A Hybrid Clustering Pipeline for

Mining Local Patterns in 3D Point Cloud

Rahim Hossain and Manar D. Samad

**Abstract:** Three-dimensional (3D) imaging provides detailed geometry of real-word objects unlike 2D image texture. The rudimentary form of 3D imaging is point clouds that are distinctly different from image pixels in terms of structure and processing methods. The 3D computer vision literature primarily retrieves global shape patterns in 3D data for object and face recognition tasks. In contrast, mining local deformations patterns in 3D point clouds is a nontrivial task. This paper proposes a computational pipeline for mining local patterns in 3D point clouds and identifying informative segments of point clouds for data dimensionality reduction. We investigate the efficacy of several clustering algorithms in point cloud segmentation and propose a multi-stage clustering pipeline with parametric modeling of local patterns in 3D point clouds. The local patterns have achieved an area under the curve of 0.72 in classifying seven emotional expressions using 3D human facial point clouds. Our pipeline demonstrates the efficacy of raw 3D point coordinates in mining local patterns without involving any feature engineering or deep learning. Therefore, the proposed pipeline can be used 1) to rapidly obtain interpretable and informative local patterns and 2) as a baseline method for learning and evaluating local patterns in 3D point cloud data.

**Introduction:** With recent advances in high-resolution 3D scanning technology and extensive use of LIDAR imaging, 3D point cloud data are now available for many applications. 3D data represent true geometry of real-world objects, which is often compromised in conventional texture of 2D imaging and even in Kinect-based depth sensing. Despite the richness in geometric information, 3D data are unstructured and unordered with inconsistencies in point counts, which turns machine learning (ML) of such data structure challenging. Existing literature either maps 3D geometric features (e.g., depth, curvature) onto 2D grid texture~\cite{} or creates stacks of 2D image slices from 3D model~\cite{} to leverage traditional image processing and recognition pipelines. Such arbitrary 3D to 2D data projections for technical convenience may inadvertently discard useful information of true geometry in 3D data~\cite{Frenet}. Therefore, mining effective patterns directly from 3D data, without introducing lossy 3D-2D data projections, is a nontrivial task.

The task of mining patterns in 3D data has been investigated in objects~\cite{} and human face recognition~\cite{} applications. The analysis of the human face in 3D is an active area of research that facilitates several emerging areas, including computational behavioral sciences~\cite{JOLT paper, 3DASD}, biometrics and affective computing~\cite{}, and forensics and medicine~\cite{xx}. A majority of these applications mine holistic patterns or global shape in 3D to differentiate between a ‘car’ and a ‘table’ or between the face of ‘John’ from that of ‘David’. In contrast, a more challenging task is to recognize local patterns in 3D data that can differentiate between facial deformations related to several emotional expressions. The task of learning local patterns from 3D data has been previously addressed by extracting and analyzing changes in local geometric and shape features~\cite{} . However, many of these analyses require grid-based sampling and interpolation of the 3D face that is an approximation of the actual 3D geometry. This data approximation may retain the global shape but can compromise local geometric patterns. Therefore, the mining of local patterns directly from 3D facial point cloud data is a baseline approach that is to be compared with other sophisticated computational models. The baseline model will reveal the efficacy of local patterns in 3D point clouds before applying any feature engineering. Therefore, it is expected that feature engineering engineered, and deep learning models must outperform this baseline model. In this paper, we propose a baseline model to learn local patterns directly from 3D point cloud coordinates by addressing several challenges as follows.

First, 3D point clouds are inconsistent in resolution and the number of points vary from one cloud to another. A common strategy is to sample and interpolate 3D point cloud on to a uniform 2D grid that facilitates operations similar to pixel-based processing. Instead, we propose distance-based clustering methods as a means to directly segment facial point clouds into distinct and meaningful facial regions. Second, clustering algorithms often yield different labels for the same segment of the point cloud that depends on initial enumeration of the seeds. This makes identifying the same spatial region with a consistent label across different point clouds challenging. In this paper, labels for different segments are consistently identified across all face samples prior to local pattern extraction and classification. Local patterns mined from these segments are compared across faces for classifying seven prototypical facial expressions, including happy, sad, afraid, angry, disgust, surprise, and neutral. Several clustering algorithms have been tested to evaluate the facial segmentation task. Third, to achieve a constant dimensional feature space for point clouds of varying size, we learn the mixture of Gaussian parameters at each segment of the 3D point cloud. These parametric features are used to identify facial segments with most varying local patterns due to facial expressions. The localization and selection of facial segments can facilitate further interpretation and optimization of local feature mining using 3D point clouds. The remainder of the paper is organized as follows. Section II outlines the methodology behind the proposed pipeline, Section III discusses the results. The paper concludes in Section IV.

