ICT Cubes Decode Webservice: From Slat Sequences to Cleartext

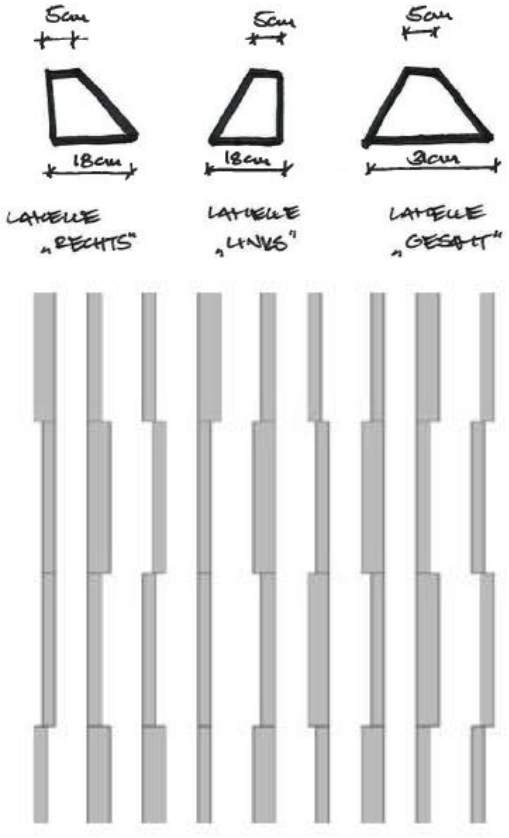
Georg Böcherer and Sebastian Baur
Institute for Communications Engineering, TU Munich
georg.boecherer@tum.de,baursebastian@mytum.de

AG Kommunikationstheorie
Institute for Theoretical Information Technology
RWTH Aachen University
June 21, 2013

RWTH Information and Communication Technology Cubes

Slats

Three types of slats: left, right, middle



Purpose:

- Make stacked floors appear as a cube
- Keep building from heating up in the sun

Side effect:

- Slats shadow the rooms

Design constraints

- Sequence of slats should look random
- Balance between cooling and shadowing:
  40% left, 40% right, 20% middle

Idea: Write a text to the facade

Challenge:

- Find a code that maps a text to a sequence of slats
- The generated slat sequence should respect the design constraints

Tool 1: Data compression

- Compressing (zip,rar,...) data generates a binary sequence that resembles the output of flipping a coin

Tool 2: Distribution matching

- Distribution matching maps the output of flipping a coin to a sequence that resembles the output of a desired source.[1]

[1] G. Böcherer, Capacity-achieving probabilistic shaping for noisy and noiseless channels, Ph.D. dissertation, RWTH Aachen University, 2012.

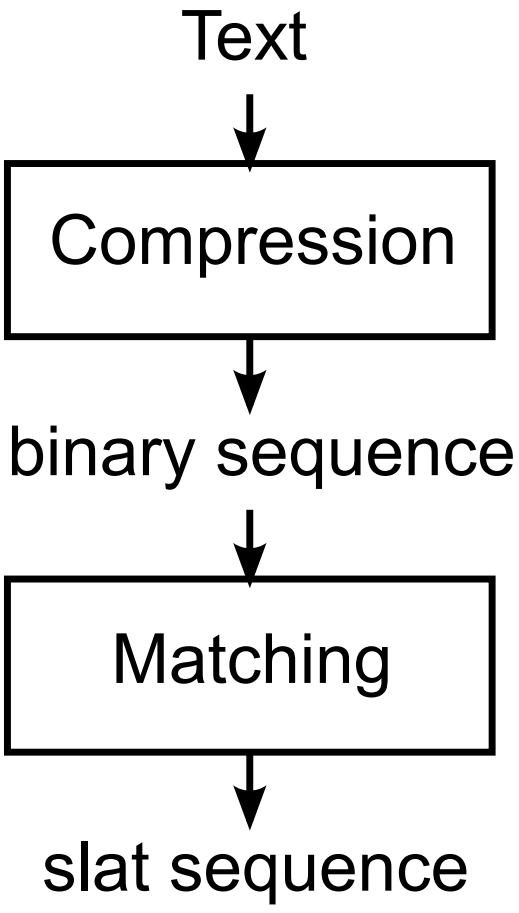
Tool 3: Law of large numbers

Think of a source that generates slats with the probabilities

$$\Pr(\text{left}) = \Pr(\text{right}) = 0.4, \qquad \Pr(\text{middle}) = 0.2$$

For a long sequence of slats generated by this source, the empirical distribution is close to 40% left, 40% right, 20% middle.

