The “Castelet”: a Dynamically Reconfigurable Stage for Performing Arts

1 Introduction

The hero jumped down the wall, escalated the hill, slid down the rock, climbed the stairs and finally left the villain behind...In this classical scene, the hero has to move on different types of terrain. Recreating such a varied spectrum of terrains in a theater scene still remains a challenge.

Although the use of technology is widespread in modern performing arts such as dance, stage performances, music concerts, and circus shows, it has mainly focused on visual, lighting, and sound effects. However, the field of stage machinery has been slow to adopt advanced technology. Stage setups are often custom built for a single use and offer few degrees of freedom (DOFs), mostly rotations and vertical/horizontal displacements. In addition, these setups are difficult to scale, cannot be moved easily between sites, and are often very expensive. Ideally, modern stage machinery should offer many DOFs, be modular, portable, and dynamically reconfigurable.

This paper presents the “Castelet”, a dynamically reconfigurable stage for performing arts. The “Castelet” is a modular and portable stage machinery composed of simple building blocks which, when combined, provide many DOFs and allow the implementation of a rich variety of complex surfaces which can be modified rapidly and dynamically during a performance. A 1/10 scale prototype of the setup is described in the next section. This prototype cannot only be used as a small-scale model for designing shows but can also be used for performing puppet shows for instance.

2 Basic Concept of the “Castelet”

The basic principle for implementing the “Castelet” is to use an array of polygons - in our case square blocks - that can move up and down. In the implemented prototype, a module consists of a 3 x 3 array of square blocks, and a stage consists of an assembly of modules (see Figure 1(b) and (c)). The motion of the blocks in a module are coupled in order to limit the number of actuators. The idea of using arrays of geometric components (pins, tubes, etc.) has been primarily applied to haptic/tactile devices [Iwata et al., 2001]. An interesting concept based on arrays of tetrahedra for landscape exploration was proposed in [Clark et al. 2004].

The key element of the “Castelet” is the Planar Interpolation Mechanism (PIM) shown in Figure 1(a). The PIM is a parallel mechanism for which the vertical motion of horizontal blocks $P_1$ and $P_3$ is induced by two linear actuators. The central block $P_2$ is passive and its vertical motion results from the motion of the two other blocks and, as its name implies, is an interpolation of their motion. The geometry of the PIM does not guarantee equal height steps $h_1$ and $h_2$, for each value of $\theta$. This is depicted in Figure 1(d) which shows the error $\Delta$ between the actual position of $F$ compared to its ideal position given by

$$\rho_1 + \rho_2 \over 2$$  

as a function of $\theta$. The parameters governing the geometry of the PIM can be chosen so as to minimize $\Delta$ and to guarantee a null interpolation error when the module is at rest or when it is fully deployed (respectively $\theta_0$ and $\theta_1$, $\theta_2$ in Figure 1(d)). Combining 6 PIMS orthogonally (e.g. along each line and column of a 3 x 3 matrix) with four actuators located at each corner results in a 3 x 3 module with 4 DOFs like the one shown in Figure 1(b). A module is also equipped with 36 light emitting diodes (LEDs) (4 per block) for combining lighting patterns to motion effects of a stage.

The motion and lighting of a stage built from several modules is controlled by a user interface on a PC which allows the configuration of a stage, the selection of pre-defined dynamic stage motion patterns (such as slope, sinewave, staircase, etc.) and lighting effects or program custom patterns.

The motion pattern selected/programmed for a stage is sent to custom electronic boards responsible for controlling the actuators on the modules. A board on a module is composed of a DSP processor and firmware for implementing the control algorithms for the actuators and LEDs as well as a CANbus network interface for managing the communication between the module and the PC.

3 References