**Methods:**

We propose a hybrid of clustering algorithms to extract and classify local patterns from 3D point clouds. The local patterns are evaluated in the classification of seven facial expressions. The proposed pipeline is developed and evaluated in Python programming language using the scikit-learn package~\cite{}. The pipeline for feature extraction is shown in Algorithm 1 and is further discussed below.

**Data set:** The proposed pipeline for analyzing 3D point clouds is evaluated using the Binghamton University 3D facial expressions (BU-3DFE) data set~\cite{}. The data set has 3D faces with seven benchmarked facial expressions. The seven prototypical expressions are: happy, angry, afraid, sad, disgust, surprise, and neutral. The 3D facial data are acquired using 3dMD – a commercially available high-resolution 3D facial imaging system~\cite{}. Additionally, the data set provides well cropped 3D faces that are free of outlier points. We extract and subsequently analyze 3D point coordinates (x, y, z) of the provided 3D facial models in WRL (a virtual reality modeling language extension) format. The class labels provided with the data set are used to perform supervised classification of seven facial expressions.

**Segmentation of 3D point clouds:** Image segmentation is an active area of research that studies similarity within a group of pixel intensities or texture features. In contrast to 2D imaging, 3D point clouds form a surface topology with varying point resolution and count. Assuming the symmetric shape of the human face, we segment eight distinct facial regions. First, we employ and evaluate different clustering algorithms and distance metrics to group the 3D facial point clouds into eight distinct facial segments (facial patches). For facial point cloud segmentation, we compare the performances of K-means clustering (with Euclidean and Mahalanobis distances), spectral clustering, and Gaussian mixture model using their corresponding Silhouette plots. The algorithm that yields most meaningful facial segments is chosen for the subsequent local pattern analysis.

\begin{algorithm}

\vspace{5pt}

\caption{Local feature extraction via hybrid clustering}

\begin{algorithmic}

\STATE Input: 3D point cloud, A = $[a]\_{mx3}$

\STATE Output: n-dimensional local feature vector, fvect

\STATE A $\leftarrow$ A - A [A [z] == max(A[z])]

\STATE patches $\leftarrow$ cluster (A, n\\_patch = 8)

\FOR {t = 1 $\rightarrow$ n\\_patch}

\FOR {i = 1 $\rightarrow$ 3}

\STATE patch = patch [t] - centroid (patch[t])

\STATE \{w[i], $\mu$[i, :], cov\\_mat\} = gmm (patch, clusterID = i)

%\STATE = gmm (patch, n\\_cluster = 3)

%\STATE cov\\_matrix = gmm (patch, n\\_cluster = 3)

\STATE $\sigma^2$[i, :] = diag (cov\\_matrix)

\STATE fvect $\leftarrow$ append (w[i], $\mu$[i,:], $\sigma^2$[i,:])

\ENDFOR

\ENDFOR

\end{algorithmic}

\end{algorithm}

**Alignment of 3D point clouds:** Theregistration of 3D point clouds is important, especially when raw 3D coordinate data are used for the pattern analysis. We perform the registration at two levels (See Algorithm 1): 1) the nose tip is identified as the highest Z-axis coordinate that is used to align and adjust all faces such that all nose tips are at the origin of the 3D Cartesian coordinate and 2) after segmentation of the facial patches, all patches are aligned to a common patch centroid.

**Local feature extraction:** The eight 3D facial patches obtained from the previous step are further grouped into Q clusters by learning Q 3D Gaussian distributions. The Gaussian distribution parameters are learned via the expectation maximization (EM) algorithm. The EM algorithm learns a set of Gaussian parameter vectors: S = {w, $\mu$, $\sigma$} for each cluster distribution that maximizes the likelihood function as below.

\begin(equation)

P(x\_j|c\_i) = exp {Gaussian (x – mu\_i) COV},

\end {equation}

which is a similarly measure between the input 3D point coordinate x\\_j and the cluster centroid $\mu\_i$ of the c\\_i cluster.

