Math Spotting in Technical Documents Using Handwritten Queries

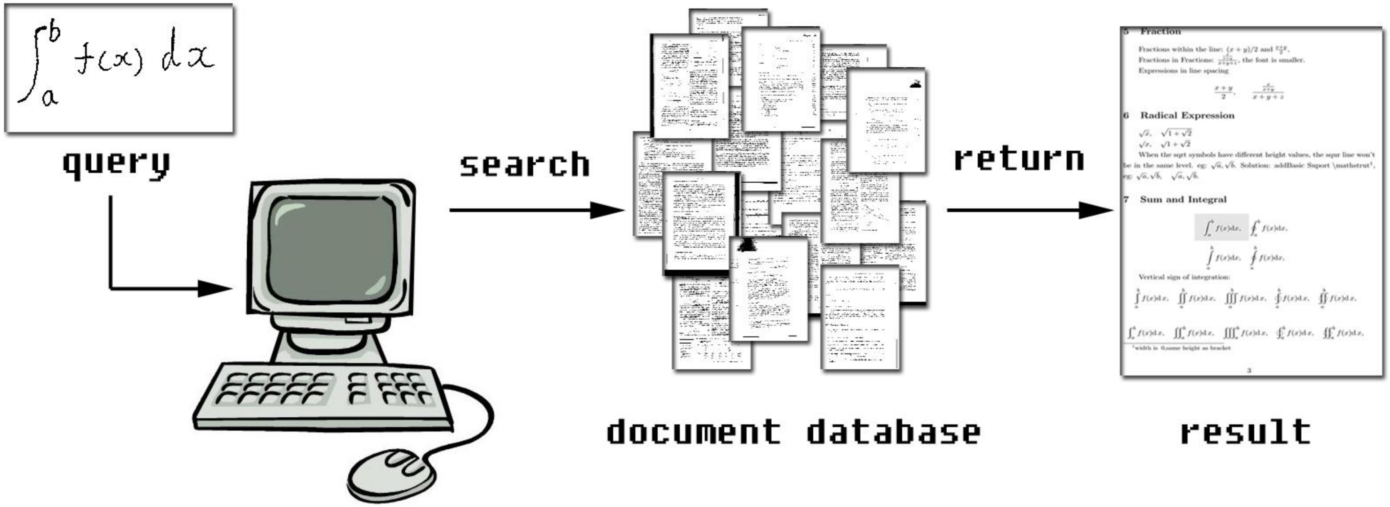
Li Yu and Richard Zanibbi

Document and Pattern Recognition Lab

Rochester Institute of Technology, NY, USA

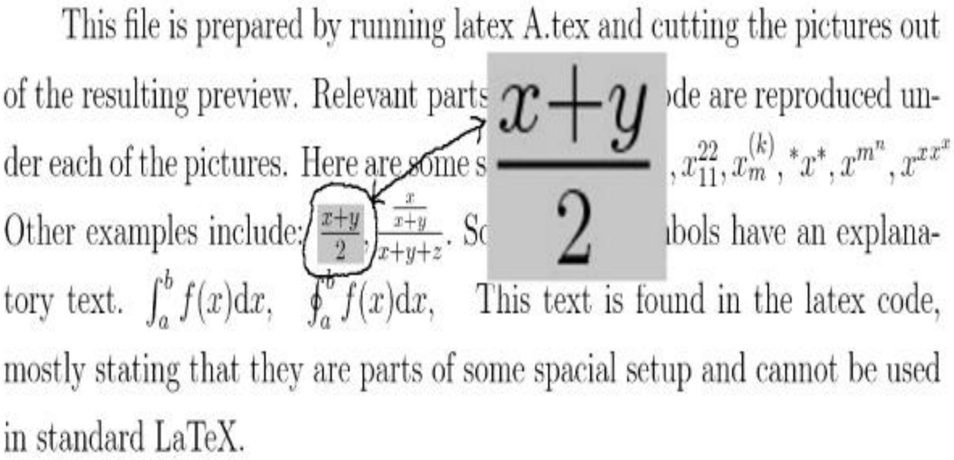
lxy1783@rit.edu, rlaz@cs.rit.edu

Math spotting



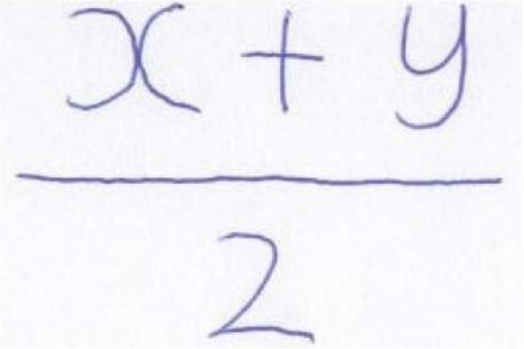
- OCR (optical character recognition) avoided
- Structure feature & Visual feature