Approach



Text: Quotes from researchers

shannon the fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point **plato** let no one ignorant of mathematics enter here **schottky** space permits only an approximate statement of the theory **nyquist** when the output of an amplifier is connected to the input through a transducer the resulting combination may be either stable or unstable **hamming** it is better to do the right problem the wrong way than the wrong problem the right way **turing** machines take me by surprise with great frequency **fourier** the profound study of nature is the most fertile source of mathematical discoveries **wiener** there are no answers only cross references **gallager** good communication is central to a civilized society **knuth** beware of bugs in the above code i have only proved it correct not tried it **mackay** combine a simple pseudo random code with a message passing decoder **bell** you cannot force ideas successful ideas are the result of slow growth **kolmogorov** the human brain is incapable of creating anything which is really complex **gauss** when i have clarified and exhausted a subject then i turn away from it in order to go into darkness again **zuse** it is not true that virtually all news in a totalitarian state is false **marconi** every day sees humanity more victorious in the struggle with space and time **bernoulli** it would be better for the true physics if there were no mathema

Compression code[2]

_ : 000	a : 0111	b : 101110
c : 11110	d : 00110	e : 110
f : 10000	g : 010100	h : 11111
i : 0110	j : 00111000100	k : 00111001
l : 10110	m : 01011	n : 1001
o : 1010	p : 001111	q : 00111000101
r : 0010	s : 0100	t : 1110
u : 10001	v : 0011101	w : 101111
x : 001110000	y : 010101	z : 0011100011

[2] F. Altenbach, G. Böcherer, R. Mathar, Short Huffman codes producing 1s half of the time, presented at ICSPCS 2011.

Matching code[3]

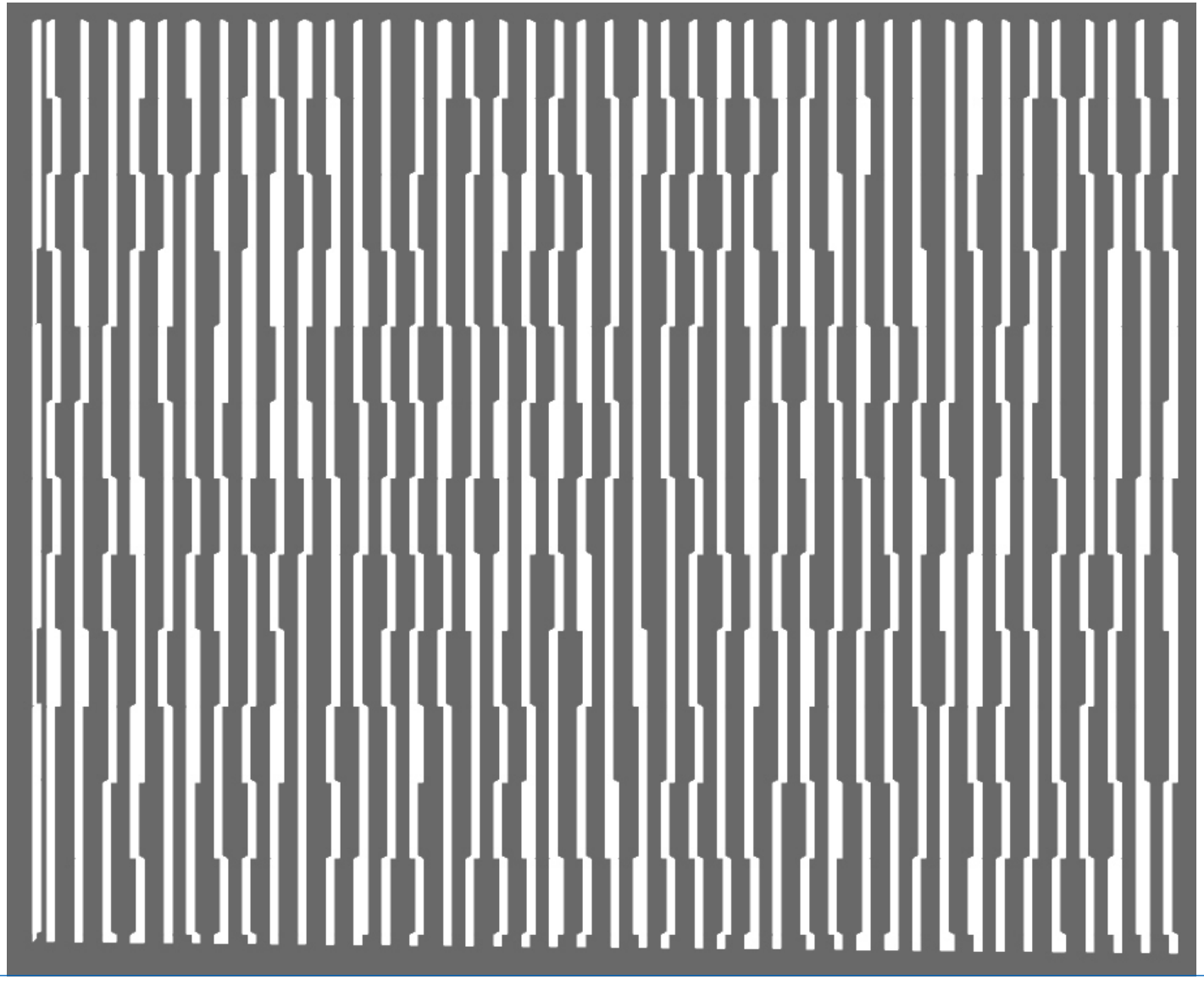
0010 : lll	1101 : llr	00000 : llm
1100 : lrl	1111 : lrr	00011 : lrm
00010 : lml	01101 : lmr	0000111 : lmm
1110 : rll	1001 : rlr	01100 : rlm
1000 : rrl	1011 : rrr	01111 : rrm
01110 : rml	01001 : rmr	000010 : rmm
01000 : mll	01011 : mlr	001101 : mlm
01010 : mrl	1010 : mrr	001100 : mrm
001111 : mml	001110 : mmr	0000110 : mmm

