A study of BSD pixel timing

June 10, 2000

S. Teige

Department of Physics, Indiana University, Bloomington IN 47405

Abstract

An algorithm to idenTify pixels in the BSD is described.

1 Introduction

Observation of the online event display seemed to indicate there was a large fraction of events with more than one pixel in the BSD. Detailed analysis shows that, using a simple, ADC based pixel finding algorithm only about 50% of the events have a single pixel.

2 Old Algorithm

The ADC based algorithm required a TDC hit is a broad time window and a large signal in the corresponding ADC channel to define a hit in a BSD paddle. Figure 1 shows an ADC distribution from a typical BSD paddle. Demanding a large ADC value insures that a hit will be in a pixel but at the expense of pixel finding efficiency. Cutting at 105 ADC counts yields 53% single pixel events with 37& of all events having no pixels and an average pixel multiplicity of 0.8. Loosening the ADC cut to 45 counts yields 64% single pixel events but increases the average pixel multiplicity of 1.6, an indication that the fraction of fake pixels in increasing.

3 New Algorithm

Examination of TDC differencs for pixel components suggested a different pixel finding algorithm. Figure 2 shows the difference between the TDC values for two scintillator paddles that can be a part of a pixel. Clearly, there is a narrow (850 picoseconds) component of the distribution and this component is taken to exist because of in-time coincidences between these two paddles. A true pixel would have three such in time coincidences while a spurious pixel (caused by random, single hits in different paddles) would be out of time. The

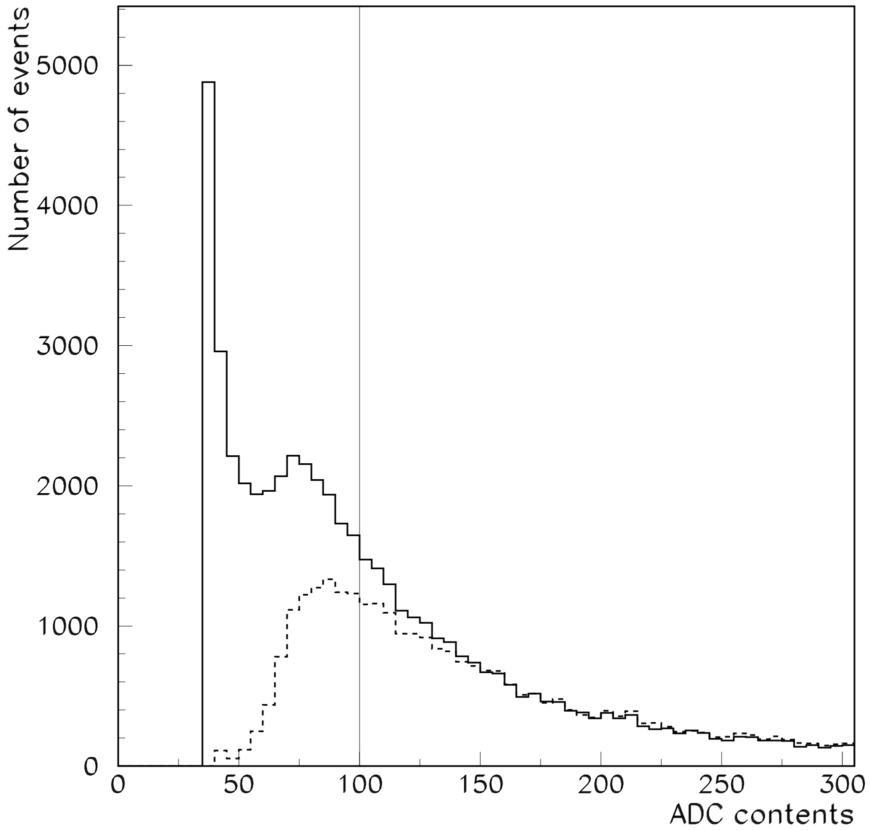

Figure 1: A typical BSD ADC distribution. The solid curve is the raw distribution. Seen is the tail of the pedestal and evidence for a minimum ionizing particle peak. The dashed curve is the distribution for events where the TDC based algorithm found a pixel containing this paddle. Above approximately 100 ADC counts (indicated by the vertical line) , the paddle is always in a pixel. The dashed curve shows that a large fraction of events are pixels but have and ADC value less than 100 counts.

mean value and the width of all possible BSD coincidences was determined by fits similar to that shown in figure 2.

To find a pixel, the loose overall timing cut was retained but the ADC cut was reduced to 40 counts, a value within the tail of the pedestal. A chi-squared was constructed using the measured value of the TDC differences between all three combinations of left, right and straight paddles in a pixel candidate and the means and widths determined above. Figure 3 shows the resulting pixel multiplicity distribution.

4 Discussion

A time based pixel finding algorithm has been described and tested. The yield of single pixel events is improved while the fraction of zero pixel and greater than one pixel events has been reduced. The narrow time windows used in the definition of a coincidence gives confidence that the pixels are true-threefold coincidences. The algorithm requires that the mean and width of many (about 600) time differences be determined, almost certainly as a function of run number. Determination of these quantities is straightforward and the map software to keep track of them already exists.

