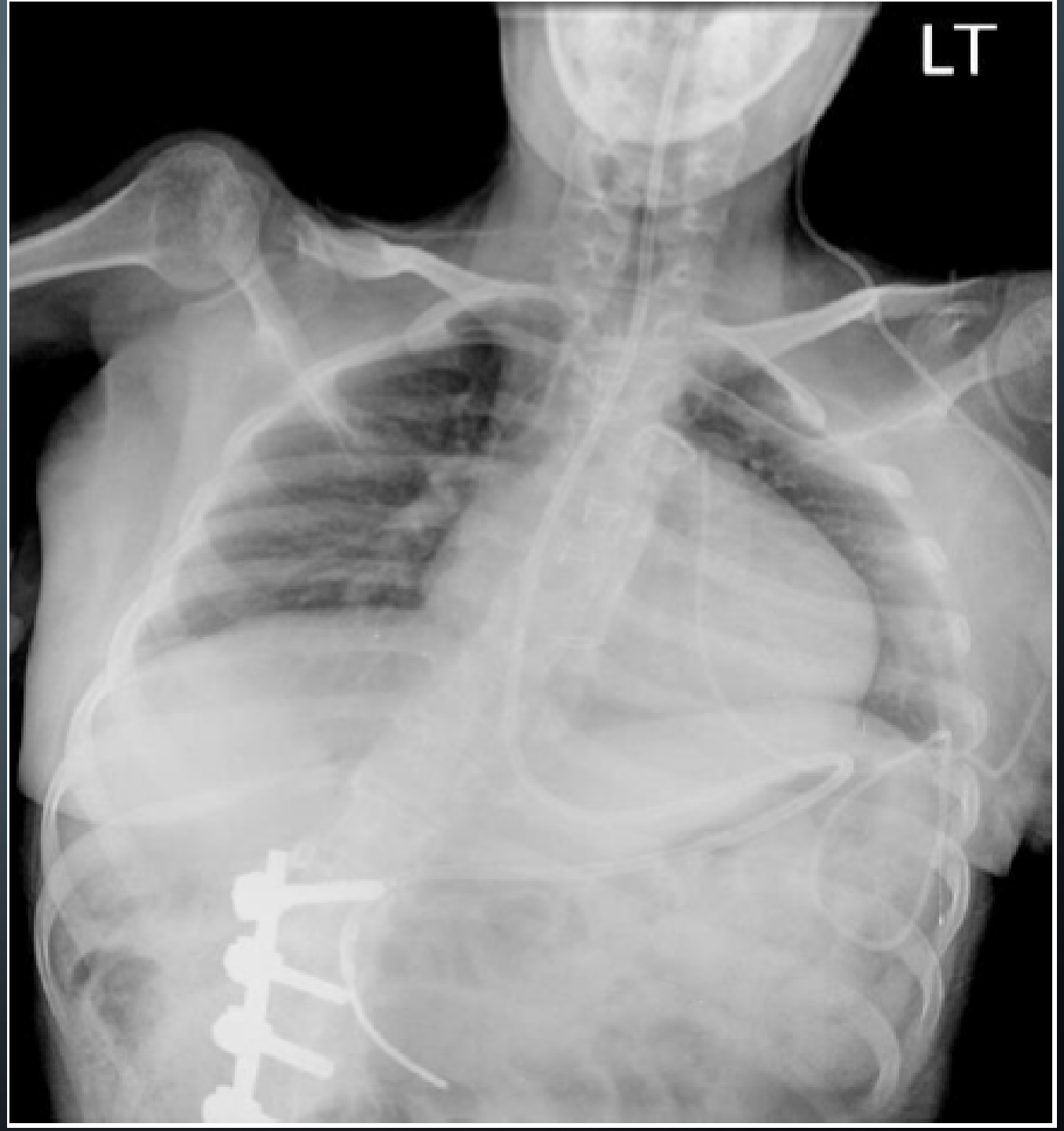
WHAT DO YOU SEE?