Document image and query image



This file is prepared by running latex A.tex and cutting the pictures out of the resulting preview. Relevant parts $\frac{x+y}{2}$ ode are reproduced under each of the pictures. Here are some s , $x_{11}^{22}, x_m^{(k)}, {}^*x^*, x^{m^n}, x^{x^{x^x}}$ Other examples include: $\frac{x+y}{2}$, $\frac{\frac{x}{x+y}}{x+y+z}$. Sc mbols have an explanatory text. $\int_a^b f(x)\mathrm{d}x$, $\oint_a^b f(x)\mathrm{d}x$, This text is found in the latex code, mostly stating that they are parts of some spacial setup and cannot be used in standard LaTeX.

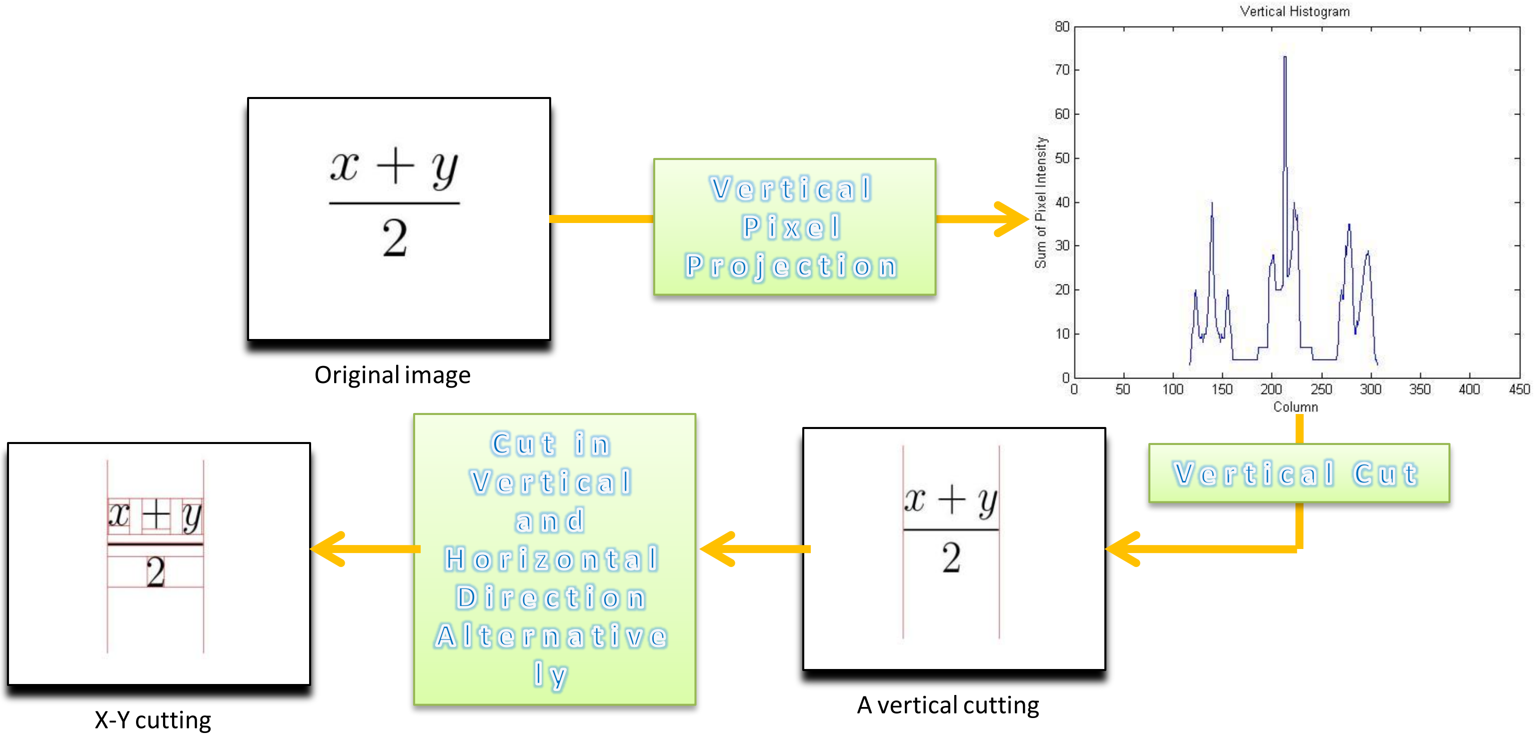
Document image



$\frac{x+y}{2}$

Query image

X-Y cutting



Original image

Vertical Pixel Projection

Vertical Cut

A vertical cutting

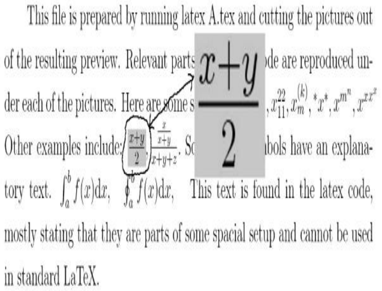
Cut in Vertical and Horizontal Direction Alternatively