The soft cluster assignment of each 3D coordinate point x\\_j is determined by the posterior probability P(c\\_i|x\\_j) of the cluster c\\_i given the input x\\_j as below.

\begin(equation)

P(c\_i| x\_j) = \frac{P(P(x\_j|c\_i) \* w\_i)}{\sum\_{i=1}^{i=Q} P(P(x\_j|c\_i) w\_i }

\end {equation}

where, x\\_j is a 3D point in Cartesian coordinate of a facial patch to be clustered, c\\_i represents one of the eight clusters and, w$\in\Re$, $\mu\in\Re^3$, COV$\in\Re^{3x3}$ are weight, mean vector and covariance matrix of each Gaussian distribution corresponding to a cluster c\\_i. A sample point $x\_j$ in a facial patch belongs to the $c\_i$ cluster that yields the maximum P(c\\_i|x\\_j).

There are Q weight parameters (w) for Q Gaussian clusters. The 3D mean vectors ($\mu$) from all Gaussian clusters yield a total of 3XQ parameter values for a facial patch. We extract the diagonal vector from the 3x3 covariance matrix (COV) as $\sigma^2$ of each Gaussian cluster. Total number of $\sigma^2$ parameters for each facial patch is, therefore, 3XQ. Thus, each patch yields a total of (3\*Q + 3\*Q + Q) or 7Q parameters. Thus, individual 3D facial point cloud is represented by 8\*Q\*7 or 56\*Q features where eight is the number of patches per 3D face.

**Importance of local features and facial segments:** The importance of individual 56\*Q features is interpretable since each feature has a physical significance and can be related to a spatial location of the point cloud. An unsupervised way to look at the feature importance is to measure individual feature variances across seven facial expressions of the same human subject. Features and facial patches with low or zero variability suggest that these are not sensitive to local variations in the face. Therefore, they may not be good candidates to look for local patterns. On the other hand, features with highest variances across facial expressions within individual facial patch can identify the facial regions that are most sensitive to local deformations.

**Classification and model evaluation:** We perform classification of seven facial expressions using the one-versus-all scheme. A nested10x2 fold cross-validation (CV) is proposed for the model evaluation where a two-fold inner CV is nested within each of ten-fold outer CV. The two-fold CV split the nine-fold training data 50%-50% to identify the best classifier hyperparameters via grid search. The best model hyperparameters are then used to train the classifier model on the nine-fold data to test on the left-out data fold. We use the gradient boosted classifier model because of its superiority among existing state-of-the-art classifier models. A mean area under the receiver operating characteristics curve (AUC) is used to evaluate the overall classification performance since AUC accounts for both model sensitivity and specificity at various classification scores. The classification performance reveals 1) how good is the point cloud segmentation and 2) how informative are the proposed parametric features in representing local patterns of point clouds.

**Results:**

The proposed pipeline is evaluated using 50 3D point cloud samples per facial expressions with a total of 350 face samples. The key results are discussed below.

**Facial segmentations:** One of the main challenges in facial segmentation is to maintain consistent labels for all facial segments. To be consistent with the cluster labels and locations across 3D facial point clouds, we use a reference face to first obtain a set of centroids for eight segments using the K-means algorithm. These centroids are then used as initial seed locations for segmenting the remainder of the facial point clouds. We apply and compare four clustering routines: 1) K-means Euclidean, 2) K-means Mahalanobis, 3) Gaussian mixture model, 4) spectral clustering with K-means in the facial segmentation task. Figure~\ref{fig:01} shows representative results of facial segmentations. The K-means algorithm identifies most meaningful facial parts with uniform cluster segments as reflected in its respective Silhouette plot. Therefore, we use K-means algorithm with Euclidean distance for facial segmentation in the subsequent steps of the proposed pipeline.