GRAPHICAL PROGRAMMING

ARE WE SEEING THE SAME THING?

AZADEH (AZZY) ASSADI

CSC 2125H1S

DR. M. CHECHIK

WHY LOOKING ISN'T ALWAYS SEEING: READERSHIP SKILLS AND GRAPHICAL PROGRAMMING

- Objective:
  - How programmers actually use different representations
  - Challenges the assumption that graphical representations are unproblematically more 'transparent', more accessible, comprehensible, and memorable than textual
  - No single representation is a panacea, but that we need to identify appropriate criteria for choosing representational 'horses' for cognitive 'courses'

GOOD GRAPHICS RELIES ON SECONDARY NOTATION

- Secondary notation comes from analog mapping used by electrical engineers
- "The use of layout and perceptual cues (element as such as adjacency, clustering, white space, labelling, …) to clarify information (such as structure, function or relationships) or to give hints to the reader" (Petre, M and Green T.R.G., 1992)
- This is NOT part of the formal system
- Subject to personal style and individual skills
  - Experience and personal skill impact the graphical representation based on how secondary notations are used
  - E.g. Novice's design considered more difficult to comprehend b/c of the layout – different grouping (not logical, confused signal flow, neglected conventions, etc.)
  - Potential for miss-cues if notations not used correctly

WHAT IS THE DIFFERENCE HERE? GRAPHIC OR TEXT?

3a

```
while ((used!=1) || (a[0] !=1))
{ if (a[0] & 0x1) { k=1; for (c = 0; c <= used; c++)
{ a[c] = 3 * a[c] + k; k = a[c] / 10; a[c] = a[c] % 10;}
if (a[used])
{ used++; if (used >= 72)
{ printf ("Run out of space\n"); exit(1);}}
} else {k = 0;
for (c = used - 1; c >= 0; c--)
{ a[c] = a[c] + 10*k; k = a[c] & 0x1; a[c] = a[c] >>1;}
if (a[used - 1] == 0) used--; }count++;
}
```

3b

```
while ((used!=1)||(a[0] !=1))
    { if(a[0]&0x1)
        { k=1;
          for (c=0; c<=used; c++)
            { a[c]=3*a[c]+k;
              k=a[c]/10;
              a[c]=a[c]%10;
            }
        if (a[used])
            { used++;
              if (used>=72)
                { printf ("Run out of space\n");
                  exit(1);
                }
            }
        }
    else
        { k=0;
          for(c=used-1; c>=0; c--)
            { a[c]=a[c]+10*k;
              k=a[c]&0x1;
              a[c]=a[c]>>1;
            }
          if (a[used-1]==0) used--;
        }
      count++;
    }
```

EXPERIENCE INFLUENCES VIEWING STRATEGY

- Reading comprehension using various graphical and textual representations of nested conditional structures
- Results:
  - Graphics were significantly slower than text in all conditions
  - Strategy differences were strongly related to prior experience (more experience = more flexible and appropriate performance relative to the question and secondary notation)
- Petri representations (Moher et al, 1993)
- Similar experiment
- Results:
  - No instances in which graphical representations out-performed their textual counterparts
    - Performance strongly depended on layout of the Petri nets
  - Difference due to secondary notations

NOVICES VS EXPERTS: OBSERVED STRATEGY DIFFERENCES

NOVICE

- Rigid strategies and more chaotic
- Did not account for the structure of task or of the notation
- Mismatch of strategy and notation
- Lack of secondary notation therefore tended not to match strategy to the question
- ***Confused during a reading or mistrusted their responses***

EXPERT

- Consistent as a group with different subjects choosing similar strategies
- Used practice trials to identify strategy for style of question
- Strategy matched the question
- Used text to guide their graphic reading when presented together.

