FINAL REPORT

REPORTING PERIOD 01/01/2007-06/30/2009

DEVELOPMENT OF CRITERIA AND IDENTIFICATION OF PARTICLE CLUSTER SIZE BASED ON MEASUREMENTS OF VOID FRACTION IN GAS-SOLID SYSTEMS

Collaborators

David Roelant, Ph.D
Seckin Gokaltun, Ph.D

APPLIED RESEARCH CENTER
FLORIDA INTERNATIONAL UNIVERSITY

Prepared for

Dr. Ronald Breault
U.S. Department of Energy
National Energy Technology Laboratory
Under Grant Agreement: DE-FG26-07NT43062

August 19, 2009

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Abstract

A circulating fluidized bed (CFB) built at FIU was used to study particle motion in the riser in order to simulate flow regimes in a cold gasifier. High speed imaging was used in order to capture the dynamics of the particles flowing in the riser. The imaging method used here is called the shadow sizing technique which allowed the determination of particle areas and trajectories at various flow rates in the riser. The solid volume fraction and particle velocities calculated using the images acquired during the experiments can be related to granular temperature in order to detect formations of clusters in the riser section of the CFB. The shadow sizing technique was observed to be an effective method in detecting dynamics of particles in motion and formation of clusters when supported with high-speed imaging.

Table of Contents

Abstract.....	2
Table of Contents	3
Executive Summary	4
Literature Review	5
Experimental Set-up	6
Data acquisition	10
Image Processing	13
Removing dead pixels and spots from the images.....	14
Finding the centroid and total area of each particle.....	17
Calculating the instantaneous solid volume fraction	19
Tracing the trajectory of each particle	19
Experimental Results.....	21
Conclusions.....	26
References.....	26
Milestone Status	29
Project Budget Status	29
Appendix.....	32
Matlab code for image processing	32

Executive Summary

The United States of America is the world's largest energy producer, consumer, and net importer. Increased demand on imported petroleum, the ongoing deregulation of the energy industry, and environmental concerns associated with the use of fossil fuel for production of electricity and transportation fuels are all contributing to an increasing interest in better utilization not only of fossil fuels available in abundance such as coal, but also in unconventional fossil fuels such as oil shale, and tar sands.

A shortage of fossil fuels for production of gasoline by new technologies and electricity demand better utilization of these fossil fuels. Attempts to use bubbling fluidized beds with internal heat exchange tubes were not successful due to severe tube erosion problems, which were unknown before the construction of the large pilot-scale and demonstration plants. Such failures and future energy goals concerning the environment necessitate an understanding of gasification, catalytic cracking, and combustion in risers and reactors. Developers of gasifiers, combustors, chemical reactors, and owners of energy power plants are looking for more timely and cost-effective methods to predict the performance of their power generation components. Therefore, they have been incorporating simulation in their design and evaluation processes. Several computational fluid dynamics (CFD) codes have been developed to simulate the hydrodynamics, heat transfer, and chemical reactions in fluidized bed gasifiers and other power generation equipments where gas-solid flow is dominant. These computer codes are based on accepted equations governing multiphase flow. However, detailed information on gas-solids flow structure, especially identification of particle cluster size and solid concentration, is needed for validation of such CFD codes.

The validation process requires comparison with sufficient experimental data obtained through advanced diagnostics in order to identify basic criteria for gas-solid systems such as particle cluster size, concentration, and granular temperature. The research includes utilization of advanced experimental approach coupled with new mathematical and statistical analysis methods to identify particle clusters based on measurements void fraction.

The experimental work presented here involved imaging of solids concentrations using shadow particle sizing to obtain measurements of particle number density and volume fraction. The shadow sizing technique was used to simultaneously measure particle velocities and void fraction.

New mathematical analysis methods have been developed to identify criteria for particle cluster size and determine the inverse solid concentration, at which granular temperature (turbulent kinetic energy of the particles) reaches its maximum and inverses its behavior. This can be accomplished by indirectly evaluating changes in granular temperature distribution, for different particle groupings, using accepted quantitative proportionality between void fraction and granular temperature.

Literature Review

The occurrence of particle clusters in fluidized beds has been observed since Yerushalmi *et al.* (1975) presented his hypothesis. Horio *et al.* (1988) and Hartge *et al.* (1988) detected clusters in fluidized beds. Sinclair and Jackson (1989) applied the granular flow model to a fully developed gas-solid flow. Ding and Gidaspow (1990) derived an expression for solid viscosity and pressure of a dense gas-solid flow by using the Boltzmann integral-differential equation and assuming a Maxwellian frequency distribution for the particle velocity. Gidaspow (1994) extended the formulation to both dilute and dense cases by considering a non-Maxwellian velocity distribution. Gidaspow, and Huilin (1998) showed that a relation exists between pressure, temperature, and solid fraction analogous to the ideal gas law. Noymer and Glicksman (1998) addressed the effect of clusters on wall heat transfer coefficients.

Lun *et al.*, (1984) and Lun and Savage (1987) presented a relationship between granular temperature and void fraction. A systematic criterion for identification of particle clusters was proposed by Soong *et al.* (1993). Brereton and Grace (1993) and Dejin *et al.* (1996) proposed intermittency and heterogeneity indexes to characterize the degree of cluster formation. Spahn *et al.*, (1997) provided evidence in support of the fact that granular temperature may provide the information to establish quantitative cluster criteria. Sharma *et al.* (2000) presented the parametric effects of particle size and superficial gas velocity on the cluster duration time, occurrence frequency, and solid concentration.

The existence of solid clusters characterizes mesoscopic behavior. The cluster concept evolved as a result of the recognition of a large slip velocity between the gas and solid particles in the Circulating Fluidized Bed (CFB). Particle clusters are an indication of the heterogeneity in the mesoscale. A complete characterization of the hydrodynamics of a CFB requires the determination of the voidage and velocity profiles.

