An Annotation Tool for Digital Restoration of Wall Paintings

A. Barreiro Díaz\textsuperscript{1,}*, I. Munoz-Pandiella\textsuperscript{1,2,}*, C. Bosch\textsuperscript{3}, C. Andujar\textsuperscript{1}\* \\
\textsuperscript{1}Universitat Politècnica de Catalunya, Spain \\
\textsuperscript{2}Universitat de Barcelona, Spain \\
\textsuperscript{3}Universitat de Vic - Universitat Central de Catalunya, Spain

Abstract
Antique paintings are essential to study and understand our past. Paintings, and specifically mural paintings, are delicate artworks that are affected by multiple deterioration conditions. Weathering and human interventions cause different damage problems, and physical and chemical changes degrade their visual color appearance. As a consequence, art historians and archaeologists require a huge effort to attempt to rebuild their original appearance. The annotation of digital images of the paintings is a valuable tool in this process. In this paper we analyze major requirements from art historians concerning the annotation of painting regions from the point of view of digital restoration. We also describe a tool prototype (based on TagLab) intended to facilitate the annotation and segmentation of mural paintings. The tool assists art historians in formulating multiple hypotheses on the original appearance by supporting multiple annotation layers for degradation and color, providing both hand-drawn and semi-automatic segmentation, and offering web-based dissemination and sharing of the annotations through the W3C Web Annotation Data Model.

CCS Concepts
\begin{itemize}
  \item Applied computing \rightarrow Fine arts; \item Computing methodologies \rightarrow Image processing;
\end{itemize}

1. Introduction and Previous Work
Antique paintings are a major component of our cultural heritage. Besides their intrinsic aesthetic qualities to express ideas and emotions, antique paintings play an essential role in helping scholars to understand our past. Unfortunately, antique paintings suffer a degradation process over the years that hinder their study and prevent viewers to fully appreciate the artworks.

Mural paintings are affected by multiple deterioration processes \cite{ICO15, SMFPSu20}: humidity, weathering, ambient conditions, structural problems and anthropogenic interventions are some of the factors affecting mural paintings. These factors result in degradation, flaking and cracking, and thus the loss of artistic content. Even in well-conserved regions, the passage of time produces physical and chemical changes that modify the visual appearance of the colors. As a consequence, the colors seen by contemporary visitors might depart significantly from the color palette used by the artists, and reconstructed tonal ranges are often speculative.

Virtual restoration aims at recovering the original state of a digitized artifact or art work, as it would appear after a real, physical restoration \cite{Lim17}. Many authors have focused on the restoration of paintings, either by removing the effects of color degradation \cite{PP00, MAM21}, cracks \cite{GP98}, or detached and lost parts \cite{ODH12}. Complex interventions, however, still often rely on manual work \cite{Lim17, LP20}. Prior to restoration, a diagnostic stage is often necessary to analyse the state of conservation of the painting and its constituent materials. In this matter, multispectral analysis has become a common practice in order to identify pigments and degradation patterns \cite{ODH12}. From this data, a common mapping stage is typically done by experts to annotate damaged regions and identify painting regions, so there is a need of appropriate tools to facilitate this task \cite{PGF18}.

The more content is missing, the more critical is to facilitate archaeologists and art historians to formulate and share different hypotheses on the missing parts. Such hypotheses cover a large range of topics, such as the original color palette, the shapes, tones, and textures in the missing parts, as well as the resources used by the artists; furthermore, such annotations must be linked to the different pictorial phases and restoration campaigns.

All this knowledge can greatly benefit from digital image annotation tools, and thus software designed specifically for Cultural Heritage often allows users to annotate the models. Examples include 3DHOP \cite{PCS18}, Cher-ob \cite{WSA18} and Aioli \cite{CCDL20}. Ponchio et al. \cite{PCDS20} and Croce et al. \cite{CCDL20} review several methods for 2D/3D annotations over CH models. Most approaches focus either on how hand-drawn annotations are entered and presented, or provide fully-automated solutions for specific problems.

In this paper we present a prototype tool to assist art historians in the process of annotating mural paintings. Our tool fills the gap between automatic tools for specific problems (crack detection, segmentation, inpainting) and manual (hand-drawn) techniques. The
3. Proposed solution

Structuring annotation projects Figure 3 illustrates the structure of a project using a representative example. A project (e.g. a particular wall within a monument) consists of a collection of maps. Maps have associated an RGB image (e.g. a photo of the painting) and optionally a depth map. Users can add an arbitrary number of annotation layers to a map: degradation, presumed color at a specific decoration phase, artistic resources, and so on. Degradation and color layers have dictionary-based labels that are part of the project configuration, although users can create and edit labels through the application GUI. Each annotation layer includes an arbitrary number of regions (defined by one outer contour and zero or more inner contours) that can be assigned a label and different user-defined attributes. The organization of the annotations into different layers is key for avoiding clutter and to create independent annotations when studying different problems such as color or degradation.