DETERMINING WHAT IS RELEVANT

NOVICE

- Distracted by syntax or surface features
- A visible symbol is interpreted as a relevant symbol – if its there it must be relevant
- Uneasy about completeness of their review
- Unable to satisfy themselves that they had read the diagram thoroughly
- Unable to recognize secondary notational cues

EXPERT

- Better able to develop overviews and significance of detail
- Strategies based on different information
- Used secondary notation cues to limit search to limited portions of the structure
- Acquired abilities to ‘see’ what is ‘invisible to non-experts
- Graphics reading was uniformly slower than text

WHAT MAKES EXPERTS EXPERT

- Not just domain knowledge but knowing how and when to use it
- Knowing that at times certain things are left out to capture other more relevant things – i.e. context specific inclusion and omission
- Secondary notations that are buried in experience and heuristics and rarely formalized or codified – i.e. experts just know
  - They just know expert's work from another or a novice
- They don't play by any hard and fast rules – i.e. apply and break rules in a systematic way while still maintain consistency of application that gives cues to structure and makes it understandable to other experts

WHY GRAPHICAL REPRESENTATIONS SO APPEALING?

- ? An alternative to text
- ? Higher level of description for the desired action. Ie. More info with less clutter – difficult to compare with text on par
- ? Provides 'gestalt' – i.e. insight into the structure as a whole? But programmers couldn't recognize structural similarities among graphical representations – did find it in textual representation
- ? Higher abstraction level – easier to derive a mental model if relations are captured.

Possibility of too literal transcription of the domain

- ? Accessible – cobol effect, pictures may seem less daunting but can take longer to read and easy to misunderstand if novice
- ? More comprehensible
- ? More memorable
- ? more fun – secondary notations are outside the rules so there is more freedom to 'play around'; illusion of accessibility

ROLE OF GRAPHICS IN PROGRAMMING NOTATION

TEXT

- Text = graphics with very limited vocabulary
- Precision of expression
- Doesn't rely on perceptual responses
- Easily translated from visual to other modes

GRAPHICS

- Graphics = unlimited vocabulary
- Lacks precision
- Impression of the whole
- Rules of interpretation not clearly defined

CONCLUSIONS

- Graphical readership is an acquired skill
  - Novices confuse visibility with relevance
  - Experts take advantage of secondary notation cues to enable them to recognize sub-term groupings to match patterns
  - Readership skills in perception and interpretation are learned