To handle complicated phenomena in the vertical pneumatic transport of solids, two theoretical approaches are proposed, namely the Eulerian and the discrete particle approaches. The Eulerian model considers the particulate phase as a continuous fluid interpenetrating and interacting with the fluid phase (Gidaspow *et al.*, 1989). The kinetic theory of granular flow is used in the Eulerian model to offer a theoretical framework for simulating gas-solid flow with particles of different size and/or different density (Van Wachem *et al.*, 2001a, b). However, severe difficulties are encountered: first many closure laws related to the mutual interaction of particles belonging to different classes have to be formulated; moreover, the universality of the constants used is questionable (Cao and Ahmadi, 1995).

Discrete particle models have become very useful and versatile tools in studying the dynamics of gas/particle flows. In this approach, each particle is treated by solving Lagrangian equations of motion for all the particles of the system, with a prescribed set of initial conditions. Once the flow and particle properties are known, the interface quantities between both phases can be calculated. It offers a more natural way to overcome the aforementioned problems, since each individual particle is tracked in the simulation. Moreover, it provides a powerful tool to investigate the detailed phenomena at the individual particle scale and to examine local phenomena such as particle to bed (or bed to particle) heat and mass transfers.

This approach was used to simulate gas–solid fluidization in the last decade. Phenomena such as bubbling, slugging, and solid transport in a circulating fluidized bed (CFB) can be simulated (Tsuji *et al.*, 1993; Hoomans *et al.*, 1996, 2001; Helland *et al.*, 2000; Van Wachem *et al.*, 2001a, b). Some researchers simulated clusters in CFB. Tanaka and Tsuji (1991) investigated the cluster formation in a vertical riser and the particle-induced instability in gas–solid flows. They showed that particle–particle interactions play an important role in cluster formation and cause flow instabilities even when the mean concentration is about 0.5%. Tanaka and Tsuji (1991) observed that gas velocity decrease and particle loading increase result in instability and inhomogeneity. The Direct Monte Carlo model was used by Ito *et al.* (1998) to simulate the dynamics of clusters. The individual particle behavior can only be obtained statistically because particle collisions are described from statistics. Ouyang and Li (1999) developed a particle-motion-resolved discrete model to simulate heterogeneous structure in gas–solid fluidization. Helland *et al.* (2000) studied the cluster formation in gas-particle CFB. They studied the influence of porosity and observed a large difference in the local flow as a function of porosity.

To simulate the cluster formation, several researchers (Ito *et al.*, 1998; Helland *et al.*, 2000) considered that the particles were distributed uniformly in the riser as an initial condition. This assumption is not realistic for predicting the cluster formation in CFB.

In the particle-motion-resolved discrete model (Ouyang and Li, 1999), the interactions forces between particle and fluid, and vice versa, were considered separately, which does not obey Newton’s third law. Zhou *et al.* (2002) emphasizes the influence of particle properties, porosity function, and velocity on the global particle flow structure and the formation and development of local regions of higher particle concentration, i.e., particle clusters.

Experimental Set-up

Initially two conceptual designs for the experimental setup were considered in order to control and measure the flow rate of bulk solids into the riser. The first design incorporated a screw feeder in the circulating fluidized bed setup. The screw feeder was driven by a variable speed motor and discharged a known volume of bulk solids per revolution. The other design considered regulating the flow of bulk solids into the riser through a rotary air-lock valve. Similarly, the rotary air-lock valve was driven by a variable speed motor and discharges a known volume of bulk solids per revolution. In order to determine the necessary operating discharge rates for both the screw feeder and the rotary air-lock valve, bulk discharge rates were calculated at several superficial gas velocities to maintain certain solid concentrations inside the riser. The rotary air-lock valve technique has been selected to better suit the needs.

After completing the design reviews for various types of Circulating Fluidized Beds (CFB) in the literature, the final design that was built at FIU included a 6-inch diameter, 10-ft long clear acrylic riser, gas-solid separator (cyclone), standpipe, air-lock valve, and air conveyor. Bulk solids flowed from the stand pipe to the air-lock valve, through which predetermined amounts of solids discharged into the air conveyor and then to the riser. The air-solid mixture fluidized in the riser before it enters the cyclone. The stand pipe was 10-inch in diameter and 3-feet height. The airlock valve had a 10-inch diameter inlet and outlet, a cast iron body, and a mild steel rotor. It was equipped with a 1 HP variable speed motor with 100% rotor capacity of 865 ft³/hr at 31 rpm. The initial CFB assembled at FIU is shown in Figure 1.

(a) Schematic Model

(b) Experiment loop

Figure 1. Initial circulating fluidized bed

Figure 2. Schematic of the Experimental Set-up

The schematic for the experimental CFB setup is shown in Figure 2. Compressed air was provided from the compressor to the system. The flow rate was adjusted with the valve at the entrance before the flow meter. The rotary airlock supplied the solid particles at a desired rate. The air-particle mixture entered the acrylic riser from the bottom, and the fluidized particles were imaged at the riser. The mixture then flowed through the separator where the solid particles were separated and air passed out to the atmosphere. Any remaining solid particles in the air were then captured in the filter bubbler. The solid particles were collected in to the stand pipe.

There have been several configuration changes to the FIU CFB design. The rotary valve was removed since it was getting clogged with polystyrene particles resulting in a discontinuous mass flow in the riser section. The rotary valve was replaced with a PVC pipe that would feed the particles flowing up in the riser back into the horizontal pipe at the bottom of the CFB setup (Figure 4 and Figure 5). With the current configuration there is no mechanism to control the solid concentration rate in the riser.

In order to remove the electrostatic buildup, an aluminum tape was placed inside the riser, which was grounded. The filter that was covering the outlet of the cyclonic separator was torn as a result of running the compressor over 200 cfm. In order to collect the particles that escaped from the top of the cyclonic separator, a metal bin was used by attaching three radial-flow filter cartridges under the lid of the bin (Figure 6 and Figure 7). These adjustments were brought into the CFB design in order to temporarily resolve issues quickly and to continue image acquisition before the project ended. Better permanent solutions will be developed in case more funding for the project is received.