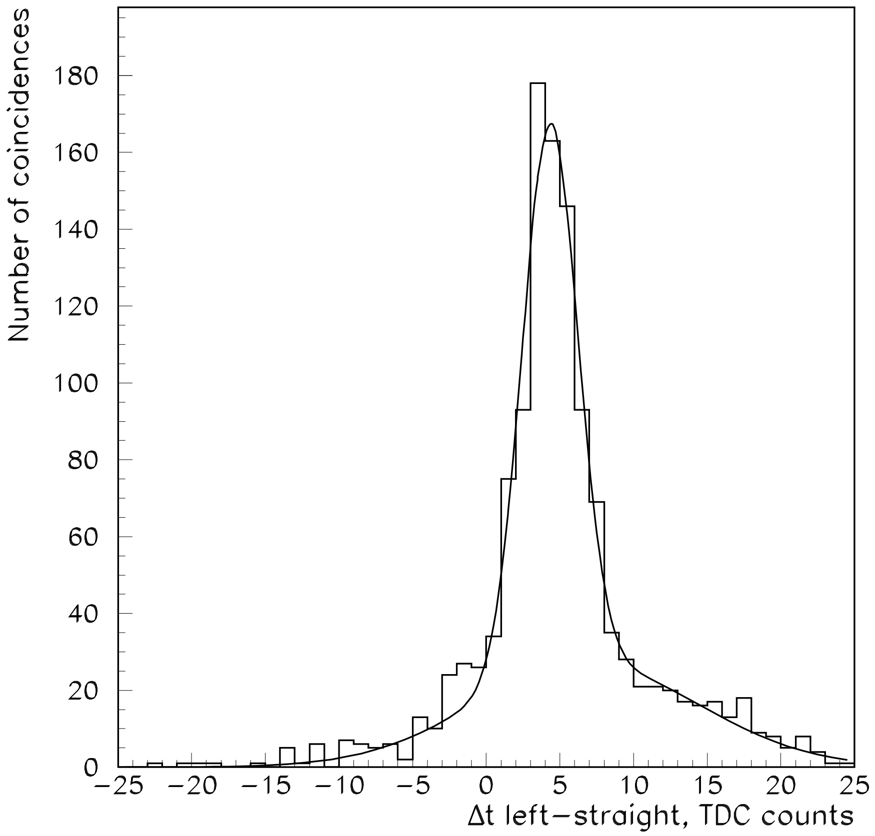

Figure 2: The difference between the TDC value for a "straight" paddle and the TDC value for a "left" paddle. The fit is to a sum of two gaussian distributions. The narrow gaussian has a width of 1.7 TDC counts (850 picoseconds) and corresponds to an in-time coincidence between these two paddles. Similar distributions are seen for left-right and right-straight coincidences.

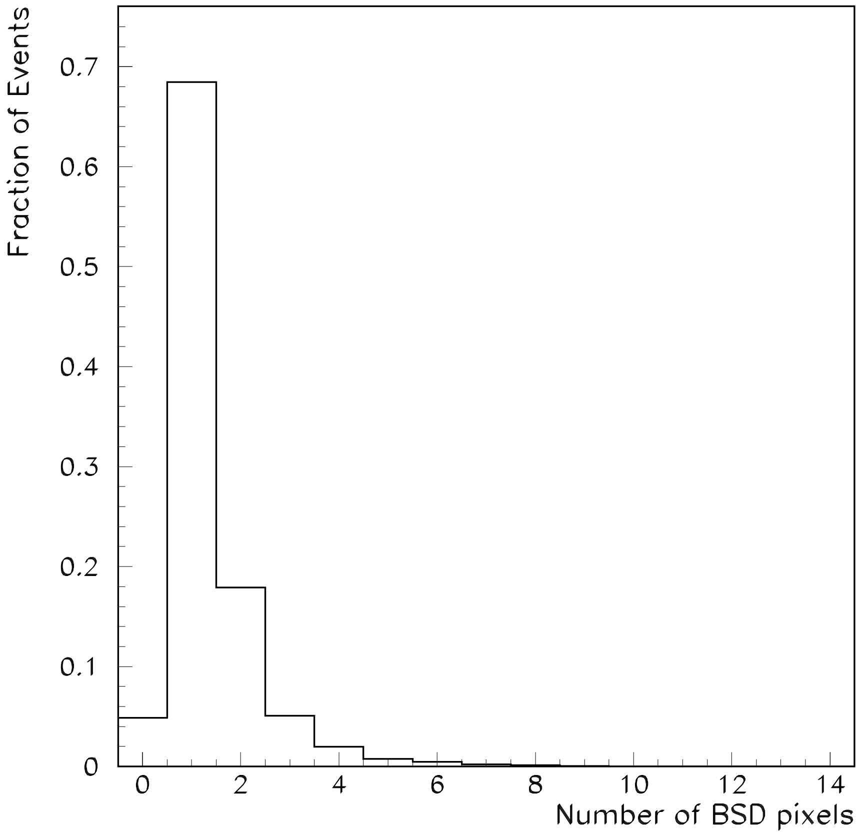

Figure 3: The pixel multiplicity distribution resluting from the new algorithm. 69% of all events have a single pixel and only 5.2% have none. The average multiplicity is 1.3.