

1/9

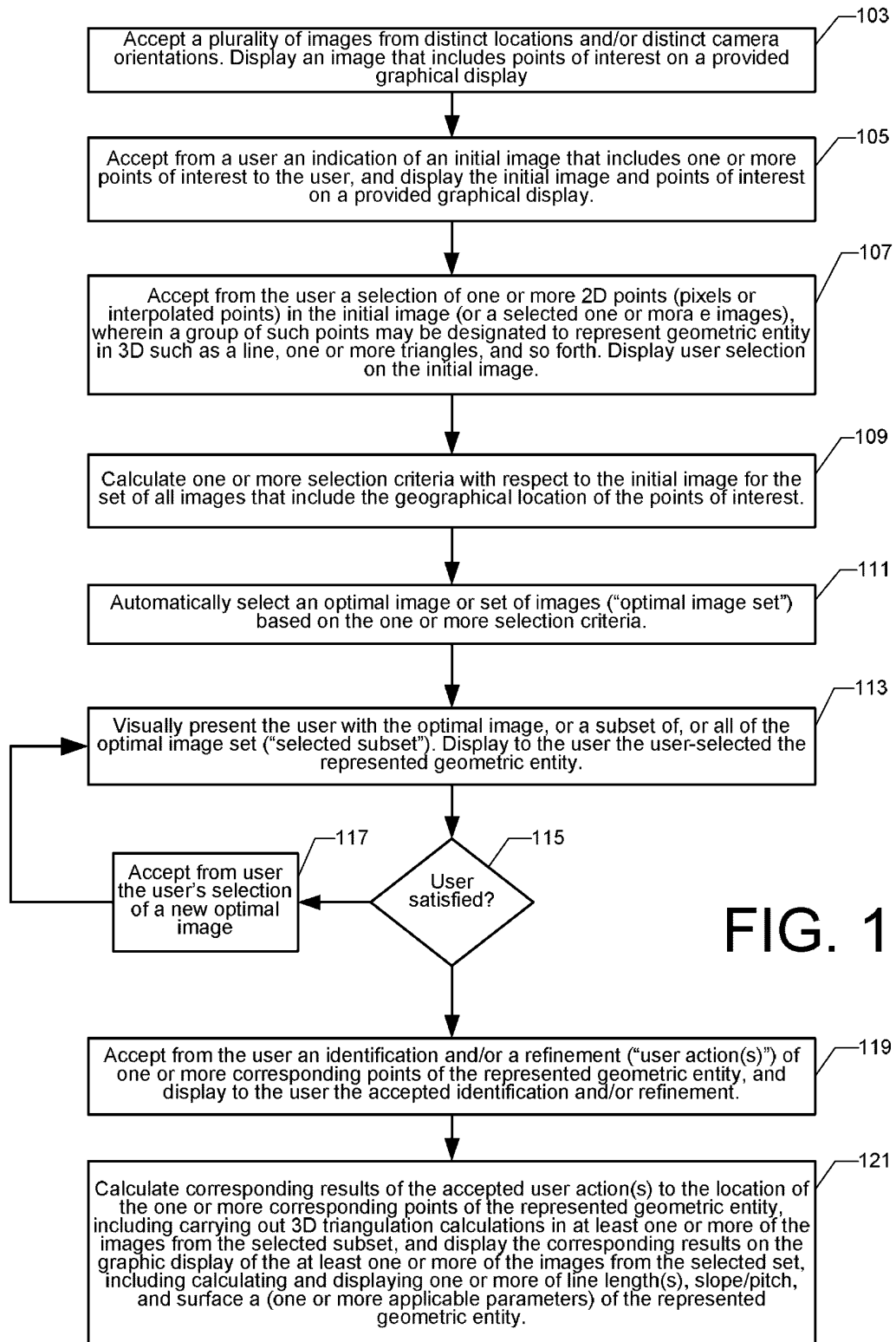


FIG. 1

2/9

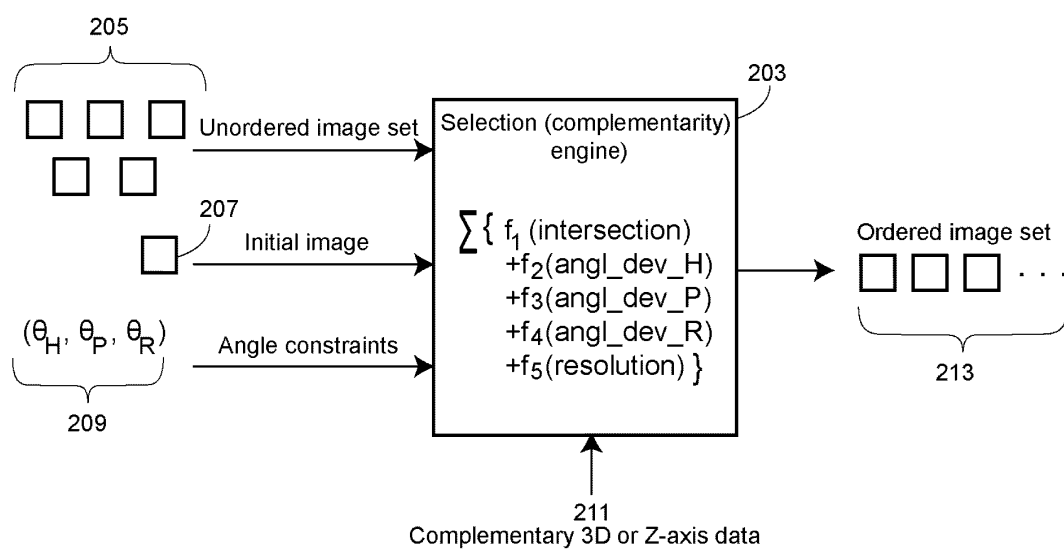


FIG. 2

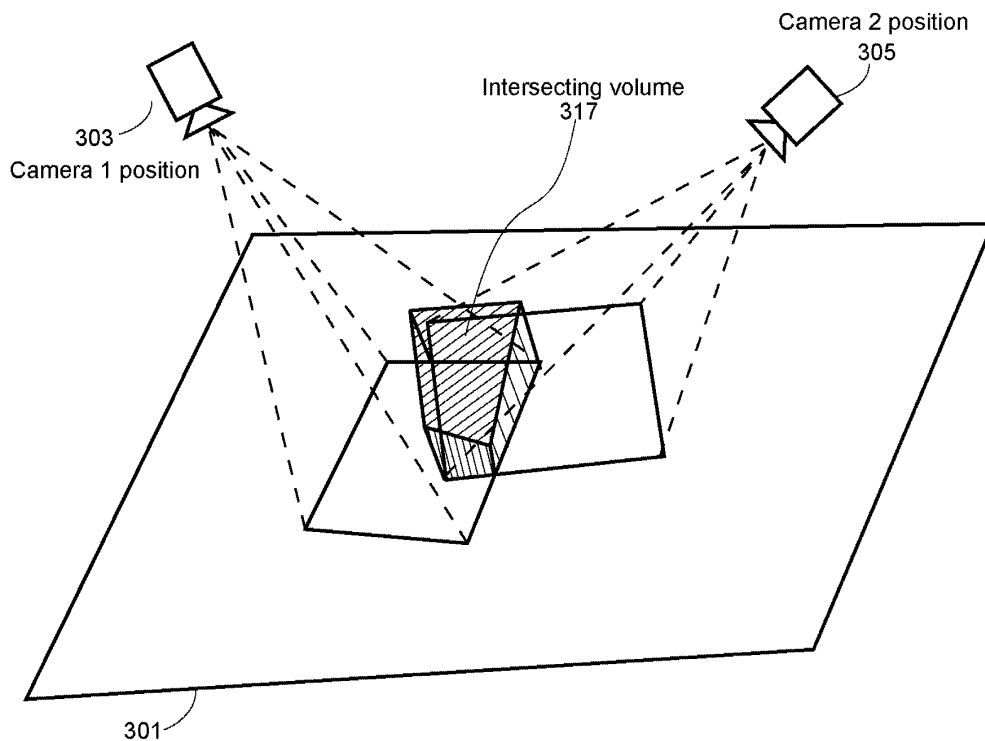


FIG. 3

3/9

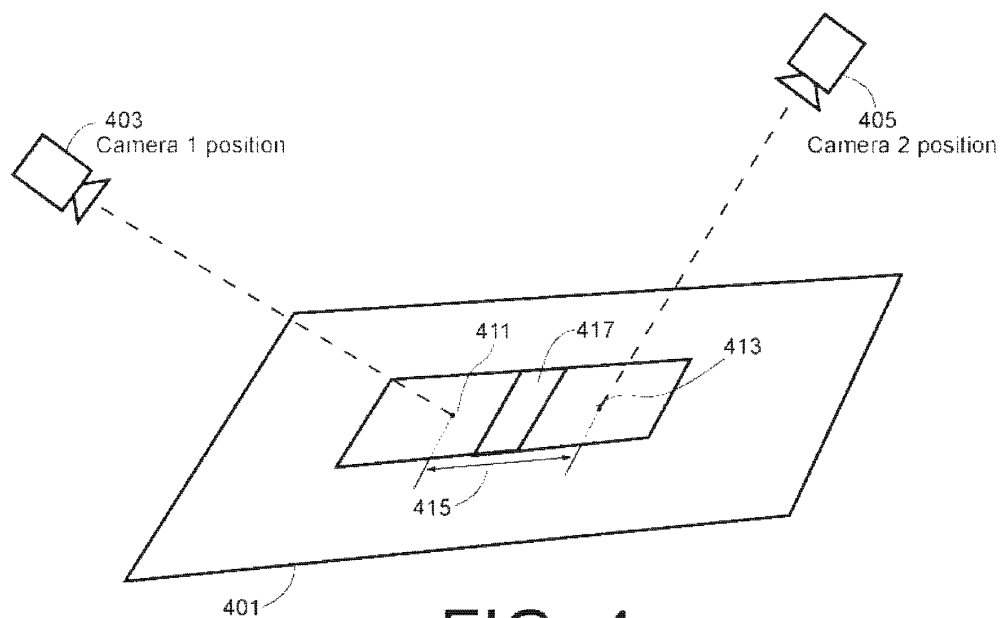


FIG. 4

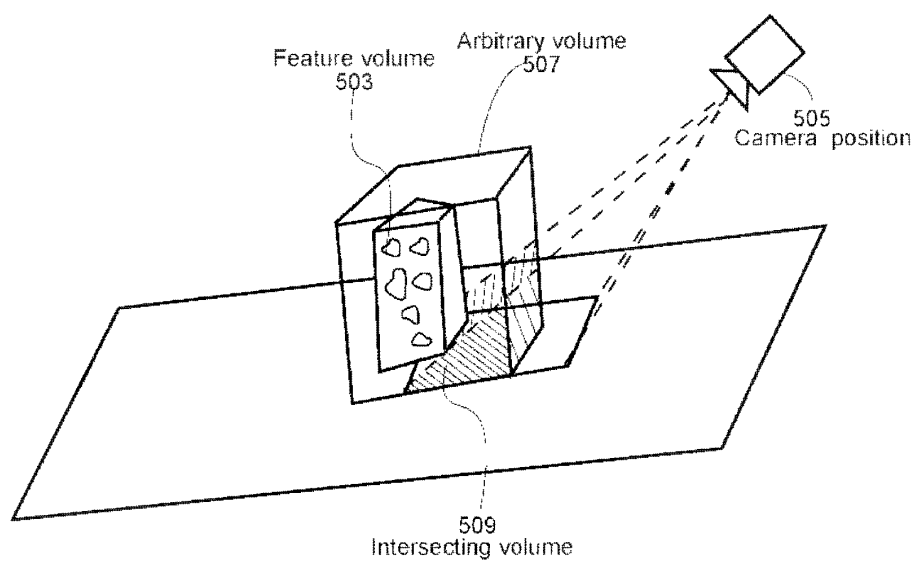


FIG. 5

4/9

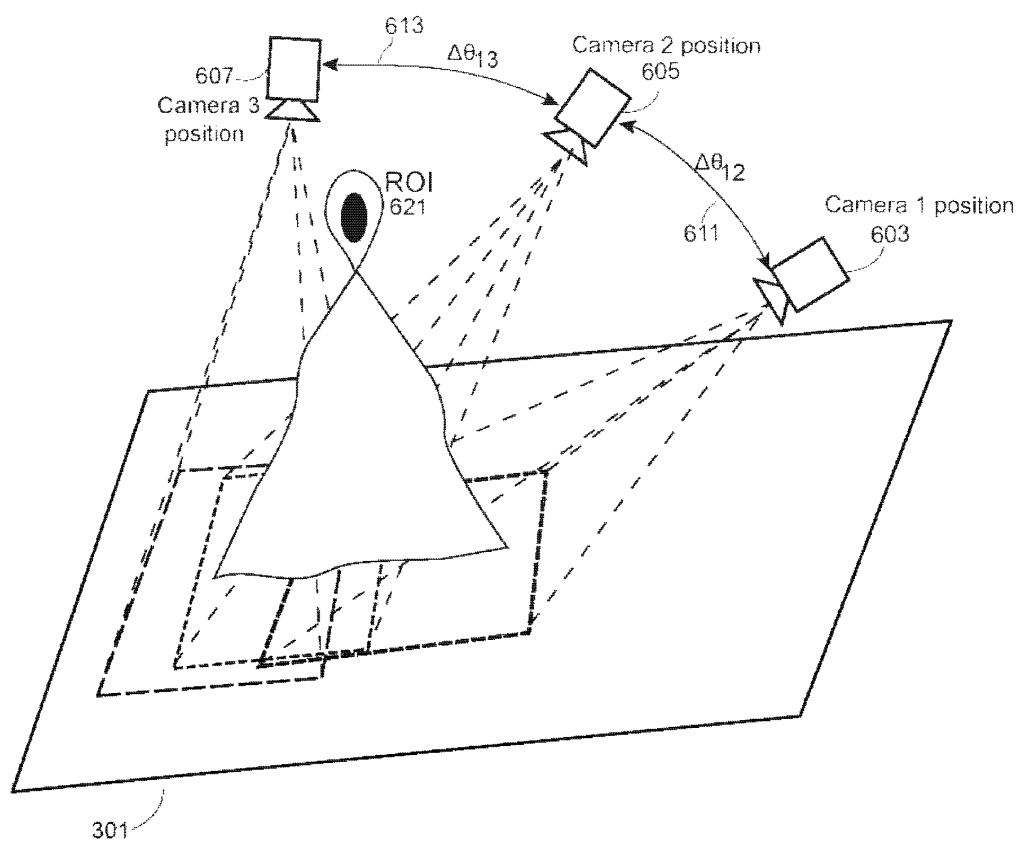


FIG. 6

```

// Photo sorting criteria:
// -----
// * Position: distance from the desired position in meters
// * Heading: deviation from given heading in degrees
