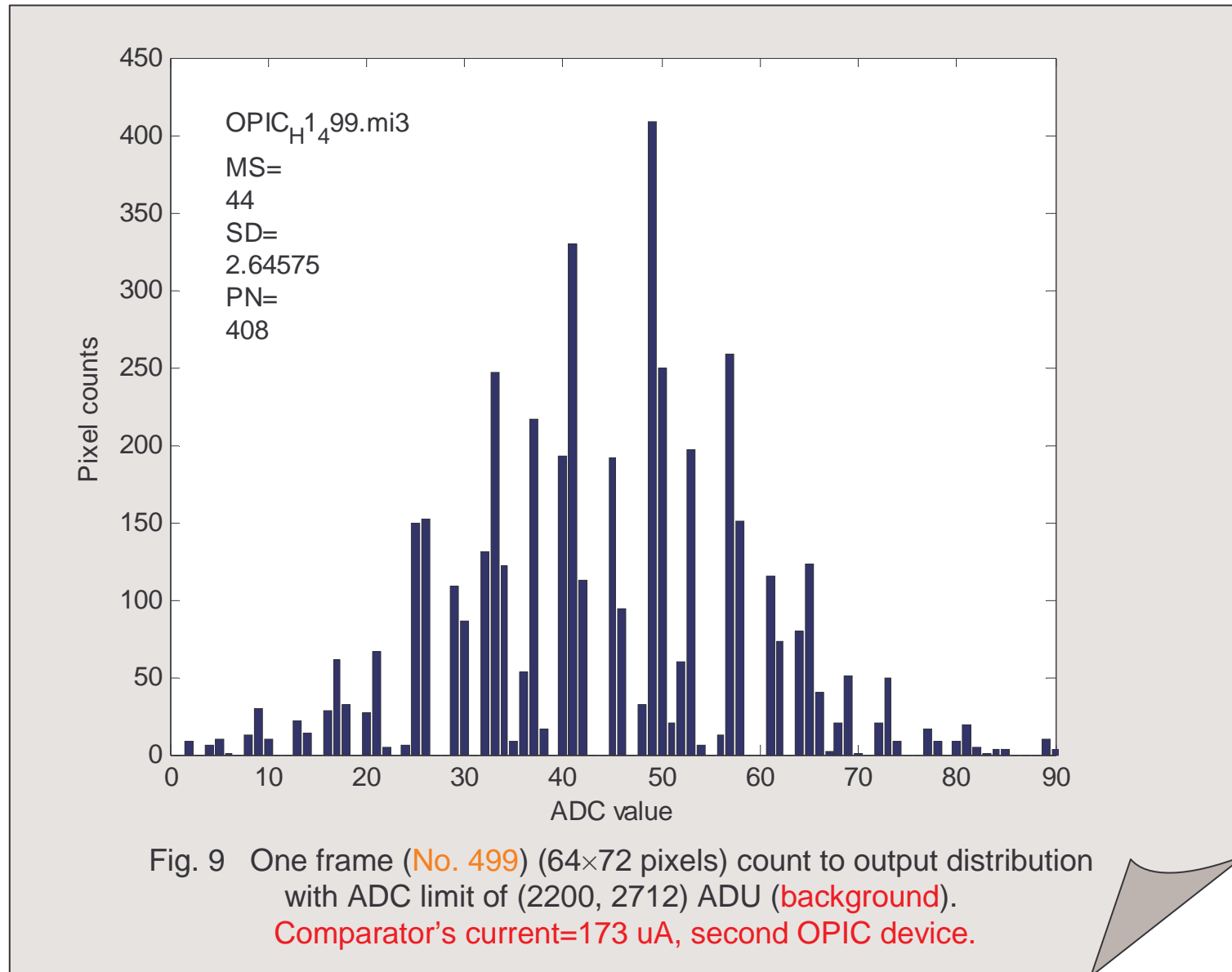


OPIC Investigations

ADC Linearity etc

Problems reported by Users