X-Y cutting

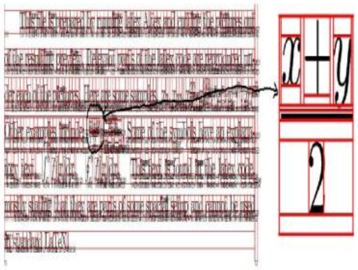
G. Nagy and S. Seth, “Hierarchical representation of optically scanned documents,” *Proc. of ICPR, (1984) 347-349.*

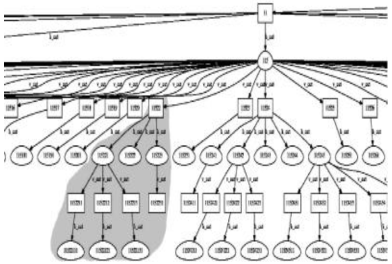
X-Y cut and X-Y tree

Page

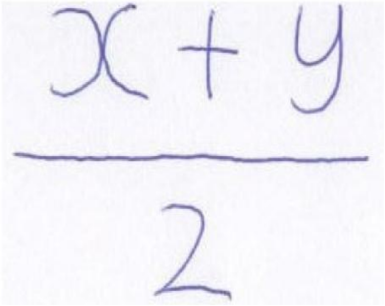


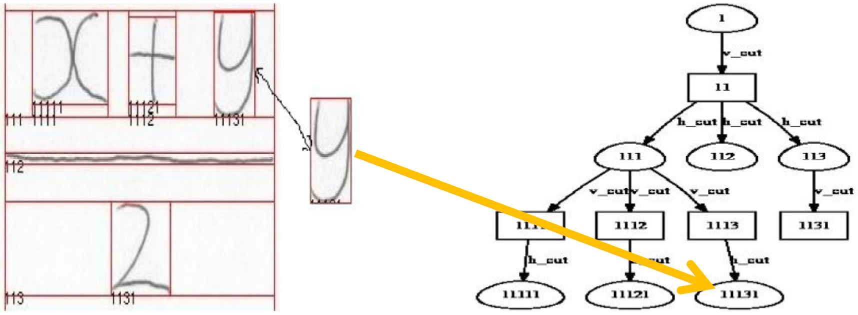
This file is prepared by running latex A.tex and cutting the pictures out of the resulting preview. Relevant parts $\frac{x+y}{2}$ de are reproduced under each of the pictures. Here are some s , $x_{11}^{22}, x_m^{(k)}, {}^*x^*, x^{m^n}, x^{xx^x}$ Other examples include: $\frac{x+y}{2}$, $\frac{\frac{x}{x+y}}{x+y+z}$. So bols have an explanatory text. $\int_a^b f(x)\mathrm{d}x$, $\oint_a^b f(x)\mathrm{d}x$, This text is found in the latex code, mostly stating that they are parts of some spacial setup and cannot be used in standard LaTeX.





Query





Images

X-Y cutting

X-Y trees

Sub-tree matching

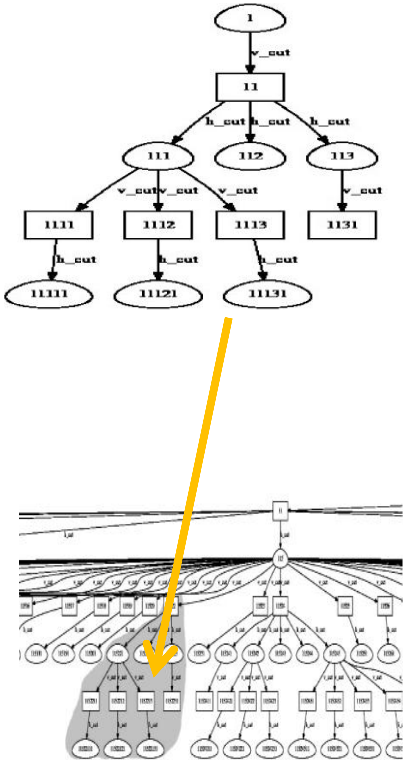
- What if we can find a matched sub-tree in the page tree?

- What we want?

  Speed & Accuracy

- Problems?