// * Pitch: deviation from given pitch in degrees
//
// Photos are given a score for each criteria.
// Scores can be positive or negative, sometimes the goal is to have more
// deviation, sometimes less.
// Combined score for each photo determines sort order (higher scores first).

// eslint-disable-next-line max-params
const photoScore = (photo, center, heading, pitch, weights) => {
  const positionScore = getMetersFromCenter(photo, center) * weights.position;
  const headingScore = degreesFromHeading(photo, heading) * weights.heading;
  const pitchScore = angleDifference(cameraPitch(photo), pitch) * weights.pitch;

  return positionScore + headingScore + pitchScore;
};

const bestPhoto = (center, heading, pitch, weights) => (photoA, photoB) => {
  const scoreA = photoScore(photoA, center, heading, pitch, weights);
  const scoreB = photoScore(photoB, center, heading, pitch, weights);

  return scoreB - scoreA;
};

export const sortByBestForThumbnailList = (photos, center, heading) => {
  const weights = {
    position: -1 / 100, // -1 point for every 100m from perfectly centered
    heading: -1 / 22.5, // -1 point for every 22.5 degrees of heading deviation
    pitch: 1 / 45 // +1 point for every 45 degrees of pitch deviation
  };

  const pitch = 90; // camera pointing down, more deviation from this is better

  // using Array.from() to prevent modifying the original array
  return Array.from(photos).sort(bestPhoto(center, heading, pitch, weights));
};

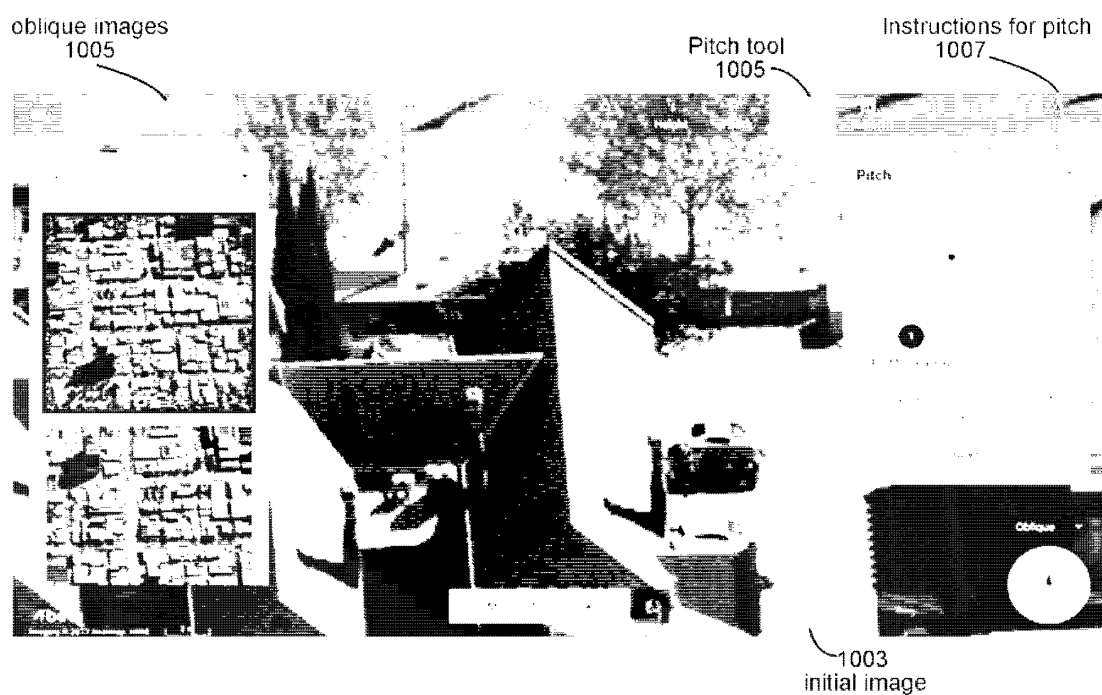
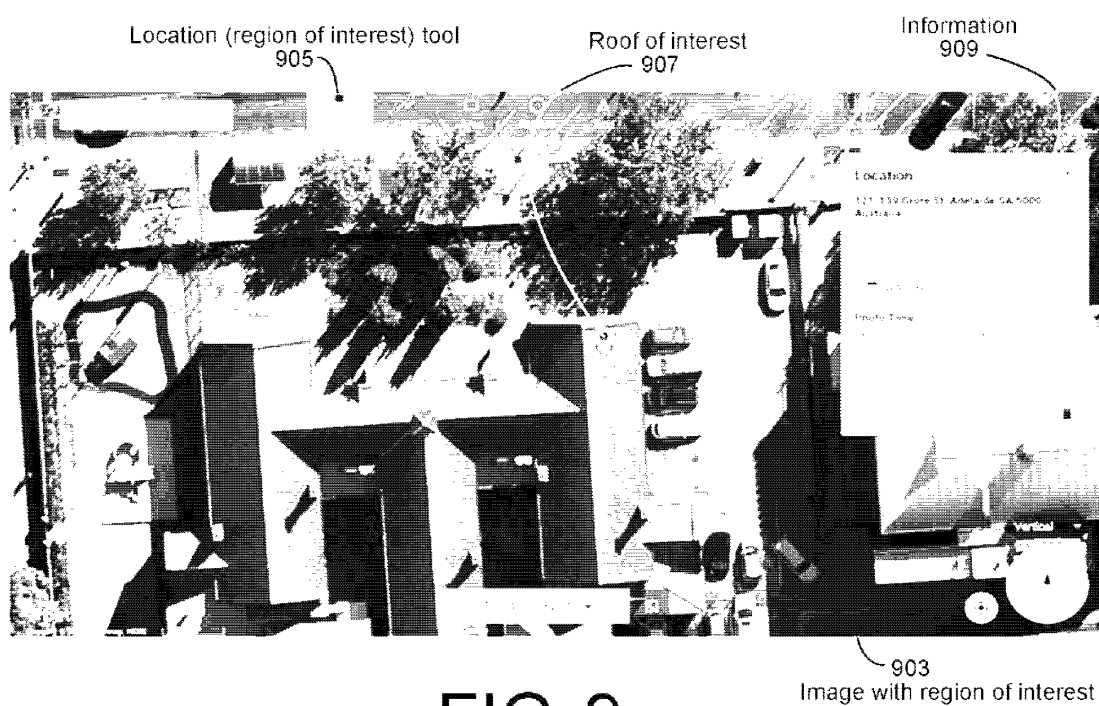
export const sortByBestForRefinement = (photos, center, heading, pitch) => {
  const weights = {
    position: -1 / 200, // 1 point for every 200 meters from perfectly centered
    heading: -1 / 22.5, // -1 point for every 22.5 degrees of heading deviation
    pitch: 1 / 22.5 // +1 point for every 22.5 degrees of pitch deviation
  };

  // Using Array.from() to prevent modifying the original array
  return Array.from(photos).sort(
    bestPhoto(center, heading, pitch, weights)
  );
};