CONCLUSIONS

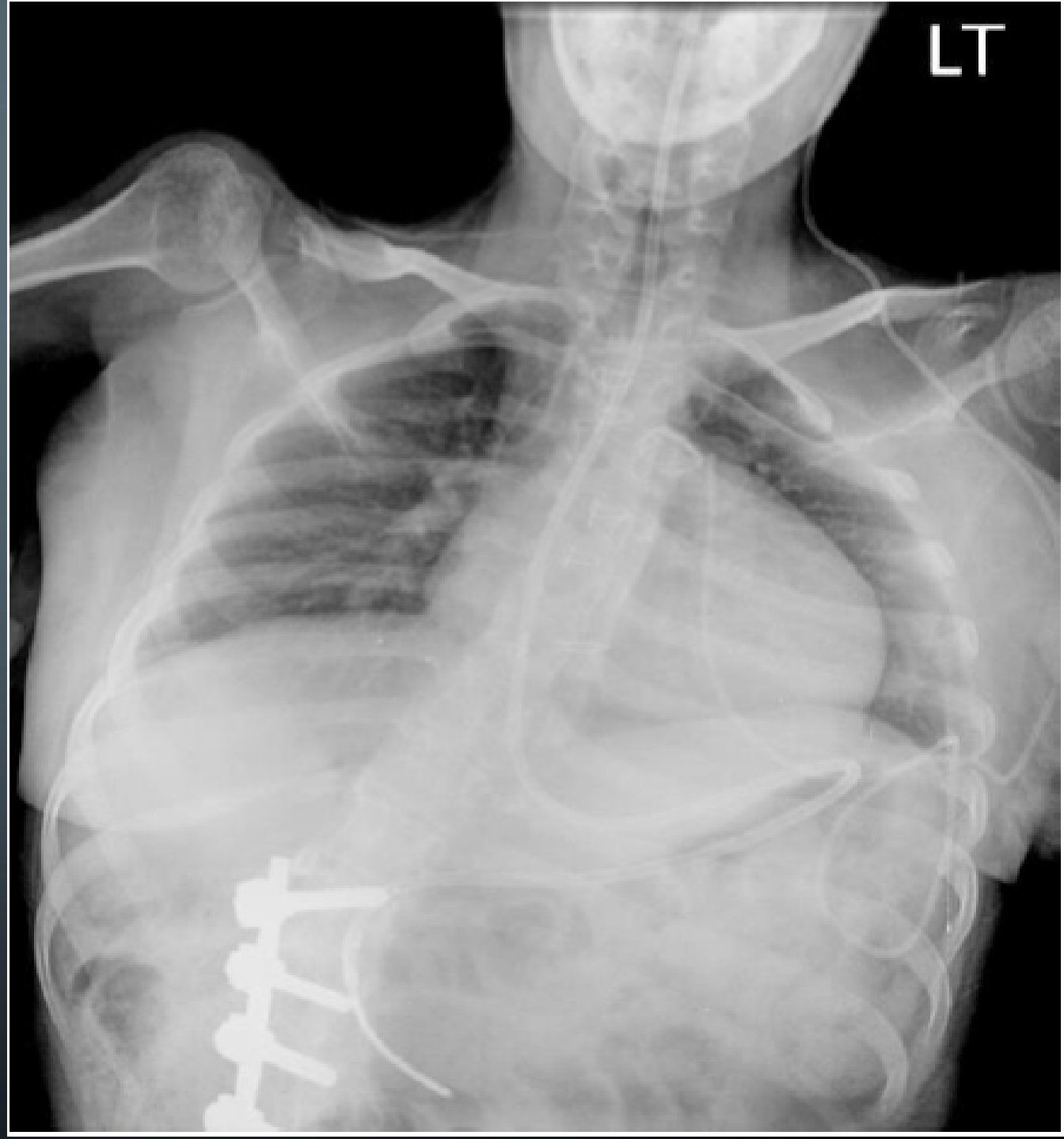
- Experts and novices have different notational needs
  - Experts should utilize expert languages with a broader scope of secondary notations as they would be more likely to benefit from it and more likely to create complex programs that would benefit from it
  - Novices need more constrained systems in which secondary notations are minimized to reduce potential for miss-cueing and misunderstanding
  - Less skilled groups are less likely to benefit from secondary notation
  - Graphics require readership and production skills the same way as text does