Figure 3. Latest CFB configuration at FIU

Figure 4. 10" PVC pipe used instead of air-lock rotary valve

Figure 5. Standpipe with the rotary valve removed

Figure 6. Torn filter

Figure 7. Particle collector bin

Data acquisition

Initially the Particle Image Velocimetry (PIV) system was tested and configured for data acquisition purposes. The laser source was tested and aligned successfully. Two high speed digital cameras have been tested and configured with the PIV system. Preliminary testing of the PIV system was completed utilizing a bench scale setup. The setup consisted of small beaker containing water and Polyamid seeding particles, which have a mean particle diameter of 20 μm . The water and seeding particles were rotated by a magnetic stirrer. The images obtained by the camera were not clear despite the fact that a filter was used to reduce the optical noise. The PIV system included Flow Manager™ software that was used to analyze the images obtained by the camera and develop a velocity vector map for the seeding particle. The diameter of particles could not be calculated.

The PIV system had the capability of measuring particle diameters utilizing the built-in Interferometric Particle Imaging (IPI) technique. IPI was based on the interference of the reflection and refraction glare points. Any phenomenon that might affect one of the glare points made the size measurements impossible. Thus, the particles had to be transparent, homogeneous, and spherical. Liquid droplets, air bubbles, and glass balls were good candidates for the IPI technique. Good image quality (contrast) would be obtained for a relative refractive index, defined as the particle refractive index divided by the medium refractive indices steered clear of unity. To ensure sufficient contrast of the interference pattern, the intensity of the two glare points had to be comparable. This condition was used to determine the position of the camera and depends on the relative refractive index, polarization, and wavelength of the laser sheet.

The PIV and IPI techniques could be combined with the help of a camera mount in order to simultaneously get the velocity and size of the particles from shots obtained by two cameras. Unfortunately, the measurements were taken in a planar coordinate and did not provide information regarding the particles' position in the plane normal to the laser sheet. Such information was very important in determining the proximity of particles with respect to each other in the space, so that accurate estimates of the void fraction could be obtained. Therefore, the IPI component of the PIV system could not be used to obtain measurements of particle size.

Instead, we reviewed closely the capabilities of other techniques capable of identifying the particles coordinates in the space. In shadow sizing, the object is backlit with a light source and a camera acquires the shadow image of the object. The shadow of the particles gives the shape and position in the image plane. Image processing software with particle-tracking velocimetry (PTV) capabilities would determine the size and the location of the center of each particle as well as its velocity.

This technique was thought to help in determining not only particles' sizes, but also their coordinates in the space, thus giving a more accurate void fraction. Particles' velocities can also be measured by cross correlating two successive shadow images obtained from this technique. Typical shadow sizing setup is shown in Figure 8. Based on this review, all necessary measurements can be obtained by utilizing shadow sizing technique alone, and there will be no need to carry out PIV measurements as planned earlier.

Figure 8. Typical setup for a shadow sizing experimental setup

The diameters of the particles from the images are obtained from the shadow of the particles. As long as the particle stays in the depth of field, the particle shape is distinct. As the distance from the subject plane of the particle increases, there is a gradual reduction in intensity at the edges of the particle captured by the camera. Particle diameters will be calculated by measuring the pixel area of the shadow of the particles. If the particle is located at some distance from the focal plane of the objective lens, the resulting shadow becomes defocused. The defocused images no longer have the sharp periphery at the edges, and there is a continuous graduation of intensity between the particle center and the background. This continuous graduation of the intensity profile can be used to determine the location of the particle relative to the plane of best focus, also giving the displacement in the third direction.

A CCD camera (C8484-05CP) was used initially, which had a resolution of 1344 (H) x 1024 (V) at full frame mode and a 12.2 fps frame rate. The effective area was 8.67 mm (H) x 6.60 mm (V), which was less than the size of the horizontal field of view (8.8 mm) of the telecentric lens (Edmund optics 55-350) that was mounted on the CCD camera. A LED light source was placed behind the particles to provide an even illumination of the flow field. A matching filter was mounted on the lens corresponding to the LED light source.

Figure 9. CCD camera and lens shown during acquisition

Figure 10. Properties of a telecentric lens

In order to increase the frame rate of the camera, the binning mode of the camera was used during acquisitions. The camera had three different modes for binning: 2x2, 4x4, and 8x8 with frame rates 22.3, 40.9 and 68 fps, respectively. Binning is the combination of two or more CCD image sensor pixels to form a new “super-pixel” prior to readout and digitizing (Figure 11).

Figure 11. 2x2 binning mode uses 16 pixels in order to create 4 new pixels

The advantages of binning were that it reduced the noise to signal ratio, increased the brightness, and increased the frame rate. On the other hand, it reduced the resolution as more pixels were binned together.

Even at the highest binning mode, the frame rate of the camera was not able to capture sharp images of particles in motion. To further increase the frame rate of the images, the CCD camera was replaced with a high-speed camera (Vision Research v5.0) that had 3800 pictures per second (pps) shooting capability at a resolution of 512 by 512 pixels. The maximum frame rate was 60000 pps at 256 pixels in horizontal resolution and 32 pixels in vertical resolution (Table 1).

Table 1. Vision Research high speed camera V5.0 maximum recording speed vs. image size

	Horizontal Resolution (pictures per second)		
	1024	512	256
Vertical Resolution (pictures per second)	1024	1000	1900
	512	2100	3800
	256	4200	7400
	128	8200	14000
	64	15000	25000
	32	27000	43000

An adaptor was purchased from Vision Research in order to connect the telecentric lens with the high speed camera. Images were obtained at 150 cfm and 180 cfm by varying the frame rate to obtain the best recording. The camera records the frames into its internal memory of 1024 MB, which can store 1024 images for 1 second of continuous recording at 1000 pps.

Image Processing

The CCD camera that was used initially outputted an AVI video file made up of frames acquired during the acquisition. Several frames were selected from the video file and saved as separate images using Quicktime software. To analyze the images, the ImageJ open-source software developed by the National Institute of Health was chosen to be used.

The procedure to calculate the solid volume fraction on a sample image was tested for still image shots of polystyrene particles. The polystyrene particles were placed on a transparent surface and the full frame mode of the CCD camera (Figure 12) was used. A region of interest (1.19x0.92 mm) was selected from the original image and saved as a separate image file.

Figure 12. Polystyrene particles placed on a transparent surface

The ImageJ software was used to read this image file and remove the background from the foreground. The image was then converted into binary format, where the foreground objects were represented in black and background was represented in white. The software was used to automatically fill the white spots inside the particles. The watershed tool was used to separate the particles that were touching each other. Finally the particle analyzer tool was used to count the particles and calculate the area of each particle (Figure 13).