[3] G. Böcherer, F. Altenbach, M. Malsbender, R. Mathar, Writing on the facade of RWTH ICT Cubes: cost constrained geometric Huffman coding, presented at ISWCS 2011, best paper award.

Checking design constraints: Cooling and shadowing

	left	right	middle
Generated sequence	38.7%	40.9%	20.4%
Design constraint	40%	40%	20%

Checking design constraints: Appearance



Reading the Facade

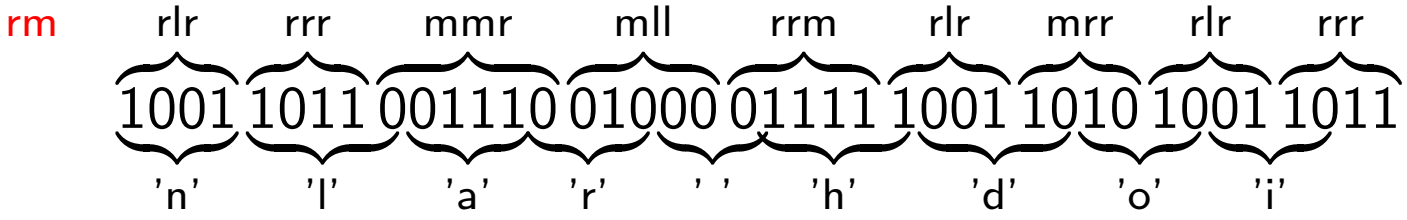
Two step approach:

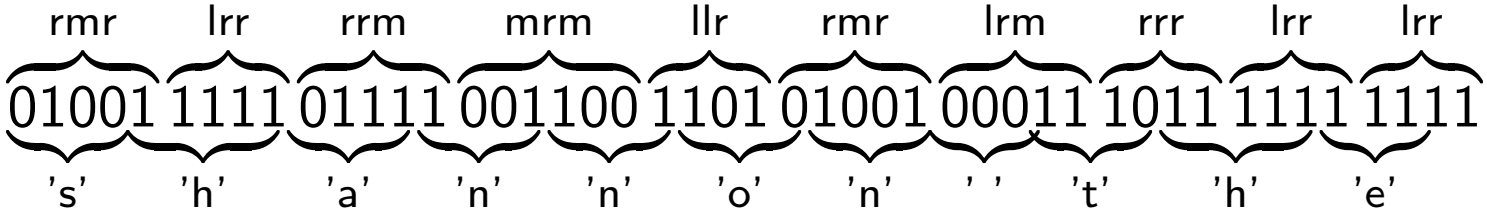
1. Read (a part of) the slat sequence
2. Decode the slat sequence

$\Rightarrow$ we start with **step 2.**[4]