CONCLUSIONS

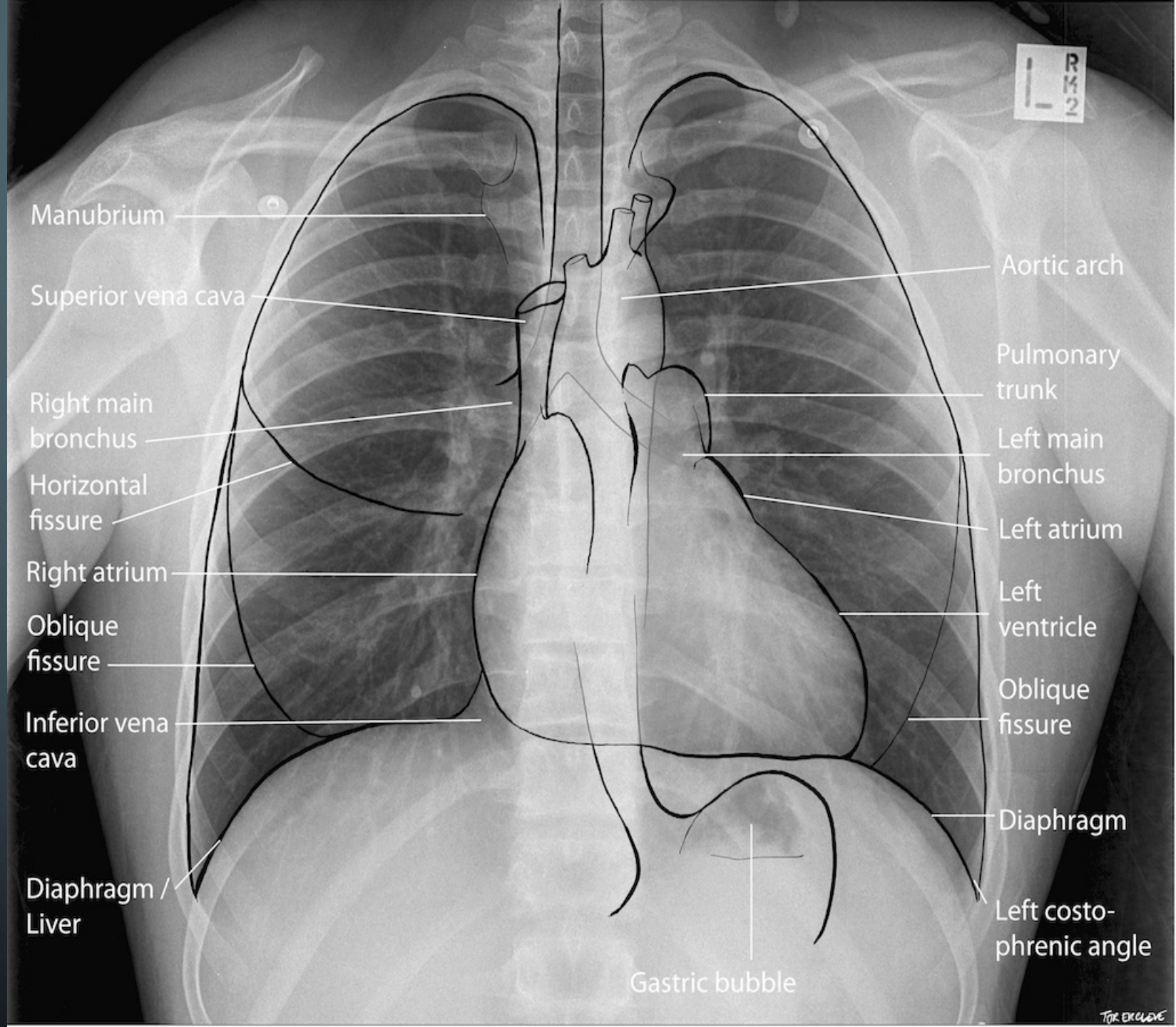
- Accept the bad with the good:
  - Individual skill and insight is key in understanding/recognizing cues
  - Need to be able to link perceptual cues to important information which requires guiding the reader with appropriate cues
  - There is no universal way of interpreting graphics
  - There is greater capacity to go wrong compared to text
  - Icons have the benefit of built in mnemonics but can become too detailed/complex (e.g Prograph) and can become too fragmented

CONTEXT: THIS IS A POST-OP FILM TAKEN ON A PATIENT WHO HAS JUST HAD A VP SHUNT PUT IN.