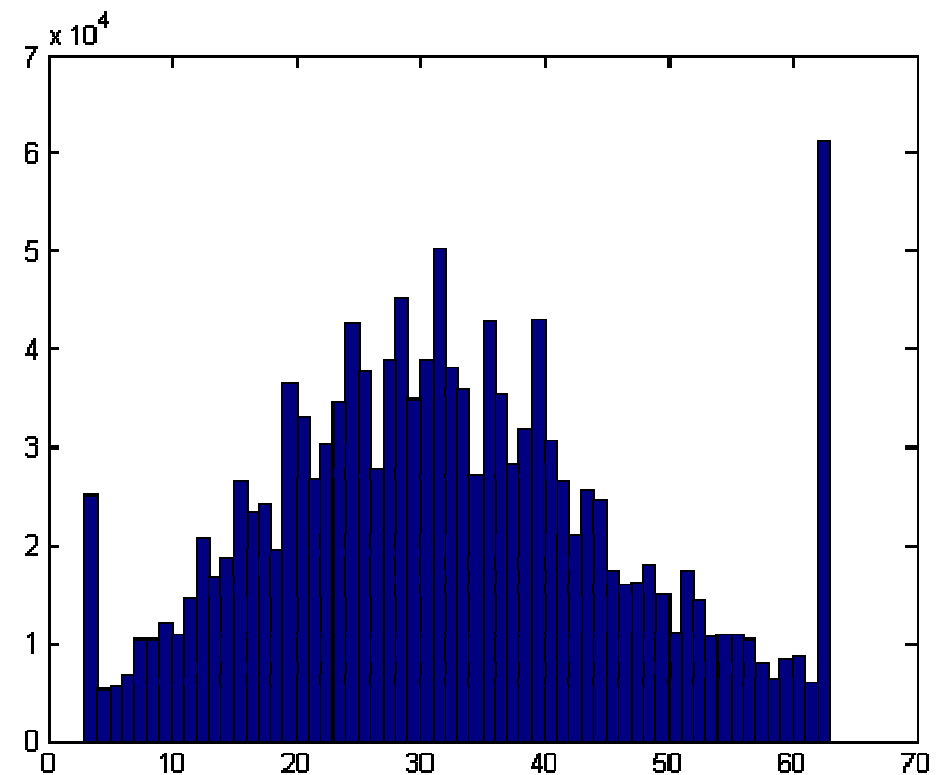
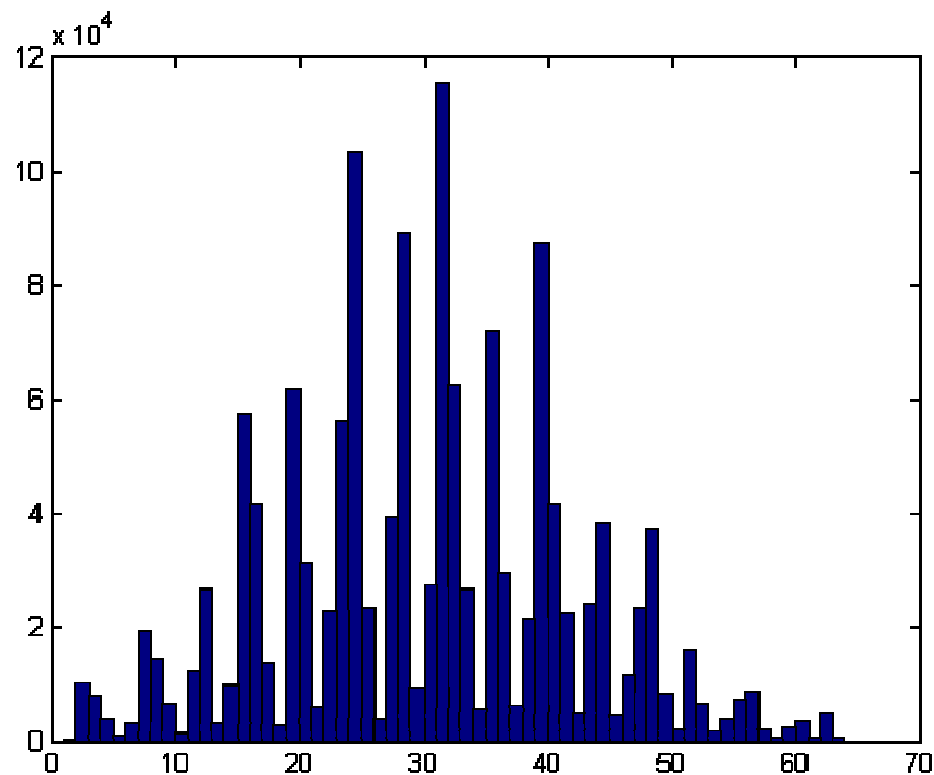
Problem solving