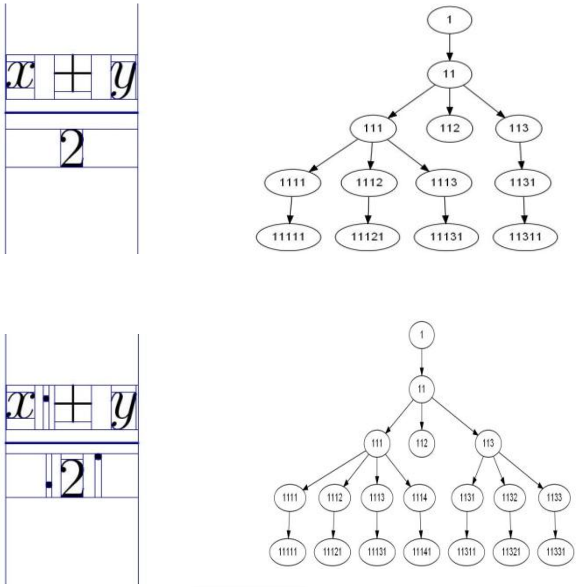
  Inexact matching



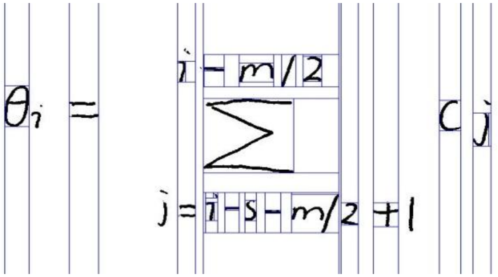
X-Y tree for query

X-Y tree for page

Noise and “Bad Division”

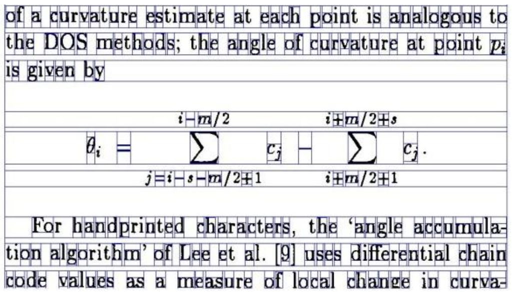


Noise



$$\theta_i = \sum_{j=i-s-m/2+1}^{i-m/2} c_j$$

Cutting in Query



of a curvature estimate at each point is analogous to the DOS methods; the angle of curvature at point p_i is given by

$$\theta_i = \sum_{j=i-s-m/2+1}^{i-m/2} c_j - \sum_{i+m/2+1}^{i+m/2+s} c_j .$$

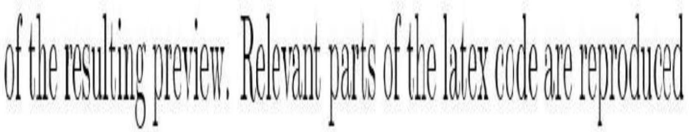
For handprinted characters, the ‘angle accumulation algorithm’ of Lee et al. [9] uses differential chain code values as a measure of local change in curva-

Cutting in Page

Bad Division

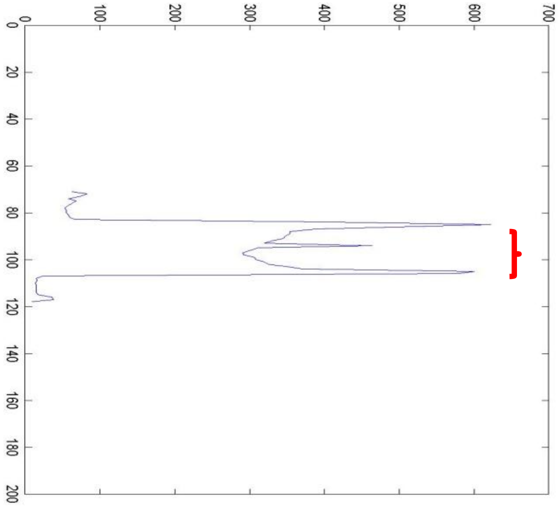
- **Avoid noise**
- **Control the way in which regions are cut**
- **Rectangles whose size smaller than thresholds will be ignored**