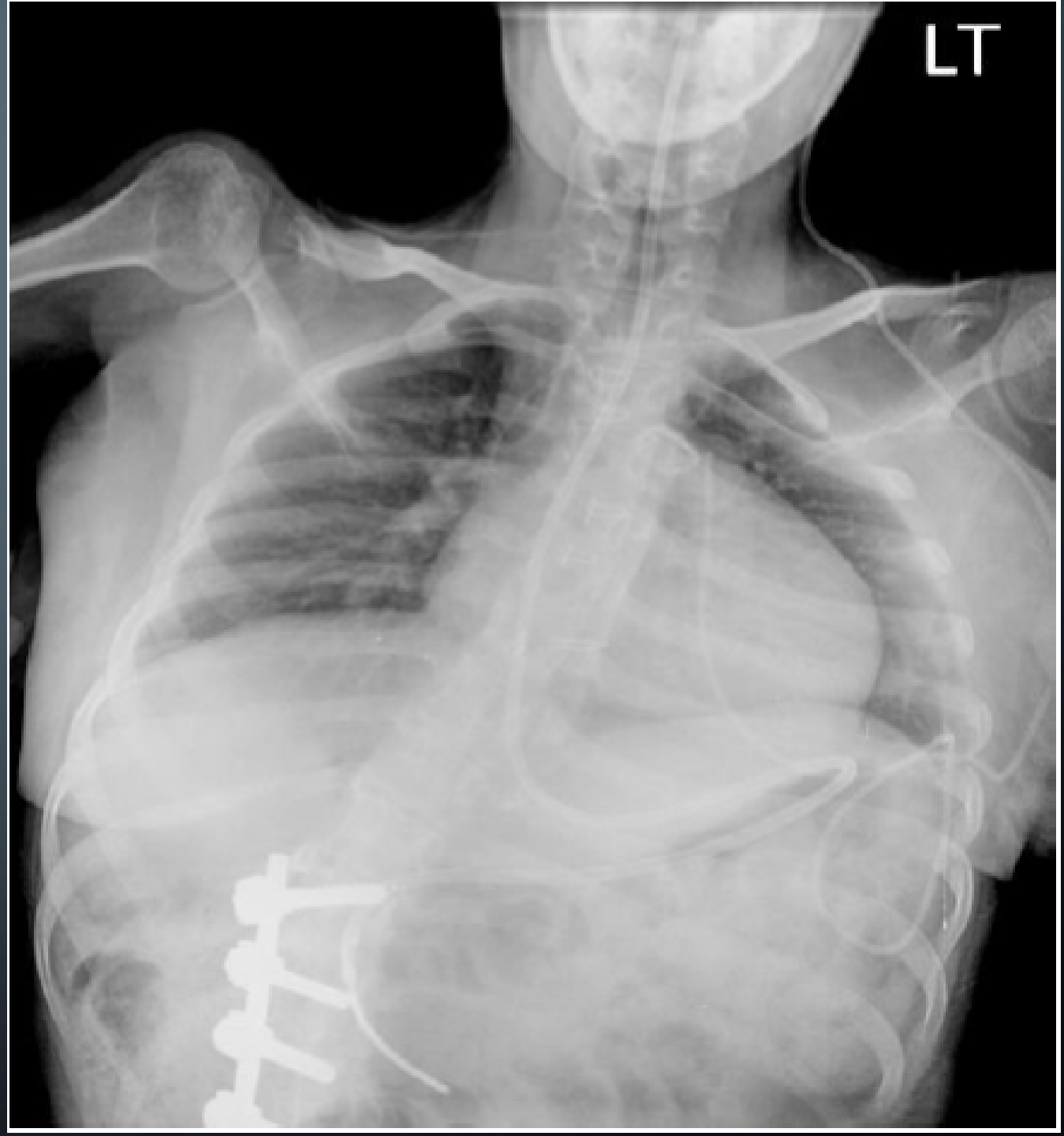
QUESTION: WHAT DO YOU SEE?