Figure 13. Identification of particles in a region of interest

The area of each particle was used to calculate the corresponding volume. This was used to

calculate the solid volume fraction as
$$\epsilon_s = \frac{\sum_i^n V_i}{Ah}$$
 where for the given case above, $n=21$, $A=0.158 \text{ mm}^2$ and $h=1 \text{ mm}$ gives $\epsilon_s = 0.99\%$.

Later when the CCD camera was replaced with a high speed camera, the image analysis procedure had to be updated as well. The high-speed camera outputted in Quicktime movie format, which was made up of the frames acquired during the acquisition. A Matlab script was written in order to analyze the video stream and perform the following operations:

- Read the video file,
- Measure the background grayness level,
- Remove the dead pixels from the images,
- Identify the particles and trace their boundaries for each frame,
- Calculate the areas of particles and find the instantaneous solid volume fraction,
- Calculate the displacement of each particle vertically and horizontally,
- Calculate the axial and horizontal speed of each particle, and
- Trace the trajectory of each particle by storing the cell centers in time.

The functions listed above were achieved using the Image Processing Toolbox in Matlab described below.

Removing dead pixels and spots from the images

Matlab v7.8 does not allow reading video files with the *.MOV extensions in MS Windows operating systems. In order to read the video files Quicktime Pro software was used to convert the *.MOV files into *.AVI video format. The resulting video file was read into Matlab using the “mmreader” command as in:

```
>>obj = mmreader('H:\NETL\...\050809_T10_2000fps_150_512512.avi');
```

The mmreader outputs the number of frames into “numFrames” variable as in:

```
>>numFrames = obj.NumberOfFrames;
```

And any frame can be selected by using the “read” function. The 1600th frame was selected for the video at 150 cfm in order to measure the grayness level, since there were no particles in this frame.

```
>>image1 = read(obj, 1600);
```

Figure 14. 1600th frame is used at 150 cfm to capture the background grayness and identify the dead pixels and spots on the lens

The background image was converted into grayscale by,

```
>>i2=rgb2gray(image1);
```

And the global grayness threshold was calculated using the “graythresh” command:

```
>>leveld = graythresh(i2);
```

The locations of the dead pixels were stored into the “deadpx” matrix by first converting the image into binary image format using the grayness threshold level “leveld” and the command “im2bw” and then subtracting the binary image from 1 as in:

```
>>deadpx = 1-im2bw(i2,1.1*leveld);
```


Using the `imshow(deadpx)` command the dead pixels can be viewed as in Figure 15:

Figure 15. Image that is used to separate the dead pixels and spots from the frames at 150 cfm

The frames that were to be worked on were converted to grayscale image format, and then they were converted to a binary image by using the grayness threshold value. An example for the 247th frame in the video is given below:

```
>> im1=247; % Select the frame to work on
>> image = read(obj, im); % Read the frame into image matrix
>> i3=rgb2gray(image); % Convert the frame into grayscale
>> levelf = graythresh(i3); % Calculate the global grayness threshold
>> frame = 1-im2bw(i3,1.16*levelf)-deadpx; %Convert to binary image and remove
the dead pixels.
```

The raw image at 247th frame is shown in Figure 16. After removing the dead pixels and converting to binary format, the image obtained is given in Figure 17. It was observed that dead pixels could not be removed completely unless another command was used to remove the small pixels, which were residue from the removed spots:

```
>> frame1 = bwareaopen(frame, 15); %Remove small pixels (15 is selected here)
```

The resulting cleaned up frame is shown in Figure 18. This figure shows that the particles were removed from the black background quite accurately.

Figure 16. Raw image

Figure 17. Image after removing dead pixels at 150 cfm

Figure 18. Image after removing small pixels

Finding the centroid and total area of each particle

The centroids and areas of particles were calculated using the “regionprops” command in Matlab. Using the frame shown in Figure 18, the “regionprops” command calculates the connected pixels in different regions (white areas) in a binary image and outputs the actual number of pixels in a region, which gives the area of that region. The center of mass of each region is also calculated; this gives the axial and vertical coordinates of the centroids. The example for the frame shown in Figure 18 is given below:

```
>>s =regionprops(frame1, 'Area', 'centroid');
>>centroids = cat(1, s.Centroid);
```

The centroids are listed in pixel units as given in Table 2.

Table 2. Coordinates of centroids in pixel units

X	Y
34.98	471.39
70.06	499.73
84.35	323.11
111.87	49.663
473.52	129.97

Once the centroids were detected, the total area occupied by each particle was calculated. This was done in conjunction with tracing the boundaries of the particles using the “bwboundaries” command in Matlab. The script used for this is given as:

```
>>factor=8.8/size(image,1);%Scaling factor from pixels to mm (for 8.8mm wide
telecentric lens)
>> [B,L] = bwboundaries(frame1,'noholes');
for k=1:size(centroids,1)
b = B{k};
subplot(2,2,1),plot(b(:,2),b(:,1),'g','LineWidth',1);
sareas=sareas+s(k).Area;%Area of each particle is summed into sareas variable
in pixel units
i=i+1;
particles(i,1)=centroids(k,1);%horizontal coordinate
particles(i,2)=centroids(k,2);%vertical coordinate
particles(i,3)=im;%frame number
end;
```

The “bwboundaries” function was used to trace the boundaries of each white region in a binary image. The area of each particle was summed into the “sareas” variable to calculate the total area occupied by the particles. The centroids of each particle was assigned to a new variable called “particles(i)” where x coordinates, y coordinates, and the frame number were stored. Figure 19 shows the boundaries of particles superimposed on the raw image and the centroids plotted on the frame in red circles.