Thresholds



of the resulting preview. Relevant parts of the latex code are reproduced

One Line In Document

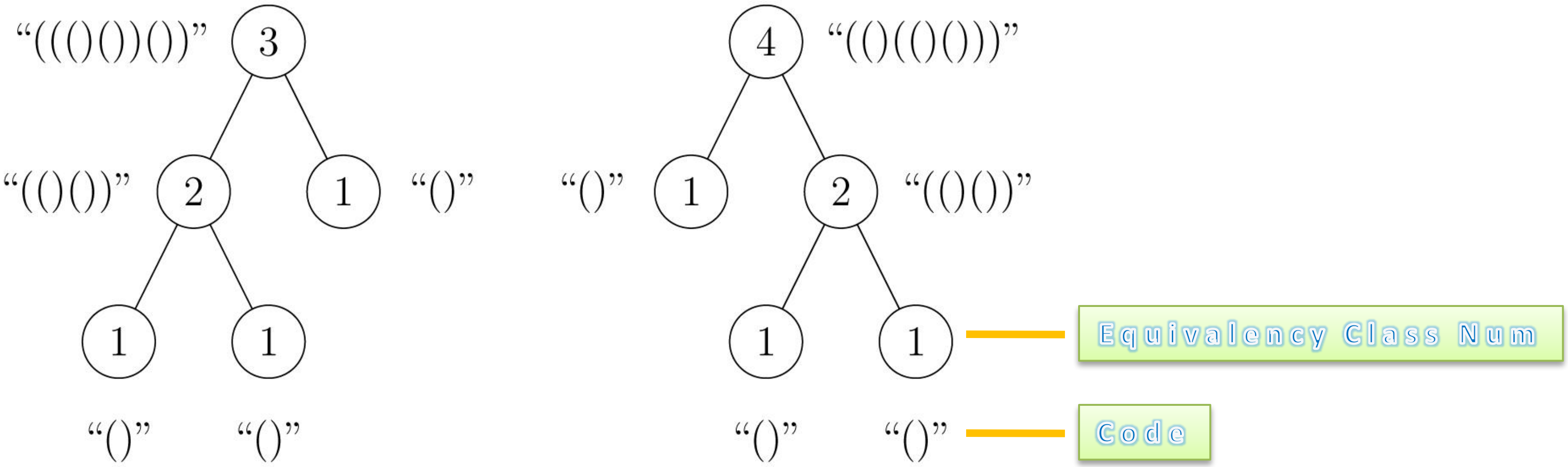


Width of Peaks

Horizontal Projection

- Dominant height/width of characters
- C_h = Mode(h_1,h_2,…h_n), where h_n represents the heights of lines in one page
- W_h = Mode(W_1,W_2,…W_n), where W_n represents the widths of blank spaces in one line
- Scaled linearly based on the current region’s height and width