- 1) On-Chip gray-code signal is coupling to the sensitive charge storage/diode node in the pixel, primarily through feedback capacitance from the comparator when switching. This can be minimised by increasing the current in the source follower (better able to reject coupling onto its output node) and reducing the current in the comparator (slower switching reduces charge injection).
- 2) Each of the Gray-code bits seem to couple to the ADC result, not just a single dominant bit that could be isolated. A test code ('JCODE'), of 64 words was developed that always flipped two bits at the same time in opposite directions to minimise overall charge injection. Promising results! *(Every code word contains 4 ones and 4 zeros)*
- 3) But: when using the 'JCODE', the software reports a high proportion (~15%) of erroneous codes (codes containing 3 ones and 5 zeros, indicating the zero-to-one transition is not always stored in the time available). This suggests the on-chip buffering is inadequate to drive the full column fast enough.

Solution 1: New Bias Settings

| | Original | New |
|-----------|----------|-------|
| COMP_VREF | 100uA | 100uA |
| SF_BIAS | 50uA | 250uA |

Can also be reduced for improvements, 50uA min.



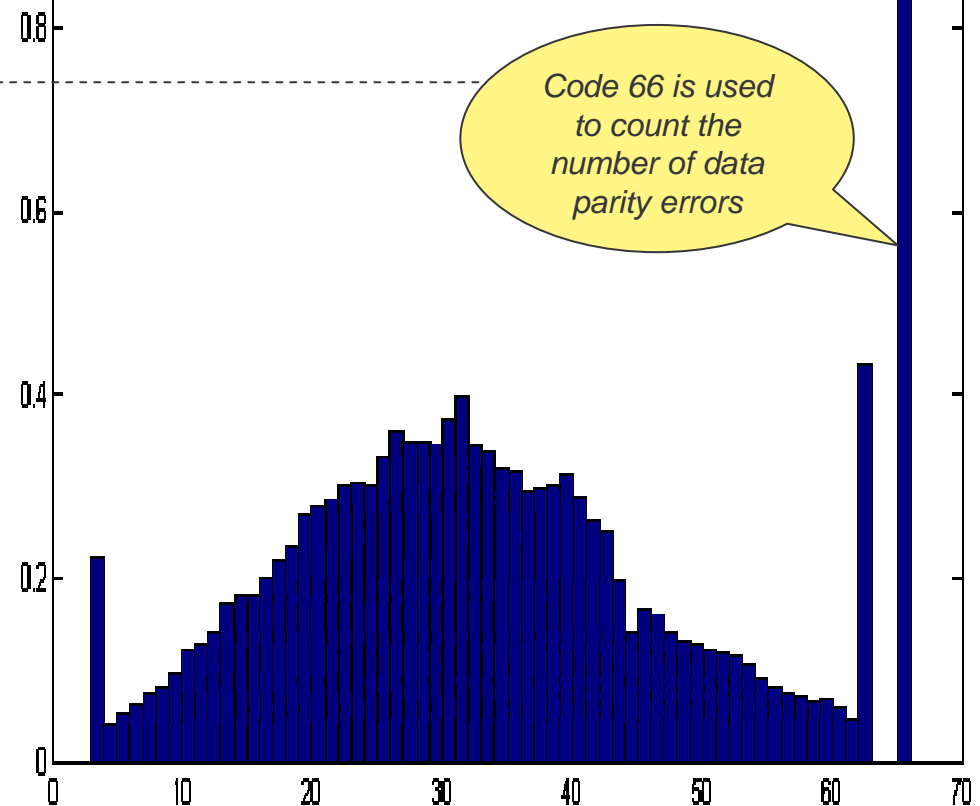
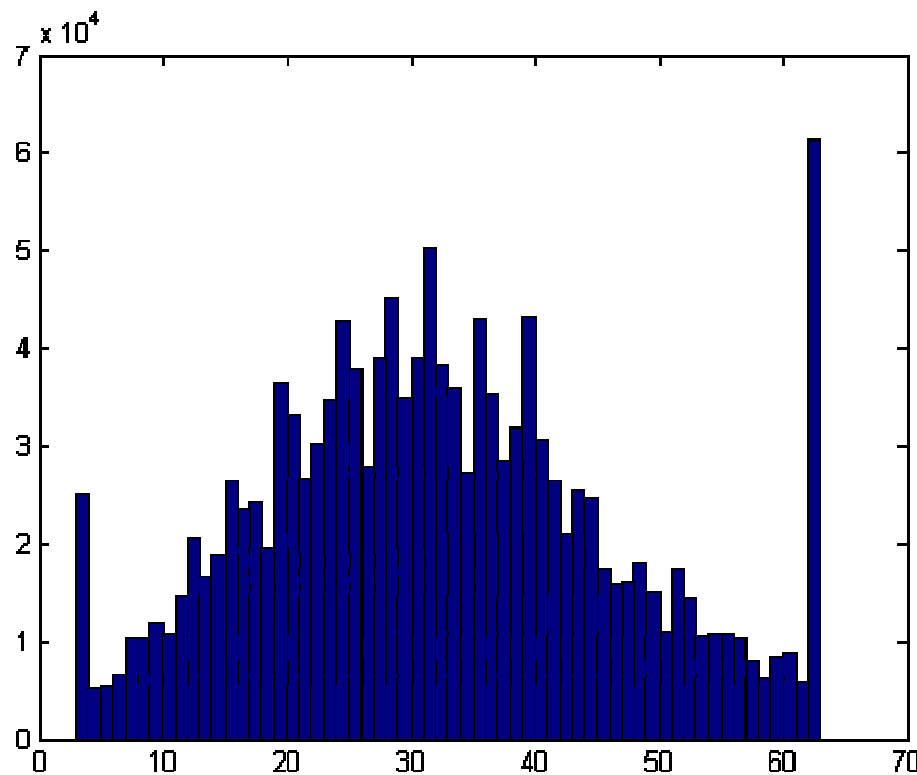
Solution 2: New coding scheme

DAC step size = 2: 64 codes (steps) DAC: 2100 à 2228
Histogrammed data relates to 500 frames, centre region: 52x52

Bias settings: COMP_VREF=100uA. SF_BIAS = 250uA

GRAYCODE

JCODE



Solution 3: Error detection

- a) Use new coding scheme, all codes have 4 ones and 4 zeros
- b) Use a 7-bit gray code with 1 parity bit

For full-scale ADC (DAC step size ~16)

- Charge coupling is minimal for large DAC steps, so standard gray code can be used. Existing data taken is likely to contain errors (maximum numerical error = one ADU)
- Could be improved by adding a parity bit to the conversion (reduces ADC to 128 codes, 7 bit) but full range could be achieved by re-writing the ADC routine to use both in-pixel registers, restoring 8 bit conversion.

For fine ADC (DAC step size ~2)

- Charge coupling is dominant at such a fine voltage sweep, so JCODE should be used to minimise charge injection
- Conversion is limited to 64 codes
- Invalid data can be discarded for device characterisation, noise measurements etc.

'JCODE' example: 4 bits

6 valid codes exist
(equal # 1s and 0s)

0011

0101

0110

1001

1010

1100

A valid sequence