Figure 19. Boundaries of particles are traced in green color (on the left) while centroids of each particle are plotted on frame using red circles (on the right)

Calculating the instantaneous solid volume fraction

The total area of particles was used to calculate the solid volume fraction as $\varepsilon = \frac{\sum_i^n A_i}{A_t}$ where for the given case above, $n=5$, $A_t=512 \times 512 \text{ pixels}^2$, which gives $\varepsilon = 0.5173\%$. In Matlab this was done by using the commands below:

```
>>svolume(im)=sareas/(size(frame1,1)*size(frame1,2));
```

The time corresponding to the current frame was calculated using:

```
>>xvolume(im)=(im)/obj.FrameRate;%Time in seconds
```

where “im” is the frame number.

Tracing the trajectory of each particle

The particle tracking was done using a code developed at the Physics Department at Georgetown University (<http://physics.georgetown.edu/matlab/>). The tracing code was placed in the same folder with the Matlab code and it was called an external function “track” as in:

```
>>res=track(particles,100);
```

The “track” function required that the coordinates of centroids were listed in a variable (“particles” used here), with the associated frame numbers. Then the “track” algorithm identified the same particles that belong to different frames and created tags for each of them. A new list

was created that included the coordinates of particles, the associated frame numbers, and the particle tags. Below such a list is shown for the frames 247 and 248:

Table 3. Output of the "track" algorithm

X Coordinate	Y Coordinate	Frame #	Particle Tag #
34.984	471.394	247	1
33.860	407.039	248	1
70.069	499.734	247	2
72.481	437.603	248	2
84.355	323.118	247	3
84.020	283.667	248	3
111.872	49.663	247	4
110.718	5.555	248	4
473.529	129.977	247	5
466.061	58.231	248	5
1.421	193.789	248	6
28.182	130.914	248	7
155.954	218.818	248	8

In Figure 20 two consecutive frames, frame 247 and frame 248, were superimposed in order to demonstrate the result of the “track” algorithm. The result of the “track” algorithm is also shown in Figure 20 with red lines to show the trajectory of the particles between the two frames.

In our analysis the particle velocities were calculated after the tracking step was finished. This was done by calculating the distances traveled by each individual particle in vertical and horizontal directions and multiplying with the frame rate of the video. This reads in the code as:

```
>>for i=1:size(res(:,4),1)-1 %for the number of all particles that moved
if (res(i+1,4) == res(i,4)) % if the ID of the next particle is the same with
the current one
    velx(i+1,2)=(res(i+1,1)-res(i,1))*factor*obj.FrameRate;%Calculate x-
velocity between two frames
    vely(i+1,2)=(res(i+1,2)-res(i,2))*factor*obj.FrameRate;%Calculate y-
velocity between two frames
    vel(i+1,2)=(sqrt(velx(i+1,2)^2+vely(i+1,2)^2))*factor*obj.FrameRate;%Ca
lculate total-velocity between two frames
end;
velx(i+1,1)=res(i+1,3);%Assign the time stamp
vely(i+1,1)=velx(i+1,1);%Assign the time stamp
vel(i+1,1)=velx(i+1,1);%Assign the time stamp
end;
```

Figure 20. Frame 248 superimposed on frame 247 with trajectories shown

Experimental Results

The first images of polystyrene particles at two different inlet flow rates were obtained using the CCD camera. Using the 8x8 binning mode of the CCD camera, several images were captured at the riser with polystyrene particles. Figure 21 shows a set of images captured at 150 cfm inlet flow rate. Depending on the distance to the focal, the shadow of the particles had a halo area surrounding them. The number density of particles in these shots was found to be too small to achieve significant size cluster formations. The frame area of the camera was also small, which did not allow for the video capture of a high number of particles.

The flow rate was increased from 150 cfm to 180 cfm to obtain a higher density flow compared to the one shown in Figure 21. Although it was observed that a much denser flow field can be obtained at this flow rate (see Figure 22), the highest frame rate attained at 8x8 binning was not sufficient to obtain still images of particles.

Figure 21. Images of polystyrene particles at 150 cfm

Figure 22. Images of polystyrene particles at 180 cfm

High speed imaging was performed using the Vision Research V5.0 high speed camera with the telecentric lens for two flow conditions at 150 cfm and 180 cfm inlet flow rate. The videos recorded were analyzed using the Matlab script (described above) in order to calculate solid volume fractions and particle velocities. The solid volume fraction distribution during the 150 cfm tests are plotted in Figure 25. The solid fraction distribution was observed to have an average value of 0.3%.

Figure 23. Image of the polystyrene particles in the riser obtaining using Vision 5.0 high-speed camera

(a) Frame 1

(b) Frame 2

(c) Frame 3

Figure 24. Three consecutive frames at 180 cfm with the boundaries traced

The trajectories obtained at 150 cfm inlet flow rate have shown that the tracking algorithm was successful in identifying the same particles between different frames; however, there have been some errors observed, as shown in blue boxes in Figure 25. The velocity values calculated, using the displacements shown in Figure 25, were calculated by obtaining the magnitude of the resultant velocity vector of each particle.

Figure 25. Solid volume fraction profile at 150 cfm

Figure 26. Particle trajectories and mean particle velocity profile at 150 cfm

Figure 27. Solid volume fraction profile at 180 cfm

Similar results are shown in Figure 27 for the inlet flow rate at 180 cfm where a higher solid volume fraction was obtained compared to the flow at 150 cfm. The average value was 3%.

At this flow regime, the formation of clusters was observed although the number of particles that made up the clusters was not large. One problem with this was that the tracking algorithm could not differentiate the boundaries between attached particles; this is why the cluster was treated as one particle. The trajectories obtained during the first seconds of the 180 cfm trial were plotted in Figure 29.

Figure 28. Clusters at 180 cfm were treated as single particles in the tracking algorithm

Figure 29. Trajectories and particle mean velocities during the 180 cfm test

Conclusions

In this research various imaging techniques have been tested in order to analyze particle flows in a circulating fluidized bed (CFB) built at FIU. The analysis included calculation of solid volume fractions in the riser section of the CFB that was used to identify cluster formations at different flow regimes. It was observed that the shadow sizing method was able to provide information about the temporal variation of the volume fraction in the riser. This was achieved by using a Matlab script that can identify the particle boundaries at consecutive frames captured with a high-speed camera. Additionally the code had the capability to track particle centroids that would yield instantaneous particle velocity calculations. The solid volume fraction and particle velocities can be related to granular temperature in order to identify cluster formations using the shadow sizing method.