```

FIG. 8

6/9



7/9

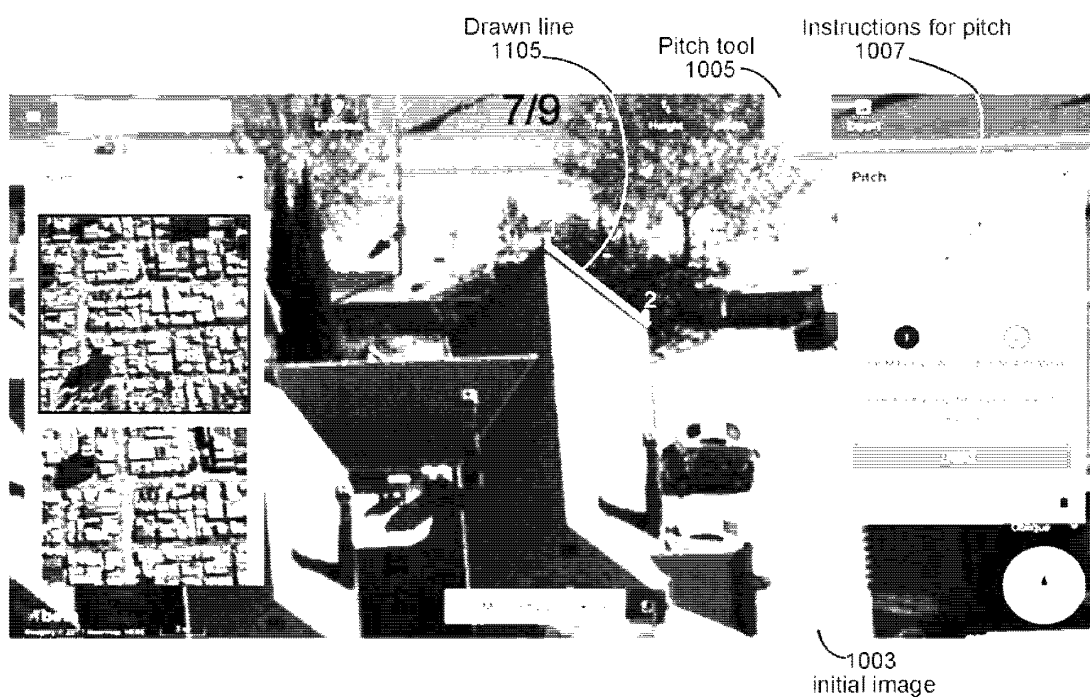


FIG. 11

