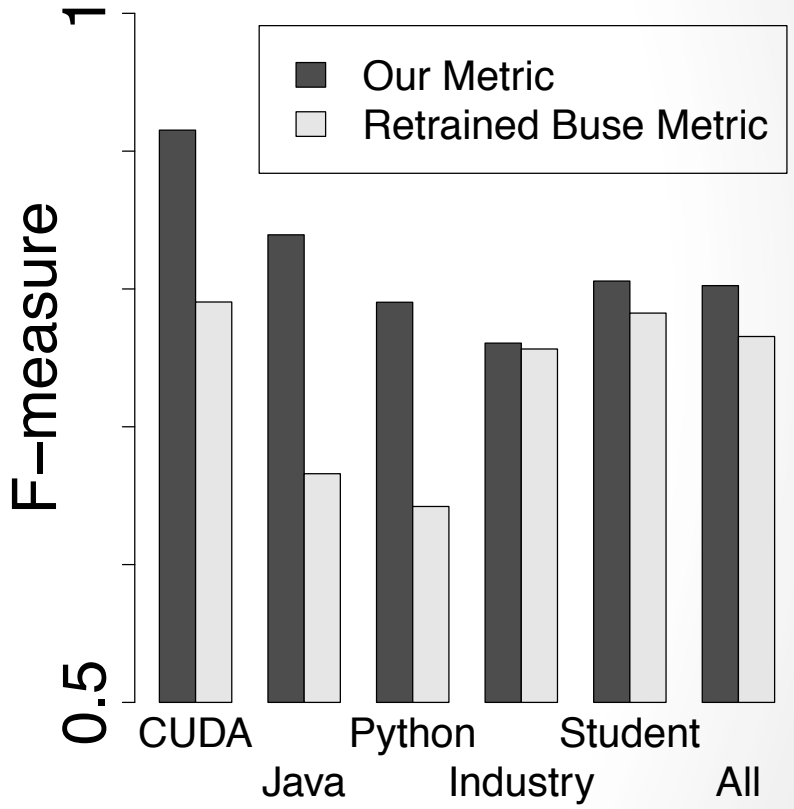
$$\text{precision} = \frac{TP}{TP + FP}$$

$$\text{recall} = \frac{TP}{TP + FN}$$

		Predicted ✔	Predicted ✖
Actual	✔	TP	FN
	✖	FP	TN

Head-to-Head F-Measure

- Multi-language
  - **5%** improvement
- Single-language
  - **16-26%** improvement



Predictors of Readability

All Languages, All Lengths

Category	Description	+/-
Syntax	Line Length	-
Syntax	Long lines	-
Visual	Operator area	-
Structural	1D DFT of syntax	-
Visual	2D DFT of comments	+
Visual	String area to keyword area	+
Alignment	Min alignment length	+

5+ Years Industry Experience

Category	Description	+/-
Syntax	Long lines	-
Syntax	Whitespace	-
Visual	Comment area	+
Structural	1D DFT of whitespace	-

Predictors of Readability

All Languages, All Lengths

Category	Description	+/-
Syntax	Line Length	-
Syntax	Long lines	-
Visual	Operator area	-
Structural	1D DFT of syntax	-
Visual	2D DFT of comments	+
Visual	String area to keyword area	+
Alignment	Min alignment length	+

5+ Years Industry Experience

Category	Description	+/-
Syntax	Long lines	-
Syntax	Whitespace	-
Visual	Comment area	+
Structural	1D DFT of whitespace	-

Predictors of Readability

Java

Category	Description	+/-
Structural	1D DFT of whitespace	-
Syntax	Long lines	-
Syntax	Lines between identifiers	-
Syntax	Keywords	+
Structural	1D DFT of syntax	-

Python

Category	Description	+/-
Syntax	Identifiers	-
Linguistic	Identifier components	-
Visual	Operator area to keyword area	-
Structural	Operator to identifier tokens	+
Structural	1D DFT of syntax	-

Conclusion

- **Visual and spatial features** can significantly improve the accuracy of readability metrics.
  - **Different features** are more predictive for **different languages**.
- **Largest** human study of readability ratings to date.
  - Survey data is available **online.**

Questions?