Equivalency Class



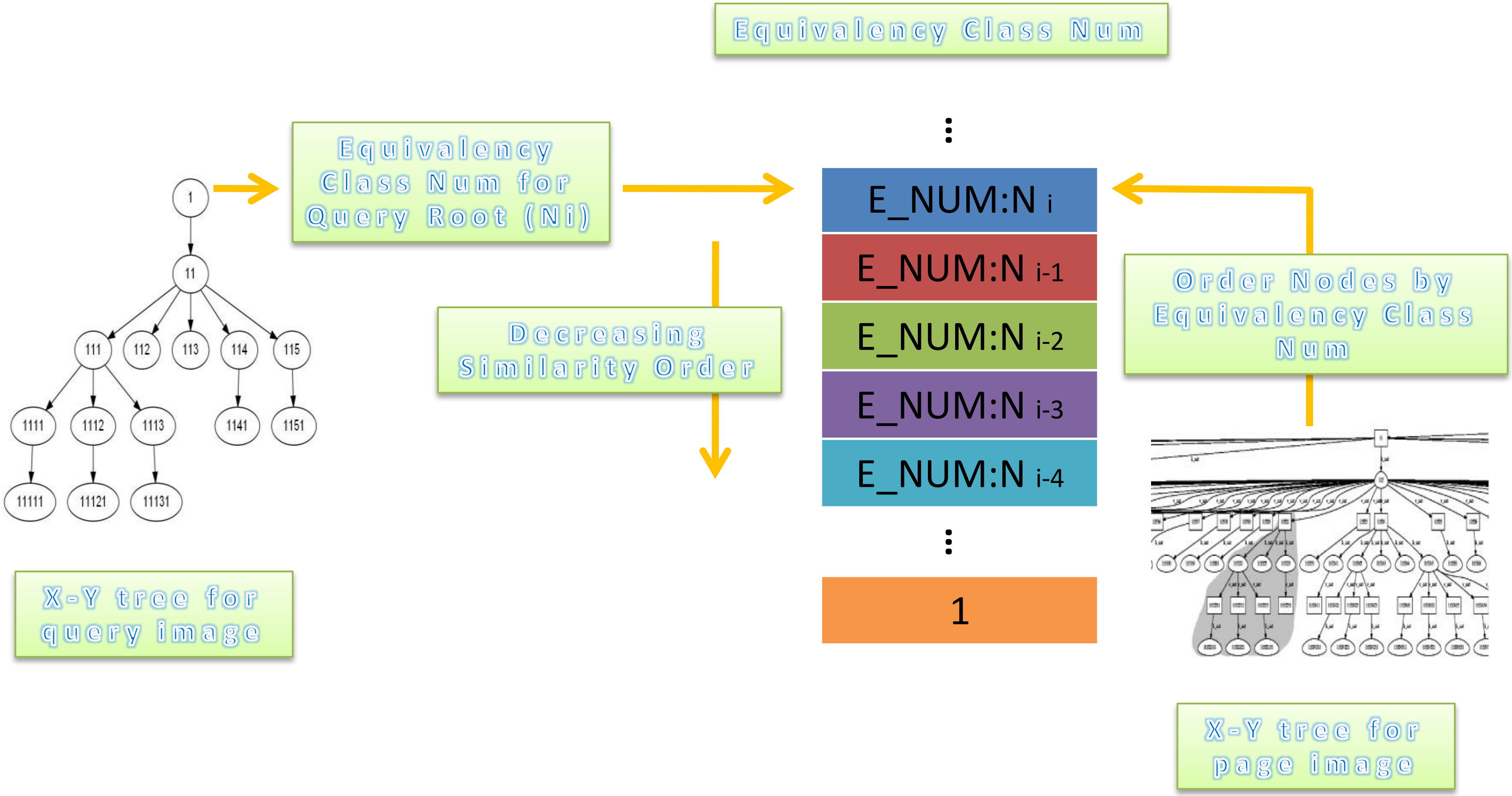
Equivalency Class Num

Code

- **Two trees have same code (equivalence class number) if and only if they are isomorphic**
- **Bottom-up algorithm with linear time in the size of the trees**

A.V. Aho, J.E. Hopcroft, and J.D. Ullman. The design and analysis of computer algorithms. Addison Wesley, Reading, Mass, 1974.

Ranking by Equivalency Class



Equivalency Class Num

Equivalency Class Num for Query Root (Ni)

Decreasing Similarity Order

Order Nodes by Equivalency Class Num

⋮

E_NUM:N$_{i}$

E_NUM:N$_{i-1}$

E_NUM:N$_{i-2}$

E_NUM:N$_{i-3}$

E_NUM:N$_{i-4}$

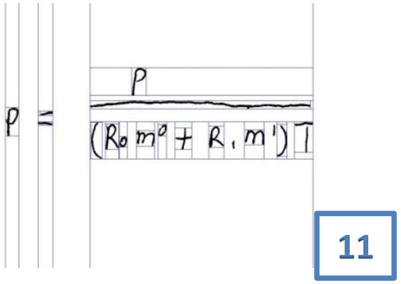
⋮

1

X-Y tree for query image

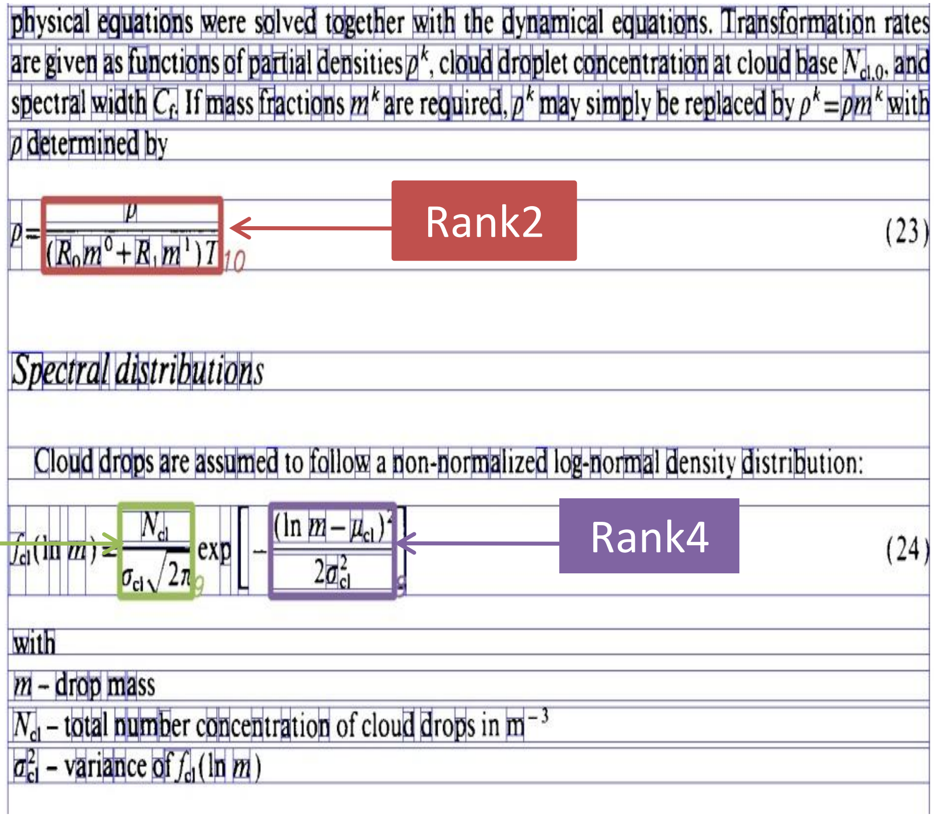
X-Y tree for page image

Ranking by Equivalency Class



Query

- **The query are included in the page**



physical equations were solved together with the dynamical equations. Transformation rates are given as functions of partial densities ρ^k, cloud droplet concentration at cloud base $N_{cl,0}$, and spectral width C_f. If mass fractions m^k are required, ρ^k may simply be replaced by $\rho^k = \rho m^k$ with ρ determined by