Easy sharing A crucial need of art historians is to share easily the annotated images with colleagues. Ideally, other experts should be able to inspect and edit the annotations in commodity hardware (PC, tablets), using a web-based solution.

Table 1 summarizes and compares the needs described above in terms of the complexity of the image regions to be annotated, the required labels, their visual representation, and typical region attributes.

3. Proposed solution

Structuring annotation projects Figure 3 illustrates the structure of a project using a representative example. A project (e.g. a particular wall within a monument) consists of a collection of maps. Maps have associated an RGB image (e.g. a photo of the painting) and optionally a depth map. Users can add an arbitrary number of annotation layers to a map: degradation, presumed color at a specific decoration phase, artistic resources, and so on. Degradation and color layers have dictionary-based labels that are part of the project configuration, although users can create and edit labels through the application GUI. Each annotation layer includes an arbitrary number of regions (defined by one outer contour and zero or more inner contours) that can be assigned a label and different user-defined attributes. The organization of the annotations into different layers is key for avoiding clutter and to create independent annotations when studying different problems such as color or decay. The proposed structure is quite similar to that of [PGF20]; the most notable differences are the fact that we require multiple annotation layers (each one with its own label dictionary) and therefore a different default dictionary is required for each type of annotation layer (e.g. degradation labels, color labels, artistic resource labels).
Intended workflow Figure 2 shows one suitable workflow for the annotation of mural paintings. A first step is to define the default dictionaries containing the labels for the different types of annotation layers. For example, the degradation dictionary might include the labels non-degraded, degraded, and support, whereas the color dictionary would include names for the different colors in a mural painting. After this step, art historians are expected to create a new project for each part of the monument, add one or more maps (photos) with the desired annotation layers, and then populate them by selecting regions using a variety of tools (hand-free drawing, border editing, and other selection tools typically found in image processing applications). Each selected region is then associated to a label and text notes and attributes. Annotation layers can be exported using the W3C Web Annotation Model, which is supported by a number of JS libraries.

Implementation details: creating annotations None of the annotation platforms we analysed completely fulfilled the requirements in Section 2. We thus opted to (a) adapt TagLab [PGF+20] to suit our needs, and (b) use the W3C Web Annotation Model together with a simple HTML+JS application for sharing the annotations. Most of the functionality we required for the annotation was already available in TagLab [PGF+20], a segmentation application currently focused on marine corals. TagLab uses a project structure quite close to that required by art historians, and provides a collection of selection tools (some of them relying on DeepLab V3+ [Jfz19] segmentation network) that greatly facilitate the selection of regions on the paintings. Furthermore, TagLab allows users to train a segmentation network from an example segmentation of an image. The resulting model can be used to provide an initial automatic segmentation for similar images. We adapted TagLab interface to the particular task of annotating mural paintings. Major changes included the addition of multiple types of annotation layers (e.g. degradation layer, color layer), the addition of specific default dictionaries for the corresponding labels, and a new GUI to select a subset of the labels in the default dictionaries. We also added a per-region confidence value. This field is user-adjustable and it is also automatically filled from the probability map provided by DeepLab V3+ [Jfz19], in the case of automatic segmentations. Figure 5 shows a general view of the adapted interface, and Figure 4 a couple of dialogues.

Implementation details: sharing annotations Unfortunately, TagLab requires a Python kernel and multiple dependencies (e.g. PyTorch) which might not be suitable for easy sharing of the annotations to other art historians. We addressed this issue by exporting the annotations in a JSON file following the W3C Web Annotation Model. The model support SVG-based polygonal contours as well as varied text-based annotations. The resulting JSON files can be inspected using a simple web-based application based on the JS library Annotorious, which allows importing, rendering and editing the annotations (both the contours and the labels).
We would also like to thank Begoña Cayuela (Tupsar) for her contribution, and suitability of the tool, from an art history perspective.

Acknowledgements
We would like to thank Roser Piñol (UB) for her helpful comments on the requirements, user interface adaptation, and suitability of the tool, from an art history perspective. We would also like to thank Begoña Cayuela (Tupsar) for her suggestion on integrating the annotation tool with the W3C standard annotation model. This work has been partially supported by the projects TIN2017-88515-C2-1-R/2-R funded by MCIN/AEI/10.13039/501100011033 / FEDER “A way to make Europe”, by the EU Horizon 2020, JPICH Conservation, Protection and Use initiative (JPICH-0127) and the Spanish Agencia Estatal de Investigación (grant PCI2020-111979).

References