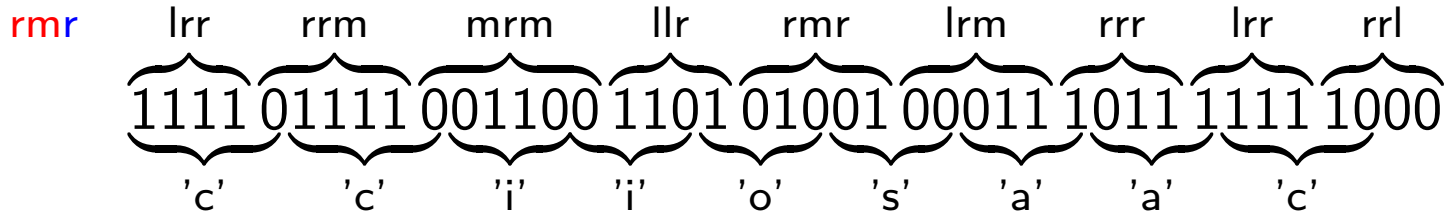
[4] Sebastian Baur: ICT Cubes Decode Webservice: Von Lamellensequenzen zu Klartext, B.Sc. thesis, TU Munich, 2013.

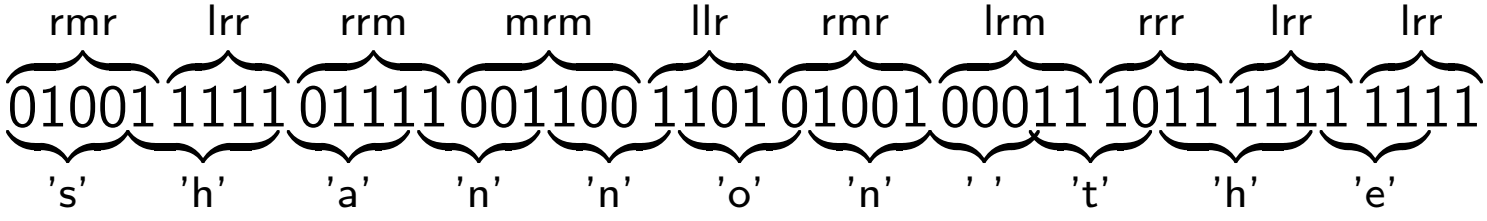
Synchronization Problem on Slat Level



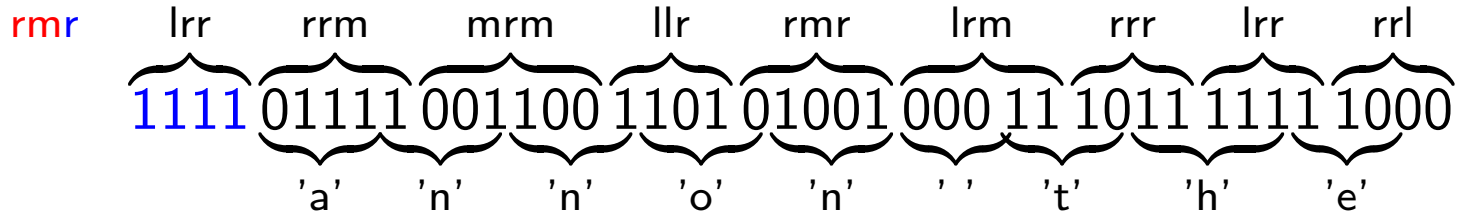


Synchronization Problem on Bit Level





Example for Correct Decoding



Observation

- Slat offset periodic $\Rightarrow$ no automatic synchronization
- Bit offset aperiodic $\Rightarrow$ automatic synchronization

Since we want to decode very short parts of the slat sequence, we do **not** rely on automatic synchronization.