$$\rho = \frac{p}{(R_0 m^0 + R_1 m^1)T} \qquad (23)$$

Rank2

Spectral distributions

Cloud drops are assumed to follow a non-normalized log-normal density distribution:

$$f_{cl}(\ln m) = \frac{N_{cl}}{\sigma_{cl}\sqrt{2\pi}} \exp\left[-\frac{(\ln m - \mu_{cl})^2}{2\sigma_{cl}^2}\right] \qquad (24)$$

Rank3

Rank4

with

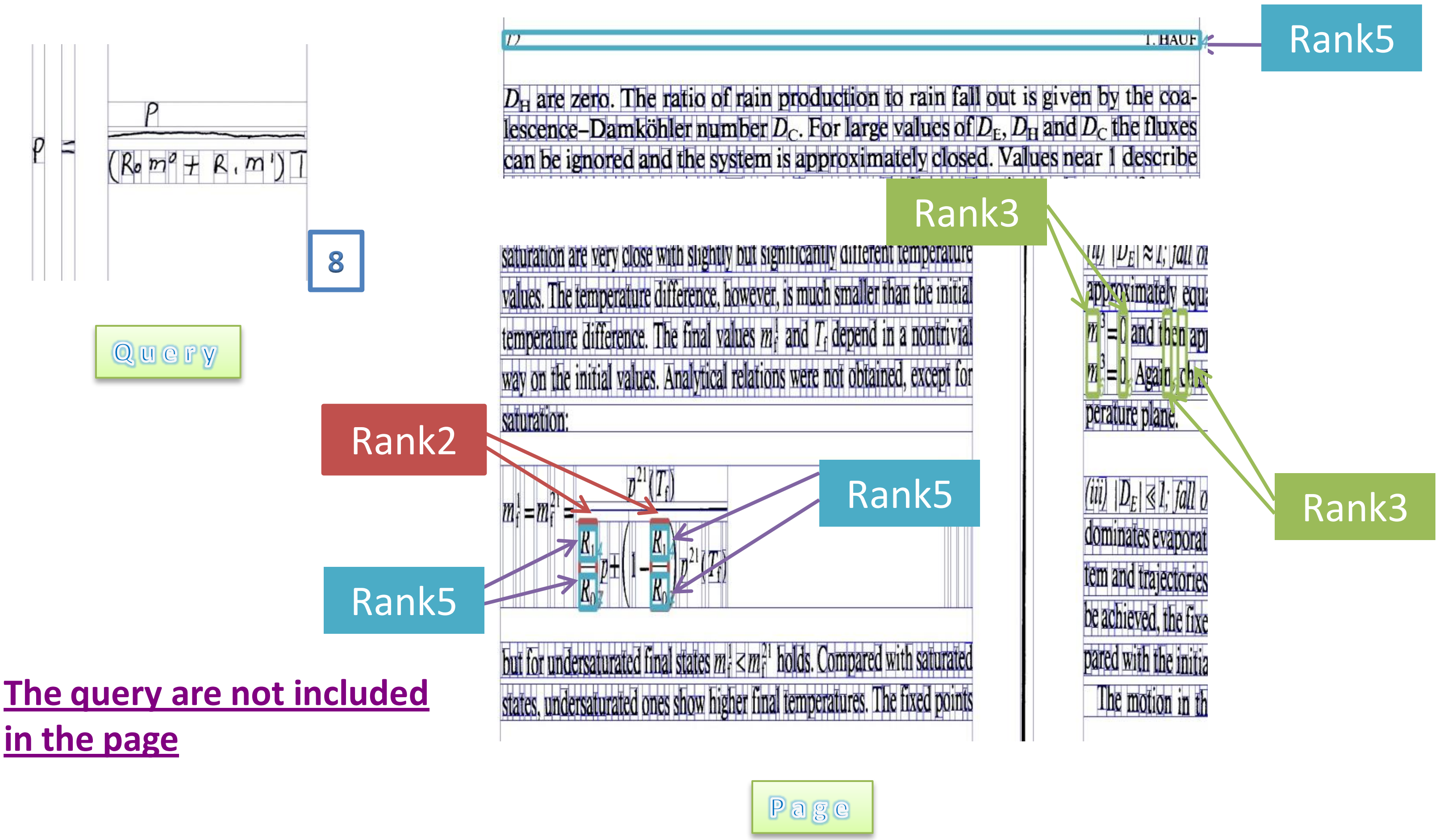
m – drop mass

N_{cl} – total number concentration of cloud drops in m^{-3}

σ_{cl}^2 – variance of $f_{cl}(\ln m)$

Page

Ranking by Equivalency Class



12 T. HAUF

D_H are zero. The ratio of rain production to rain fall out is given by the coalescence–Damköhler number D_C. For large values of D_E, D_H and D_C the fluxes can be ignored and the system is approximately closed. Values near 1 describe