However, the current analysis needs to be improved by implementing calibration of the shadow halo area for particles that are out of the field of focus. This is required in order to calculate the areas of particles more accurately. Another issue with the current analysis was that the particles with adjacent boundaries were treated as a single particle. The interface of particles that are in contact should be separated in order to determine the number of particles that form the clusters.

Lastly, there are some changes required for the current CFB design at FIU such as upgrading the solid particle injection system to avoid clogging, using a larger capacity memory storage to record continuous videos at high speeds and a reliable filtration system for the cyclonic particle separator.

References

- Brereton, C. M. H., Grace, J. R., (1993), "Microstructural aspects of the behavior of circulating fluidized beds," *Chem. Eng. Sci.* 48, pp. 2565–2572.
- Cao, J., Ahmadi, G., (1995), "Gas-particle two-phase turbulent flow in a vertical duct," *Int. J. Multiphase Flow*, 21, pp. 1203–1228.
- Clayton Crowe, Martin Sommerfeld and Yutaka Tsuji, 1998, *Multiphase flows with droplets and particles*, CRC Press LLC.
- Dejin, L., Kwauk, M., Hongzhong, L., (1996), "Aggregative and particulate fluidization - the two extremes of a continuous spectrum," *Chem. Eng. Sci.*, 51, pp. 4045–4063.
- Ding, J., Gidaspow, D., (1990), "A bubbling fluidization model using kinetic theory of granular flow," *AIChE Journal*, 36, p. 523.
- Gidaspow, D., Huilin, L., (1998), "Equation of state and radial distribution functions of FCC particles in a CFB," *AIChE Journal*, 44, p. 279.
- Hartge, E. U., Rensnet, D., Werther, J., (1988), "Solids concentration and velocity patterns in circulating fluidized beds", in: P. Basu, J.F. Large (Eds.), *Circulating Fluidized Bed Technology II*, Pergamon, New York, pp. 165–180.
- Helland, E., Occelli, R., Tadrist, L., (2000), "Numerical study of cluster formation in a gas-

particle circulating fluidized bed,” *Powder Technology*, 110, pp. 210–221.

Hoomans, B.P.B., Kuipers, J.A.M., Briels, W.J.W., Van Swaaij, P.M., (1996) “Discrete particle simulation of bubble and slug formation in a two-dimensional gas-fluidized bed: a hard-sphere approach,” *Chem. Eng. Sci.*, 51, pp. 99–118.

Horio, M., Morshita, K., Tachibana, O., Murata, M., (1988), “Solid distribution and movement in circulating fluidized beds,” P. Basu, J.F. Large (Eds.), *Circulating Fluidized Bed Technology II*, Pergamon, New York.

Ito, M., Tsukada, M., Shimamura, J., Horio, M., (1998), “Prediction of cluster size and slip velocity in circulating fluidized beds by a DSMC model,” Fan, L.S., Knowlton, T.M. (Eds.), *Fluidization IX. Engineering Foundation Inc.*, New York, pp. 525–532.

Liang-Shih Fan and Chao Zhu, 1998, *Principles of Gas-Solid Flows*, Cambridge Series in Chemical Engineering.

Lun, C., Savage, S., Jeffery, D., and Chepurnity, N., (1984), “Kinetic theory for granular flow: Inelastic particles in Couette flow and slightly inelastic particles in a granular flowfield,” *Journal of Fluid Mechanics*, 140, pp. 223.

Noymer, D. P., Glicksman, L. R., (1998), “Cluster motion and particle-convective heat transfer at the wall of a circulating fluidized bed,” *Int. J. Heat Mass Transfer*, 41, pp. 147–158.

Ouyang, J., Li, J., (1999), “Discrete simulations of heterogeneous structure and dynamic behavior in gas–solid fluidization,” *Chem. Eng. Sci.*, pp. 5427–5440.

Sharma, K. A., Tuzla, K., Matsen, J., Chen, J. C., (2000), “Parametric effects of particle size and gas velocity on cluster characteristics in fast fluidized beds,” *Powder Technology*, 111, pp. 114–122.

Sinclair, J.L., Jackson, R., (1989), “Gas–particle flow in a vertical pipe with particle–particle interactions,” *AIChE Journal*, 35, p. 1473.

Soong, C. H., Tuzla, K., Chen, J. C., (1993), “Identification of particle clusters in circulating fluidized beds,” in: A.A. Avidan (Ed.), *Circulating Fluidized Bed Technology IV Engineering Foundation*, New York, pp. 615–620.

Spahn, F., Schwarz, U., and Kurths, J., (1997), “Clustering of granular assemblies with temperature dependent restitution under Keplerian differential rotation,” *Physical Review Letters*, 78, pp. 1596–1599.

Tsuo, Y. P., Gidaspow, D., (1990), “Computation of flow patterns in circulating fluidized beds,” *AIChE Journal*, 36, p. 885.

Tsuji, Y., Kawaguchi, T., Tanaka, T., (1993), “Discrete particle simulation of two-dimensional fluidized bed,” *Powder Technology*, 77, pp. 79–87.

Van Wachem, B.G.M., Schouten, J.C., Van den Bleek, C.M., Krishna, R., Sinclair, J.L., (2001), “CFD modeling of gas fluidized beds with a bimodal particle mixture,” *AIChE J.*, pp. 1292–1301.

Van Wachem, B.G.M., Van der Schaff, J., Schouten, J.C., Krishna, R., Van den Bleek, C.M., (2001), “Experimental validation of Lagrangian–Eulerian simulations of fluidized beds,” *Powder Technology*, 116, pp. 155–165.

Yerushalmi, J., Glicksman, M. J., Graff, R. A., Dobers, S., Squires, A. M., (1975), "Production of gaseous fuels from coal in the fast fluidized bed," *Fluid. Technol.*, pp. 437–469.

Zhou, H., Flamant, G., Gauthier, D., Lu, J., (2002), "Lagrangian approach for simulating the gas-particle flow structure in a circulating fluidized bed riser," *Int. J. Multiphase Flow*, 28, pp. 1801–1821.

Milestone Status

Project Gantt Chart (Table 1) illustrates the progress toward completing the project tasks.