**Figure 1. Facial 3D point cloud segmentation and Silhouette plots using (a, b) K-means (Euclidean), (b) K-means (Mahalanobis), (c) Gaussian mixture model, and (d) spectral clustering. K-means algorithm appears to segment the point cloud in anatomically meaningful way.**

**Feature extraction and interpretation:** Following the segmentation of individual faces into eight patches, we further group each patch into three clusters (Q = 3) using the Gaussian Mixture Model (GMM). The three clusters from each patch yield a total of 24 patches per 3D facial point cloud. Each of the 24 patches is modeled using seven Gaussian parameters (w, 3D $\mu$, and 3D $\sigma$), which yield a total of 168 features (24X7) per 3D point cloud. Table~\ref{tab:01} shows variances across seven facial expressions in three Gaussian parameters (w, $\mu$, $\sigma$) for each Gaussian cluster and facial patch pair. The variance values reveal which facial patches are most effective in capturing local patterns that differentiate one facial expression from another. As shown in Table ~\ref{tab:01}, patch 2 (Lower right cheek) has no variation in the weight of Gaussian cluster 3 (weight = 0) and two clusters are enough to represent local patterns for that patch. The patch 8 (Upper right cheek) shows similar trend as patch 2 suggesting that the cheek regions may be too flat to capture additional variations in the third Gaussian distributions. The patch 1 (the Upper lip and nose region) appears to yield most variances in the $\sigma$ parameter. Other two parameters are mostly varying in the patch 1 and patch 2 (Lower lip and chin), both of which cover the entire mouth region. Intuitively, the mouth region is the most active part in any facial expression. Patches 3 and 4 cover the rigid forehead regions of the face that intuitively yield least variations in the proposed parameter space.

**Table 1. The variance of parameter values across seven facial expressions. Higher the variance, more discriminating is the facial patch or the parameter for local pattern recognition. The best patch for each parameter is also identified in a separate row and the lowest variances for $\mu$ and $\sigma$ parameters are highlighted for each cluster. GMM stands for Gaussian mixture model. w = cluster weight, $\mu$ = cluster mean, $\sigma$ = diagonal of the covariance matrix of the cluster.**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | GMM cluster 1 | | | GMM cluster 2 | | | GMM cluster 3 | | |
| Patch Number | Facial Region | w | $\mu$ | $\sigma^2$ | w | $\mu$ | $\sigma$ | w | $\mu$ | $\sigma$ |
| 1 | Upper lip and nose | 0.017 | 399 | 12128 | 0.021 | 237 | 12804 | 0.016 | 414 | 13821 |
| 2 | Lower right cheek | 0.001 | 379 | 2826 | 0.002 | 411 | 3083 | 0.000 | 420 | 3153 |
| 3 | Left eye with forehead | 0.011 | **1**93 | 6553 | 0.006 | **233** | **2567** | 0.016 | 232 | 8699 |
| 4 | Right eye with forehead | 0.007 | 327 | 5756 | 0.009 | 419 | 3597 | 0.020 | **111** | 9715 |
| 5 | Lower left cheek | 0.001 | 317 | **1260** | 0.003 | 396 | 2687 | 0.003 | 351 | 6028 |
| 6 | Lower  lip and chin | 0.006 | 452 | 5408 | 0.013 | 512 | 9349 | 0.030 | 241 | 7574 |
| 7 | Upper left cheek | 0.003 | 227 | 6352 | 0.003 | 305 | 10860 | 0.004 | 245 | 8605 |
| 8 | Upper right cheek | 0.001 | **192** | 6703 | 0.001 | 270 | 12292 | 0.003 | 457 | **1119** |
| Best Patch |  | 1 | 6 | 1 | 1 | 6 | 1 | 6 | 8 | 1 |

**Classification performance:** In seven-class classification, we have achieved a mean AUC of 0.72 and the confusion matrix is shown in Figure~\ref{fig:03}. The diagonal of the confusion matrix appears dominant suggesting that the proposed pipeline and features are effective in discriminating local patterns for seven facial expressions.

The diagonal of the confusion matrix shows that happy, surprise, neutral, and sad are the best classified expressions. Fear, anger, and disgust expressions are the most challenging expressions to classify presumably they may represent common facial patterns for negative emotions in contrast to positive emotion like happiness. These negative expressions use eyebrows almost equally as the lower part of the face, which may not be well captured in 3D point cloud data. No neutral and sad expressions are confused as the happy expression. However, sad faces are predicted as neutral equally as neutral faces are confused as sad. The expression of sadness in face can be mild and appear similar to neutral expressions as they are confused during classification. These results show what 3D point clouds can yield at baseline without any feature engineering.

****

**Figure 3.** Confusion matrix for classifying seven prototypical facial expressions using parametric modeling of 3D point cloud data.

**Conclusions:**

}