saturation are very close with slightly but significantly different temperature values. The temperature difference, however, is much smaller than the initial temperature difference. The final values m_f^1 and T_f depend in a nontrivial way on the initial values. Analytical relations were not obtained, except for saturation:

$$m_f^1 = m_f^{21} = \frac{p^{21}(T_f)}{\frac{R_1}{R_0}p + \left(1 - \frac{R_1}{R_0}\right)p^{21}(T_f)}$$

but for undersaturated final states $m_f^1 < m_f^{21}$ holds. Compared with saturated states, undersaturated ones show higher final temperatures. The fixed points

(ii) $|D_E| \approx 1$; fall o
approximately equ
$m^3 = 0$ and then ap
$m^3 = 0$. Again, ch
perature plane.

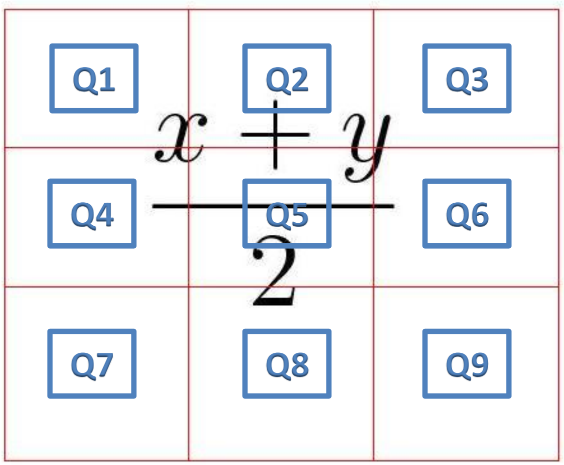
(iii) $|D_E| \ll 1$; fall o
dominates evaporat
tem and trajectories
be achieved, the fixe
pared with the initia
The motion in th

- **The query are not included in the page**

Other Rankings

- Ranking by Number of Nodes:
  - Divide the page nodes into bins based on their size.
  - Start with the size of the query root.
  - Search for the page nodes in decreasing size order.
- Ranking by both equivalence class number and number of nodes:
  - Generate the equivalence class number for both query and page.
  - Start with the query root and by decreasing order.
  - Find all the exact sub-matches in the page tree.

Visual Feature



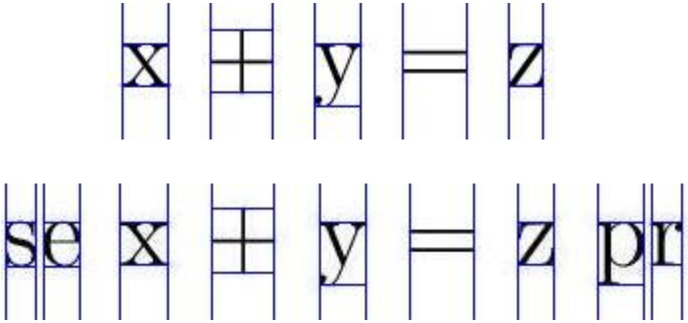
$$Dis\tan ce=\sum_{i=1}^{9}\frac{(Q_i-P_i)^2}{P_i}$$

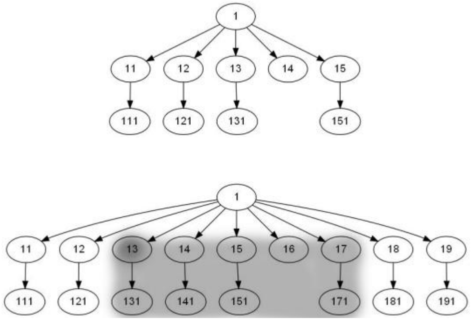
Where Qi and Pi represents the sum of pixel intensity in the sub-region in query and candidate respectively

- **Dividing the region into nine sub-regions and computing sum of pixel intensity respectively**
- **Ranking the candidates by decreasing visual similarity**

Problems and Future work

- **The situation where the target is “scattered” in the page.**





- q03vp03.htm

Problems and Future work

- **Different Rankings**
- **More visual features && comparison**
- **Document image indexing**

Thanks

Question?