Table 1. Project Gantt Chart (simplified)

Task Description	Time	Start	End	Status as of 6/30/09
Task 1- Develop Project Management Plan	23 days	1/1/2007	1/31/2007	Completed
Task 2- Experimental setup and testing of circulating fluidized bed riser	244 days	2/1/2007	1/8/2008	Completed
Task 3- Configure and test shadow sizing system	65 days	7/30/2007	10/26/2007	Completed
Task 4- Experimental data collection	75 days	1/9/2008	4/22/2008	Completed
Task 5- Image analysis to obtain velocity and void measurements	113 days	1/23/2008	6/27/2008	Completed
Task 6- Identification of void fraction associated with maximum granular temperature	45 days	03/09/2008	06/08/2009	Completed
Task 7- Mathematical analysis of collected data to identify criteria for cluster size	10 days	06/08/2008	06/20/2009	Completed

Project Budget Status

Table 4 and corresponding Figure 30 illustrate the cost status of the project through June 2009.

Total actual incurred cost for the current quarterly period is \$20,769. The cumulative expenditures amount to \$126,813 versus the baseline estimate of cumulative cost \$134,928.

Three primary drivers for under running the budget are: (1) needed to look at higher densities of particles than originally analyzed and presented at the last DOE NETL review meeting; (2) a Ph.D. student left FIU and a new researcher identified at the same time as project end date was passed without resolution of a No Cost Extension (NCE); and (3) only the shadow sizing technique was used while measurements with the PIV method were not considered.

Table 4. Project Financial Status (from Project Management System)

			Jan-08	Feb-08	Mar-08	Apr-08	May-08	Jun-08	Jul-08	Aug-08	Sep-08	Oct-08	Nov-08	Dec-08	Jan-09	Feb-09	Mar-09	Apr-09	May-09	Jun-09
	Plan Total		1920	10000	4100	5334	4134	6134	4150	4150	4100	4050	3952	3930	3600	3480	3292	3172	2760	2200
	Plan Cumulative		62390	72390	76490	81824	85958	92092	96242	100392	104492	108542	112494	116424	120024	123504	126796	129968	132728	134928
Task 1	Develop Project Management Plan	Plan	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		Actual	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Task 2	Experimental setup and testing	Plan	1100	5000	2000	2400	588	1287	0	0	0	0	0	0	0	0	0	0	0	0
		Actual	1093	4495	2140	2400	500	2560	0	1850	0	0	0	0	0	0	3250	2050	0	0
Task 3	Configure and test shadow sizing system	Plan	820	5000	1500	620	630	620	0	0	0	0	0	0	0	0	0	0	0	0
		Actual	812	5334	1400	840	200	1520	0	0	0	0	0	0	0	0	0	0	0	0
Task 4	Experimental data collection	Plan	0	0	600	1224	1200	779	1394	1387	984	980	956	944	840	840	760	600	400	0
		Actual	0	0	625	1800	500	2500	0	100	-3107	900	940	0	900	0	900	2900	2308	0
Task 5	Image analysis: velocity and void	Plan	0	0	0	1090	1041	1724	1395	1393	1230	1200	1196	1186	960	840	800	760	560	400
		Actual	0	0	0	928	500	2310	164	785	300	1100	1180	0	311	0	300	966	2420	1190
Task 6	Identification of void fraction associated	Plan	0	0	0	0	675	1724	1361	1370	1236	1230	1200	1180	1080	960	920	852	840	840
		Actual	0	0	0	0	31	1860	0	289	200	200	424	0	0	0	0	1120	2250	900
Task 7	Mathematical analysis of collected data	Plan	0	0	0	0	0	0	0	0	650	640	600	620	720	840	812	960	960	960
		Actual	0	0	0	0	0	0	0	0	40	201	0	0	0	0	0	1240	1816	1608
	Actual Total		1905	9829	4165	5968	1731	10750	164	3024	-2567	2401	2544	0	1211	0	4450	8276	8794	3698
	Actual Cumulative		62375	72204	76369	82337	84068	94818	94982	98006	95439	97840	100384	100384	101595	101595	106045	114321	123115	126813
	Variance		15	186	121	-513	1890	-2726	1260	2386	9053	10702	12110	16040	18429	21909	20751	15647	9613	8115

Figure 30. Project Cost for the Period Nov'07-June'09

Appendix

Matlab code for image processing

```

closeall;
clearall;
path(path, 'H:\NETL\Image Processing');
%-----
%1) Read the Video file - (*.mov is not supported)
obj = mmreader('H:\NETL\Experiment
Files\050809\050809_T10_2000fps_150_512512.avi');
numFrames = obj.NumberOfFrames; % Get the number of frames
image1 = read(obj, 1600); % Select the image to work on to find dead
pixels
%1-1) Read the file to be worked on
%obj2 = mmreader('H:\NETL\Experiment
Files\050809\050809_T12_2000fps_185_512512_Reduc_f.avi');
%numFrames2 = obj2.NumberOfFrames; % Get the number of frames
%-----
%2) Find the dead pixels
i2=rgb2gray(image1); % Convert to grayscale
leveld = graythresh(i2); % Calculate the global threshold
deadpx = 1-im2bw(i2,1.1*leveld); % Convert to binary image
%imshow(deadpx)
%-----
%3) Select a frame and remove the dead pixels
if(1)
figurenumber=0; figurenumber=figurenumber+1;fig=figure(figurenumber);
%   set(fig,'DoubleBuffer','on');
%   set(gca,...% 'XLim',[0 1], 'YLim',[0 100],...
%       'NextPlot','replace','Visible','off')
%   mov = avifile('particles150cfm4.avi','compression','none','fps',5);
%   mov.quality = 100;
im1=245;im2=250;
i=0;
for im=im1:im2;
image = read(obj, im); %
%figure;imshow(image)
i3=rgb2gray(image); % Convert to grayscale
levelf = graythresh(i3); % Calculate the global threshold
frame = 1-im2bw(i3,1.16*levelf)-deadpx; % Convert to binary image with
white background
%figure;imshow(frame);title('Image after removing dead
pixels','fontsize',12);
frame1 = bwareaopen(frame, 15);%Remove small pixels (60 is selected to
remove the deadpixels)
%figure;imshow(frame1);title('Image after removing small
pixels','fontsize',12);
%-----
%4) Find centroids and areas and display on original image
s = regionprops(frame1, 'Area', 'centroid');
centroids = cat(1, s.Centroid);
subplot(2,2,1), imshow(i3)
hold(imgca, 'on')

