**Segmentation, characterization and superimposition of PSP damage-induced dental X-ray artifacts**

Wenda Zhao

Contributors: Hershel Stark, John Valen

Pascal Tyrrell Lab  
Department of Medical Imaging  
University of Toronto

**Introduction**

Phosphor storage plates (PSPs) are commonly used sensors for digital intraoral radiography. They are made of a polyester base coated with phosphor particles which store a latent image in the form of energy when excited by X-ray. The energy is then released as blue fluorescent light under a helium-neon laser beam and the image is shown on a computer system1. In comparison to this indirect way of acquiring images, complementary metal oxide semiconductor (CMOS) sensors are directly connected through a cable to the computer and can display the image immediately after exposure. PSPs have the advantage of allowing easier placement in the oral cavity and causing less discomfort to the patients2. These sensors are meant to be reusable. However, after a prolonged period of use, damages accumulate on the plates and can introduce artifacts to the acquired images (see figure 1). CMOS sensors, on the other hand, are free of such damages. A study in Turkey showed that 20% of the images obtained via PSPs over a six-month period contained visible artifacts3. And PSPs-induced artifacts have been shown to adversely affect the diagnostic ability in digital mammography4. Thus, this has raised the concerns of possible missed diagnoses when using damaged sensors in dental radiography. Clinics are now dealing with this problem by discarding the plates after a certain times of use. Previous study comparing images of PSPs with different used times has suggested that each plate can be used up to 200 times without showing statistically significant changes5. However, in real life this decision is made rather arbitrarily by clinicians. Thus, a standard model needs be established to characterize and quantify the artifacts and to give an accurate prediction as in whether a damaged sensor would impede diagnosis or not based on its characteristics. Clinicians can therefore be informed of when to discard a plate without hurting diagnosis, as a consequence of doing so too late, or wasting money, as a consequence of doing so too early. In order to accomplish this, two preliminary goals need to be achieved in preparation for building such model. The first is the characterization of dental image artifacts, which the model would then take as the input. Previous attempts have been made to categorize these artifacts. In a study done by Kalathingal *et al.*, the authors subjectively rated the damaged plates into five categories of severity in order to study the effect of wiping as a means to remove surface contaminations6. And in a recent paper published by Çalışkan and Sumer, the artifacts were placed into five categories based their appearances: cracking, scratches, peelings of edges, bite-marks and crescent-shaped bending7. These attempts have opened up discussions on this topic but the findings were also at high risk of subjectivity and bias. Thus, a more robust approach to segment and characterize these artifacts is needed. The second goal is that we need to establish a “gold standard” of whether a provided artifact will affect diagnosis. And in order to achieve that, dental images with pathologies needs to be provided in the background of a representative sample of artifacts for dentists to examine. However, acquisition of such images in clinics is unlikely since it is considered unethical to knowingly use damaged sensors on patients. One way solve this problem is to superimpose images of the plates along onto artifact-free teeth images taken with CMOS sensors, and use the superimposed images for testing.

Thus, in preparation for building a model that can predict the effect of dental image artifacts on diagnosis, we face the question of whether we can segment, characterize those artifacts and superimpose them onto clean teeth images. And we hypothesize that we can achieve that by using an algorithmic and machine learning approach. Our objectives are to segment the artifacts from the image; to quantify and characterize them; to cluster the segmented artifacts into meaningful groups based on the characterization; and to superimpose images of PSPs containing artifacts onto clean teeth images taken with CMOS sensors to mimic real dental images taken on damaged PSPs.



**Figure 1**. Phosphor Storage Plates image with artifacts.

**Methods**

**Fast normalized cross-correlation**

There is a “C” shaped object at the right bottom corner of all the PSP images (see figure 1). It is marked on the sensor itself to help with orienting the plate. But for our study, this “C” shape could potentially be picked up by the segmentation algorithm and interfere with the process. Thus, we utilized a scikit-image8 package for template matching to locate and remove the “C”. The algorithm uses fast normalized cross-correlation9 to find the occurrence of a temple in an image. We cropped out a 40 pixels by 40 pixels piece of one original PSP image with the “C” in the middle to serve as a template for the algorithm. The algorithm returns a correlation matrix showing the similarity between the template and pixels across the entire tested image, with the maximum being the centre of the matched area. We then set all pixels within a radius of 20 from the matched point whose value is larger than 45 to a value of 20.

**Density-based spatial clustering of applications with noise (DBSACN)**

DBSCAN is a data clustering algorithm initially proposed by Ester *et al.*10 in 1996, and does, as the name suggests, density-based clustering that groups spatially close points together. The algorithm searches the ɛ neighborhood of a point, which is the multidimensional space within a distance ɛ, calculated by some distance metrics, usually Euclidian distance, from that point. If there are more than a certain number of points, denoted MinP, in that neighborhood, the data point is then considered a core point and a cluster is started. All the points within the neighborhood of a core point are termed reachable from the core point and are all part of that cluster. The ɛ neighborhoods of all cluster members, if again contains more than MinP points, are also termed reachable from the original core point and are thus grouped into the same cluster, so on and so forth. In other words, reachability can be passed on from one core point to another core point in its neighborhood. Any point that is not reachable from other points is grouped into noise. The DBSCAN algorithm contains two hyperparameters: ɛ and MinP. In our study, we chose 20 for both based on clustering results from several testing values. In order to reduced the noise, we also thresholded our images by setting any pixels below 65 to 0. We used a scikit-learn11 package for DBSCAN and our inputs are in the form of sparse matrices: SciPy12 COO matrices of triplet (row, column, intensity) tuples. The algorithm returns a cluster label for each pixel on the image.

**Feature Computation**

We picked the following 9 features to be calculated from each segmented artifacts: number of pixels to represent the area of artifacts; sum of intensity to represent the volume; range along x and y axes to represent the length and width; x and y coordinates of the centroid to represent the location; maximum and median intensity to represent the intensity; and a categorical variable of shape to be either line, triangle, rectangle or circle. Because centrosymmetric locations on the plates are equivalent in clinical settings, we used the location of centroid with respect to the centre of the image instead of to the origin of the coordinate plane, which is usually the bottom left corner. We also omitted minimum intensity as a feature because we applied a lower threshold of 65 during DBSCAN segmentation, which made 65 to be the minimum in many artifacts.

For shape detection, we utilized the contour of the segmented artifacts. An Open CV13 package of contour approximation was used to approximate the contour with a shape of less vertices within a range of error ɛ, which we picked to be 4% of the contour perimeter. The algorithm is an implementation of the Ramer–Douglas–Peucker algorithm14, 15, and it returns a list of vertices for the approximating shape. With two vertices, it means the shape of interest can be approximated by a line; whereas three vertices indicates a triangle; four being a rectangle. We put any artifact that can be approximated by a polygon with more than four vertices into the category of circle.

**K Prototypes**