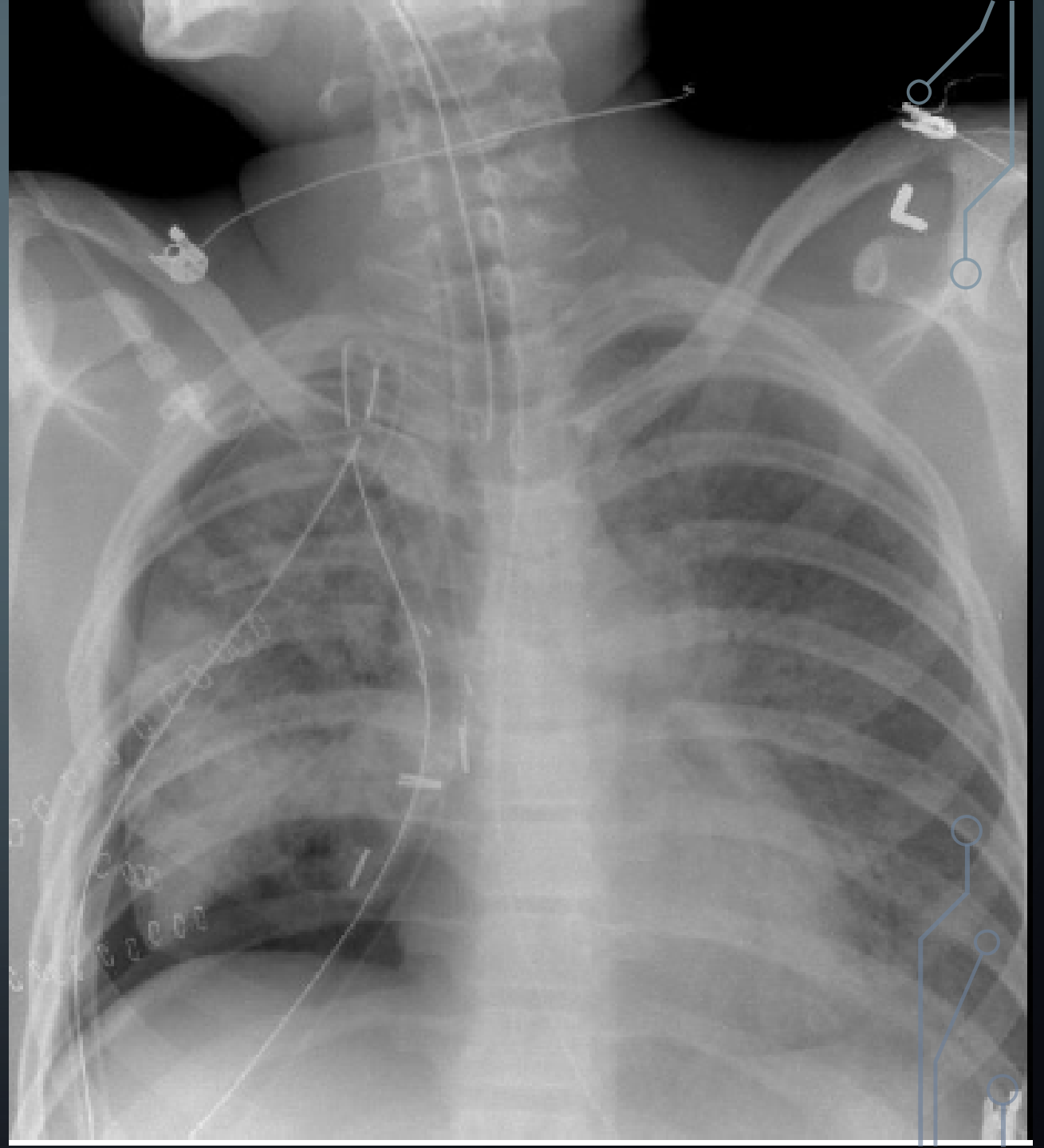
SOME LANDMARKS



WHAT DO YOU SEE?



WHAT DO YOU SEE?



REFERENCES

- Moher, T.G., Mak, D.C., Blumenthal, B., and Leventhal, L.M. Comparing the comprehensibility of textual and graphical programs: The case of Petri nets. In C.R. Cook, J.C. Scholtz, and J.C. Spohrer, Eds., *Empirical Studies of Programmers: Fifth Workshop*. Ablex, 1993, 137–161.
- Petre, M. Why looking isn't always seeing: readership skills and graphical programming. Communications of the ACM, 38(6), 1995
- Petre, M., and Green, T. R. G. Requirements of graphical notations for professional users: Electronics CAD systems as a case study. *Le Travail Humain 55,* 1 (1992), *47–*70.

THANK YOU