```

```

%if size(centroids) > 0
%   subplot(2,2,2), plot(imgca,centroids(:,1), centroids(:,2), 'r.')
%end;
%-----
%5) Trace the boundaries
sareas=0.0;
factor=8.8/size(image,1);%Scaling factor from pixels to mm (for 8.8mm
wide telecentric lens)
[B,L] = bwboundaries(frame1,'noholes');
for k=1:size(centroids,1)
b = B{k};
    subplot(2,2,1),plot(b(:,2),b(:,1),'g','LineWidth',1);
sareas=sareas+s(k).Area;%Area of each particle is summed into sareas
variable in pixel units
i=i+1;
particles(i,1)=centroids(k,1);%horizontal coordinate
particles(i,2)=centroids(k,2);%vertical coordinate
particles(i,3)=im;%frame number
end;
%-----
%6) Calculate solid volume fraction:
svolume(im)=sareas/(size(frame1,1)*size(frame1,2));
%xvolume(im)=(im-im1)/obj.FrameRate;%Time in seconds
xvolume(im)=(im)/obj.FrameRate;%Time in seconds
subplot(2,2,[3 4]), plot(xvolume(im1:im), svolume(im1:im),'b-o')
xlabel('Time (s)','fontsize',14);ylabel('Solids Volume
Fraction','fontsize',14);set(gca,'FontSize',14);
%-----
%7) Plot the motion of centroids if the frame is not empty
if isempty(centroids) < 1
    subplot(2,2,2),
plot(factor*centroids(:,1),factor*centroids(:,2),'ro'),axis([0 8.8 0
8.8],'square');set(gca,'YDir','reverse');grid on;
xlabel('Horizontal axis (mm)','fontsize',14);ylabel('Vertical axis
(mm)','fontsize',14);title('Motion of the
centroids','fontsize',14);set(gca,'FontSize',14);
end
%   F = getframe(gcf);
%   mov = addframe(mov,F);
end;
%mov = close(mov);
if(1)
res=track(particles,100);%(The value 100 has to be adjusted according
to the video. The bigger the better.)
figurenumber=figurenumber+1;fig=figure(figurenumber);
for i=1:size(res(:,4),1)-1 %for the number of all particles that moved
if (res(i+1,4) == res(i,4)) % if the ID of the next particle is the
same with the current one
    velx(i+1,2)=(res(i+1,1)-
res(i,1))*factor*obj.FrameRate;%Calculate x-velocity between two frames
vely(i+1,2)=-(res(i+1,2)-res(i,2))*factor*obj.FrameRate;%Calculate y-
velocity between two frames

vel(i+1,2)=(sqrt(velx(i+1,2)^2+vely(i+1,2)^2))*factor*obj.FrameRate;%Ca
lculate total-velocity between two frames

```

```

        subplot(1,2,1),plot(factor*res(i:i+1,1),factor*res(i:i+1,2),'r-
o'),axis([0 8.8 0 8.8],'square');set(gca,'YDir','reverse');grid
on;holdon;
xlabel('Horizontal axis (mm)','fontsize',14);ylabel('Vertical axis
(mm)','fontsize',14);title('Trajectories of the
centroids','fontsize',14);set(gca,'FontSize',14);
end;
velx(i+1,1)=res(i+1,3);%Assign the time stamp
vely(i+1,1)=velx(i+1,1);%Assign the time stamp
vel(i+1,1)=velx(i+1,1);%Assign the time stamp
end;
velsort=sort(vely,1);
    lim1=1;
fori=1:size(res(:,4),1)-1
if (velsort(i+1,1) >velsort(i,1)) || (i == size(res(:,4),1)-1)
    lim2=i;
ifi == size(res(:,4),1)-1
lim2=size(res(:,4),1);
meanvel(velsort(i+1,1)+1)=mean(velsort(lim1:lim2,2));%calculate the
mean of velocities at each frame
maxvel(velsort(i+1,1)+1)=max(velsort(lim1:lim2,2));%calculate the max
of velocities at each frame
minvel(velsort(i+1,1)+1)=min(velsort(lim1:lim2,2));%calculate the min
of velocities at each frame
stdvel(velsort(i+1,1)+1)=std(velsort(lim1:lim2,2));%calcualte the std
of velocities at each frame
frametime(velsort(i+1,1)+1)=velsort(i,1);
else
meanvel(velsort(i+1,1))=mean(velsort(lim1:lim2,2));%calculate the mean
of velocities at each frame
maxvel(velsort(i+1,1))=max(velsort(lim1:lim2,2));%calculate the max of
velocities at each frame
minvel(velsort(i+1,1))=min(velsort(lim1:lim2,2));%calculate the min of
velocities at each frame
stdvel(velsort(i+1,1))=std(velsort(lim1:lim2,2));%calcualte the std of
velocities at each frame
frametime(velsort(i+1,1))=velsort(i,1);
    lim1=lim2+1;
end
end
end
        subplot(1,2,2);%plot(velsort(:,1),velsort(:,2),'ko');
holdon;
%errorbar(frametime,meanvel,stdvel);
        plot(frametime(im1+1:im2+1)/obj.FrameRate,meanvel(im1+1:im2+1),'k-
o');plot(frametime(im1+1:im2+1)/obj.FrameRate,maxvel(im1+1:im2+1),'b-
+');plot(frametime(im1+1:im2+1)/obj.FrameRate,minvel(im1+1:im2+1),'r-
+');
xlabel('Time (s)','fontsize',14);ylabel('Particle Velocity
(mm/s)','fontsize',14);axis square;%axis([0 max(velsort(:,1)) 0
max(velsort(:,2))],'square')
        legend('Mean particle velocity (mm/s)','Maximum particle velocity
(mm/s)','Minimum particle velocity
(mm/s)','fontsize',14,0);set(gca,'FontSize',14);
%-----
end
end

```