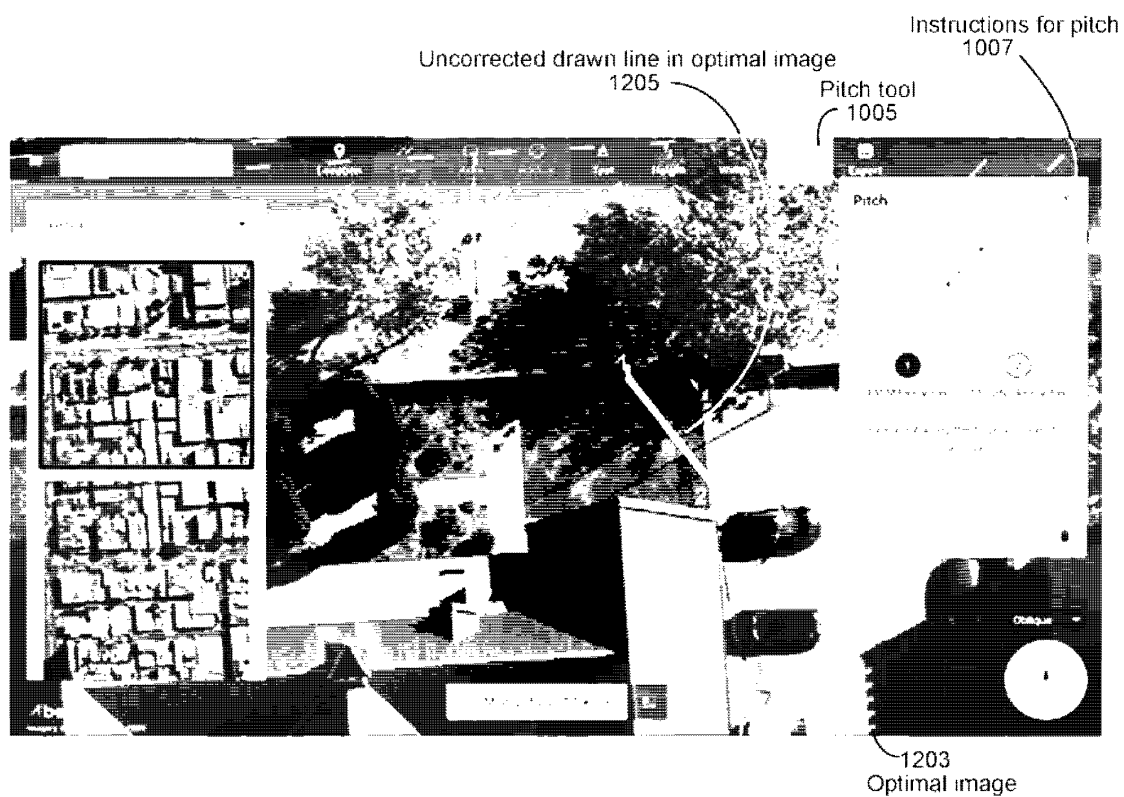


FIG. 12

8/9

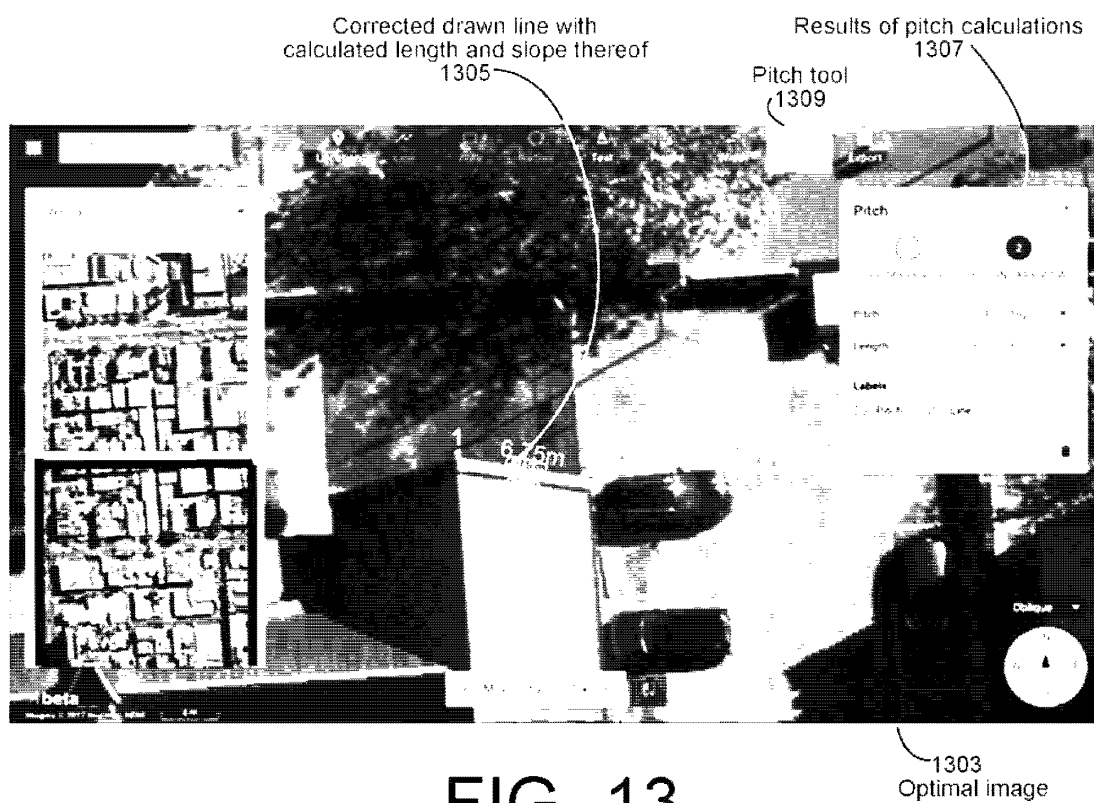


FIG. 13

9/9