Approach

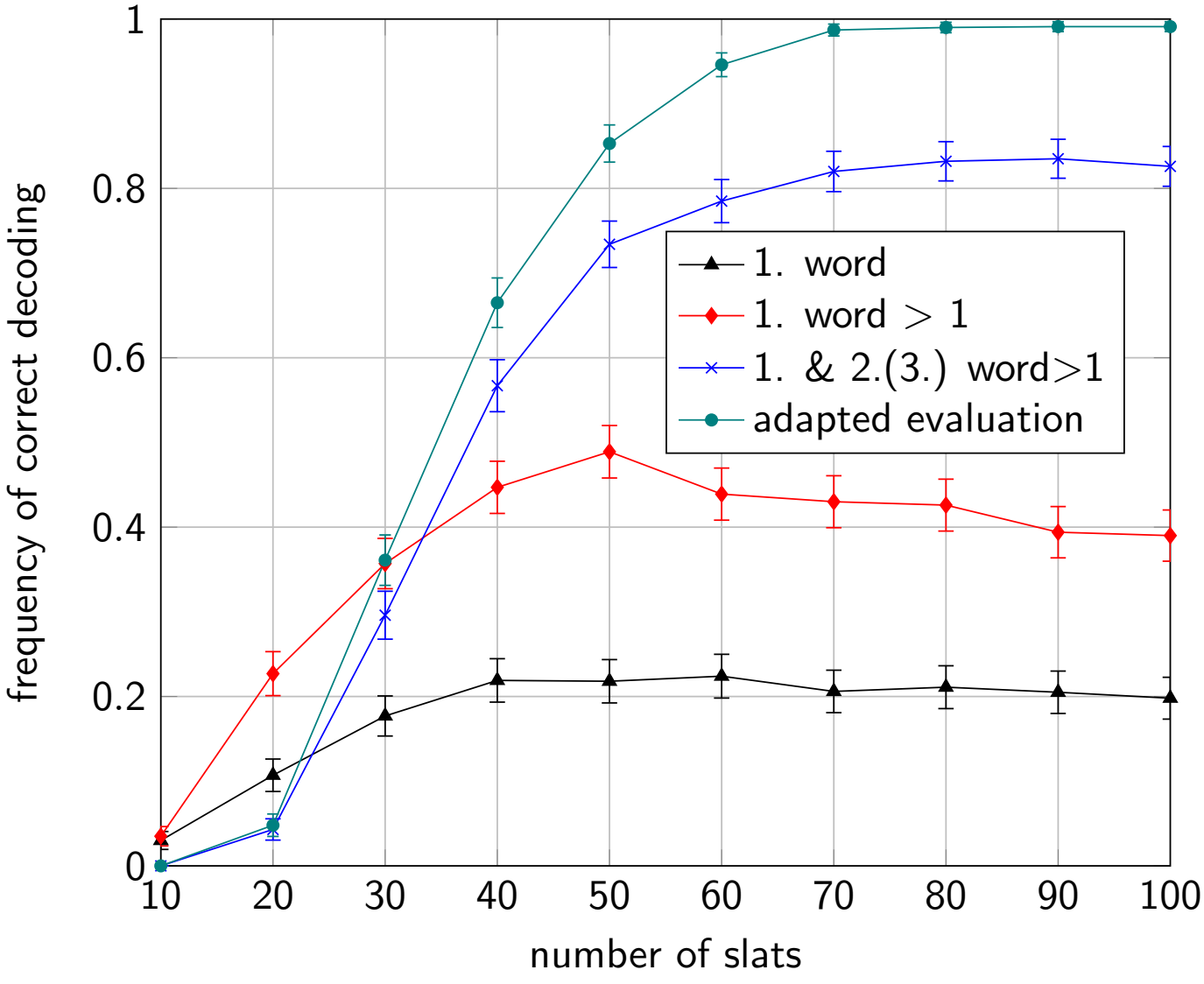
Let $\hat{t}$ be the decoded text and suppose we have the boolean function `isSynchronized`$(\hat{t})$.

for `slatOffset` $= 0 \ldots 2$
  for `bitOffset` $= 0 \ldots$ `maxLength`
    decode with `slatOffset` and `bitOffset`
    if[`isSynchronized`$(\hat{t})$]
      return $\hat{t}$
    end if
  end for
end for

Implementing `isSynchronized`

- Pilot symbols: less quotes!
- Comparison with cleartext: loss of flexibility!
- **Dictionary**

Testing Dictionary Based Synchronization



Implementation

- Implemented in google programming language **Go**
- Deployed in the cloud using google's appengine

Webservice Interface

- HTTP GET request
- Slat sequence as query:

  http://ict-cubes.appspot.com/?slats=rlrrrrmmrmllrrmrlrmrrrlrrrrl
- Call from MATLAB:

  resp=urlread("http://ict-cubes.appspot.com/?slats=rlrrrrmmrmllrrmrlrmrrrlrrrrl")
- Response as JSON string:

  {“ClearText“:“shannon the fundamental“,“Length“:24}

Outlook

From **photo** to **clear text**:

- Make plain view of the 8 facades available to the reader
- Match photo to plain view
- Pass slat sequence covered by photo to the webservice.